



Analisis Bisnis secara Real Time dalam Transformasi Digital

Dr. Budi Raharjo, S.Kom., M.Kom., MM.



YAYASAN PRIMA AGUS TEKNIK

Dr. Budi Raharjo, S.Kom., M.Kom., MM.

Analisis Bisnis secara Real Time dalam Transformasi Digital



YAYASAN PRIMA AGUS TEKNIK

YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-623-8120-81-9 (PDF)



9 786238 120819

Analisis Bisnis secara Real Time dalam Transformasi Digital

Penulis :

Dr. Budi Raharjo, S.Kom., M.Kom., MM.

ISBN : 9 786238 120819

Editor :

Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

Penyunting :

Dr. Joseph Teguh Santoso, M.Kom.

Desain Sampul dan Tata Letak :

Irdha Yuniyanto, S.Ds., M.Kom.

Penebit :

Yayasan Prima Agus Teknik Bekerja sama dengan
Universitas Sains & Teknologi Komputer (Universitas STEKOM)

Anggota IKAPI No: 279 / ALB / JTE / 2023

Redaksi :

Jl. Majapahit no 605 Semarang

Telp. (024) 6723456

Fax. 024-6710144

Email : penerbit_ypat@stekom.ac.id

Distributor Tunggal :

Universitas STEKOM

Jl. Majapahit no 605 Semarang

Telp. (024) 6723456

Fax. 024-6710144

Email : info@stekom.ac.id

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara apapun tanpa ijin dari penulis

KATA PENGANTAR

Segala puji bagi Allah, Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan buku ajar yang berjudul ***“Analisis Bisnis secara Real Time dalam Transformasi Digital”***. Tujuan dari penulisan buku ini tidak lain adalah untuk membantu para mahasiswa di dalam memahami seluruh lanskap perangkat, sensor, manajemen perangkat, analisis real-time, dan analisis prediktif. Buku ini menjelaskan bagaimana dunia usaha dapat memanfaatkan platform untuk mewujudkan transformasi digital mereka. Dengan demikian, hal ini memberikan pengetahuan dan metodologi terkini bagi para pendidik pendidikan tinggi, peneliti, mahasiswa, dan pemangku kepentingan industri.

Microsoft Azure Internet of Things (IoT) adalah kumpulan layanan cloud, komponen edge, dan SDK yang dikelola Microsoft yang memungkinkan pengguna untuk menyambungkan, memantau, dan mengontrol aset IoT Anda dalam skala besar. Dalam istilah yang lebih sederhana, solusi IoT terdiri dari perangkat IoT yang berkomunikasi dengan layanan cloud. SDK Perangkat IoT dan IoT Hub mendukung protokol komunikasi umum seperti HTTP, MQTT, dan AMQP untuk komunikasi perangkat-ke-cloud dan cloud-ke-perangkat. Dengan Azure IoT Edge pengguna dapat memindahkan sebagian beban kerja IoT dari layanan cloud Azure ke perangkat pengguna. IoT Edge dapat mengurangi latensi dalam Solusi penggunaannya, mengurangi jumlah data yang dipertukarkan perangkat Anda dengan cloud, dan mengaktifkan skenario offline. Azure Sphere adalah platform aplikasi tingkat tinggi yang aman dengan fitur komunikasi dan keamanan bawaan untuk perangkat yang tersambung ke internet. Ini mencakup unit mikrokontroler aman, sistem operasi berbasis Linux kustom, dan layanan keamanan berbasis cloud yang menyediakan keamanan berkelanjutan dan terbaru.

Buku ini dibagi menjadi 11 Bab, pada bab 1 “Bisnis Secara Real-Time” Bab ini memberikan konteks bisnis untuk topik teknis yang dibahas dalam buku ini dan ada pendekatan logis dan peta jalan teknologi yang akan menunjukkan jalan menuju kesuksesan transformasi menuju bisnis real-time. Pendekatan berprinsip dan arsitektur referensi diperkenalkan yang memberikan peta jalan tentang cara merancang dan mengimplementasikan IoT yang sangat skalabel, aman, dan solusi SaaS analitik tingkat lanjut di Azure. Bab 2 “DevOps Menggunakan PowerShell, ARM, dan VSTS”, Bab ini merinci bagaimana pembaca dapat menggunakan Azure PowerShell, templat Azure Resource Manager (ARM), dan Visual Studio Team Services untuk mengotomatiskan langkah-langkah penyediaan, pembuatan, dan penerapan layanan yang dihosting Azure. Dan pada bab 3 “Manajemen Perangkat Menggunakan IoT Hub”, Bab ini membahas fitur Manajemen Perangkat Azure IoT Hub yang mendukung perintah dan kontrol, kembaran perangkat, dan metode langsung.

Bab 4 “Sensor, Perangkat, dan Gerbang”, Bab ini memberikan gambaran sekilas tentang dunia sensor dan perangkat. Ini menyentuh beberapa skenario sensor dan perangkat yang lebih

umum yang akan pembaca temui dan bagaimana mereka berhubungan dan bekerja sama untuk menciptakan jaringan yang konsisten dan andal dari berbagai hal yang terhubung. Pola implementasi firmware perangkat tercakup. Bab 5 “pemrosesan Real-Time Menggunakan Azure Stream Analytics”, bab ini, kami memeriksa penggunaan Microsoft Azure Streaming Analytics untuk menciptakan pekerjaan guna memproses aliran data masuk dari berbagai sensor kami, melakukan transformasi dan pengayaan data, dan terakhir, memberikan hasil keluaran dalam berbagai format data. Dan bab 6 “pemrosesan batch dengan data factory dan data lake store”, dalam bab ini, kami mengkaji penggunaan Azure Data Factory dan Azure Data Lake serta di mana, mengapa, dan bagaimana teknologi ini sesuai dengan kemampuan bisnis modern yang berjalan dengan kecepatan Internet. Bab 7 “analisis Tingkat lanjut dengan azure data lake analytics”, dalam bab ini, kita akan mempelajari penggunaan Azure Data Lake Analytics (ADLA), yang merupakan perangkat “big data” baru Microsoft yang berjalan di atas Azure Data Lake.

Bab 8 “analisis Tingkat lanjut menggunakan pembelajaran mesin dan R”, dalam bab ini, kita menjelajahi dunia baru pembelajaran mesin dan analisis prediktif yang menarik menggunakan Azure Machine Learning dan bahasa pemrograman R. Bab 9 visualisasi data, peringatan, dan pemberitahuan dengan power BI”, dalam bab ini, kami mengeksplorasi penggunaan visualisasi data, peringatan, dan notifikasi untuk membantu bisnis modern saat ini menyediakan komunikasi yang berguna bagi karyawan dan pelanggan agar berhasil mengelola operasi mereka secara real-time. Bab 10 “keamanan dan identitas”, bab ini menguraikan kerangka kerja untuk menganalisis potensi vektor ancaman serta alat dan protokol yang disediakan Azure untuk memitigasi ancaman ini. Kami juga memeriksa cara mengimplementasikan aplikasi multi-penyewa menggunakan Azure Active Directory B2C. Bab 11 yang merupakan menjadi bab terakhir dalam buku ini “epilog”, dalam bab ini penulis merefleksikan setiap topik yang dibahas dalam buku ini dan memberikan beberapa saran tentang bagaimana pembaca dapat memulai transformasi digital pembaca.

Buku ini menjelaskan bagaimana dunia usaha dapat memanfaatkan platform kami untuk mewujudkan transformasi digital mereka. Buku ini menjelaskan bagaimana dunia usaha dapat memanfaatkan platform kami untuk mewujudkan transformasi digital mereka. Akhir kata semoga buku ini bermanfaat bagi para pembaca.

Semarang, Januari 2024

Penulis

Dr. Budi Raharjo S.Kom., M.Kom., M.M.

DAFTAR ISI

Halaman Judul	i
Kata Pengantar	ii
Daftar Isi	iv
BAB 1 BISNIS SECARA REAL-TIME	1
1.1. Pendekatan Platfrom	1
1.2. Platfrom Bisnis Waktu Nyata	2
1.3. Internet Of Things (IoT) dan Data Besar	6
1.4. Arsitektur Referensi Bisnis Real-Time	8
1.5. Referensi Dan Implementasi Kesehatan Dan Keselamatan Pekerja	16
1.6. Ringkasan.....	19
BAB 2 DEVOPS MENGGUNAKAN POWERSHELL, ARM, dan VSTS	20
2.1. Pendahuluan	20
2.2. DevOps and Azure	22
2.3. Manajer Sumber Daya Azure	27
2.4. Membuar Proyek VSTS	33
2.5. Menciptakan Definisi Otomatisasi	40
2.6. Ketentuan Menggunakan Azure Classic Comands	50
2.7. Ringkasan	78
BAB 3 MANAJEMEN PERANGKAT MENGGUNAKAN IOT HUB	79
3.1. Siklus Hidup Manajemen Perangkat	79
3.2. Pusat IoT Azure	82
3.3. Konfigurasi dan Pemantauan	86
3.4. Dashbor Manajemen Perangkat	95
3.5. Ringkasan	108
BAB 4 SENSOR, PERANGKAT, DAN GERBANG	109
4.1. Sensor	109
4.2. Perangkat	112
4.3. Buat Perangkat Cerdas	122
4.4. Modifikasi Simulator Tim	141
4.5. Ringkasan	143
BAB 5 PEMROSESAN REAL-TIME MENGGUNAKAN AZURE STREAM ANALYTICS	144
5.1. Arsitektur Lambada	144
5.2. Analisis Real - Time	146
5.3. Memprediksi Hasil Untuk Keunggulan Kompetitif	148
5.4. Memilih Mesin Analisis Streaming Terkelola Di Azure	150
5.5. Implemenetasi Referensi: Azure Streaming Analytics	152
5.6. Merencanakan Keluaran Analaisis Streaming	158
5.7. Konversi Tipe Data	161

5.8.	Konfigurasi Definisi Fungsi Analisis Streaming	174
5.9.	Query SQL Pekerjaan Analisis Streaming	187
5.10.	Ringkasan	192
BAB 6	PEMROSESAN BATCH DENGAN DATA FACTORY Dan DATA LAKE STORE	193
6.1.	Ikhtisar Pabrik Data Azure	193
6.2.	Definisi JSON	198
6.3.	Penyimpanan Azure Data Lake	206
6.4.	Perbarui File Input Data Referensi Untuk Azure Stream Analytics	209
6.5.	Aktivitas Sumber Daya Pembaruan Azure ML Data Factory	223
6.6.	Pindahkan Data Dari Penyimpanan Blob Ke Data Lake	236
6.7.	Ringkasan	246
BAB 7	ANALISIS TINGKAT LANJUT DENGAN AZURE DATA LAKE ANALYTICS	247
7.1.	Analisis Azure Data Lake	248
7.2.	Alat Azure Data Lake Untuk Visual Studio	258
7.3.	Fitur Adla, Mengkueri Data Di Penyimpanan	263
7.4.	Manajemen Dan Administrasi Yang Disederhanakan	270
7.5.	Gambaran Umum Tugas Yang Harus Diselesaikan	273
7.6.	Pengujian Csv Azure Data Lake Baru Untuk Tiga File Input	275
7.7.	Ringkasan	298
BAB 8	ANALISIS TINGKAT LANJUT MENGGUNAKAN PEMBELAJARAN MESIN DAN R 300	300
8.1.	Apa Itu Pembelajaran Mesin?	300
8.2.	Pembelajaran Mesin Azure.....	306
8.3.	Ikhtisar Server Microsoft R	312
8.4.	Perluas Eksperimen Pembelajaran Mesin Dengan Modul Bahasa R	314
8.5.	Ringkasan	341
BAB 9	VISUALISASI DATA, PERINGATAN, DAN NOTIFIKASI DENGAN POWER BI	342
9.1.	Lanskap pelaporan modern	342
9.2.	Ihtisar Power BI	344
9.3.	Sumber data Power BI	348
9.4.	Arsitektur Biaya Pelaporan Cloud	353
9.5.	Fungsi Azure	358
9.6.	Modifikasi Query SQL Streaming Analytics	363
9.7.	Visualisasi Pembacaan Sensor Menggunakan R Dan Power BI Desktop	370
9.8.	Membangun Visualisasi Power BI Untuk Prediksi Pembelajaran Mesin	380
9.9.	Fungsi Azure Untuk Peringatan	385
9.10.	Ringkasan	405
BAB 10	KEAMANAN DAN IDENTITAS	407
10.1.	Pemodean Ancaman	407
10.2.	Protokol Keamanan	411
10.3.	Identitas	413
10.4.	Pelatihan Membuat Penyewa Azure B2C	418

10.5. Perbarui Aplikasi Untuk Menghubungi API Platform	436
BAB 11 EPILOG	444
Daftar Pustaka	446

BAB 1

BISNIS SECARA REAL-TIME

Setiap bisnis saat ini sedang mengalami transformasi digital akibat kekuatan disruptif di pasar, mulai dari pesaing yang lahir di cloud hingga meningkatnya tuntutan pelanggan, mitra, dan karyawan untuk terlibat melalui pengalaman digital modern. Mereka berevolusi dari hanya mengandalkan data historis menjadi belajar menggunakan data historis dan real-time untuk mendorong inovasi, mengembangkan strategi bisnis, dan mengotomatisasi proses bisnis penting.

Ketika bisnis berkembang dan bertransformasi untuk memanfaatkan data real-time, mereka akan memberikan dampak melalui efisiensi operasional serta menciptakan peluang pendapatan baru. Misalnya, produsen produk dapat mengumpulkan informasi tentang kinerja produk mereka di lokasi terpencil dan mengotomatiskan penjadwalan teknisi layanan lapangan hanya jika diperlukan. Gerai ritel dapat menyediakan inventaris real-time untuk mendorong pengalaman belanja Omni-channel bagi pelanggannya. Perusahaan yang mempunyai kebutuhan untuk meningkatkan keselamatan pekerja dapat melacak kondisi lingkungan seperti suhu, kelembaban, dan kecepatan angin serta biometrik karyawan seperti detak jantung, suhu tubuh, dan laju pernapasan untuk dapat menentukan apakah kondisi fisik karyawan akan menimbulkan dampak buruk. masalah keselamatan pekerja.

Dalam setiap skenario ini, perusahaan menyadari bahwa untuk tetap kompetitif, meningkatkan efisiensi operasional, dan melibatkan pelanggan lebih dalam, mereka harus belajar memanfaatkan pola dan praktik pengembangan perangkat lunak modern. Mereka bertransformasi menjadi penyedia Software-as-a-Service (SaaS) yang ahli dalam ilmu data dan Internet of Things (IoT). Mereka belajar membuat aplikasi yang menghubungkan orang, tempat, dan benda. Mereka menyediakan visualisasi data, peringatan, dan pemberitahuan secara real-time. Mereka mengintegrasikan produk-produk yang terhubung ini dengan sistem lini bisnis yang ada dan menyediakan otentikasi tanpa batas bagi pelanggan, mitra, dan karyawan melalui pengalaman mendalam dan indah yang dapat digunakan di perangkat apa pun dan tersedia 24/7. Mereka bertransformasi menjadi Bisnis Real-Time.

1.1 PENDEKATAN PLATFORM

Untuk memaksimalkan investasi dalam transformasi ke Bisnis Real-Time, perlu untuk meningkatkan penggunaan kembali kemampuan bisnis dan teknologi umum. Hal ini paling efektif dicapai dengan memanfaatkan pendekatan platform. Pendekatan platform berupaya menciptakan serangkaian kemampuan mendasar yang diakses melalui API terkelola sehingga banyak aplikasi vertikal dapat dibangun pada substrat umum ini, seperti yang digambarkan pada Gambar 1.1.

Terlepas dari industri vertikal, ada serangkaian komponen umum yang dapat digunakan kembali yang Anda perlukan untuk mengoperasikan solusi IoT. Setelah kemampuan tersebut diterapkan, sebuah bisnis dapat membangun banyak aplikasi yang menargetkan berbagai jenis pengguna dan pasar vertikal yang berbeda. Salah satu manfaat besar dari pendekatan ini adalah bahwa semua data dari semua pasar vertikal tersebut dimasukkan ke dalam sub-sistem analitik tingkat lanjut yang umum di mana wawasan lintas bisnis dapat diambil. Wawasan ini menghasilkan apa yang kami sebut aplikasi X-Factor, yaitu peluang pendapatan baru yang tidak dapat ditemukan karena sifat model bisnis sebelumnya yang tertutup.



Gambar 1.1 IoT dan model Platform Analisis Tingkat Lanjut

1.2 PLATFORM BISNIS WAKTU NYATA

Transformasi ke Bisnis Real-Time membutuhkan investasi pada sumber daya manusia, proses, dan alat. Untuk memenuhi harapan pelanggan Anda dan menyediakan platform bagi bisnis untuk meningkatkan dampak dengan cepat, diperlukan pendekatan baru dalam desain, pengembangan, dan penerapan produk perangkat lunak Anda. Solusi Bisnis Real-Time adalah aplikasi SaaS yang mendukung siklus rilis yang sering, bekerja pada perangkat apa pun, dan menyediakan skema otentikasi multi-penyewa serta akses aman ke informasi mendasar bagi pelanggan, mitra, dan karyawan.

Bisnis Real-Time menyiratkan bahwa Anda mengadopsi sensor, suar, dan teknologi perangkat pintar terbaru untuk menghubungkan produk, lingkungan fisik, dan orang-orang untuk menghasilkan data real-time. Hal ini menyiratkan bahwa Anda memanfaatkan teknik analitik tingkat lanjut seperti pemrosesan aliran, pengurangan peta, dan pembelajaran mesin untuk melakukan analisis real-time pada aliran data konstan yang mengalir ke sistem. Melalui analisis tersebut, Anda akan dapat memberikan peringatan dan pemberitahuan tingkat visual dan sistem serta visualisasi data bernilai tinggi bagi konsumen informasi. Selain itu, Anda akan dapat berintegrasi dengan lini sistem bisnis yang ada untuk mengotomatisasi proses bisnis penting.

Seperti digambarkan pada Gambar 1-2, agar sukses dalam perjalanan ini, organisasi harus menerapkan serangkaian prinsip teguh yang menjadi pedoman dan aturan yang digunakan untuk memengaruhi setiap keputusan produk dalam prosesnya.



Gambar 1.2. Metodologi, Proses, Arsitektur, Platform

Prinsip-prinsip ini mencakup metodologi pengembangan perangkat lunak yang terdefinisi dengan baik, proses pengembangan perangkat lunak standar yang mencakup tidak hanya pengembangan tetapi juga operasi, penerapan pola arsitektur perangkat lunak yang memberikan skalabilitas, elastisitas, dan ketangkasan, dan platform perangkat lunak yang menyediakan layanan dasar untuk perangkat lunak modern.

Metodologi: Rekayasa Lean

Lean engineering berakar pada bidang manufaktur yang konsep utamanya adalah memaksimalkan nilai pelanggan sekaligus meminimalkan pemborosan, yaitu menciptakan nilai lebih bagi pelanggan dengan sumber daya yang lebih sedikit. Diterapkan pada siklus hidup pengembangan perangkat lunak, tim produk melibatkan pelanggan di awal proses pengembangan dengan mengoperasionalkan produk yang layak minimal (MVP) dan meminta umpan balik. Dengan menggunakan pendekatan ini, tim produk perangkat lunak dapat lebih mudah menyesuaikan atau bahkan mengoreksi arah sehingga meningkatkan kualitas produk. Siklus hidup lean engineering, seperti yang digambarkan pada Gambar 1.3, disebut Build-Measure-Learn dan mendorong penyampaian berkelanjutan, analisis berkelanjutan, dan umpan balik berkelanjutan.

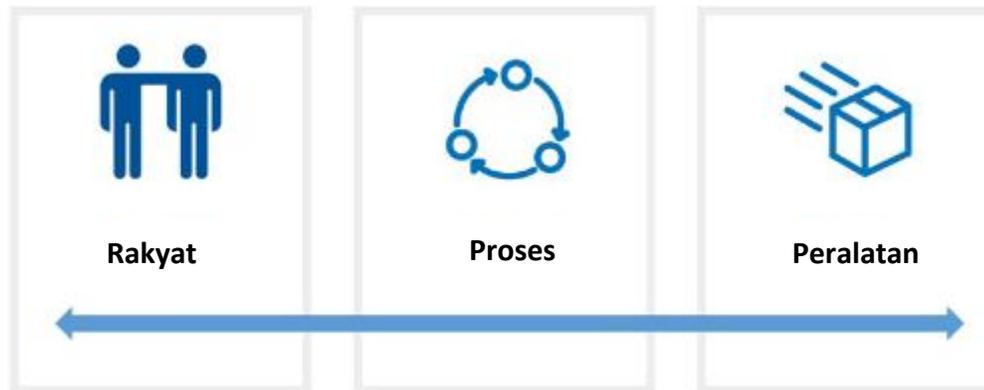


Gambar 1.3. Siklus rekayasa lean

Fase Build (membangun) mewakili aktivitas pengembangan dan penerapan; fase Pengukuran berfokus pada pemantauan dan pelaporan kesehatan perangkat lunak; Pembelajaran adalah tentang melibatkan pelanggan untuk mengumpulkan umpan balik yang kemudian digunakan untuk mendorong siklus hidup pengembangan produk berikutnya. Pembuatan dasbor, baik yang dibuat khusus atau disediakan oleh alat pihak ketiga, memberikan analisis real-time dan historis yang dapat digunakan untuk memperoleh wawasan dengan cepat dan mengarahkan upaya pengembangan produk ke arah yang memenuhi kebutuhan pelanggan Anda.

Proses: DevOps

Istilah DevOps adalah gabungan antara pengembangan dan operasi. Mashup menyiratkan hubungan yang lebih dalam dan kolaboratif antara organisasi pengembangan dan operasi. Tujuan dari kolaborasi tersebut adalah untuk menentukan bagaimana orang, proses, dan alat digabungkan untuk mengotomatisasi siklus hidup pengembangan perangkat lunak. DevOps, seperti yang digambarkan pada Gambar 1.4, menyiratkan penciptaan tim lintas fungsi, menggabungkan pengembang, penguji, dan arsitek bersama dengan operasi yang bersama-sama memiliki seluruh jalur penerapan mulai dari pembangunan, pengujian, pementasan hingga produksi.



Gambar 1-4. Kerangka kerja DevOps

Hal ini mengharuskan tim-tim ini bekerja secara kolaboratif untuk mengadopsi proses dan alat yang umum. Penjelasan sederhana ini memiliki implikasi besar bagi sebuah organisasi. Hal ini tidak terjadi dalam semalam dan harus dilakukan secara bertahap menggunakan tim kecil yang mengadopsi metode baru dan praktik terbaik dan kemudian bertransisi menjadi ahli di bidangnya, mentransfer pengetahuan mereka ke seluruh staf.

Arsitektur: Layanan Mikro

Atribut utama dari layanan mikro adalah:

- **Otonom dan Terisolasi:** Layanan mikro adalah unit mandiri yang secara fungsional memiliki ketergantungan yang longgar pada layanan lain dan dirancang, dikembangkan, diuji, dan dirilis secara independen.
- **Dapat digunakan kembali, Toleran terhadap Kesalahan, dan Responsif:** Layanan mikro harus dapat diskalakan dengan tepat tergantung pada skenario penggunaan. Mereka harus toleran terhadap kesalahan dan memberikan kerangka waktu yang masuk akal untuk pemulihan jika terjadi kesalahan. Terakhir, mereka harus responsif, memberikan kinerja yang wajar berdasarkan skenario eksekusi.
- **Dapat diprogram:** Layanan mikro mengandalkan API dan kontrak data untuk menentukan cara interaksi dengan layanan dicapai. API mendefinisikan sekumpulan titik akhir yang terlihat di jaringan, dan kontrak data menentukan struktur pesan yang dikirim atau dikembalikan.
- **Dapat Dikonfigurasi:** Layanan mikro dapat dikonfigurasi. Agar dapat digunakan kembali dan memenuhi kebutuhan setiap sistem yang memilih untuk menggunakan kemampuannya, layanan mikro harus menyediakan sarana yang dapat disesuaikan dengan skenario penggunaan.
- **Otomatis:** Siklus hidup layanan mikro harus sepenuhnya otomatis, mulai dari desain hingga penerapan.

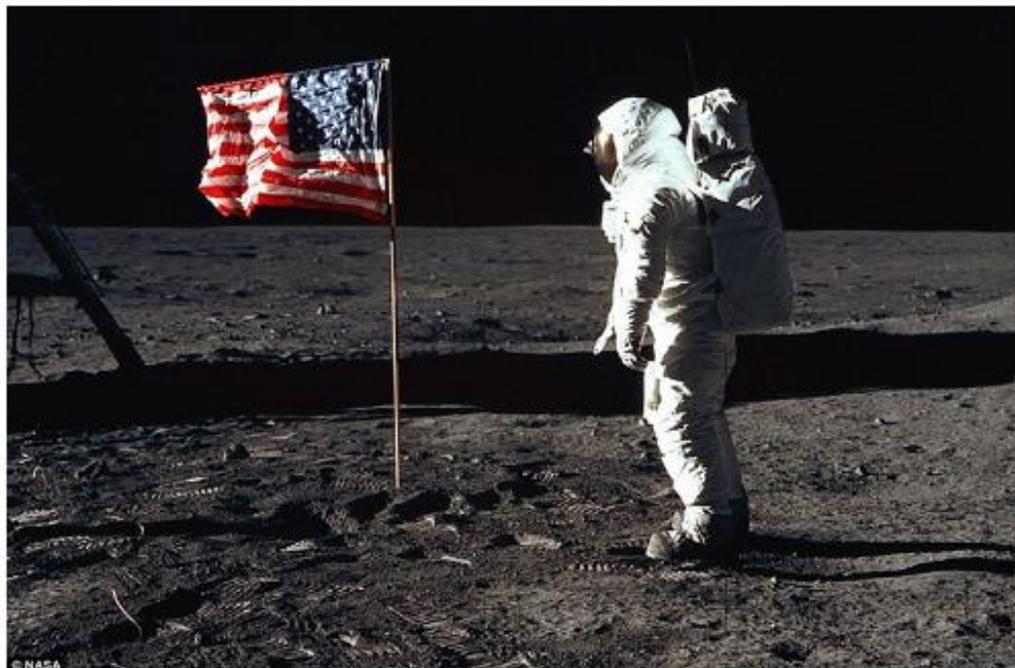
Arsitektur layanan mikro telah menjadi pola populer dalam pengembangan aplikasi SaaS yang sangat skalabel dan toleran terhadap kesalahan karena menyediakan arsitektur aplikasi yang lebih kuat, mendukung siklus rilis berkecepatan tinggi, dan menekankan pada API terkelola sebagai sarana untuk mengakses data yang mendasarinya, analitik, dan operasi bisnis. Salah satu manfaat pendekatan ini adalah API berpotensi menjadi saluran pendapatan baru bagi bisnis.

Peron: Awan

Aplikasi modern memerlukan infrastruktur dan platform perangkat lunak yang dapat menyediakan ketersediaan tinggi, toleransi kesalahan, skala elastis, penyimpanan dan komputasi sesuai permintaan, API, dan alat untuk mengotomatisasi sepenuhnya setiap interaksi dengan platform. Platform cloud juga harus menyediakan layanan dasar yang mencakup IoT, big data, dan analisis prediktif.

1.3 INTERNET OF THINGS (IOT) DAN DATA BESAR

IoT bukanlah hal baru. Kemampuan untuk menghubungkan perangkat ke jaringan, mengumpulkan telemetri, dan menampilkan informasi tersebut untuk mengumpulkan wawasan dan bertindak telah ada selama beberapa waktu. NASA memelopori konsep data yang dikumpulkan oleh sensor dan dikirim melintasi ruang dan waktu untuk dianalisis hampir secara real-time sehingga status dapat divisualisasikan, diperoleh wawasan, dan tindakan diambil dalam keadaan darurat (lihat Gambar 1-5).



Gambar 1.5. Neil Armstrong (kredit gambar NASA)

Pengendali Misi NASA, terlihat pada Gambar 1-6, terdiri dari ratusan orang, masing-masing dengan koleksi monitornya sendiri yang menyediakan visualisasi data metrik utama yang berasal dari modul perintah atau pakaian astronot. Data tersebut merupakan ukuran langsung status misi dan keselamatan. Sungguh luar biasa jika Anda memikirkan pencapaian organisasi tersebut dengan kondisi teknologi pada saat itu.



Gambar 1.6. Kontrol Misi NASA (kredit gambar NASA)

Titik Tipping

Satu-satunya hal yang dimiliki NASA yang membuat mereka unik adalah anggarannya. Miliaran dolar memungkinkan NASA untuk mengirim manusia ke bulan dan, dalam prosesnya, mendefinisikan IoT untuk kita semua.

Kemungkinan besar Anda tidak memiliki anggaran sebesar NASA, namun Anda beruntung. Tidak perlu lagi menghabiskan banyak uang untuk mengaktifkan IoT pada produk Anda dan menghubungkannya ke cloud untuk mengumpulkan telemetri, mengubah dan menyimpan data, mengumpulkan wawasan, dan bertindak. Sensor dan papan mikroprosesor mini tidak mahal dan semakin murah serta bertenaga setiap saat. Kemampuan mengembangkan kode untuk mengumpulkan pembacaan sensor, terhubung ke titik akhir cloud yang aman, dan mengirim pesan tidak pernah semudah ini.

Apa yang mendorong rasa haus akan IoT ini adalah data. Data ini akan mengungkapkan kualitas produk dan cara penggunaannya oleh pelanggan, serta memberi Anda kemampuan untuk menghitung waktu rata-rata hingga kegagalan komponen-komponennya dan memberikan nilai bisnis langsung melalui otomatisasi pemeliharaan preventif terjadwal. Dengan menggunakan

analisis prediktif dan preskriptif, Anda dapat memberikan pengalaman pelanggan yang lebih baik, meningkatkan kualitas produk, dan menciptakan keunggulan kompetitif.

Data besar

Apa yang Anda pelajari dengan sangat cepat dengan IoT adalah ketika Anda memiliki orang, tempat, dan benda yang dilengkapi sensor, maka semuanya menjadi tentang data. Ada sekitar 2 miliar PC di dunia pada tahun 2017 dan sekitar 10 miliar perangkat seluler. Pada tahun 2020, diperkirakan akan ada lebih dari 50 miliar perangkat terhubung yang menyalurkan data berukuran Exabytes ke cloud. Bagaimana kita dapat menyerap, mentransformasi, menyimpan, dan menganalisis data ini, dan yang lebih penting lagi, bagaimana kita melakukan kueri dan memvisualisasikan data sehingga kita dapat dengan cepat mendapatkan wawasan dan mengambil tindakan? Kita perlu mempelajari keterampilan baru dan memanfaatkan kemampuan cloud baru untuk menangani masuknya data dalam jumlah besar ini.

Analisis Tingkat Lanjut

Jika data adalah bijihnya, maka pengetahuan adalah emasnya. Salah satu tujuan pengumpulan semua data ini adalah untuk dapat memperoleh wawasan dari visualisasi data real-time yang meningkatkan nilai bisnis dan mengotomatiskan proses bisnis penting. Pola pemrosesan data IoT yang khas adalah menyiapkan rute jalur dingin, hangat, dan panas untuk data di mana jalur dingin menyediakan penyimpanan jangka panjang, jalur hangat menyediakan penyimpanan untuk dasbor waktu nyata, pemrosesan batch cepat dan batch lambat, dan jalur panas mengirimkan pesan melalui antrian ke layanan mikro yang menyediakan layanan pemberitahuan waktu nyata untuk peringatan dan alarm.

Analisis prediktif adalah pilihan populer untuk solusi real-time karena merupakan proses penambangan data yang berfokus pada memprediksi keadaan di masa depan. Model data dibuat dari data historis atau sampel. Model-model ini digunakan bersama dengan algoritma statistik untuk memeriksa aliran data real-time dan membuat prediksi. Perusahaan manufaktur dapat menggunakan analisis prediktif untuk menentukan kapan komponen produk mereka akan rusak dan menggunakannya untuk mengotomatiskan proses penjadwalan teknisi layanan lapangan untuk mengunjungi dan melakukan pemeliharaan. Sebuah klinik medis dapat menggunakan analisis prediktif untuk memeriksa data genom guna melihat apakah suatu populasi ditakdirkan untuk suatu kondisi medis. Sebuah gerai ritel dapat menggunakan pendekatan ini yang diterapkan pada tingkat persediaan waktu nyata untuk memprediksi kapan suatu produk populer akan habis.

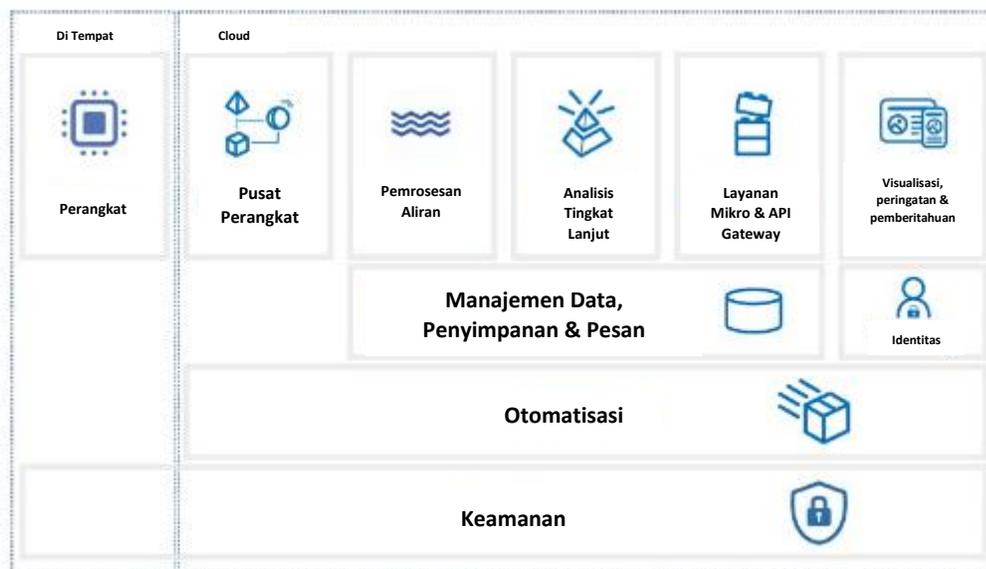
1.4 ARSITEKTUR REFERENSI BISNIS REAL-TIME

Untuk mewujudkan manfaat solusi Bisnis Real-Time memerlukan serangkaian kemampuan bisnis dan teknis yang menentukan proses yang Anda gunakan untuk menyerap peristiwa perangkat, melakukan analisis tingkat lanjut, mengumpulkan wawasan, dan bertindak. Bisnis perlu mengembangkan model yang menunjukkan bagaimana proses akan ditingkatkan untuk mengurangi biaya dan saluran pendapatan baru apa yang akan diidentifikasi dan

dimanfaatkan untuk meningkatkan keuntungan. Misalnya, produsen dapat mengurangi biaya dengan hanya mengirimkan teknisi lapangan untuk melakukan pemeliharaan bila diperlukan. Pabrik yang sama mungkin menemukan bahwa mereka dapat menawarkan kontrak layanan yang lebih fleksibel sehingga meningkatkan penjualan dan pendapatan kontrak layanan bulanan.

Tim pengembangan produk yang bertanggung jawab untuk mendukung model bisnis baru ini memerlukan arsitektur referensi yang memberikan peta jalan tentang bagaimana mereka dapat membangun sistem real-time dari perangkat yang terhubung dan layanan cloud. Gambar 1.7 menggambarkan arsitektur referensi untuk Bisnis Real-Time. Setiap komponen arsitektur mewakili kombinasi layanan cloud dasar untuk mengimplementasikan pipeline data real-time, analitik tingkat lanjut, penyimpanan data besar, API, dan skrip otomatisasi pendukung serta protokol keamanan.

Bersama-sama, komponen-komponen ini memberikan kemampuan teknis platform Bisnis Real-Time.



Gambar 1.7. Arsitektur referensi Bisnis Real-Time

Perangkat

Setiap skenario IoT memerlukan analisis dan strategi tentang cara terbaik untuk mengaktifkan sensor pada produk, lingkungan, atau orang-orang untuk mengumpulkan data yang diperlukan secara efisien dan aman untuk mendorong kasus bisnis. Ini mungkin melibatkan tag RFID untuk lokasi dan identifikasi produk; sensor lingkungan untuk suhu, kelembapan, atau kecepatan angin; sensor mekanis untuk melacak putaran roda gigi atau iterasi pompa hidrolik; atau dalam kasus manusia, sensor biometrik untuk detak jantung, suhu kulit, atau glukosa darah.

Selain sensor, Anda memerlukan perangkat dengan sistem operasi tertanam seperti Linux atau Windows 10 IoT, untuk menghosting kode yang mengelola konektivitas ke sensor dan

lingkungan fisik tempat perangkat digunakan, mengamankan konektivitas melalui kabel, nirkabel dan/atau jaringan seluler, autentikasi aman ke cloud, pengiriman pesan dari perangkat ke cloud seperti detak jantung dan telemetri, serta pengiriman pesan dari cloud ke perangkat untuk perintah dan kontrol serta manajemen perangkat. Skenario tingkat lanjut dapat menambahkan analitik, pemfilteran, aturan dan peringatan bisnis, serta pemberitahuan di edge. Perangkat juga akan berpartisipasi dalam protokol manajemen perangkat untuk mengelola status perangkat, peningkatan firmware, dan operasi kendali jarak jauh lainnya.

Pusat Perangkat

Hub perangkat adalah layanan yang dihosting di cloud yang menyediakan konektivitas perangkat yang aman, penyerapan telemetri, serta perintah dan kontrol jarak jauh. Layanan ini harus menyediakan kemampuan ini dalam skala besar sehingga, seiring bertambahnya jumlah perangkat yang terhubung, layanan tidak akan pernah gagal. Layanan ini mungkin menyediakan penyimpanan sementara untuk semua pesan masuk. Penyimpanan sementara ini memungkinkan analisis peristiwa pesan secara real-time, analisis pesan dalam periode waktu singkat, serta kemampuan untuk kembali ke masa lalu untuk menilai kembali peristiwa tersebut.

Manajemen perangkat biasanya dimasukkan ke dalam layanan hub perangkat. Manajemen perangkat menyediakan kemampuan untuk mendaftarkan perangkat menggunakan pengidentifikasi uniknya seperti nomor seri. Setelah terdaftar, perangkat akan dapat terhubung dengan aman ke hub perangkat untuk tujuan komunikasi. Manajemen perangkat dapat memanfaatkan konsep kembaran perangkat, yang merupakan representasi digital dari status perangkat yang tetap tersinkronisasi dengan perangkat fisik. Kembaran perangkat memberikan kemampuan untuk menyinkronkan nilai properti perangkat di cloud dengan perangkat di lapangan, yang pada dasarnya melakukan konfigurasi status yang diinginkan (DSC) dalam skala besar.

Pemrosesan Aliran

Pemrosesan aliran adalah layanan yang dihosting di cloud yang menyediakan analisis real-time pada telemetri masuk. Layanan ini memungkinkan Anda melakukan kueri terhadap seluruh pesan masuk secara real-time, memilih pesan dengan jenis tertentu atau yang berisi nilai tertentu, menerapkan agregasi dan perhitungan dari waktu ke waktu (windowing), mengubah pesan, mengidentifikasi kondisi alarm, dan kemudian mengambil tindakan berdasarkan hasilnya. dari analitik. Dalam kebanyakan kasus, layanan ini merutekan pesan yang dihasilkan ke lokasi penyimpanan, API, atau antrean pesan untuk diproses lebih lanjut.

Manajemen Data, Penyimpanan, dan Pesan

Untuk mendapatkan hasil maksimal dari data real-time yang kini mengalir melalui cloud, Anda perlu menyediakan berbagai jenis penyimpanan dan manajemen data, masing-masing dioptimalkan untuk langkah berikutnya dalam jalur pemrosesan data.

- *Penyimpanan Blob*: Penyimpanan Blob biasanya digunakan untuk pengarsipan peristiwa.

- *Simpan dan Teruskan Pesan*: Untuk menyediakan integrasi dengan layanan mikro berbasis peristiwa atau sistem lini bisnis lokal, mekanisme antrean dan pub/sub menyediakan pesan yang digabungkan secara longgar.
- *NoSQL dan Basis Data Relasional*: Untuk kemampuan kueri berbasis waktu dan integrasi dengan aplikasi tradisional dan dasbor modern, Anda dapat memanfaatkan layanan penyimpanan relasional atau NoSQL.
- *Data Lake*: Sistem File Terdistribusi Hadoop (HDFS) yang menyediakan penyimpanan data besar dan akses kueri lintas bahasa dan dapat digunakan dengan mesin analisis tingkat lanjut.
- *Ekstrak, Transformasi, dan Muat*: Anda mungkin ingin menggunakan layanan ETL untuk melakukan operasi integrasi dan transformasi data.

Analisis Tingkat Lanjut

Analisis tingkat lanjut adalah istilah umum untuk semua kemungkinan analisis data besar yang ingin Anda lakukan pada data real-time Anda. Hal ini mungkin melibatkan penggabungan data real-time dengan data historis dan referensi serta memanfaatkan kemampuan kueri terdistribusi dari mesin Hadoop, menggunakan alat ETL untuk berintegrasi ke dalam gudang data atau melatih model prediktif untuk mengotomatiskan pemeliharaan preventif. Kemajuan terbaru dalam Azure Cognitive API dan bot kecerdasan buatan juga dapat digunakan bersama dengan data real-time ini untuk menciptakan pengalaman percakapan baru yang mendalam bagi pelanggan Anda.

Layanan mikro dan API Gateway

Layanan mikro menyediakan kemampuan bisnis atau platform melalui API yang terdefinisi dengan baik, kontrak data, konfigurasi, dan penyimpanan data mendasar yang diperlukan agar dapat berfungsi. Ini menyediakan fungsi ini dan hanya fungsi ini. Ia melakukan satu hal dan melakukannya dengan baik. Layanan mikro mewakili kemampuan bisnis yang ditentukan menggunakan desain berbasis domain, diimplementasikan menggunakan praktik terbaik berorientasi objek, diuji pada setiap langkah dalam jalur penerapan, dan diterapkan melalui otomatisasi sebagai layanan yang otonom, terisolasi, sangat skalabel, dan tangguh dalam infrastruktur cloud terdistribusi. Solusi IoT mungkin memiliki tiga jenis layanan mikro:

- ❖ **Transaksional**: Layanan mikro transaksional bertanggung jawab untuk menulis pesan ke penyimpanan yang sesuai.
- ❖ **Didorong Peristiwa**: Layanan mikro berbasis peristiwa mendengarkan antrean pesan dan bertindak jika ada pesan yang masuk ke antrean. Layanan mikro ini biasanya digunakan untuk mendorong proses bisnis peringatan dan notifikasi atau berintegrasi dengan lini sistem bisnis lain yang memerlukan penanganan pesan khusus.
- ❖ **Kontrak API**: Layanan mikro ini memanfaatkan titik akhir ReST dan model data JSON serta menyediakan permasalahan lintas sektoral dan kemampuan bisnis yang ingin Anda hadapi pada aplikasi apa pun yang memakan waktu.

Gerbang API menyediakan proksi API aman yang menggabungkan API ReST Anda, mengatur API ke dalam produk, menyediakan akses terbatas ke produk API melalui definisi grup pengembang, menyediakan kemampuan berlangganan, menyediakan injeksi kebijakan, pembatasan, kuota, dll., dan menyediakan analitik di tingkat produk, API, dan operasi. Gerbang API memberikan kemampuan untuk memisahkan API Anda menjadi akses pribadi, semi-pribadi (akses mitra), atau publik, lalu memonetisasi API tersebut untuk membuat saluran pendapatan baru.

Visualisasi, Peringatan, dan Pemberitahuan

Kini setelah Anda dapat menyerap, menganalisis, dan menyimpan data real-time, Anda dapat membuat aplikasi yang berhubungan dengan pelanggan, mitra, dan karyawan yang memberikan visualisasi data yang berdampak, peringatan dan pemberitahuan visual dan berpusat pada perangkat menggunakan API dan layanan mikro berbasis peristiwa. Anda dapat memanfaatkan layanan pihak ketiga seperti Twilio dan Send Grid untuk memberikan pemberitahuan teks, suara, dan email. Selain itu, Anda mungkin ingin menggunakan layanan notifikasi seluler yang dihosting di cloud untuk pembaruan waktu nyata di perangkat seluler.

Identitas

Anda ingin memberikan akses ke aplikasi yang Anda buat kepada pelanggan, mitra, dan karyawan Anda. Layanan identifikasi menyediakan mekanisme autentikasi dan otorisasi sistem masuk tunggal atau multipenyewa sehingga orang yang masuk ke aplikasi Anda hanya dapat melihat data dan fungsionalitas aplikasi yang disediakan oleh peran mereka. Layanan identitas dapat menyediakan fitur-fitur ini dalam model Business-to-Consumer (B2C) dan Business-to-Business (B2B), termasuk integrasi dengan layanan direktori perusahaan Anda.

Otomatisasi

Merancang, mengembangkan, menerapkan, dan mengoperasionalkan solusi IoT memerlukan penerapan pendekatan otomatis pada siklus hidup produk perangkat lunak. Istilah populer saat ini adalah DevOps. DevOps menyiratkan bahwa Anda diatur ke dalam model tim produk perangkat lunak yang menekankan kualitas produk saat kode bergerak melalui jalur penerapan otomatis. Tim memiliki proses yang terdefinisi dengan baik dan menggunakan seperangkat alat untuk mengotomatiskan pekerjaannya, mengurangi kesalahan, dan meningkatkan kualitas. Tim ini memanfaatkan platform cloud untuk mengotomatiskan penyediaan infrastruktur cloud, melakukan manajemen pembangunan, pengujian, dan rilis, serta pemantauan dan pengumpulan metrik kesehatan runtime.

Keamanan

Solusi IoT memiliki empat zona keamanan; lokal, perangkat, gateway cloud, dan layanan cloud. Zona lokal adalah lingkungan fisik tempat perangkat disebarkan. Zona perangkat mewakili perangkat IoT, cara konfigurasinya dengan zona lokal dan cara kerja internalnya, sistem operasi, dan aplikasi. Zona gateway cloud mewakili titik akhir yang dapat diakses oleh publik tempat perangkat terhubung dan berkomunikasi. Terakhir, zona layanan cloud menyediakan akses ke

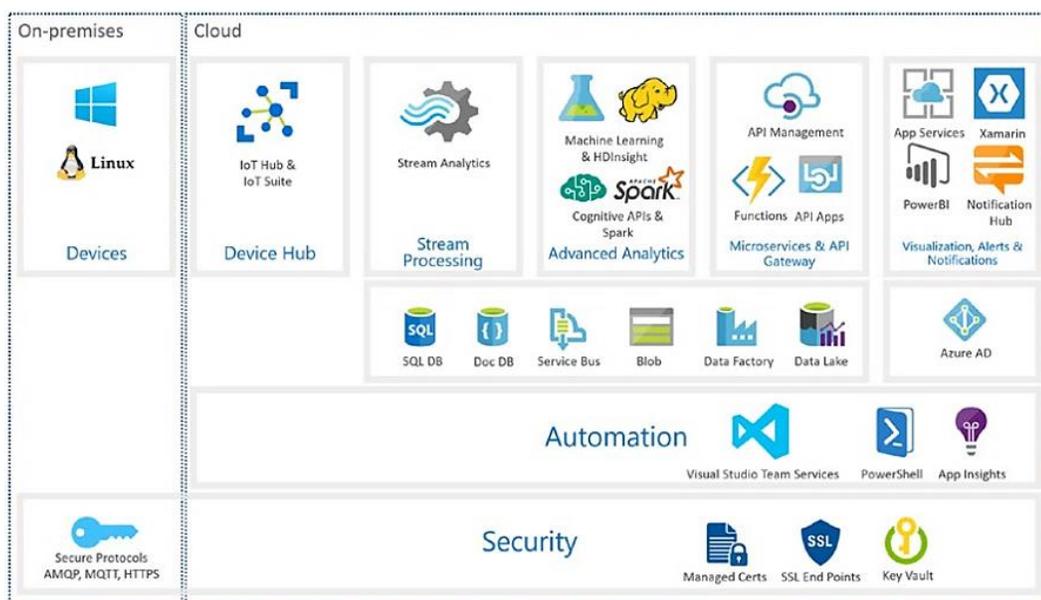
pesan masuk ke layanan internal yang dihosting di cloud yang menyediakan penyerapan data, pemrosesan aliran, penyimpanan, analitik tingkat lanjut, dan integrasi aplikasi.

Berikut adalah beberapa hal yang perlu diingat sehubungan dengan pengamanan IoT dan solusi analitik tingkat lanjut:

- Amankan jaringan kabel/nirkabel tempat perangkat Anda berjalan.
- Pastikan sistem operasi tertanam yang menjalankan perangkat Anda adalah yang terbaru dan dapat ditambah dari jarak jauh melalui proses peningkatan firmware otomatis.
- Gunakan TLS untuk mengamankan koneksi dari perangkat pintar dan/atau edge gateway ke cloud.
- Enkripsi pesan saat diam dan dalam pertargungan.
- Pastikan titik akhir cloud publik Anda diamankan menggunakan SSL dan Anda menggunakan beberapa bentuk autentikasi dan otorisasi seperti Basic, OAuth, sertifikat terkelola, atau Kebijakan Akses Bersama.
- Memanfaatkan layanan identifikasi untuk akses ke aplikasi Anda.
- Memanfaatkan layanan manajemen kunci untuk memberikan tata kelola dan akses aman terbatas ke sertifikat Anda.

Microsoft Azure IoT dan Cortana Intelligence Suite

Microsoft menyediakan rangkaian IoT yang kaya fitur dan lengkap serta sumber daya analitik tingkat lanjut yang dapat digunakan untuk membangun platform Bisnis Real-Time Anda. Untuk setiap kemampuan dalam arsitektur referensi, ada satu atau beberapa layanan Azure yang menyediakan fungsi tersebut melalui layanan akhir yang dapat diskalakan dan dikonfigurasi. Kuncinya adalah mengetahui cara menyatukan semuanya menggunakan kombinasi kode, konfigurasi, dan praktik terbaik (lihat Gambar 1.8).



Gambar 1-8. Arsitektur referensi dan pemetaan Azure

Otomatisasi

Azure menyediakan dukungan lengkap untuk integrasi berkelanjutan dan proses pengiriman berkelanjutan menggunakan PowerShell, Templat Azure Resource Manager (ARM), dan Visual Studio Team Services (VSTS). Alat sumber terbuka dan pihak ketiga juga didukung dengan menyediakan ekosistem alat dan utilitas otomatisasi yang kaya dan beragam, apa pun pilihan sistem operasi atau bahasa pemrograman Anda. Lihat Bab 2 untuk detail selengkapnya tentang otomatisasi menggunakan PowerShell, Templat ARM, dan VSTS.

Keamanan dan Identitas

Keamanan adalah perhatian utama dalam sistem terdistribusi dan dengan diperkenalkannya perangkat, sensor, beacon, edge gateway, dan protokol jaringan baru, kita harus rajin dalam mendefinisikan ancaman keamanan pada sistem dan merancang solusi dari awal dengan keamanan di pikiran. Azure menyediakan layanan dasar untuk pengelolaan identitas pengguna akhir secara mandiri yang, dikombinasikan dengan model untuk manajemen langganan dan identifikasi pelanggan dan organisasi, Anda dapat memberikan pengalaman yang lancar bagi pelanggan, mitra, dan karyawan Anda. Lihat Bab 10 untuk detail tentang keamanan dan identitas menggunakan Azure AD B2C.

Perangkat

Microsoft menyediakan IoT Gateway SDK, pustaka lintas platform sumber terbuka yang menyediakan infrastruktur dan modul plug-and-play untuk menciptakan solusi gateway IoT. Dengan menggunakan SDK, Anda dapat mengembangkan aplikasi yang memungkinkan perangkat berkomunikasi dengan Azure IoT Hub. SDK menyediakan kumpulan modul yang dapat melakukan operasi seperti agregasi dan transformasi pesan, analisis matematis, penyimpanan lokal, peringatan dan pemberitahuan lokal, dan sebagainya. Modul berkomunikasi secara pipeline dan menyampaikan pesan melalui perantara pesan. Anda dapat memperluas SDK dengan mengembangkan modul Anda sendiri. Lihat Bab 4 untuk rincian tentang perangkat.

Pusat Perangkat

IoT Hub adalah layanan Azure yang menyediakan pendaftaran perangkat, manajemen perangkat, penyerapan telemetri, serta perintah dan kontrol. IoT Hub menyediakan perpesanan perangkat-ke-cloud dan cloud-ke-perangkat yang andal dalam skala besar, memungkinkan komunikasi yang aman menggunakan kredensial dan autentikasi keamanan per perangkat, mencakup pemantauan ekstensif terhadap konektivitas dan properti perangkat melalui kembaran perangkat, dan menyediakan akses melalui serangkaian SDK khusus bahasa termasuk C, C#, Java, Node, dan Python. Lihat Bab 3 untuk cakupan Azure IoT Hub dan manajemen perangkat.

Pemrosesan Aliran

Di Azure, kemampuan ini disebut Azure Stream Analytics atau ASA. ASA memungkinkan Anda dengan cepat mengembangkan dan menerapkan layanan mikro kecil yang menentukan input, output, dan kueri yang memilih subkumpulan pesan untuk pemrosesan analitis.

Pemrosesan analitis dapat memanfaatkan windowing, menggabungkan pesan dari waktu ke waktu, penerapan aturan terhadap nilai dalam pesan, dan panggilan eksternal ke Azure Functions atau Machine Learning API untuk menentukan di mana merutekan pesan untuk langkah dalam alur analisis data. Bab 5 membahas Analisis Aliran secara mendetail.

Manajemen Data, Penyimpanan, dan Pesan

Ada banyak opsi penyimpanan di Azure termasuk namun tidak terbatas pada penyimpanan blob, SQL Database, Cosmos DB, dan Data Lake, Sistem File Terdistribusi Hadoop sebagai kemampuan layanan yang memungkinkan Anda mengelola data dalam jumlah besar dan ukuran file dalam rentang petabyte. Azure menyediakan Pabrik Data untuk melakukan operasi ETL pada data yang disimpan di Azure atau berpindah masuk dan keluar dari lokasi penyimpanan lokal. Dan yang terakhir, Azure menyediakan Bus Layanan untuk menyimpan dan meneruskan pesan, skenario pub/sub, dan penyerapan data volume tinggi menggunakan Event Hubs.

Analisis Tingkat Lanjut

Azure menyediakan serangkaian kemampuan analitik yang terkelola sepenuhnya termasuk HDInsight, Hadoop as a Service dari Microsoft, pembelajaran mesin untuk analitik prediktif, dan analitik Data Lake untuk distribusi skala besar. analitik yang memanfaatkan Yarn dan U-SQL. Microsoft juga menyediakan serangkaian API kognitif yang memberikan kecerdasan pembelajaran mesin tingkat lanjut untuk pemahaman bahasa, pengenalan wajah, analisis sentimen, dan banyak lagi. Lihat Bab 8 untuk cakupan Azure Machine Learning dan bahasa pemrograman R.

Layanan mikro dan API Gateway

Saat Anda membangun solusi di Azure dan layanan dasarnya, Anda perlu memanfaatkan arsitektur layanan mikro untuk mengoptimalkan penggunaan fitur sesuai permintaan Azure dan untuk dapat mendukung jalur pengembangan pengiriman berkelanjutan dan siklus rilis berkecepatan tinggi. Anda memiliki beberapa pilihan dalam cara mengemas dan menyebarkan layanan mikro di Azure termasuk *App Services*, *Service Fabric*, *Azure Container Services (ACS)*, dan *Docker*.

Untuk layanan mikro yang mengekspos ReST API, Azure menyediakan Manajemen API untuk menerbitkan API Anda menggunakan model berlangganan baik untuk pengembang internal maupun eksternal. Mereka dapat menerapkan kebijakan, mengumpulkan statistik, dan memberikan keamanan untuk melindungi mereka dari penyalahgunaan. Bab 2 memberikan detail tentang Azure API Management.

Visualisasi, Peringatan, dan Pemberitahuan

Fitur utama dari solusi Bisnis Real-Time adalah visualisasi data real-time yang disediakan melalui dasbor yang dapat diperbarui sendiri, peringatan, dan pemberitahuan yang disediakan melalui dasbor dan perangkat seluler serta melalui integrasi sistem, yang dapat mencakup aplikasi yang dihosting di cloud seperti seperti Dynamics CRM dan aplikasi lini bisnis lokal seperti SAP.

Ada banyak sekali teknologi yang memungkinkan Anda menciptakan pengalaman pengguna seperti ini, mulai dari web responsif menggunakan Node.JS, Angular.JS, D3, dan pustaka JavaScript populer lainnya hingga kerangka kerja seluler asli atau lintas platform seperti Xamarin dan off-road. produk visualisasi data yang tersedia seperti PowerBI.

1.5 REFERENSI DAN IMPLEMENTASI KESEHATAN DAN KESELAMATAN PEKERJA

Saat Anda menelusuri bab-bab buku ini, kami akan membahas setiap komponen dalam arsitektur referensi. Kami telah menyediakan implementasi referensi untuk menunjukkan pola dan praktik yang memanfaatkan Azure IoT dan Cortana Intelligence Suite. Solusinya disebut Kesehatan dan Keselamatan Pekerja. Repositori telah dirancang untuk mendukung perkembangan linier sepanjang buku seiring dengan perkembangan setiap bab dari bab-bab sebelumnya. Untuk mengikuti latihan dalam buku ini, Anda memerlukan akun Azure.

■ Catatan Anda dapat mendaftar akun Azure dengan mengunjungi <http://azure.net>.

Latar Belakang

Skenario solusinya adalah Anda memberikan solusi SaaS yang memungkinkan pelanggan Anda menghubungkan karyawan mereka yang bekerja dalam kondisi berbahaya menggunakan pakaian dan perangkat yang dapat dikenakan dengan sensor. Sensor-sensor tersebut menyediakan serangkaian data biometrik yang dikumpulkan secara real-time dan digunakan untuk memberi masukan pada mesin analitik prediktif yang akan memberikan peringatan sebelum seorang karyawan mencapai tingkat kelelahan atau stres.

Telah ditentukan tiga perusahaan fiktif yang menghadirkan pelanggan Anda. Setiap perusahaan memiliki 15 karyawan yang diawasi.

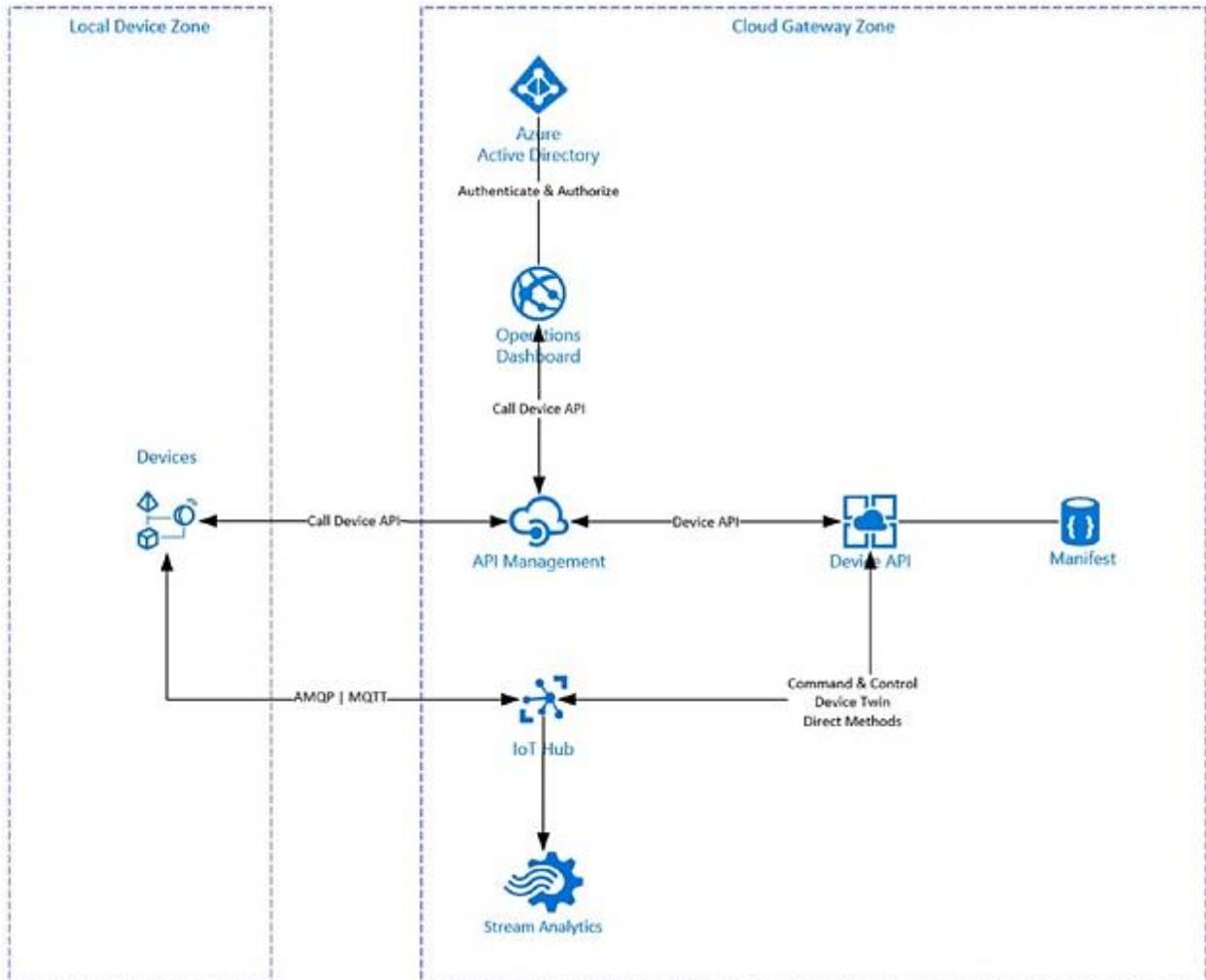
- *WigiTech*: Perusahaan teknologi yang ingin memantau karyawan pabrik yang bekerja dalam kondisi berbahaya
- *Menara Tinggi*: Perusahaan jasa utilitas yang karyawannya melakukan pemeliharaan menara komunikasi yang berada di atas gedung pencakar langit
- *The Complex Badger*: Sebuah perusahaan angkutan truk yang mengkhususkan diri dalam memindahkan alat berat masuk dan keluar dari lokasi pertambangan dan penebangan kayu ingin memantau karyawannya untuk memastikan mereka mampu menangani kondisi berkendara yang sulit

Arsitektur Solusi

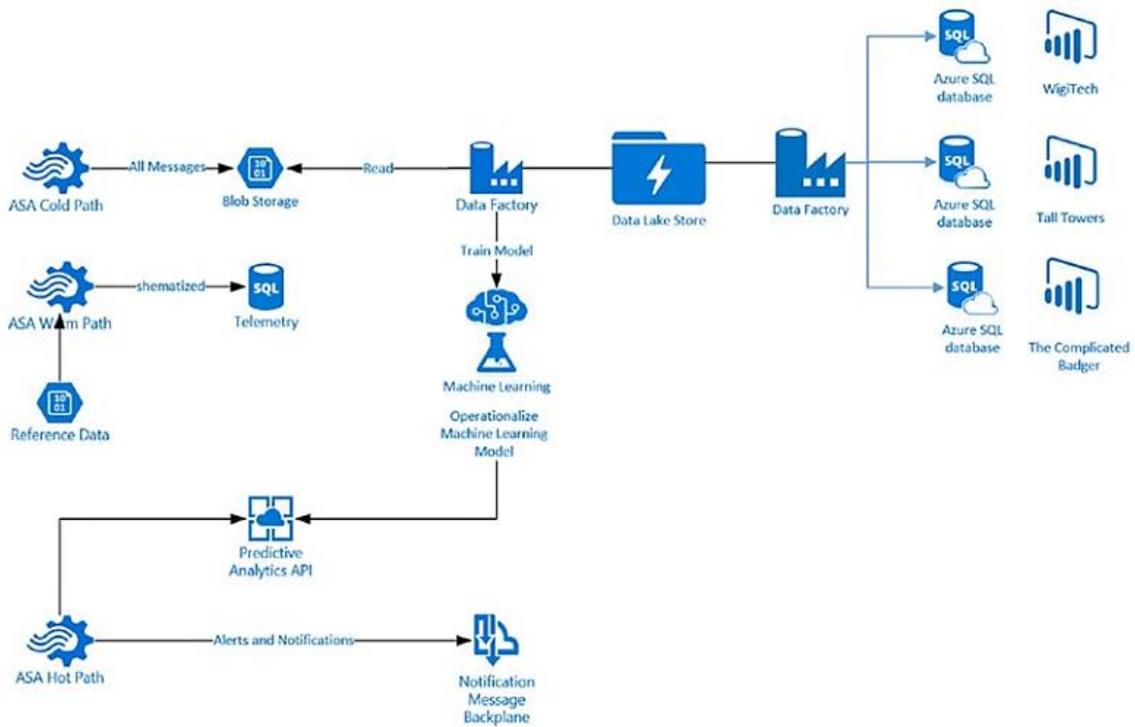
Arsitektur solusi paling baik digambarkan sebagai tiga subsistem yang bekerja sama (lihat Gambar 1-9, 1-10, dan 1-11):

- *Sub-Sistem IoT*: Layanan yang diperlukan untuk menyediakan konektivitas perangkat, komunikasi dua arah, dan manajemen perangkat
- *Sub-Sistem Analisis*: Layanan yang diperlukan untuk melakukan analisis waktu nyata dan perutean peristiwa, penyimpanan data besar dan pengelolaan data, serta analisis prediktif

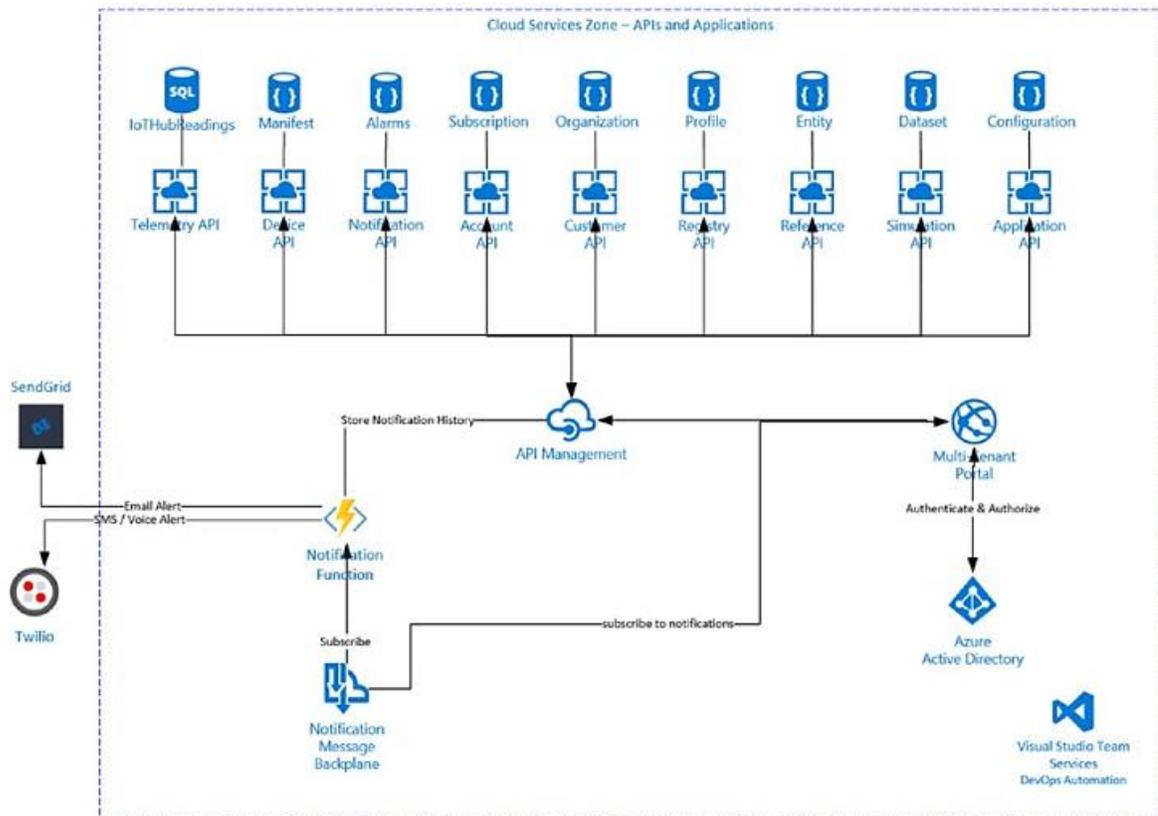
- *Sub-Sistem Layanan Aplikasi:* Layanan yang diperlukan untuk menyediakan akses aman ke aplikasi, API, dan data yang mendasarinya serta alat yang menyediakan otomatisasi untuk penyediaan, pembuatan, pengujian, penerapan, dan pemantauan



Gambar 1-9. Sub-sistem IoT



Gambar 1-10. Sub-sistem analitik



Gambar 1.11 Application services sub-system

Mengunduh Repositori

Jika Anda ingin memanfaatkan latihan langsung di setiap bab, kami telah menyediakan repositori kode implementasi referensi sebagai titik awal. Repositori terletak di sini:

<https://github.com/brtbook/brt>

Kloning repositori ke lingkungan lokal Anda. Latihan di Bab 2 akan merinci pengaturan dan konfigurasi lingkungan pengembangan Anda. Perlu diperhatikan bahwa langganan Azure diperlukan dan Anda akan dikenakan biaya untuk layanan yang Anda provisi di lingkungan Azure Anda.

11.6 RINGKASAN

Pada bab ini, kami memperkenalkan konsep Bisnis Real-Time, tantangan serta manfaatnya. Kami memberikan ikhtisar tentang penyewa inti pengembangan perangkat lunak modern, lean engineering, DevOps, layanan mikro, dan cloud.

Kami memperkenalkan arsitektur referensi Bisnis Real-Time, yang memberikan peta jalan tentang bagaimana Anda dapat merancang SaaS IoT yang sangat skalabel dan solusi analitik tingkat lanjut. Terakhir, kami memetakan arsitektur referensi ke sumber daya dasar di Azure yang kami perlukan untuk membawa solusi kami ke pasar dengan cepat.

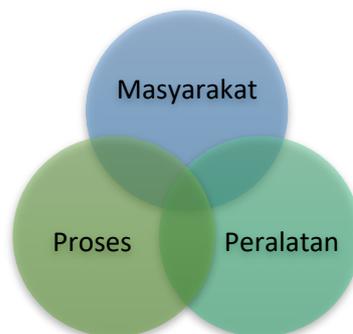
Setiap bab berikutnya akan mencakup area arsitektur referensi bersama dengan layanan Azure relevan yang dipetakan ke area tersebut. Kami akan menggunakan implementasi referensi untuk Kesehatan dan Keselamatan Pekerja untuk menunjukkan pola dan praktiknya.

BAB 2

DEVOPS MENGGUNAKAN POWERSHELL, ARM, DAN VSTS

2.1 PENDAHULUAN

DevOps dapat diringkas sebagai penyatuan orang, proses, dan alat (lihat Gambar 2.1), penyelarasan tim pengembangan dan operasi, otomatisasi proses pengembangan, pengujian dan rilis, serta pemilihan serangkaian alat yang konsisten untuk memfasilitasi otomatisasi siklus penyediaan, pembuatan, pengujian, dan rilis. Tujuan penerapan pendekatan DevOps adalah untuk menciptakan siklus hidup pengembangan produk yang efisien yang menghilangkan, semaksimal mungkin, kesalahan yang terjadi melalui langkah manual, yaitu kesalahan yang disebabkan oleh kesalahan manusia.



Gambar 2.1. Kerangka kerja DevOps

Melalui kombinasi studi empiris dan pengalaman langsung, para profesional perangkat lunak menyadari bahwa tim kecil lintas fungsi adalah tim yang optimal untuk pembuatan perangkat lunak modern. Aplikasi rekayasa Lean, Agile, Scrum, dkk. semakin populer untuk mengisi kekosongan metodologi dan proses, dan alat-alat tersebut telah berevolusi untuk mempromosikan otomatisasi kepada masyarakat kelas satu di bidang teknik. Setiap organisasi harus menemukan jalurnya sendiri menuju DevOps. Cara perusahaan mengukur kesuksesan akan berbeda-beda berdasarkan tempat di mana mereka paling menderita. Oleh karena itu, metrik yang digunakan untuk mengukur keberhasilan akan berbeda untuk setiap organisasi. Beberapa metrik umum mencakup penerapan yang lebih sering, pemulihan kegagalan yang lebih cepat, tingkat kegagalan yang lebih rendah, dan waktu pemasaran. Menentukan metrik yang paling penting bagi organisasi Anda memerlukan refleksi diri dan kolaborasi antara semua tim yang akan terkena dampak perubahan bawaan.

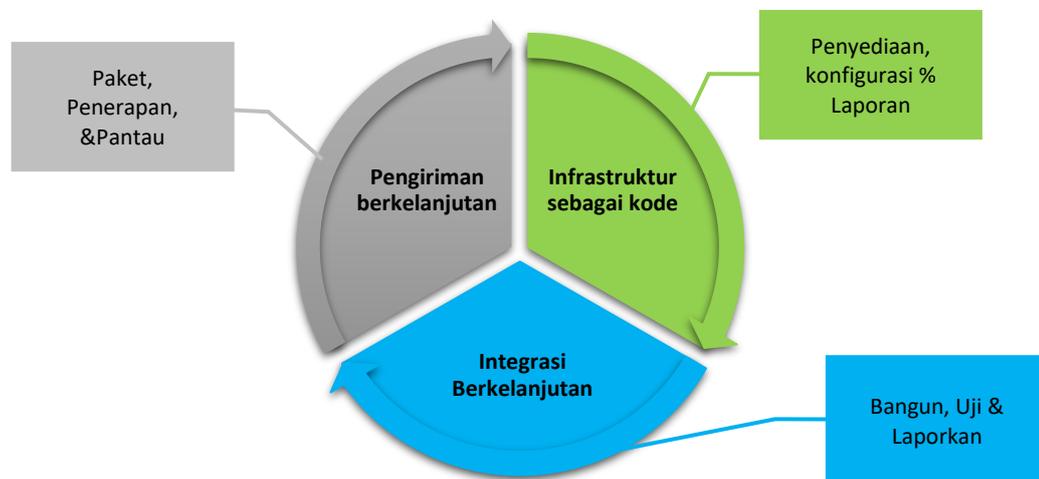
Masyarakat

Saat memperkenalkan DevOps ke dalam organisasi Anda, aspek sumber daya manusia bisa menjadi hal yang paling menantang. Memperkenalkan perubahan yang berdampak ke dalam suatu organisasi akan sulit dilakukan tanpa dukungan dari pimpinan dan dukungan dari staf teknis. Untuk memfasilitasi perubahan ini, perusahaan Anda mungkin perlu memperkenalkan struktur organisasi baru dan model tim baru, yang meruntuhkan tembok

antara pengembangan dan operasi. Staf teknik mungkin perlu mempelajari keterampilan baru, mengubah kebiasaan sehari-hari, dan belajar bekerja lebih kolaboratif. Anda mungkin perlu merekrut keterampilan baru, melatih kembali staf, dan, favorit saya, menyewa perusahaan jasa profesional untuk membantu transisi ini.

Proses

Proses didefinisikan sebagai serangkaian tindakan atau langkah yang diambil untuk mencapai suatu hasil. Tujuan dari memiliki siklus hidup pengembangan produk yang terdefinisi dengan baik adalah untuk menghasilkan perangkat lunak yang berkualitas tinggi dan berharga dengan cara yang efisien, cepat, dan andal, sehingga memungkinkan siklus rilis yang sering yang memenuhi keinginan bisnis dan pelanggan. Ada tiga tahapan utama dalam proses DevOps: Infrastruktur sebagai Kode, Integrasi Berkelanjutan, dan Pengiriman Berkelanjutan, seperti yang ditunjukkan pada Gambar 2-2.



Gambar 2-2. proses DevOps

Infrastruktur sebagai Kode

Infrastruktur sebagai Kode adalah proses mendefinisikan konfigurasi Infrastruktur sebagai Layanan (IaaS) dan Platform-as-a-Service (PaaS) yang terdiri dari jaringan, mesin virtual, dan layanan perangkat lunak menggunakan templat dan kemudian mengatur pembuatannya melalui skrip otomatisasi dan utilitas. Dengan menggunakan pendekatan ini, dimungkinkan untuk membuat instance dan mengkonfigurasi semua layanan komputasi, penyimpanan, dan perangkat lunak yang diperlukan untuk sistem terdistribusi on-demand yang canggih.

Integrasi Berkelanjutan

Integrasi berkelanjutan adalah proses di mana kode dikembangkan dan diperiksa ke dalam repositori kode sumber yang memicu pengujian unit dan fungsional secara otomatis. Jika pengujian gagal, pengembang diberi tahu melalui otomatisasi item pekerjaan yang ditambahkan ke backlog masing-masing.

Integrasi berkelanjutan memberikan umpan balik cepat kepada pengembang mengenai kualitas kode mereka. Kode hanya diterima di cabang master jika melewati

pemeriksaan kualitas ini. Jika build gagal, perbaikan build menjadi prioritas tertinggi. Proses ini harus berjalan “terus menerus”, sesuai dengan namanya, karena proses ini meletakkan dasar untuk pengiriman berkelanjutan, memastikan bahwa perangkat lunak yang dibangun saat ini stabil dan dapat diterapkan sepanjang siklus hidupnya.

Pengiriman Berkelanjutan

Pengiriman berkelanjutan adalah proses dimana perangkat lunak dikemas dan diterapkan ke lingkungan cloud. Sangat penting bahwa jika ada masalah dengan langkah-langkah ini, perbaikannya menjadi prioritas dibandingkan memberikan fitur produk baru. Langkah-langkah yang menjadi tanggung jawab proses pengiriman berkelanjutan adalah:

- *Pengemasan*: Proses dimana perangkat lunak dikemas untuk diterapkan. Ini bisa sesederhana membuat file ZIP yang digunakan oleh proses penerapan untuk menggunakan teknologi kontainer seperti Docker dan Azure Container Services.
- *Deployment*: Proses dimana perangkat lunak yang dikemas disebarkan ke cloud. Hal ini mungkin melibatkan penyalinan paket atau gambar kontainer lokal ke cloud atau dari satu lokasi cloud ke lokasi cloud lainnya.
- *Pemantauan*: Proses dimana tim dapat memantau perangkat lunak yang berjalan untuk menentukan apakah penerapannya sehat dan berfungsi.

Karena kami memanfaatkan Azure untuk menghosting solusi ini, kami dapat menggunakan Azure untuk menyediakan infrastruktur sesuai permintaan untuk membangun, menguji, mengemas, dan menerapkan perangkat lunak.

Peralatan

Seluruh industri telah tumbuh berdasarkan integrasi berkelanjutan dan pengiriman berkelanjutan. Ada banyak sekali produk dan alat baik dari vendor perangkat lunak independen maupun komunitas open source. Alat-alat ini memberikan kemampuan di seluruh siklus hidup pengembangan produk perangkat lunak termasuk manajemen kode sumber, otomatisasi bangunan, kerangka pengujian dan otomatisasi pengujian, alat manajemen proyek, pelacakan bug, Lingkungan Pengembangan Terpadu (IDE), otomatisasi proses pengemasan dan penerapan, konfigurasi keadaan yang diinginkan, dan daftarnya terus berlanjut.

Pilihan alat Anda akan bergantung pada kerangka pengembangan, platform cloud, keterampilan bahasa, dan kemungkinan investasi yang ada pada alat oleh perusahaan Anda. Untuk meningkatkan kualitas integrasi berkelanjutan dan proses pengiriman berkelanjutan, integrasi alat akan menjadi faktor utama. Rekomendasinya adalah fokus pada bagaimana setiap kerangka kerja, bahasa, alat otomatisasi, dan platform yang Anda gunakan memungkinkan Anda menghilangkan langkah-langkah manual dan memberikan pelaporan serta visibilitas mengenai kesehatan lingkungan build dan runtime Anda.

2.2 DEVOPS DAN AZURE

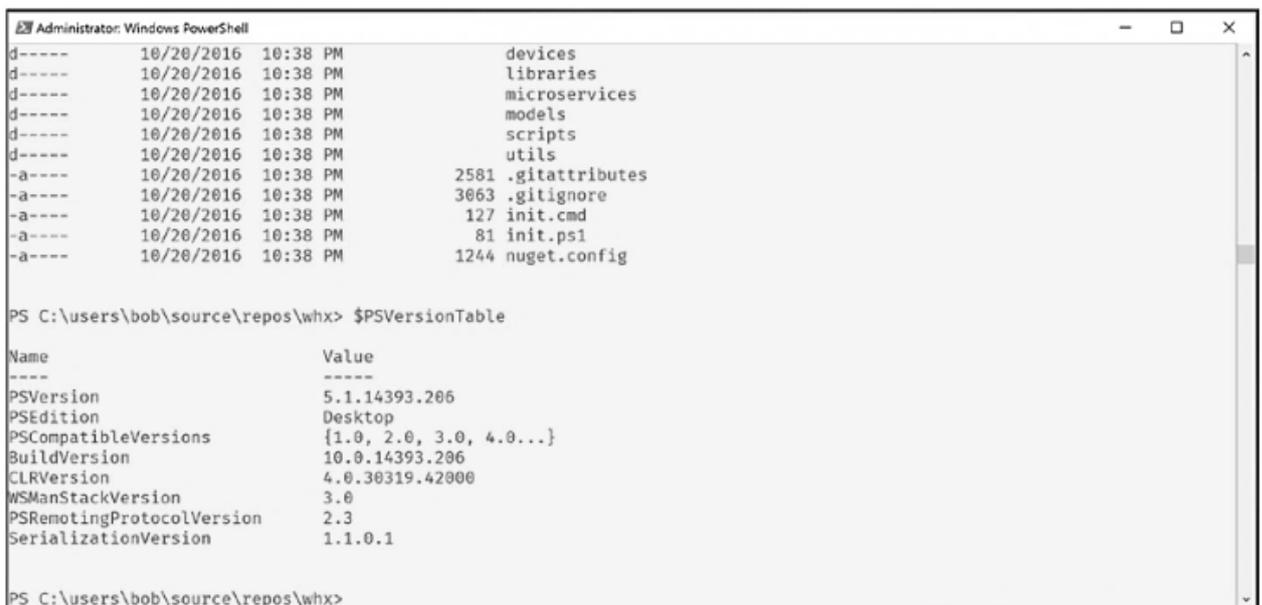
Azure menyediakan serangkaian SDK, API, alat, kerangka kerja, dan utilitas baris perintah untuk mengelola sumber daya Azure dan menentukan serta mengelola proses integrasi berkelanjutan dan pengiriman berkelanjutan Anda. Azure telah berevolusi untuk

menyediakan tidak hanya dukungan untuk alat dan bahasa pengembangan tradisional Microsoft, yaitu.

Visual Studio, .NET Framework, dan C#, tetapi juga memberikan dukungan penuh untuk tim yang ingin memanfaatkan *nix, Java, node.js, Docker, Jenkins, Octopus, Chef, Puppet, Jira, CA Rally atau hampir semua open source lainnya sumber atau produk ISV Continuous Integration/Continuous Deliver (CICD). Baik Anda mengembangkan di .NET, Java, Ruby, Python, PHP, atau Node, tingkat kemampuan yang sama tersedia untuk membuat proses pengiriman berkelanjutan otomatis. Tidak mungkin untuk mencakup semua kemungkinan kombinasi, jadi untuk tujuan kami, kami akan fokus pada penggunaan PowerShell dan Visual Studio Team Services bersama dengan Templat Azure Resource Manager untuk mengimplementasikan proses CICD yang sepenuhnya otomatis yang menyediakan langkah-langkah penyediaan, pembuatan, dan penerapan untuk solusi kami.

PowerShell

Windows PowerShell adalah otomatisasi tugas berorientasi objek dan kerangka manajemen konfigurasi dari Microsoft yang terdiri dari shell baris perintah dan bahasa skrip terkait yang dibangun di .NET Framework. PowerShell telah dimigrasikan ke .NET Core dan menjadi sumber terbuka mulai Agustus 2016, menjadikannya alat otomatisasi yang layak di Mac dan Linux. Konsol PowerShell (lihat Gambar 2.3) adalah utilitas baris perintah yang dapat Anda gunakan untuk menjalankan skrip PowerShell. Saat menggunakan PowerShell untuk mengelola Azure dari desktop, Anda disarankan untuk menjalankan utilitas ini sebagai administrator.



```

Administrator: Windows PowerShell
d----- 10/20/2016 10:38 PM          devices
d----- 10/20/2016 10:38 PM          libraries
d----- 10/20/2016 10:38 PM          microservices
d----- 10/20/2016 10:38 PM          models
d----- 10/20/2016 10:38 PM          scripts
d----- 10/20/2016 10:38 PM          utils
-a----- 10/20/2016 10:38 PM          2581 .gitattributes
-a----- 10/20/2016 10:38 PM          3063 .gitignore
-a----- 10/20/2016 10:38 PM          127 init.cmd
-a----- 10/20/2016 10:38 PM          81 init.ps1
-a----- 10/20/2016 10:38 PM          1244 nuget.config

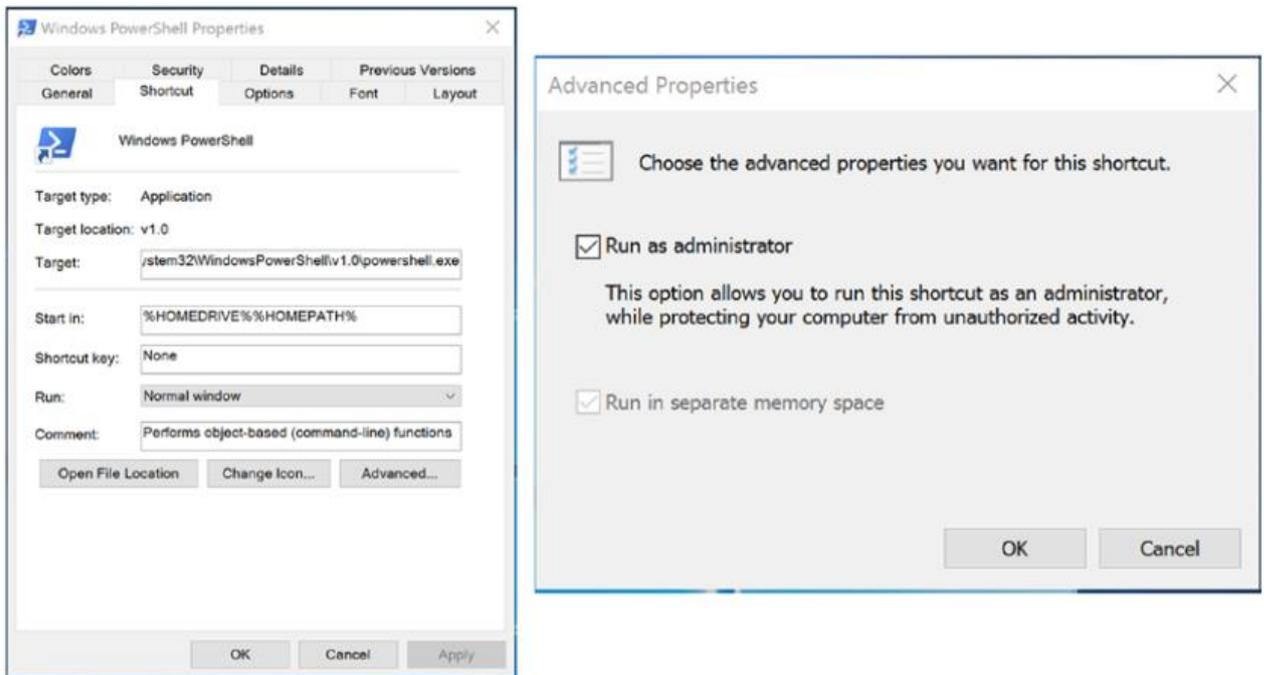
PS C:\users\bob\source\repos\whx> $PSVersionTable

Name                           Value
----                           -
PSVersion                       5.1.14393.206
PSEdition                       Desktop
PSCompatibleVersions             {1.0, 2.0, 3.0, 4.0...}
BuildVersion                    10.0.14393.206
CLRVersion                      4.0.30319.42000
WSManStackVersion               3.0
PSRemotingProtocolVersion       2.3
SerializationVersion            1.1.0.1

PS C:\users\bob\source\repos\whx>

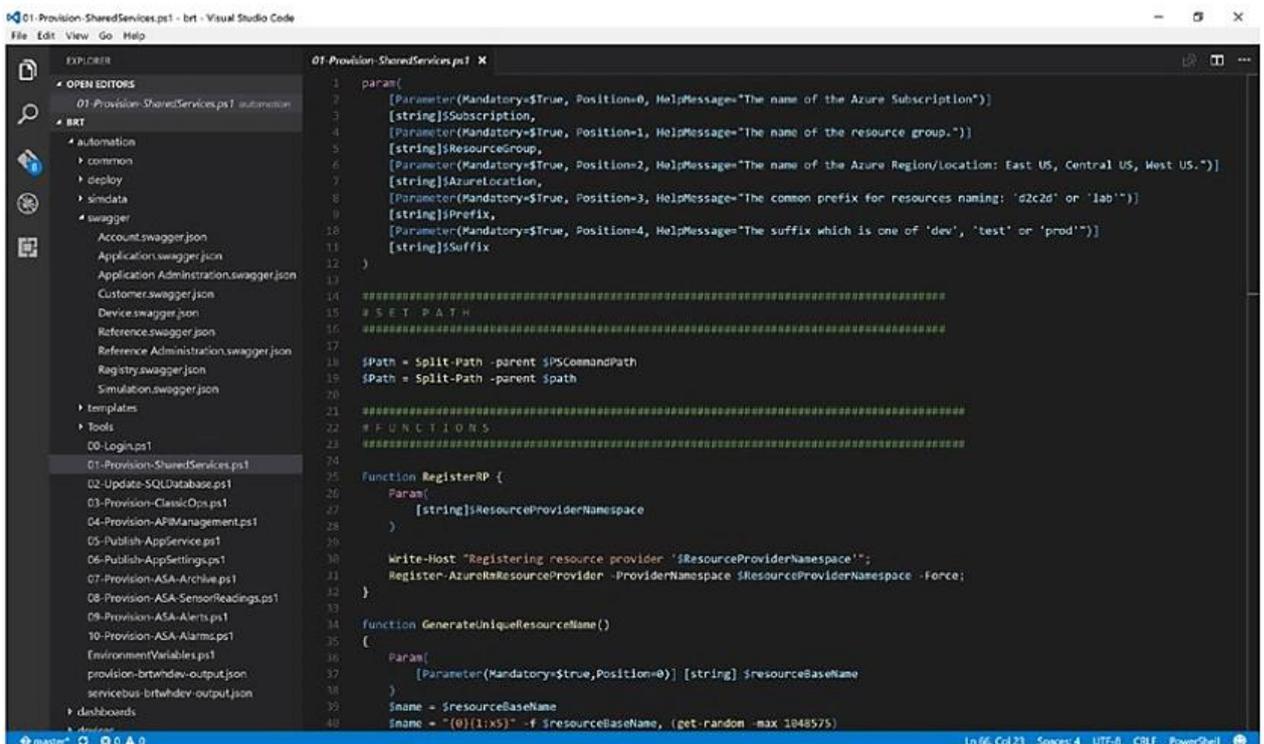
```

Gambar 2.3. Konsol PowerShell



Gambar 2.4. Konfigurasi untuk dijalankan sebagai administrator

Untuk mengembangkan skrip PowerShell, Anda dapat menggunakan Visual Studio, Visual Studio Code (lihat Gambar 2.5) atau editor teks favorit Anda. Visual Studio Code adalah alat lintas platform yang menyediakan pewarnaan sintaksis, integrasi Git, dan dukungan debugging untuk PowerShell serta bahasa pemrograman lain seperti Node.js. Untuk menjalankan skrip PowerShell, disarankan untuk menggunakan konsol PowerShell.



Gambar 2.5. Kode Visual Studio

Konstruksi Skrip

Skrip PowerShell yang disediakan dalam repo Git yang menyertainya menggunakan beberapa teknik PowerShell standar. Teknik pertama adalah sebagian besar skrip akan mengambil sekumpulan parameter masukan standar sehingga dapat digunakan kembali. Di bagian atas setiap skrip, terdapat bagian param yang mendefinisikan lima parameter standar:

```
param(
    [string]$Subscription, #Azure subscription
    [string]$ResourceGroup, #resource group
    [string]$AzureLocation, #region, i.e. East US, Japan
    [string]$Prefix, #the prefix for naming services
    [string]$Suffix #the suffix for naming services
)
```

Berikut adalah contoh mengoper parameter ini ke skrip dari baris perintah:

```
PS C:\> .\somescript.ps1
-subscription bobs-azure
-resourcegroup dev001
-azurelocation "East US"
-prefix bob
-suffix de
```

Skrip PowerShell berada di repositori kode. Terkadang skrip perlu mereferensikan file yang terletak di repositori. Untuk mereferensikan file di seluruh repositori, lokasi jalur saat ini diambil dan kemudian digunakan untuk membuat jalur ke file lain.

```
$Path = Split-Path -parent $PSCmdPath
$Path = Split-Path -parent $Path
```

Variabel PowerShell adalah nama yang dimulai dengan \$. Untuk mendeklarasikan variabel, Anda menggunakan sintaks ini:

```
$storageAccountName = $Prefix + "blobstorage" + $suffix
$storageAccountType = "standard_LRS"
```

Perhatikan penggunaan parameter umum \$Prefix dan \$Suffix. Parameter ini digunakan untuk memastikan nama unik dibuat untuk setiap sumber daya Azure. Pada awal setiap skrip, cache kesalahan dibersihkan dan waktu mulai dicatat sehingga runtime skrip dapat dihitung dan dilaporkan pada akhir skrip.

```
$Error.Clear()

# Mark the start time.
$StartTime = Get-Date

$ErrorActionPreference = "stop"

#.....
```

```
#bagian isi skrip di sini
#.....

#tanda waktu selesai
$FinishTime = Get-Date

#keluaran konsol
$TotalTime = ($FinishTime - $StartTime).TotalSeconds
Write-Verbose -Message "Elapse Time (Seconds): $TotalTime" -Verbose
```

Azure PowerShell

Azure PowerShell adalah lingkungan skrip yang dapat Anda gunakan untuk mengontrol dan mengotomatiskan pembuatan, konfigurasi, dan pengelolaan sumber daya Azure serta kode dan konfigurasi kustom Anda di Azure. Azure PowerShell menyediakan cmdlet (perintah PowerShell) yang merangkum API ReST Azure. Untuk mengonfigurasi lingkungan Azure PowerShell, unduh dan jalankan instalasi Azure SDK.

Setelah instalasi selesai, jalankan konsol PowerShell sebagai administrator dan jalankan perintah berikut.

```
PS C: \> Set-ExecutionPolicy Unrestricted
```

Pasang Modul Azure Resource Manager:

```
PS C:\> Instal-modul azureRM
PS C: \> Impor-Modul AzureRM
```

Instal Modul Azure Classic:

```
PS C:\> Instal-modul azure
PS C: \> Impor-Modul Azure
```

Memanggil Azure Cmdlet

Cmdlet Azure PowerShell menggabungkan API Azure ReST dan memudahkan penerapan templat Azure Resource Manager (ARM) serta menjalankan perintah untuk mengonfigurasi dan mengelola layanan yang ada. Templat Sumber Daya Azure dibahas secara lebih rinci nanti di bab ini. Misalnya, untuk memilih langganan Azure, Anda akan menggunakan cmdlet `Set-AzureRmContext`:

```
Set-AzureRmContext -SubscriptionName $Subscription;
```

Potongan kode berikut ini akan memeriksa apakah grup sumber daya ada, dan jika tidak ada, grup sumber daya akan membuatnya:

```
$rg = Get-AzureRmResourceGroup
    - Name $ResourceGroup
    - ErrorAction SilentlyContinue
if (!$rg)
{
```

```

New-AzureRmResourceGroup
  - Name $ResourceGroup
  - Location $AzureLocation
}
Else
{
  Write-Verbose"Using existing resource group"
}

```

2.3 MANAJER SUMBER DAYA AZURE

Implementasi referensi terdiri dari banyak layanan dan komponen perangkat lunak yang bekerja sama. Azure Resource Manager (ARM) adalah konstruksi di Azure untuk mengatur dan menghubungkan sumber daya untuk solusi Anda dan menyediakan kemampuan untuk membuat, mengonfigurasi, memperbarui, dan menghapusnya sebagai satu unit. Mari kita definisikan terminologinya:

- ❖ *Sumber Daya*: Item yang dapat dikelola dan tersedia melalui Azure. Beberapa sumber daya umum adalah mesin virtual, akun penyimpanan, aplikasi web, database, jaringan virtual, dll.
- ❖ *Grup Sumber Daya*: Kumpulan sumber daya terkait yang diberi nama untuk solusi Azure. Grup sumber daya dapat menyertakan semua sumber daya untuk solusi, atau hanya sumber daya yang ingin Anda kelola sebagai grup. Anda memutuskan bagaimana Anda ingin mengalokasikan sumber daya ke grup sumber daya berdasarkan apa yang paling masuk akal bagi organisasi Anda. Bukan hal yang aneh untuk menunjuk kelompok sumber daya untuk pengembangan, pengujian, pementasan, dan produksi melalui konvensi penamaan.
- ❖ *Penyedia Sumber Daya*: Layanan yang memasok sumber daya yang dapat Anda sebarkan dan kelola melalui manajer sumber daya. Setiap penyedia sumber daya menawarkan operasi untuk bekerja dengan sumber daya yang disebarkan. Beberapa penyedia sumber daya umum adalah Microsoft.Compute, yang memasok sumber daya mesin virtual, Microsoft. Penyimpanan, yang memasok sumber daya akun penyimpanan, dan Microsoft.Web, yang memasok sumber daya yang terkait dengan aplikasi web.
- ❖ *Templat Manajer Sumber Daya*: Templat manajer sumber daya, juga disebut sebagai templat ARM, adalah file JavaScript Object Notation (JSON) yang mendefinisikan satu atau beberapa sumber daya untuk disebarkan ke grup sumber daya. Ini juga mendefinisikan ketergantungan antara sumber daya yang dikerahkan. Templat ini dapat digunakan untuk menyebarkan sumber daya secara konsisten dan berulang kali.
- ❖ Ada banyak manfaat mendefinisikan lingkungan Azure Anda menggunakan templat JSON daripada menerapkan semuanya dalam skrip.
- ❖ Templat dapat dibuat versinya, ditambahkan ke repositori kode Anda, dan tetap disinkronkan dengan kode yang mengimplementasikan solusi.
- ❖ Anda dapat memanfaatkan template ini berulang kali dan konsisten sepanjang siklus proses pengiriman berkelanjutan Anda.

- ❖ Anda dapat menerapkan kontrol akses berbasis peran pada sumber daya yang disebarkan untuk menentukan siapa yang dapat mengakses kemampuan apa di lingkungan tersebut.
- ❖ Anda dapat menerapkan tag ke sumber daya Azure individual. Mengkueri lingkungan Azure Anda untuk sumber daya yang diberi tag memberi Anda tampilan yang mencakup seluruh grup sumber daya. Ini dapat berguna jika Anda ingin membuat tampilan penagihan untuk semua sumber daya yang terkait dengan pengujian atau akuntansi sebagai contoh.

Templat ARM

Templat ARM adalah file JSON yang menentukan sumber daya yang akan disediakan dan dikonfigurasi dalam grup sumber daya. File template ARM memiliki enam bagian.

Bagian 1. \$schema

Lokasi file skema JSON yang menjelaskan versi bahasa templat.

"\$schema":

<https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#>

Bagian 2. Versi Konten

Versi templat (misalnya 1.0.0.0).

"contentVersion": "1.0.0.0"

Bagian 3. Parameter

Mendefinisikan parameter input ke template. Penggunaan parameter masukan yang baik membuat skrip Anda dapat digunakan kembali dan dinamis. Di sini kami menentukan parameter input untuk lokasi Azure, nama akun penyimpanan, dan sebagainya. Parameter masukan ini membuat skrip menjadi dinamis dan dapat digunakan kembali. Nilai dapat diakses di area lain templat menggunakan notasi ini: `[parameters('param-name')]`.

```
"parameters": {
  "azureLocation": {"type": "string"},
  "storageAccountName": {"type": "string"},
  "servicebusNamespace": {"type": "string"},
  "docDbAccount": {"type": "string"},
  ...
}
```

Bagian 4. Variabel

Fragmen skrip JSON yang dapat digunakan kembali yang menyediakan pengaturan umum atau penyederhanaan konstruksi kode yang kompleks. Variabel bernama ini dapat diakses di area lain templat menggunakan notasi: `[variabel('nama-variabel')]`.

```
"variables": {
  "iotHubResourceId":
    "[resourceId(
```

```

    'Microsoft.Devices/Iothubs',parameters('iotHubName')]]",
    "iotHubKeyName": "iothubowner",
    ...

```

Bagian 5. Sumber Daya

Bagian templat ini mencantumkan sumber daya yang ingin Anda provisi di Azure, seperti instans DocumentDB, namespace bus layanan, atau situs web. Untuk mengidentifikasi jenis sumber daya, gunakan sintaksis penyedia sumber daya, misalnya:

```
'Microsoft.Storage/storageAccounts'
```

Atau

```
'Microsoft.DocumentDb/databaseAccount'.
```

Berikut cuplikan bagian sumber daya di templat ARM:

```

"resources": [
  {
    "type": "Microsoft.Storage/storageAccounts",
    "name": "[parameters('storageAccountName')]",
    "apiVersion": "2015-06-15",
    "location": "[parameters('azureLocation')]",
    "properties": {"accountType":
      "[parameters('storageAccountType')]"}
  },
  ...

```

Bagian 6. Keluaran

Bagian output memungkinkan Anda meminta pengaturan yang dihasilkan dari pembuatan sumber daya Azure sehingga Anda dapat menggunakannya kembali dengan skrip dan templat lain. Nilai-nilai ini biasanya berupa item seperti string koneksi dan merupakan kunci kebijakan akses bersama.

```

"outputs":{
  "iotHubHostName":{
    "type":"string",
    "value":"[reference(
      Variables('iotHubResourceId')).hostName]"
  },
  "iotHubKey"
  "type":"string"
  "value":"[listkeys(
    Variables('iotHubKeyResource'),
    Variables('iotHubVersion')).primaryKey]"

```

```
},
...
```

Untuk menjalankan templat, Anda harus terlebih dahulu membuat file JSON parameter yang cocok yang berisi parameter input seperti yang ditentukan oleh templat. Kode PowerShell berikut akan menghasilkan file JSON yang berisi parameter input untuk templat ARM dan kemudian meneruskan file tersebut bersama templat ke cmdlet **New-AzureRMResourceGroupDeployment**. Ketika perintah selesai, parameter output ditulis ke file dengan **provision-[ResourceGroupName]-output.json**.

```
Try
{
  $JSON = @"
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",

  "parameters": {
    "azureLocation": {"value": "$AzureLocation"},
    "storageAccountName": {"value": "$storageAccountName"},
    "storageAccountType": {"value": "$storageAccountType"},
    "servicebusNamespace": {"value": "$servicebusNamespace"},
    "docDbAccount": {"value": "$databaseAccount"},
    "iotHubName": {"value": "$iotHubName"},
    "prefix":{"value":"$Prefix"},
    "suffix":{"value":"$Suffix"},
  }
}
"@
  $ParamsPath = $Path
    + "\Automation\Templates\whx-arm-template-params.json

  # write the parameters file to disk
  $JSON | Set-Content -Path $ParamsPath

  $TemplatePath = $Path
  + "\Automation\Templates\whx-arm-template.json"
  $OutputPath = $Path
  + "\Automation\provision-$ResourceGroup-output.json"

  # validate the template files
  Test-AzureRmResourceGroupDeployment
    -ResourceGroupName $ResourceGroup
    -TemplateFile $TemplatePath
    -TemplateParameterFile $ParamsPath

  # perform the deployment
  New-AzureRmResourceGroupDeployment
    -ResourceGroupName $ResourceGroup
    -TemplateFile $TemplatePath
    -TemplateParameterFile $ParamsPath
```

```

    | ConvertTo-Json
    | Out-File "$OutputPath"
}
Catch
{
    Write-Verbose -Message $Error[0].Exception.Message
    Write-Verbose
    -Message "Exiting due to exception: ARM Template Failed."
}

```

File keluaran akan berisi parameter masukan ke templat serta setiap pengaturan keluaran yang diminta seperti yang diminta oleh templat. Dengan menyimpan output sebagai file JSON, skrip lain apa pun dalam proses penyediaan, pembuatan, dan penerapan dapat mengurai file dan mengakses nilai yang disimpan. Untuk menyediakan kemampuan penguraian ini untuk skrip PowerShell lainnya dalam solusi kami, Anda dapat membuat skrip PowerShell yang memuat file JSON keluaran dan menginisialisasi objek PowerShell untuk memudahkan akses pengaturan.

```

$provisionOutputPath = $Path
+ "\automation\provision-$ResourceGroup-output.json"
$provisionInfo = ConvertFrom-Json
-InputObject (Gc $provisionOutputPath -Raw)

```

Untuk memasukkan skrip ini ke file lain, Anda dapat menggunakan perintah PowerShell ini:

```

$includePath = $Path + "\Automation\EnvironmentVariables.ps1"
."$includePath

```

Untuk mengakses salah satu parameter masukan atau nilai keluaran dari eksekusi templat ARM, gunakan objek `$provisionInfo` sebagai berikut:

```

$IoTHubHostName = $provisionInfo.Outputs.iotHubHostName.Value
$IoTHubKey = $provisionInfo.Outputs.iotHubKey.Value
$IoTHubConnectionString =
    $provisionInfo.Outputs.iotHubConnectionString.Value
$IoTHubname = $provisionInfo.Parameters.iotHubname.Value

```

Sekarang setelah Anda memahami cara cmdlet Azure PowerShell dan templat ARM digunakan, bagian selanjutnya dari bab ini akan membawa Anda melalui serangkaian latihan yang akan menggunakan repo Git bersama dengan Visual Studio Team Services untuk mengatur provisi, membangun, dan penerapan proses CI/CD.

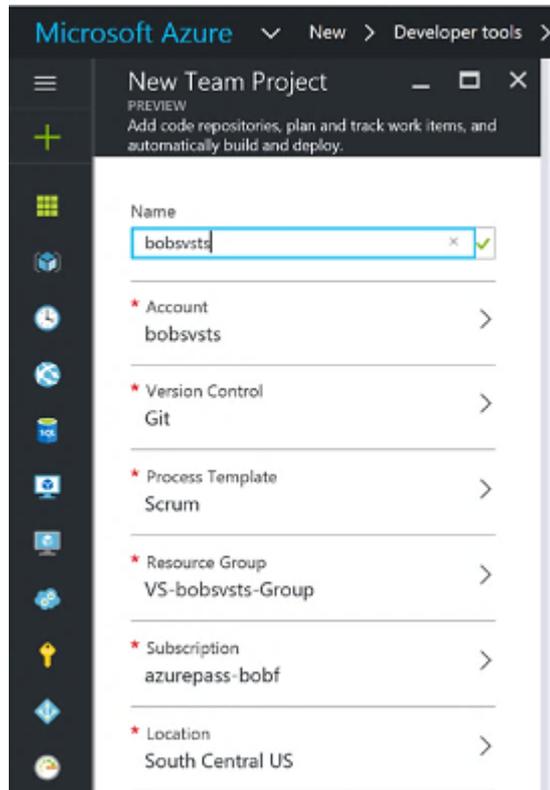
Layanan Tim Visual Studio

Visual Studio Team Services (VSTS) adalah kontrol kode sumber, manajemen proyek, integrasi berkelanjutan, dan penawaran Perangkat Lunak sebagai Layanan (SaaS) pengiriman

berkelanjutan di Azure. Untuk memulai VSTS, Anda cukup membuat instans di langganan Azure Anda, dengan memberikan nama unik untuk instans tersebut.

Buat Instan Layanan Tim Visual Studio

Di Portal Azure, pilih Baru ► Alat Pengembang ► Proyek Tim dan isi formulir dengan nama instans, kontrol versi, dan templat proses Anda (lihat Gambar 2-6).

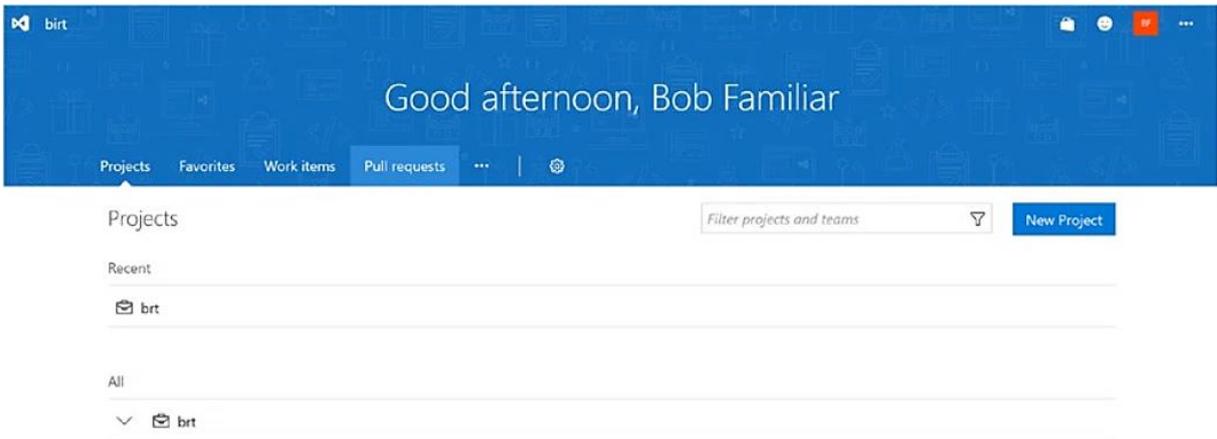


Gambar 2.6. Contoh Layanan Tim Visual Studio baru

Setelah proses ini selesai, Anda dapat menavigasi ke instans VSTS Anda di URL `http://[your-vstshostname].visualstudio.com` dan mulai mengonfigurasi lingkungan pengembangan tim Anda.

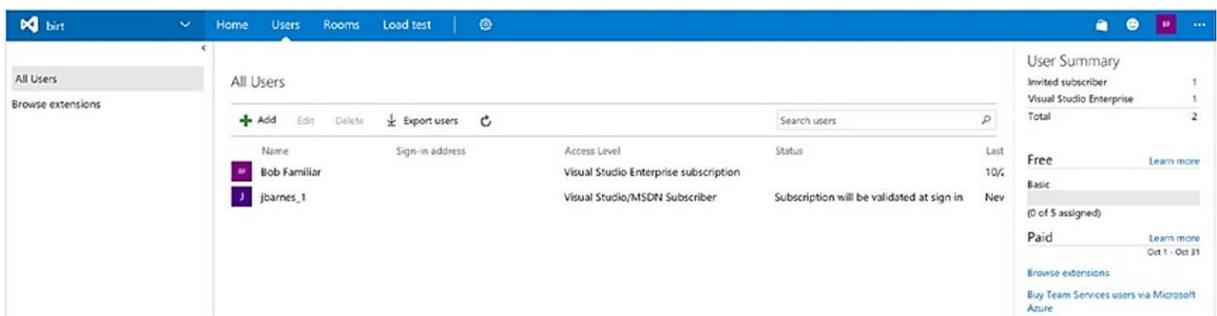
Menambahkan Anggota Tim Ke VSTS

Setelah instance Anda disediakan, Anda dapat memberikan akses ke VSTS untuk rekan satu tim Anda dengan memilih opsi Pengguna dari halaman beranda akun VSTS (lihat Gambar 2-7).



Gambar 2.7. Halaman beranda akun VSTS

Dengan mengklik elipsis (...) di menu, Anda dapat mengakses halaman Pengguna. Anda dapat mendaftarkan pengguna dengan instance VSTS ini di halaman ini. Setelah pengguna terdaftar pada layanan ini, mereka dapat ditambahkan ke masing-masing proyek. VSTS menyediakan lima akun pengguna Basic gratis dan akun pengembang MSDN dalam jumlah tidak terbatas. Anda juga dapat menambahkan akun pemangku kepentingan dalam jumlah tidak terbatas; pemangku kepentingan adalah pengguna yang tidak melakukan tugas pengembangan namun mengakses fitur pelaporan layanan (lihat Gambar 2.8).



Gambar 2.8. Mengelola pengguna VSTS

2.4 MEMBUAT PROYEK VSTS

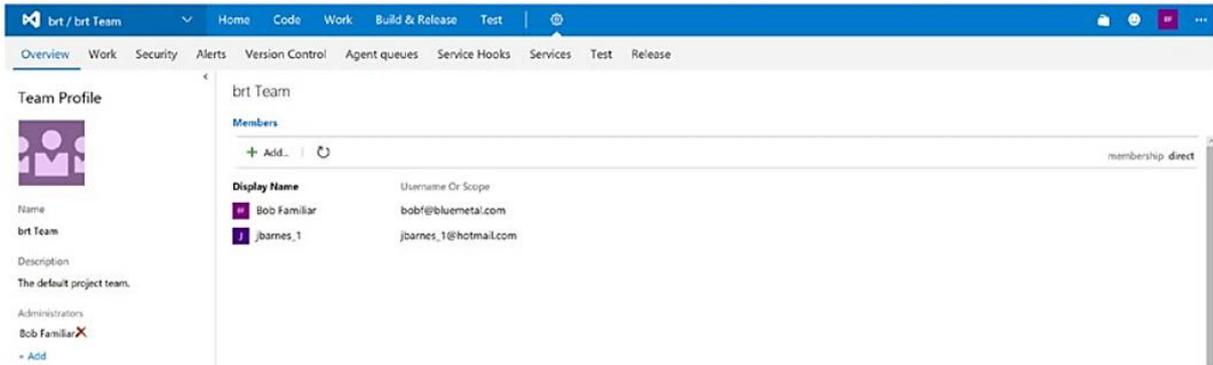
Setelah Anda masuk, Anda disajikan dengan layar Selamat Datang dan dapat mulai membuat repositori proyek, mengkloning repositori tersebut ke lingkungan pengembangan lokal Anda, menambahkan file sumber, dan melakukan perubahan tersebut kembali ke repositori. Saat Anda membuat proyek, Anda juga memiliki pilihan untuk menggunakan format repositori Team Foundation Server (TFS) dan memilih jenis templat proses yang Anda sukai—Agile, Scrum, atau CMMI (lihat Gambar 2.9).

Gambar 2.9. Buat proyek di Visual Studio Team Services

Setelah Anda membuat proyek, Anda perlu menambahkan pengguna ke tim proyek. Dari halaman beranda proyek, klik ikon roda gigi di toolbar untuk membuka halaman konfigurasi proyek (lihat Gambar 2-10).

Gambar 2.10. Halaman beranda proyek VSTS

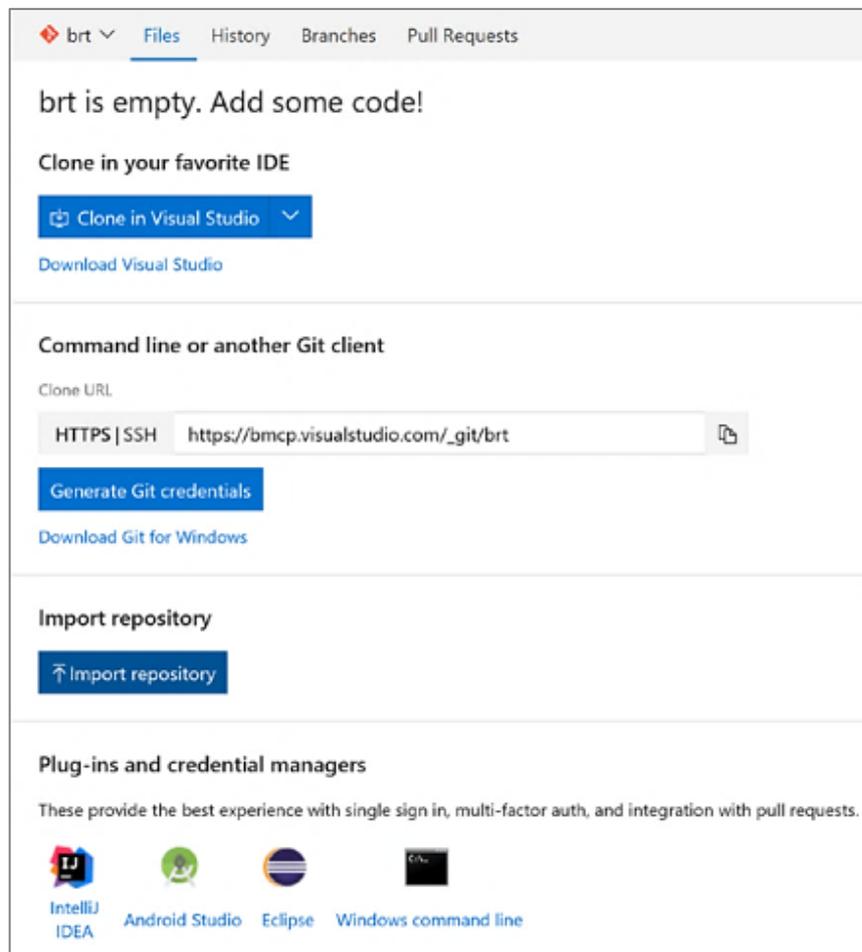
Dari sini Anda dapat menambahkan rekan satu tim ke proyek (lihat Gambar 2-11).



Gambar 2.11. Tambahkan anggota tim proyek

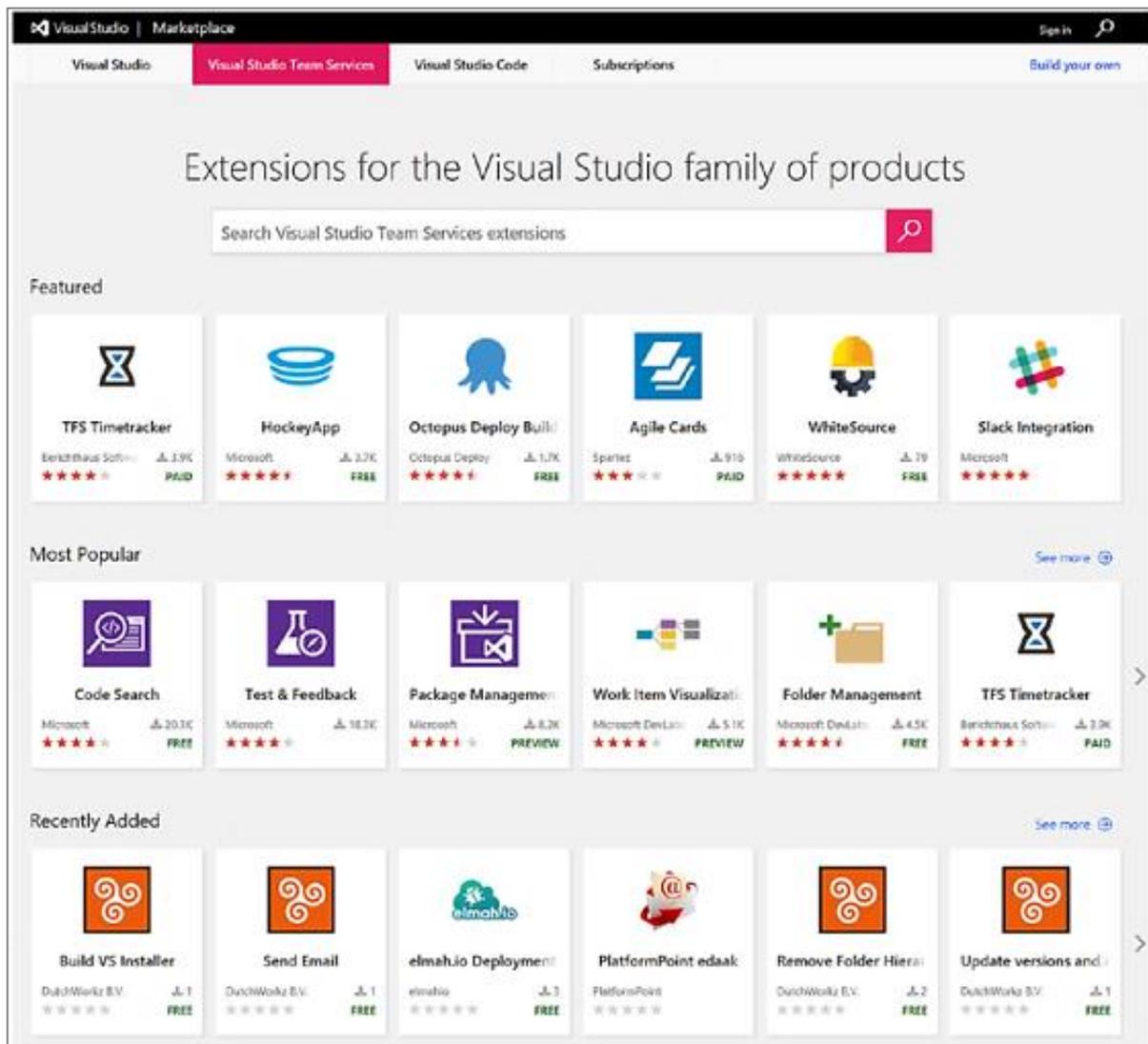
Konfigurasi Repo Git

Jika Anda mengklik opsi menu Kode, Anda akan dibawa ke halaman konfigurasi repositori kode. Anda dapat mengakses titik akhir repositori untuk kloning, mengunduh Git untuk Windows, mengimpor repositori yang ada, dan mengunduh plugin untuk IDE Anda (lihat Gambar 2-12). Setelah Anda mengonfigurasi repo, Anda dapat menggunakan perintah Git atau alat Git lainnya untuk mengkloning repo ke lingkungan lokal Anda dan mulai membuat kode.



Gambar 2.12. Halaman konfigurasi Git untuk proyek VSTS baru

Ada plugin untuk banyak IDE populer termasuk Eclipse, IntelliJ, Android Studio, Visual Studio, Visual Studio Code, dan lainnya (lihat Gambar 2-13).

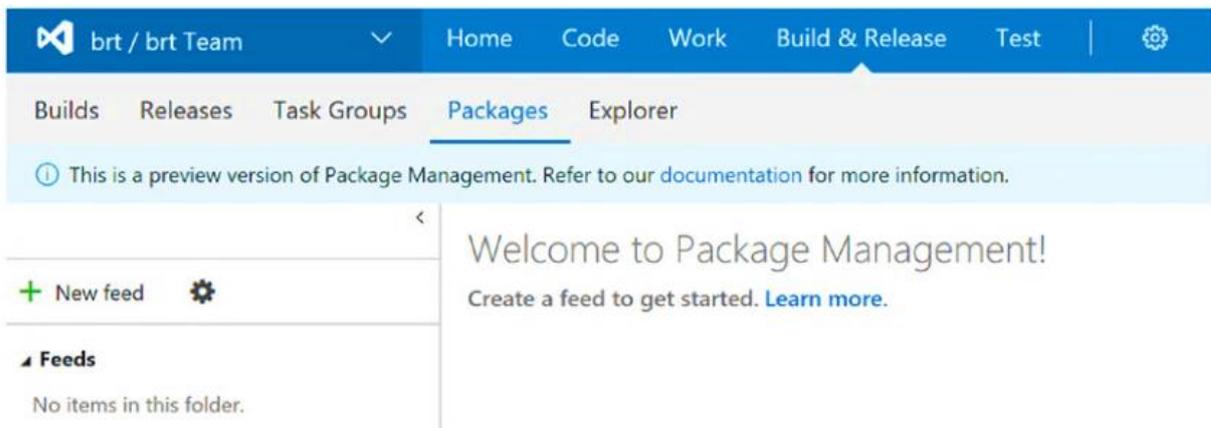


Gambar 2.13. Pasar Visual Studio

Ada pasar ekstensi yang menyediakan integrasi dengan Octopus, Git Hub, Slack, Trello, dan banyak lagi. Anda juga dapat membuat ekstensi Anda sendiri.

Konfigurasi Feed Paket

VSTS memiliki ekstensi manajemen paket yang memungkinkan Anda menyiapkan umpan repositori yang dihosting Azure yang dapat digunakan untuk mengaktifkan alur kerja pengiriman berkelanjutan. Layanan ini dapat menghosting komponen dan paket serta membuatnya tersedia untuk build dan rilis Anda. Pada saat penulisan ini, ekstensi Manajemen Paket mendukung NuGet, dan terdapat rencana untuk mendukung jenis paket tambahan seperti Docker, Maven, dan lainnya di masa mendatang (lihat Gambar 2-14).

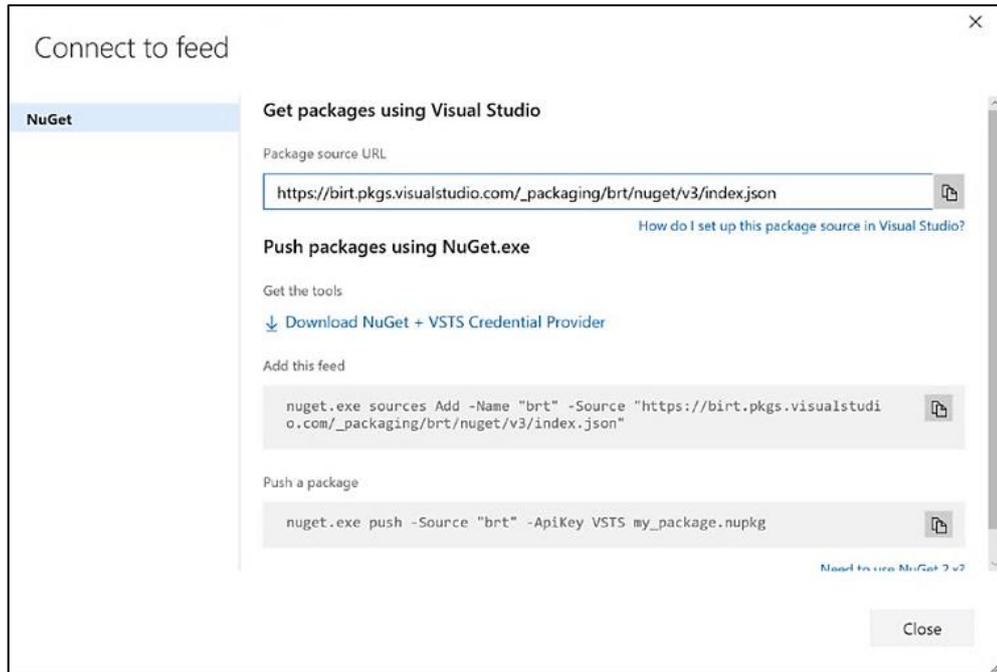


Gambar 2.14. Umpan paket Layanan Tim Visual Studio

Setelah ekstensi Manajemen Paket dipasang, kunjungi halaman Paket dan klik Umpan Baru untuk membuat umpan NuGet Anda. Tetapkan nama, tambahkan deskripsi, dan tentukan siapa yang dapat menyumbangkan paket (lihat Gambar 2.15).

Gambar 2.15. Buat umpan paket baru

Setelah Anda membuat umpan, Anda akan dapat mengakses detail koneksi. Klik tombol Hubungkan ke Umpan untuk menampilkan kotak dialog Hubungkan ke Umpan. Di sini Anda akan melihat URL sumber paket serta mengunduh penyedia kredensial VSTS (lihat Gambar 2-16).

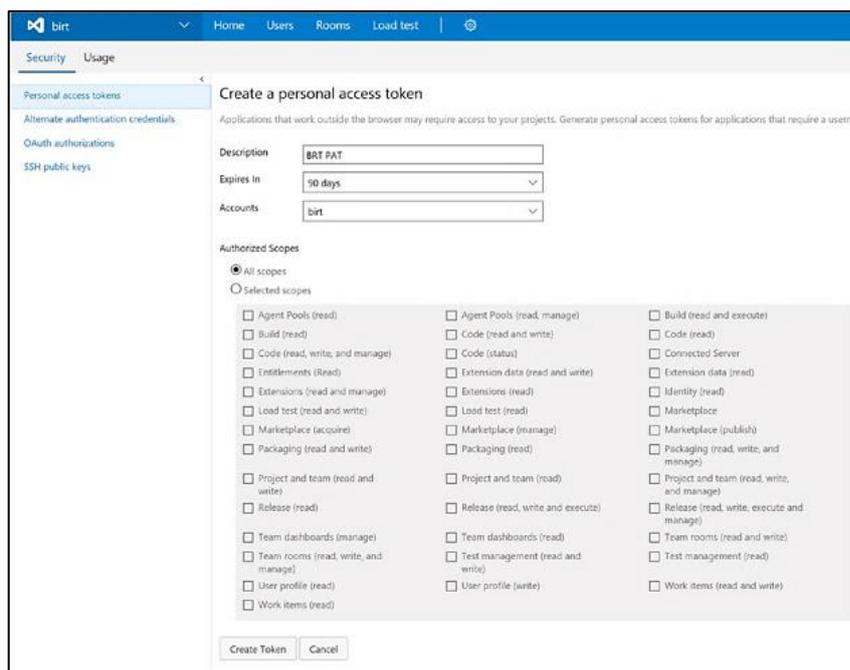


Gambar 2.16. Hubungkan ke umpan

Hal ini diperlukan agar lingkungan pengembangan lokal dapat mengautentikasi ke feed. Setiap anggota tim pengembangan perlu melakukan ini di lingkungannya sendiri.

Konfigurasi Feed Paket Nuget

Untuk mengautentikasi ke feed paket, Anda perlu membuat token akses pribadi dan menyiapkan NuGet.config untuk solusinya. Dari halaman beranda Akun VSTS, klik akun Anda di toolbar dan pilih Keamanan. Pilih Token Akses Pribadi lalu klik Tambah. Beri nama token Anda dan klik Buat Token (lihat Gambar 2-17).



Gambar 2.17. Buat token akses pribadi

Salin token yang dihasilkan dan sisihkan karena Anda akan menggunakannya pada langkah berikutnya. Menggunakan konsol PowerShell, navigasikan ke folder tingkat atas repo Anda dan jalankan perintah ini:

```
PS C:\> Invoke-WebRequest https://dist.nuget.org/win-x86-commandline/latest/nuget.exe -OutFile nuget.exe
```

Perintah ini akan mengunduh salinan `NuGet.exe`. Tambahkan file `NuGet.config` ke root repo Anda. Berikut adalah file dasar `NuGet.config` yang dapat Anda gunakan. Simpan ini ke root repo Anda.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <packageSources>
    <clear />
    <add key="vss-package-management"
      value="https://www.myget.org/F/vss-packagemanagement/api/v2"/>
    <add key="nuget.org"
      value="https://api.nuget.org/v3/index.json" />
  </packageSources>
  <activePackageSource>
    <add key="All" value="(Aggregate source)" />
  </activePackageSource>
</configuration>
```

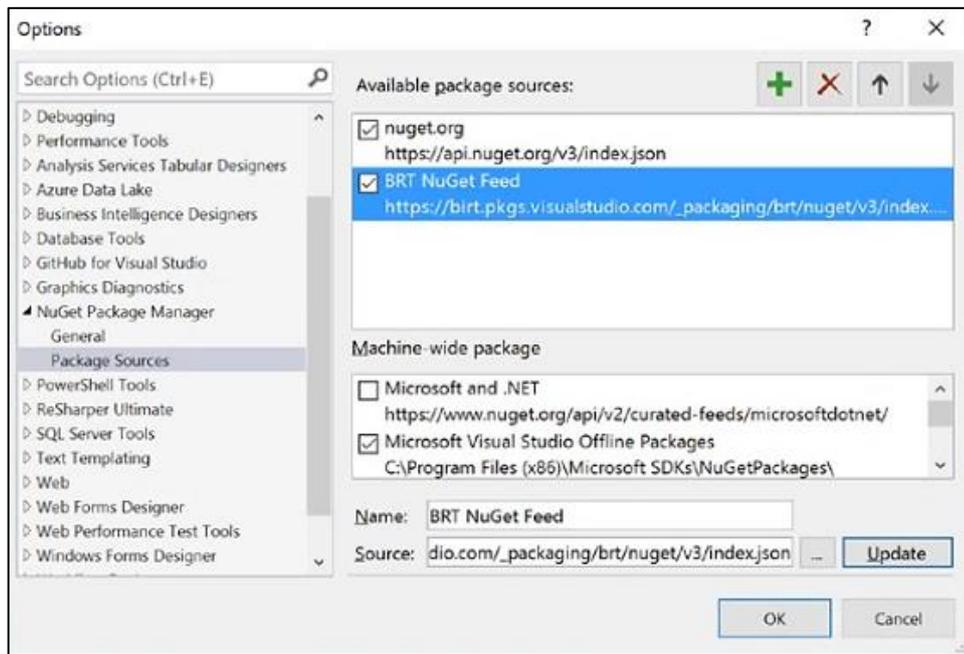
Selanjutnya, jalankan perintah berikut untuk menambahkan feed Anda ke file `NuGet.config` dan untuk menyimpan kredensial autentikasi feed Anda:

```
PS C:\> .\nuget.exe sources add -name {your feed name}
-source {your feed URL}
```

Anda akan dipromosikan untuk mendapatkan nama pengguna. Ini bisa berupa apa saja. Selanjutnya Anda dipromosikan untuk mendapatkan kata sandi. Salin dan tempel token akses pribadi Anda dan tekan Enter. Umpan Anda akan ditambahkan ke file `NuGet.config` dan lingkungan pengembangan Anda akan secara otomatis diautentikasi ke umpan untuk mengunduh dan menerbitkan paket. Terakhir, hapus salinan `NuGet.exe` dari direktori Anda saat ini.

```
PS C:\> rm nuget.exe
```

Untuk menambahkan feed ini ke Visual Studio, dari menu klik **Tools** ► **NuGet Package Manager** ► **Package Manager Settings**. Klik tanda + untuk menambahkan sumber paket dan tempelkan URL sumber paket ke kolom Sumber. Beri nama yang bersahabat (lihat Gambar 2.18).



Gambar 2.18. Pengaturan Manajer Paket

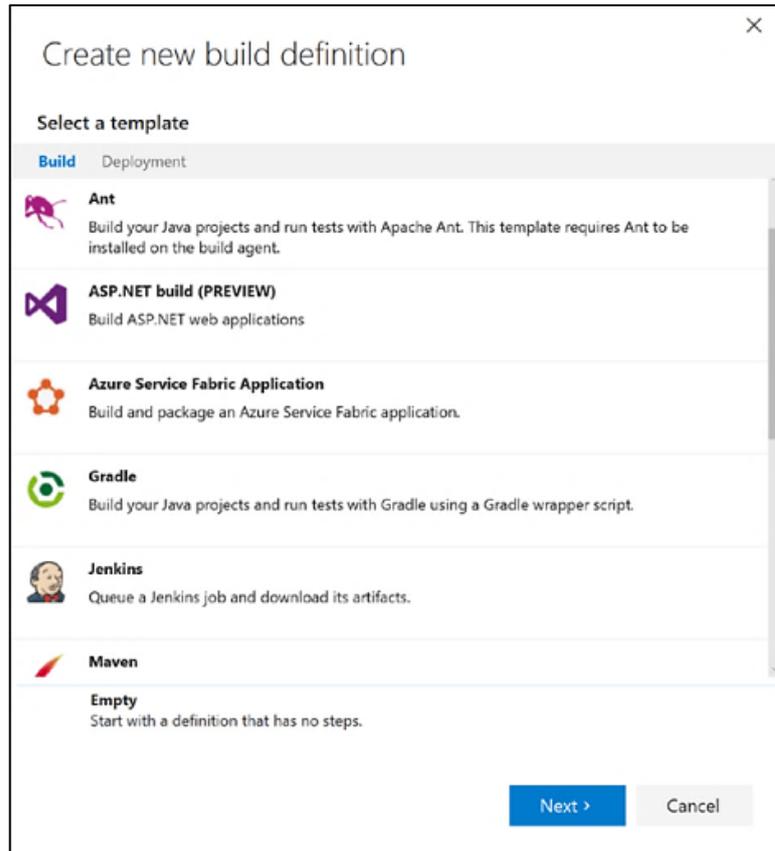
Umpan paket menyediakan satu sumber paket NuGet berversi untuk tim Anda, sehingga mendorong penggunaan kembali dan kolaborasi. Pada latihan berikutnya, Anda akan membuat paket NuGet dan mempublikasikannya ke feed yang baru dikonfigurasi ini.

2.5 MENCIPTAKAN DEFINISI OTOMATISASI

Sekarang setelah lingkungan pengembangan Anda dikonfigurasi, Anda dapat mulai membuat definisi otomatisasi yang akan menentukan proses integrasi berkelanjutan dan pengiriman berkelanjutan.

Repo memiliki folder berikut:

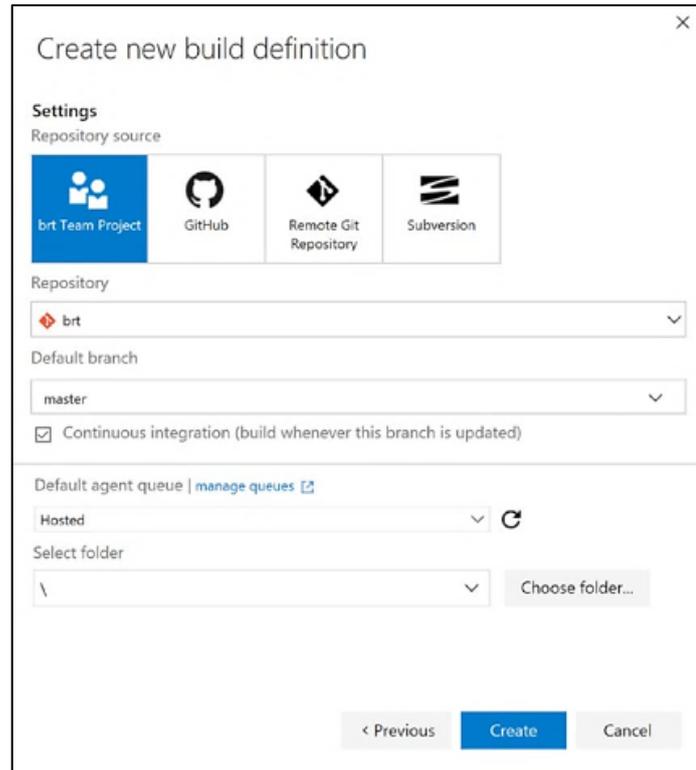
- *Otomatisasi*: skrip PowerShell dan templat ARM
- *Dasbor*: Dasbor pelanggan dan karyawan serta dasbor operasi
- *Perangkat*: Simulator perangkat
- *Perpustakaan*: Perpustakaan umum untuk ReST dan DocumentDB
- *Microservices*: Solusi dan layanan lintas sektoral
- *Model*: Model data yang digunakan oleh perangkat dan layanan mikro
- *Utilitas*: Aplikasi utilitas



Gambar 2.19. Buat daftar templat definisi build baru

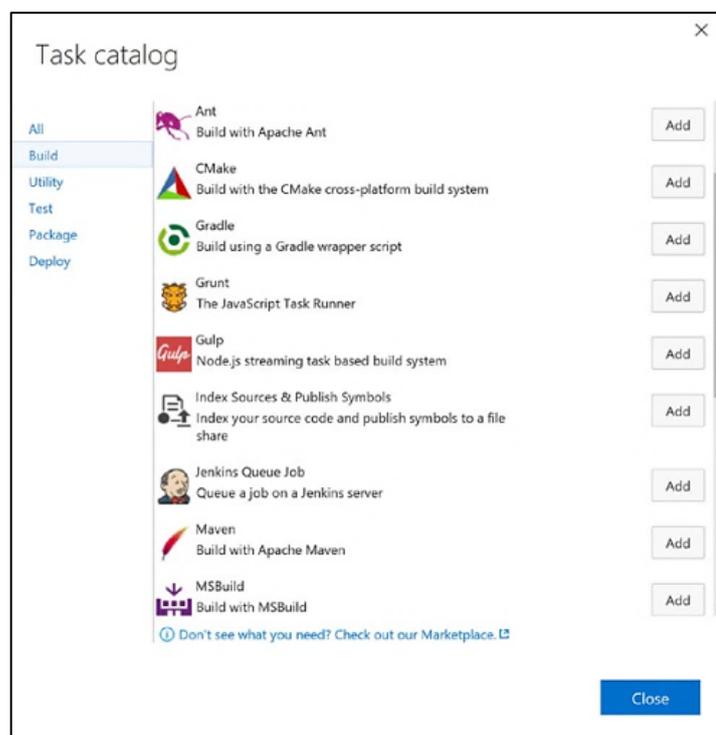
Tugas pertama adalah membuat definisi otomatisasi yang membangun paket Library dan Model NuGet. Definisi build akan dijalankan terhadap kode sumber yang telah dikomit ke repo dan akan memanfaatkan sumber daya komputasi yang disediakan secara otomatis di Azure untuk menjalankan langkah-langkah build. Klik Build & Release di bilah menu atas VSTS lalu klik New Definition (lihat Gambar 2.19).

VSTS memiliki beberapa templat bawaan. Untuk skenario ini, pilih templat kosong dan klik tombol Berikutnya. Pada halaman berikutnya dari wizard, ambil pengaturan default tetapi centang kotak untuk mengaktifkan integrasi berkelanjutan (lihat Gambar 2.20).



Gambar 2.20. Buat pengaturan definisi build baru

Klik Tambahkan langkah Pembuatan. Katalog tugas yang dihasilkan berisi tugas pembangunan, utilitas, pengujian, pengemasan, dan penerapan yang dapat Anda gunakan bersama-sama untuk menentukan pembangunan (lihat Gambar 2-21).

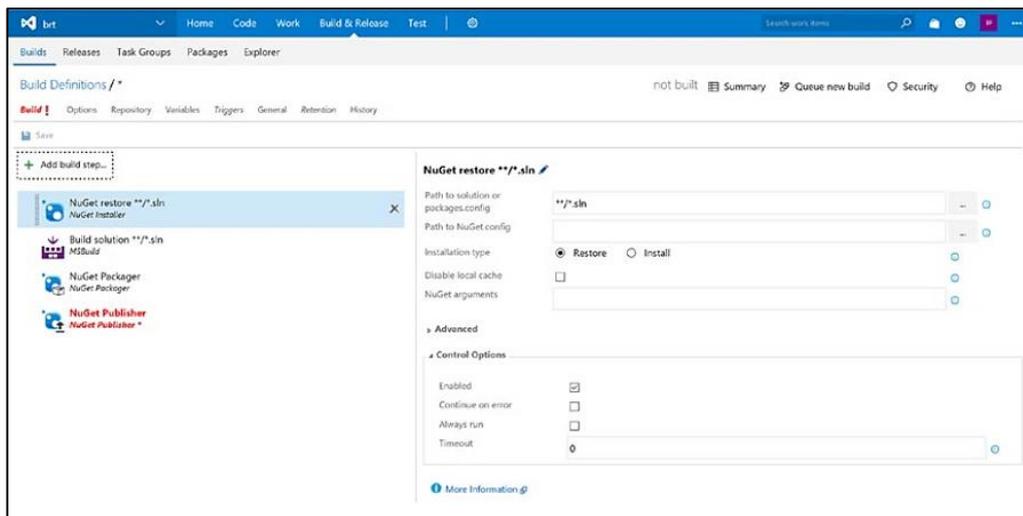


Gambar 2.21. Katalog tugas

Definisi build ini memerlukan empat tugas. Anda dapat memilih masing-masing dari empat tugas saat berada di kotak dialog ini dan menambahkannya ke definisi build.

- Pilih kategori Paket dan tambahkan Tugas Penginstal NuGet
- Pilih kategori Build dan tambahkan Tugas MSBuild
- Pilih kategori Paket dan tambahkan Tugas NuGet Packager
- Tambahkan Tugas Penerbit NuGet
- Klik tombol Tutup

Saat Anda kembali, definisi build Anda akan terlihat seperti yang digambarkan pada Gambar 2.22.

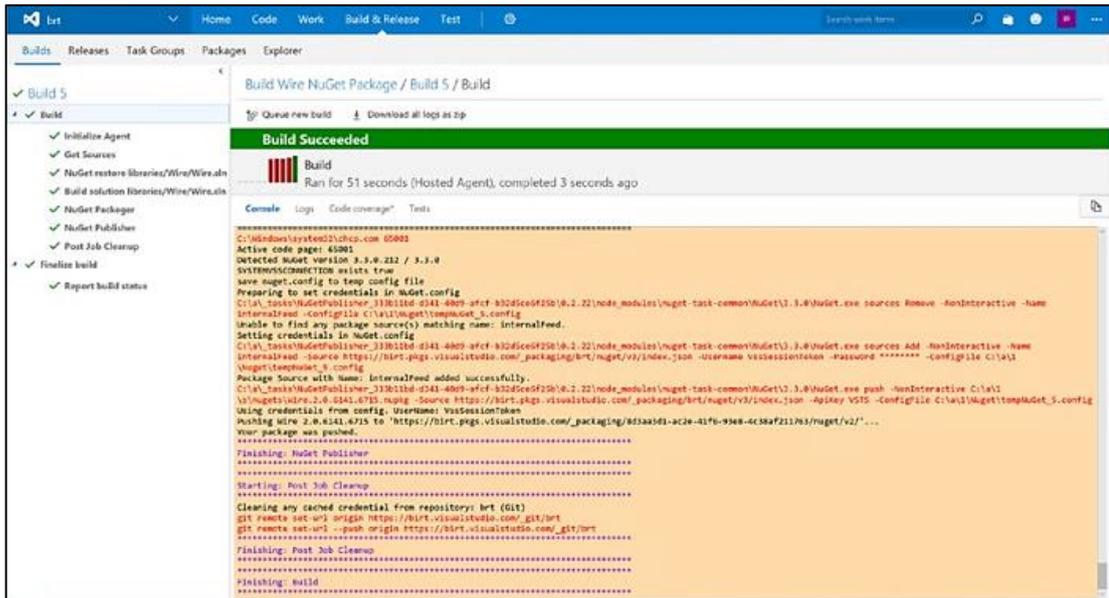


Gambar 2.22. Bangun definisi yang sedang dikembangkan

Setiap tugas memiliki kumpulan pengaturan. Isi pengaturan untuk setiap tugas seperti yang ditunjukkan pada Tabel 2-1 untuk menentukan build perpustakaan kawat. Definisi build akan menghasilkan paket NuGet yang dipublikasikan ke feed paket yang Anda buat sebelumnya. Perhatikan bahwa beberapa pengaturan disembunyikan dan Anda perlu mengklik tombol Lanjutkan untuk menampilkan pengaturan tambahan.

Tabel 2-1. Definisi Pembuatan Paket Wire NuGet

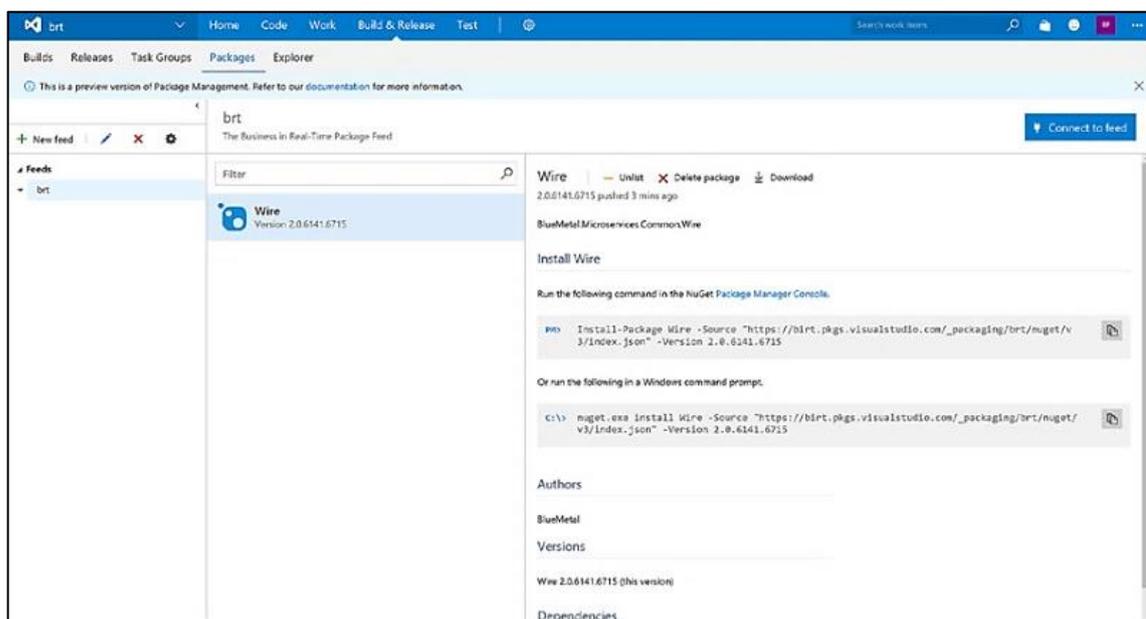
NuGet Installer Settings	
Path to Solution	libraries/Wire/Wire.sln
Path to NuGet.Config	nuget.config
NuGet Arguments	-outputdirectory packages
MSBuild Settings	
Path to Solution	libraries/Wire/Wire.sln
Configuration	debug
NuGet Packager	
Path to CSProj	libraries/Wire/Wire/Wire.csproj
Package Folder	nugets
Include Referenced Projects	Yes (checked)
Configuration to Package	debug
NuGet Publisher	
Path to NuPkg	nugets/wire*.nupkg
Feed Type	Internal
Internal Feed URL	[URL to your package feed]



Gambar 2.23. Membangun kesuksesan

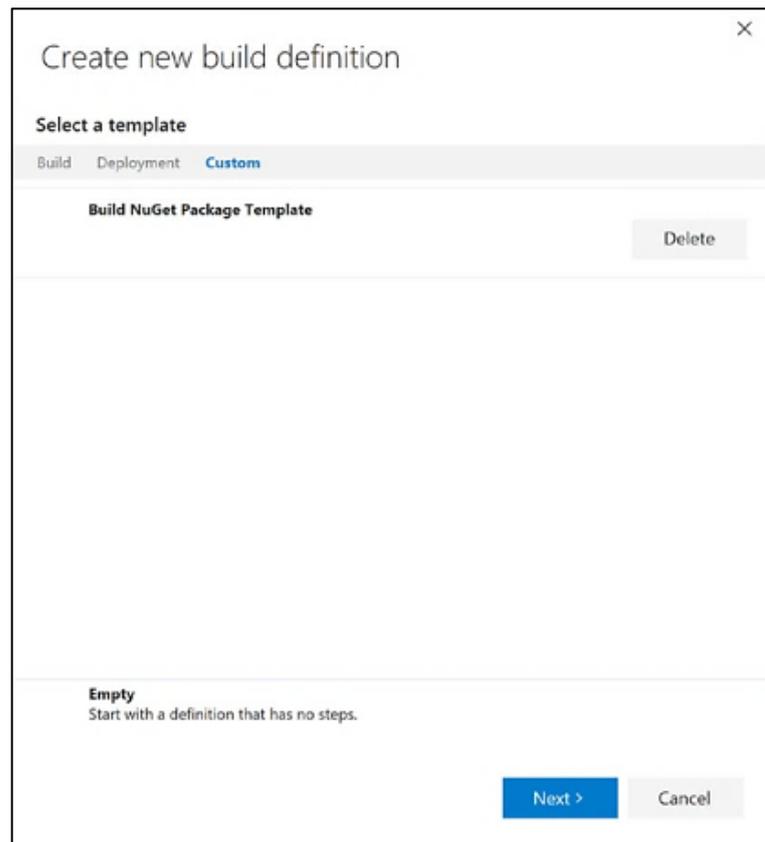
Klik Save dan beri nama seperti Build Wire NuGet Package. Untuk menguji definisi build, klik link Antrean Build Baru di sisi kiri atas layar. Permintaan pembangunan akan dimasukkan ke dalam antrean dan akan dijalankan ketika agen pembangunan telah tersedia. Agen pembangunan adalah komputer virtual yang dihosting Azure yang disediakan sesuai permintaan untuk menjalankan definisi pembangunan Anda. Definisi build akan mengantri dan memulai proses build setelah agen build tersedia. Jika ada kesalahan build, Anda dapat menggunakan output konsol untuk men-debug masalah apa pun dengan pengaturan tugas (lihat Gambar 2.23). File log build yang disediakan juga disediakan.

Paket feed Anda sekarang berisi paket Wire NuGet, seperti yang ditunjukkan pada Gambar 2.24.



Gambar 2.24. Paket Wire NuGet diterbitkan

VSTS menyediakan fasilitas templating sehingga Anda dapat mendesain definisi build satu kali dan menggunakannya kembali untuk langkah build serupa. Pada halaman Build Definition, klik elipsi di sebelah definisi `Build Wire NuGet Package` untuk menampilkan menu dan pilih `Save As a template` beri nama template seperti `Build NuGet Package Template` dan klik OK. Selanjutnya klik tombol `+ Baru` dan klik `Kustom` di toolbar kotak dialog. Anda akan melihat template yang baru Anda buat. Pilih template dan klik `Next` lalu klik `Create` (lihat Gambar 2-25).



Gambar 2.25. Buat definisi build baru dari templat kustom

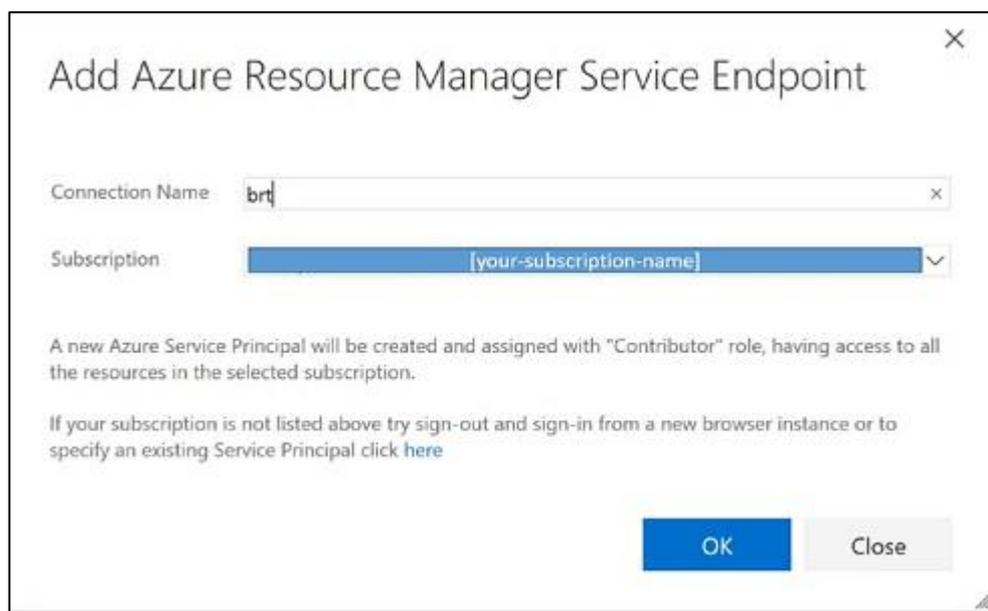
Perbarui pengaturan agar mengarah ke perpustakaan Toko di folder `libraries/store`. Simpan definisi dan antri build untuk menerbitkan Paket Store NuGet. Tambahkan definisi build untuk model Referensi, yang ada di folder `models/reference`. Ini akan membuat paket NuGet yang mewakili model data untuk layanan mikro Referensi. Setelah Anda menyelesaikan latihan ini, Anda akan memiliki tiga paket NuGet di feed Anda—`Wire`, `Store`, dan `Referensi`.

Buat Definisi Bangunan Penyediaan

Definisi build provisi menggunakan skrip PowerShell dan templat ARM untuk membuat dan mengonfigurasi sumber daya Azure dasar yang diperlukan solusi kami. Untuk latihan ini, kami membuat definisi build yang mengeksekusi skrip yang membuat grup sumber daya kami dan menyediakan layanan bersama berikut:

- Ruang nama bus layanan
- Akun penyimpanan
- DokumenDB
- Basis data SQL
- Pusat IoT

Sebelum membuat definisi build ini, Anda perlu mengonfigurasi titik akhir layanan dari lingkungan VSTS ke langganan Azure Anda sehingga definisi build dapat tersambung ke langganan tersebut dan menjalankan templat PowerShell dan ARM. Arahkan kursor ke ikon roda gigi di toolbar dan pilih Layanan dari menu tarik-turun. Klik + Tambahkan titik akhir Layanan Baru dan pilih Azure Resource Manager dari daftar (lihat Gambar 2.26).

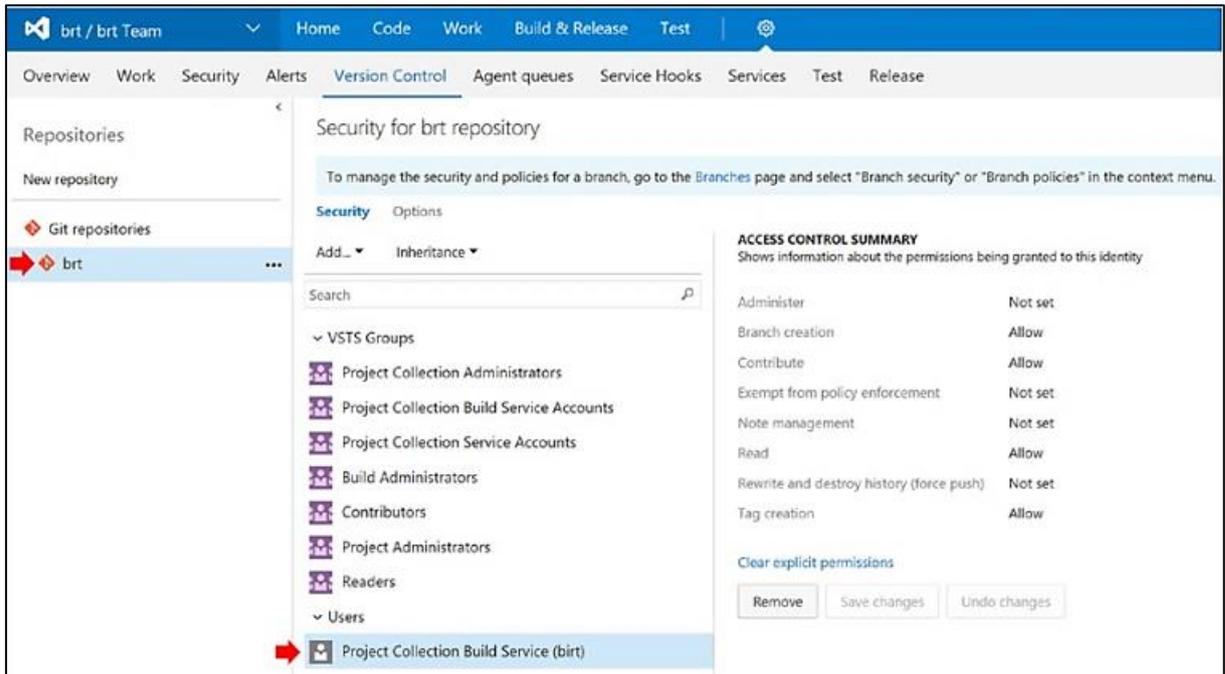


Gambar 2.26. Tambahkan Titik Akhir Layanan Azure Resource Manager

Pilih langganan Anda dari drop-down dan berikan nama yang ramah. Klik Oke. Definisi build ini akan menghasilkan file output yang berisi string koneksi dan kunci ke layanan bersama. Periksa file ini ke dalam repo sehingga dapat digunakan oleh skrip definisi build berikutnya. Untuk melakukannya, berikan izin kepada Agen Pembuat untuk berkontribusi ke repositori.

Klik Kontrol Versi di bilah menu dan kemudian nama repo Git Anda. Pilih Project Collection Build Service dan izinkan izin berikut, lalu simpan perubahan Anda (lihat Gambar 2-27).

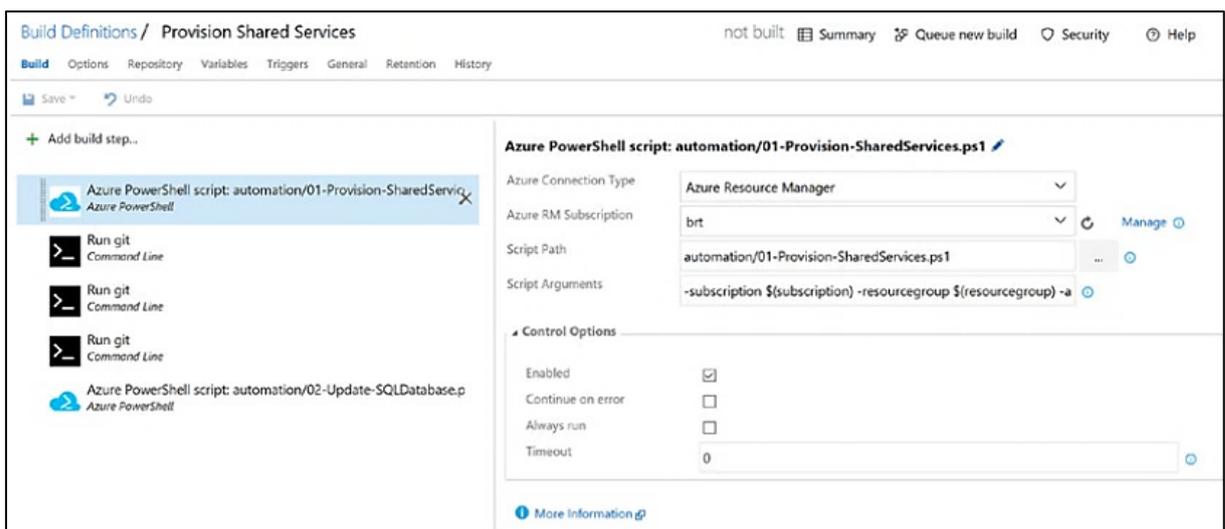
- Pembuatan cabang: Izinkan
- Berkontribusi: Izinkan
- Baca: Izin yang diwariskan
- Pembuatan Tag: Izin yang diwariskan



Gambar 2.27. Konfigurasi keamanan pada repositori

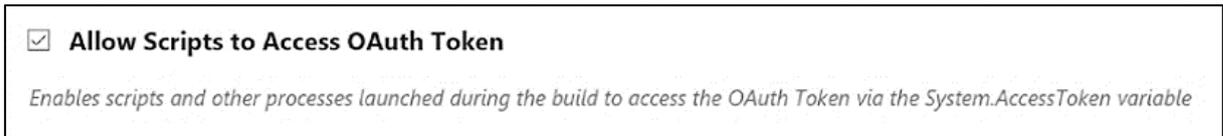
Buat templat definisi build kosong. Klik tombol + Tambahkan langkah pembuatan... dan tambahkan empat tugas ke definisi (lihat Gambar 2-28).

- Pilih kategori Deploy dan tambahkan dua tugas bertipe Azure PowerShell
- Pilih kategori Utilitas dan tambahkan tiga tugas dari jenis baris perintah
- Pindahkan tugas baris perintah di antara dua tugas Azure PowerShell, seperti yang digambarkan dalam Gambar 2-28



Gambar 2.28. Tugas definisi Penyediaan Bangun

Pada tab Opsi, centang opsi Izinkan Skrip Mengakses Token OAuth (lihat Gambar 2.29).



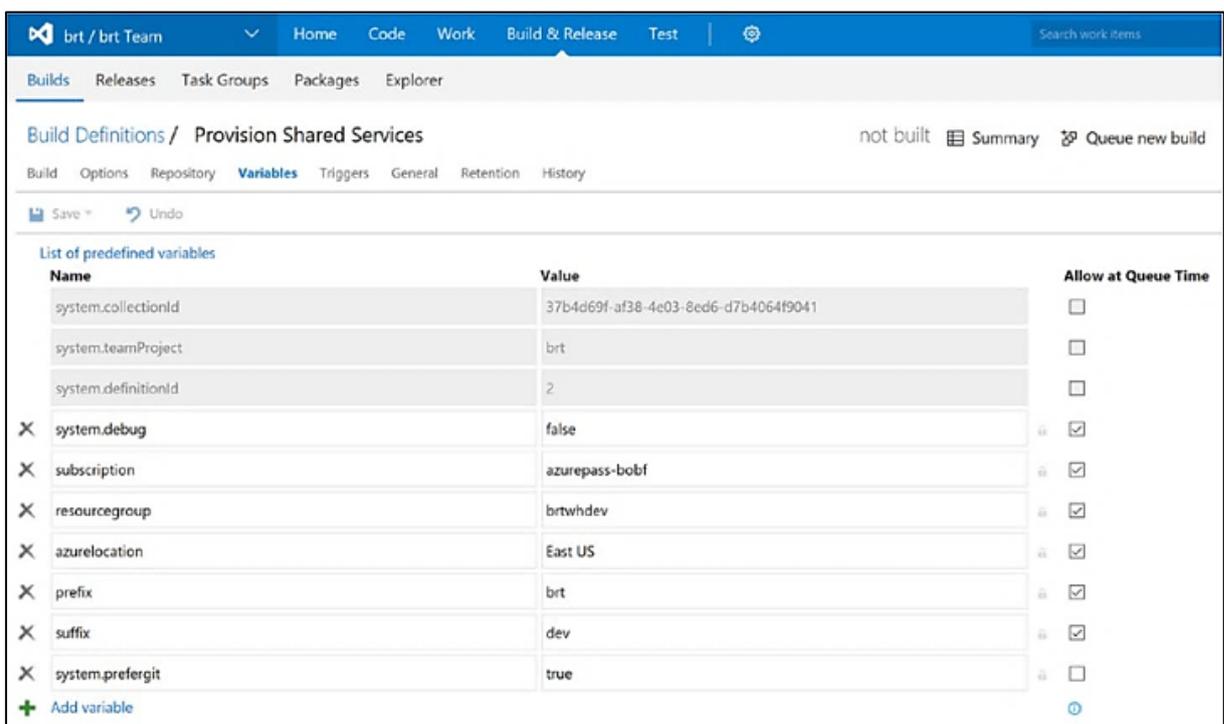
Gambar 2.29. Izinkan Skrip Mengakses Token OAuth

Sebelum menentukan pengaturan untuk setiap tugas, kita perlu mengkonfigurasi variabelnya. Variabel memungkinkan Anda membuat nama yang mudah dipahami untuk nilai yang dapat diubah saat Anda menjalankan build.

Setiap definisi build dapat memiliki kumpulan variabelnya sendiri. Skrip PowerShell, seperti yang Anda ingat, menggunakan serangkaian parameter baris perintah. Anda akan menggunakan variabel untuk memberikan nilai parameter ini saat runtime. Klik Variabel di menu dan tambahkan definisi variabel yang tercantum pada Tabel 2-2 dan digambarkan pada Gambar 2-30.

Tabel 2-2. Definisi Variabel untuk Definisi Pembuatan Ketentuan

Nama	Nilai	Izinkan Pada Waktu Q
Berlangganan	[nama-langganan anda]	Ya
Grup sumber daya	[nama-grup sumber daya]	Ya
Lokasi Biru	[wilayah biru]	Ya
Awalan	[awalan]	Ya
Akhiran	[akhirian]	Ya
System preflight	Benar	Tidak



Gambar 2.30. Bangun variabel definisi

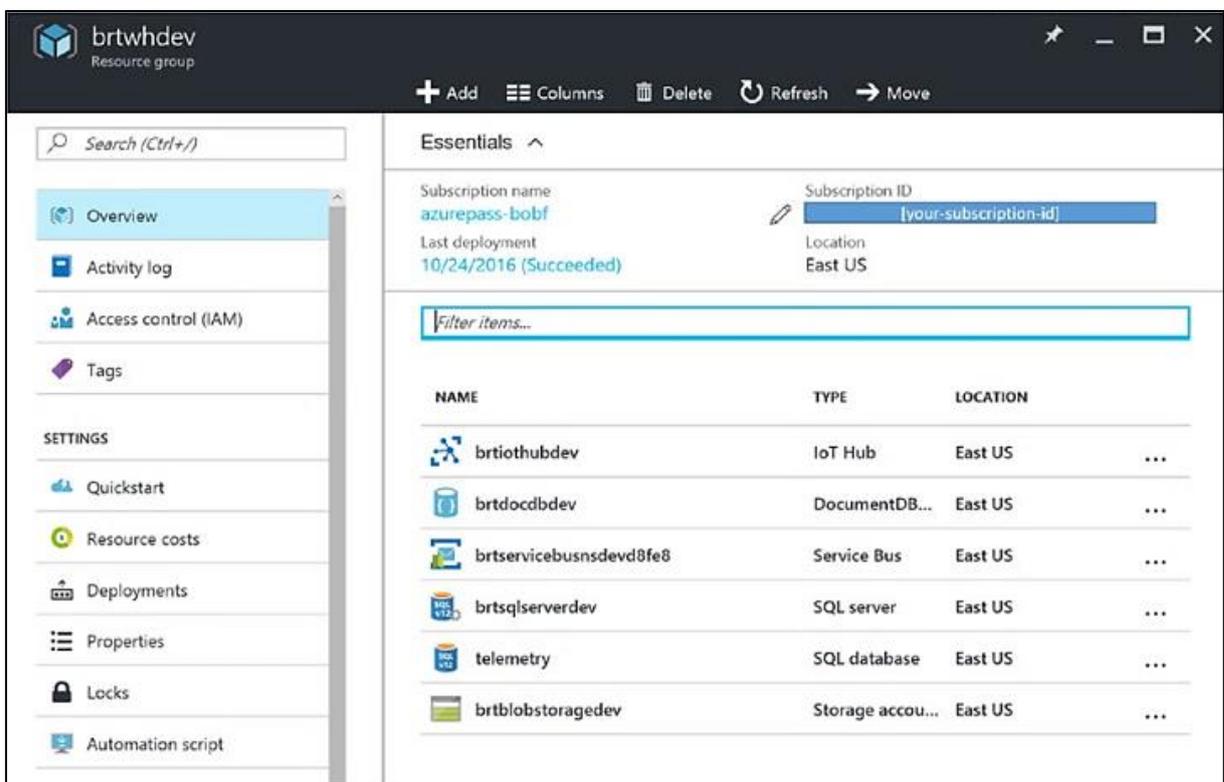
Klik Bangun di menu dan isi pengaturan untuk Azure PowerShell dan tugas baris perintah. Skrip PowerShell pertama akan menerapkan templat ARM yang telah ditinjau di awal bab ini. Skrip dan templat menyediakan layanan bersama dan kemudian menghasilkan file output yang berisi string koneksi dan kunci untuk layanan bersama. Tugas baris perintah menjalankan perintah Git untuk menambahkan file yang dihasilkan ke repositori dan mengkomitnya. Terakhir, kita akan menjalankan skrip PowerShell yang menghasilkan tabel relasional dalam database SQL. Kami akan menggunakan variabel untuk meneruskan parameter dinamis ke skrip. Isi pengaturan untuk setiap tugas build yang diuraikan pada Tabel 2.3.

Tabel 2.3. Penyediaan Layanan Bersama Membangun Definisi

Task 1: Azure PowerShell Task Settings	
Azure Connection Type	Azure Resource Manager
Azure RM Subscription	[your-subscription-name]
Script Path	automation/01-Provision-SharedServices.ps1
Script Arguments	-subscription \$(subscription) -resourcegroup \$(resourcegroup) -azurelocation '\$(azurelocation)' -Prefix \$(prefix) -Suffix \$(suffix) -verbose
Task 2: Command-Line Task Settings	
Tool	Git
Arguments	add provision-\$(resourcegroup)-output.json
Working Folder (Advanced)	automation
Fail on Standard Error	Yes
Task 3: Command-Line Task Settings	
Tool	Git
Arguments	commit -m "commit provision output"
Working Folder (Advanced)	automation
Fail on Standard Error	Yes
Task 4: Command-Line Task Settings	
Tool	Git
Arguments	push origin HEAD:master
Working Folder (Advanced)	automation
Fail on Standard Error	No
Continue Error	Yes
Task 5: Azure PowerShell Task Settings	
Azure Connection Type	Azure Resource Manager
Azure RM Subscription	[your-subscription-name]

Script Path	automation/02-Update-SQLDatabase.ps1
Script Arguments	-subscription \$(subscription) -resourcegroup \$(resourcegroup) -azurelocation '\$(azurelocation)' -Prefix \$(prefix) -Suffix \$(suffix) -verbose

Simpan definisi dengan memberikan nama seperti Provision Shared Services dan mengantri build. Ketika skrip ini selesai, Anda akan membuat grup sumber daya di langganan Anda yang berisi layanan bersama; menghasilkan file keluaran yang berisi string koneksi dan kunci untuk definisi build lainnya; dan memeriksa file baru tersebut ke dalam repo Git (lihat Gambar 2-31).



Gambar 2.31. Grup sumber daya dengan layanan bersama yang disediakan

2.6 KETENTUAN MENGGUNAKAN AZURE CLASSIC COMMANDS

Ada beberapa operasi yang memerlukan skrip PowerShell untuk dijalankan dalam apa yang disebut mode Azure Classic (artinya operasi tersebut tidak berfungsi dalam konteks ARM). Perintah ini berkaitan dengan pembuatan kontainer penyimpanan blob dan mengakses serta menyimpan string koneksi bus layanan. Perintahnya berada dalam file bernama 03-Provision-ClassicOps.ps1 di folder otomatisasi. Mari kita tinjau kodenya.

get the path

```
$Path = Split-Path -parent $PSCmdPath
$Path = Split-Path -parent $Path
```

```

# pull in the environment variables
$includePath = $Path + "\Automation\EnvironmentVariables.ps1"
."$includePath"

# get the storage key and context
$storageKey = (Get-AzureRmStorageAccountKey -AccountName
    $storageAccountName -ResourceGroupName
    $ResourceGroup)[0]

$StorageContext = New-AzureStorageContext
    -StorageAccountName $storageAccountName
    -StorageAccountKey $storageKey.Value

# create the storage containers
New-AzureStorageContainer -Context $StorageContext
    -Name $ArchiveContainerName -Permission Off
    -ErrorAction SilentlyContinue
New-AzureStorageContainer -Context $StorageContext
    -Name $RefDataContainerName -Permission Off
    -ErrorAction SilentlyContinue
New-AzureStorageContainer -Context $StorageContext
    -Name $ImageContainerName -Permission Off
    -ErrorAction SilentlyContinue

# Upload the rules file to the reference data container
$refdata = $path+"\automation\deploy\rules\$TempSensorRulesFilename"
Set-AzureStorageBlobContent -Context $StorageContext
    -Container $RefDataContainerName -File $refdata
    -Force

# Upload the image files to the image container
# $_.mode -match "-a---" scans the data directory
# and only fetches the files, filtering out directories
$imagedir = $path + "\automation\deploy\images"
    $files = Get-ChildItem $imagedir -force | Where-Object {$_.mode -
    match "-a---"}

# iterate through all the files and start uploading data
foreach ($file in $files)
{
    #fq name represents fully qualified name
    $fqName = $imagedir + "\" + $file.Name

    #upload the current file to the blob
    Set-AzureStorageBlobContent -Blob $file.Name
        -Context $StorageContext
        -Container $ImageContainerName
        -File $fqName -Force

```

```

}

# save the service bus connection string
$sbr = Get-AzureSBAuthorizationRule -Namespace $serviceBusNamespace

$JSON =@"
{
  "ServiceBusConnectionString":
    "$sbr.ConnectionString",
}
"@

$ServiceBusInfo=$Path+"\automation\servicebus-$resourcegroup-output.json"
$JSON | Set-Content -Path $ServiceBusInfo

```

Sebelum Anda dapat membuat definisi build untuk menjalankan skrip provisi ini, Anda perlu membuat koneksi antara lingkungan build dan langganan Azure untuk menjalankan perintah Azure Classic. Navigasikan ke halaman Konfigurasi Layanan, tambahkan titik akhir tipe Azure Classic, dan isi formulir (lihat Gambar 2-32).

Gambar 2.32. Konfigurasi koneksi Azure Classic

Buat definisi build kosong dan tambahkan tugas Deploy tipe Azure PowerShell Script dan kemudian tiga tugas utilitas tipe baris perintah. Konfigurasi pengaturan yang ditunjukkan pada Tabel 2.4.

Tabel 2.4. Penyediaan Definisi Pembuatan Sumber Daya Klasik

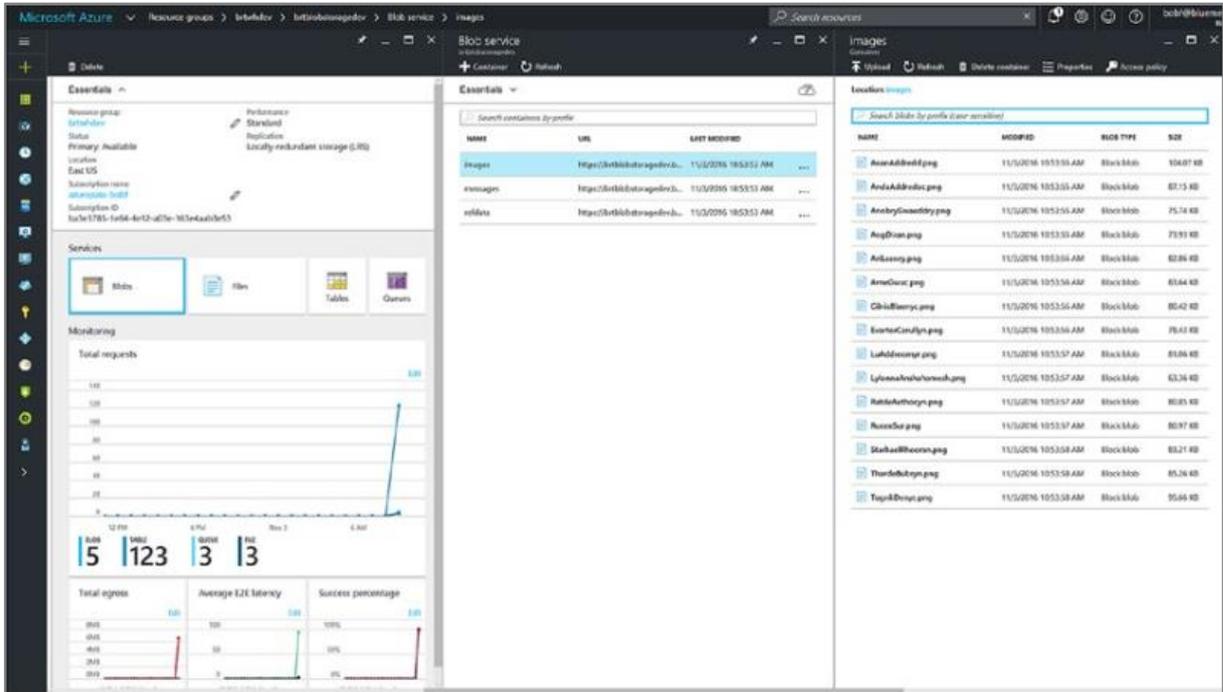
Task 1: Azure PowerShell Task Settings	
Azure Connection Type	Azure Classic
Azure Classic Subscription	[your-classic-subscription-connection-name]
Script Path	automation/03-Provision-ClassicalOps.ps1
Script Arguments	-subscription \$(subscription) -resourcegroup \$(resourcegroup) -verbose
Task 2: Command-Line Task Settings	
Command	Git
Arguments	add servicebus-\$(resourcegroup)-output.json
Working Folder	automation
File on Standard Error	Yes
Task 3: Command-Line Task Settings	
Command	Git
Arguments	commit -m "commit provision output"
Working Folder	automation
File on Standard Error	Yes
Task 4: Command-Line Task Settings	
Command	Git
Arguments	push origin HEAD:master
Working Folder	automation
File on Standard Error	No
Continue on Error	Yes

Tentukan variabel untuk definisi build ini seperti yang ditunjukkan pada Tabel 2-5.

Tabel 2-5. Definisi Variabel untuk Penyediaan Definisi Pembuatan Sumber Daya Klasik

Name	Value	Set a Q Time
subscription	[your-subscription-name]	Yes
resourcegroup	[resource-group-name]	Yes

Untuk memvalidasi skrip, navigasikan ke akun penyimpanan Anda dan periksa kontainer penyimpanan blob. Anda akan melihat beberapa file gambar telah diunggah ke wadah gambar. Ini adalah foto kepala karyawan (lihat Gambar 2-33).



Gambar 2-33. Wadah gambar dibuat dan file diunggah

Sekarang juga ada file baru di folder otomatisasi bernama `servicebus-[resource-group-name]-output.json` yang berisi string koneksi bus layanan. File ini juga telah diperiksa ke dalam repositori.

Buat Definisi Pembangunan Deployment

Dalam latihan ini, Anda akan membuat definisi build penerapan untuk API Referensi layanan mikro yang akan menjalankan langkah-langkah berikut:

- Membangun dan mengemas layanan mikro Referensi
- Membuat database Referensi dan kumpulan Entitas di Dokumen DB
- Unggah kumpulan sampel data referensi ke DocumentDB
- Menyediakan Paket App Service dan App Service di grup sumber daya
- Menyebarkan paket API Referensi ke App Service
- Konfigurasi pengaturan App Service dengan info koneksi untuk DocumentDB

Karena Anda telah membuat umpan NuGet baru dengan versi baru paket NuGet yang diperlukan oleh proyek layanan mikro, Anda perlu membuka solusi API Referensi, yang ada di folder layanan `microservices/reference`, dan membuat solusinya. Melakukan pembangunan lokal akan memperbarui referensi NuGet. Setelah selesai, periksa pembaruan pada repositori.

Buat definisi build baru menggunakan template kosong dan tentukan variabel seperti yang ditunjukkan pada Tabel 2.6.

Tabel 2.6. Definisi Variabel untuk Definisi Pembuatan API Referensi Deployment

Name	Value	Set at Q Time
BuildPlatform	Any CPU	Yes
BuildConfiguration	[debug release]	Yes
Subscription	[your-subscription-name]	Yes
Resourcegroup	[resource-group-name]	Yes
Azure	location[azure-region]	Yes
Prefix	[project-code]	Yes
Suffix	[dev tst prd]	Yes
ServiceName	ReferenceAPI	No
ServicePlan	AppServicePlan	No
Database	Reference	No
Collection	Entity	No
DeployData	false	Yes

Tambahkan tugas ke definisi build seperti yang ditunjukkan pada Tabel 2.7.

Tabel 2.7. Tugas untuk Definisi Pembuatan API Referensi Deploy

Category	Task
Package	NuGet Installer
Build	Visual Studio Build
Deploy	Azure PowerShell
Deploy	Azure App Service Deployment ARM
Deploy	Azure PowerShell

Konfigurasi setiap langkah pembangunan seperti yang ditunjukkan pada Tabel 2-8.

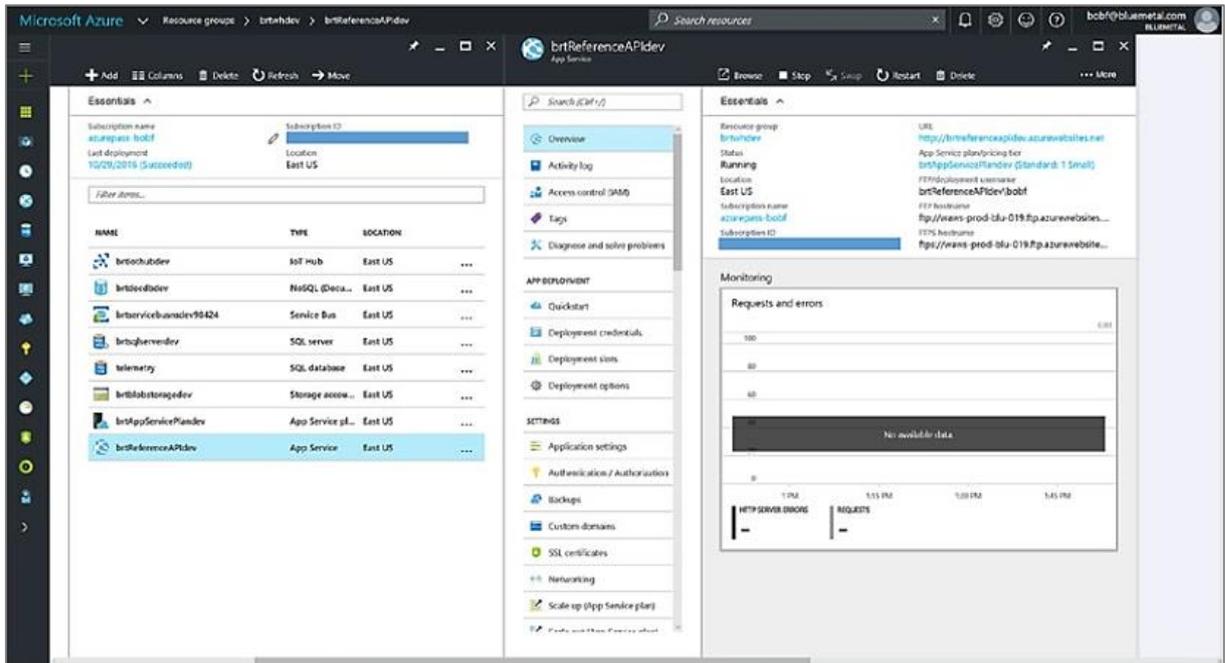
Tabel 2-8. Terapkan Definisi Pembuatan API Referensi

Task 1: NuGet Installer Task Settings	
Path to solution	microservices/Reference/ReferenceAPI.sln
Path to nuget.config	nuget.config
NuGet Arguments	-outputdirectory packages
Task 2: Visual Studio Build Task Settings	
Solution	microservices/Reference/ReferenceAPI.sln
MSBuild Arguments	/p:DeployOnBuild=true;DefaultPackageFilename=ReferenceAPI.zip /p:PublishUrl=\$(Build.StagingDirectory)/ReferenceAPI
Platform	\$(BuildPlatform)

Configuration	<code>\$(BuildConfiguration)</code>
Task 3: Azure PowerShell Task Settings	
Azure Connection Type	Azure Resource Manager
Azure RM Subscription	<code>[your-subscription-connection-name]</code>
Script Path	<code>automation/05-Publish-AppService.ps1</code>
Script Arguments	<code>-Subscription \$(Subscription) -ResourceGroup \$(ResourceGroup) -AzureLocation '\$(AzureLocation)' -Prefix \$(Prefix) -Suffix \$(Suffix) -ServiceName \$(ServiceName) -ServicePlan \$(ServicePlan) -Database \$(Database) -Collection \$(Collection) -DeployData:\$\$ (DeployData)</code>
Task 4: Azure App Service Deployment ARM Task Settings	
Azure RM Subscription	<code>[your-subscription-connection-name]</code>
App Service Name	<code>\$(Prefix)ReferenceApi(\$Suffix)</code>
Package Folder Name	<code>microservices/reference/ReferenceAPI/obj/Debug/Package/ReferenceAPI.zip</code>
Publish Using Web Deploy	Yes
Remove Additional Files	Yes
Task 5: Azure PowerShell Task Settings	
Azure Connection Type	Azure Resource Manager
Azure RM Subscription	<code>[your-subscription-connection-name]</code>
Script Path	<code>automation/06-Publish-AppSettings.ps1</code>
Script Arguments	<code>-Subscription \$(Subscription) -ResourceGroup \$(ResourceGroup) -AzureLocation '\$(AzureLocation)' -Prefix \$(Prefix) -Suffix \$(Suffix) -ServiceName \$(ServiceName) -Database \$(Database) -Collection \$(Collection)</code>

Simpan definisi build dengan memberikan nama seperti `Deploy Reference Microservice` dan antri build. Atur variabel `Deploy Data` ke `true`. Hal ini akan memicu pembuatan Basis Data Referensi dan Koleksi Entitas serta mengunggah data referensi sampel ke dalam koleksi `DocumentDB`.

Setelah pembangunan selesai, Anda akan melihat Paket App Service dan API Referensi App Service tercantum di grup sumber daya Anda. Paket App Service menentukan pengaturan peningkatan dan perluasan skala untuk setiap Layanan Aplikasi yang terkait dengan rencana tersebut (lihat Gambar 2-34).



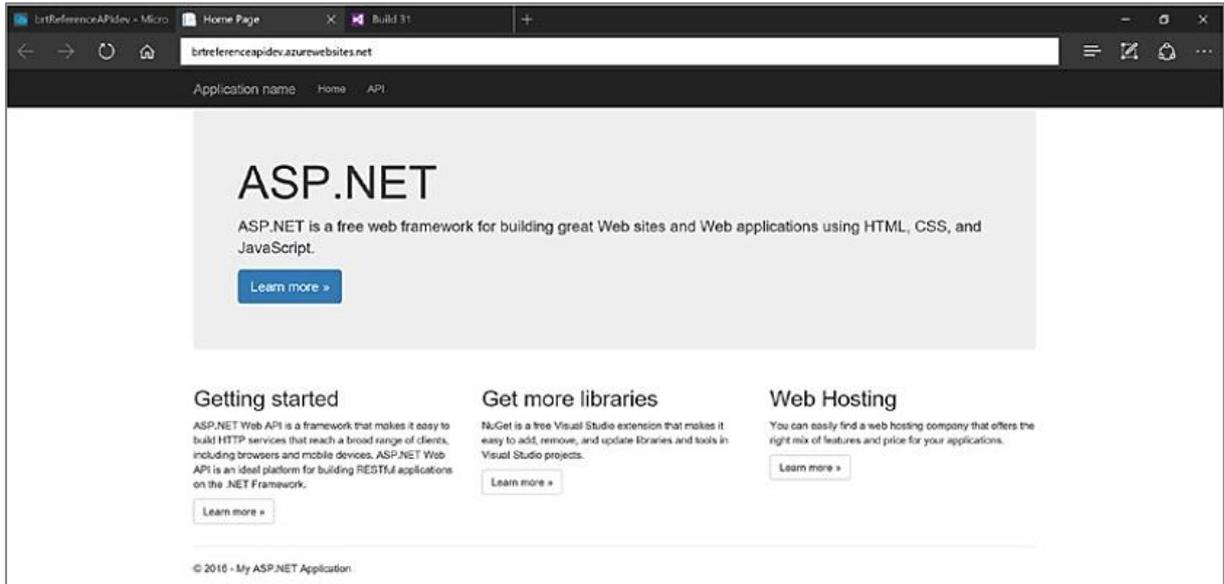
Gambar 2-34. Penyebaran layanan mikro referensi

Jika Anda mengklik pengaturan aplikasi, Anda akan melihat bahwa pengaturan runtime yang diperlukan layanan ini telah dikonfigurasi berdasarkan definisi penerapan (lihat Gambar 2-35). Nilai untuk pengaturan ini diambil dari file yang dihasilkan oleh definisi build penyediaan dan dimasukkan secara dinamis saat layanan ini dibuat dan disebar.

App settings			
collection	Entity	<input type="checkbox"/> Slot setting	...
database	Reference	<input type="checkbox"/> Slot setting	...
docdburi	https://brtdocdbdev.docum...	<input type="checkbox"/> Slot setting	...
apiss	810ecd57-e305-43ec-89ad-...	<input type="checkbox"/> Slot setting	...
docdbkey	N8ZLwHk5SE8zogBRcF6lvZ...	<input type="checkbox"/> Slot setting	...
<input type="text" value="Key"/>	<input type="text" value="Value"/>	<input type="checkbox"/> Slot setting	...

Gambar 2-35. Pengaturan aplikasi dinamis

Anda dapat menguji layanan mikro Referensi dengan mengklik URL di layar Ikhtisar. Layanan akan berputar dan membawa Anda ke halaman beranda default aplikasi API (lihat Gambar 2-36).



Gambar 2-36. Referensi halaman beranda default Microservice

Pada latihan berikutnya, Anda akan menyediakan manajemen API dan mengonfigurasi proksi API. Setelah selesai, kita dapat menguji API dan memvalidasi bahwa database dan koleksi di DocumentDB telah diinisialisasi dengan benar.

Buat Definisi Bangun Apim Ketentuan

Manajemen API adalah layanan Azure untuk menerbitkan API ke pengembang. Anda menggunakan manajemen API untuk menentukan titik akhir proksi yang dapat Anda lacak analitiknya, menentukan akses keamanan, menggunakan model berlangganan, memasukkan kebijakan seperti pembatasan atau header khusus, dan banyak lagi.

Setelah manajemen API disediakan, Anda akan memiliki Portal Administrator dan Portal Pengembang. Portal Administrator digunakan untuk mengelola API, grup pengembang, dan produk serta kebijakan API, serta meninjau laporan analitik. Portal Pengembang memberi pengembang kemampuan untuk mendaftar dan berlangganan API yang aksesnya telah diberikan kepada mereka. Setelah berlangganan, pengembang menerima kunci berlangganan yang harus digunakan pada setiap panggilan ke API. Portal ini juga menyediakan halaman konsol untuk setiap API tempat pengembang dapat mengakses dokumentasi dan menguji API untuk melihat cara kerjanya. Untuk membuat definisi build untuk menyediakan manajemen API, mulailah dengan definisi build kosong dan tambahkan satu tugas build dari kategori Deploy, tambahkan tugas build Azure PowerShell dengan pengaturan yang ditunjukkan pada Tabel 2-9.

Tabel 2.9. Definisi Pembuatan APIM Penyediaan

Task 1: Azure PowerShell Task Settings	
Azure Connection Type	Azure Resource Manager
Azure RM Subscription	[your-subscription-connection-name]
Script Path	automation/04-Provision-APIManagement.ps1

Script Arguments	<pre> -subscription \$(subscription) -resourcegroup \$(resourcegroup) -azurelocation '\$(azurelocation)' -Prefix \$(prefix) - Suffix \$(suffix) -Organization '\$(organization)' - APIServiceName \$(apiservicename) -APIAdminEmail \$(APIAdminEmail) - verbose </pre>
-------------------------	--

Tentukan variabel yang ditunjukkan pada Tabel 2-10.

Tabel 2-10. Definisi Variabel untuk Definisi Pembuatan APIM Penyediaan

Name	Value	Set at Q Time
Subscription	[your-subscription-name]	Yes
ResourceGroup	[resource-group-name]	Yes
AzureLocation	[azure-region]	Yes
Prefix	[project-code]	Yes
Suffix	[dev tst prd]	Yes
Organization	[your company name]	Yes
APIServiceName	[project-code]	Yes
APIAdminEmail	[your-email]	Yes

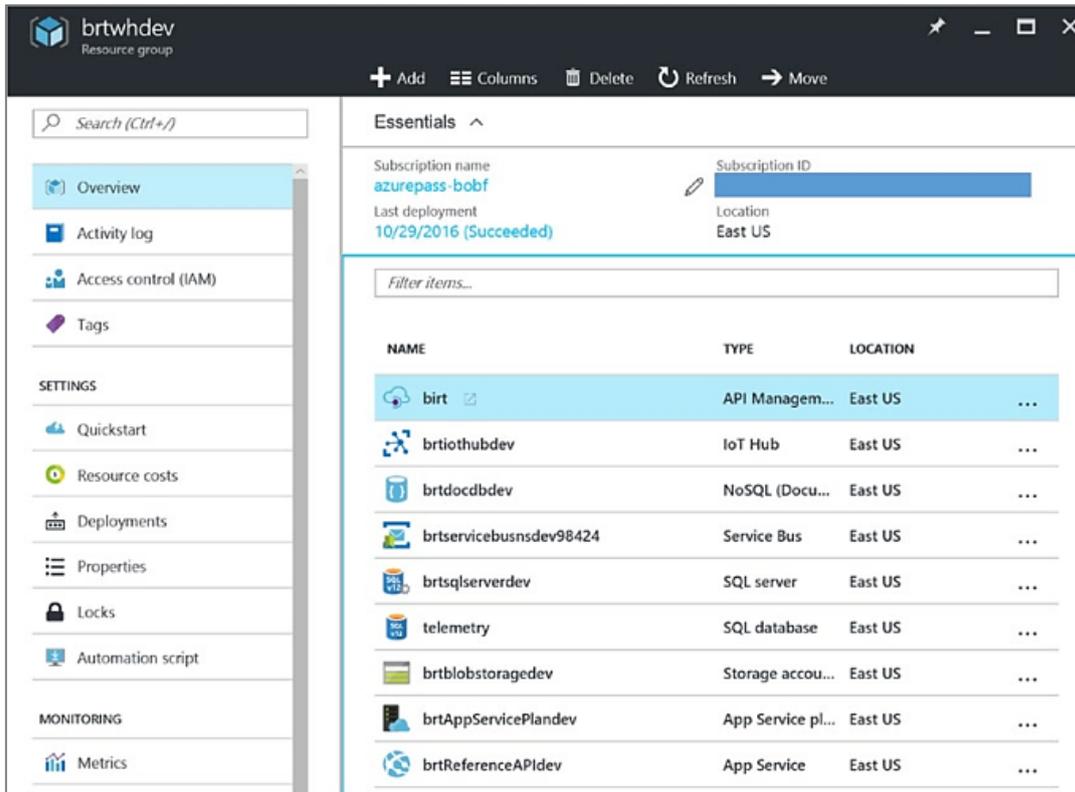
Jika Anda mengalami masalah, Anda juga dapat menjalankan skrip penyediaan dari laptop Anda menggunakan konsol PowerShell. Jalankan perintah ini dari folder otomatisasi.

```

PS C:\> .\00-Login.ps1
PS C;\> .\04-Provision-APIManagement.ps1

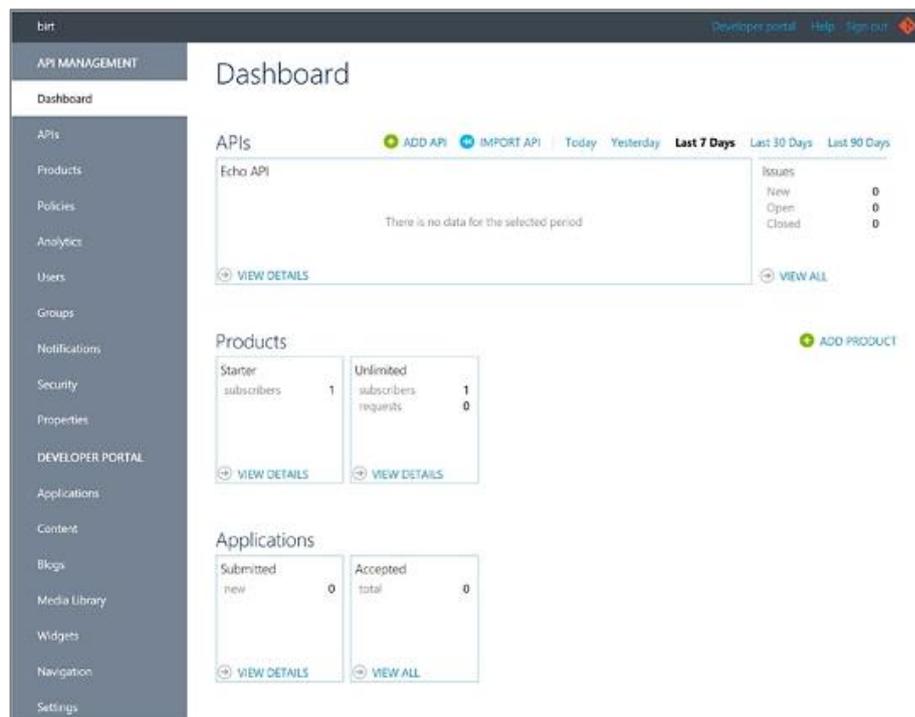
```

Setelah proses penyediaan selesai, Anda akan melihat contoh manajemen API di grup sumber daya Anda, seperti yang ditunjukkan pada Gambar 2-37.



Gambar 2-37. Manajemen API disediakan

Klik nama layanan di grup sumber daya untuk menavigasi ke bilah Manajemen API. Dari sana Anda dapat meluncurkan Portal Penerbit atau Portal Pengembang. Luncurkan Portal Penerbit (lihat Gambar 2-38).



Gambar 2-38. Portal Penerbit APIM

Portal Penerbit menyediakan akses ke semua pengaturan dan fungsi manajemen API. Mari kita periksa fitur penerbit:

- *Dasbor*: Dasbor memberikan ringkasan ikhtisar produk, grup, dan aktivitas API.
- *Produk*: Produk adalah konstruksi organisasi untuk API. Dengan mengatur API ke dalam produk, Anda dapat mengontrol siapa yang dapat mengakses API tersebut.
- *Grup*: Grup menyediakan konstruksi organisasi untuk keanggotaan. Misalnya, Anda dapat membuat grup pengembang yang mewakili berbagai tim, pengembang internal dan eksternal, operasi, dll. Grup tersebut kemudian ditetapkan ke produk, yang membatasi akses hanya ke API yang dipublikasikan sebagai bagian dari produk tersebut.
- *Pengguna*: Pengguna memberikan visibilitas administrator ke langganan pengembang. Anda dapat menambahkan pengembang dan mengundang pengembang ke layanan.
- *API*: API mendefinisikan proksi API, termasuk dokumentasi, operasi, dan pengaturan keamanan.
- *Kebijakan*: Kebijakan menyediakan mekanisme yang dapat digunakan untuk memasukkan operasi pada permintaan masuk dan respons keluar seperti memasukkan header, menetapkan kuota, mengonfigurasi CORS, dan sebagainya.
- *Analytics*: Analytics memberikan laporan penggunaan dan kesehatan untuk setiap operasi API, penggunaan bandwidth, dan visibilitas ke mana pengguna memanggil API mana.
- *Pemberitahuan*: Peringatan administrator dilaporkan di sini.
- *Keamanan*: Keamanan memungkinkan Anda mengonfigurasi pengaturan keamanan untuk layanan Manajemen API termasuk integrasi Direktori Aktif dan akses ke API Manajemen.
- *Properti*: Nilai properti adalah string yang mungkin berisi rahasia dan dapat direferensikan dari kebijakan. Gunakan properti untuk menggunakan kembali nilai di seluruh kebijakan dan menghindari menentukan rahasia dalam kebijakan.
- *Portal Pengembang*: Bagian Portal Pengembang pada Portal Administrator memberi Anda kemampuan untuk menyesuaikan Portal Pengembang.

Dalam latihan berikutnya, Anda akan mengonfigurasi lingkungan APIM dan menentukan proksi untuk API Referensi.

Tambahkan Kelompok Apim

Klik Grup di menu navigasi Portal Penerbit dan klik Tambahkan Grup. Berikan nama dan deskripsi BRT `Developers` dan klik Simpan. Buatlah grup kedua yang disebut BRT `Operators` (lihat Gambar 2-39).

Add group

Name
BRT Developers
Unique name of the group.

Description
BRT Developers
Description of the group's purpose and its members.

Save Cancel

Gambar 2-39. Buat grup APIM

Tambahkan Produk Apim

Klik Produk lalu Tambah Produk. Berikan judul dan deskripsi masing-masing BRT . DEV dan BRT Developer APIs. Karena Anda menyediakan lingkungan pengembang, sebaiknya gunakan konvensi penamaan yang memperjelas bahwa API ini adalah bagian dari lingkungan tersebut. Secara default, API memerlukan langganan. Anda memiliki opsi untuk meminta persetujuan administrator untuk mengakses API di produk ini. Klik Simpan dan kemudian buat produk kedua bernama BRT.OPS dengan deskripsi BRT Operations APIs (lihat Gambar 2-40).

Add new product

Title
BRT.DEV
Display name of the product as it would appear on the developer and admin portals.

Description
BRT Developer APIs
Product descriptions usually explain product's purpose and highlight included APIs.

Require subscription
Developers will be required to subscribe to the product and use subscription key to access APIs included in it.

Require subscription approval
All subscription requests will be subject to approval. Configure subscription request email notifications on the Notifications page.

Allow multiple simultaneous subscriptions
Allow developers to have multiple subscriptions on the same product.

Save Cancel

Gambar 2.40. Buat sebuah produk

Klik nama produk dan pilih tab Visibilitas. Di sinilah Anda dapat mengontrol grup mana yang memiliki visibilitas terhadap produk. Tetapkan grup Pengembang BRT ke produk BRT.DEV dan grup Operator BRT ke produk BRT.OPS.

Definisikan Proxy Api

Untuk menyederhanakan proses pendefinisian proxy API, kami akan mengimpor definisi Swagger. Swagger adalah format JSON standar untuk mendefinisikan ReST API. Berikut adalah fragmen Swagger untuk API referensi:

```
{
  "swagger": "2.0",
  "info": {
    "title": "Reference.Dev",
    "version": "1.0"
  },
  "host": "[prefix]referenceapi[suffix].azurewebsites.net",
  "basePath": "/reference",
  "schemes": [
    "https"
  ],
  "paths": {
    "/entities/domain/{domain}": {
      "get": {
        "operationId": "Get Entities by Domain",
        "parameters": [
          {
            "name": "domain",
            "in": "path",
            "description": "",
            "required": true,
            "type": "string"
          },
          {
            "name": "Ocp-Apim-Subscription-Key",
            "in": "header",
            "description": "subscription key in header",
            "type": "string"
          }
        ],
        "responses": {}
      }
    },
    ...
  }
}
```

Definisi swagger telah disediakan untuk setiap layanan mikro. Anda dapat menemukannya di folder `automation/swagger`. Anda perlu memperbarui file-file ini agar mengarah ke layanan yang Anda sebarkan di Azure. Perbarui file `Reference.Swagger.json` dan `Reference.Adminstration.Swagger.json` dengan mengganti placeholder `[prefix]` and `[suffix]` dengan nilai yang Anda gunakan

selama penerapan. Klik API lalu Impor API. Pilih Dari File dan format Swagger, navigasikan ke direktori `automation/swagger`, dan pilih `Reference.Swagger.json`.

Pilih API Baru dan masukkan akhiran URI `dev/v1/reference`. Hal ini menciptakan titik akhir baru untuk API yang menyediakan lingkungan dan pembuatan versi tambahan. Terakhir, tambahkan API ke grup BRT.DEV dan klik Simpan (lihat Gambar 2-41).

Gambar 2-41. Impor API referensi

Ulangi langkah-langkah ini menggunakan file `swagger Reference.Adminstration.Swagger.json`.

Titik akhir API ini disediakan oleh layanan yang berjalan sama, API Referensi, namun operasi administrasi hanya tersedia untuk tim operasi. Masukkan akhiran URI `dev/v1/reference/admin`, tambahkan API ke grup BRT.OPS, dan klik Simpan (lihat Gambar 2-42).

Gambar 2-42. Impor API administrasi referensi

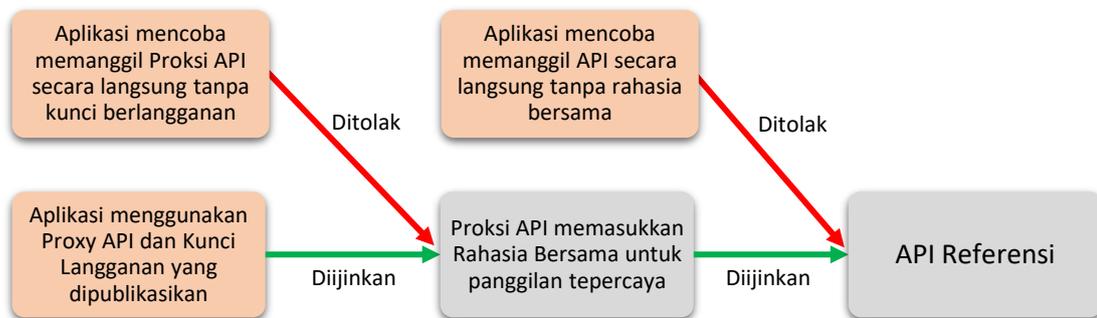
Agar pengembang dan operator terdaftar dapat memiliki visibilitas terhadap API, produk API perlu dipublikasikan. Klik Produk lalu klik nama produk lalu tombol Publikasikan. Lakukan ini untuk produk BRT.DEV dan BRT.OPS (lihat Gambar 2-43).

Gambar 2-43. Publikasikan produk

Buat Properti API

Ada beberapa cara untuk mengamankan REST API termasuk OAuth, sertifikat terkelola, dan teknik lainnya. Aplikasi ini akan menggunakan kombinasi kunci berlangganan yang disediakan oleh manajemen API dan rahasia bersama yang hanya diketahui oleh layanan API dan manajemen API. Untuk menyiapkan mekanisme rahasia bersama, Anda akan menggunakan kombinasi properti dan kebijakan manajemen API.

Seperti yang Anda lihat pada Gambar 2-44, jika aplikasi mencoba memanggil Proxy API tanpa kunci berlangganan, akses ditolak. Jika aplikasi mencoba melewati manajemen API dan memanggil API secara langsung tanpa rahasia bersama, akses akan ditolak. Dalam kedua kasus tersebut, jika panggilan tidak dilakukan menggunakan SSL, pemanggilan akan gagal (lihat Gambar 2.44).



Gambar 2-44. Mengamankan API dengan kunci langganan dan rahasia bersama

Properti digunakan untuk mendefinisikan rahasia dalam definisi kebijakan. Konfigurasi API untuk aplikasi ini memerlukan SSL dan rahasia bersama diteruskan ke layanan dalam header yang disebut `apiss`. Jika header tidak ada atau nilai di header tidak cocok dengan nilai rahasia bersama yang diketahui API, akses akan ditolak.

Nilai rahasia bersama default disediakan dalam file `EnvironmentVariables.ps1` di folder otomatisasi. Anda harus memperbarui nilai dalam file ini menggunakan utilitas konsol `GenSharedSecret` yang dapat ditemukan di folder `utility`. Setelah Anda memiliki nilai rahasia bersama yang baru, klik Properti di Portal Penerbit, lalu Tambahkan Properti. Buat properti dengan nama `SSHeaderName` dan atur nilainya menjadi `apiss`. Centang kotak Ini Rahasia. Ulangi langkah ini dan buat properti dengan nama `SSHeaderValue`. Tetapkan nilai rahasia bersama yang Anda buat (lihat Gambar 2-45).

Gambar 2-45. Buat properti

■ **Catatan:** Jika Anda memperbarui rahasia bersama, ingatlah untuk menerapkan ulang API Anda sehingga pengaturan aplikasi diperbarui dengan nilai rahasia bersama yang baru.

Definisikan Kebijakan API

Sekarang setelah Anda menentukan properti, Anda dapat menggunakan kebijakan Set Header pada permintaan masuk untuk memasukkan rahasia bersama yang menjadikan manajemen API sebagai pemanggil layanan mikro yang tepercaya.

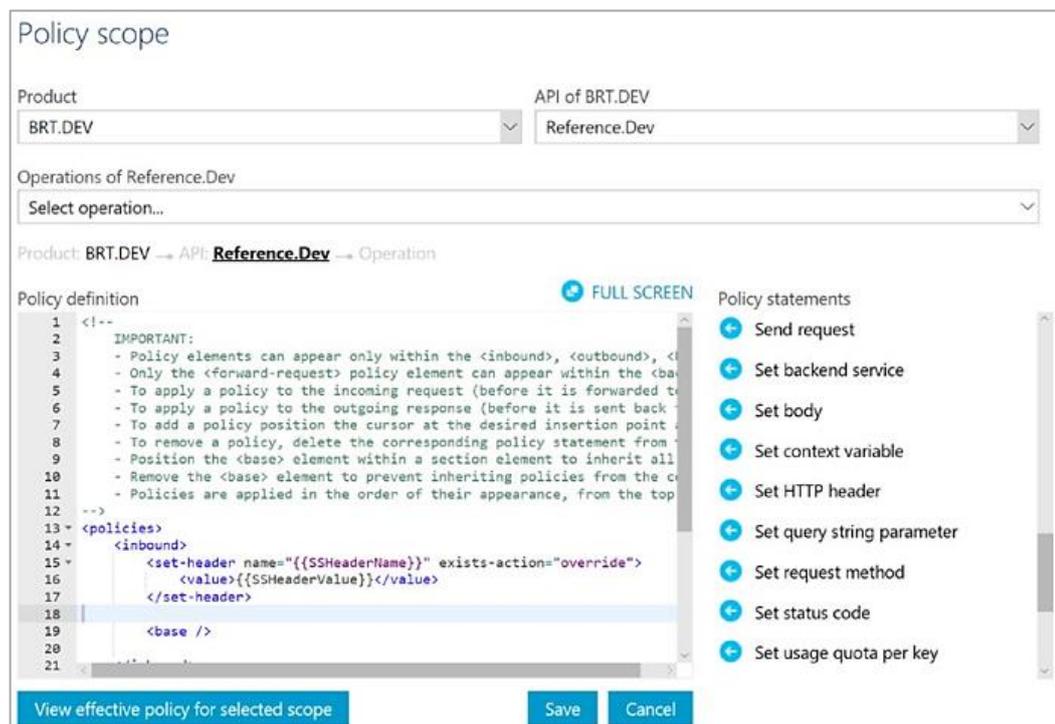
Klik Kebijakan, pilih produk BRT.DEV, lalu API Referensi.DEV.

Klik Tambahkan Kebijakan di editor dan letakkan kursor Anda tepat setelah tag <inbound> (lihat Gambar 2-46).

Gambar 2-46. Editor Kebijakan

Gulir ke bawah daftar kebijakan dan masukkan kebijakan Set Header ke dalam dokumen kebijakan. Ubah header menggunakan sintaks properti. Untuk mereferensikan nama header rahasia bersama, gunakan `{{SSHeaderName}}`, dan untuk mereferensikan nilai, gunakan `{{SSHeaderValue}}`. Tetapkan `exists-action=override`. Klik Simpan (lihat Gambar 2-47).

```
<set-header name="{{SSHeaderName}}" exists-action="override">
  <value>{{SSHeaderValue}}</value>
</set-header>
```

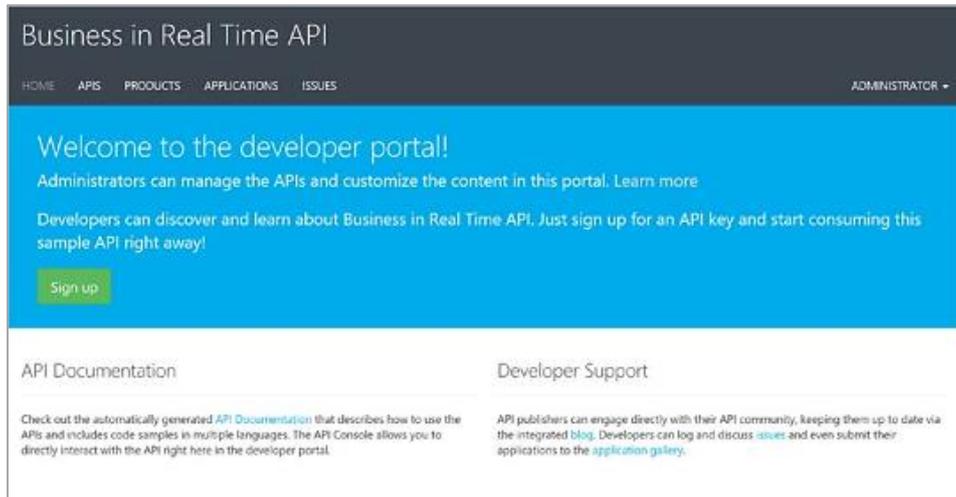


Gambar 2-47. Tetapkan kebijakan header menggunakan sintaks referensi properti

Ulangi langkah-langkah ini untuk API Referensi.Administrasi.Dev.

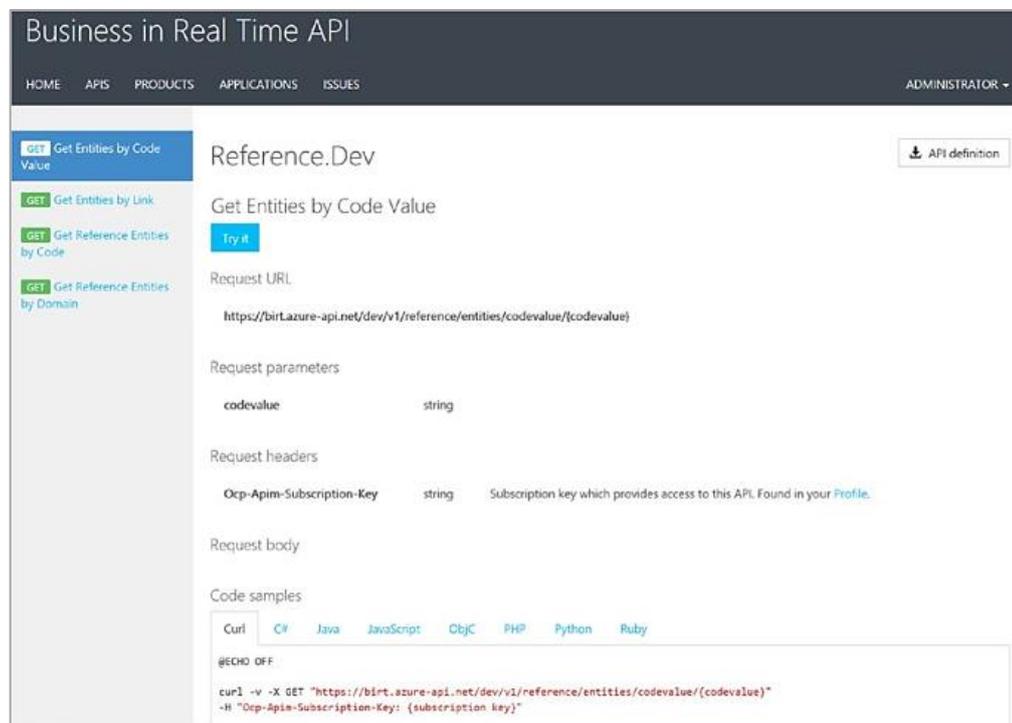
Pengujian API

Dari Portal Administrator, navigasikan ke Portal Pengembang Manajemen API (lihat Gambar 2-48). Portal ini dibuat secara otomatis dan menyediakan portal layanan mandiri bagi pengembang untuk mengakses API Anda. Saat pengembang mengunjungi portal ini pertama kali, mereka akan diminta untuk mendaftar. Administrator akan menerima pemberitahuan email dan juga akan melihat pemberitahuan di area Pemberitahuan di Portal Penerbit. Setelah keanggotaan disetujui, pengembang akan dapat melihat produk yang dipublikasikan. Karena Anda adalah administrator pengelolaan API, Anda sudah memiliki akses langsung.



Gambar 2.48 Portal Pengembang

Klik pada API untuk melihat daftar API yang diterbitkan. Pilih API Referensi.DEV. Anda akan melihat daftar operasi. Halaman operasi menyediakan URL permintaan dan contoh kode serta kemampuan untuk memanggil API untuk mempelajari cara kerjanya. Pilih operasi Dapatkan Entitas Referensi berdasarkan Domain dan kemudian klik Cobalah (lihat Gambar 2-49).

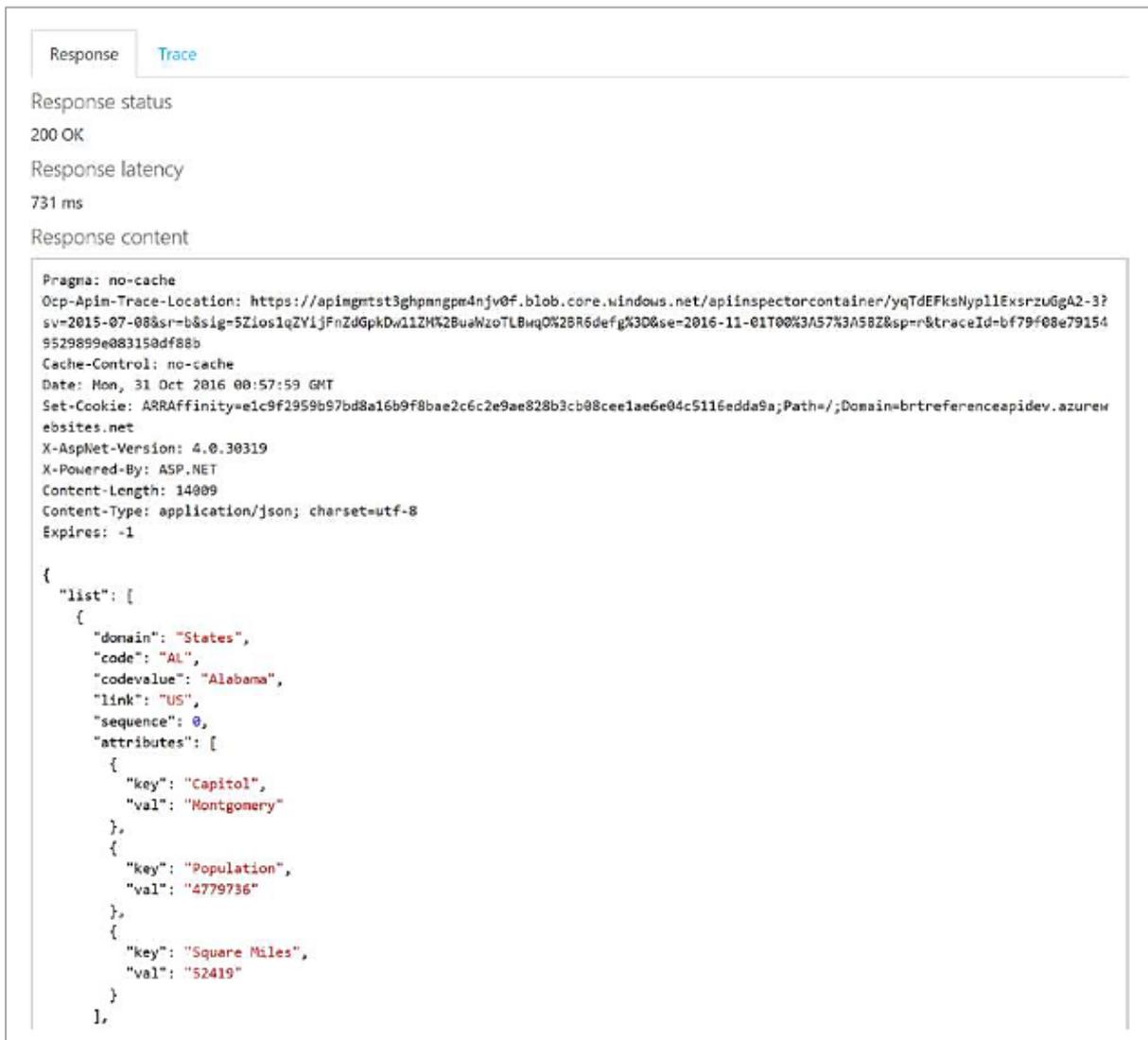


Gambar 2-49. Referensi operasi API

Data referensi adalah informasi yang umum pada banyak aplikasi seperti daftar negara bagian, kode negara, kode pos, kode bahasa, dan sebagainya. Anda tidak ingin mengizinkan sembarang aplikasi mengubah data ini dan merusak daftar pencarian; oleh karena itu, operasi Referensi telah dipartisi menjadi API publik hanya-baca (API Referensi) dan API dengan

kemampuan pembaruan (API Administrasi Referensi). Sebelumnya, kami mengonfigurasi API Referensi agar dapat dilihat oleh pengembang dan API Administrasi Referensi agar hanya dapat dilihat oleh operator.

Operasi “Dapatkan Entitas Referensi berdasarkan Domain” mengambil domain entitas, misalnya Serikat, dan mengembalikan semua entitas yang merupakan bagian dari domain tersebut. Masukkan Negara di bidang domain dan kemudian klik tombol Kirim (lihat Gambar 2-50).



```

Response status
200 OK
Response latency
731 ms
Response content
Pragna: no-cache
Ocp-Apim-Trace-Location: https://apimgmtst3ghpmngpm4njv0f.blob.core.windows.net/apiinspectorcontainer/yqTdeFksNyp11ExsrzuG42-3?sv=2015-07-08&sr=b&sig=5Zios1qZV1jFnZdGpkDw11ZM%2BuaWzoTLBmq0%2BR6defg%3D&se=2016-11-01T00%3A57%3A58Z&sp=r&traceId=bf79f08e791549529899e083150df88b
Cache-Control: no-cache
Date: Mon, 31 Oct 2016 08:57:59 GMT
Set-Cookie: ARRAffinity=e1c9f2959b97bd8a16b9f8bae2c6c2e9ae828b3cb08cee1ae6e04c5116edda9a;Path=/;Domain=brtreferenceapidev.azurewebsites.net
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Content-Length: 14009
Content-Type: application/json; charset=utf-8
Expires: -1

{
  "list": [
    {
      "domain": "States",
      "code": "AL",
      "codevalue": "Alabama",
      "link": "US",
      "sequence": 0,
      "attributes": [
        {
          "key": "Capitol",
          "val": "Montgomery"
        },
        {
          "key": "Population",
          "val": "4779736"
        },
        {
          "key": "Square Miles",
          "val": "52419"
        }
      ]
    }
  ]
}

```

Gambar 2-50. Respons dari API referensi

Dapatkan Kunci Berlangganan Produk

Untuk mengambil kunci langganan produk yang diterbitkan, navigasikan ke halaman Profil di Portal Pengembang (lihat Gambar 2-51). Anda akan melihat ada dua set kunci untuk Produk BRT.DEV dan produk BRT.OPS. Setiap produk menyediakan dua kunci, primer dan sekunder. Setiap kali Anda diberikan kunci di Azure, Anda selalu diberikan dua kunci sehingga Anda dapat menjaga kunci utama tetap pribadi dan membagikan kunci sekunder. Jika masalah

keamanan muncul, Anda selalu dapat menggunakan kunci utama untuk mengakses layanan dan membuat ulang kunci sekunder untuk mencegah akses tambahan oleh pengguna kunci sebelumnya.

The screenshot displays the 'Business in Real Time API' developer portal. The top navigation bar includes 'HOME', 'APIS', 'PRODUCTS', 'APPLICATIONS', 'ISSUES', and 'ADMINISTRATOR'. The main content area is divided into three sections: 'Profile', 'Your subscriptions', and 'Your applications'. The 'Profile' section shows user details like email (bobf@bluemetal.com) and organization name (Business in Real Time). The 'Your subscriptions' section lists two active subscriptions: BRT.DEV and BRT.OPS, each with details on keys and start dates. The 'Your applications' section is currently empty, showing 'No results found'.

Gambar 2-51. Halaman profil Portal Pengembang

Catat kunci sekunder untuk kedua produk, karena Anda akan menggunakannya untuk mengakses API pada langkah selanjutnya.

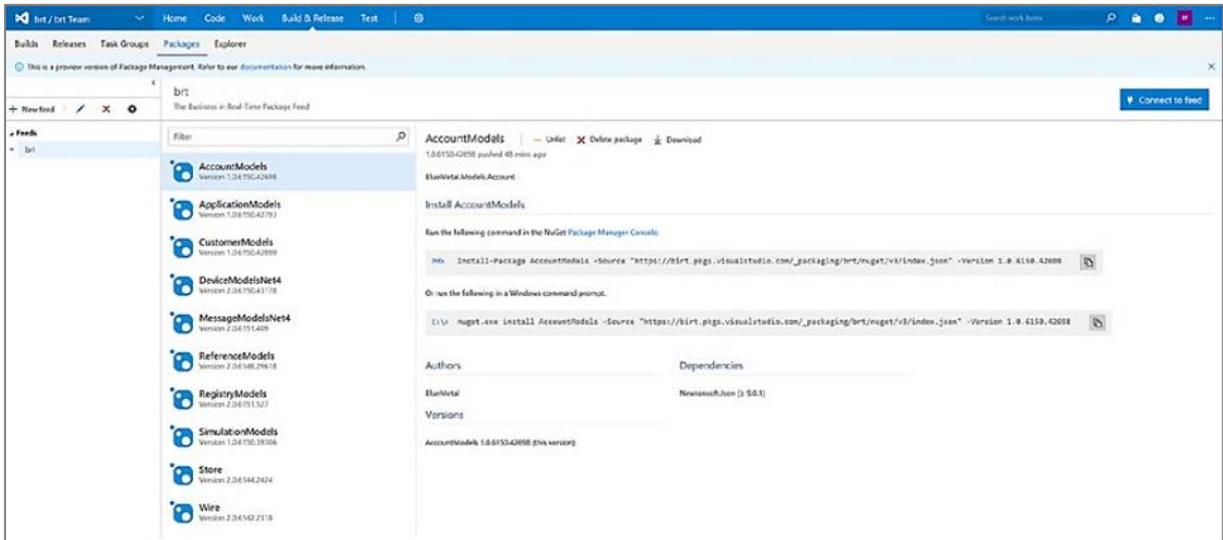
Selesaikan Penerapan

Untuk menyelesaikan penerapan, Anda perlu membuat definisi build untuk model dan layanan mikro lainnya. Ingatlah bahwa Anda dapat menyimpan definisi build yang ada sebagai templat sehingga Anda tidak perlu memulai dari awal saat membuat definisi baru. Gunakan fitur ini untuk membuat templat untuk penerapan layanan mikro berdasarkan definisi build Referensi Penyebaran Layanan Mikro. Membuat dan mengantri definisi build untuk paket NuGet yang tercantum pada Tabel 2-11. Feed NuGet lengkap digambarkan pada Gambar 2-52.

Tabel 2-11. Bisnis dalam NuGets Waktu Nyata

Project Name	Location
Account Model	models/account
Application Model	models/application
Customer Model	models/customer

Device ModelNet4	models/device/net4/
Message ModelNet4	models/message/net4
Registry Model	models/registry
Simulation Model	models/simulation



Gambar 2-52. Pakan paket lengkap

Untuk mengaitkan solusi layanan mikro dengan umpan NuGet, Anda perlu membuka masing-masing solusi di Visual Studio dan melakukan pembangunan lokal serta memeriksa solusi yang diperbarui. Selanjutnya, buat definisi build penerapan berikut:

- Menerapkan Layanan Mikro Akun
- Menyebarkan Layanan Mikro Aplikasi
- Menyebarkan Layanan Mikro Pelanggan
- Menerapkan Layanan Mikro Perangkat
- Menyebarkan Layanan Mikro Registri
- Menerapkan Layanan Mikro Simulasi

Saat Anda menjalankan build ini untuk pertama kalinya, atur parameter input `DeployData` ke `true` sehingga database dan koleksi DocumentDB dibuat. Nama database dan koleksi untuk masing-masing layanan mikro ditunjukkan pada Tabel 2-12.

Tabel 2-12. Basis Data dan Koleksi

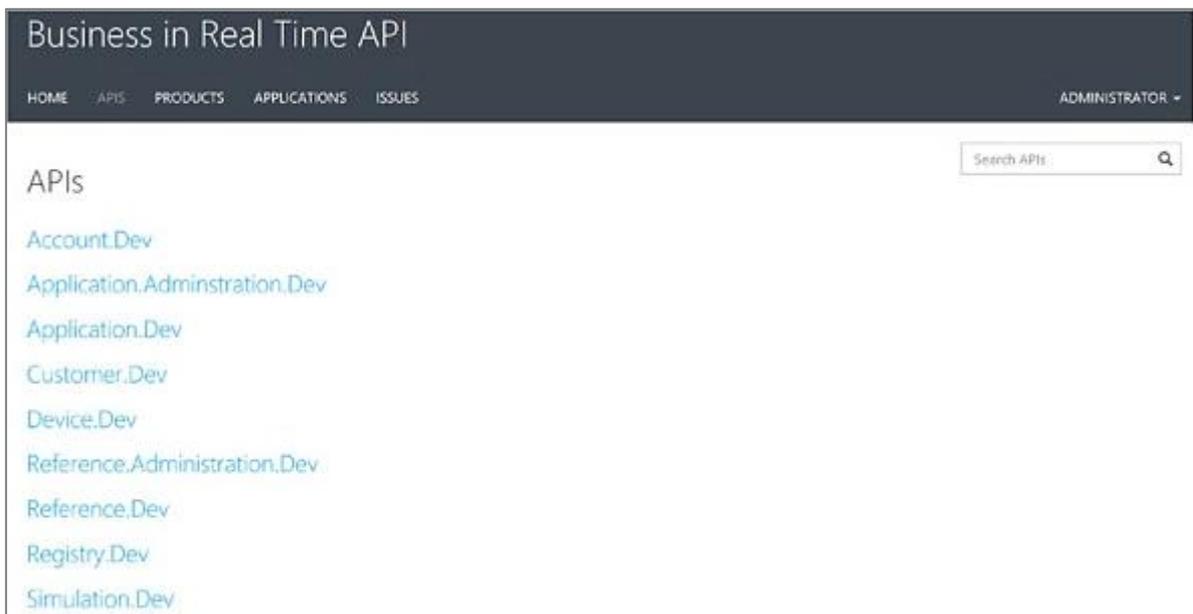
Account	Database Collection	Account Subscription
Application	Database Collection	Application Configuration
Customer	Database Collection	Customer Organization
Device	Database Collection	Device Manifest

Registry	Database Collection	Registry Profile
Simulation	Database Collection	Simulation DataSet

Perbarui dan impor definisi Swagger dan konfigurasi kebijakan rahasia bersama untuk API berikut:

- API Akun
- API Aplikasi
- API Administrasi Aplikasi
- API Pelanggan
- API Perangkat
- API Registri
- API Simulasi

Seperti yang ditunjukkan pada Gambar 2-53, penerapan dan konfigurasi layanan mikro telah selesai. Pada latihan berikutnya, Anda akan melakukan bootstrap penerapan dengan data sampel dan menyediakan sekumpulan perangkat simulasi IoT.



Gambar 2-53. Kumpulan API lengkap

Bootstrap Penyerapan

Setelah Anda menyelesaikan proses penyediaan dan penerapan serta mengonfigurasi API terkelola, Anda dapat menggunakan utilitas Bootstrap untuk menginisialisasi lingkungan dengan data sampel dan menyediakan serangkaian perangkat simulasi.

Utilitas Bootstrap dapat ditemukan di folder `utilities\bootstrap`. Utilitas akan melakukan operasi berikut.

1. *Buat Profil Registri:* Buat tiga profil perusahaan palsu dan 16 profil karyawan per perusahaan. Perusahaan palsu tersebut adalah:
 - WigiTech: Produsen komponen dan produk berteknologi tinggi
 - Menara Tinggi: Perusahaan layanan utilitas yang mengkhususkan diri dalam melayani menara radio di gedung pencakar langit
 - The Complex Badger: Sebuah perusahaan angkutan truk yang mengkhususkan diri dalam memindahkan alat berat
2. *Buat Organisasi Pelanggan:* Tentukan kumpulan tiga tim per perusahaan di mana setiap tim terdiri dari lima karyawan. Karyawan ke-16 per perusahaan mewakili manajemen.
3. *Buat Langganan Akun:* Tentukan tiga akun pelanggan untuk tiga perusahaan dan tentukan tingkat langganannya.
4. *Buat Konfigurasi Aplikasi:* Karena setiap perusahaan menggunakan solusi multi-penyewa yang sama, informasi ini mewakili penyesuaian solusi untuk perusahaan itu. Beberapa contoh sederhananya adalah logo, warna, dll., namun dapat diperluas ke penyesuaian yang lebih canggih yang mungkin memerlukan akses ke modul tingkat lanjut.
5. *Buat Manifes Perangkat:* Buat registrasi perangkat dan manifes terkait untuk setiap karyawan. Setiap perangkat dikaitkan dengan seorang karyawan dan mewakili serangkaian sensor yang mengumpulkan data biometrik.

Operasi ini perlu dijalankan secara berurutan, karena setiap langkah dalam proses pembuatan data sampel didasarkan pada langkah sebelumnya. Untuk menggunakan utilitas Bootstrap, Anda perlu memperbarui aplikasi agar dapat menggunakan API yang Anda kelola. Muat solusinya di Visual Studio dan buka file app.config. Perbarui Pengaturan Aplikasi dengan menambahkan kunci langganan Anda dari manajemen API dan nama host manajemen API untuk setiap API.

```
<add key="DevSubKey"
value="subscription-key=[your-dev-key]" />
<add key="OpsSubKey"
value="subscription-key=[your-dev-key]" />
<add key="ProfileAPI"
value="https://[your-apim-host].azure-api.net/dev/v1/registry"/>
<add key="AccountAPI"
value="https://[your-apim-host].azure-api.net/dev/v1/account"/>
<add key="ApplicationAPI"
  value="https://[your-apim-host].azure-api.net/dev/v1/application"/>
<add key="CustomerAPI"
value="https://[your-apim-host].azure-api.net/dev/v1/customer"/>
<add key="DeviceAPI"
  value="https://[your-apim-host].azure-api.net/dev/v1/device"/>
```

■ Catatan Utilitas Bootstrap menggunakan Microsoft Access Database Engine untuk membaca data sampel profil dari Excel.

Jika lingkungan pengembangan Anda tidak menginstal runtime ini, utilitas akan mengeluarkan pengecualian. Instal Access Database Engine dari lokasi ini:

<https://www.microsoft.com/en-us/download/details.aspx?id=13255>

Bangun dan jalankan utilitas (lihat Gambar 2-54). Anda akan disajikan dengan menu. Pilih opsi 1 dan tekan Enter.

```

file:///C:/Users/bob/Source/Repos/brt/utilities/Bootstrap/Bootstrap/bin/Debug/Bootstrap.EXE
* BUSINESS IN REAL - TIME BOOTSTRAP *
Execute these steps in the specified order.
Select an operation from the menu and press [Enter]
1. Create Registry Profiles
2. Create Customer Organizations
3. Create Account Subscriptions
4. Create Application Configurations
5. Create Device Manifests
0. Exit
Choice [0-5]:

```

Gambar 2-54. Utilitas bootstrap

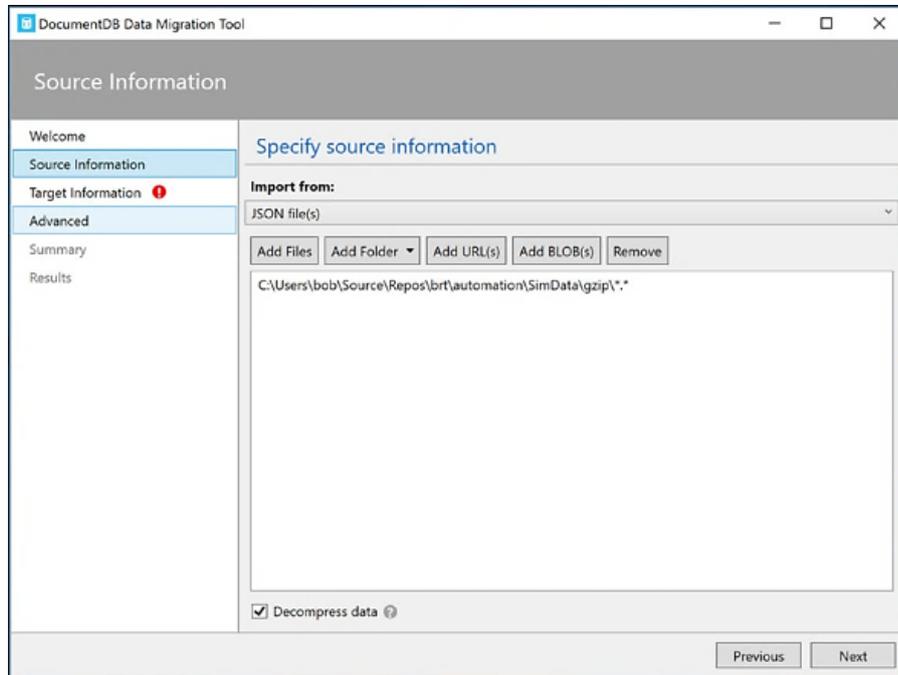
Anda akan kembali ke menu utama setelah menyelesaikan setiap langkah. Lanjutkan melalui opsi menu 2 hingga 5 untuk menyelesaikan proses pembuatan data sampel. Jika Anda mengalami masalah apa pun, kemungkinan besar masalah tersebut terkait dengan pengaturan konfigurasi API. Tinjau pengaturan API terkelola Anda dan uji dalam Portal Pengembang untuk memastikan API Anda berfungsi dengan baik.

Beban Simulasi Data

Untuk memuat data simulasi, Anda akan menggunakan Alat Migrasi Data DocumentDB (DT). Ini adalah utilitas gratis dari Microsoft yang menyediakan versi aplikasi baris perintah dan GUI. Versi baris perintah DT digunakan oleh penerapan skrip PowerShell untuk membuat database dan koleksi DocumentDB serta mengunggah data referensi sampel. Karena ukuran data simulasi jauh lebih besar daripada data referensi, kami akan melakukan operasi ini dari lingkungan pengembangan kami.

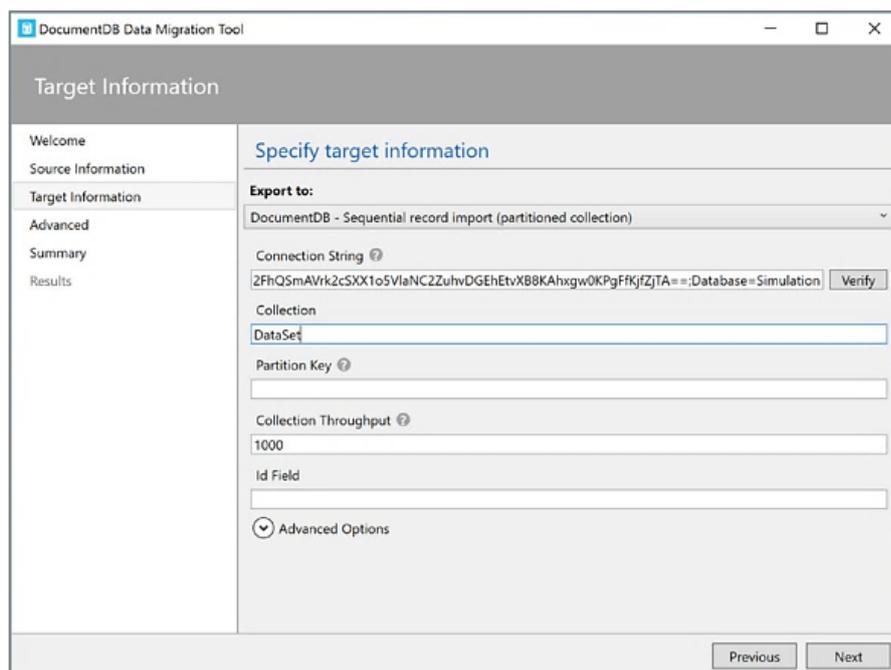
Data simulasi disediakan sebagai kumpulan dokumen JSON yang di-gzip di folder `automation\simdata\gzip`. Terdapat ~64KB dokumen data simulasi di mana setiap dokumen mewakili kumpulan pembacaan biometrik dari 1 dari 15 rekan satu tim. Utilitas

desktop DT ada di folder `automation\Tools\dt\dt-1.7` dan disebut `DTUI.EXE`. Jalankan utilitas dan konfigurasi sumbernya menjadi folder gzip. Centang kotak `Decompress Data` (lihat Gambar 2-55).



Gambar 2-55. Utilitas migrasi data DocumentDB

Klik Berikutnya dan tambahkan string koneksi ke instance DocumentDB Anda. Tambahkan nama database, Simulasi, ke akhir koneksi menggunakan sintaks `Database=Simulasi;`. Tetapkan nama koleksi menjadi `DataSet` (lihat Gambar 2-56).



Gambar 2-56. Berikan string koneksi, database, dan nama koleksi

Klik Berikutnya hingga Anda mencapai layar terakhir dan menjalankan operasi impor. Utilitas akan mendekomposisi file data dan mengunggahnya ke DocumentDB. Setelah proses selesai, Anda dapat memvalidasi proses dengan menguji API simulasi menggunakan manajemen API. Navigasikan ke Portal Pengembang dan pilih titik akhir Dapatkan Himpunan Data Simulasi berdasarkan Nama, klik Coba, lalu berikan nama `teammate1` sebagai parameternya. Klik Kirim. Jika impor data berhasil, Anda akan menerima kode respons 200 dan JSON akan ditampilkan di jendela keluaran (lihat Gambar 2-57).

The screenshot displays the 'Simulation.Dev' interface in the Azure API Management portal. The left sidebar contains a list of API actions, with 'GET Get a Simulation Dataset by Name' selected. The main area shows the configuration for this API call, including query parameters, headers, and authorization details. The 'name' query parameter is set to 'teammate1'. The 'Ocp-Apim-Trace' header is set to 'true', and the 'Ocp-Apim-Subscription-Key' header is masked with asterisks. The 'Request URL' is `https://birt.azure-api.net/dev/v1/simulation/datasets/name/teammate1`. The 'HTTP request' section shows the raw request: `GET https://birt.azure-api.net/dev/v1/simulation/datasets/name/teammate1 HTTP/1.1` with headers: `Host: birt.azure-api.net`, `Ocp-Apim-Trace: true`, and `Ocp-Apim-Subscription-Key: *****`. Below the request, the 'Response' tab is active, showing a status of '200 OK', a latency of '46739 ms', and a JSON response content.

Response **Trace**

Response status
200 OK

Response latency
46739 ms

Response content

```

Pragma: no-cache
Ocp-Apim-Trace-Location: https://apimgmtst3ghpmpgm4njv0f.blob.core.windows.net/apiinspector/container/yqTdeFksNyp11Exsrzu6gA2-4?sv=2015-07-08&sr=b&sig=glet8uQLo22uyd90Fg1jfnf10b0y42Bh8wa1QaLAN2BF8g4%3D&se=2016-11-03T23%3A49%3A23Z&sp=r&traceId=8f57a73457aa472fb3a2b9bff2a50cf7
Cache-Control: no-cache
Date: Wed, 02 Nov 2016 23:50:09 GMT
Set-Cookie: ARRAffinity=56646929c62d805e2f4e8ce9726ccc1bfe6e992d29e06ca36cedelba3e46d768;Path=/;Domain=brtsimulationapidev.azurewebsites.net
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Content-Length: 163886
Content-Type: application/json; charset=utf-8
Expires: -1

```

```

{
  "name": "teammate1",
  "rows": [
    "664da22f-12cd-406e-a03a-8332a597dbb2",
    "48ae820e-8cdc-4d58-8297-331cb60c0920",
    "3985e739-95a7-417b-84aa-351c418d81e4",
    "866efe1c-67dd-4d78-9917-fcd4f85092bc",
    "28bd1c58-1b95-4446-9765-f22925e13af0",
    "29c61a38-c7e1-467a-ab06-7e4afbc6aef",
    "1476fb96-29f4-4c83-b961-e837b85710be",
    "0eab747e-f0f0-4f3d-aff4-b685b1f9ad56",
    "9f8dc477-0440-44c8-841a-5a1507cad06",
    "cc5e302f-1e42-44a8-91ee-57f1ae37cb7b",
    "d792976d-4937-4364-966f-cc5fa70edf6f",
    "4752d00d-2869-4bf4-b0f4-c169c9b0f5ff",
    "d12382cc-6106-40bc-aea1-f9bdc7772fb",
    "24e18d53-6d06-4dd6-88fb-d72f4cb1b7a6",
    "cf3ff929-fc49-4026-9f4b-b418eeda704c",
    "404d4e5b-b521-4cb3-b0ef-aec6e0282d34",
    "458a4511-6043-4f1f-9735-cd2256e681de",
    "acc50856-08c2-4cd0-b029-1487232b4a50",
    "8b63e03-1687-4452-b815-cbd7a5b01463",
    "ace33cc1-2b96-47c9-9aca-63c252f0c3c7",
    "bb0ff865-495f-4786-8790-5a1a475c0c5c",
    "35af05b1-8809-425e-bd7b-0eec2b7a9000",
    "a6d6d5ac-5d7b-4287-8306-6be559e81b88",
    "c616eb17-8958-4258-8dfb-d74fb54df11e",
  ]
}

```

Gambar 2.57. Simulasi Test API

2.7 RINGKASAN

DevOps adalah tentang manusia, proses, dan alat. Hal ini menyiratkan penciptaan tim lintas fungsi, menggabungkan pengembang, pengujian, dan arsitek bersama dengan personel operasi yang bersama-sama memiliki seluruh jalur penerapan, mulai dari pembangunan hingga pengujian hingga pementasan dan produksi. Hal ini mengharuskan tim-tim ini bekerja secara kolaboratif untuk mengadopsi proses dan alat yang umum. Kemampuan Anda untuk menghadirkan perangkat lunak modern yang memanfaatkan arsitektur layanan mikro dan platform cloud berakar pada kemampuan Anda untuk mengatur tim, menentukan proses berkelanjutan, dan menggunakan seperangkat alat dan teknologi yang tepat untuk mengotomatiskan siklus hidup produk perangkat lunak Anda.

Dalam bab ini, kita memeriksa bagaimana seseorang dapat menggunakan PowerShell, templat manajemen sumber daya Azure, dan Visual Studio Team Services untuk membuat definisi otomatisasi yang menangani provisi layanan Azure, membuat paket dan layanan perangkat lunak, serta menyebarkan build perangkat lunak ke Azure. Setiap langkah dari proses integrasi berkelanjutan dan pengiriman berkelanjutan dijalankan di Azure, memberikan pengalaman yang lancar bagi tim DevOps. Selain itu, skrip dirancang agar operasi ini dapat dilakukan terhadap lingkungan pengembangan, pengujian, atau produksi hanya dengan memodifikasi parameter input.

Jika Anda telah berhasil menyelesaikan latihan dalam bab ini, Anda berada di posisi yang tepat untuk memaksimalkan pembelajaran Anda dari bab berikutnya seiring kami terus mengembangkan IoT yang baru disediakan dan solusi analitik tingkat lanjut.

BAB 3

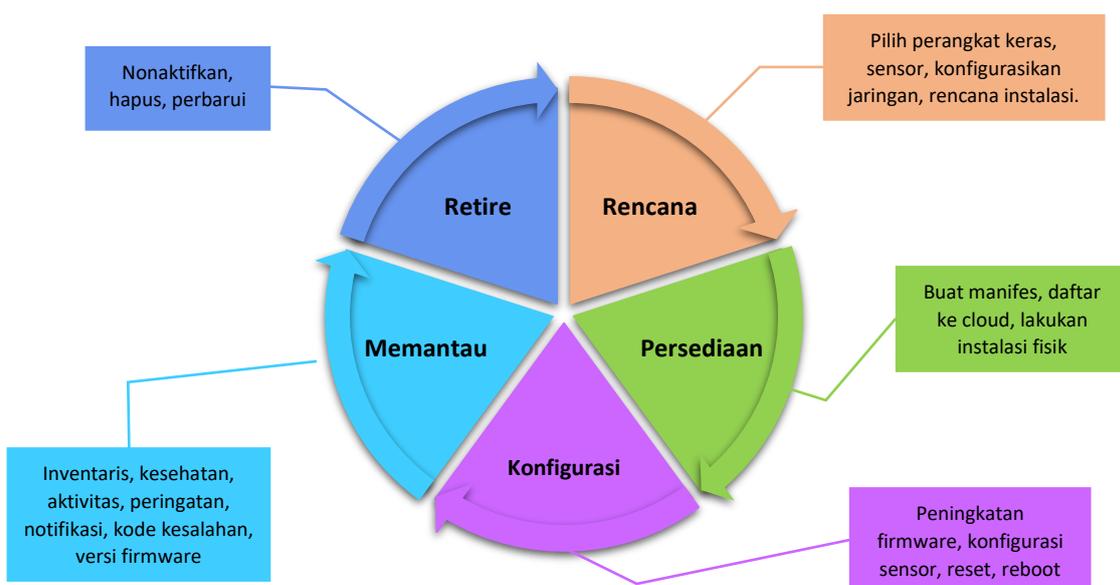
MANAJEMEN PERANGKAT MENGGUNAKAN IOT HUB

Menghubungkan orang, tempat, dan berbagai hal ke cloud, meskipun bukan hal yang sepele, mungkin merupakan salah satu aspek IoT yang lebih mudah karena teknik dan protokolnya didefinisikan dengan sangat baik. Pekerjaan sebenarnya dimulai ketika Anda memiliki ribuan, mungkin jutaan, perangkat yang terhubung ke cloud dan Anda perlu mengelola operasi sehari-hari dari sistem yang sangat terdistribusi ini. Selain memantau dan mengelola layanan cloud yang menyediakan analitik, penyimpanan, dasbor, peringatan, dan notifikasi, Anda juga perlu memantau dan mengelola beacon, perangkat, dan edge gateway Anda.

Apa inventaris perangkat saya? Apakah perangkat dihidupkan dan terhubung? Apakah mereka melakukan pembacaan sensor? Apakah mereka mengirim pesan telemetri? Apakah ada perangkat yang melaporkan kesalahan? Apakah mereka perlu di-boot ulang atau dikonfigurasi ulang? Versi firmware apa yang dijalankan, dan apakah kami perlu mengupgrade perangkat apa pun ke revisi firmware baru? Kondisi perangkat ini perlu didiagnosis dan dikelola dari jarak jauh dan aman oleh tim operasi menggunakan serangkaian layanan perangkat lunak yang diakses melalui dasbor Manajemen Perangkat yang cerdas.

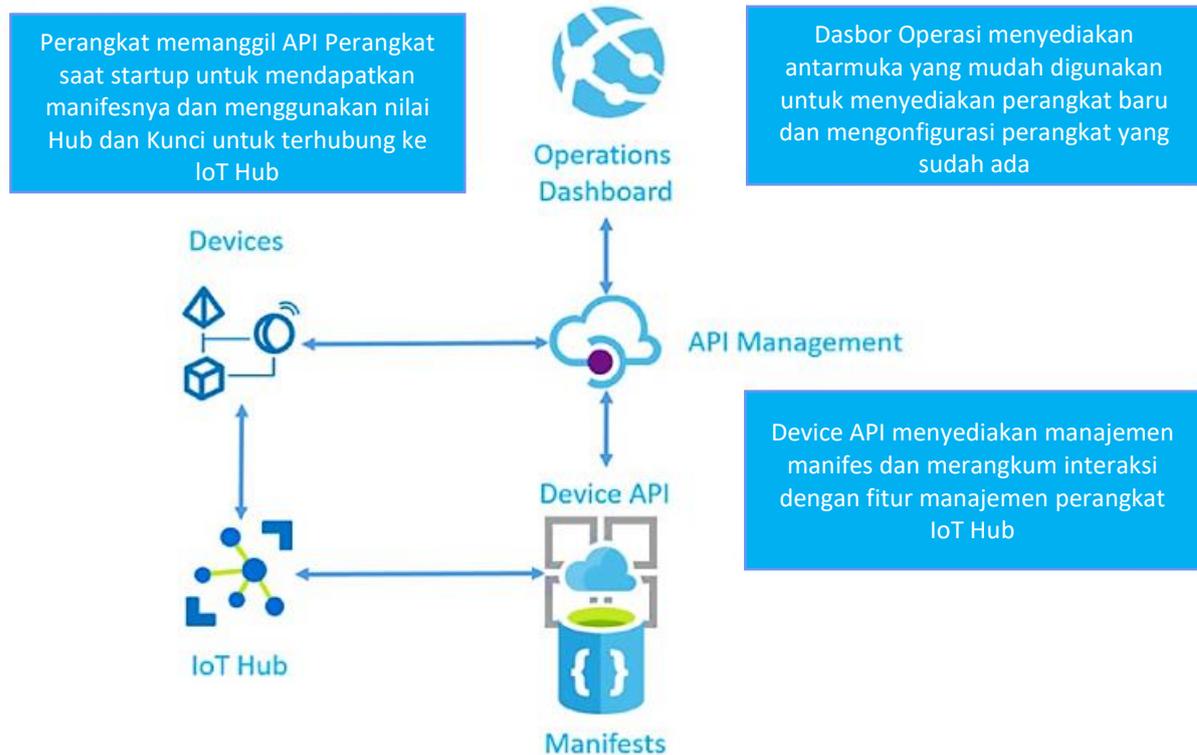
3.1 SIKLUS HIDUP MANAJEMEN PERANGKAT

Tim operasi akan memerlukan seperangkat alat perangkat lunak yang mendukung proses yang terdefinisi dengan baik untuk mengelola perangkat sepanjang masa pakainya, mulai dari penyediaan dan konfigurasi hingga penghentian (lihat Gambar 3-1).



Gambar 3-1. Siklus hidup manajemen perangkat

Bab ini akan membawa Anda melewati setiap fase siklus hidup manajemen perangkat. Di bagian akhir bab ini terdapat kumpulan latihan yang menunjukkan pola manajemen perangkat umum termasuk reboot, upgrade firmware, dan konfigurasi sensor. Sampel Bisnis Real-Time menyediakan dasbor operasi, API perangkat terkelola, dan simulator perangkat yang bekerja bersama dengan Azure IoT Hub untuk mendemonstrasikan layanan perangkat lunak manajemen perangkat (lihat Gambar 3-2).



Gambar 3.2. Layanan perangkat lunak siklus hidup manajemen perangkat

Perencanaan

Saat merencanakan strategi perangkat IoT, ada beberapa bidang penelitian, desain, dan proses yang perlu dipertimbangkan. Setiap solusi IoT memiliki tantangan uniknya sendiri, mengingat lingkungan tempat perangkat tersebut digunakan. Seperti halnya proyek apa pun, tahap perencanaan sangat penting untuk keberhasilan. Tujuannya adalah untuk menghilangkan asumsi dan mengurangi risiko.

Berikut beberapa contoh pertanyaan yang, jika dijawab, akan membantu menghilangkan asumsi dan mengurangi risiko sehingga menghasilkan strategi pengelolaan perangkat yang jelas:

- *Platform Perangkat Keras dan Perangkat Lunak:* Apakah Anda akan menggunakan perangkat IoT dan edge gateway yang tersedia atau Anda memerlukan enclosure yang dibuat khusus? Bagaimana cara kerjanya? Bagaimana kondisi lingkungan tempat perangkat dan edge gateway akan dikerahkan? Suhu? Kelembaban? Debu? Berapa umur perangkat keras yang diharapkan? Sistem operasi apa yang akan diinstal?
- *Komunikasi:* Mekanisme komunikasi apa yang Anda perlukan? Seluler, nirkabel, kabel, atau kombinasi apa pun?

- *Sensor*: Sensor apa yang akan digunakan? Apa saja kebutuhan dayanya? Bagaimana cara menghubungkannya ke platform perangkat keras? Koneksi langsung, gateway sensor, Modbus, BACnet, CAN bus, atau lainnya? Berapa kisaran pembacaan yang menentukan kondisi normal, peringatan, dan peringatan untuk setiap sensor? Seberapa sering pesan dikirim untuk setiap jenis pembacaan, dan data apa yang terkandung dalam pesan tersebut?
- *Analisis*: Apakah Anda memerlukan analisis terkini? Jika ya, operasi matematika atau analisis Machine Learning apa yang perlu dilakukan? Berapa banyak bacaan yang perlu dikumpulkan sebelum menerapkan operasi ini? Pemfilteran apa, jika ada, yang perlu diterapkan?
- *Penyimpanan Lokal*: Apakah Anda memerlukan penyimpanan lokal pada tingkat tertentu untuk analisis edge atau peninjauan di lokasi?
- *Peringatan dan Alarm Lokal*: Berdasarkan analisis edge, apakah Anda perlu membuat peringatan dan pemberitahuan alarm dari perangkat atau edge gateway? Bagaimana hal itu akan dilakukan?
- *Konfigurasi Jaringan*: Bagaimana perangkat dan edge gateway akan terhubung ke jaringan? Apakah itu perangkat tambahan di jaringan yang ada? Di segmen mereka sendiri? Apa saja persyaratan firewallnya? Bagaimana jaringan dikelola secara lokal?
- *Keamanan*: Bagaimana perangkat dan edge gateway diamankan secara fisik dan digital? Bagaimana perangkat dan edge gateway berkomunikasi secara lokal? Bagaimana data dienkripsi saat terbang dan diam?
- *Proses Instalasi Perangkat Keras*: Siapa yang bertanggung jawab atas pengaturan dan konfigurasi perangkat dan gateway? Apakah hal ini memerlukan keterampilan jaringan dan keamanan, atau bisakah seseorang yang tidak memiliki pengetahuan tentang jaringan dan keamanan mengatur lingkungannya? Dokumentasi dan materi pelatihan apa yang perlu dibuat dan disediakan? Apakah ini tersedia dalam bentuk hardcopy atau secara digital melalui aplikasi seluler yang cerdas?
- *Pengelompokan Perangkat*: Bagaimana Anda mengelompokkan perangkat dan mengontrol akses sesuai dengan kebutuhan organisasi Anda?

Mengembangkan strategi manajemen perangkat yang komprehensif adalah dasar dari solusi IoT yang sukses. Hal ini memerlukan kolaborasi antara operasional dan insinyur perangkat lunak dan perangkat keras untuk merancang pendekatan yang aman dan mudah dikelola terhadap penyediaan, penerapan, dan proses pengoperasian yang akan menjangkau puluhan ribu, mungkin jutaan, perangkat.

Penyediaan

Penyediaan adalah proses pemasangan sistem operasi dan firmware, pembuatan ID perangkat unik, pendaftaran perangkat di cloud, dan dokumen digital (manifes) dibuat untuk menyimpan dan melacak detail perangkat sepanjang masa pakainya. Tugas menghasilkan pengidentifikasi unik mungkin melibatkan integrasi dengan sistem lini bisnis atau sistem Perencanaan Sumber Daya Perusahaan (ERP) seperti SAP. Sistem ERP sering digunakan untuk menyediakan data master untuk kode produk, melacak inventaris, dan menyediakan layanan

pelaporan aset. Sistem CRM dapat memberikan informasi pelanggan seperti alamat fisik perusahaan, produk dan layanan apa yang telah dibeli oleh pelanggan, dan lokasi di mana perangkat dan gateway akan dipasang. Selain itu, informasi seperti koordinat geolokasi dan detail seperti nomor bangunan, lantai, dan ruangan dapat diberikan.

Metadata perangkat akan memberikan layanan perangkat lunak rincian yang diperlukan untuk mengatur proses manajemen perangkat yang komprehensif. Mari kita periksa bagaimana informasi ini digunakan oleh layanan perangkat lunak Device API dan Azure IoT Hub.

3.2 PUSAT IOT AZURE

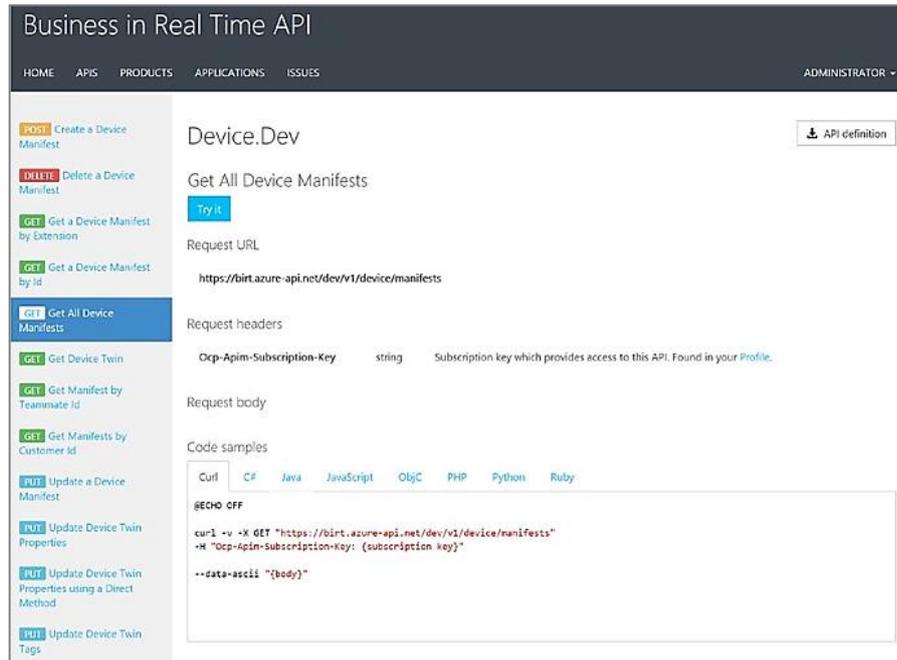
Azure IoT Hub adalah layanan terkelola sepenuhnya yang menyediakan layanan komunikasi dan manajemen aman untuk perangkat Anda. Ini termasuk:

- Komunikasi perangkat-ke-cloud dan cloud-ke-perangkat untuk komunikasi dua arah yang aman serta transfer file menggunakan pesan perintah dan kontrol asinkron serta metode permintaan dan balasan sinkron
- Penyimpanan sementara berbasis hub peristiwa yang sangat skalabel yang menyediakan penyimpanan pesan dari satu hingga tujuh hari dan akses berbasis kursor ke aliran peristiwa yang menyediakan pengelolaan dan sinkronisasi status
- Komunikasi aman menggunakan WebSockets pada protokol port 443, AMQP, MQTT, atau HTTPS
- Memantau konektivitas perangkat, aliran peristiwa, dan peristiwa manajemen identitas
- SDK untuk banyak bahasa dan sistem operasi populer
- Kemampuan manajemen perangkat
- Perutean pesan deklaratif bawaan ke layanan Azure lainnya

Azure IoT Hub menyediakan serangkaian kemampuan manajemen perangkat termasuk Pendaftaran Perangkat, Kembaran Perangkat, Metode Langsung, Pekerjaan, dan Kueri. Sebelum kita mendalami Azure IoT Hub, mari kita periksa layanan mikro perangkat, yaitu ReST API terkelola yang merangkum panggilan ke IoT Hub serta mengelola repositori dokumen yang melacak metadata setiap perangkat sepanjang masa pakainya.

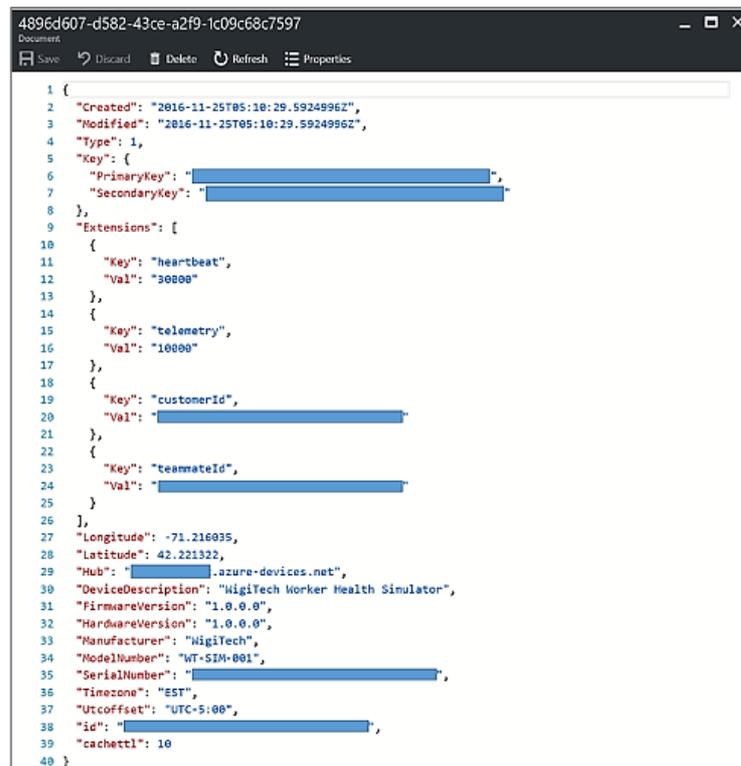
API Perangkat dan Manifes Perangkat

Implementasi referensi Bisnis dalam Waktu Nyata menyediakan API terkelola yang disebut Perangkat yang digunakan untuk membuat dan mengelola metadata tentang setiap perangkat serta merangkum operasi yang menggunakan fitur manajemen perangkat IoT Hub. Jika Anda melakukan latihan di Bab 2, Anda akan memiliki instance API Perangkat terkelola yang diterapkan sepenuhnya, seperti yang ditunjukkan pada Gambar 3-3.



Gambar 3.3. API perangkat yang dikelola

Device API menyediakan operasi untuk membuat, memperbarui, dan menanyakan dokumen JSON yang disebut Manifest. Manifest dibuat pada awal proses penyediaan dan diinisialisasi dengan informasi yang diperlukan organisasi untuk melacak perangkat sepanjang siklus hidupnya. Setiap kali perangkat dikonfigurasi, diperbarui, atau dicabut provisinya, manifest diperbarui, sehingga memberikan catatan historis (lihat Gambar 3-4).



Gambar 3-4. Dokumen JSON manifes perangkat

Manifes perangkat menyediakan kumpulan properti berguna seperti:

- *Dibuat*: Tanggal/waktu dokumen manifes dibuat.
- *Dimodifikasi*: Tanggal/waktu dokumen manifes terakhir diperbarui.
- *Tipe*: Jenis perangkat; simulator, perangkat pintar, edge gateway, ponsel, dll.
- *Nomor Seri*: ID unik untuk perangkat. Nilai ini digunakan untuk mendaftar ke Azure IoT Hub dan untuk mengidentifikasi pesan secara unik.
- *Bujur/Lintang*: Geolokasi perangkat. Nilai ini dapat ditetapkan pada waktu penyediaan dan kemudian diperbarui secara dinamis jika perangkat memiliki kemampuan GPS. Hal ini sangat berguna terutama untuk perangkat yang sedang bergerak atau skenario di mana Anda ingin memetakan lokasi perangkat.
- *Nomor Pabrik/Model*: Nomor pabrik dan model perangkat fisik.
- *Versi Firmware/Versi Perangkat Keras*: Detail versi firmware yang diinstal pada perangkat dan versi perangkat keras.
- *Ekstensi*: Daftar pasangan kunci/nilai yang dapat digunakan untuk memperluas model data. Penerapan kami menggunakan koleksi ekstensi untuk menyimpan detak jantung dan irama telemetri, ID profil pelanggan, dan ID profil rekan tim yang terkait dengan perangkat.
- *Hub dan Kunci*: Nama host dan kunci keamanan IoT Hub.

Device API menyediakan operasi Buat Manifes yang akan mendaftarkan perangkat ke IoT Hub dan menyimpan manifes di DocumentDB. Berikut adalah contoh kode C# yang membuat manifes untuk perangkat simulasi dan memanggil API Perangkat untuk melakukan registrasi. Manifes diteruskan sebagai payload ke panggilan ReST.

```
// initialize a new device manifest
var manifest = new Manifest
{
    Latitude = [latitude-value],
    Longitude = [longitude-value],
    SerialNumber = [unique-device-id],
    Manufacturer = "[hardware-manufacturer-name]",
    ModelNumber = "[hardware-model-number]",
    FirmwareVersion = "[firmware-version]",
    HardwareVersion = "[hardware-version]",
    DeviceDescription = "simulator",
    Type = DeviceTypeEnum.Simulator,
    Timezone = "EST",
    UtcOffset = "UTC-5:00"
};

// add property extensions
manifest.Extensions.Add(
    new DeviceProperty("heartbeat", "30000"));
manifest.Extensions.Add(
    new DeviceProperty("birthdate", "10000"));
```

```

manifest.Extensions.Add(
    new DeviceProperty("customerId", customerProfile.id));
manifest.Extensions.Add(
    new DeviceProperty("teammateId", teammate.id));

// invoke the Create Manifest operation on the Device API
var uri = @"https://[apim-host].azureapi.net/dev/v1/device/manifests";
var uriBuilder = new UriBuilder(uri)
{
    Query = "subscription-key=[dev-key]"
};

var json = JsonConvert.SerializeObject(manifest);

Rest.Post(uriBuilder.Uri, json);

```

Panggilan API akan mendaftarkan perangkat ke IoT Hub, menambahkan nama host IoT Hub dan kunci simetris ke manifes, dan menyimpan manifes di DocumentDB.

Untuk menyambung ke IoT Hub, perangkat memerlukan tiga informasi: (1) ID perangkat unik, (2) nama host IoT Hub, dan (3) kunci simetris yang dihasilkan saat perangkat didaftarkan. Anda dapat menyematkan nilai-nilai ini di firmware. Namun, jika salah satu detail ini berubah, pembaruan firmware diperlukan untuk menyegarkan perangkat dengan informasi baru. Teknik lainnya adalah dengan menggunakan teknik penerapan tanpa sentuhan di mana perangkat IoT memanggil API saat startup untuk mengambil metadata. Satu-satunya informasi yang tertanam dalam firmware adalah ID perangkat unik dan titik akhir API yang mengembalikan manifes perangkat.

Implementasi kami menggunakan teknik API tanpa sentuhan ini. Perangkat memanggil Device API saat startup untuk mengambil manifesnya. Manifes akan memiliki dua informasi tambahan nama host dan kunci simetris. Pendekatan ini memberikan fleksibilitas karena lokasi IoT Hub dapat diubah dan perangkat, setelah reboot atau melalui perintah jarak jauh, akan memuat ulang manifesnya dan terhubung ke instans IoT Hub baru tanpa modifikasi pada firmware. Berikut contoh kode Node.js untuk perangkat yang menggunakan proses startup ini:

```

'use strict';

// refence the Azure IoT Hub SDK
var Client = require('azure-iot-device').Client;
var Protocol = require('azure-iot-device-mqtt').Mqtt;

var Manifest;
var Profile;
var DeviceClient;

```

```

// Add your unique device Id
var DeviceId = '[unique-device-id]';

var startup = function()
{
var uri = 'https://[apim-host].azureapi.net/dev/v1/device/manifests/id/'+
  DeviceId;
REST(uri, 'GET', function(data)
{
  Manifest = data;

// create the IoT Hub connection string
var connectionString = 'HostName=' + Manifest.Hub + ";" +
"DeviceId="+Manifest.SerialNumber+";"+"SharedAccessKey="+
  Manifest.Key.PrimaryKey

// connect to IoT Hub
DeviceClient=Client.fromConnectionString(connectionString, Protocol);
DeviceClient.open(function(err)
{
  if (err) {
    console.error('could not connect ' + err);
  } else {
    console.log('client connected to IoT Hub');

    //start sending telemetry
    sendTelemetry();
  }
});
});
}

```

3.3 KONFIGURASI DAN PEMANTAUAN

Setelah perangkat terdaftar dan terhubung, tim operasi akan memerlukan kemampuan untuk menyesuaikan konfigurasi perangkat di tingkat OS, firmware, dan aplikasi serta melakukan operasi ini dari jarak jauh melalui dasbor operasi umum. Untuk mendukung operasi konfigurasi yang sedang berlangsung di seluruh ekosistem perangkat Anda, IoT Hub menyediakan fitur manajemen perangkat berikut:

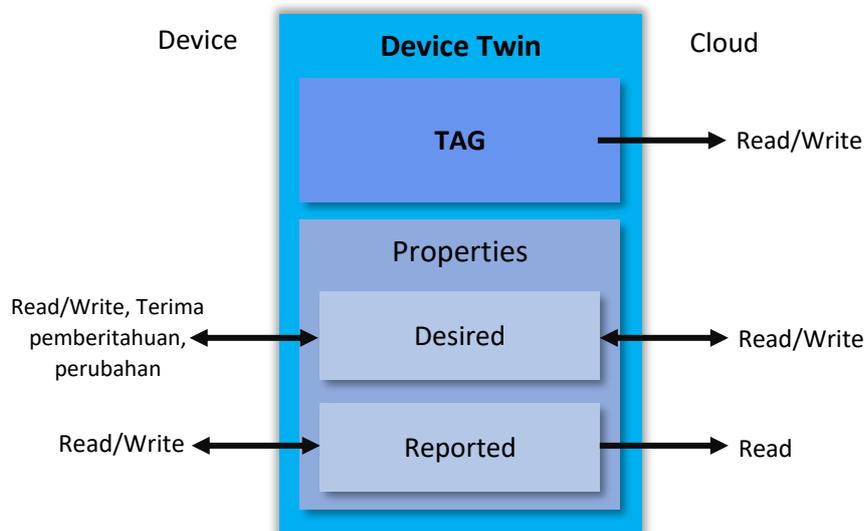
- *Device Twin*: Representasi digital virtual dari perangkat di cloud yang tetap tersinkronisasi dengan perangkat fisik
- *Tag*: Properti kembaran perangkat yang memungkinkan Anda mengatur informasi meta pada perangkat sehingga perangkat dapat dikueri berdasarkan nilai tag dan operasi yang dilakukan pada perangkat yang dipilih
- *Metode Langsung*: Kemampuan layanan yang dihosting di cloud untuk meminta operasi tingkat perangkat, seperti reboot atau pembaruan firmware, dilakukan menggunakan pola permintaan/respons

- *Kueri*: Pilih subkumpulan perangkat atau pekerjaan berdasarkan properti kembar perangkat dan nilai tag
- *Pekerjaan*: Saat operasi Device Twin, Tag, dan Metode Langsung dilakukan di beberapa perangkat, pekerjaan melacak kemajuan dan melaporkan status

Perangkat Kembar

Perangkat kembar adalah dokumen JSON yang menyimpan data meta dan konfigurasi. Si kembar tetap tersinkronisasi dengan perangkat fisik yang terhubung. Metadata kembar digunakan untuk melakukan kueri dan mendukung operasi konfigurasi yang berjalan lama seperti peningkatan firmware. Kembar dikelola oleh IoT Hub dan dibatasi ukurannya hingga 8K per koleksi properti—Tag, Diinginkan, dan Dilaporkan—dengan total 24KB per kembar. Perangkat kembar berisi:

- *Tag*: Atribut metadata yang dapat digunakan untuk mengarahkan kueri perangkat, misalnya, “pilih semua perangkat di gedung 3, lantai 27”. Nilai-nilai ini hanya dapat dibaca atau ditulis oleh backend.
- *Properti yang Diinginkan*: Properti yang digunakan untuk membuat permintaan konfigurasi oleh backend.
- *Properti yang Dilaporkan*: Properti konfigurasi seperti yang dilaporkan oleh perangkat. Properti yang diinginkan dan dilaporkan digunakan bersama untuk mengomunikasikan status yang diinginkan oleh backend dan status sebenarnya oleh perangkat (lihat Gambar 3-5).



Gambar 3.5. Model perangkat kembar

Tag

Tag digunakan untuk mengelompokkan perangkat secara logis berdasarkan kebutuhan organisasi. Untuk menambahkan tag ke perangkat kembar, Anda membuat fragmen JSON dengan format berikut:

```
var tags = '
  { "tags":
```

```

        { "tag-collection-name" :
          { "tag-name" : "tag-value", ... }
        }
      }

```

Misalnya:

```

var tags = '
  { "tags":
    { "location" :
      { "building" : "23", "floor" : "7" }
    }
  }'

```

Untuk menggunakan SDK Layanan IoT Hub, Anda mereferensikan paket Microsoft.Azure.Devices NuGet. Anda dapat menggunakan objek `RegistryManager` untuk mengambil kembarannya dan menerapkan pembaruan:

```

using Microsoft.Azure.Devices;
...
var _registryManager =
    RegistryManager.CreateFromConnectionString(iothubconnstr);
...
// get the twin
var twin = await _registryManager.GetTwinAsync(myDeviceId);

// update the twin with new tags
_registryManager.UpdateTwinAsync(myDeviceId, tags, twin.ETag)

```

Properti yang Diinginkan dan Dilaporkan

Properti yang diinginkan dan dilaporkan juga dapat digunakan untuk mengonfigurasi perangkat. Daripada mengirimkan perintah tertentu atau menggunakan metode langsung, backend dapat mengatur properti yang diinginkan. Perangkat diberitahu bahwa ada properti baru yang diinginkan dan dapat bertindak bila diperlukan. Perangkat mengontrol bagaimana dan kapan menerapkan perubahan konfigurasi. Untuk menyetel properti yang diinginkan, Anda membuat fragmen JSON dengan format ini:

```

var property = '
  { "properties":
    { "desired" :
      { "propertyCollectionName" :
        { "property" : "value" , ... }
      }
    }
  }';

```

Misalnya :

```
var properties = '
  { "properties":
    { "desired" :
      { "tempSensorConfig" :
        {
          "configId" : "1",
          "cadence" : "10000"
        }
      }
    }
  }';
```

Fragmen ini mendefinisikan permintaan untuk mengatur irama pengiriman pesan telemetri untuk sensor suhu menjadi setiap 10 detik. Layanan backend dapat memperbarui perangkat kembar dengan properti baru yang diinginkan ini menggunakan RegistryManager.

```
//get the twin
var twin = await _registryManager.GetTwinAsync(myDeviceId);

//update the twin's desired properties
_registryManager.UpdateTwinAsync(myDeviceId,properties,twin.ETag);
```

Perangkat dapat menggunakan kembaran perangkat untuk melaporkan nilai properti saat ini dan mengonfigurasi pengendali peristiwa untuk perubahan properti. Fragmen Node.js ini akan dijalankan pada perangkat untuk mengambil kembaran perangkat dan memperbarui properti yang dilaporkan.

```
client.getTwin(function(err, twin) {
  if (err) {
    console.error('could not get twin');
  } else {
    console.log('retrieved device twin');

    //reported temp sensor cadence
    twin.properties.reported.tempSensorConfig = {
      configId:"0",
      cadence:"30000",
    }

    //if a desired property is updated...
    twin.on('properties.desired', function(desiredChange) {
      console.log("received change");
      var currentTempSensorConfig=twin.properties.reported.tempSensorConfig;
```

```

//if the update is different than reported...
if(desiredChange.tempSensorConfig.configId !==
    currentTempSensorConfig.configId) {

    //make the configuration change
    initConfigChange(twin);
}
});
}

var initConfigChange = function(twin) {

    //get the reported configuration for temp sensor
    var currentTempSensorConfig =
        twin.properties.reported.tempSensorConfig;

    //get the pending change
    currentTempSensorConfig.pendingConfig =
        twin.properties.desired.tempSensorConfig;

    //report status
    currentTempSensorConfig.status = "Pending";

    var patch= { tempSensorConfig: currentTempSensorConfig };

    //report that an update is pending...
    twin.properties.reported.update(patch, function(err) {
    if (err) { console.log('Could not report properties');
        } else { console.log('Reported pending config change');

    setTimeout(function() {

        //complete the property update completeConfigChange(twin);}, 60000);
        }
    });
}

var completeConfigChange = function(twin) {

    var currentTempSensorConfig=twin.properties.reported.tempSensorConfig;

    currentTempSensorConfig.configId=
        currentTempSensorConfig.pendingConfig.configId;

    currentTempSensorConfig.cadence =
        currentTempSensorConfig.pendingConfig.cadence;

    currentTempSensorConfig.status = "Success";
    delete currentTempSensorConfig.pendingConfig;
}

```

```

var patch = { tempSensorConfig: currentTempSensorConfig };

patch.telemetryConfig.pendingConfig = null;

//complete the property update
twin.properties.reported.update(patch, function(err) {
if (err) {
    console.error('Error reporting properties: ' + err);
} else {
    console.log('Reported completed config change');
}
});
});
});

```

Metode Langsung

IoT Hub menyediakan mekanisme pesan perintah dan kontrol yang menggunakan pola simpan dan teruskan. Pesan dapat dimasukkan ke dalam antrian di backend melalui IoT Hub. Perangkat, jika terhubung dan mendengarkan, akan menerima pesan secara asinkron dan dapat bertindak berdasarkan pesan tersebut saat diterima. Pengirim pesan akan diberitahu melalui mekanisme panggilan balik bahwa pesan telah sampai atau gagal terkirim. Waktu aktif pesan perintah dan kontrol ini adalah dari 1 hingga 48 jam dan dapat dikonfigurasi oleh pengirim.

Metode langsung memberikan aplikasi cloud kemampuan untuk melakukan panggilan ke perangkat menggunakan pola permintaan/respons sinkron. Panggilan ini akan berhasil atau langsung gagal. Metode langsung memanfaatkan objek `ServiceClient` dalam paket `Microsoft.Azure.Devices NuGet` bersama dengan kelas `CloudToDeviceMethod`, yang memberikan nama fungsi yang akan dikenali perangkat dan batas waktu untuk interaksi permintaan/respons.

Misalnya, contoh kode ini akan mengirimkan permintaan reboot ke perangkat dengan batas waktu 30 detik:

```

using Microsoft.Azure.Devices;

ServiceClient _serviceClient;

_serviceClient = ServiceClient.CreateFromConnectionString(
    iothubconnstr);

var method = new CloudToDeviceMethod("reboot")
{
    ResponseTimeout = TimeSpan.FromSeconds(30)
};

_serviceClient.InvokeDeviceMethodAsync(myDeviceId, method);

```

Dalam contoh Node.js berikut, kita menggunakan objek `DeviceClient` untuk mendaftarkan fungsi `onReboot` agar dipanggil ketika metode langsung bernama `reboot` dipanggil. Fungsi `onReboot` akan segera mengirimkan respons, memperbarui perangkat kembar dengan status `reboot`, dan kemudian melakukan `reboot`. Perangkat memperbarui kembaran perangkat sehingga backend dapat menanyakan status operasi `reboot`.

```
'use strict';

var Client = require('azure-iot-device').Client;
...
// register the reboot handler
DeviceClient.onDeviceMethod('reboot', onReboot);

// called when the reboot direct method is called
var onReboot = function(request, response) {

  // Respond to the cloud app that invoked the direct method
  response.send(200, 'Reboot started', function(err) {
    if (!err) {
      console.error('An error occurred ' + err.toString());
    } else {
      console.log('Response to method sent successfully.');
```

```

    }
  });

  // Add your device's reboot API for physical restart.
  console.log('Rebooting!');
};

```

Pertanyaan

IoT Hub mendukung bahasa kueri mirip SQL untuk mengambil kumpulan perangkat kembar dan pekerjaan. Koleksi kembar perangkat IoT Hub disebut perangkat. Misalnya, untuk memilih semua perangkat kembar, Anda akan menggunakan kueri ini:

```
Select * from devices
```

Kueri berikutnya ini akan mengembalikan semua perangkat kembar untuk perangkat di gedung 23:

```
Select * from devices where tags.location.building= '23'
```

Dimungkinkan juga untuk melakukan kueri di seluruh tag dan properti yang dilaporkan dan diinginkan. Misalnya:

```
Select * from devices where tags,location.region= 'US' and
properties.reported.connectivity IN ["wi-fi", "wired"]
```

Anda juga dapat menentukan kueri yang mengembalikan status untuk memantau operasi yang berjalan lama. Misalnya:

```
select properties.reported.firmwareUpgrade.status AS status,
       count() AS numberOfDevices
from devices
group by properties.reported.firmwareUpgrade.status
```

Kueri ini mengembalikan JSON berikut:

```
[
  {
    "status": "Success",
    "numberOfDevices": 3
  },
  {
    "status": "Pending",
    "numberOfDevices": 2
  },
  {
    "status": "Error",

```

```

        "numberOfDevices": 1
    }
]

```

To execute a query, use the `RegistryManager` class:

```

var query = _registryManager.CreateQuery(
    "select * from devices", 100);

while (query.HasMoreResults)
{
    var page = await query.GetNextAsTwinAsync();
    foreach (var twin in page)
    {
        //perform configuration operation
    }
}

```

Argumen kedua pada `CreateQuery()` adalah ukuran halaman. Perulangan operasi dapat menggilir perangkat secara kembar halaman demi halaman.

Pekerjaan

Pekerjaan dapat digunakan untuk memperbarui properti yang diinginkan, memperbarui tag, dan memanggil metode langsung. Mereka dapat dijalankan pada satu perangkat atau pada kumpulan perangkat. IoT Hub melacak semua pekerjaan yang telah atau sedang dilakukan. Untuk menjadwalkan pekerjaan, IoT Hub Service SDK menyediakan `JobClient`. Anda dapat menggunakan objek ini bersamaan dengan definisi metode langsung untuk memulai pekerjaan:

```

CloudToDeviceMethod directMethod = new
    CloudToDeviceMethod("reboot", TimeSpan.FromSeconds(5),
    TimeSpan.FromSeconds(5));

JobResponse result = await
    jobClient.ScheduleDeviceMethodAsync(jobId,
    "deviceId='myDeviceId'", directMethod, DateTime.Now, 10);

```

Anda dapat menggunakan mekanisme kueri untuk menanyakan status pekerjaan. Pertimbangkan pertanyaan ini:

```

select * from devices.jobs
    where devices.jobs.deviceId = 'myDeviceId'
    and devices.jobs.jobType = 'firmwareUpgrade'
    and devices.jobs.status = 'completed'
    and devices.jobs.createdTimeUtc > '2016-09-01'

```

Ini akan mengembalikan semua pekerjaan pemutakhiran firmware yang berhasil diselesaikan untuk perangkat dengan ID unik myDeviceId.

Proses penghentian perangkat atau kumpulan perangkat melibatkan penghapusan dari lingkungan fisik serta pembaruan layanan perangkat lunak backend untuk menghapus atau mengarsipkan metadata perangkat. Implementasi referensi Bisnis Real-Time menyediakan fungsi penghapusan yang menghapus metadata dari IoT Hub serta dari koleksi DocumentDB sebagai fitur utilitas Bootstrap. Ia menggunakan API Perangkat untuk melakukan operasi di semua perangkat yang terdaftar.

Penyiapan Latihan

Latihan ini memanfaatkan layanan Azure dan layanan mikro yang disebarkan di Bab 2. Dalam latihan berikut, Anda mengonfigurasi dasbor Manajemen Perangkat dan simulator perangkat yang mengimplementasikan pola manajemen perangkat yang dibahas dalam bab ini.

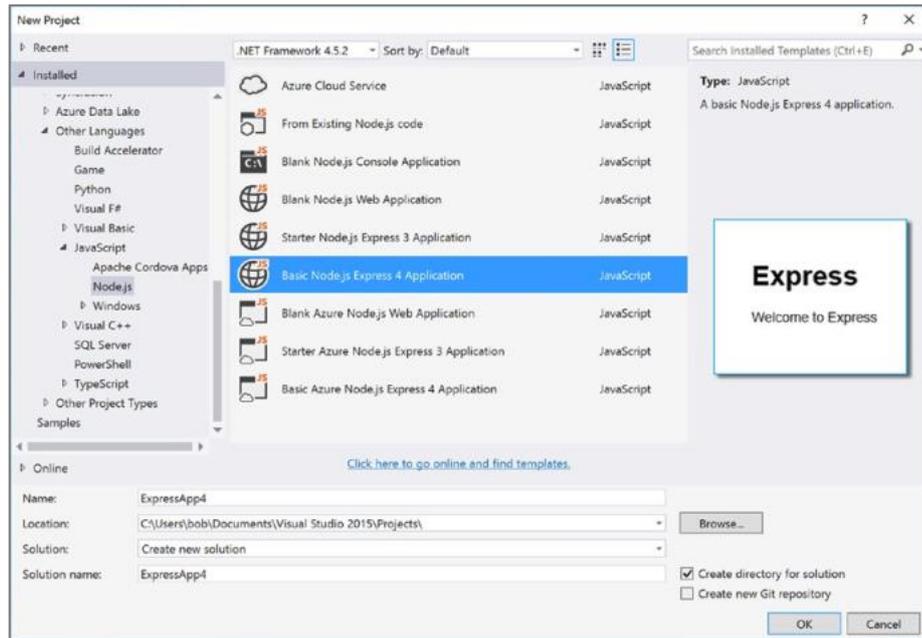
Dasbor dan perangkat dibuat menggunakan Node.js. Untuk menjalankan dan menguji fitur manajemen perangkat, Anda menggunakan Visual Studio, Visual Studio Code, dan utilitas yang disebut Device Explorer, yang merupakan bagian dari Azure IoT SDK.

Untuk melakukan operasi ini, lingkungan pengembangan Anda memerlukan tambahan berikut:

1. Unduh dan instal Node.js v6.9.x:
<https://nodejs.org/en/>
2. Menggunakan jendela shell perintah, navigasikan ke folder devices/device-node dan jalankan perintah ini untuk menginstal paket node untuk solusi perangkat:
C:> npm install azure-iot-device- - save
C:> npm install azure-iot-device-mqtt - -save
3. Unduh dan instal alat Visual Studio Node.js:
<https://visualstudio.com/vs/node-js/>
4. Unduh dan instal kode Visual Studio:
<https://code.visualstudio.com>

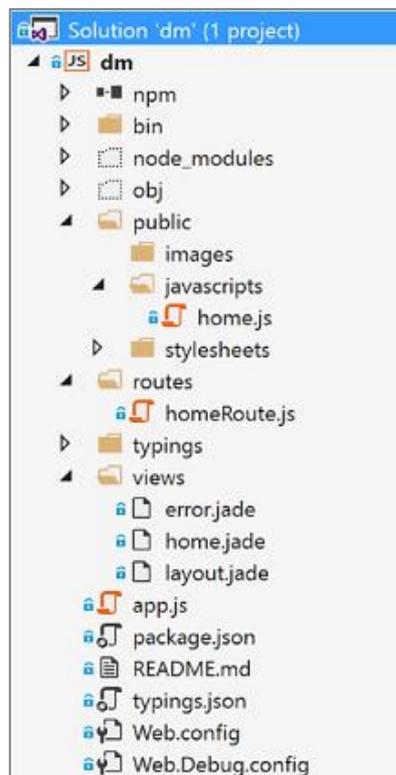
3.4 DASBOR MANAJEMEN PERANGKAT

Buka solusi `dm.sln` di folder `dashboards/dm`. Proyek Visual Studio ini dibangun menggunakan template Visual Studio Node.js Express 4, seperti yang ditunjukkan pada Gambar 3-6, dan memanfaatkan Jade, mesin template HTML berkinerja tinggi, dan Node untuk menentukan server HTTP yang menyajikan HTML- UI berbasis CSS-JavaScript.



Gambar 3.6. Templat proyek Visual Studio Node.js

Kode tata letak Jade ditemukan di file `views/layout.jade` dan `views/home.jade`. JavaScript yang dijalankan pada klien ditemukan di file `public/javascripts/home.js` (lihat Gambar 3-7).



Gambar 3.7. Penjelajah solusi DM

Untuk mengonfigurasi aplikasi agar berfungsi di lingkungan Anda, buka file `home.js` dan perbarui dua variabel di bagian atas file dengan kunci langganan dan nama host API Management dari lingkungan Anda.

```
Var devKey = '[your-dev-key]';
Var apimHost = 'https://[your-apim-host].azure-api.net';
```

Jalankan aplikasi. Fungsi JavaScript pertama yang dipanggil adalah `GetAllCompanies()`, dan ditemukan di file `home.js`. Kode secara dinamis menghasilkan elemen HTML `SELECT` dengan memanggil API Registri dan meminta semua profil tipe organization (yaitu, tipe = 1).

```
function GetAllCompanies()
{
    var selectContent =
        '<option value="">Select a company</option>';

    $.getJSON(
        apimHost + '/dev/v1/registry/profiles/type/1?subscription-key=' +
        devKey,
        function ( data ) {

            $.each( data.list, function ()
            {
                selectContent +=
                    '<option value="" +
                    this.id + "">' +
                    this.companyname + '</option>';
            });

            $( '#companyList select' ).html( selectContent );
            $( document.getElementById(
                'companyList' ).selectedIndex = 0 );
        });
}
```

Pilih perusahaan WigiTech dan Anda akan disajikan perangkat yang didaftarkan ke karyawan WigiTech (lihat Gambar 3-8). Perangkat dalam skenario kami dipasang ke tubuh karyawan dan memberikan pembacaan biometrik secara terus menerus. Pengaturan irama detak jantung dan pesan telemetri perangkat disajikan sebagai bidang yang dapat diedit, dan terdapat tombol perintah untuk menjalankan perintah Reboot, Perbarui Firmware, dan Konfigurasi Irama.

Business in Real-Time

Manage Your Connected Team

WigTech

Serial Number	Model	Heartbeat	Telemetry	Reboot	Upgrade	Configure
fb57e0a4-d3f7-443c-af10-310c7f09ccba	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence
21d5d6ca-9632-4a79-9caa-632c0cc78c32	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence
0f6a3116-63b9-4ac2-9961-e106846b76f7	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence
71abbd52-986f-44cb-a8dc-d6d07fee3322	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence
f0dd5745-11a7-48aa-9262-d7e220537e7c	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence
751c6d0c-a03f-4112-b7e7-77f7fe6f9eb3	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence
4b9f2d4b-e012-43cb-8262-66a53567f1be	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence
4a061209-84ab-49ad-8f00-f3597b932590	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence
4abf70b7-e04d-4ccb-a9e6-8053916a6672	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence
54cc5c03-eb06-495d-ba2b-405bc79e5204	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence
c5ae7c45-d24c-4239-b4d8-a6a91f741043	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence
47ae5112-f50b-4a36-aa9c-9caea0808cdd	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence
caa19850-23f2-494a-a02e-684c7adb884c	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence
5a3600ca-ad6b-470e-9f2f-cdab1c1608d8	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence
9b16c815-230e-4d8d-93e7-a41e60946008	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence
1920dcc7-a92b-4e40-b124-a88bce85fa7a	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence
1d41e4b-2600-4d83-98b7-dd734578a82f	WT-SIM-001	30000	10000	Reboot Device	Update Firmware	Configure Cadence

Gambar 3-8. Dasbor manajemen perangkat

Operasi Reboot dan Update Firmware menggunakan mekanisme metode langsung untuk berkomunikasi dengan perangkat. Konfigurasi Irama menggunakan properti yang diinginkan untuk meminta perubahan konfigurasi pada perangkat. Data yang digunakan untuk membuat tabel ini berasal dari manifes untuk setiap perangkat yang terdaftar. Saat Anda memilih perusahaan dari drop-down, fungsi `GetDevices()` dipanggil. Fungsi ini memanggil API “Dapatkan Manifes berdasarkan ID Pelanggan”, yang mengembalikan daftar manifes yang terkait dengan perusahaan yang diidentifikasi oleh `customerId`.

```
function GetDevices( select )
{
    var uri = apimHost +
        '/dev/v1/device/manifests/customer/' +
        select.value + '?subscription-key=' + devKey;

    $.getJSON( uri, function ( data ) {
        manifests = data.List;
    ...

```

Sisa kode dalam fungsi ini melanjutkan pembuatan tabel perangkat secara dinamis menggunakan kumpulan manifes. Saat Anda mengklik tombol Reboot, Update Firmware, atau Configure Cadence, fungsi JavaScript terkait akan dipanggil. Fungsi-fungsi ini memanggil titik akhir API Perangkat untuk melakukan tugas manajemen dan konfigurasi perangkat.

Mari kita periksa fungsi Reboot. Fungsi tersebut meneruskan nomor seri perangkat yang akan di-boot ulang. Itu membangun fragmen JSON untuk objek JSON yang disebut `twinPropertyRequest`. Kelas ini diteruskan ke metode Device API dan berisi nomor seri, nama metode langsung, dan daftar parameter apa pun. Metode `reboot direct` tidak mengambil parameter apa pun, sehingga fragmen JSON hanya diinisialisasi dengan nomor seri dan nama metode langsung, sebagai berikut:

```
function Reboot( serialNumber )
{
    var command = {
        DeviceId: serialNumber,
        Name: 'reboot',
        Properties: []
    };
}
```

Selanjutnya, fungsi tersebut memanggil titik akhir Device API untuk memanggil metode langsung yang meneruskannya twinPropertyRequest JSON.

```
var twinPropertyRequest = JSON.stringify(command);

var uri = apimHost +
    '/dev/v1/device/twin/properties/direct';

$.ajax( {
    url: uri,
    type: "PUT",
    data: twinPropertyRequest,

    headers: {
        "Ocp-Apim-Trace": "true",
        "Ocp-Apim-Subscription-Key": devKey,
        "Content-Type": "application/json"
    },
    success: function ()
    {
        alert( "success" );
    },
    error: function ( xhr, status, error )
    {
        var err = eval( "(" + xhr.responseText + ")" );
        alert( status + ": " + err.Message );
    }
});
}
```

Device API mengakses TwinPropertyRequest dari isi permintaan dan menyiapkan panggilan ke metode langsung menggunakan objek CloudToDeviceMethod dan ServiceClient.

```
public void UpdateTwinPropertiesDirect(
    TwinPropertyRequest twinPropertyRequest)
{
    // Update Twin Properties using a Direct Method
    var method = new CloudToDeviceMethod(twinPropertyRequest.Name)
}
```

```

    { ResponseTimeout = TimeSpan.FromSeconds(30) };
...
_serviceClient.InvokeDeviceMethodAsync(
    twinPropertyRequest.DeviceId, method));

```

Fungsi `UpdateFirmware()` juga menggunakan teknik metode langsung, namun selain perintah pembaruan firmware, fungsi ini meneruskan parameter yaitu URI ke file gambar firmware.

```

function UpdateFirmware( serialNumber )
{
    var command = {
        DeviceId: serialNumber,
        Name: 'firmwareUpdate',
        Properties: [
            {
                Key: 'fwPackageUri',
                Val: 'https://[uri-to-firmware]'
            }
        ]
    };
    var twinPropertyRequest = JSON.stringify(command);
    var uri = apimHost + '/dev/v1/device/twin/properties/direct';

    $.ajax( {
        url: uri,
        type: "PUT",
        data: twinPropertyRequest,
        headers: {
            "Ocp-Apim-Trace": "true",
            "Ocp-Apim-Subscription-Key": devKey,
            "Content-Type": "application/json"
        },
        success: function ()
        {
            alert( "success" );
        },
        error: function ( xhr, status, error )
        {
            var err = eval( "(" + xhr.responseText + ")" );
            alert( status + ": " + err.Message );
        }
    });
}

```

Biasanya perangkat mengirimkan pesan detak jantung untuk memberi sinyal bahwa perangkat masih berfungsi dan dapat berkomunikasi. Irama detak jantung menentukan

berapa lama perangkat menunggu antara pengiriman pesan detak jantung. Demikian pula, irama telemetri menentukan berapa lama perangkat menunggu antara pengiriman pesan telemetri.

Pengaturan default untuk nilai-nilai ini, 30 detik untuk detak jantung dan 10 detik untuk telemetri, disimpan dalam manifes perangkat dan digunakan untuk menginisialisasi perangkat saat startup. Melalui aplikasi Device Management, kita dapat memperbarui pengaturan ini saat perangkat berjalan dengan mengirimkan permintaan perubahan Device Twin Desired Property.

Fungsi `Configure()` memanfaatkan teknik Properti yang Diinginkan untuk meminta perubahan konfigurasi pada pengaturan detak jantung dan irama telemetri. Ia menggunakan struktur data `TwinPropertyRequest` yang sama untuk meneruskan nilai baru untuk pengaturan detak jantung dan irama telemetri.

```
function Configure( serialNumber, inputId ) {

    var hbInputId = 'hb' + String(inputId);
    var tlInputId = 'tl' + String(inputId);

    var heartbeat;
    var telemetry;
    $(heartbeat = document.getElementById( hbInputId ).value );
    $(telemetry = document.getElementById( tlInputId ).value);

    var request = {
        DeviceId: serialNumber,
        Name: 'cadenceConfig',
        Properties: [
            {
                Key: 'heartbeat',
                Val: heartbeat
            },
            {
                Key: 'telemetry',
                Val: telemetry
            }
        ]
    };

    var twinPropertyRequest = JSON.stringify( request );
    var uri = apimHost + '/dev/v1/device/twin/properties';

    $.ajax( {
        url: uri,
        type: "PUT",
        data: twinPropertyRequest,
        headers: {
```

```

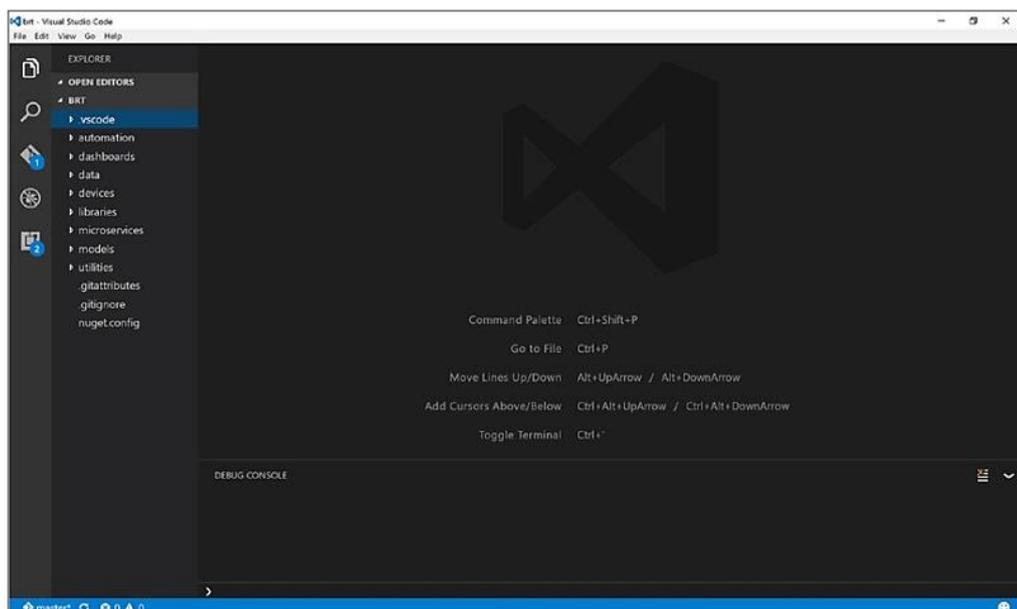
    "Ocp-Apim-Trace": "true",
    "Ocp-Apim-Subscription-Key": devKey,
    "Content-Type": "application/json"
  },
  success: function ()
  {
    alert( "success" );
  },
  error: function ( xhr, status, error )
  {
    var err = eval( "(" + xhr.responseText + ")" );
    alert( status + ": " + err.Message );
  }
});
}

```

Pada latihan berikutnya, Anda menentukan simulator perangkat untuk menerima panggilan metode langsung dan kejadian properti yang diinginkan. Salin nomor seri dari halaman web untuk perangkat pertama dalam daftar dan sisihkan karena Anda memerlukannya untuk memperbarui simulator perangkat yang akan menerima perintah yang datang dari konsol.

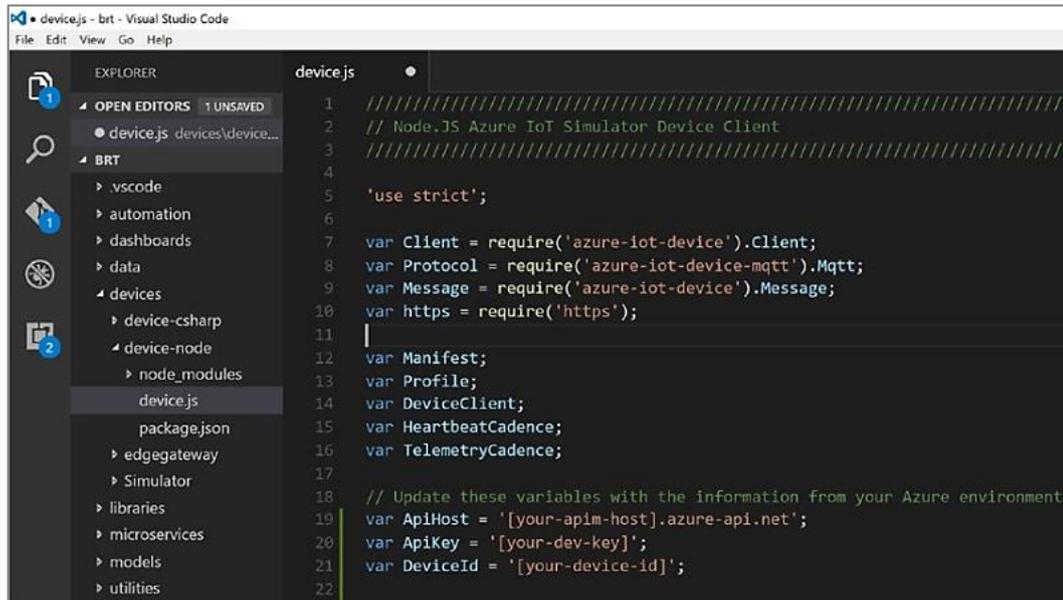
Simulator Perangkat

Untuk sepenuhnya menguji layanan perangkat lunak manajemen perangkat, Anda memerlukan perangkat. Dalam latihan ini, Anda memodifikasi kode untuk perangkat dan, menggunakan dasbor Manajemen Perangkat, menguji perintah Reboot, Peningkatan Firmware, dan Konfigurasi Irama. Mulai Visual Studio Code dan, dengan menggunakan fitur folder terbuka, buka folder akar dari repositori kode implementasi referensi (lihat Gambar 3-9).



Gambar 3-9. Kode Visual Studio

Proyek perangkat ada di `devices/device-node`. Arahkan ke folder itu dan buka file `device.js`. Perbarui tiga variabel di bagian atas file dengan nama host API Management Anda, kunci pengembang API Management Anda, dan ID Perangkat (nomor seri) yang Anda salin dari dasbor Manajemen Perangkat pada latihan sebelumnya (lihat Gambar 3-10).



```

1 ////////////////////////////////////////////////////////////////////
2 // Node.JS Azure IoT Simulator Device Client
3 ////////////////////////////////////////////////////////////////////
4
5 'use strict';
6
7 var Client = require('azure-iot-device').Client;
8 var Protocol = require('azure-iot-device-mqtt').Mqtt;
9 var Message = require('azure-iot-device').Message;
10 var https = require('https');
11
12 var Manifest;
13 var Profile;
14 var DeviceClient;
15 var HeartbeatCadence;
16 var TelemetryCadence;
17
18 // Update these variables with the information from your Azure environment
19 var ApiHost = '[your-apim-host].azure-api.net';
20 var ApiKey = '[your-dev-key]';
21 var DeviceId = '[your-device-id]';
22

```

Gambar 3-10. Perbarui variabel ApiHost, ApiKey, dan DeviceId

File `device.js` disusun menjadi beberapa bagian berikut:

- *Kelas Pesan*: Menentukan pesan detak jantung dan telemetri
- *Fungsi Pembantu*: Rutinitas pembantu
- *Reboot Handler*: Kode dipanggil ketika perintah reboot dikeluarkan ke perangkat
- *Pengendali Pembaruan Firmware*: Kode dipanggil ketika perintah pembaruan firmware dikeluarkan ke perangkat
- *Pengendali Properti yang Diinginkan*: Kode dipanggil ketika peristiwa perubahan properti yang diinginkan diterima
- *Utama*: Fungsi pengontrol untuk setiap pengoperasian perangkat utama

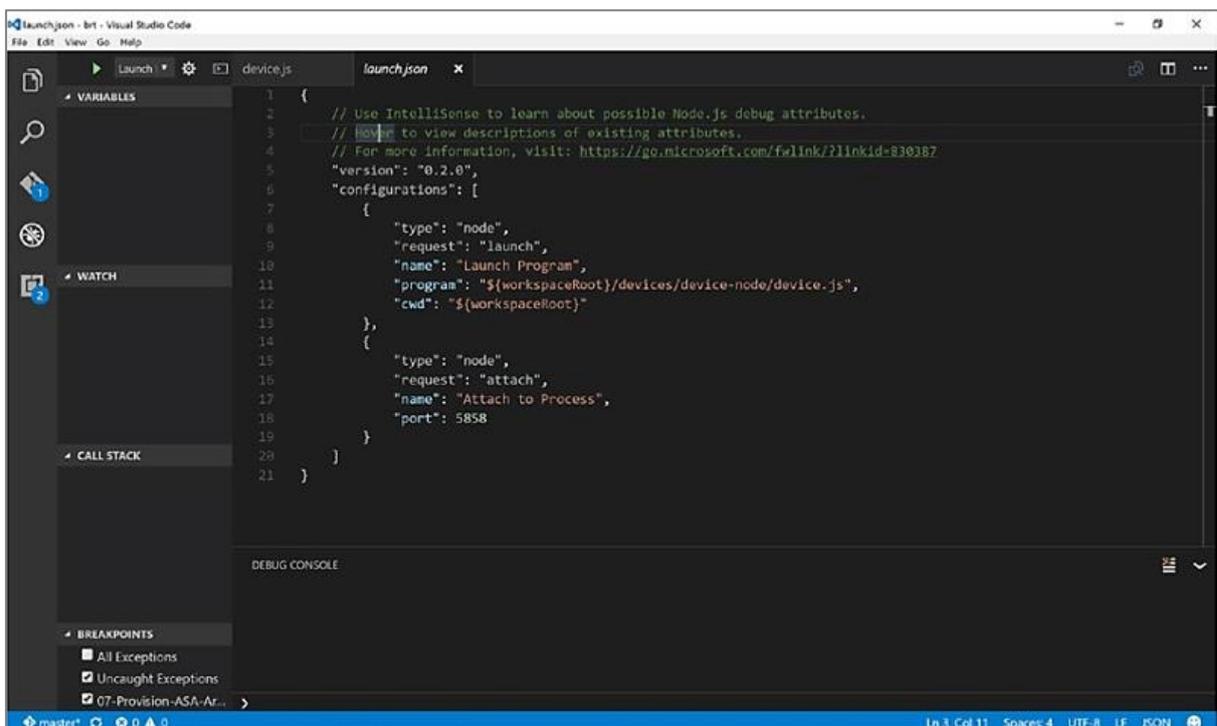
Pertama mari kita periksa fungsi `main()`. Rutinitas ini memberi Anda kendali atas setiap pengoperasian utama perangkat. Operasi ini meliputi:

- `getDeviceManifest()`: Memanggil API Perangkat menggunakan ID unik perangkat untuk mengambil manifes
- `getUserProfile()`: Memanggil Registry API untuk mengambil profil pengguna yang terkait dengan perangkat ini
- `connectIoTHub()`: Menyambungkan ke IoT Hub
- `setRebootHandler()`: Mengatur event handler metode langsung reboot
- `setFirmwareUpdateHandler()`: Menyiapkan pengendali peristiwa metode langsung pembaruan firmware

- `setDesiredPropetyChangeHandler()` : Mengatur properti kembar yang diinginkan untuk mengubah event handler
- `sendHeartbeat()` : Mengirim pesan detak jantung
- `sendTelemetry()` : Mengirim pesan telemetri

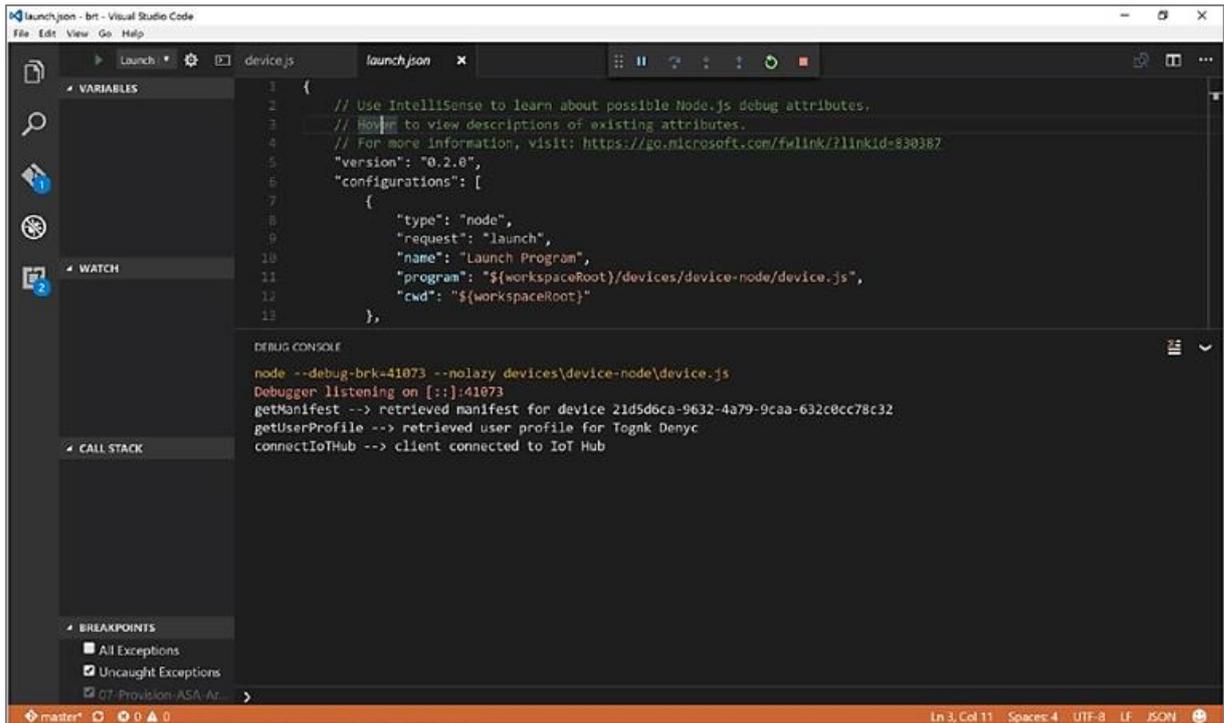
Sebagian besar panggilan dikomentari sehingga Anda dapat menghubungkannya secara online satu per satu. Sebelum Anda melakukannya, mari siapkan Visual Studio Code untuk debugging.

Klik ikon debug di selokan sebelah kiri, lalu klik ikon roda gigi di bilah menu. Pastikan pengaturan program di file `launch.json` mengarah ke file `device.js` (lihat Gambar 3-11).



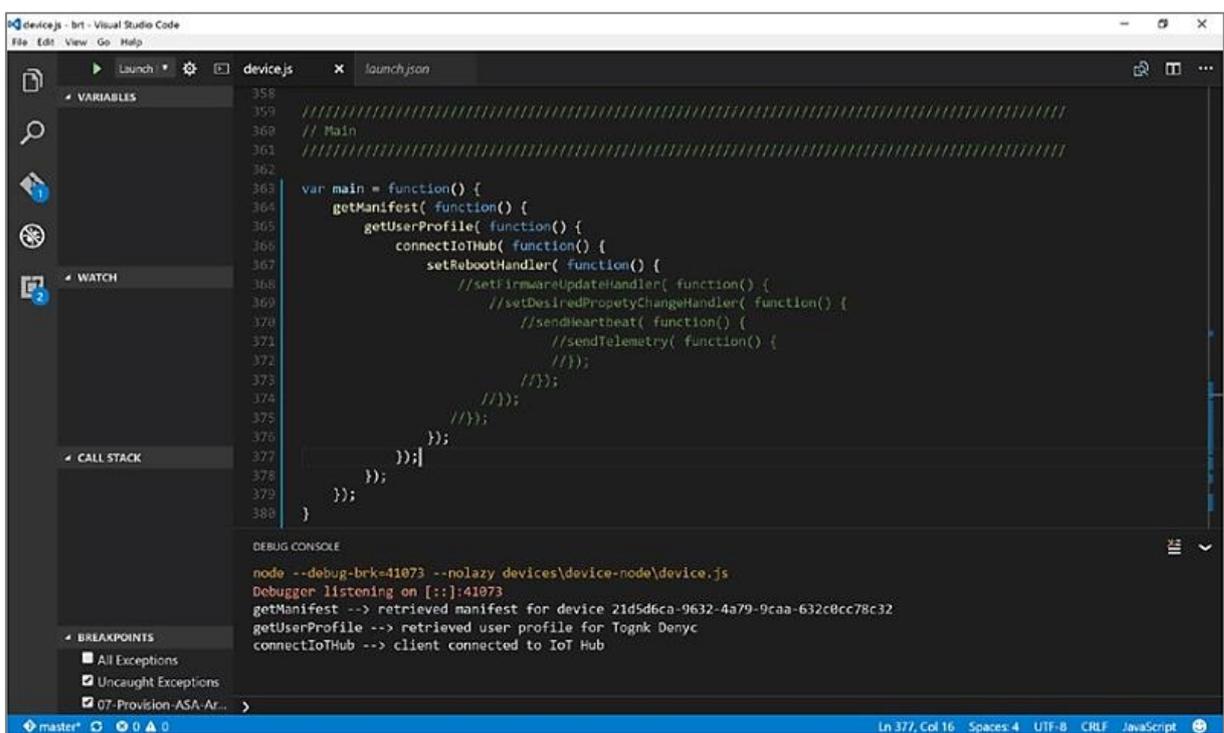
Gambar 3-11. Siapkan file Launch.Json untuk debugging

Klik tombol jalankan di bilah menu. Perhatikan kontrol debugger di bagian atas IDE. Perangkat akan mengambil manifestnya, mencari profil pengguna untuk pengguna yang ditetapkan pada perangkat ini, dan terhubung ke IoT Hub (lihat Gambar 3-12). Kode ini menggunakan fungsi `console.log()` dari Node untuk mengeluarkan informasi pelacakan ke jendela konsol debug.



Gambar 3-12. Luncurkan debugger

Hentikan debugger. Ubah fungsi main () sehingga fungsi setRebootHandler () akan dipanggil saat berikutnya Anda menjalankan aplikasi (lihat Gambar 3-13).

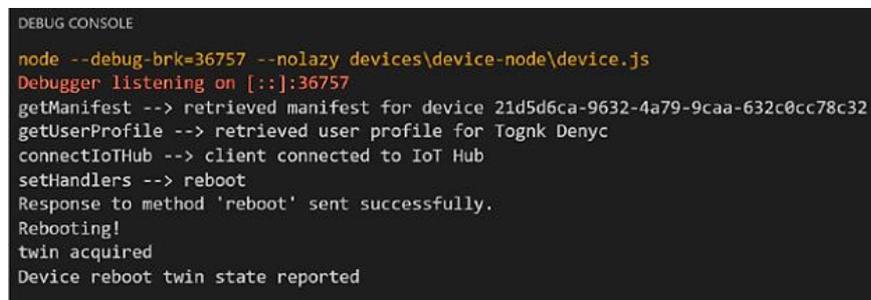


Gambar 3-13. Aktifkan panggilan setRebootHandler ()

Karena kita akan mengaktifkan event handler reboot, Anda juga ingin menjalankan aplikasi konsol Manajemen Perangkat sehingga Anda dapat menguji metode reboot langsung. Fungsi `setRebootHandler()` menggunakan objek `DeviceClient` untuk menyiapkan pengendali kejadian metode langsung. Ketika metode langsung yang disebut `reboot` dipanggil, fungsi panggilan balik bernama `onReboot()` akan dipanggil.

```
var setRebootHandler = function (callback)
{
    callback();
    console.log('setHandlers --> reboot');
    DeviceClient.onDeviceMethod('reboot', onReboot);
}
```

Seperti yang dibahas sebelumnya dalam bab ini, fungsi `onReboot()` akan mengirimkan respons langsung ke pemanggil sehingga aplikasi cloud mengetahui apakah panggilan tersebut berhasil. Selanjutnya memperbarui perangkat kembar dengan status reboot dan kemudian memulai proses reboot. Jalankan aplikasi di debugger. Output konsol debug akan mendokumentasikan bahwa `setRebootHandler()` aktif. Menggunakan dasbor Manajemen Perangkat, klik tombol Reboot untuk perangkat. Konsol debug akan mendokumentasikan bahwa metode langsung berhasil dipanggil (lihat Gambar 3-14).



```
DEBUG CONSOLE
node --debug-brk=36757 --nolazy devices\device-node\device.js
Debugger listening on [::]:36757
getManifest --> retrieved manifest for device 21d5d6ca-9632-4a79-9caa-632c0cc78c32
getUserProfile --> retrieved user profile for Togkn Denyc
connectIoTHub --> client connected to IoT Hub
setHandlers --> reboot
Response to method 'reboot' sent successfully.
Rebooting!
twin acquired
Device reboot twin state reported
```

Gambar 3-14. Reboot metode langsung berhasil

Hentikan debugger.

Batalkan komentar pada `setFirmwareUpdateHandler()` dan jalankan kembali aplikasi perangkat. Menggunakan dasbor Manajemen Perangkat, klik tombol Perbarui Firmware. Periksa output konsol debug di Visual Studio Code untuk melihat bahwa kejadian metode langsung telah ditangani. Perhatikan bagaimana kembaran perangkat diperbarui seiring setiap langkah proses pembaruan firmware diterapkan (lihat Gambar 3-15).

```

DEBUG CONSOLE
node --debug-brk=35589 --nolazy devices\device-node\device.js
Debugger listening on [::]:35589
getManifest --> retrieved manifest for device 21d5d6ca-9632-4a79-9caa-632c0cc78c32
getUserProfile --> retrieved user profile for Tognk Denyc
connectIoTHub --> client connected to IoT Hub
setHandlers --> firmware upgrade
setHandlers --> reboot
onFirmwareUpdate --> Response to method 'firmwareUpdate' sent successfully.
onFirmwareUpdate --> package uri == https://[uri-to-firmware]
onFirmwareUpdate --> device twin acquired.
twin state reported --> {"iothubDM":{"firmwareUpdate":{"fwPackageUri":"https://[uri-to-firmware]","status":"waiting","error":null,"startedWaitingTime":"2016-11-27T00:29:00.637Z"}}}
twin state reported --> {"iothubDM":{"firmwareUpdate":{"status":"downloading"}}}
Downloading image from https://[uri-to-firmware]
twin state reported --> {"iothubDM":{"firmwareUpdate":{"status":"downloadComplete","downloadCompleteTime":"2016-11-27T00:29:04.639Z"}}}

twin state reported --> {"iothubDM":{"firmwareUpdate":{"status":"applying","startedApplyingImage":"2016-11-27T00:29:12.642Z"}}}

twin state reported --> {"iothubDM":{"firmwareUpdate":{"status":"applyComplete","lastFirmwareUpdate":"2016-11-27T00:29:12.642Z"}}}
>|

```

Gambar 3-15. Pembaruan firmware metode langsung berhasil

Dalam pengujian akhir ini, batalkan komentar pada sisa fungsi `main()` sehingga pengendali peristiwa properti yang diinginkan diaktifkan dan perangkat akan mulai mengirimkan pesan detak jantung dan telemetri. Jalankan aplikasi perangkat dan, dengan menggunakan dasbor Manajemen Perangkat, ubah pengaturan detak jantung dan irama telemetri masing-masing menjadi 15000 dan 5000 (lihat Gambar 3-16). Konsol debug akan mendokumentasikan:

- Pesan dikirim menggunakan pengaturan irama asli
- Properti yang diinginkan diterima
- Properti kembaran perangkat yang dilaporkan diperbarui menggunakan pengaturan dari properti yang diinginkan
- Perangkat mulai mengirimkan pesan detak jantung dan telemetri menggunakan nilai irama baru

```

DEBUG CONSOLE
sendTelemetry --> 10000
Sending message: {"Id":"31fa2d13-92cf-42b3-2a07-1d235b496212","Longitude":-71.216035,"Latitude":42.221322,"MessageType":4,"UserId":"058d7760-e868-434e-aa1d-bbc438d5ebb6","Age":72,"Height":170,"Weight":270,"HeartRateBPM":0,"HeartRateRedZone":0,"HeartRateVariability":0,"BreathingRate":0,"Temperature":0,"Steps":0,"Velocity":0,"Altitude":0,"Cadence":0,"Speed":0,"HIB":0,"Status":1}
send status: MessageEnqueued
sendTelemetry --> 10000
Sending message: {"Id":"9a4016d7-309f-4a88-f5d2-f5ad763b846f","Longitude":-71.216035,"Latitude":42.221322,"MessageType":4,"UserId":"058d7760-e868-434e-aa1d-bbc438d5ebb6","Age":72,"Height":170,"Weight":270,"HeartRateBPM":0,"HeartRateRedZone":0,"HeartRateVariability":0,"BreathingRate":0,"Temperature":0,"Steps":0,"Velocity":0,"Altitude":0,"Cadence":0,"Speed":0,"HIB":0,"Status":1}
send status: MessageEnqueued
sendHeartbeat --> 30000
Sending message: {"Id":"46d9a114-ba83-4cb3-18fb-6fb5ef913335","Longitude":-71.216035,"Latitude":42.221322,"Timestamp":"2016-11-27T00:38:30.253Z","MessageType":1,"Ack":{"Node Device Is Active"}}
sendTelemetry --> 10000
Sending message: {"Id":"d464b9e7-4d16-48eb-5b40-33e2a60edda1","Longitude":-71.216035,"Latitude":42.221322,"MessageType":4,"UserId":"058d7760-e868-434e-aa1d-bbc438d5ebb6","Age":72,"Height":170,"Weight":270,"HeartRateBPM":0,"HeartRateRedZone":0,"HeartRateVariability":0,"BreathingRate":0,"Temperature":0,"Steps":0,"Velocity":0,"Altitude":0,"Cadence":0,"Speed":0,"HIB":0,"Status":1}
send status: MessageEnqueued
send status: MessageEnqueued
twin.properties.desired --> property change: {"cadenceConfig":{"heartbeat":"15000","telemetry":"5000"},"$version":20}
InitConfigChange --> starting
InitConfigChange --> pending config change: {"cadenceConfig":{"heartbeat":"30000","telemetry":"10000","status":"Pending","pendingConfig":{"heartbeat":"15000","telemetry":"5000"}}}
completeConfigChange --> starting
completeConfigChange --> completed config change: {"cadenceConfig":{"heartbeat":"15000","telemetry":"5000","status":"Success"}}
sendTelemetry --> 5000
Sending message: {"Id":"0d73929b-31be-4c8b-110b-044569f28dcd","Longitude":-71.216035,"Latitude":42.221322,"MessageType":4,"UserId":"058d7760-e868-434e-aa1d-bbc438d5ebb6","Age":72,"Height":170,"Weight":270,"HeartRateBPM":0,"HeartRateRedZone":0,"HeartRateVariability":0,"BreathingRate":0,"Temperature":0,"Steps":0,"Velocity":0,"Altitude":0,"Cadence":0,"Speed":0,"HIB":0,"Status":1}
send status: MessageEnqueued
sendTelemetry --> 5000
Sending message: {"Id":"c1b50077-6c4a-4118-00dd-f98548c63b62","Longitude":-71.216035,"Latitude":42.221322,"MessageType":4,"UserId":"058d7760-e868-434e-aa1d-bbc438d5ebb6","Age":72,"Height":170,"Weight":270,"HeartRateBPM":0,"HeartRateRedZone":0,"HeartRateVariability":0,"BreathingRate":0,"Temperature":0,"Steps":0,"Velocity":0,"Altitude":0,"Cadence":0,"Speed":0,"HIB":0,"Status":1}
send status: MessageEnqueued
>

```

Gambar 3-16. Properti irama diterapkan

3.5 RINGKASAN

Manajemen perangkat adalah komponen kunci dari IoT end-to-end dan solusi analisis tingkat lanjut. Manajemen perangkat memberikan dukungan siklus hidup yang lengkap untuk ekosistem perangkat Anda melalui serangkaian layanan perangkat lunak yang bekerja sama untuk perencanaan, penyediaan, konfigurasi, pemantauan, dan pada akhirnya menghentikan perangkat yang digunakan.

BAB 4

SENSOR, PERANGKAT, DAN GERBANG

Bab 2 dan 3 berfokus pada proses inti yang harus diterapkan saat memulai perjalanan transformasi digital IoT dan analitik tingkat lanjut, yaitu DevOps dan Manajemen Perangkat. Bab ini membahas dunia sensor, perangkat, dan gateway. Meskipun kami tidak dapat mencakup semua kemungkinan permutasi perangkat keras dan perangkat lunak, kami akan membahas beberapa skenario umum yang kami temui dan bagaimana keduanya berhubungan dan bekerja sama untuk menciptakan jaringan yang konsisten, aman, dan andal.

4.1 SENSOR

Tujuan dari sensor adalah untuk mengukur. Sensor mengambil pengukuran parameter fisik dan mengubahnya menjadi nilai yang dapat dibaca secara elektrik menggunakan sirkuit analog atau digital. Bentuk fisik, bahan konstruksi, dan elektronik bergantung pada aplikasinya. Ada banyak kelas sensor yang dibedakan berdasarkan akurasi, ketahanan lingkungan, jangkauan, detail resolusi, kemampuan kalibrasi, dan biaya (lihat Gambar 4-1).



Sensor Suhu dan Kelembapan

Sensor Tekanan Negatif (Vakum)

Sensor Gerak PIR

Gambar 4-1. Berbagai jenis sensor

Terdapat industri produsen sensor yang kuat yang menyediakan solusi perangkat keras sensor untuk mengumpulkan kualitas terukur, termasuk namun tidak terbatas pada:

- Posisi, kedekatan, dimensi, jarak, kemiringan, dan gerak
- Kualitas udara, polusi udara, kadar karbon dioksida, hidrokarbon, hidrogen, metana, dan oksigen
- Arus listrik, arus eddy, medan listrik, medan magnet, dan tegangan
- Kondisi cuaca termasuk suhu, kelembapan, titik embun, aliran panas, dan asap
- Aliran darah manusia dan hewan, kadar glukosa darah, tekanan darah, suhu tubuh, dan detak jantung
- Parameter akustik seperti kenyaringan, kebisingan, resonansi, ultrasound, dan arah

- Pengukuran fluida termasuk pengukur tekanan, massa, aliran volume, curah, dan ketinggian
- penglihatan dan identifikasi termasuk sensor gambar, deteksi kode, dan deteksi objek
- Optik dan luminositas termasuk radiasi, cahaya, kekeruhan, UV, dan cahaya tampak

Daftar ini bisa terus bertambah. Poin utamanya adalah bahwa dunia sensor itu kaya dan menakjubkan, dan biaya sensor-sensor ini serta konverter analog ke digital dan perangkat keras mikroprosesor untuk terhubung dan berkomunikasi telah menjadikan IoT sebagai komoditas yang ada di mana-mana. Kita bisa mengukur apa saja dan mengubahnya menjadi bilangan floating-point.

Jika Anda menggunakan *Single-Board Computer* (SBC) seperti Raspberry Pi, Anda dapat menyambungkan sensor analog menggunakan pelindung adaptor *Analog-to-Digital* (ADC). Adaptor ini membaca tegangan yang berasal dari sensor analog dan mengubahnya menjadi nilai digital. Anda juga dapat berkomunikasi dengan sensor melalui antarmuka UART, I2C, dan GPIO.

Bukan hal yang aneh untuk menghubungkan SBC Anda ke perangkat perantara seperti *Programmable Logic Controller* (PLC). Sensor dapat dipasang langsung ke PLC, atau PLC dapat mengumpulkan pembacaan dari perangkat keras yang mendukung sensor yang terpasang ke PLC. PLC dan komputer papan tunggal berkomunikasi menggunakan antarmuka serial atau Ethernet. Perangkat lunak SBC menggunakan perpustakaan seperti Modbus atau Ethernet/IP untuk berkomunikasi dengan PLC.

Catatan Modbus adalah protokol komunikasi serial yang awalnya diterbitkan oleh Schneider Electric pada tahun 1979 untuk digunakan dengan produk PLC-nya. Karena kemudahan penggunaan dan popularitasnya, Modbus telah menjadi standar de facto protokol untuk berkomunikasi dengan PLC. Ethernet/IP adalah protokol jaringan industri yang mengadaptasi Common Industrial Protocol (CIP) ke Ethernet standar, dan CIP adalah protokol populer lainnya untuk aplikasi otomasi industri.

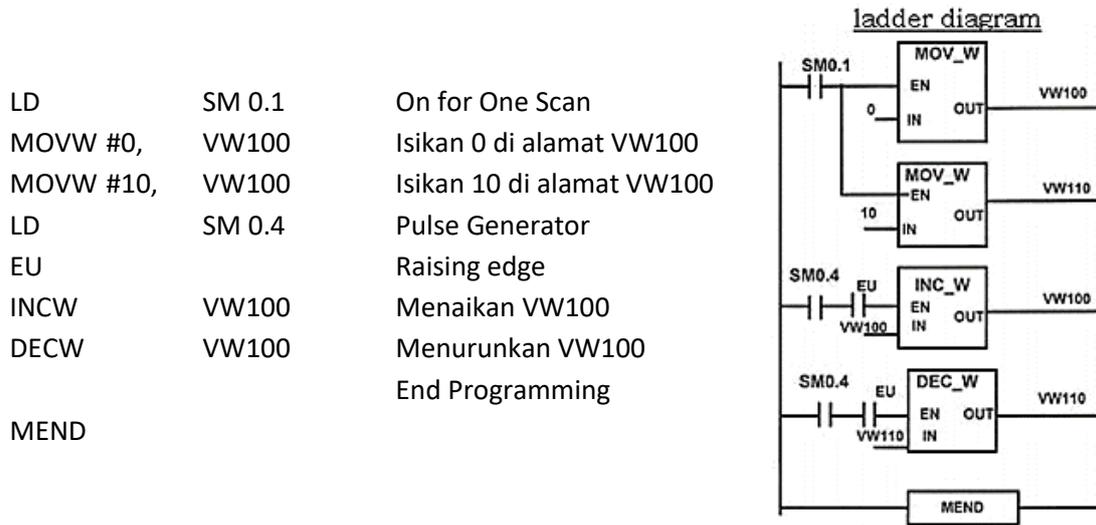
Pengontrol Logika yang Dapat Diprogram

Pengontrol Logika yang Dapat Diprogram (PLC) memainkan peran penting dalam IoT industri. PLC adalah komputer tangguh yang digunakan untuk menyediakan antarmuka yang dapat diprogram ke mesin industri, kamera, dan sensor. PLC dapat digunakan, misalnya, untuk mengontrol katup pada tangki penyimpanan bahan kimia; itu memonitor level cairan menggunakan sensor dan membuka katup untuk mengeluarkan cairan ketika sensor dipicu.

PLC menggunakan bahasa pemrograman khusus untuk menyesuaikan cara mereka mengontrol mesin dan sensor yang terpasang. Ada lima pendekatan pemrograman PLC yang digunakan saat ini:

- Diagram blok fungsi
- Diagram logika tangga
- Teks terstruktur
- Daftar instruksi
- Bagan fungsi berurutan

Berikut adalah contoh program PLC yang menambah pencacah setiap kali generator pulsa menyala (lihat Gambar 4-2).



Gambar 4.2. Contoh program PLC dari plcmanual.com

PLC sering digunakan untuk menghubungkan peralatan industri ringan hingga berat ke sistem Supervisory Control And Data Acquisition (SCADA), sebuah solusi yang menyediakan kontrol proses untuk pabrik dan pabrik. Kami menjumpai PLC ketika bekerja dengan klien yang ingin memanfaatkan IoT untuk menghubungkan produk, pabrik, dan pabrik jarak jauh mereka ke cloud guna memanfaatkan analitik prediktif untuk menghitung waktu rata-rata hingga kegagalan sistem jarak jauh mereka (lihat Gambar 4-3). Jika Anda dapat memperkirakan waktu terjadinya kegagalan, Anda dapat mengurangi biaya pengoperasian dengan mengoptimalkan pemeliharaan terjadwal.



Schneider Electric

ABB

Horner

Gambar 4-3. Contoh PLC industri ringan dan berat

PLC tidak mendukung IoT secara default. Meskipun SBC saat ini umum digunakan untuk menyediakan konektivitas dan komunikasi cloud, modem GSM telah menangani tugas ini selama bertahun-tahun. Mari kita terjun ke dunia perangkat dengan terlebih dahulu melihat versi awal perangkat berkemampuan jaringan dan bagaimana perangkat tersebut berevolusi menjadi apa yang sekarang kita sebut perangkat pintar.

4.2 PERANGKAT

Sebagaimana dinyatakan dalam Bab 1, IoT bukanlah hal baru. Bagi industri tertentu, kemampuan untuk melacak dan berkomunikasi sangat penting bagi bisnis mereka seperti industri kontainer pengiriman, mesin penjual otomatis, dan angkutan truk. Sebelum ledakan “benda” terhubung yang memanfaatkan kemajuan terkini dalam teknologi mikroprosesor dan teknologi nirkabel berdaya rendah, penggunaan modem seluler telah digunakan secara luas untuk menyediakan komunikasi mesin-ke-mesin (M2M). Merupakan hal yang umum bagi perusahaan untuk memanfaatkan cloud publik sekaligus mendukung ekosistem M2M lama mereka menggunakan infrastruktur IoT yang sama.

Modem GSM

Modem GSM adalah perangkat berkemampuan seluler. Mereka dapat menerima kartu SIM dan perangkat lunak yang tertanam dalam host dan terhubung ke penyedia seluler seperti ATT atau Verizon. Mereka menggunakan port serial RS232, RS485, atau RS422 untuk terhubung ke PLC yang menyediakan data untuk dikirim (lihat Gambar 4-4). Setelah terhubung ke operator, perangkat lunak yang tertanam menggunakan koneksi VPN ke jaringan perusahaan untuk terhubung ke alamat IP yang dikenal—server—yang menerima data masuk dan memberikan tanggapan. Koneksi dari operator melalui VPN ke server dan transmisi data tidak berkelanjutan, seperti yang kita harapkan di dunia jaringan yang ada di mana-mana dan hal-hal yang terhubung. Iramanya biasanya beberapa kali sehari dan dalam beberapa kasus lebih jarang untuk mengendalikan biaya.



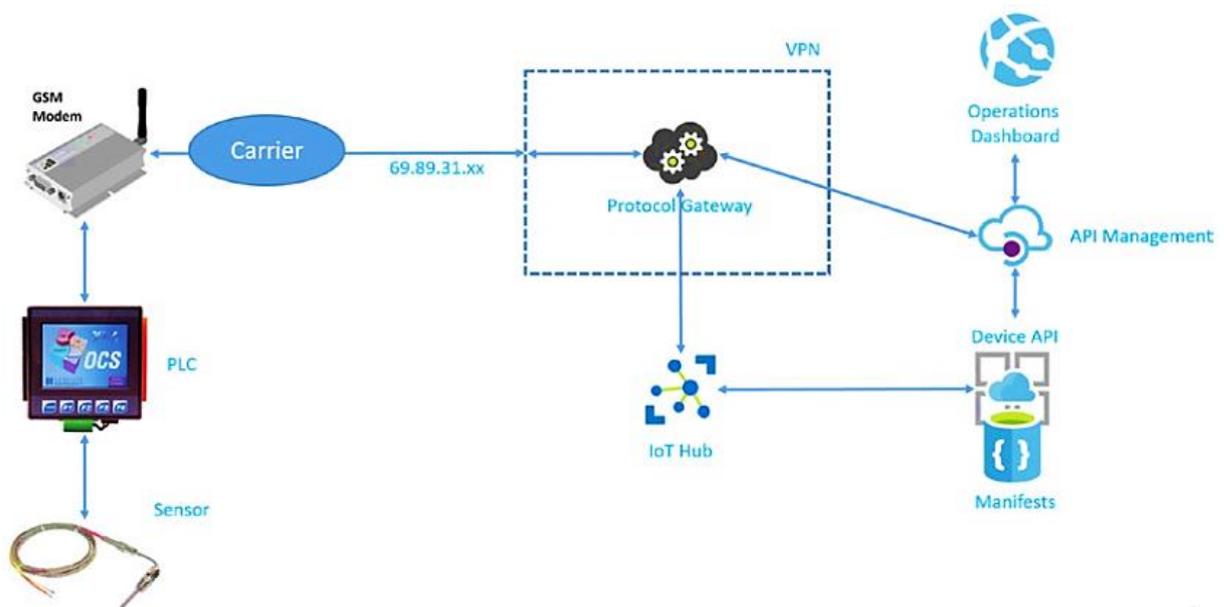
Gambar 4-4. Modem GSM serikat insinyur Elementz

Untuk mengintegrasikan modem GSM dengan sub-sistem IoT yang dihosting Azure, kami perlu menyediakan proksi yang berada di antara modem GSM dan Azure IoT Hub. Proksi ini menampilkan dirinya pada modem GSM sebagai server beralamat IP yang berada di belakang jaringan VPN aman dan mengelola interaksi dengan cloud. Kami menyebut proxy ini sebagai gateway protokol karena tugasnya adalah menerjemahkan format data lama dan perintah modem GSM yang didukung oleh perangkat lunak tertanam modem ke format pesan kanonik yang diharapkan oleh layanan kami.

Gerbang Protokol

Gateway protokol adalah layanan cloud yang diterapkan dalam konteks VPN dan dibuat dapat diberi alamat IP melalui konfigurasi cloud. Gateway protokol menyediakan lapisan terjemahan pesan dan perintah-dan-kontrol antara modem GSM dan Azure IoT Hub. Gateway protokol juga bertindak sebagai proksi manajemen perangkat, menjalankan proses startup perangkat seperti yang diuraikan dalam Bab 3, memanfaatkan API perangkat untuk mencari manifes perangkat dan menggunakan informasi tersebut untuk terhubung ke IoT Hub atas nama modem GSM.

Microsoft menyediakan Azure Protocol Gateway sebagai titik awal untuk implementasi gateway protokol. Azure Protocol Gateway adalah kerangka kerja yang memungkinkan komunikasi dua arah dengan Azure IoT Hub. Model pemrograman gateway protokol juga memungkinkan Anda menyambungkan komponen khusus untuk pemrosesan khusus seperti autentikasi, transformasi pesan, kompresi, dekompresi, serta enkripsi dan dekripsi lalu lintas antara perangkat jarak jauh dan IoT Hub. Karena modem GSM kadang-kadang terhubung, fitur ini dapat digunakan untuk menambahkan plugin untuk menyimpan perintah manajemen perangkat yang diterapkan saat modem melakukan panggilan berikutnya (lihat Gambar 4-5).



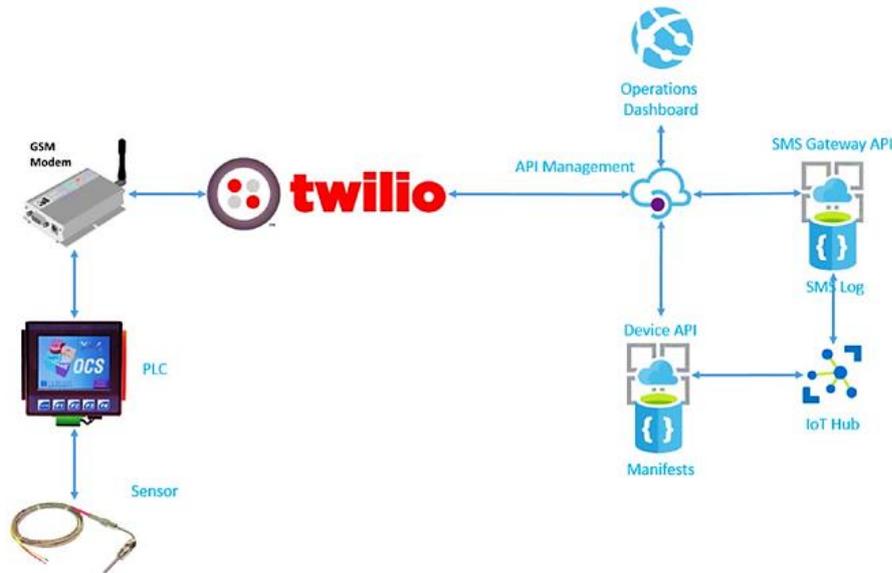
Gambar 4-5. Integrasi modem GSM menggunakan gateway protokol

Modem GSM dan SMS

Karena modem GSM pada dasarnya adalah sebuah ponsel tanpa keyboard, layar, speaker, dan jack headphone (apakah masih ada?), dan dapat menggunakan Layanan Pesan Singkat (SMS) untuk mengirim dan menerima pesan. Meskipun ini bukan pilihan komunikasi yang paling dapat diandalkan, ini masih dapat berperan dalam mengkomunikasikan informasi kecil untuk skenario yang kadang-kadang terhubung dengan cara yang hemat biaya.

Untuk mendukung komunikasi SMS, kita bisa menggunakan layanan seperti Twilio untuk menyediakan komunikasi dua arah dengan modem GSM. Untuk melengkapi rantai komunikasi, kami dapat menambahkan layanan ReST tambahan yang menyediakan titik akhir

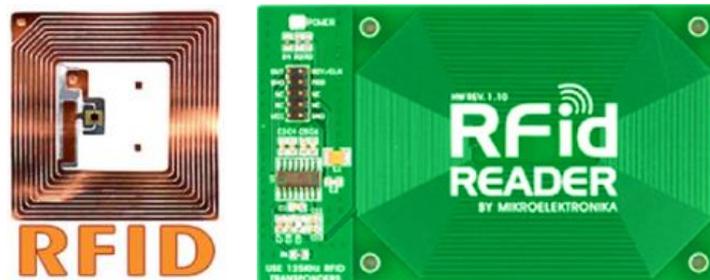
untuk Twilio dan menyediakan terjemahan protokol antara pesan SMS yang datang dari Twilio dan layanan kami. Sebut saja layanan ini sebagai SMS Gateway API. Modem GSM mengirimkan pesan SMS ke Twilio yang selanjutnya meneruskan pesan tersebut ke SMS Gateway API. SMS Gateway mengikuti proses startup perangkat yang diuraikan dalam Bab 3, memanggil API perangkat untuk mengambil manifest modem GSM dan menggunakan informasi tersebut untuk menyambung ke IoT Hub (lihat Gambar 4-6).



Gambar 4.6. Integrasi modem GMS menggunakan Twilio dan API SMS Gateway khusus

RFID

Identifikasi Frekuensi Radio (RFID) adalah sistem yang menggunakan tag yang ditempelkan atau disematkan pada suatu benda. Tag bisa pasif atau aktif. Tag aktif bertenaga baterai dan diaktifkan ketika pembaca RFID berada dalam jarak dekat. Tag berisi ID unik, nomor seri, dan data khusus aplikasi. Misalnya, jika tag RFID ditempelkan pada pakaian di toko ritel, tag tersebut akan berisi nomor seri unik beserta detail produk pakaian tersebut seperti nomor SKU, nama, warna, ukuran, dll.



Gambar 4.7. Tag dan pembaca RFID

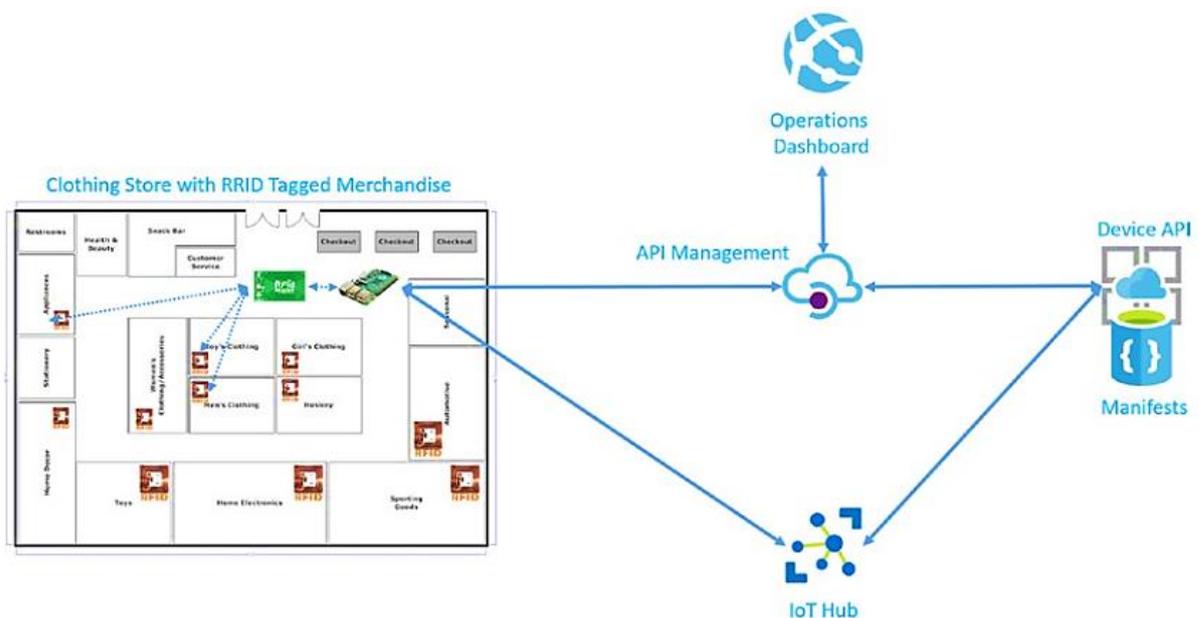
Untuk berkomunikasi dengan tag, pembaca RFID radio dua arah mengirimkan sinyal ke tag dan menerima data yang disimpan sebagai respons (lihat Gambar 4-7). Tag mungkin hanya dapat dibaca atau dapat ditulis, yaitu pembaca RFID dapat memperbarui konten tag. Sistem pembaca RFID memiliki jangkauan transmisi/penerimaan antara 1 hingga 2.000 kaki.

Rentangannya dapat dikalibrasi sehingga tag RFID hanya dibaca ketika berada dalam zona yang ditentukan dengan jelas di sekitar pembaca. Hal ini berguna, misalnya, jika Anda ingin mengatur zona pada ban berjalan sehingga tag terbaca ketika melewati bawah pembaca.

Saat digunakan dalam skenario IoT, tag RFID dan pembaca RFID digunakan bersama dengan perangkat pintar yang berkomunikasi ke cloud serta mengubah, memfilter, dan menggabungkan data RFID.

Misalnya, toko pakaian dapat memberi tag RFID pada semua barang dagangan dan memasang pembaca RFID di langit-langit untuk terus berkomunikasi dengan tag RFID guna melacak lokasi produk dan informasi SKU. Dengan menggunakan informasi ini, Anda akan dapat membuat tampilan inventaris toko dan lokasi produk secara real-time menggunakan layanan IoT. Meskipun beberapa pengecer menggunakan RFID, mereka melacak inventaris secara manual menggunakan perangkat genggam RFID dan mereka tidak mengoordinasikan data inventaris toko dengan tingkat inventaris yang diiklankan di situs web mereka. Solusi yang menyediakan inventaris dan lokasi produk secara real-time yang divisualisasikan melalui web dan aplikasi seluler untuk manajemen toko, karyawan, dan pelanggan akan menjadi hal yang transformatif dalam industri ritel.

Perangkat pintar, dalam skenario ini, akan menyediakan penyaringan dan transformasi data selain menangani komunikasi dengan layanan cloud (lihat Gambar 4-8).



Gambar 4.8. Toko pakaian menggunakan RFID untuk melacak inventaris dan lokasi produk secara real-time

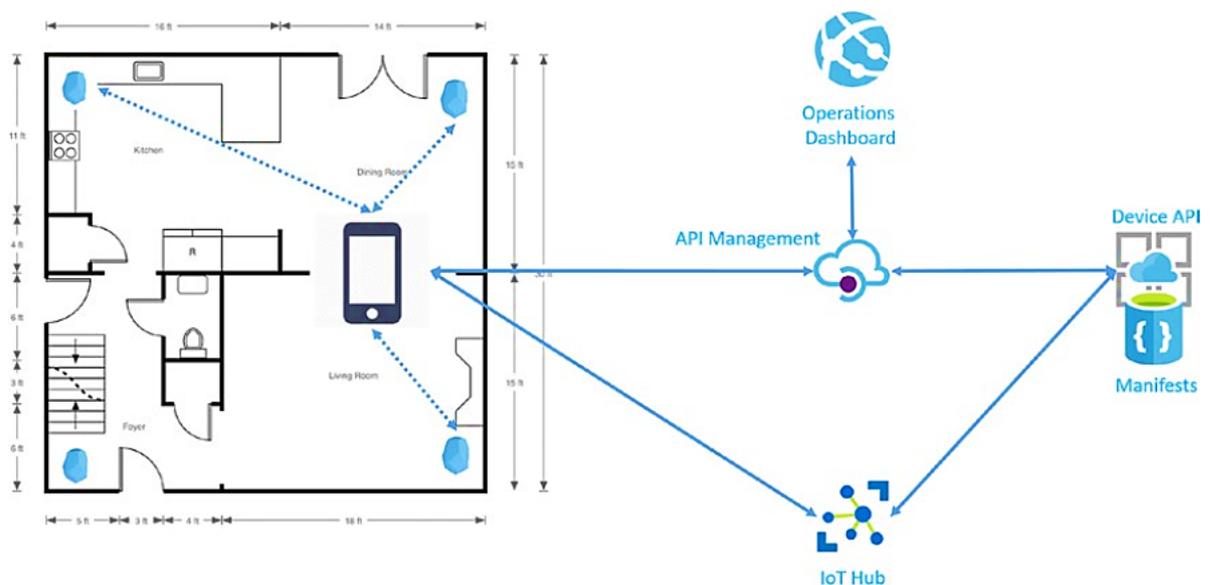
Suar Bluetooth

Bluetooth adalah standar teknologi nirkabel untuk bertukar data dalam jarak pendek. Suar Bluetooth adalah perangkat kecil bertenaga baterai yang dapat dipasang atau tertanam pada berbagai benda. Penerima berkemampuan Bluetooth apa pun seperti ponsel atau perangkat pintar dengan tumpukan Bluetooth terpasang dapat terhubung dan berkomunikasi

dengan suar saat berada dalam jangkauan. Jangkauannya bervariasi menurut kelas perangkat, mulai dari kurang dari satu meter hingga lebih dari 100 meter (lihat Gambar 4-9).



Gambar 4.9. Suar Bluetooth yang didekonstruksi



Gambar 4.10. Suar Bluetooth dan ponsel digunakan untuk kesadaran lokasi

Skenario yang umum adalah menggunakan suar Bluetooth yang ditempatkan di dalam gedung bersama dengan aplikasi ponsel untuk memberikan kesadaran lokasi. Dimungkinkan untuk menggunakan beberapa suar Bluetooth untuk melakukan triangulasi dan menentukan tidak hanya koordinat 2D dalam suatu ruang tetapi juga menghitung ketinggian yang menyediakan lokasi 3D (lihat Gambar 4-10). Di ruang tamu dengan bantuan, ini dapat digunakan untuk menentukan apakah seseorang sedang berdiri atau berbaring di lantai dan mungkin membutuhkan bantuan. Solusi IoT dapat secara otomatis mengingatkan petugas pertolongan pertama, mematikan peralatan, dan membuka kunci pintu ketika petugas pertolongan pertama tiba.

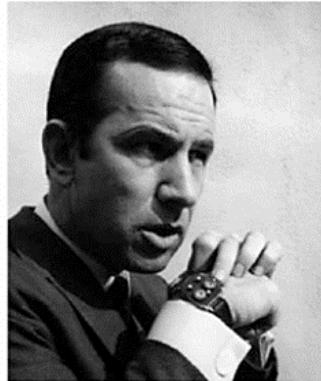
Ayo Pintar

Beacon RFID dan Bluetooth hanya berguna bila dikombinasikan dengan perangkat pintar. Jadi, apa itu perangkat pintar? Hollywood telah melakukan sebagian besar pekerjaan visioner di bidang ini. Kita hanya perlu melihat film dan TV populer untuk menemukan

prototipe awal perangkat dan teknologi yang sekarang kita gunakan setiap hari. Don Adams, yang membintangi serial TV terkenal *Get Smart* dari tahun 1965 hingga 1970, dapat dilihat pada Gambar 4-11 yang menunjukkan versi awal perangkat pintar yang sekarang kita gunakan setiap hari.



The Mobile Phone



Smart Watch



Skype for Business

Gambar 4.11. Prototipe perangkat pintar yang visioner (Manajemen General Artists Corporation-GAC. Dittransfer dari en.wikipedia ke commons.)

Perangkat pintar, seperti Single-Board Computer (SBC), terdiri dari papan mikroprosesor seukuran kartu kredit dengan chipset, memori, penyimpanan, dan kemampuan I/O yang kuat. Ia dapat terhubung ke dan membaca sensor menggunakan teknologi seperti Modbus untuk PLC, sinyal radio untuk tag RFID, dan Bluetooth untuk suar. Ia juga dapat menggunakan saluran I/O pada motherboard untuk berkomunikasi dengan sensor yang terpasang. Ini mungkin menggunakan modem kabel, nirkabel, atau seluler untuk terhubung ke jaringan lokal atau operator dan ke cloud publik. Itu dapat menjalankan aplikasi yang memproses data sensor, menerapkan operasi matematika, dan menangani semua interaksi dengan cloud dan API. Dengan kata lain, perangkat pintar bisa secerdas yang Anda programkan. Penting untuk diingat bahwa ponsel adalah perangkat pintar dan sering digunakan dalam skenario IoT seperti yang dijelaskan sebelumnya. Ponselnya sendiri sudah dibenamkan sensor seperti GPS dan accelerometer. Itu juga dapat menjalankan aplikasi yang menggunakan Bluetooth untuk terhubung ke suar Bluetooth saat berada di dekatnya.

Komputer Papan Tunggal

Raspberry Pi, Qualcomm DragonBoard, dan Intel Edison adalah contoh SBC kecil, murah, namun kuat yang dapat menjalankan sistem operasi tertanam seperti Linux dan Windows 10 IoT. Mereka dapat dikustomisasi menggunakan pelindung (papan tambahan), sensor, dan layar LCD yang dipasang menggunakan pin atau kabel. Papan mikroprosesor ini juga mendukung penyambungan layar HDMI, kartu MicroSD, dan perangkat USB (lihat Gambar 4-12).



Gambar 4.12. Contoh SBC dengan pelindung dan kit sensor

SBC dapat digunakan untuk membuat prototipe solusi perangkat keras IoT Anda dan bahkan menjadi fondasi ekosistem perangkat Anda. Anda mungkin perlu menutup SBC Anda dengan material industri ringan hingga berat tergantung pada kondisi lingkungan di mana perangkat akan digunakan (lihat Gambar 4-13). Apakah perangkat berada di dalam atau di luar ruangan atau terkena panas atau dingin ekstrem, debu, kelembapan, atau getaran? Hal ini perlu dipertimbangkan ketika merancang solusi perangkat keras Anda.



Gambar 4.13. Contoh enclosure Raspberry Pi

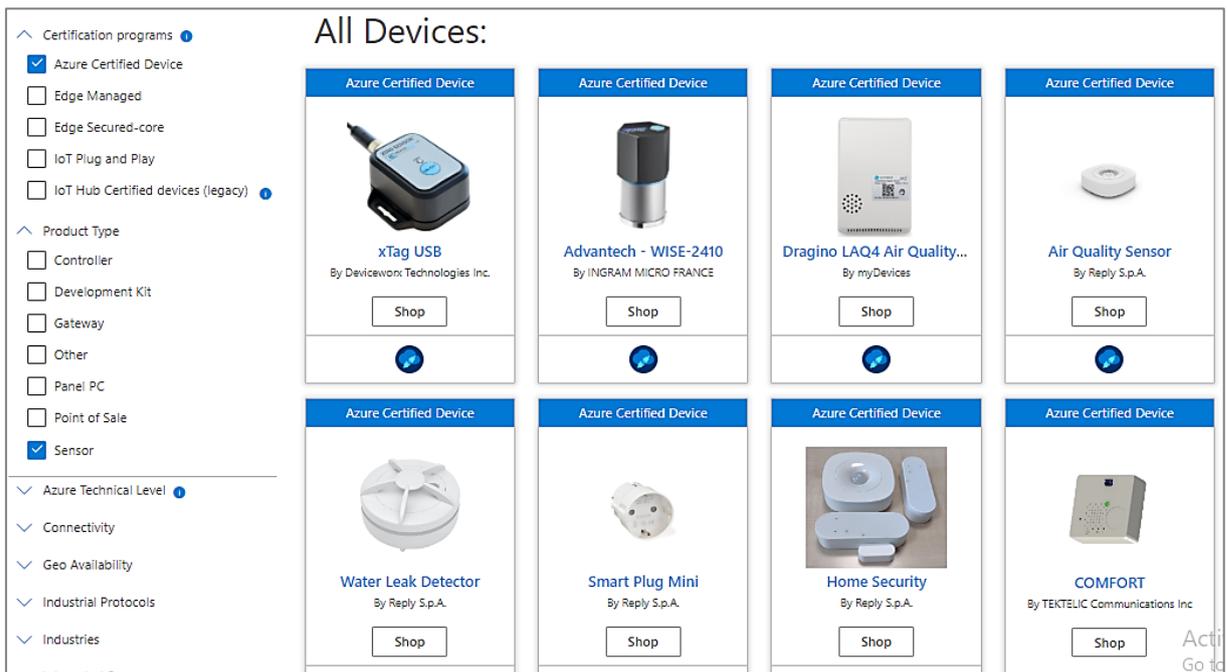
Untuk mengembangkan perangkat lunak mikrokontroler tertanam, Microsoft menyediakan SDK Perangkat IoT dalam beberapa bahasa pemrograman yang berbeda, termasuk Python, Node.js, Java, C, C++, dan C#.

SDK Perangkat Azure IoT dapat digunakan dengan berbagai platform dan perangkat sistem operasi (OS). Persyaratan minimumnya adalah platform perangkat:

- *Mampu membuat koneksi IP*: Hanya perangkat berkemampuan IP yang dapat berkomunikasi langsung dengan Azure IoT Hub.
- *Mendukung Keamanan Lapisan Transportasi (TLS)*: Diperlukan untuk membangun saluran komunikasi yang aman dengan Azure IoT Hub.
- *Dukungan SHA-256*: Diperlukan untuk menghasilkan token aman untuk mengautentikasi perangkat dengan layanan.
- *Memiliki Jam Waktu Nyata atau Kode Implementasi untuk Terhubung ke Server Network Time Protocol (NTP)*: Diperlukan untuk membuat koneksi TLS dan untuk menghasilkan token aman untuk otentikasi.

- *Memiliki Setidaknya 64KB RAM:* Jejak memori SDK bergantung pada SDK dan protokol yang digunakan serta platform yang ditargetkan. Jejak terkecil dicapai dengan menggunakan mikrokontroler penargetan C SDK.

Anda dapat menemukan daftar lengkap platform OS tempat berbagai SDK telah diuji di katalog perangkat Azure Certified untuk IoT (lihat Gambar 4-14). Perlu diperhatikan bahwa Anda mungkin masih dapat menggunakan SDK pada OS dan platform perangkat keras yang tidak tercantum. Semua SDK bersifat open source dan dirancang agar portabel. Katalog Perangkat IoT Bersertifikasi Azure terletak di <https://catalog.azureiotsuite.com/>.



Gambar 4.14. Katalog perangkat IoT bersertifikasi Azure

Perangkat Lunak Mikrokontroler

Kini setelah Anda memiliki perangkat kecil dan ringkas yang menjalankan sistem operasi tertanam yang modern, aman, dan tumpukan jaringan lengkap, Anda dapat memanfaatkan kekuatan tersebut untuk mengimplementasikan perangkat lunak pengontrol yang akan membaca data sensor dan berintegrasi langsung dengan IoT Hub dan API ReST.

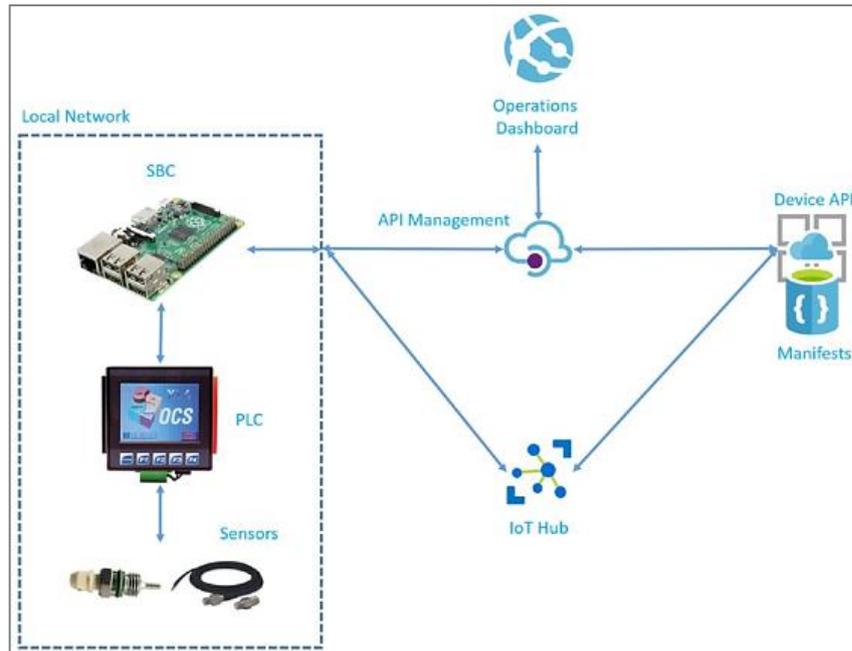
Ada tiga pola pesan inti yang digunakan oleh perangkat pintar:

- *Detak jantung:* Mengirim pesan ping sesekali secara berkala
- *Telemetri:* Sering mengirim telemetri secara berkala
- *Perintah dan Kontrol:* Menerima perintah dan data yang masuk seperti yang dibahas dalam Bab 3, IoT Hub menyediakan dua pola pesan tambahan yang merupakan variasi perintah dan kontrol:
- *Peristiwa Kembar Perangkat:* Menerima peristiwa perubahan properti perangkat
- *Metode Langsung:* Menyediakan pengendali fungsi untuk pemanggilan permintaan/respons jarak jauh

Masing-masing operasi ini ditangani secara asinkron, memastikan bahwa perangkat memiliki performa yang baik dan tidak ada satu operasi pun yang menghalangi operasi lainnya. Selain SDK klien perangkat, kami ingin memanggil API Perangkat kami untuk mengambil manifes perangkat. Jika Anda ingat, manifes perangkat berisi URI ke instans IoT Hub Anda dan kunci simetris untuk diautentikasi oleh perangkat. Contoh C# berikut menunjukkan urutan startup untuk perangkat yang menjalankan Windows 10 IoT.

```
// reference the Azure Devices Client SDK
using Microsoft.Azure.Devices.Client;
...
// device constants; Id, Device API and Dev Key
private const string DeviceSerialNumber = "[your-device-id]";
private const string DeviceApi =
    "https:// [your-apim-host].azure
        api.net/dev/v1/device/manifests/id/" + DeviceSerialNumber;
private const string SubscriptionKey =
    "subscription-key= [your-apim-devkey]";
...
// properties for manifest and IoT Hub client
private static Manifest _deviceManifest;
private static DeviceClient _deviceClient;
...
// invoke the Device API to retrieve the device manifest
_deviceManifest = await GetDeviceManifest();
    // connect to IoT Hub
_deviceClient = DeviceClient.Create(
    _deviceManifest.Hub,
    AuthenticationMethodFactory.
        CreateAuthenticationWithRegistrySymmetricKey(
    _deviceManifest.SerialNumber,
    _deviceManifest.Key.PrimaryKey),
    TransportType.Amqp);
```

Beberapa baris kode ini memberikan integrasi yang aman dengan layanan cloud Anda (lihat Gambar 4.15). Kami memeriksa kode ini secara lebih rinci nanti di bab ini saat Anda membuat perangkat lunak yang tertanam di perangkat pintar Anda sendiri.



Gambar 4-15. Integrasi perangkat pintar ke layanan cloud

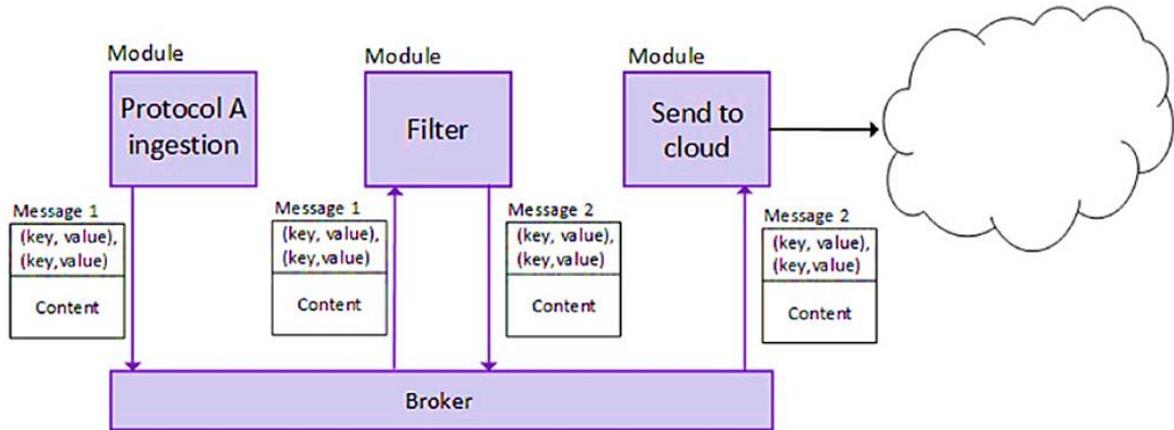
Gerbang Tepi

Edge gateway adalah perangkat pintar canggih yang bertindak sebagai perantara antara ekosistem perangkat dan cloud. Dengan memperkenalkan perantara ini, Anda dapat menerapkan operasi lebih lanjut pada data sensor secara lokal sebelum dikirim ke cloud. Operasi lokal ini termasuk namun tidak terbatas pada:

- *Agregasi*: Membangun kumpulan pembacaan sensor
- *Logging*: Menyediakan penyimpanan sementara dari peristiwa, koleksi, peringatan terkini, dll.
- *Analisis*: Menerapkan operasi matematika pada kumpulan pembacaan sensor seperti rata-rata, deviasi standar, Fast Fourier Transform, regresi linier, dll.
- *Penyaringan*: Menerapkan filter ke kumpulan pembacaan sensor
- *Aturan*: Menerapkan aturan bisnis yang ditentukan oleh ambang batas atas dan bawah yang ditentukan oleh properti sensor
- *Peringatan*: Mengidentifikasi pembacaan di luar batas dan peringatan sinyal menggunakan pemberitahuan fisik seperti lampu berkedip dan sirene mendesing atau memanfaatkan kemampuan seluler untuk mengirim pesan teks dan suara
- *Windowing*: Mengumpulkan telemetri dan/atau kondisi peringatan selama jangka waktu tertentu dan hanya melaporkan bila ambang batas windowing telah terlampaui, misalnya, melakukan peringatan ketika terdapat lebih dari lima kesalahan setiap menitnya

Perangkat lunak mikrokontroler yang berjalan pada perangkat edge gateway menentukan jalur data untuk setiap jenis telemetri yang menjadi tanggung jawabnya. Pipeline data terdiri dari satu atau lebih modul yang bertindak atas data yang mengalir melalui pipeline. Urutan operasi penting, dan Anda juga ingin dapat mengonfigurasi modul saat runtime.

Microsoft menyediakan Azure IoT Gateway SDK untuk membantu pengembangan gateway edge. SDK ini menyediakan kerangka kerja untuk mendefinisikan modul pluggable yang berkomunikasi melalui tulang punggung perantara pesan. Setiap modul yang Anda tentukan dan konfigurasi melakukan operasinya pada pesan yang masuk dan menulis pesan yang mungkin diubah kembali ke broker untuk modul berikutnya.



Gambar 4-16. Arsitektur gerbang Azure IoT

Dalam latihan berikut, Anda mempelajari cara membuat perangkat pintar sederhana dan menyebarkan aplikasi ke SBC. Selain itu, Anda akan memperbarui simulator perangkat yang digunakan untuk menghasilkan data biometrik untuk solusi IoT.

4.3 BUAT PERANGKAT CERDAS

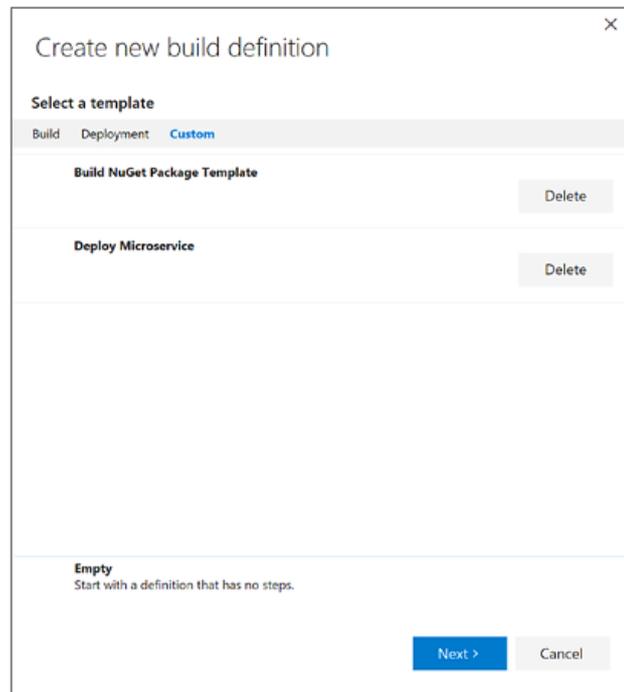
Dalam latihan ini, Anda membuat perangkat pintar dasar menggunakan bahasa C# dan template Universal Windows Application (UWP). Dengan menggunakan templat ini, solusinya dapat berjalan secara lokal di desktop Anda dengan menyediakan simulator, dan dapat diterapkan ke SBC yang menjalankan Windows 10 IoT.

Persyaratan untuk latihan ini:

- Sistem Operasi Windows 10
- Visual Studio 2015 dengan Pembaruan 3
- SDK Windows 10
- Templat Visual Studio IoT Inti Windows 10
- Azure IoT SDK

Sebelum kita mendalami proyek perangkat, Anda perlu membuat versi UWP dari dua pustaka utama sehingga Anda dapat memanfaatkannya dalam perangkat lunak mikrokontroler Anda: DeviceModels dan MessageModels. Jika Anda telah menyelesaikan latihan di Bab 2, Anda akan memiliki versi perpustakaan ini yang menargetkan .NET 4.6 dan digunakan oleh layanan yang dihosting di cloud. Kode untuk versi UWP perpustakaan ini sudah menjadi bagian dari repositori kode. Langkah selanjutnya ini akan membuat definisi build dan memungkinkan Anda menambahkannya ke feed paket NuGet Anda.

1. Buat definisi build baru menggunakan template Build NuGet Package yang Anda buat di Bab 2 (lihat Gambar 4-17).



Gambar 4-17. Buat definisi build dari templat

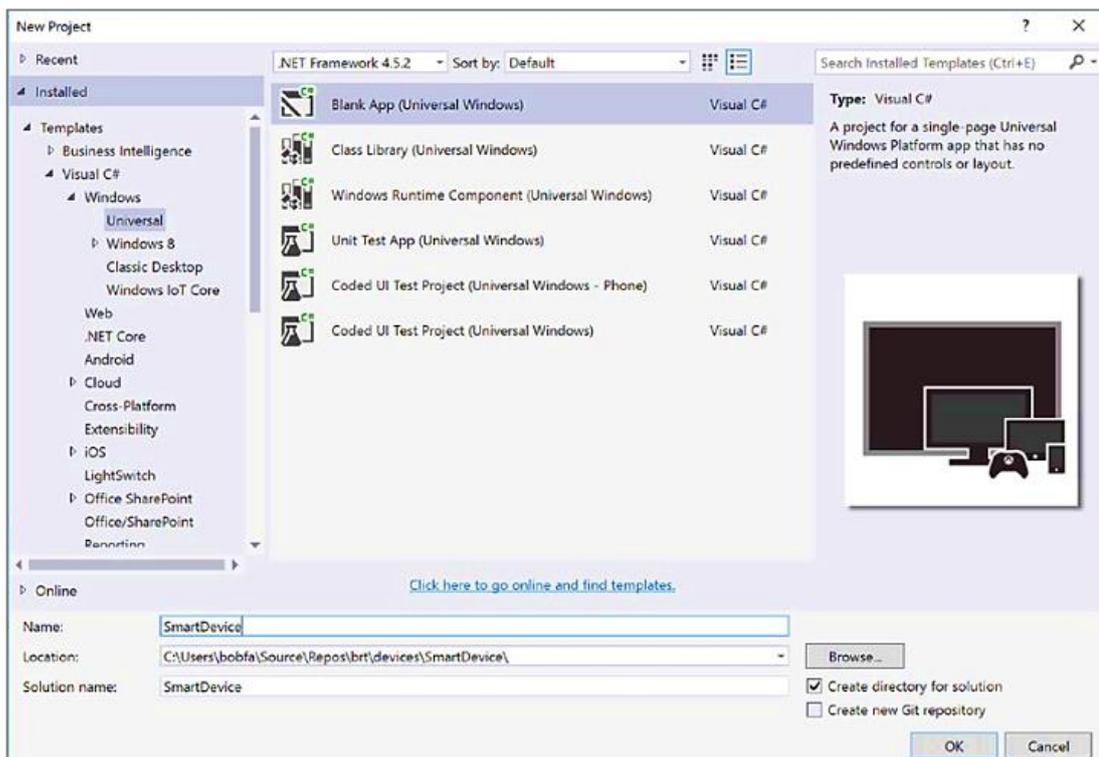
2. Isi pengaturan definisi build untuk solusi `MessageModelUWP` seperti yang ditunjukkan pada Tabel 4-1. Solusi ini telah disediakan sebagai bagian dari repo implementasi referensi.

Tabel 4-1. Bangun Definisi Pembuatan Paket NuGet MessageModelUWP

<u>NuGet Installer Settings</u>	
Path to solution	<code>Models/Message/UWP/MessageModels.sln</code>
Path to NuGet.Config	<code>nuget.config</code>
NuGet arguments	<code>-outputdirectory packages</code>
NuGet version	3.5
<u>MSBuild Settings</u>	
Path to solution	<code>models/Message/UWP/MessageModels.sln</code>
Configuration	<code>debug</code>
<u>NuGet Packager</u>	
Path to CSProj	<code>models\Message\UWP\MessageModels\MessageModels.csproj</code>
Package Folder	<code>nugets</code>
Include referenced projects	Yes (checked)
Configuration to package	<code>debug</code>
<u>NuGet Publisher</u>	
Path to NuPkg	<code>nugets/Message*.nupkg</code>
Feed type	Internal

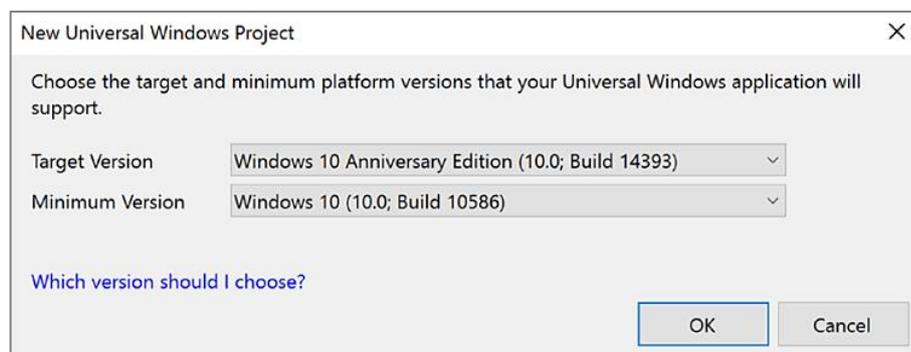
Internal feed URL	[URL to your package feed]
NuGet Version	3.5

3. Simpan definisi build. Buat definisi build lain untuk solusi `DeviceModelsUWP` menggunakan pengaturan yang sama. Masukkan dua definisi build baru ke dalam antrean untuk menambahkan paket NuGet ini ke feed Anda.
4. Buat proyek di Visual Studio bernama `SmartDevice` menggunakan template Blank App (Universal Windows) (lihat Gambar 4-18).



Gambar 4-18. Buat proyek aplikasi universal Windows

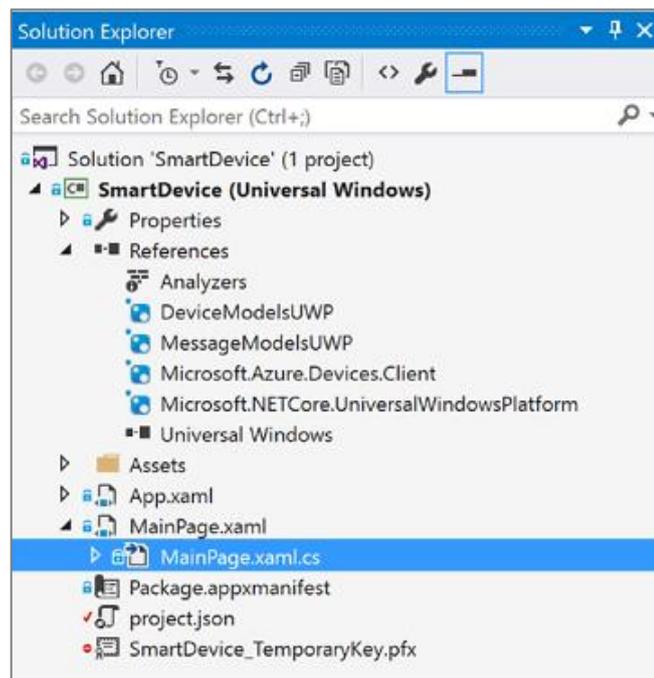
Anda akan disajikan kotak dialog yang menginformasikan versi target dan versi minimum Windows 10 yang akan didukung oleh proyek (lihat Gambar 4-19). Pilih default.



Gambar 4-19. Proyek Windows universal baru

5. Setelah proyek selesai diinisialisasi, klik kanan pada node Referensi di Solution Explorer dan pilih Kelola Paket NuGet dari menu.

Pilih NuGet.org sebagai lokasi sumber paket dan cari `Azure.Devices`. Tambahkan referensi ke paket NuGet `Microsoft.Azure.Devices.Client`. Tambahkan juga referensi ke paket `Newtonsoft.Json` NuGet. Pilih feed NuGet Anda sebagai sumber paket dan tambahkan referensi ke paket NuGet `DeviceModelUWP` dan `MessageModelUWP` yang baru saja Anda buat. Setelah selesai, referensi solusi Anda akan muncul seperti yang digambarkan pada Gambar 4-20.



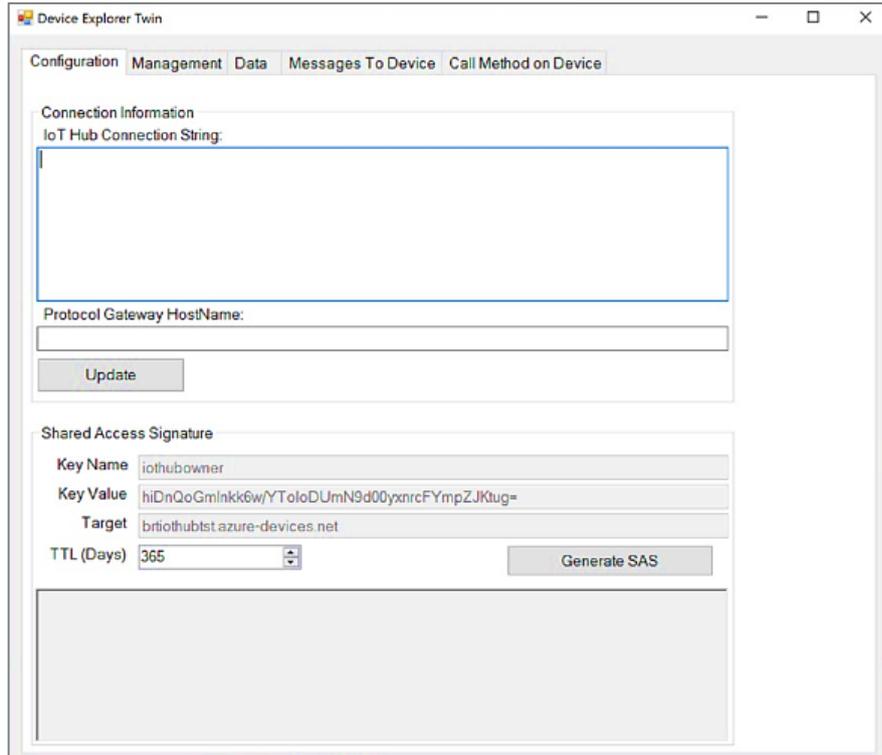
Gambar 4-20. Referensi Perangkat Cerdas

6. Buka file dan tambahkan pernyataan penggunaan ini di bagian atas.

```
using Microsoft.Azure.Devices.Client;
using brt.Models.Message;
using brt.Models.Device;
using Newtonsoft.JSON;
```

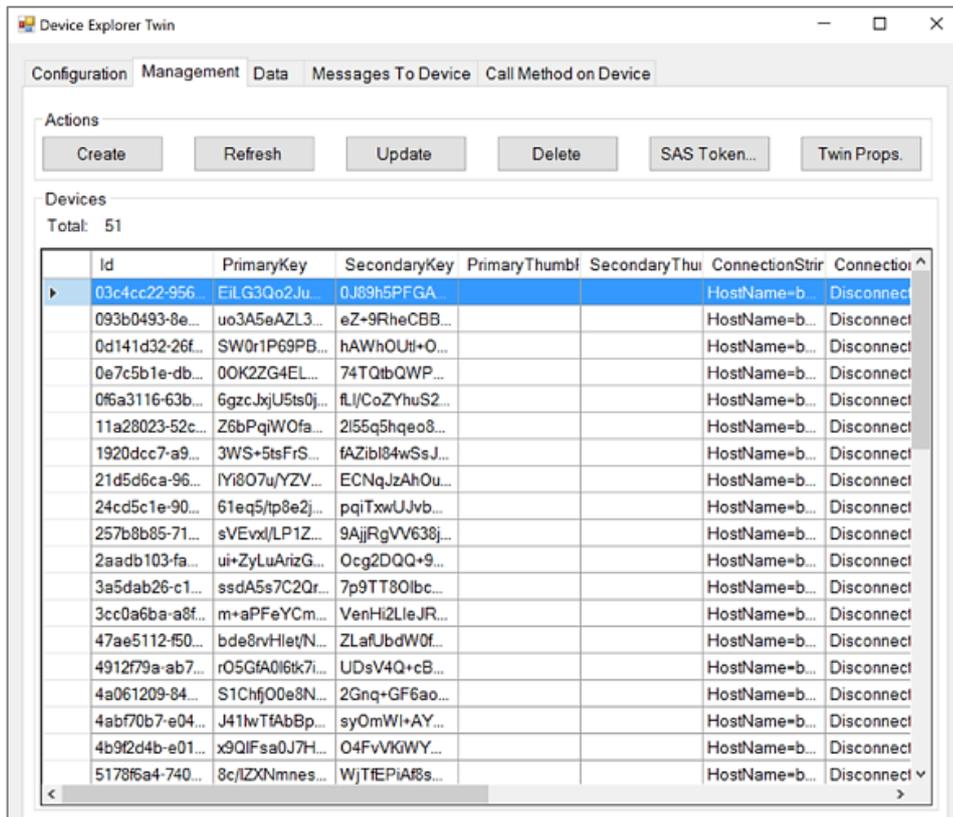
Anda memerlukan ID unik untuk perangkat pintar Anda. Kami mendaftarkan beberapa perangkat menggunakan utilitas Bootstrap di Bab 2. Anda dapat menggunakan utilitas Device Explorer yang merupakan bagian dari Azure IoT SDK untuk mencari ID perangkat terdaftar di IoT Hub untuk digunakan pada perangkat pintar Anda.

7. Buka solusi Device Explorer di Visual Studio lain dan jalankan aplikasi (lihat Gambar 4-21).



Gambar 4-21. Penjelajah Perangkat

Masukkan string koneksi ke instans IoT Hub Anda pada tab Konfigurasi dan klik tombol Perbarui. Arahkan ke tab Manajemen dan klik Daftar (lihat Gambar 4-22).



Gambar 4-22. Device Explorer mencantumkan perangkat di IoT Hub

Pilih perangkat lalu klik kanan dan pilih untuk menyalin data untuk perangkat yang dipilih. Tempelkan informasi tersebut ke Notepad sehingga Anda dapat mengakses ID unik untuk digunakan sebagai nomor seri perangkat pintar. Anda juga memerlukan nama host API Management dan kunci pengembang untuk menginisialisasi properti yang menentukan identitas perangkat. Anda dapat mengambil pengaturan tersebut dari portal Pengembang Manajemen API.

8. Tambahkan variabel kelas berikut ke kelas MainPage dalam solusi UWP Anda. Gunakan ID perangkat yang Anda salin dari Device Explorer untuk digunakan sebagai nomor seri perangkat. Perbarui URI API untuk menggunakan nama host dan kunci langganan APIM Anda.

```
// Device Identity
private const string DeviceSerialNumber = "[device-id]";
private const string DeviceApi = "https://[your-apim
    host].azure-api.net/dev/v1/device/manifests/id/"+
    DeviceSerialNumber;
private const string SubscriptionKey =
    "subscription-key=[your-apim-devkey] ";
private static Manifest _deviceManifest;
private static DeviceClient _deviceClient;

// Thread Management
private static Task _heartbeatTask;
private static Task _telemetryTask;
```

9. Buka file MainPage.xaml dan tambahkan event handler MainPage_OnLoaded() ke elemen <Page>. Penangan didefinisikan dalam file XAML dan fungsinya dihasilkan dalam file MainPage.xaml.cs.

```
<Page
...
    Loaded="MainPage_OnLoaded">
...
</Page>
```

10. Tambahkan kata kunci async ke tanda tangan metode MainPage_OnLoaded().

```
private async void MainPage_OnLoaded(
    object sender, RoutedEventArgs e)
{
    throw new System.NotImplementedException();
}
```

11. Pada file Mainpage.Xaml, ganti bagian elemen <Grid> dengan kode XAML ini:

```

<Grid Background="{ThemeResource
    ApplicationPageBackgroundThemeBrush}">
  <Grid.RowDefinitions>
    <RowDefinition Height="50*" />
    <RowDefinition Height="50*" />
  </Grid.RowDefinitions>

  <TextBlock Grid.Row="0"
    HorizontalAlignment="Center"
    VerticalAlignment="Center"
    FontSize="24">
    Windows 10 IoT Device
  </TextBlock>
  <StackPanel Grid.Row="1" Margin="10,10,10,10">
    <TextBox x:Name="Status"
      Margin="10"
      IsReadOnly="True"
      TextAlignment="Center" />
  </StackPanel>
</Grid>

```

12. Tambahkan dua stub metode `private static` untuk kode yang akan menangani rangkaian pesan latar belakang.

```

private static void StartHeartbeat()
{
}

private static void StartTelemetry()
{
}

```

13. Tambahkan metode `private static` bernama `GetDeviceManifest()` yang memanggil Device API untuk mengambil manifes perangkat.

```

private static async Task<Manifest> GetDeviceManifest()
{
    var client = new HttpClient();
    var uriBuilder = new UriBuilder(DeviceApi)
    {
        Query = SubscriptionKey
    };
    var json = await client.GetStringAsync(uriBuilder.Uri);
    var deviceManifest =
        JsonConvert.DeserializeObject<Manifest>(json);
    return deviceManifest;
}

```

14. Ganti isi event handler `MainPage_OnLoaded()` dengan kode ini.

```
Status.Text = "Main Page Loaded";

//get the device manifest
_deviceManifest = await GetDeviceManifest();

try
{
    //connect to IoT Hub
    _deviceClient = DeviceClient.Create(
        _deviceManifest.Hub,
        AuthenticationMethodFactory
            .CreateAuthenticationWithRegistrySymmetricKey(
                _deviceManifest.SerialNumber,
                _deviceManifest.Key.PrimaryKey),
        TransportType.Amqp);

    Status.Text = $"{_deviceManifest.SerialNumber}
        Connected to Azure IoT Hub";
}
catch (Exception connectionErr)
{
    Status.Text = connectionErr.Message;
}

StartHeartbeat();
StartTelemetry();
```

Kode ini akan memanggil API Perangkat untuk mengambil manifes perangkat dan menyambung ke IoT Hub menggunakan properti manifes perangkat. Jika berhasil, pesan status ditampilkan pada antarmuka pengguna aplikasi. Untuk mengimplementasikan pola pesan detak jantung, kita akan memulai tugas latar belakang dan dalam tugas tersebut secara terus-menerus mengirimkan pesan detak jantung berdasarkan pengaturan irama, yang didefinisikan sebagai properti ekstensi dalam manifes perangkat. Pesan detak jantung ditentukan di perpustakaan `MessageModelsUWP`. Ini adalah pesan sederhana yang berisi string pengakuan, nomor seri perangkat, serta garis bujur dan lintang perangkat.

15. Tambahkan kode berikut ke isi metode `StartHeartbeat()`.

```
var cadence = Convert.ToInt32(
    _deviceManifest.Extensions["heartbeat"]);

_heartbeatTask = Task.Factory.StartNew(async () =>
{
    while (true)
```

```

{
    // create a heartbeat message
    var heartbeat = new Heartbeat
    {
        Ack =
            $"({_deviceManifest.SerialNumber} is functioning",
            Longitude = _deviceManifest.Longitude,
            Latitude = _deviceManifest.Latitude,
            DeviceId = _deviceManifest.SerialNumber
        };
    var json = JsonConvert.SerializeObject(heartbeat);
    var message = new Message(Encoding.ASCII.GetBytes(json));

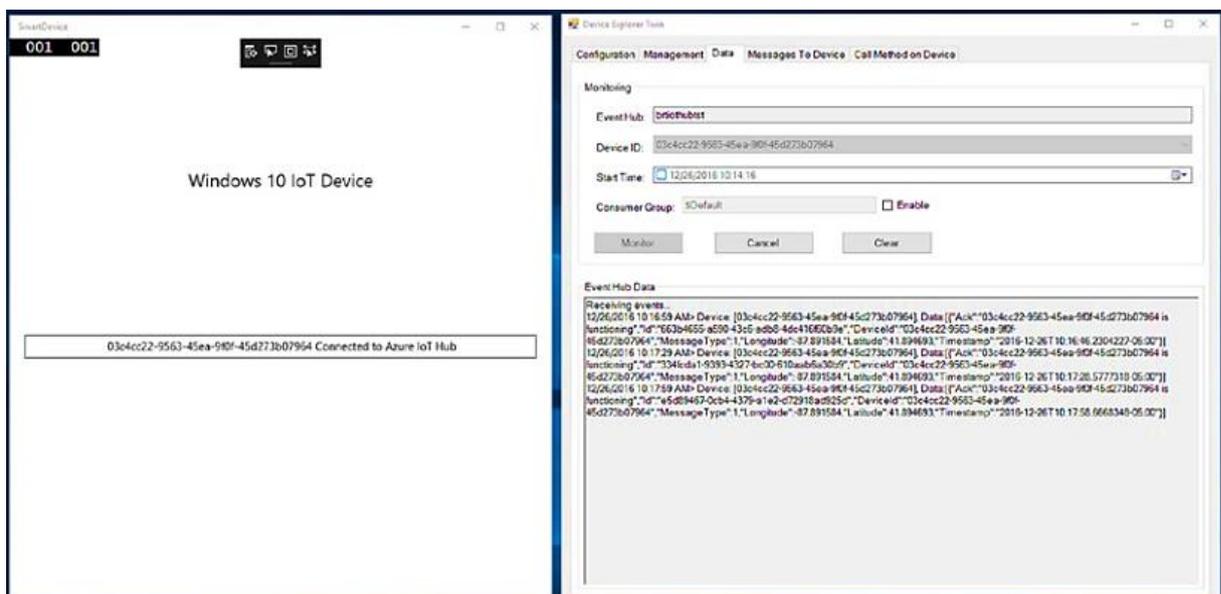
    await _deviceClient.SendEventAsync(message);

    await Task.Delay(cadence);
}
});

```

16. Untuk menguji perangkat pintar Anda, jalankan utilitas Device Explorer, pilih tab Data, dan mulailah memantau pesan yang masuk dari perangkat Anda dengan memilih ID-nya dari drop-down.

Mulai aplikasi Perangkat Cerdas Anda. Ini akan mulai mengirimkan pesan detak jantung setiap 30 detik (lihat Gambar 4-23).



Gambar 4-23. Perangkat pintar mengirimkan pesan detak jantung

Saat ini kami belum memiliki sensor yang terhubung. Kami akan melakukan simulasi pengukuran suhu dengan kisaran nilai yang diharapkan antara 60 dan 70. Pembacaan ini akan dikirim lebih sering daripada pesan detak jantung. Kami akan menggunakan kelas

SimpleSensorReading seperti yang didefinisikan di perpustakaan MessageModelsUWP. Seperti pesan detak jantung, pesan ini akan berisi nomor seri perangkat serta garis bujur dan lintang. Selain itu, ini akan berisi pembacaan floating point untuk suhu yang disimulasikan.

17. Tambahkan kode berikut ke metode StartTelemetry().

```

var cadence = Convert.ToInt32(
        _deviceManifest.Extensions["telemetry"]);

var random = new Random();

_telemetryTask = Task.Factory.StartNew(async () =>
{
    while (true)
    {
        //the temperature will be simulated
        //it will be a value between 60 and 70

        var sensorReading = new SimpleSensorReading
        {
            Longitude = _deviceManifest.Longitude,
            Latitude = _deviceManifest.Latitude,
            DeviceId = _deviceManifest.SerialNumber,
            Reading = random.Next(60, 70)
        };

        var json =
            JsonConvert.SerializeObject(sensorReading);

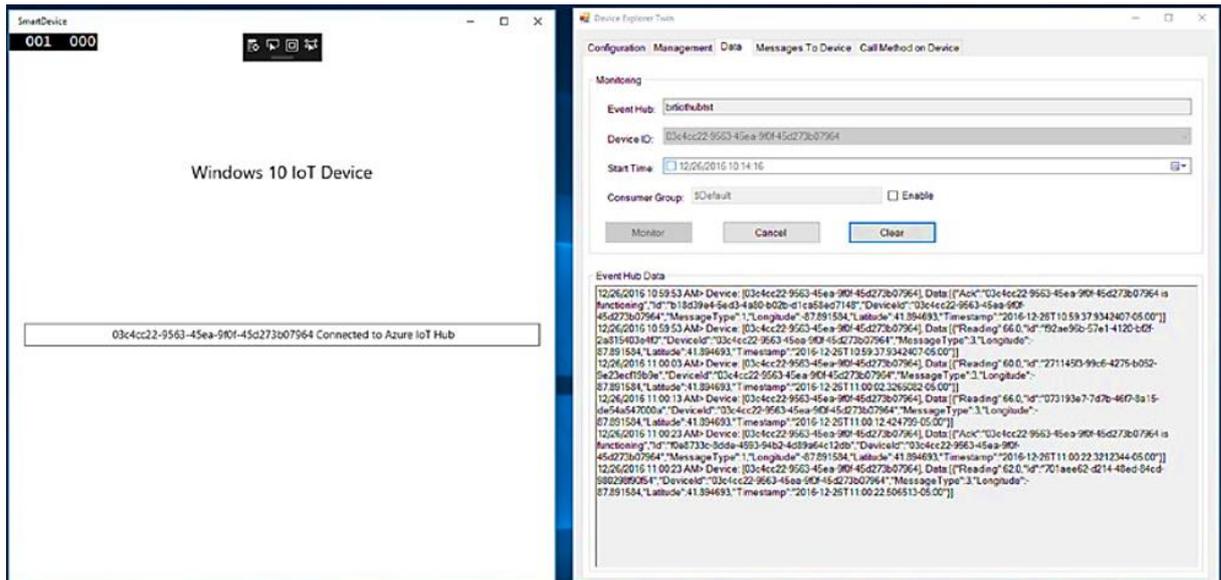
        var message = new Message(
            Encoding.ASCII.GetBytes(json));

        await _deviceClient.SendEventAsync(message);

        await Task.Delay(cadence);
    }
});

```

18. Jalankan aplikasi Perangkat Cerdas dan tinjau pesan masuk di IoT Hub menggunakan Device Explorer. Anda akan melihat beberapa pesan telemetri masuk dan detak jantung sesekali (lihat Gambar 4.24).



Gambar 4.24. Detak jantung dan telemetri tiba di IoT Hub

Deploy Aplikasi Perangkat Cerdas Anda Ke SBC

Sekarang kita memiliki aplikasi simulator Perangkat Cerdas yang berfungsi, kita dapat menerapkannya ke SBC. Dalam latihan ini, Anda akan memperbarui aplikasi Perangkat Cerdas Anda untuk menggunakan sensor suhu yang terpasang pada Qualcomm DragonBoard. Perhatikan bahwa latihan ini bukan merupakan persyaratan untuk menjalani latihan lain dalam buku ini. Di sini disediakan untuk kelengkapan.

Persyaratan untuk latihan ini adalah sebagai berikut:

- Papan Naga 410c
- Kit Pemula 96Boards Linker
- Utilitas Pembaruan DragonBoard untuk Windows 10 IoT
- OS Windows 10 IoT Core untuk DragonBoard
- tetikus USB
- papan ketik USB
- Layar dan kabel HDMI
- Kabel Mini-USB ke USB

DragonBoard 410c adalah komputer papan tunggal yang dibuat dengan prosesor seri Qualcomm® Snapdragon™ 400. Ini memiliki kekuatan pemrosesan tingkat lanjut, Wi-Fi, Bluetooth, dan GPS, semuanya dikemas dalam papan seukuran kartu kredit. Berdasarkan prosesor Snapdragon 410E berkemampuan 64-bit, DragonBoard 410c dirancang untuk mendukung pengembangan perangkat lunak berbasis Linux atau Windows (lihat Gambar 4-25).



Gambar 4-25. Ikhtisar DragonBoard 410c

1. Siapkan perangkat keras Anda seperti yang digambarkan pada Gambar 4-26. DragonBoard terhubung ke layar HDMI dan ke laptop pengembangan Anda menggunakan kabel mini-USB. Ada juga mouse USB dan keyboard USB yang terpasang pada DragonBoard.

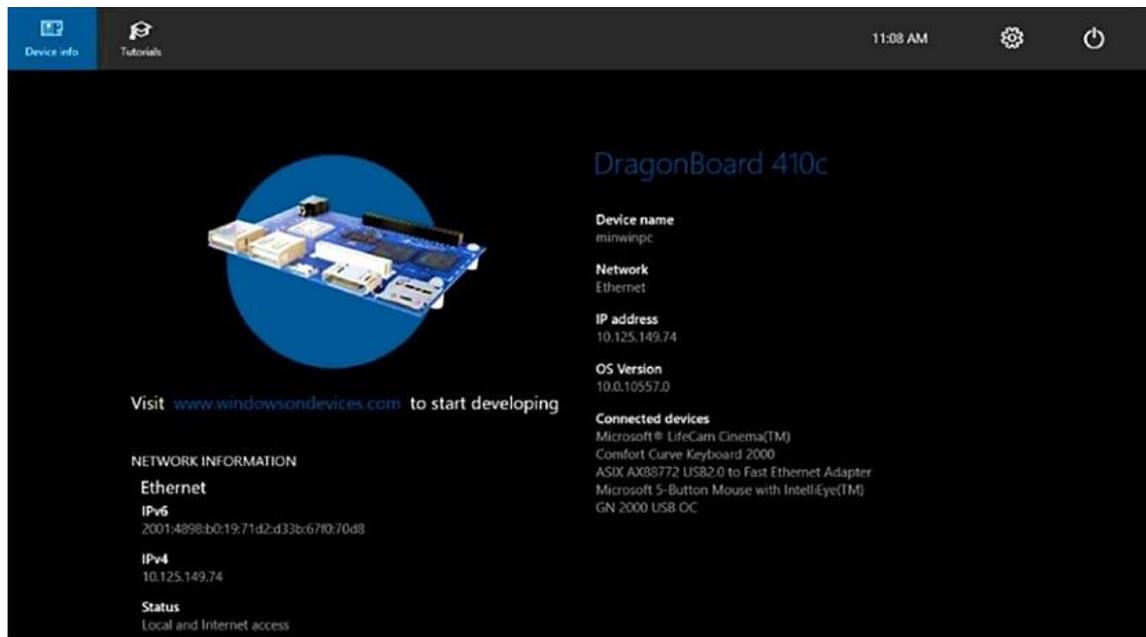


Gambar 4-26. Pengaturan DragonBoard

2. Flash DragonBoard dengan sistem operasi Windows 10 Core IoT.
 - Colokkan adaptor HDMI, keyboard USB, dan mouse ke DragonBoard.
 - Gunakan adaptor mini-USB ke USB untuk menghubungkan board ke PC Anda.

Setelah perangkat di-flash dan terhubung ke jaringan nirkabel, Anda dapat mengelola perangkat menggunakan portal Perangkat Windows. Portal Perangkat Windows adalah aplikasi yang merupakan bagian dari Windows 10 IoT Core dan berjalan secara lokal di DragonBoard. Aplikasi ini dapat diakses dari PC Anda dengan menggunakan browser dan menavigasi ke halaman beranda aplikasi yang berjalan di perangkat. Anda memerlukan alamat

IP perangkat Anda di jaringan Anda. Layar beranda DragonBoard akan menampilkan alamat IP perangkat, seperti yang ditunjukkan pada Gambar 4-27.



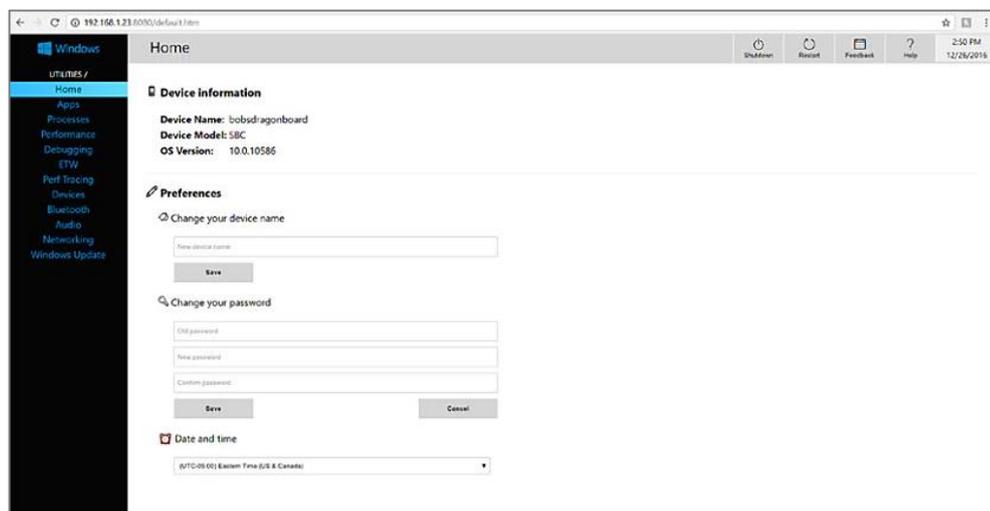
Gambar 4-27. Layar beranda DragonBoard

3. Menggunakan browser, navigasikan ke `http://[deviceIP]:8080` dan masuk ke portal perangkat menggunakan nama pengguna dan kata sandi default:

Username: Administrator

Password: p@ssword

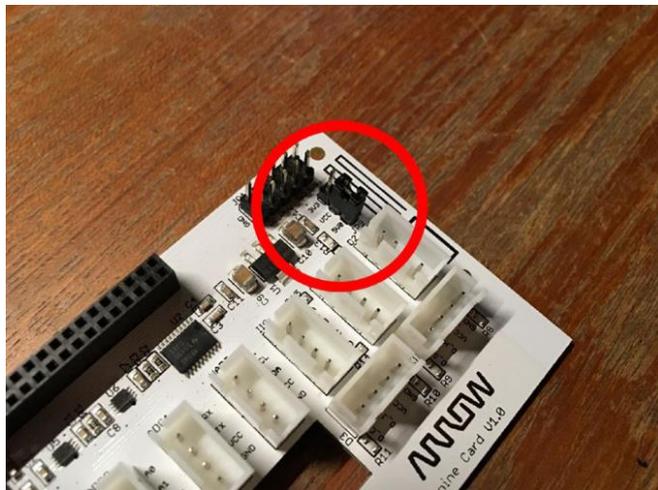
Dengan menggunakan aplikasi ini, Anda dapat mengatur nama perangkat, kata sandi administrator, dan banyak fitur perangkat lainnya dengan menavigasi menu di sisi kiri antarmuka pengguna (lihat Gambar 4-28).



Gambar 4.28. Portal perangkat Windows

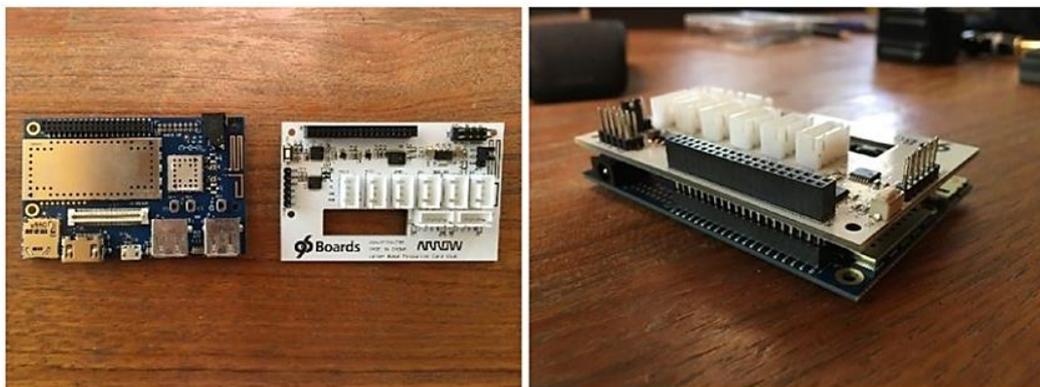
Semua sensor kit Linker dapat ditambahkan ke DragonBoard melalui kartu mezzanine Linker. Kartu mezzanine Linker memiliki delapan konektor yang mendukung Analog, UART, I2C, dan GPIO, serta dua saluran input analog menggunakan chip MCP3004 ADC. Ada penerjemah tingkat tegangan dua arah, yang memungkinkan terjemahan dua arah tegangan rendah antara node tegangan 1,2-V, 1,5-V, 1,8-V, 2,5-V, 3,3-V, dan 5-V. Ini kompatibel dengan modul 3.3V atau 5V dan memudahkan penyambungan periferal. Anda dapat memilih voltase yang sesuai dengan memasang jumper pada pemilih voltase JP9. Kami akan menggunakan mezzanine dalam mode 5-V karena itulah yang dibutuhkan oleh sensor yang akan kami pasang.

4. Matikan perangkat dan lepaskan keyboard dan mouse. Atur tegangan menjadi 5V menggunakan jumper JP9, seperti yang ditunjukkan pada Gambar 4-29.



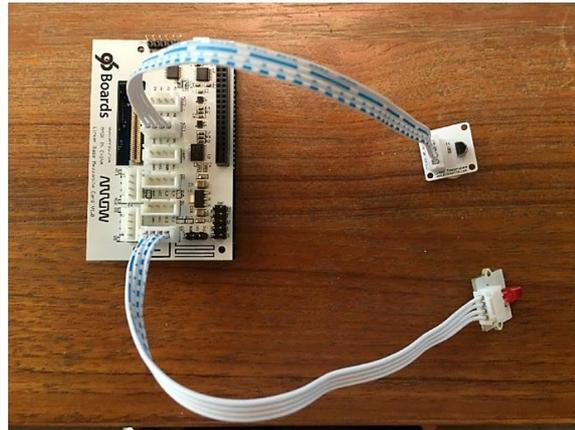
Gambar 4.29. Atur jumper tegangan ke 5V

5. Tumpuk papan mezzanine di atas DragonBoard dan masukkan secara hati-hati konektor I/O serbaguna (GPIO) ke dalam konektor GPIO pada DragonBoard (lihat Gambar 4-30).



Gambar 4.30. Sebelum dan sesudah penumpukan mezzanine

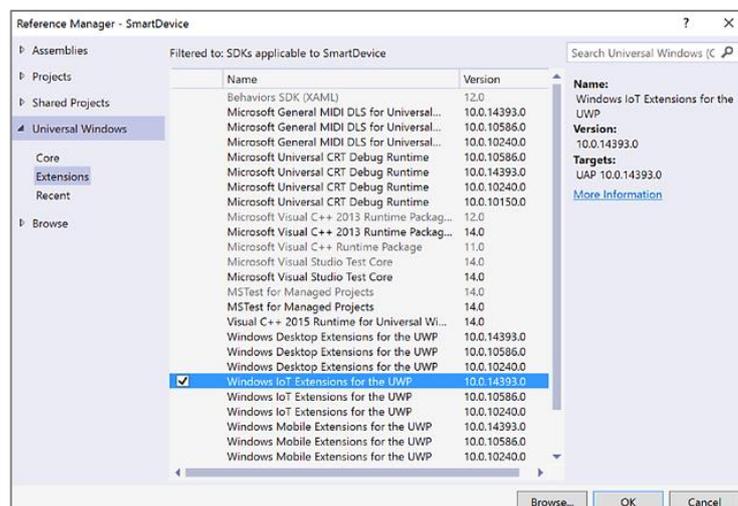
6. Pasang sensor LED merah dan sensor termal seperti yang ditunjukkan pada Gambar 4-31.
 - Dengan menggunakan kabel adaptor, pasang salah satu ujung ke modul LED merah dan ujung lainnya ke konektor berlabel DC2 di mezanin Linker.
 - Dengan menggunakan kabel adaptor kedua, pasang salah satu ujung ke sensor termal dan ujung lainnya ke konektor berlabel ADC1 di mezanin Linker.



Gambar 4-31. Sensor terpasang

7. Pasang keyboard, mouse, dan monitor HDMI ke kabel daya. Anda sekarang siap untuk menguji konfigurasi perangkat keras lengkap.
8. Tambahkan referensi ke Windows IoT Extensions for UWP ke solusi Perangkat Cerdas Anda. Perpustakaan ini akan menyediakan kelas tingkat tinggi untuk bekerja dengan komputer papan tunggal.

Klik kanan pada node Referensi di Solution Explorer dan pilih Tambahkan Referensi. Pilih Universal Windows ► Extensions di menu sebelah kiri. Centang kotak di sebelah Windows IoT Extensions for the UWP for version 10.0.14393 lalu klik OK (lihat Gambar 4-32).



Gambar 4.32. Tambahkan referensi ke ekstensi Windows IoT

9. Tambahkan pernyataan penggunaan berikut di bagian atas file `MainPage.xaml.cs`:

```
using Windows.Devices.Gpio;
using Windows.Devices.Spi;
using Windows.Devices.Enumeration;
```

10. Tambahkan variabel `private` berikut ke kelas `MainPage`. Ini digunakan untuk menghubungkan dan berkomunikasi dengan antarmuka I/O tujuan umum (GPIO) dan bus antarmuka perifer serial (SPI).

```
// GPIO
private static GpioPinValue _value;
private const int LedPin = 13;
private static GpioPin _led;

// SPI
private static byte[] _readBuffer =
    new byte[3] { 0x00, 0x00, 0x00 };
private static byte[] _writeBuffer =
    new byte[3] { 0x01, 0x80, 0x00 };
private const string SpiControllerName = "SPI0";
private const int SpiChipSelectLine = 0;
private static SpiDevice _spiDisplay;
```

11. Tambahkan metode `private static` untuk menginisialisasi GPIO. Antarmuka GPIO akan digunakan untuk menyalakan dan mematikan LED merah.

```
private static void InitGPIO()
{
    _led = GpioController.Default.OpenPin(LedPin);
    _led.Write(GpioPinValue.Low);
    _led.SetDriveMode(GpioPinDriveMode.Output);
}
```

12. Tambahkan metode `private static` asinkron untuk menghidupkan dan mematikan LED. Hal ini dilakukan dengan mengatur nilai lokasi pin LED ke tinggi (hidup), tidur selama beberapa milidetik, dan kemudian mengatur pin ke rendah (mati). Hal ini akan menyebabkan lampu LED berkedip dan kemudian mati.

```
public static async void BlinkLED(int duration)
{
    _led.Write(GpioPinValue.High);
    await Task.Delay(duration);
    _led.Write(GpioPinValue.Low);
}
```

13. Tambahkan metode `private static` untuk menginisialisasi SPI. SPI akan digunakan untuk berkomunikasi dengan sensor suhu menggunakan konverter analog-ke-digital pada papan mezzanine.

```
private static async void InitSPI()
{
    try
    {
        var settings = new SpiConnectionSettings(SpiChipSelectLine)
        {
            ClockFrequency = 500000,
            Mode = SpiMode.Mode0
        };

        var spiAqs = SpiDevice.GetDeviceSelector(
            SpiControllerName);
        var deviceInfo = await
            DeviceInformation.FindAllAsync(spiAqs);

        _spiDisplay = await SpiDevice.FromIdAsync(
            deviceInfo[0].Id, settings);
    }
    catch (Exception ex)
    {
        throw new Exception("SPI Initialization Failed", ex);
    }
}
```

14. Tambahkan metode pembantu `private static` untuk mengubah array byte yang dikembalikan dari sensor menjadi nilai floating-point.

```
public static double ConvertToDouble(byte[] data)
{
    int result = 0;
    int i = Convert.ToInt32("1100000000", 2);
    result = (data[1] << 8) & i;
    result |= (data[2] & 0xff);
    return result;
}
```

15. Perbarui metode `StartTelemetry()` agar menggunakan antarmuka SPI untuk membaca sensor suhu dan mengaturnya sebagai pembacaan suhu dari pesan sensor sederhana. Ini menggantikan penggunaan generator nomor acak.

```
// get the temperature setting from the sensor
_spiDisplay.TransferFullDuplex(
```

```

    _writeBuffer, _readBuffer);

// convert the value to Fahrenheit
var temperature = ConvertToDouble(_readBuffer);
temperature = ((temperature * 5.0) / (1023 - 0.5)) * 100);

var sensorReading = new SimpleSensorReading
{
    Longitude = _deviceManifest.Longitude,
    Latitude = _deviceManifest.Latitude,
    DeviceId = _deviceManifest.SerialNumber,
    Reading = temperature
};

```

16. Tambahkan panggilan ke metode `BlinkLED()` tepat setelah mengirim pesan suhu ke IoT Hub. Berikan nilai 500ms selama durasi kedipan.

```

await _deviceClient.SendEventAsync(message);
BlinkLED(500);

```

17. Perbarui awal metode `MainPage_OnLoaded()` untuk memanggil metode `GPIOInit()` dan `SPIInit()`.

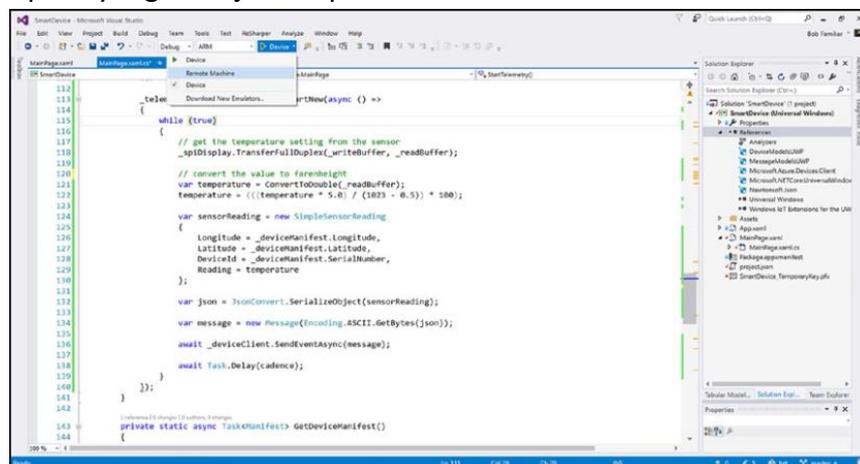
```

Status.Text = "Main Page Loaded";
InitGPIO();
Status.Text = "GPIO Initialized";
InitSPI();
Status.Text = "SPI Initialized";

```

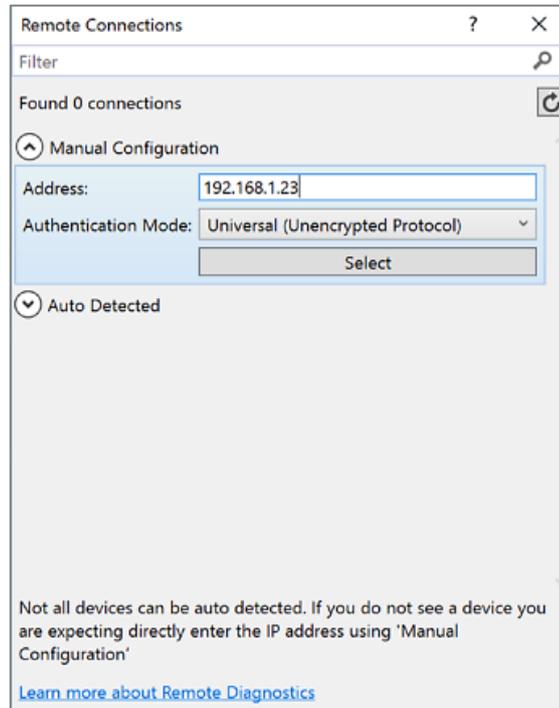
#Kami sekarang siap untuk menguji perangkat pintar kami.

18. Untuk menyebarkan aplikasi ke perangkat, atur Build Target ke ARM lalu pilih Remote Machine, seperti yang ditunjukkan pada Gambar 4-33.



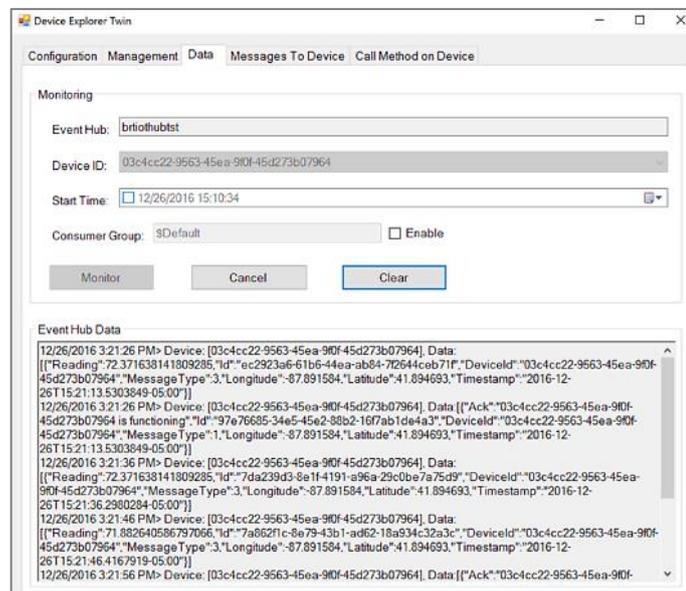
Gambar 4-33. Konfigurasi Visual Studio untuk membangun dan menyebarkan solusi ke perangkat jarak jauh

Pertama kali Anda melakukan ini Visual Studio akan meminta alamat IP perangkat. Berikan alamat IP dari layar beranda DragonBoard, seperti yang ditunjukkan pada Gambar 4-34, dan klik Pilih.



Gambar 4-34. Mengatur alamat IP SBC Anda di Visual Studio

19. Jalankan aplikasi menggunakan debugger Visual Studio. Aplikasi akan dibangun dan disebar ke DragonBoard Anda. Anda akan melihat UI pada layar HDMI dan Anda dapat melacak pesan detak jantung dan suhu yang masuk pada PC Anda menggunakan utilitas Device Explorer, seperti yang ditunjukkan pada Gambar 4-35.



Gambar 4.35. DragonBoard mengirimkan pesan sensor detak jantung dan suhu ke IoT Hub

4.4 MODIFIKASI SIMULATOR TIM

Penerapan referensi Bisnis Real-Time adalah solusi multi-penyewa. Ini menyediakan serangkaian layanan platform umum yang dapat digunakan oleh banyak pelanggan pada saat yang sama, menyediakan akses aman ke data untuk setiap pelanggan. Dalam skenario kami, setiap perusahaan telah menetapkan kelompok karyawan yang disebut tim, di mana setiap anggota tim terhubung ke cloud menggunakan rompi yang dilengkapi sensor. Perusahaan dapat memantau pembacaan biometrik dari masing-masing karyawan, yang kemudian dialihkan ke layanan analitik tingkat lanjut untuk mengidentifikasi tingkat kelelahan dan stres serta memberikan visualisasi data, peringatan, dan pemberitahuan secara real-time.

Implementasi referensi menyediakan tiga aplikasi konsol untuk mensimulasikan tim. Setiap aplikasi mensimulasikan tim terhubung yang terdiri dari 15 karyawan dari perusahaan palsu berikut:

- *WigiTech*: Perusahaan teknologi berbasis di Massachusetts yang mengaktifkan sensor pada pekerja di pabriknya yang bekerja di dalam dan di sekitar mesin berbahaya.
- *Tall Towers*: Sebuah perusahaan layanan utilitas yang memungkinkan para insinyur lapangan memantau kelelahan dan stres saat mereka memanjat menara di puncak gedung pencakar langit di pusat kota New York.
- *The Complexed Badger*: Sebuah perusahaan angkutan truk yang berbasis di Chicago yang mengkhususkan diri dalam memindahkan kargo besar dan berbahaya dan memantau kewaspadaan pengemudinya.

Implementasi referensi mencakup layanan mikro yang disebut Simulasi yang menyediakan data untuk mendorong analitik. Data sampel terdiri dari 4200 catatan pembacaan biometrik (berisi beberapa pembacaan sensor biometrik) untuk masing-masing 15 individu sehingga menghasilkan total 63 ribu catatan data. Setiap aplikasi simulator menggunakan kumpulan data yang sama. Oleh karena itu, jika Anda menjalankan ketiga simulator secara bersamaan, Anda akan mensimulasikan total 45 karyawan dari tiga perusahaan berbeda.

Setiap simulator saat startup menggunakan Registry API dan Device API untuk mengambil profil dan manifes perangkat untuk setiap rekan satu tim. Selanjutnya dimulai thread latar belakang untuk setiap rekan satu tim. Dalam thread tersebut, ia mengambil rekaman data simulasi dan, menggunakan profil pengguna, manifes, dan rekaman data simulasi, membuat pesan dan mengirimkannya ke Azure IoT Hub. Bidang dalam pesan diuraikan dalam cuplikan kode berikut.

```

UserId = teammate.Profile.id,
DeviceId = teammate.Manifest.SerialNumber,
Longitude = teammate.Manifest.Longitude,
Latitude = teammate.Manifest.Latitude,
Status = SensorStatus.Normal,
Timestamp = DateTime.Now,
Age = teammate.Profile.biometrics.age,
Weight = teammate.Profile.biometrics.weight,
Height = teammate.Profile.biometrics.height,
BreathingRate = datarow.columns[0].dataValue,

```

```

Ventilation = datarow.columns[1].dataValue,
Activity = datarow.columns[2].dataValue,
HeartRateBPM = datarow.columns[3].dataValue,
Cadence = datarow.columns[4].dataValue,
Velocity = datarow.columns[5].dataValue,
Speed = datarow.columns[6].dataValue,
HIB = datarow.columns[7].dataValue,
HeartrateRedZone = datarow.columns[8].dataValue,
HeartrateVariability = datarow.columns[9].dataValue,
Temperature = datarow.columns[10].dataValue

```

Untuk menggunakan simulator, Anda perlu membuat beberapa pembaruan kecil pada masing-masing dari tiga aplikasi simulator sehingga aplikasi tersebut menggunakan API terkelola yang Anda terapkan menggunakan latihan di Bab 2. Simulator tim ada di folder perangkat/perangkat-teamsim. Di sana Anda akan menemukan simulator untuk setiap perusahaan semu.

1. Navigasikan ke folder `devices/device-teamsim/wigitech` dan buka solusi WigiTechSim.
2. Buka file `App.config` dan perbarui pengaturan konfigurasi berikut:

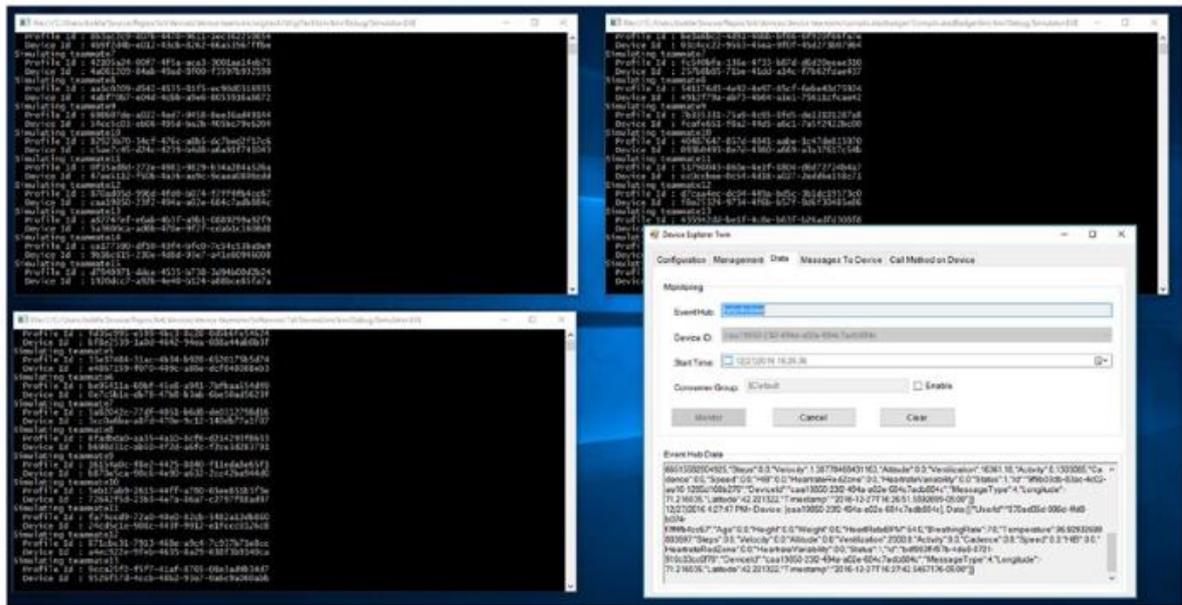
```
<add key="DeviceAPI" value="https://[your-apim-host].azure-api.net/
dev/v1/device/manifests" />
```

```
<add key="RegistryAPI" value="https://[your-apim-host].azure-api.net/
dev/v1/registry/profiles" />
```

```
<add key="SimulationAPI" value="https://[your-apim-host].azure-
api.net/dev/v1/simulation/datasets" />
```

```
<add key="SubscriptionKey" value="subscription-key=[your-dev-key]" />
```

3. Lakukan ini juga untuk setiap aplikasi simulator lainnya.
4. Jalankan Device Explorer dan masing-masing dari tiga aplikasi simulator. Setelah perangkat tersambung ke IoT Hub dan mengirimkan data, Anda dapat menggunakan Device Explorer untuk memeriksa pesan yang datang dari perangkat yang terkait dengan masing-masing simulator (lihat Gambar 4-36).



Gambar 4-36. Tiga simulator dan Device Explorer

Setiap nomor seri perangkat ditampilkan di jendela konsol sehingga Anda dapat dengan mudah menghubungkannya ke Device Explorer.

4.5 RINGKASAN

Bab ini membahas dunia sensor dan perangkat yang kaya dan menakjubkan serta berbagai cara konfigurasi yang berbeda dapat terhubung ke Azure IoT Hub. Ini memeriksa modem GSM, perangkat pintar komputer papan tunggal, dan gateway edge. Anda telah mempelajari tentang Protocol Gateway SDK dan cara menggunakannya untuk menyediakan lapisan terjemahan antara perangkat lama dan layanan cloud baru. Anda telah memeriksa arsitektur edge gateway dan IoT Gateway SDK serta cara menggunakan perangkat ini untuk menyediakan agregasi, pemfilteran, dan analitik di edge.

Melalui latihan ini, Anda mempelajari cara mengimplementasikan perangkat lunak mikrokontroler yang berjalan di perangkat jarak jauh Anda, membaca sensor, dan mengirim pesan ke IoT Hub. Terakhir, Anda mengonfigurasi simulator tim sehingga Anda dapat mendorong serangkaian analisis lanjutan yang kaya fitur yang disediakan oleh implementasi referensi. Rangkaian bab berikutnya membahas analisis tingkat lanjut tersebut dan memanfaatkan data yang berasal dari simulator. Mereka mencakup analisis aliran, pabrik data, data lake, dan Pembelajaran Mesin.

BAB 5

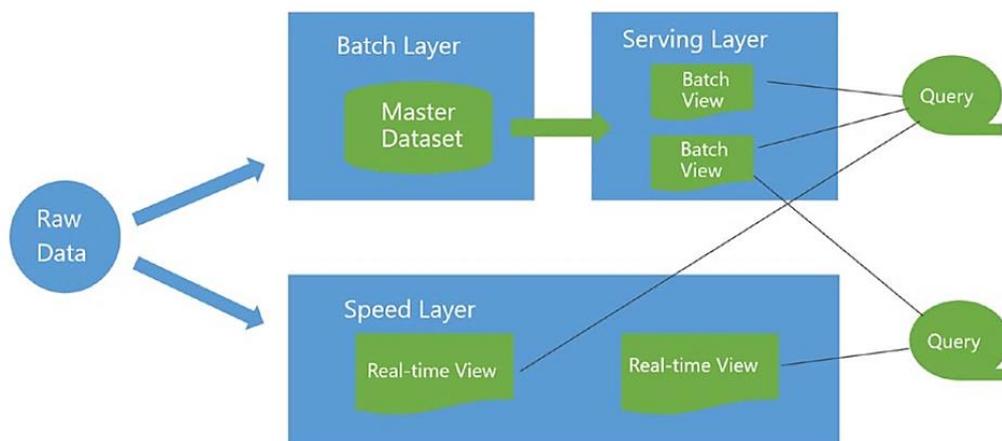
PEMROSESAN REAL-TIME MENGGUNAKAN AZURE STREAM ANALYTICS

Bab ini membahas penggunaan Microsoft Azure Streaming Analytics untuk menciptakan tugas memproses aliran data masuk dari berbagai sensor, melakukan transformasi dan pengayaan data, dan terakhir, memberikan hasil keluaran ke dalam berbagai format data. Cloud dikatakan mewakili transformasi teknologi yang terjadi sekali dalam satu generasi. Tentu saja, salah satu landasan transformasi utama ini adalah kemampuan untuk secara efisien menyerap, memproses, dan melaporkan data dalam jumlah besar dalam skala besar.

5.1 ARSITEKTUR LAMBDA

Dalam analisis data modern saat ini, strategi pemrosesan aliran baru telah diusulkan arsitektur “Lambda” dan secara luas dikaitkan dengan Nathan Marz, pencipta Apache Storm. Esensi mendasar dari arsitektur Lambda adalah arsitektur ini dirancang untuk menyerap data masuk dalam jumlah besar dengan memanfaatkan metodologi pemrosesan batch dan streaming. Atribut tambahan arsitektur Lambda mencakup hal berikut:

- Kemampuan untuk memproses beragam beban kerja dan skenario.
- Throughput tinggi ditandai dengan pembacaan latensi rendah dengan penulisan dan pembaruan yang sering.
- Menyimpan data yang masuk dalam format aslinya. Ini adalah gagasan tentang “danau data”.
- Memodelkan transformasi data sebagai rangkaian tahapan yang terwujud dari data asli.
- Infrastruktur yang sangat terukur, hampir linier, dan dapat ditingkatkan skalanya untuk meningkatkan/menurunkan skala. Gambar 5-1 menggambarkan arsitektur Lambda.

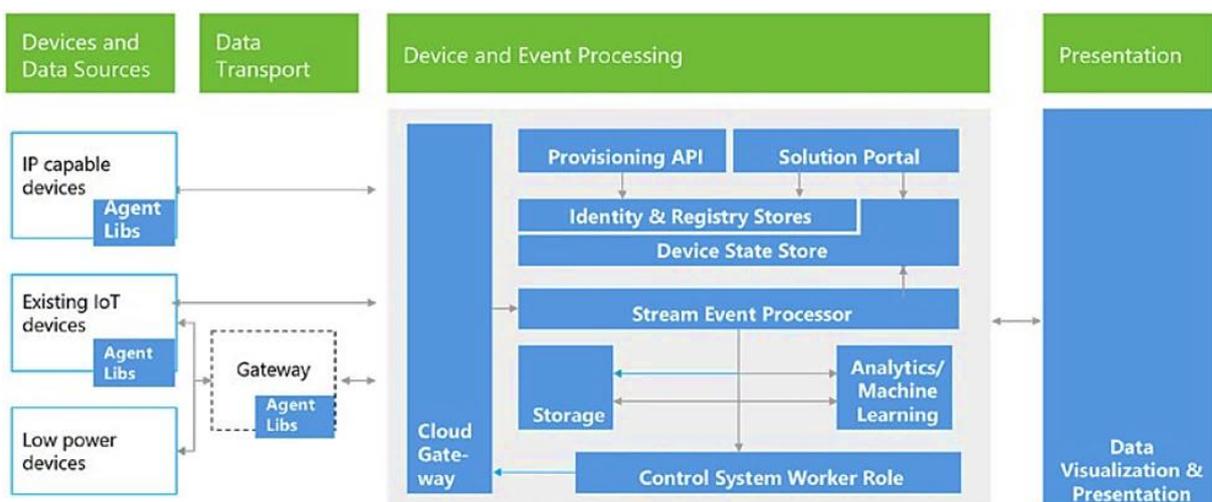


Gambar 5.1. Arsitektur Lambda

Dengan menggabungkan pemrosesan batch dan aliran dalam arsitektur yang sama, hasilnya adalah mesin analisis data yang dioptimalkan yang tidak hanya mampu memproses data, namun juga mengirimkan data yang tepat pada waktu yang tepat. Perlu dicatat bahwa data just-in-time (JIT) baru yang kini dapat ditampilkan dengan analitik streaming (melalui arsitektur Lambda) sering kali dapat menjadi sumber berbagai jenis keunggulan kompetitif bagi bisnis atau perusahaan yang mengetahui caranya. Mengeksploitasi jenis informasi ini di beragam kasus penggunaan.

Keajaiban sebenarnya datang dari mengetahui cara mengenali peluang tersembunyi yang terkubur jauh di dalam data dan dari sana mengambil tindakan untuk mengeksplorasi, mengembangkan, mengatasi kegagalan dengan cepat, dan akhirnya berhasil dalam membangkitkan kebenaran, perubahan transformasional dalam bisnis atau industri. Ini pada dasarnya adalah inti dari buku ini dan menjadi pedoman bagi penerapan referensi kita sendiri.

Microsoft telah menyadari kebutuhan akan analisis streaming dalam skala besar sebagai bagian dari solusi Internet-of-Thing (IoT) modern saat ini dan telah memasukkan kemampuan streaming ke dalam beberapa arsitektur populer, seperti Cortana Analytics serta penawaran IoT Suite yang ditunjukkan pada Gambar 5-2.



Gambar 5.2. Arsitektur rangkaian Azure IoT

Apa itu Analisis Streaming?

Salah satu keuntungan nyata dari mesin analisis streaming yang hebat adalah kemampuan untuk menyediakan analisis dan keluaran waktu nyata sehingga data dapat ditindaklanjuti dengan penundaan minimum. Untuk memberikan konteks terhadap tantangan yang terlibat dalam pengembangan solusi analisis streaming, contoh kanonis terbaik adalah skenario penghitungan jumlah mobil merah di tempat parkir versus penghitungan jumlah mobil merah yang melewati jalan bebas hambatan utama (dengan asumsi saat itu bukan jam sibuk) untuk setiap interval 10 menit dalam periode satu jam (lihat Gambar 5-3).



Gambar 5.3. Tantangan analisis streaming

Inti dari skenario ini adalah “data bergerak” versus “data diam”. Perbedaan utama di sini adalah elemen “waktu” dan kemampuan untuk menangkap dan menganalisis “potongan” data secara berkala di jutaan peristiwa yang berpotensi terjadi untuk mendeteksi pola dan anomali dalam data streaming dalam jumlah besar.

5.2 ANALISIS REAL-TIME

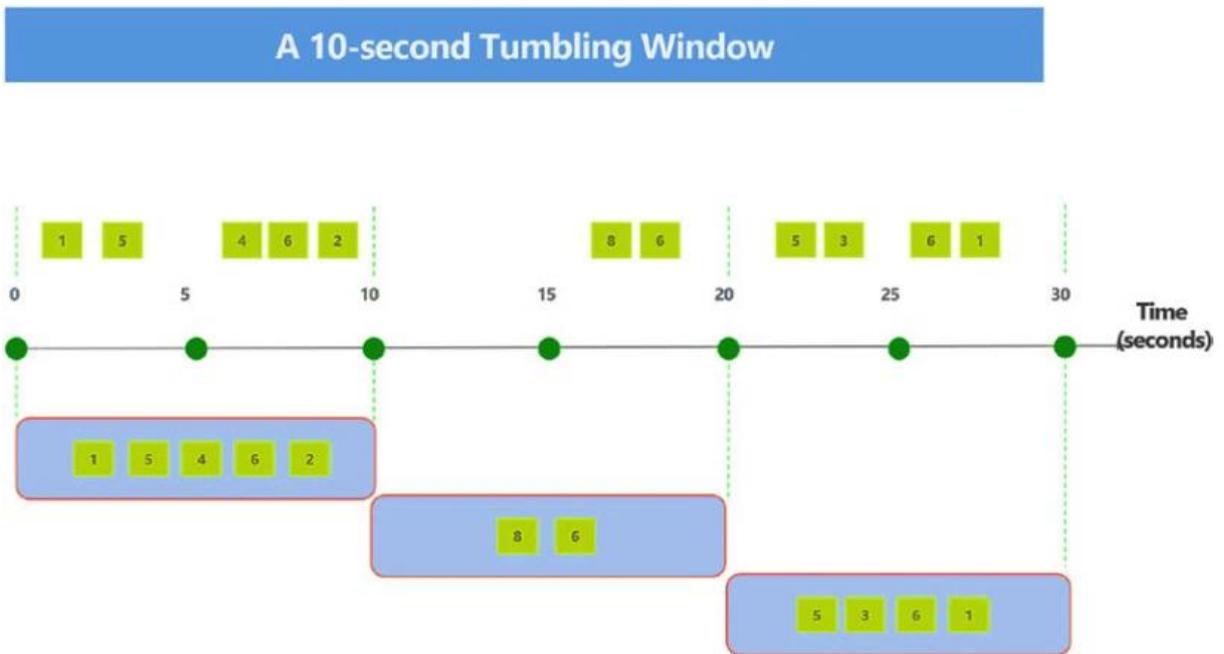
Analitik real-time adalah tentang kemampuan memproses data yang berasal dari jutaan perangkat atau aplikasi yang terhubung, dengan kemampuan bawaan untuk menyerap dan memproses jutaan peristiwa per detik. Komponen utama dari skenario ini adalah integrasi dengan pola terbitkan/berlangganan yang sangat skalabel. Persyaratan utama lainnya adalah penyederhanaan kemampuan pemrosesan pada aliran data berkelanjutan yang memungkinkan solusi untuk mentransformasi, menambah, mengkorelasikan, dan melakukan operasi temporal (berbasis waktu).

Mengkorelasikan data streaming dengan data referensi juga merupakan persyaratan inti dalam banyak kasus, karena data yang masuk sering kali perlu dicocokkan dengan data host yang sesuai.

Implementasi Streaming dan Analisis Rangkaian Waktu

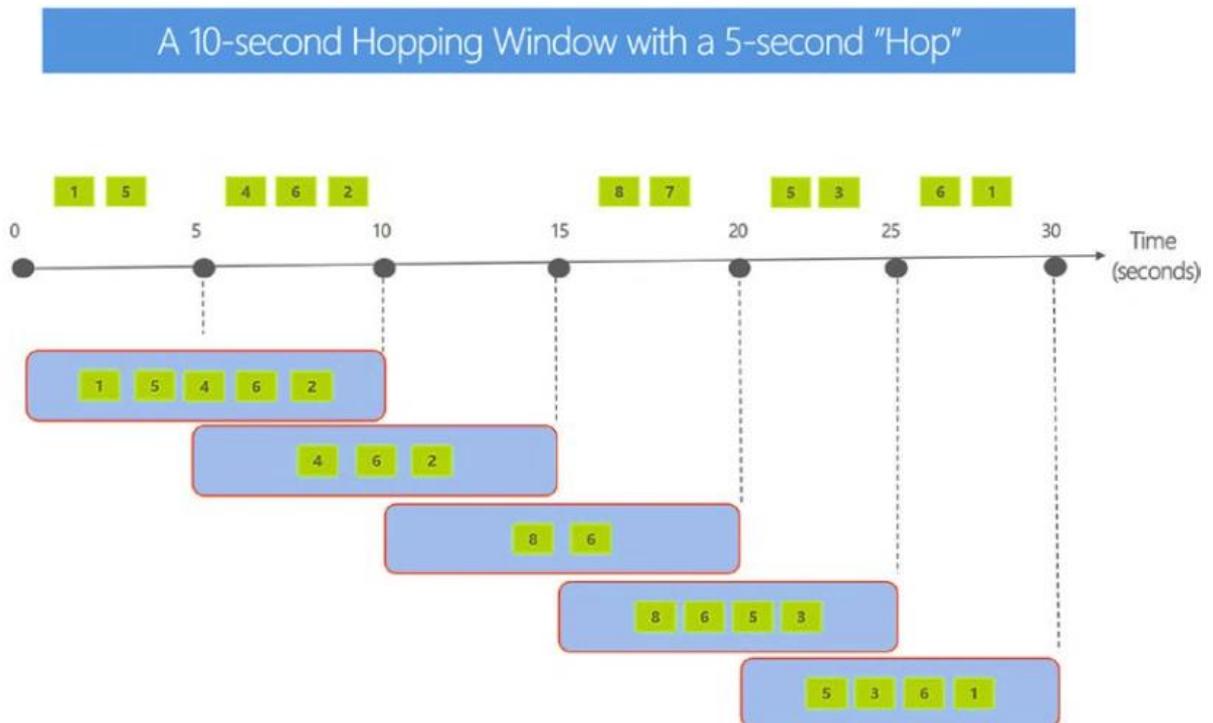
Bisnis modern saat ini menuntut analisis secara real-time untuk mendapatkan keunggulan kompetitif; mereka telah beralih dari metode “jadul” yaitu ritme pelaporan per jam, harian, mingguan, dan bulanan, kini mengandalkan data yang hanya berumur beberapa detik. Untuk menjadikan analisis dan pelaporan streaming menjadi lebih relevan, dukungan untuk perhitungan jangka waktu menjadi lebih penting. Untuk tujuan ini, serangkaian jendela deret waktu bermanfaat seperti:

- *Tumbling Windows*: Serangkaian interval waktu berukuran tetap, tidak tumpang tindih, dan berdekatan. Diagram pada Gambar 5-4 mengilustrasikan aliran dengan serangkaian peristiwa dan bagaimana peristiwa tersebut dipetakan ke dalam jendela tumbling sepuluh detik.
- *Hopping Windows*: Model jendela yang dijadwalkan tumpang tindih. Spesifikasi jendela hopping terdiri dari parameter berikut:
 - Satuan waktu.
 - Ukuran jendela, berapa lama setiap jendela bertahan
 - Ukuran hop, seberapa besar setiap jendela bergerak maju relatif terhadap jendela sebelumnya
 - <Opsional> Ukuran offset, parameter opsional keempat.



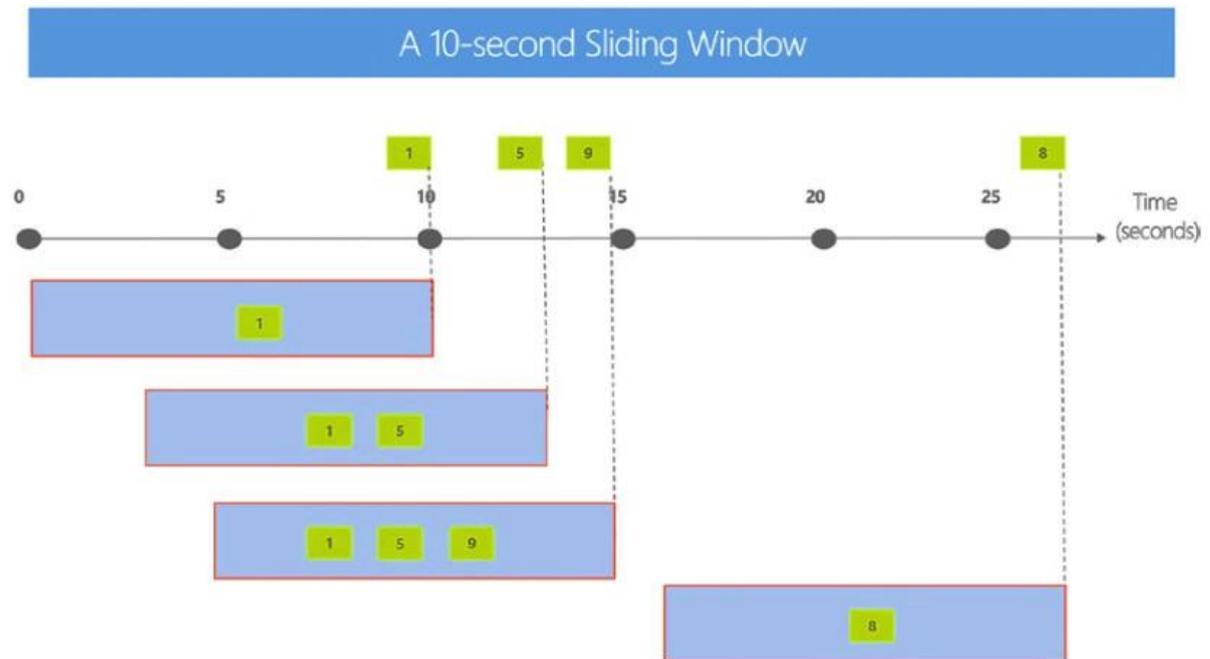
Gambar 5-4. Contoh jendela berjatuhan dalam analisis streaming

Ilustrasi pada Gambar 5-5 menunjukkan aliran dengan serangkaian peristiwa. Setiap kotak mewakili jendela lompatan dan kejadian yang dihitung sebagai bagian dari jendela tersebut, dengan asumsi lompatannya adalah 5, dan ukurannya adalah 10.



Gambar 5-5. Contoh melompati jendela dalam analisis streaming

- *Jendela Geser*: Saat menggunakan jendela geser, sistem diminta untuk secara logis mempertimbangkan semua kemungkinan jendela dengan panjang tertentu dan menampilkan peristiwa hanya pada titik waktu ketika konten jendela benar-benar berubah, dengan kata lain ketika suatu peristiwa masuk atau ada jendela. Gambar 5-6 mengilustrasikan jendela geser.



Gambar 5-6. Contoh jendela geser dalam analisis streaming

5.3 MEMREDIKSI HASIL UNTUK KEUNGGULAN KOMPETITIF

Selain mendeteksi pola dan anomali dalam data, elemen penting lainnya dalam menjalankan bisnis dengan kecepatan Internet adalah kemampuan untuk membentuk hasil bisnis baru dengan memprediksi apa yang mungkin terjadi di masa depan. Ini adalah domain permasalahan untuk analitik prediktif dan Pembelajaran Mesin, yang dapat menggunakan data historis yang dikombinasikan dengan algoritme ilmu data modern untuk memprediksi hasil di masa depan.

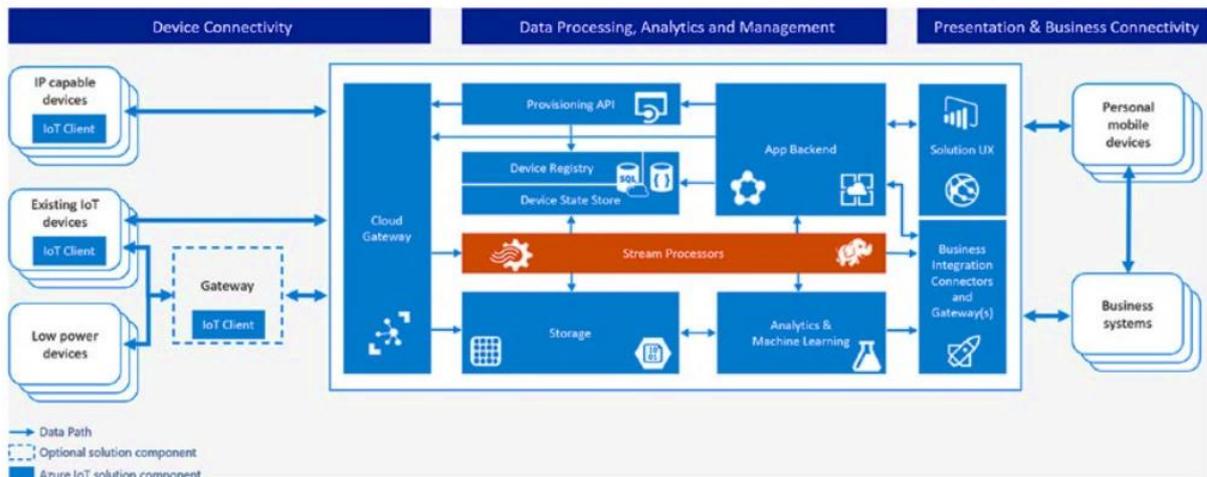
Salah satu metode utama untuk mempercepat bisnis dan membangun keunggulan kompetitif adalah kemampuan untuk mengotomatisasi, melengkapi, atau mempercepat proses pengambilan keputusan bisnis utama melalui penggunaan analisis prediktif dan Pembelajaran Mesin. Dalam dunia bisnis saat ini, keputusan dibuat setiap hari, seringkali tanpa semua fakta dan data, sehingga wawasan tambahan apa pun sering kali dapat menghasilkan keunggulan kompetitif yang sangat besar.

Di banyak industri dan sektor vertikal, banyak hambatan yang ditemukan saat ini karena kurangnya data yang dapat ditindaklanjuti. Ini adalah area utama di mana Azure Stream Analytics dan Machine Learning dapat membantu mengurangi hambatan dan mempercepat hasil. Fitur baru terbaru dari Azure Streaming Analytics adalah kemampuan untuk secara langsung memanggil Layanan Web Azure Machine Learning saat data streaming

diproses untuk memperkaya aliran data masuk atau memprediksi hasil yang mungkin memerlukan pemicuan pemberitahuan atau peringatan. Ini tentunya merupakan fitur dan kemampuan utama, jadi kami akan membahasnya secara mendetail di Bab 8.

Pemrosesan Aliran: Opsi Implementasi di Azure

Saat menggunakan Microsoft Azure, ada beberapa pilihan yang tersedia untuk mengimplementasikan lapisan Stream Processing, seperti yang diilustrasikan pada Gambar 5-7 dengan arsitektur IoT.



Gambar 5-7. Lapisan pemrosesan aliran dalam arsitektur IoT

Dengan Microsoft Azure, ada banyak opsi yang tersedia untuk membuat lapisan pemrosesan uap. Pilihannya berkisar dari menggunakan paket Perangkat Lunak Sumber Terbuka (OSS) di Mesin Virtual Linux hingga memanfaatkan Platform-as-a-Service yang terkelola sepenuhnya, seperti Azure HDInsight.

Berikut adalah opsi dasar untuk menjalankan mesin pemrosesan aliran di Azure:

Opsi Streaming: Mesin Virtual - (Infrastruktur sebagai Layanan):

- **Mesin virtual** (menjalankan Windows atau Linux)
 - ❖ Distribusi Perangkat Lunak Sumber Terbuka:
 - ❖ Pekerjaan Horton
 - ❖ Cloudera
 - ❖ Buat sendiri: Apache Storm/Spark, Apache Samza, Twitter Heron, Kafka Streams, Apache Flink, Apache Beam (alur kerja pemrosesan data), Apache Mesos (banyak sekali proyek).
 - ❖ Perhatikan bahwa opsi ini juga dapat digunakan untuk aplikasi pemrosesan aliran lokal.

Opsi Layanan Terkelola Azure (Platform sebagai Layanan – PaaS):

- **Azure HDInsight:** Spark/Storm Terkelola
 - ❖ Pada dasarnya mengelola Hadoop yang 100% kompatibel di Azure.
 - ❖ HBase sebagai database transaksional NoSQL berbentuk kolom yang berjalan di blob Azure.
 - ❖ Apache Storm sebagai layanan streaming untuk pemrosesan hampir real-time.

- ❖ Dukungan Hadoop 2.4 untuk perolehan kueri 100x pada kueri Hive.
- ❖ Dukungan Mahout untuk Machine Learning dan Hadoop.
- ❖ Antarmuka pengguna grafis untuk kueri sarang.
- **Azure Streaming Analitik**
 - ❖ Memproses data real-time di Azure menggunakan bahasa SQL sederhana.
 - ❖ Mengonsumsi jutaan kejadian real-time dari IoT atau Event Hubs yang dikumpulkan dari perangkat, sensor, infrastruktur, dan aplikasi.
 - ❖ Melakukan analisis sensitif terhadap waktu menggunakan bahasa mirip SQL terhadap beberapa aliran waktu nyata dan data referensi.
 - ❖ Output ke penyimpanan persisten, dasbor, atau kembali ke perangkat.

5.4 MEMILIH MESIN ANALISIS STREAMING TERKELOLA DI AZURE

Dengan ketersediaan kemampuan Apache Storm dan Spark Streaming di HDInsight, serta Azure Streaming Analytics, Microsoft telah menyediakan beberapa opsi untuk teknologi kepemilikan dan sumber terbuka untuk menerapkan solusi analitik streaming. Perlu dicatat bahwa kedua platform analitik terkelola Azure memberikan manfaat dari solusi PaaS terkelola, namun ada beberapa kemampuan utama yang berbeda yang membedakan keduanya dan harus dipertimbangkan saat menentukan solusi arsitektur streaming akhir Anda. Pada akhirnya, pilihan akhir akan dipersempit menjadi beberapa pertimbangan utama, yang dijelaskan pada Tabel 5.1.

Tabel 5-1. Perbandingan Antara Azure Streaming Analytics dan HDInsight

Fitur	Analisis Aliran Azure	Badai Apache HDInsight
Format Data Masukan	Format masukan yang didukung adalah Avro, JSON, dan CSV.	Format apa pun dapat diterapkan melalui kode khusus.
Bahasa Kueri SQL	Dukungan bahasa SQL yang mudah digunakan tersedia dengan sintaksis yang sama seperti ANSI SQL.	Tidak, pengguna harus menulis kode dalam Java C# atau menggunakan Trident API
Operator Sementara	Agregat berjendela dan gabungan temporal didukung secara langsung.	Operator temporal harus diimplementasikan melalui kode khusus.
Ekstensibilitas Kode Kustom	Tersedia melalui fungsi yang ditentukan pengguna JavaScript.	Ya, tersedia untuk menulis kode khusus dalam C#, Java, atau bahasa lain yang didukung di Storm.
Dukungan untuk UDF (Fungsi Buatan Pengguna)	UDF dapat ditulis dalam JavaScript dan dipanggil sebagai bagian dari kueri pemrosesan aliran waktu nyata.	UDF dapat ditulis dalam C#, Java, atau bahasa pilihan Anda.
Harga	Stream Analytics diberi harga berdasarkan jumlah unit streaming yang diperlukan. Unit	Untuk Apache Storm di HDInsight, unit pembeliannya berbasis kluster, dan dikenakan biaya berdasarkan waktu

	adalah perpaduan komputasi, memori, dan throughput.	kluster berjalan, terlepas dari pekerjaan yang disebarakan.
Layanan Kontinuitas Bisnis / Ketersediaan Tinggi dengan SLA terjamin	SLA waktu aktif 99,9%. Pemulihan otomatis dari kegagalan. Pemulihan operator temporal penuh status sudah ada di dalamnya.	SLA waktu aktif 99,9% dari cluster Storm. Apache Storm adalah platform streaming yang toleran terhadap kesalahan. Tanggung jawab pelanggan untuk memastikan pekerjaan streaming mereka berjalan tanpa gangguan.
Data referensi	Data referensi tersedia dari Azure Blobs dengan ukuran maksimal cache pencarian dalam memori sebesar 100 MB. Penyegaran data referensi dikelola oleh layanan.	Tidak ada batasan ukuran data. Konektor tersedia untuk HBase, DocumentDB, SQL Server dan Azure. Konektor yang tidak didukung dapat diterapkan melalui kode khusus. Penyegaran data referensi harus ditangani dengan kode khusus.
Integrasi dengan Pembelajaran Mesin	Melalui konfigurasi model Azure Machine Learning yang diterbitkan sebagai fungsi selama pembuatan pekerjaan Azure Streaming Analytics.	Tersedia melalui Storm Bolts.

Perlu dicatat bahwa HDInsight mendukung Apache Storm dan Apache Spark Streaming. Masing-masing kerangka kerja ini menyediakan kemampuan streaming dan kemampuan ini mungkin layak untuk dianalisis lebih lanjut tergantung pada kebutuhan implementasi spesifik Anda. Berikut ini perbandingan tambahan dari tiga implementasi streaming terkelola (melalui HDI dan ASA) yang dimungkinkan di Microsoft Azure:

Feature	Storm on HDI	SparkStreaming on HDI	Azure Streaming Analytics
Programming Model	Java, C#	Scala, Python, Java	SQL Query Language
Delivery Guarantee	At-least-once (Exactly once w/ Trident)	Exactly Once	At least once
State Management	Ya	Ya	Ya
Processing Model	Event at-a-time	Micro-batching	Real-time event processing
Scaling	Manual	Manual	Automatic
Open Source	Yes	Yes	NA

Pilihan Teknologi Streaming: Pertimbangan Keputusan

Saat mengevaluasi platform analitik streaming terkelola di Azure, pertimbangan tambahan harus diberikan pada faktor-faktor tersebut untuk membuat keputusan yang lebih tepat mengenai pilihan Anda atas platform streaming terkelola Azure:

- * * Keahlian dan latar belakang tim pengembangan
- * * Keahlian dalam menulis query SQL versus menulis kode
- * * Tingkat keahlian yang diperlukan: Analis versus pengembang untuk pertanyaan
- * * Upaya dan kecepatan pengembangan

- *✱ Menggunakan konektor OOB dibandingkan menggunakan komponen OSS
- *✱ Pemecahan masalah dan diagnostik
- *✱ Pencatatan log kustom, log operasi Azure
- *✱ Skalabilitas, penyesuaian, dan harga

Masalah dengan Solusi Streaming Lainnya

Terlepas dari pilihan platform analitik streaming terkelola Anda di Azure, ada banyak keuntungan memilih platform yang berjalan di Azure dibandingkan solusi satu kali.

Opsi penerapan dengan mesin analisis streaming lainnya dapat memberikan banyak hal yang diinginkan dalam hal penyediaan solusi yang holistik dan mudah dikelola. Beberapa poin penting tercantum di sini yang seharusnya dipertimbangkan saat mengevaluasi platform analitik streaming untuk solusi Anda, sertakan hal berikut:

- *Kedalaman dan keluasan:* Tingkat keterampilan pengembangan yang diperlukan.
- *Kelengkapan:* Biasanya bukan solusi menyeluruh.
- *Keahlian:* Perlunya keterampilan khusus untuk menyiapkan dan memelihara.
- *Biaya:* Pengembangan, pengujian, dan lingkungan produksi serta perizinan.

5.5 IMPLEMENTASI AZURE STREAMING ANALYTICS

Dalam implementasi referensi kami dan di seluruh bab selanjutnya dalam buku ini, kami akan menggunakan Azure Streaming Analytics untuk mengimplementasikan solusi referensi. Azure Streaming Analytics adalah layanan cloud yang terkelola sepenuhnya untuk analisis real-time pada aliran data menggunakan bahasa kueri mirip SQL dengan semantik temporal bawaan. Ini sangat “cocok” untuk kebutuhan solusi.

Keuntungan Azure Streaming Analytics

Memilih platform analitik streaming yang tepat adalah keputusan penting yang akan berdampak besar pada keseluruhan kinerja, keandalan, skalabilitas, dan keseluruhan pengoperasian solusi Anda. Oleh karena itu, akan sangat membantu jika mendokumentasikan alasan spesifik dalam memilih Microsoft Azure Streaming Analytics dan membagikannya untuk pertimbangan arsitektur solusi di masa mendatang.

Tidak Ada Tantangan dengan Penerapan

Azure Streaming Analytics adalah layanan PaaS yang terkelola sepenuhnya di cloud, sehingga Anda dapat dengan cepat mengonfigurasi dan menerapkan dari Portal Azure atau melalui skrip penerapan PowerShell.

- Tidak ada akuisisi dan pemeliharaan perangkat keras
- Mengabaikan persyaratan keahlian penerapan
- Aktif dan berjalan dalam beberapa klik (dan dalam hitungan menit)
- Tidak ada penyediaan atau pemeliharaan perangkat lunak
- Kembangkan bisnis Anda secara global dengan mudah

Keandalan Misi Kritis

- Mencapai keandalan dan skala yang sangat penting dengan Azure Streaming Analytics
- Jaminan pengiriman tepat satu kali hingga adaptor keluaran yang menulis kejadian keluaran

- Manajemen negara untuk pemulihan otomatis
- Dijamin tidak akan ada event yang hilang atau output yang salah
- Mempertahankan urutan acara per perangkat
- Dijamin 99,9% ketersediaan SLA

Keberlangsungan bisnis

- Stream Analytics memproses data pada throughput tinggi dengan hasil yang dapat diprediksi dan tanpa kehilangan data
- Jaminan uptime (ketersediaan tiga sembilan)
- Pemulihan otomatis dari kegagalan
- Manajemen status bawaan untuk pemulihan cepat

Tidak Ada Tantangan dengan Skala:

Skalakan ke volume data apa pun sambil tetap mencapai throughput tinggi, latensi rendah, dan ketahanan terjamin.

- Elastisitas cloud untuk peningkatan atau penurunan skala
- Memutar sejumlah sumber daya sesuai permintaan
- Skalakan dari kecil ke besar bila diperlukan
- Arsitektur yang terdistribusi dan diperluas skalanya
- Penskalaan menggunakan penggeser di Portal Azure dan bukan menulis kode

Biaya Awal yang Rendah

Azure Stream Analytics memungkinkan Anda dengan cepat mengembangkan dan menyebarkan solusi berbiaya rendah untuk mendapatkan wawasan real-time dari perangkat, sensor, infrastruktur, dan aplikasi.

- Menyediakan dan menjalankan solusi streaming mulai dari \$25/bulan
- Bayar hanya untuk sumber daya yang Anda gunakan
- Kemampuan untuk menambah sumber daya secara bertahap
- Mengurangi biaya ketika bisnis memerlukan perubahan

Perkembangan yang cepat

Mengurangi gesekan dan kompleksitas serta menggunakan lebih sedikit baris kode saat mengembangkan fungsi analitik untuk memperluas skala sistem terdistribusi. Jelaskan transformasi yang diinginkan dengan sintaksis berbasis SQL, dan Stream Analytics secara otomatis mendistribusikannya untuk skala, performa, dan ketahanan.

- Kurangi bilah untuk membuat solusi pemrosesan aliran melalui Bahasa mirip SQL
- Memfilter, memproyeksikan, menggabungkan, menggabungkan aliran, menambahkan data statis dengan data streaming dengan mudah, dan mendeteksi pola atau kekurangan pola dengan beberapa baris SQL
- Semantik temporal bawaan

Pengalaman Pengembangan dan Debugging Melalui Portal Azure

Kueri di Azure Stream Analytics dinyatakan dalam bahasa kueri mirip SQL. Di Azure Stream Analytics, pesan pencatatan operasional dapat digunakan untuk tujuan debugging seperti melihat status pekerjaan, kemajuan pekerjaan, dan pesan kegagalan untuk melacak kemajuan pekerjaan dari waktu ke waktu; dari awal, pemrosesan, hingga keluaran.

- Kelola acara yang tidak berurutan dan tindakan pada acara yang datang terlambat melalui konfigurasi

Penjadwalan dan Pemantauan Bawaan

Portal Manajemen Azure dan Portal Azure menampilkan metrik performa utama yang dapat digunakan untuk memantau dan memecahkan masalah kueri dan performa pekerjaan Anda.

- Pemantauan bawaan
- Lihat sekilas kinerja sistem Anda
- Membantu Anda menemukan cara penerapan yang hemat biaya

Mengapa Pelanggan Menggunakan Azure Stream Analytics?

Bagian sebelumnya menguraikan beberapa keuntungan utama menggunakan Azure Streaming Analytics. Dengan memanfaatkan Microsoft Azure Streaming Analytics untuk provisi infrastruktur yang cepat beserta aspek pemeliharaan rendah saat berjalan pada platform analitik streaming yang terkelola sepenuhnya, Anda dapat menghindari komplikasi yang biasa tercantum berikutnya:

- Memantau dan memecahkan masalah solusinya.
- Mengembangkan solusi dan infrastruktur yang dapat berkembang seiring dengan pertumbuhan bisnis.
- Mengembangkan solusi untuk mengelola ketahanan, seperti kegagalan infrastruktur dan redundansi geografis.
- Mengembangkan solusi untuk diintegrasikan dengan komponen lain seperti Machine Learning, BI, dll.
- Mengembangkan solusi (kode) untuk penyerapan, pemrosesan sementara, dan operasi jalan keluar panas/dingin.
- Pengadaan infrastruktur: menghindari penundaan perangkat keras yang lama dan penyediaannya dalam hitungan menit.

Fokus pada membangun solusi, bukan pada infrastruktur solusi, dan kembangkan serta terapkan aplikasi lebih cepat sehingga Anda benar-benar dapat bekerja dengan kecepatan Internet.

Perlu dicatat bahwa selain banyak manfaat yang diuraikan untuk pelanggan Azure individu, Azure Streaming Analytics juga merupakan teknologi inti yang membentuk beberapa penawaran solusi berbasis Microsoft Azure lainnya, seperti:

- **Azure IoT Suite:** Microsoft menyediakan Azure IoT Suite sebagai bagian dari solusi IoT yang telah dikonfigurasi sebelumnya dan dibangun di platform Azure dan memudahkan untuk menyambungkan perangkat dengan aman dan menyerap peristiwa dalam skala besar.
- **Cortana Intelligence Suite:** Big Data yang dikelola sepenuhnya dan rangkaian analitik tingkat lanjut untuk mengubah data Anda menjadi tindakan cerdas.

Skenario Vertikal Utama untuk Dijelajahi untuk Azure Stream Analytics. Ada banyak kasus penggunaan untuk memanfaatkan analisis streaming di seluruh industri vertikal. Beberapa aplikasi yang lebih populer tercantum di sini:

- Jasa Keuangan:
 - Deteksi penipuan
 - Pelacakan aset
- Kesehatan:
 - Pemantauan pasien
- Pemerintah:
 - Pengawasan dan pemantauan
- Infrastruktur, Energi, dan Utilitas:
 - Manajemen operasi di bidang minyak dan gas
 - Bangunan pintar
- Manufaktur:
 - Pemeliharaan prediktif
 - Pemantauan jarak jauh
- Pengecer:
 - Keterlibatan dan pemasaran pelanggan secara real-time
 - Optimalisasi inventaris
- Telko/TI:
 - Infrastruktur TI dan pemantauan jaringan seluler
 - Kesadaran berbasis lokasi
- Transportasi dan Logistik:
 - Pemantauan kontainer
 - Pelacakan pengiriman barang yang mudah rusak

Solusi Kami: Memanfaatkan Azure Streaming Analytics

Dalam solusi referensi kami, kami menggunakan Azure Stream Analytics, yang merupakan mesin pemrosesan peristiwa yang terkelola sepenuhnya, hemat biaya, dan real-time yang dapat membantu kami membuka wawasan mendalam dari data kami. Azure Stream Analytics memudahkan pengaturan komputasi analitik real-time pada streaming data dari perangkat, sensor, situs web, media sosial, aplikasi, sistem infrastruktur, dan banyak lagi. Ini sempurna untuk solusi kami.

Sebelum memulai panduan penerapan arsitektur referensi spesifik, kami akan menjelajahi keseluruhan alur kerja pembuatan pekerjaan Streaming Analytics di Microsoft Azure. Stream Analytics memanfaatkan penelitian Microsoft selama bertahun-tahun dalam mengembangkan mesin streaming yang sangat disesuaikan untuk pemrosesan yang sensitif terhadap waktu, serta integrasi bahasa SQL untuk spesifikasi intuitif pekerjaan streaming.

Dengan beberapa klik di Portal Azure, Anda dapat menulis pekerjaan Stream Analytics yang menentukan tiga komponen utama Solusi Azure Streaming Analytics. Ketiga komponen tersebut adalah input, output, dan kueri U-SQL.

Pekerjaan Analisis Streaming: Definisi INPUT

Tugas pertama dalam menyiapkan pekerjaan Azure Streaming Analytics adalah menentukan input untuk pekerjaan streaming baru. Definisi masukan terkait dengan sumber data streaming yang masuk.

Perhatikan bahwa, pada saat penulisan ini, hanya ada dua tipe format data yang didukung:

- * JSON: Streaming muatan pesan dalam format JavaScript-Object-Notation.
- * CSV: Streaming data dalam format teks Nilai yang Dipisahkan Koma. Baris header juga didukung (dan direkomendasikan) untuk menyediakan fungsionalitas penamaan kolom tambahan.

Saat menentukan input aliran data, ada dua jenis definisi—Aliran Data dan Data Referensi:

- *Aliran Data*: Aliran data dilambangkan sebagai rangkaian peristiwa tak terbatas yang mengalir sepanjang waktu. Pekerjaan Stream Analytics harus menyertakan setidaknya satu input aliran data untuk digunakan dan diubah oleh pekerjaan. Pekerjaan Stream Analytics harus menyertakan setidaknya satu input aliran data untuk digunakan dan diubah oleh pekerjaan. Sumber masukan aliran data yang didukung mencakup jenis masukan berikut: (pada saat penulisan ini):
 - Aliran IoT Hub:
 - Azure IoT Hub adalah platform penyerapan peristiwa terbitkan-langganan yang sangat skalabel dan dioptimalkan untuk skenario IoT.
 - Digunakan untuk aliran perpesanan dari perangkat ke cloud dan cloud ke perangkat.
 - Dioptimalkan untuk mendukung jutaan perangkat yang terhubung secara bersamaan.
 - Dapat digunakan untuk mengirim pesan masuk dan keluar ke perangkat IoT.
 - Aliran Event Hub:
 - Mengaktifkan aliran pesan peristiwa masuk (hanya) untuk skenario perangkat-ke-cloud.
 - Mendukung jumlah koneksi simultan yang lebih terbatas.
 - Event Hubs memungkinkan Anda menentukan partisi untuk setiap pesan yang dikirim untuk meningkatkan skalabilitas.
 - Penyimpanan Blob: Digunakan sebagai sumber input untuk menyerap data massal sebagai aliran. Untuk skenario dengan data tidak terstruktur dalam jumlah besar untuk disimpan di cloud, penyimpanan blob menawarkan solusi hemat biaya dan skalabel.
- *Data Referensi*: Stream Analytics mendukung jenis masukan tambahan kedua yang disebut data referensi. Berbeda dengan data yang bergerak, data referensi bersifat statis atau perubahannya melambat.
 - Biasanya digunakan untuk melakukan pencarian dan korelasi dengan aliran data lain untuk memperkaya kumpulan data.
 - Pada saat penulisan ini, penyimpanan blob Azure saat ini merupakan satu-satunya sumber input yang didukung untuk data referensi. Blob sumber data referensi saat ini dibatasi hingga ukuran 100 MB.

Pekerjaan Analisis Streaming: Definisi OUTPUT

Kami telah menentukan sumber input untuk pekerjaan Azure Streaming Analytics baru, jadi langkah selanjutnya adalah menentukan format output untuk pekerjaan tersebut. Untuk mengaktifkan berbagai pola aplikasi, Azure Stream Analytics memiliki opsi berbeda untuk menyimpan output dan melihat hasil analisis. Hal ini memudahkan untuk melihat keluaran pekerjaan dan memberikan fleksibilitas dalam konsumsi dan penyimpanan keluaran pekerjaan untuk gudang data dan tujuan lainnya.

Perhatikan bahwa output apa pun yang dikonfigurasi dalam pekerjaan harus ada sebelum pekerjaan dimulai dan peristiwa mulai mengalir. Misalnya, jika Anda menggunakan penyimpanan blob sebagai output, pekerjaan tidak akan membuat akun penyimpanan secara otomatis. Ini perlu disediakan oleh pengguna sebelum pekerjaan Streaming Analytics dimulai. Format Output untuk Azure Streaming Analytics mencakup opsi penyimpanan berikut (pada saat penulisan ini):

- *SQL Database:* Azure SQL Database dapat digunakan sebagai output untuk data yang bersifat relasional atau untuk aplikasi yang bergantung pada konten yang dihosting dalam database relasional. Satu-satunya persyaratan untuk opsi keluaran Stream Analytics ini adalah tujuannya adalah tabel yang sudah ada di Azure SQL Database. Akibatnya, skema tabel harus sama persis dengan bidang dan jenisnya yang dihasilkan dari pekerjaan analisis streaming.
- *Gudang Data Azure SQL:* Perhatikan bahwa Gudang Data Azure SQL juga dapat ditentukan sebagai output melalui opsi output SQL Database. Fitur ini sedang dalam mode pratinjau pada saat penulisan ini.
- *Penyimpanan Blob:* Penyimpanan Blob menawarkan solusi hemat biaya dan terukur untuk menyimpan data tidak terstruktur dalam jumlah besar di cloud.
- *Event Hubs:* Konstruksi penyerapan peristiwa publikasikan-langganan yang sangat skalabel. Event Hubs sangat skalabel dan dapat menangani penyerapan jutaan peristiwa per detik. Perhatikan bahwa alasan utama menggunakan Event Hub sebagai Output dari pekerjaan Stream Analytics adalah agar data dapat menjadi input dari pekerjaan streaming lainnya. Dengan cara ini, Anda dapat “merantai” beberapa pekerjaan streaming untuk menyelesaikan skenario aplikasi, seperti memberikan peringatan dan pemberitahuan waktu nyata.
- *Penyimpanan Tabel:* Penyimpanan kunci/atribut NoSQL yang dapat dimanfaatkan untuk data terstruktur dengan batasan skema minimal. Penyimpanan tabel menyediakan penyimpanan yang sangat tersedia dan dapat diskalakan secara besar-besaran, sehingga aplikasi dapat secara otomatis melakukan penskalaan untuk memenuhi permintaan pengguna.
- *Antrean Bus Layanan:* Menyediakan pengiriman pesan First-In, First-Out (FIFO) ke satu atau lebih konsumen. Pesan kemudian diatur untuk diterima dan diproses oleh penerima sesuai urutan tanggal/waktu saat pesan tersebut ditambahkan ke antrian, dan setiap pesan diterima dan diproses hanya oleh satu konsumen pesan.

- *Topik Bus Layanan*: Meskipun antrian bus layanan menyediakan pola komunikasi satu-ke-satu dari pengirim ke penerima, topik bus layanan menyediakan bentuk komunikasi satu-ke-banyak di mana banyak aplikasi dapat berlangganan suatu topik.
- *DocumentDB*: Layanan database dokumen NoSQL yang terkelola sepenuhnya yang menawarkan kueri dan transaksi melalui data bebas skema, kinerja yang dapat diprediksi dan andal, serta perkembangan pesat.
- *Power BI*: Dapat digunakan sebagai output untuk pekerjaan Stream Analytics guna menyediakan visualisasi yang kaya untuk hasil analitis. Kemampuan ini dapat digunakan untuk dasbor operasional, pembuatan laporan, dan pelaporan berbasis metrik.
- *Data Lake Store*: Opsi keluaran ini memungkinkan Anda menyimpan data dalam berbagai ukuran, jenis, dan kecepatan penyerapan untuk analisis operasional dan eksplorasi. Saat ini, pembuatan dan konfigurasi output Data Lake Store hanya didukung di Portal Azure Classic.

5.6 MERENCANAKAN OUTPUT ANALISIS STREAMING

Tahap keluaran dari proses analisis streaming memerlukan sedikit perencanaan awal dan lanjutan, karena beberapa pemikiran harus diberikan mengenai bagaimana data akan dikirimkan dan dikonsumsi. Di sinilah nilai sebenarnya dari arsitektur Lambda modern berperan dengan gagasan jalur data “panas”, “hangat”, dan “dingin”.

Analisis dan eksploitasi yang tepat atas kemampuan pelaporan utama ini dapat menjadi pembeda antara menciptakan keunggulan kompetitif yang sesungguhnya dan sekadar menciptakan skenario kelebihan data yang bising. Nasib banyak bisnis bisa naik turun tergantung pada ketepatan waktu dan keakuratan data operasional utama. Pilih keluaran Anda dengan hati-hati.

Jalur Panas

Jalur pemrosesan data mendesak, misalnya data yang dikirim dari perangkat lapangan ke sistem IoT. Data ini biasanya memerlukan analisis segera. Ini sering digunakan untuk meningkatkan peringatan dan pemberitahuan penting lainnya. Opsi keluaran “jalur panas” untuk Azure Streaming Analytics meliputi:

- *Power BI*: Untuk integrasi streaming real-time bersama dengan dasbor visual yang kaya.
- *Event Hubs*: Untuk berintegrasi dengan pekerjaan Streaming Analytics lainnya dan hub Notifikasi keluar.
- *Antrean Bus Layanan*: Untuk integrasi dengan sistem pemberitahuan penerbit/pelanggan lainnya.
- *Topik Bus Layanan*: Untuk integrasi dengan skenario notifikasi satu-ke-banyak.

Jalan Hangat

Jalur pemrosesan untuk data perangkat yang tidak mendesak namun biasanya memiliki masa pakai terbatas sebelum menjadi basi. Data ini harus dianggap memiliki “tanggal kedaluwarsa” dan karenanya harus diproses dalam jangka waktu tertentu. Data ini juga dapat digunakan untuk menambah hasil yang dihasilkan oleh pemrosesan jalur panas guna

memberikan konteks tambahan. Contoh data jalur hangat mencakup informasi diagnostik untuk analisis kinerja, pemecahan masalah, atau pengujian A/B. Data mungkin perlu disimpan dalam penyimpanan yang relatif cepat untuk diakses (dan oleh karena itu mungkin lebih mahal daripada yang diperlukan untuk jalur dingin), namun kapasitas penyimpanannya mungkin jauh lebih kecil, karena data ini memiliki masa pakai yang terbatas dan kemungkinan besar tidak akan dipertahankan untuk jangka waktu yang lama.

Opsi keluaran “jalur hangat” untuk Azure Streaming Analytics meliputi:

- Azure SQL Database: Untuk kueri data relasional yang hampir real-time
- DocumentDB: Untuk kueri data NOSQL yang hampir real-time

Jalan Dingin

Jalur pemrosesan data yang disimpan dan diproses nantinya. Misalnya, data ini dapat diambil dari penyimpanan untuk diproses nanti dalam mode batch. Data dapat disimpan dalam penyimpanan yang relatif murah dan berkapasitas tinggi karena potensi volume tinggi dan sifat historisnya. Data biasanya digunakan untuk memberikan informasi statistik, untuk menghasilkan laporan analitis, dan untuk tujuan audit.

Opsi keluaran “jalur dingin” untuk Azure Streaming Analytics meliputi:

- *Penyimpanan Blob*: Untuk penyimpanan data umum yang berbiaya rendah dan berskala tinggi
- *Penyimpanan Tabel*: Untuk penyimpanan data pasangan nilai kunci berbiaya rendah dan berskala tinggi
- *Data Lake Store*: Untuk platform penyimpanan data historis yang tidak terbatas, berbiaya rendah, berskala tinggi, dengan kemampuan pemrosesan analitis yang mendalam.
- *Power BI*: Untuk integrasi streaming real-time bersama dengan dasbor visual yang kaya.
- *Event Hubs*: Untuk berintegrasi dengan pekerjaan Streaming Analytics lainnya dan hub Notifikasi keluar.
- *Antrean Bus Layanan*: Untuk integrasi dengan sistem pemberitahuan penerbit/pelanggan lainnya.
- *Topik Bus Layanan*: Untuk integrasi dengan skenario notifikasi satu-ke-banyak.

Power BI untuk Visualisasi Real-Time

Power BI dapat digunakan sebagai output untuk pekerjaan Stream Analytics guna memberikan pengalaman visualisasi hasil analisis yang kaya dan real-time. Kemampuan ini dapat digunakan untuk dasbor operasional, pembuatan laporan dinamis, dan bentuk pelaporan dan analisis berbasis metrik lainnya secara real-time.

Pekerjaan Analisis Streaming: Transformasi Data melalui Kueri SQL

Setelah definisi input dan output pekerjaan Azure Streaming Analytics dibuat, tugas berikutnya adalah membuat transformasi data. Di sinilah semua bagian mulai bersatu dan solusi lengkap akhirnya dapat dikonfigurasi berdasarkan masukan dan keluaran yang telah ditentukan sebelumnya.

Azure Stream Analytics menawarkan bahasa kueri mirip SQL untuk melakukan transformasi dan komputasi pada aliran data peristiwa yang masuk. Bahasa kueri Stream

Analytics adalah bagian dari sintaksis Transaction-SQL (T-SQL) standar untuk melakukan komputasi analisis streaming yang sederhana dan kompleks.

Azure Streaming Analytics SQL: Ramah Pengembang

Karena sebagian besar pengembang saat ini mungkin sudah memiliki pengetahuan yang baik tentang T-SQL, fitur ini membuatnya sangat mudah untuk menjadi sangat produktif dalam waktu yang sangat singkat saat menggunakan Azure Streaming Analytics.

Bagian penyiapan tugas Streaming Analytics ini adalah tempat pemrosesan sebenarnya akan terjadi dan kami akan memetakan, memperkaya, dan mengubah masukan data streaming yang masuk menjadi satu atau lebih keluaran streaming yang telah ditentukan sebelumnya. Perhatikan bahwa dalam satu pekerjaan Streaming Analytics, dimungkinkan untuk mengirim data yang diproses dari satu masukan ke beberapa tujuan keluaran dengan menyatukan pernyataan SQL dalam pekerjaan tersebut.

Azure Streaming Analytics (ASA): Fitur Dialek Kueri SQL

Bahasa SQL di ASA sangat mirip dengan T-SQL, yang merupakan bahasa database utama untuk mesin aplikasi database SQL modern seperti Microsoft SQL Server, IBM DB2, dan server Oracle Database.

Perlu dicatat bahwa Azure Streaming Analytics (ASA) SQL juga berisi superset fungsi yang mendukung kemampuan analitik tingkat lanjut sebagai operasi "temporal" (tanggal/waktu) seperti menerapkan jendela waktu geser, lompat, atau jatuh ke aliran peristiwa secara berurutan untuk mendapatkan data ringkasan yang dibatasi waktu langsung dari aliran peristiwa masuk.

Semua ini dicapai dengan fasih dan mudah menggunakan pernyataan T-SQL yang sudah dikenal. Untuk membantu Anda memahami lebih lanjut fitur bahasa ASA SQL Query, berikut adalah sinopsis singkat kemampuan utama yang tersedia dengan bahasa Azure Streaming Analytics SQL Query.

Bahasa Kueri SQL

- Semua tugas transformasi data ditulis secara deklaratif sebagai serangkaian pernyataan bahasa kueri mirip T-SQL.
- Tidak diperlukan pemrograman tambahan dan tidak diperlukan kompilasi kode.
- Skrip SQL mudah dibuat dan diterapkan.

Tipe Data yang Didukung

Tipe data berikut ini didukung dalam bahasa ASA SQL:

- ❖ Bigint: Bilangan bulat dalam rentang -2^{63} (-9,223,372,036,854,775,808) hingga $2^{63}-1$ (9,223,372,036,854,775,807).
- ❖ Float: Angka floating-point dalam kisaran $-1.79E+308$ hingga $-2.23E-308$, 0, dan $2.23E-308$ hingga $1.79E+308$.
- ❖ Datetime: Mendefinisikan tanggal yang dikombinasikan dengan waktu dalam sehari dengan pecahan detik yang didasarkan pada jam 24 jam dan relatif terhadap UTC (zona waktu offset 0).
- ❖ nvarchar (maks): Nilai teks terdiri dari karakter Unicode.
- ❖ record: Satu set pasangan nama/nilai. Nilai harus berupa tipe data yang didukung.

* array: Kumpulan nilai yang terurut. Nilai harus berupa tipe data yang didukung.

5.7 KONVERSI TIPE DATA

Konversi tipe data dalam tipe dalam bahasa kueri Stream Analytics dilakukan melalui fungsi CAST. Fungsi ini mengonversi ekspresi satu tipe data ke tipe data lain dalam tipe yang didukung dalam bahasa kueri Stream Analytics. Kehati-hatian harus diberikan saat menggunakan fungsi CAST pada aliran data yang tidak konsisten, karena kegagalan akan menyebabkan pekerjaan analisis streaming berhenti jika konversi tidak dapat dilakukan. Sebagai contoh bagus tentang apa yang tidak boleh dilakukan pernyataan ASA SQL di Listing 5.1 akan mengakibatkan kegagalan pekerjaan Azure Streaming Analytics.

Daftar 5.1. Contoh Penggunaan Operator CAST yang Akan Mengakibatkan Kegagalan Pekerjaan dalam Pernyataan ASA SQL

```
CAST ('Test-String' AS bigint)
```

TRY_CAST

Untuk menghindari kegagalan pekerjaan yang parah karena kegagalan konversi tipe data, sangat disarankan agar operasi SQL *TRY_CAST* digunakan sebagai gantinya. Versi ini mengembalikan nilai yang diberikan ke tipe data yang ditentukan jika pemeran berhasil atau jika tidak, panggilan akan mengembalikan null.

Pekerjaan transformasi SQL akan berlanjut dengan baik apa pun hasil panggilan *TRY_CAST*.

Daftar 5.2. Contoh Operator TRY_CAST dalam Pernyataan ASA SQL

```
SELECT TweetId, TweetTime
FROM Input
WHERE TRY_CAST( TweetTime AS datetime) IS NOT NULL
```

Fungsi Semantik Temporal

Semua operator ASA SQL kompatibel dengan properti temporal aliran peristiwa. Fungsionalitas tambahan ditambahkan ke bahasa ASA SQL melalui operator baru seperti:

- TumblingWindow
- HoppingWindow
- SlidingWindow

Operator dan Fungsi Bawaan

- Bahasa ASA SQL mendukung konstruksi T-SQL utama lainnya seperti filter, proyeksi, gabungan, agregat berjendela (temporal), dan fungsi manipulasi teks dan tanggal.
- Kueri aliran peristiwa tingkat lanjut dapat dibuat melalui ekstensi kueri canggih ini.

Fungsi Buatan Pengguna: Integrasi Pembelajaran Mesin Azure

- Bahasa ASA SQL sekarang mendukung panggilan langsung ke Azure Machine Learning (ML) melalui fungsi yang ditentukan pengguna.
- Fungsi yang ditentukan pengguna menyediakan cara yang dapat diperluas untuk pekerjaan streaming untuk mengubah data masukan menjadi data keluaran menggunakan fungsi yang ditentukan secara eksternal dan diakses sebagai bagian dari kueri SQL.
- Fungsi Pembelajaran Mesin dalam analisis aliran dapat digunakan seperti panggilan fungsi biasa dalam bahasa kueri analisis aliran.
- Fungsionalitas ini memberikan kemampuan untuk menilai peristiwa individual data streaming dengan memanfaatkan model Pembelajaran Mesin yang dihosting di Azure dan diakses melalui panggilan layanan web.
- Pada saat penulisan ini, Layanan Permintaan-Respons Azure Machine Learning (RRS) adalah satu-satunya kerangka kerja UDF yang didukung dan saat ini berada dalam mode “pratinjau”.
- Kemampuan ini memungkinkan Anda dengan mudah membangun aplikasi untuk skenario seperti analisis sentimen Twitter real-time, seperti yang diilustrasikan pada Daftar 5.3. Fungsi yang ditentukan pengguna Azure Machine Learning bernama sentimen dengan mudah dimasukkan ke dalam kueri ASA SQL. Kemampuan ini memberikan mekanisme yang kuat untuk memanfaatkan analisis prediktif guna memperkaya data aliran peristiwa yang masuk dan mengubahnya menjadi data yang dapat ditindaklanjuti.

Daftar 5-3. Contoh Fungsi Buatan Pengguna Azure Machine Learning

```
WITH subquery AS (
  SELECT text, sentiment(text) as result from input
)
SELECT text, result.[Scored Labels]
INTO output
FROM subquery
```

Jaminan Pengiriman Peristiwa Disediakan di Azure Stream Analytics

Bahasa kueri Azure Stream Analytics menyediakan ekstensi ke sintaks T-SQL untuk memungkinkan komputasi kompleks atas aliran peristiwa yang masuk. Dengan Azure Streaming Analytics, konsep terkait pengiriman peristiwa berikut ini perlu ditinjau:

- Tepat sekali pengiriman
- Duplikat catatan

Tepat Sekali Pengiriman

Jaminan pengiriman “tepat sekali” berarti semua peristiwa masukan diproses tepat satu kali oleh sistem analisis streaming. Dengan cara ini, hasilnya juga dijamin lengkap dan tidak ada keluaran yang duplikat. Dalam hal Perjanjian Tingkat Layanan (SLA) Azure, Azure

Stream Analytics menjamin tepat satu kali pengiriman ke adaptor output yang menulis peristiwa output.

Catatan Duplikat

Saat pekerjaan Stream Analytics berjalan, rekaman duplikat terkadang terjadi dalam data output. Catatan duplikat ini diharapkan terjadi, karena adaptor keluaran Azure Stream Analytics tidak menulis peristiwa keluaran dengan cara transaksional yang lengkap.

Fungsi Manajemen Waktu

Bahasa kueri Azure Stream Analytics SQL memperluas sintaksis T-SQL untuk memungkinkan komputasi kompleks melalui aliran peristiwa. Stream Analytics menyediakan konstruksi bahasa untuk menangani aspek temporal data. Misalnya, dimungkinkan untuk menetapkan stempel waktu khusus ke peristiwa aliran, menentukan jangka waktu untuk agregasi, menentukan perbedaan waktu yang diperbolehkan antara dua aliran data untuk operasi GABUNG, dll.

- *System.Timestamp*: Properti sistem yang dapat digunakan untuk mengambil stempel waktu suatu peristiwa.
- *Kebijakan Kemiringan Waktu*: Memberikan kebijakan untuk kejadian yang tidak sesuai pesanan dan kedatangannya terlambat.
- *Fungsi Agregat*: Digunakan untuk melakukan perhitungan pada sekumpulan nilai dari jangka waktu tertentu dan mengembalikan satu nilai.
- *DATEDIFF*: Diizinkan dalam predikat JOIN dan memungkinkan spesifikasi batas waktu untuk operasi JOIN.
- *Fungsi Tanggal dan Waktu*: Azure Stream Analytics menyediakan berbagai fungsi tanggal dan waktu untuk digunakan dalam membuat kueri analitik streaming yang sensitif terhadap waktu.
- *TIMESTAMP BY*: Memungkinkan penentuan nilai stempel waktu khusus.

Pentingnya Klausul TIMESTAMP BY

Di Azure Streaming Analytics, semua peristiwa masuk memiliki stempel waktu yang ditentukan dengan baik. Jika solusi diperlukan untuk menggunakan waktu aplikasi, mereka dapat menggunakan kata kunci `TIMESTAMP BY` untuk menentukan kolom dalam payload yang harus digunakan untuk memberi stempel waktu pada setiap peristiwa yang masuk untuk melakukan perhitungan temporal seperti fungsi windowing (Hopping, Tumbling, Sliding), GABUNG Temporal, dll.

Perhatikan bahwa direkomendasikan untuk menggunakan klausa `TIMESTAMP BY` pada “waktu kedatangan” sebagai praktik terbaik karena klausa `TIMESTAMP BY` dapat digunakan pada kolom jenis “datetime” apa pun dan semua format ISO 8601 didukung. Sebagai perbandingan, nilai `System.timestamp` hanya dapat digunakan dalam klausa `SELECT`. Daftar 5.4 mengilustrasikan contoh `TIMESTAMP BY` yang menggunakan kolom `TweetTime` sebagai waktu penerapan untuk semua acara yang masuk.

Daftar 5-4. Klausul TIMESTAMP BY

```
SELECT TweetId, TweetTime
FROM TweetInput
TIMESTAMP BY TweetTime
```

Azure Stream Analytics: Model Pemrograman Terpadu

Seperti yang telah kita lihat di bagian sebelumnya yang mencakup superset fitur, fungsi, dan kemampuan yang memperluas dialek SQL Azure Streaming Analytics, hasil akhirnya adalah model pemrograman berbasis SQL yang sangat kuat, namun mudah didekati, yang menyatukan aliran peristiwa, data referensi, dan ekstensi Machine Learning untuk menciptakan solusi komprehensif.

Azure Stream Analytics: Contoh Model Pemrograman SQL

Contoh Paling Sederhana

Daftar 5.5 adalah contoh yang sangat sederhana dari Query SQL streaming yang akan menyalin semua bidang dalam nama masukan input iothub ke dalam output bernama blob-output.

Daftar 5.5. Kueri ASA SQL yang paling sederhana

```
Select * into blob-output from iothub-input
```

Dalam banyak kasus, kueri SQL Azure Streaming Analytics akan lebih kompleks dan biasanya akan menggabungkan berbagai semantik temporal untuk menampilkan data yang terkait dengan jendela jangka waktu geser, lompat, atau jungkir balik dari aliran peristiwa. Seperti disebutkan sebelumnya, di sinilah kekuatan sebenarnya dari Azure Streaming Analytics benar-benar bersinar, karena sangat mudah dicapai melalui superset fungsionalitas yang telah ditambahkan Microsoft ke dialek T-SQL yang sudah dikenal. Sebagai ilustrasi, contoh jendela temporal berikut akan membuat asumsi bahwa kita membaca dari aliran masukan tweet dari Twitter.

Jendela Tumbling 10 Detik

Jendela jatuh dapat didefinisikan sebagai serangkaian interval waktu berukuran tetap, tidak tumpang tindih, dan berdekatan yang diambil dari aliran data. ASA SQL pada Daftar 5-6 berupaya menjawab pertanyaan berikut: “Beri tahu saya jumlah tweet per zona waktu setiap 10 detik”

Daftar 5-6. Contoh Pernyataan SQL Jendela Tumbling

```
SELECT TimeZone, COUNT(*) AS Count
FROM TwitterStream TIMESTAMP BY CreatedAt
GROUP BY TimeZone, TumblingWindow(second,10)
```

Hopping Windows: Jendela Lompatan 10 Detik dengan “Hop” 5 Detik

Jendela lompat dirancang untuk memodelkan jendela terjadwal yang tumpang tindih. ASA SQL pada Daftar 5-7 berupaya menjawab pertanyaan berikut:

“Setiap 5 detik, beri saya jumlah tweet selama 10 detik terakhir”

Daftar 5-7. Contoh Pernyataan SQL Jendela Hopping

```
SELECT Topic, COUNT(*) AS TotalTweets
FROM TwitterStream TIMESTAMP BY CreatedAt
GROUP BY Topic, HoppingWindow(second, 10 , 5)
```

Jendela Geser: Jendela Geser 10 Detik

Dengan jendela geser, sistem diminta untuk secara logis mempertimbangkan semua kemungkinan jendela dengan panjang tertentu dan menampilkan peristiwa ketika konten jendela benar-benar berubah, misalnya, ketika suatu peristiwa terdeteksi yang masuk atau ada di jendela. ASA SQL pada Daftar 5-8 berupaya menjawab pertanyaan berikut:

“Beri saya jumlah tweet untuk semua topik yang di-tweet lebih dari 10 kali dalam 10 detik terakhir”

Daftar 5-8. Contoh Pernyataan SQL Jendela Geser

```
SELECT Topic, COUNT(*) FROM TwitterStream
TIMESTAMP BY CreatedAt
GROUP BY Topic, SlidingWindow(second, 10)
HAVING COUNT(*) > 10
```

JOIN dengan Banyak Aliran

Mirip dengan bahasa T-SQL standar, klausa JOIN dalam bahasa kueri Azure Stream Analytics digunakan untuk menggabungkan rekaman dari dua atau lebih sumber input. Namun, klausa GABUNG di Azure Stream Analytics SQL bersifat sementara. Artinya setiap JOIN harus memberikan batasan seberapa jauh baris yang cocok dapat dipisahkan dalam waktu. ASA SQL pada Daftar 5.9 berupaya menjawab pertanyaan berikut:

“Buat daftar semua pengguna dan topik yang mereka ubah sentimennya dalam satu menit”

Daftar 5-9. JOIN Beberapa Aliran dalam ASA SQL

```
SELECT TS1.UserName, TS1.Topic
FROM TwitterStream TS1 TIMESTAMP BY CreatedAt
JOIN TwitterStream TS2 TIMESTAMP BY CreatedAt
ON TS1.UserName = TS2.UserName AND TS1.Topic = TS2.Topic
AND DateDiff(second, TS1, TS2) BETWEEN 1 AND 60
WHERE TS1.SentimentScore != TS2.SentimentScore
```

Mendeteksi Absennya Peristiwa

Pola kueri SQL ini bisa sangat berguna karena akan memberikan kemampuan untuk menentukan apakah suatu aliran tidak memiliki nilai yang cocok dengan kriteria tertentu. Misalnya, Daftar 5.10 adalah contoh kueri ASA SQL yang akan berupaya memberikan jawaban real-time untuk pertanyaan tersebut.

“Tunjukkan kepada saya jika suatu topik tidak di-tweet selama 10 detik sejak terakhir kali di-tweet.”

Daftar 5-10. Mendeteksi Ketiadaan Data di ASA SQL

```
SELECT TS1.CreatedAt, TS1.Topic
FROM TwitterStream TS1 TIMESTAMP BY CreatedAt
LEFT OUTER JOIN TwitterStream TS2 TIMESTAMP BY CreatedAt
ON TS1.Topic = TS2.Topic
AND DATEDIFF(second, TS1, TS2) BETWEEN 1 AND 10
WHERE TS2.Topic IS NULL
```

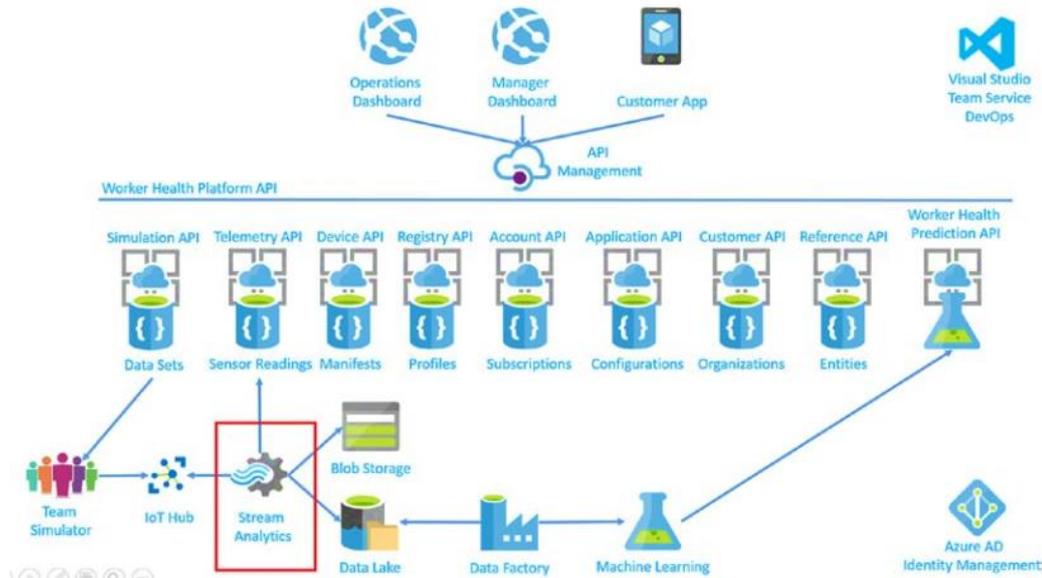
Implementasi Referensi

Sekarang kita telah membahas semua dasar-dasar yang terkait dengan konfigurasi dan penyiapan Azure Streaming Analytics, sekarang saatnya untuk menjalani langkah-langkah konfigurasi sebenarnya untuk solusi implementasi referensi kami. Di bagian selanjutnya, kita akan menelusuri konfigurasi pekerjaan analitik streaming melalui Portal Azure untuk mengimplementasikan berbagai jalur data untuk aliran data IoT masuk. Sebagai bagian dari konfigurasi, kami akan membuat dan mengonfigurasi artefak berikut di Azure:

- ❖ Pekerjaan Azure Streaming Analytics.
- ❖ Masukan: Untuk Pekerjaan Azure Streaming Analytics kami. Dalam hal ini, kita akan menggunakan dua input. Yang pertama adalah untuk aliran data masuk dari IoT Hub, yaitu input kedua untuk data referensi. Dalam contoh ini, kita akan membaca dari referensi File .CSV di penyimpanan blob Azure untuk mencocokkan informasi kesehatan pribadi anggota tim dengan pembacaan informasi sensor kesehatan real-time mereka.
- ❖ Fungsi: Terdiri dari referensi ke Layanan Web Azure Machine Learning. Untuk membantu memperkaya data dengan analisis prediktif. Dalam contoh ini, kita akan memeriksa apakah seorang anggota tim kelelahan hingga mencapai titik kelelahan.
- ❖ Output: Untuk output hasil dari Pekerjaan Azure Streaming Analytics ke dalam berbagai format penyimpanan: Panas, Hangat, dan Dingin (dari arsitektur Lambda).
- ❖ ASA SQL Query Language: Akan menggabungkan semua konfigurasi sebelumnya untuk memproses aliran data masuk dan mengirimkan hasil yang dihitung ke berbagai tujuan keluaran.

Skenario Kasus Penggunaan Bisnis

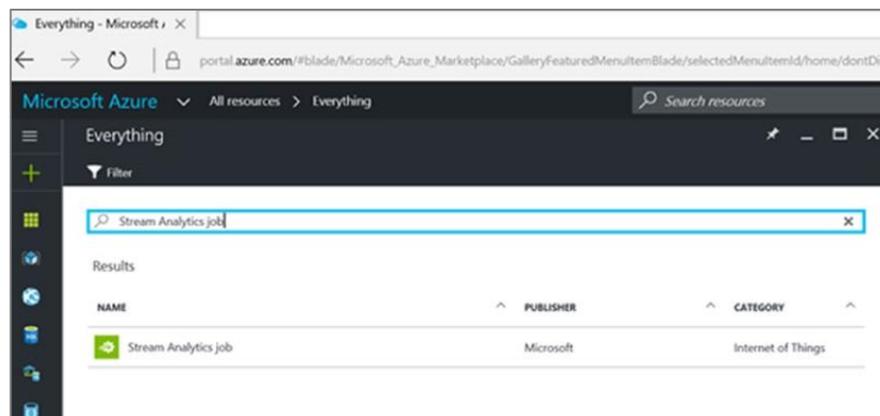
Seperti yang mungkin Anda ingat, skenario kasus penggunaan untuk implementasi referensi kami melibatkan pemantauan kesehatan pekerja selama aktivitas berat. Untuk itu, sensor IoT dipakai dan oleh anggota berbagai tim kerja serta pembacaan sensornya dikirim ke cloud Azure melalui konfigurasi IoT Hub. Langkah penting berikutnya dalam proses ini adalah penggunaan Azure Streaming Analytics untuk memproses aliran data masuk dengan cepat dan efisien. Gambar 5.8 menunjukkan penggunaan Azure Streaming Analytics sebagai mesin pemrosesan penyerapan utama dalam keseluruhan arsitektur.



Gambar 5.8. Solusi Referensi Kesehatan dan Keselamatan Pekerja

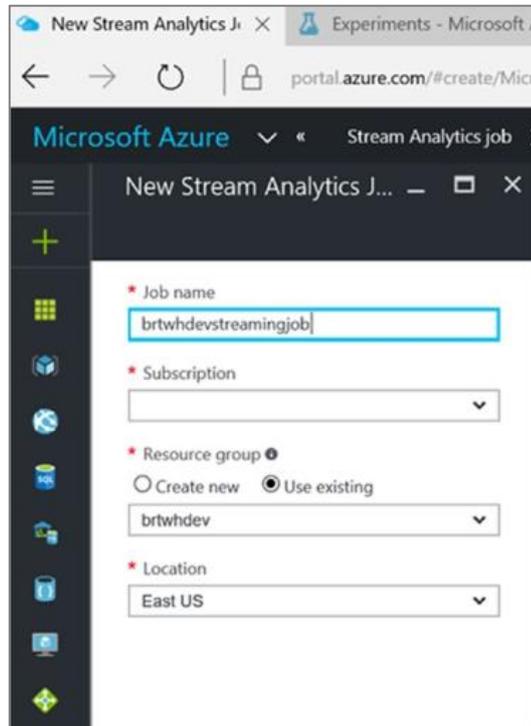
Azure Setup: Buat Pekerjaan Azure Streaming Analytics

Langkah pertama dalam proses ini adalah membuat pekerjaan Azure Streaming Analytics baru. Untuk melakukan hal ini, kami berasumsi bahwa Anda memiliki langganan Azure dan telah menyebarkan komponen infrastruktur suite Azure IoT yang dibahas dalam Bab 1-4 buku ini. Mulailah dengan menambahkan sumber daya baru ke grup sumber daya Azure yang ada dan mencari Pekerjaan Stream Analytics, seperti yang digambarkan dalam Gambar 5.9.



Gambar 5.9. Menambahkan pekerjaan analitik aliran ke grup sumber daya Azure

Setelah mengklik pekerjaan Stream Analytics, Anda akan diminta mengisi parameter untuk membuat pekerjaan Azure Stream Analytics, seperti yang ditunjukkan pada Gambar 5-10.



Gambar 5.10. Menambahkan parameter pekerjaan Stream Analytics

Isi pilihan parameter Anda untuk nilai yang sesuai:

- Nama Pekerjaan
- Berlangganan
- Kelompok Sumber Daya
- Lokasi

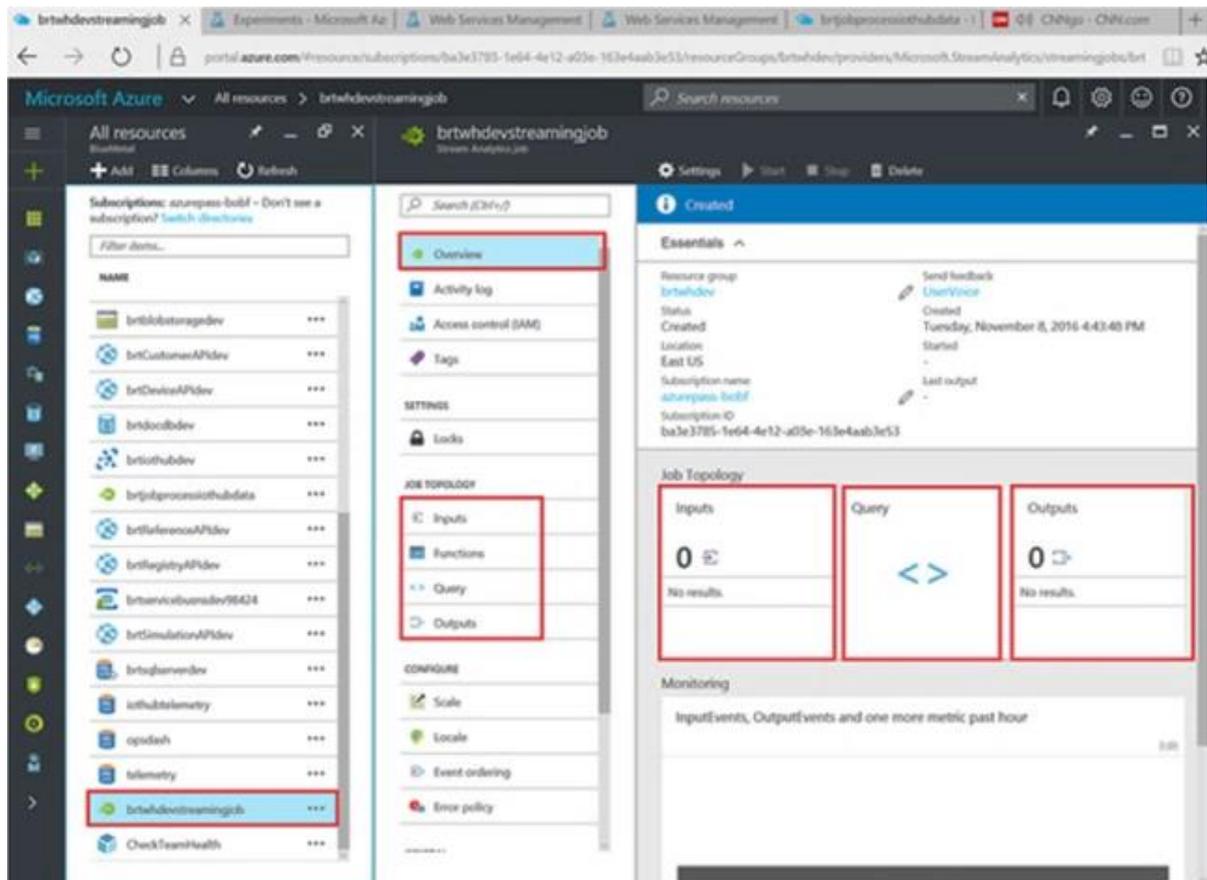
Setelah selesai, klik tombol Buat di bagian bawah layar. Masukan Anda kemudian akan divalidasi dan pekerjaan Azure Streaming Analytics baru akan dibuat setelah jangka waktu singkat. Butuh waktu kurang dari satu menit melalui portal.

Setelah Anda mengonfigurasi pekerjaan Azure Streaming Analytics, Anda dapat menambahkan dan mengonfigurasi komponen tambahan dari pekerjaan tersebut, seperti berikut ini:

- Input: Untuk menentukan aliran data masuk dan data referensi dalam Tugas Azure Streaming Analytics kami.
- Fungsi: Untuk menentukan referensi ke panggilan Azure Machine Learning Web Service. Dalam contoh kita, kita akan memeriksa apakah seorang anggota tim kelelahan hingga mencapai titik kelelahan.
- Keluaran: Untuk menentukan keluaran hasil dari Pekerjaan Azure Streaming Analytics kami ke dalam berbagai format penyimpanan dan platform pengiriman: Panas, Hangat, dan Dingin (dari arsitektur Lambda).

- SQL Query: Akan menggabungkan semua input, fungsi, dan output yang telah ditentukan sebelumnya dalam serangkaian pernyataan SQL untuk memproses aliran data masuk dan mengirimkan hasil yang dihitung ke berbagai tujuan output dan metode pengiriman/pemberitahuan.

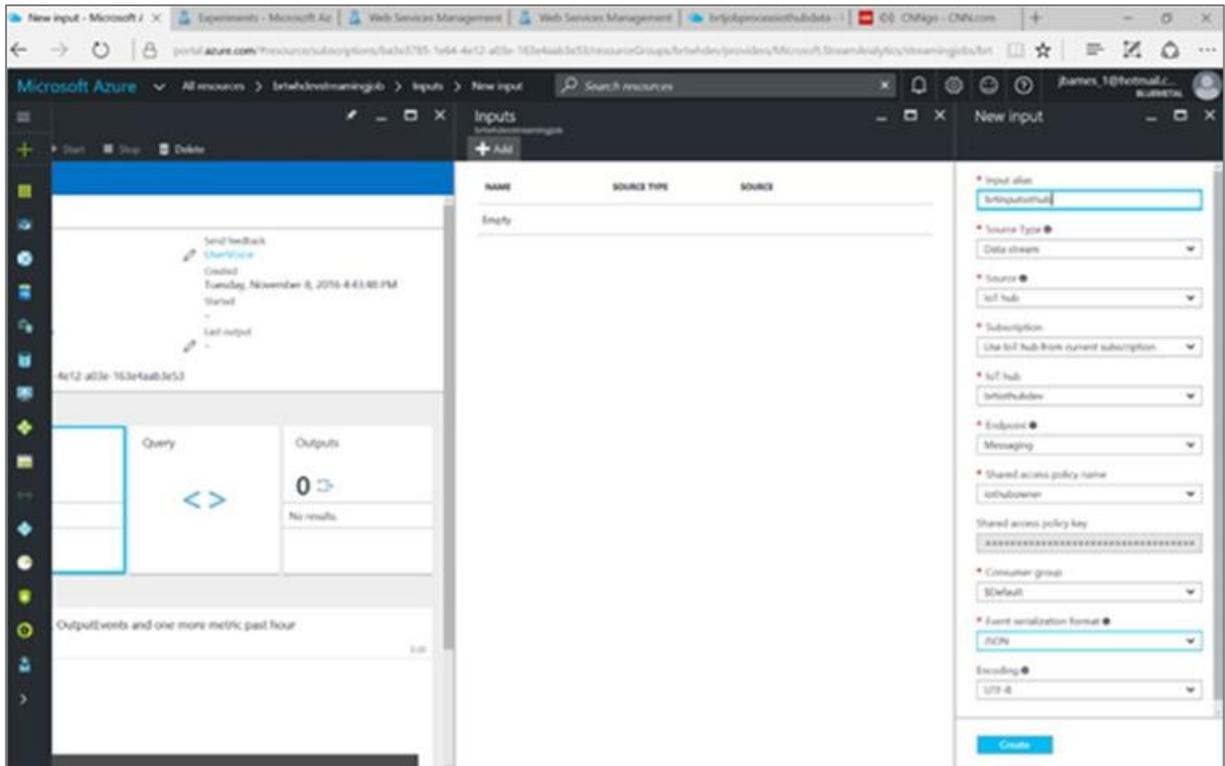
Gambar 5-11 menampilkan tangkapan layar dari pekerjaan Streaming Analytics yang baru dibuat dan input, kueri, dan output terkait.



Gambar 5.11. Pekerjaan Stream Analytics: menambahkan input, fungsi, kueri, dan output

Penyiapan Azure: Pekerjaan Streaming Analytics Dan Input Iot Hub Data Stream

Mari kita mulai dengan mengonfigurasi definisi input untuk menjelaskan aliran data masuk, yang datang melalui konfigurasi IoT Hub di Azure. IoT Hub adalah layanan pemrosesan peristiwa yang memungkinkan masuknya peristiwa dan telemetri ke cloud dalam skala besar, dengan latensi rendah dan keandalan tinggi. Mulailah dengan mengklik gambar masukan lalu + Tambah, seperti yang ditunjukkan pada Gambar 5.12.



Gambar 5.12 Pekerjaan Stream Analytics, menambahkan parameter untuk input dari IoT Hub

Isi pilihan parameter Anda untuk nilai yang sesuai:

- *Input Alias:* Ini akan menjadi nama utama yang digunakan dalam Kueri SQL apa pun untuk merujuk ke aliran input ini.
- *Jenis Sumber:* Atur ke Aliran Data. Perhatikan opsi tambahan untuk Data Referensi.
- *Sumber:* Sumber masukan yang terdeteksi yaitu Aliran Data.
- *Berlangganan:* Langganan Azure untuk sumber input ini.
- *IoT Hub:* Pilih pilihan yang sesuai untuk lingkungan Anda.
- *Titik Akhir:* Pilih Pesan. Opsi lainnya mencakup Pemantauan Operasional.
- *Nama Kebijakan Akses Bersama:* Untuk akses yang didelegasikan dan dibagikan ke sumber daya.
- *Grup Konsumen:* Semua konsumen acara membaca aliran acara melalui partisi dalam grup konsumen.
- *Format Serialisasi Acara:* Tentukan JSON. Pilihan lainnya termasuk Avro dan CSV.
- *Pengkodean:* Tentukan UTF-8.

Setelah selesai, klik tombol Buat di bagian bawah layar. Masukan Anda kemudian akan divalidasi dan definisi masukan Azure Streaming Analytics baru akan dibuat setelah jangka waktu singkat.

Azure Setup: Data Referensi Input Pekerjaan Streaming Analytics

Setelah kita mengonfigurasi jalur data input utama dari IoT Hub, selanjutnya kita akan menentukan jenis definisi input sekunder yang akan digunakan untuk data referensi. Data referensi didefinisikan sebagai data yang lebih statis atau lebih lambat berubah, yang

seringkali dapat di-refresh pada interval waktu yang lebih lama. Prasyarat untuk langkah ini adalah mengunggah file data referensi ke kontainer penyimpanan blob Azure dalam format CSV, JSON, atau Avro.

Untuk implementasi referensi kami, kami akan mengunggah file berformat .CSV yang berisi data referensi anggota tim dengan atribut yang lebih statis seperti nama, tinggi badan, berat badan, jenis kelamin, email, dll. Data referensi ini kemudian dapat digunakan dalam pernyataan GABUNG dalam bahasa Kueri SQL Azure Streaming Analytics untuk memperkaya output data dan memberikan hasil bisnis yang lebih bermakna dan dapat ditindaklanjuti.

Data referensi disimpan di penyimpanan blob Azure dan dimodelkan sebagai urutan blob dalam urutan tanggal/waktu menaik. Ini hanya mendukung penambahan pada akhir urutan dengan menggunakan tanggal/waktu yang lebih besar dari yang ditentukan oleh blob terakhir dalam urutan. Saat ini, pekerjaan Azure Stream Analytics mencari penyegaran blob hanya ketika waktu mesin maju ke waktu yang dikodekan dalam nama blob.

Misalnya, tugas kita akan mencari pola kita yang bernama TeamReferenceData{date}{time}.csv. Ia kemudian akan menemukan file ini: TeamReferenceData2016-11-1217-30.csv dan akan memprosesnya sesegera mungkin tetapi tidak lebih awal dari pukul 17.30 pada tanggal 13 November 2016 zona waktu UTC. Ia tidak akan pernah mencari file dengan waktu yang disandikan lebih awal dari file terakhir yang ditemukan.

Data Referensi INPUT: Contoh Penggunaan SQL

Dimasukkannya data referensi sebagai masukan potensial untuk pekerjaan analisis streaming berarti Anda dapat memanfaatkan SQL JOIN (INNER atau LEFT OUTER) antara aliran dan sumber data referensi untuk memperkaya model data masuk Anda.

Perhatikan bahwa data “referensi” muncul hanya sebagai masukan lain dalam kueri ASA SQL di Daftar 5-11.

Daftar 5-11. SQL JOIN Data Referensi dengan Data Streaming Masuk

```
SELECT myRefData.Name, myStream.Value
FROM myStream
JOIN myRefData ON myStream.myKey = myRefData.myKey
```

Konfigurasi INPUT Analisis Streaming: Definisi Data Referensi

Untuk memulai, navigasikan ke pekerjaan Streaming Analytics Anda di Portal Azure yang sebelumnya ditentukan dalam bab ini dan pilih opsi Input di panel navigasi sebelah kiri. Klik + Tambahkan di bagian atas halaman untuk menambahkan definisi masukan baru. Anda perlu memberikan opsi konfigurasi untuk parameter berikut:

- * Input Alias: Nama unik untuk merujuk pada definisi input ini.
- * Jenis Sumber: Pilih Data Referensi.
- * Berlangganan: Langganan Azure untuk sumber input ini.
- * Akun penyimpanan: Akun penyimpanan Anda (dalam hal ini brtblobstorgedev).
- * Kunci akun penyimpanan: Disalin dari langganan

- * Wadah: tim data-ref
- * Pola jalur: TeamReferenceData[date]{time}.csv
- * Format tanggal: YYYY-MM-DD.
- * Format waktu: HH-mm.
- * Format Serialisasi Acara: Tentukan CSV. Opsi lainnya termasuk JSON dan Avro.
- * Pembatas: Koma (,).
- * Pengkodean: Tentukan UTF-8.

Klik tombol Buat di bagian bawah layar untuk membuat definisi masukan. Tangkapan layar pada Gambar 5-13 menunjukkan parameter masukan untuk definisi referensi masukan.

The screenshot shows a configuration form for a reference data input in Stream Analytics. The fields are as follows:

- Input alias:** (empty text box)
- Source Type:** Reference data
- Subscription:** Use blob storage from current subscription
- Storage account:** brtblobstoragedev
- Storage account key:** (masked with asterisks)
- Container:** refdata-team
- Path pattern:** TeamReferenceData[date]{time}.csv
- Date format:** YYYY-MM-DD
- Time format:** HH-mm
- Event serialization format:** CSV
- Delimiter:** comma (,)
- Encoding:** UTF-8

A blue "Create" button is located at the bottom of the form.

Gambar 5.13. Pekerjaan Stream Analytics: menambahkan parameter untuk input data Referensi

Setelah masukan baru untuk data referensi kami dibuat dan diuji (proses ini hanya memakan waktu sekitar satu menit), Anda kemudian dapat menggunakan definisi masukan baru ini sebagai bagian dari kueri SQL untuk BERGABUNG dengan “data referensi” dengan data sensor yang masuk dari Pusat IoT. Biasanya, data sensor IoT diubah menjadi sangat “ramping” ketika ditransmisikan melalui jaringan, sehingga tidak ada informasi tambahan

selain informasi minimal yang diperlukan untuk menjaga ukuran paket tetap kecil dan biaya transmisi efisien.

Dengan mencocokkan aliran data yang masuk dengan data profil yang lebih lengkap dari data referensi kami, kami dapat memberikan profil data yang lebih lengkap untuk berbagai entitas yang diukur. Kemampuan ini juga memungkinkan mekanisme pendukung keputusan yang lebih terperinci, karena atribut-atribut tambahan yang bermakna kini dapat dibuka dan digunakan untuk memprediksi hasil yang lebih baik.

Azure Setup: Pekerjaan Streaming Analytics, Fungsi Untuk Menghubungi Layanan Web Azure ML

Langkah selanjutnya adalah menentukan alias untuk Layanan Web Azure Machine Learning. Layanan Web akan digunakan untuk memanggil skrip SQL pemrosesan dan digunakan untuk memeriksa apakah anggota tim mungkin menunjukkan gejala bahwa dia mungkin hampir kelelahan fisik.

Azure Machine Learning: Memprediksi Kesehatan Anggota Tim

Model prediktif Azure Machine Learning didasarkan pada serangkaian uji stres fisik yang harus dijalani setiap anggota tim setiap kuartal (setiap tiga bulan). Tujuan dari stress test adalah untuk melacak semua pembacaan sensor yang sama yang dihasilkan oleh anggota tim saat bekerja, namun dalam lingkungan simulasi, stress-test, yang dapat dengan cepat mensimulasikan kondisi kerja yang tidak menguntungkan untuk menyebabkan kelelahan fisik dan kelelahan mental. Kapan pun anggota tim merasa bahwa mereka tidak dapat melanjutkan atau menyelesaikan tugas stress test yang ada dengan aman, mereka cukup menekan tombol Lelah di rompi mereka untuk menghentikan tes dan mendaftarkan atribut fisik mereka pada titik kelelahan. Dengan cara ini, pembacaan sensor fisik utama dikorelasikan dengan respons “kelelahan” rekan satu tim terhadap tes stres. Hal ini memungkinkan kami membuat model “pelatihan” analitik prediktif, yang dapat dilatih menggunakan algoritma Pembelajaran Mesin “klasifikasi biner” untuk memprediksi hasil di masa depan berdasarkan atribut tertentu seperti pernapasan, detak jantung, kecepatan, kecepatan, suhu, denyut nadi, dll.

Memperoleh Data Pelatihan Machine Learning yang Tidak Bias

Perlu juga dicatat bahwa ada keuntungan berbeda lainnya menggunakan lingkungan simulasi “uji stres” sebagai data pelatihan untuk membuat model Pembelajaran Mesin. Hal ini disebabkan oleh fakta bahwa anggota tim mempunyai insentif finansial untuk tetap bekerja berjam-jam dalam kondisi buruk untuk mendapatkan imbalan finansial dan peningkatan kompensasi. Karena alasan ini, anggota tim mungkin enggan menunjukkan kelelahan saat mengerjakan pekerjaan lini produksi nyata untuk memaksimalkan kompensasi finansial mereka. Namun, dengan mengukur respons stress test tim di luar lingkungan kerja produksi yang tidak melibatkan insentif finansial hasil yang lebih akurat dapat diperoleh.

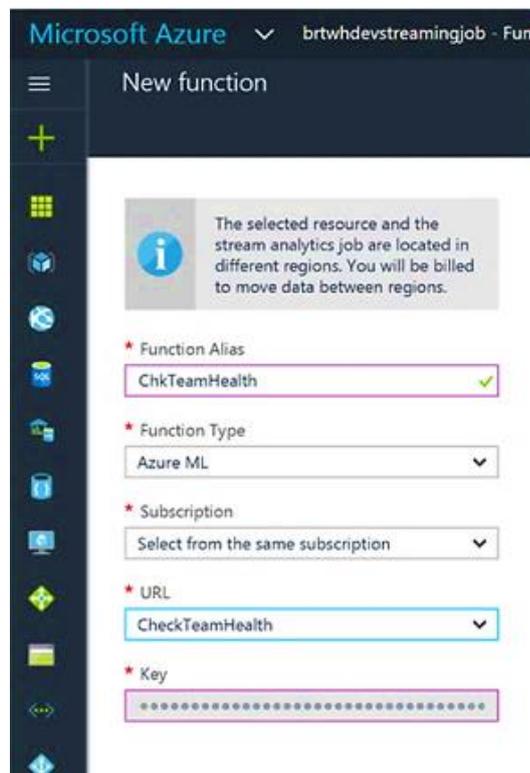
Kunci keberhasilannya adalah memperoleh data pelatihan yang tidak memihak dan akurat yang lebih mirip dengan indikator utama kelelahan dan stres yang dapat diukur, dicatat, dan digunakan untuk melatih Model Azure Machine Learning.

5.8 KONFIGURASIKAN DEFINISI FUNGSI ANALISIS STREAMING

Untuk memulai, navigasikan ke pekerjaan Streaming Analytics Anda di Portal Azure yang sebelumnya ditentukan dalam bab ini dan pilih opsi Fungsi di panel navigasi sebelah kiri. Klik + Tambahkan di bagian atas halaman untuk menambahkan definisi fungsi baru. Anda harus menyediakan hal-hal berikut:

- ❖ Alias Fungsi: Nama unik untuk merujuk pada definisi fungsi ini.
- ❖ Tipe Fungsi: Defaultnya adalah Azure ML. Catatan: opsi fungsi saat ini dibatasi hanya untuk merujuk pada definisi Azure Machine Learning Web Service. Microsoft diperkirakan akan membuka kemampuan ini untuk merujuk pada “Azure Functions” dalam waktu dekat. Fungsionalitas baru ini akan secara dramatis memperluas kemampuan bahasa Kueri SQL Azure Streaming Analytics dengan menggabungkan pemanggilan fungsi ke modul kode kustom.
- ❖ Berlangganan: Langganan Azure yang akan digunakan untuk definisi fungsi.
- ❖ URL: Mengacu pada nama penyebaran Layanan Web Azure Machine Learning.
- ❖ Kunci: Kunci keamanan untuk akses ke penyebaran Azure Machine Learning Web Service.

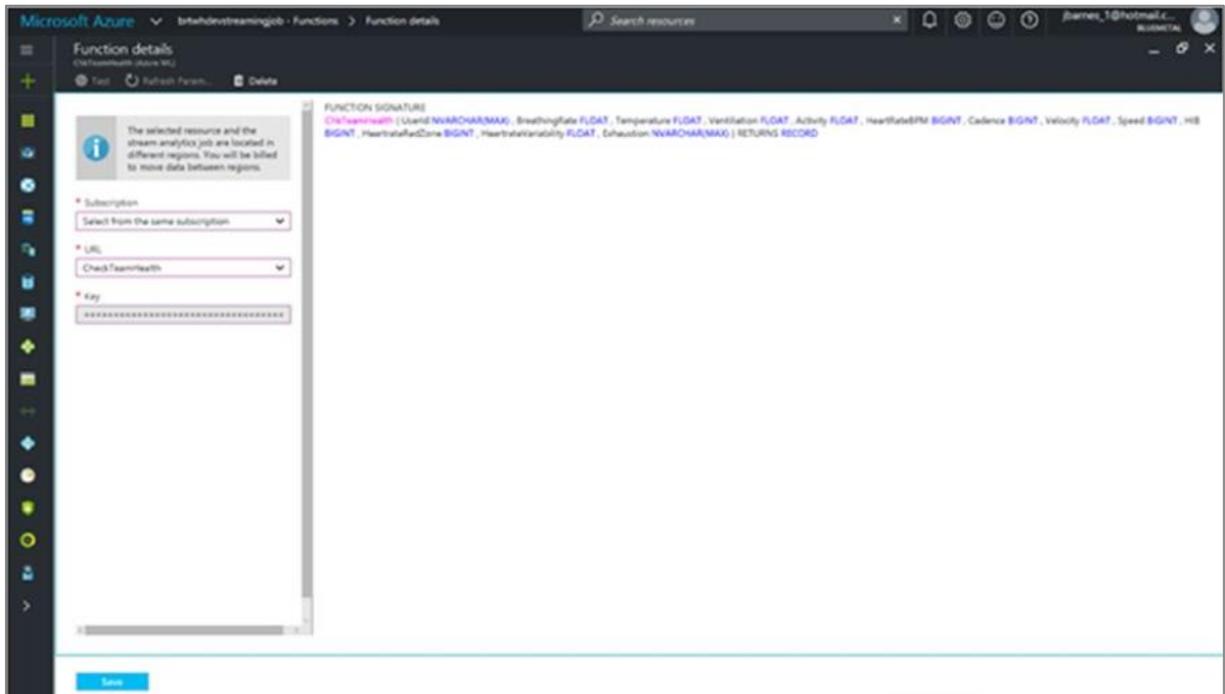
Klik tombol Buat di bagian bawah layar untuk membuat definisi fungsi. Tangkapan layar pada Gambar 5-14 menunjukkan parameter definisi fungsi untuk implementasi referensi.



Gambar 5.14. Tambahkan parameter ke pekerjaan streaming untuk referensi fungsi ke Layanan Web Azure Machine Learning yang diinginkan

Setelah fungsi dibuat dan diuji (proses ini hanya memerlukan waktu sekitar 1 menit), kita kemudian dapat menggunakan definisi fungsi baru ini sebagai bagian dari Kueri SQL untuk memanggil Layanan Web Azure Machine Learning guna memprediksi apakah anggota tim

berisiko terkena kelelahan secara fisik sehingga dapat menimbulkan lingkungan kerja yang tidak aman. Tangkapan layar pada Gambar 5-15 menggambarkan fungsi baru yang kita buat.



Gambar 5.15. Pekerjaan Stream Analytics: parameter fungsi untuk Memanggil Azure ML Web Service CheckTeamHealth

Tanda tangan fungsi juga ditampilkan, yang memperlihatkan 13 parameter dan berbagai jenis bidang yang harus dikonfigurasi dengan benar untuk membuat panggilan fungsi dari kueri SQL Streaming Analytics. Kami akan membahas detail implementasi Layanan Web Azure Machine Learning yang disebut CheckTeamHealth di Bab 8. Sebagai bagian dari latar belakang Bab 8, kami membahas pembuatan model pelatihan Azure ML berdasarkan hasil uji stres dan menggunakan algoritme "Klasifikasi Biner" untuk membuat model prediktif yang dapat menentukan dengan tepat apakah anggota tim mendekati titik fisik. kelelahan. Kami juga akan membahas mekanisme penerapan, pengujian, dan pengelolaan Layanan Web Azure ML.

Penyiapan Azure: Hasil Pekerjaan Streaming Analytics

Sekarang kita telah membuat input dan fungsi untuk pekerjaan Azure Streaming Analytics, sekarang saatnya membuat berbagai output untuk pekerjaan streaming kita. Ini adalah langkah penting dalam proses ini, karena kita akan membuat beberapa keluaran dari satu aliran masukan yang berasal dari IoT Hub.

Salah satu rangkaian keputusan penting dalam menentukan keluaran pekerjaan streaming adalah gagasan arsitektur Lambda yang telah kita ulas sebelumnya di bab ini. Azure Streaming Analytics menyediakan kemampuan untuk menentukan berbagai format output dan opsi penyimpanan yang dapat berhubungan langsung dengan gagasan jalur data Panas, Hangat, dan Dingin:

- ❖ Hot Path: untuk menampilkan informasi penting, real-time, dan dapat ditindaklanjuti, seperti peringatan dan notifikasi. Biasanya, Power BI digunakan sebagai alat visualisasi untuk membuat dasbor dan visualisasi real-time untuk data bisnis penting.
- ❖ Jalur Hangat: Biasanya lebih lambat dibandingkan opsi Jalur Panas, namun memberikan akses yang relatif cepat ke data penting melalui teknologi seperti DocumentDB dan database Azure SQL melalui kueri pengguna terekam atau ad hoc.
- ❖ Jalur Dingin: Ini biasanya merupakan opsi penyimpanan dengan biaya paling rendah dan paling lambat. Skenario kasus penggunaan yang umum adalah menyediakan kemampuan arsip atau penyimpanan yang akan bertindak sebagai sistem pencatatan untuk semua transaksi. Biasanya, penyimpanan blob Azure atau Azure Data Lake digunakan untuk implementasi ini.

Implementasi referensi kami akan berupaya menerapkan ketiga jalur data Lambda.

Pekerjaan Analisis Streaming: Penyimpanan Blob Output Jalur Dingin

Kami akan menggunakan penyimpanan blob Azure untuk opsi keluaran "jalur dingin" kami. Prasyarat untuk langkah ini adalah membuat akun Azure Storage dan kontainer di akun penyimpanan untuk menyimpan data kami.

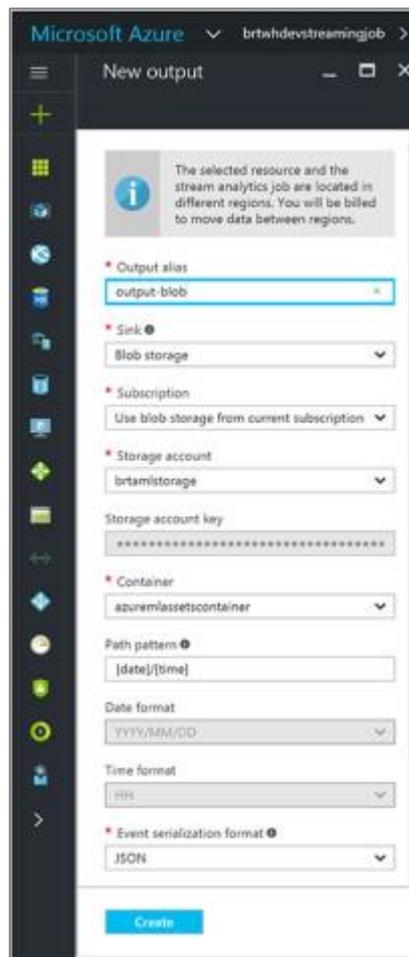
Setelah Anda membuat Akun Azure Storage dan kontainer di akun penyimpanan, Anda memerlukan kunci akses akun penyimpanan untuk langkah berikutnya. Untuk memulai definisi keluaran pertama kami, navigasikan (melalui Portal Azure) ke pekerjaan Azure Streaming Analytics yang kami tentukan sebelumnya di bab ini:

Pilih Output:

- Klik pada + Tambah untuk menambahkan definisi keluaran baru.
- Di bawah parameter Sink, pilih Blob Storage dari daftar drop-down. Tindakan ini akan mengatur kolom tersisa yang perlu Anda isi dengan benar:
- Output Alias: Nama untuk definisi output ini yang akan digunakan dalam query SQL kita.
- Sink : Mengacu pada tujuan keluaran yang dipilih; dalam hal ini, pilih penyimpanan blob.
- Berlangganan: Langganan Azure yang akan digunakan untuk definisi output ini.
- Akun Penyimpanan : Nama akun Azure Storage yang dibuat sebagai prasyarat untuk definisi output ini.
- Kunci Akun Penyimpanan : Kunci Akun yang sesuai untuk akun penyimpanan.
- Container : Nama repositori atau folder akun penyimpanan yang akan menampung data keluaran kita.
- Pola Jalur : Menunjukkan jalur file yang digunakan untuk menemukan blob Anda dalam kontainer yang ditentukan. Dalam definisi pola jalur, Anda dapat memilih untuk menggunakan kata kunci {tanggal} dan {waktu} untuk membantu membuat representasi logis dari keluaran data. Untuk implementasi kami, kami akan menggunakan [tanggal]/[waktu] untuk memisahkan data berdasarkan tanggal dan waktu. Ini akan menyediakan struktur file dalam kontainer blob yang mirip dengan jalur "tahun/bulan/hari/waktu".

- Format Tanggal : Format defaultnya adalah YYYY/MM/DD. Terima defaultnya.
- Format Waktu : Format defaultnya adalah HH. Terima defaultnya.
- Format Serialisasi Acara : Pilihannya adalah JSON, CSV, dan Avro. Kami akan menggunakan JSON untuk definisi keluaran kami, terutama karena aliran data masuk dalam format yang sama (JSON).

Tangkapan layar pada Gambar 5-16 menggambarkan opsi parameter untuk menentukan definisi output untuk penyimpanan blob.



Gambar 5-16. Pekerjaan Stream Analytics: parameter definisi keluaran jalur dingin menggunakan penyimpanan blob

Setelah Anda memasukkan semua parameter yang diperlukan, klik tombol Buat di bagian bawah layar dan definisi keluaran baru akan segera diuji dan ditambahkan ke pekerjaan Azure Streaming Analytics.

Pekerjaan Analisis Streaming: Output Jalur Hangat: Azure SQL Database. Untuk "jalur hangat" kami, kami akan menggunakan database Azure SQL untuk opsi keluaran kami. Prasyarat wajib untuk langkah ini adalah membuat server database Azure, database Azure, dan tabel Azure untuk menyimpan data. Daftar 5-12 adalah skrip Transact-SQL yang akan membuat tabel database Azure SQL bernama `IoTHubSensorReadings`.

Daftar 5-12. Definisi Tabel SQL untuk [IotHubSensorReadings]

```

CREATE TABLE [dbo].[IotHubSensorReadings] (
    [UserId] [char](256) NOT NULL,
    [Age] [float] NOT NULL,
    [Height] [float] NOT NULL,
    [Weight] [float] NOT NULL,
    [HeartRateBPM] [float] NOT NULL,
    [BreathingRate] [float] NOT NULL,
    [Temperature] [float] NOT NULL,
    [Steps] [float] NOT NULL,
    [Velocity] [float] NOT NULL,
    [Altitude] [float] NOT NULL,
    [Ventilization] [float] NOT NULL,
    [Activity] [float] NOT NULL,
    [Cadence] [float] NOT NULL,
    [Speed] [float] NOT NULL,
    [HIB] [float] NOT NULL,
    [HeartRateRedZone] [float] NOT NULL,
    [HeartrateVariability] [float] NOT NULL,
    [Status] [int] NOT NULL,
    [Id] [char](256) NOT NULL,
    [DeviceId] [char](256) NOT NULL,
    [MessageType] [int] NOT NULL,
    [Longitude] [float] NOT NULL,
    [Latitude] [float] NOT NULL,
    [Timestamp] [datetime2](7) NOT NULL,
    [EventProcessedUtcTime] [datetime2](7) NOT NULL,
    [PartitionId] [int] NOT NULL,
    [EventEnqueuedUtcTime] [datetime2](7) NOT NULL
)

```

Perhatikan bahwa nama kolom di tabel `IotHubSensorReadings` sama persis dengan nama kolom aliran data masukan kami. Ini bukanlah langkah yang perlu, tapi seperti yang akan segera Anda lihat, ini membantu mempermudah saat tiba waktunya untuk menulis pernyataan kueri SQL. Untuk memperbaiki ketidakcocokan nama kolom, kita cukup menggunakan klausa T-SQL `AS` untuk mengganti nama kolom masuk menjadi kolom tujuan dalam pernyataan `SELECT` kita.

Setelah artefak prasyarat Azure SQL Database (Server, Database, dan Tabel) telah dibuat, Anda perlu mengumpulkan kredensial untuk database (nama pengguna dan kata sandi) untuk langkah berikutnya dalam proses konfigurasi membuat definisi output pekerjaan streaming. Untuk membuat definisi keluaran pekerjaan Azure Streaming Analytics untuk jalur data "hangat", mulailah dengan menavigasi (melalui Portal Azure) ke definisi pekerjaan streaming yang telah ditentukan sebelumnya dan memilih Output.

Klik + Tambahkan untuk menambahkan definisi keluaran baru:

- Di bawah parameter Sink, pilih database SQL dari daftar drop-down. Tindakan ini akan mengatur kolom tersisa yang perlu Anda isi dengan benar:
- Output Alias: Nama untuk definisi output ini yang akan digunakan dalam query SQL kita.
- Sink : Mengacu pada tujuan keluaran yang dipilih; dalam hal ini, pilih database SQL.
- Berlangganan: Langganan Azure yang akan digunakan untuk definisi output ini.
- Database : Nama database Azure SQL yang kita buat sebagai prasyarat di awal bab ini.
- Nama Server: Nama Azure SQL Server yang kita buat sebagai prasyarat di awal bab ini. Perhatikan bahwa konvensi penamaan harus mengikuti format berikut:
- <Nama Server Anda>.database.windows.net
- Nama Pengguna: Kredensial nama pengguna untuk server database Azure SQL.
- Kata Sandi: Kredensial kata sandi untuk server database Azure SQL.
- Tabel: Nama tabel database Azure SQL yang kita buat sebagai prasyarat di awal bab ini. Contoh implementasi referensi kami menggunakan tabel bernama IoTHubSensorReadings.

Tangkapan layar yang ditunjukkan pada Gambar 5.17 menggambarkan opsi parameter untuk menentukan definisi keluaran untuk database SQL.

Gambar 5.17. Pekerjaan Stream Analytics untuk parameter definisi keluaran jalur hangat menggunakan database SQL

Setelah Anda memasukkan semua parameter yang diperlukan, klik tombol Buat di bagian bawah layar dan definisi keluaran baru akan segera diuji dan ditambahkan ke pekerjaan Azure Streaming Analytics.

Pekerjaan Analisis Streaming: Jalur Panas OUTPUT Power BI

Untuk "jalur panas" kami, kami akan menggunakan Power BI untuk opsi keluaran kami. Seperti yang dinyatakan sebelumnya, Power BI adalah alat visualisasi data kaya yang memungkinkan Anda membuat dasbor dan visualisasi yang kaya dan real-time untuk data bisnis penting. Power BI menampilkan dasbor yang interaktif dan dapat dibuat serta diperbarui dari berbagai sumber data secara real-time. Pada bab selanjutnya dalam buku ini, kita akan menggunakan Power BI sebagai dasbor operasional real-time seperti halnya dasbor pada mobil. Ini menampilkan informasi penting tentang kendaraan, seperti kecepatannya, tingkat bahan bakarnya, atau suhu oli. Dalam aplikasi referensi kami, kami akan memantau data kesehatan dan aktivitas anggota tim secara real time. Prasyarat wajib untuk langkah ini adalah menyiapkan dan mengonfigurasi pengguna Azure Active Directory yang dapat kita gunakan untuk mengautentikasi Power BI selama proses definisi output. Azure Active Directory adalah bagian dasar dari proses Autentikasi Power BI dan menyimpan pengguna, grup, dan domain selain pengaturan dan opsi konfigurasi lainnya. Aplikasi Power BI terintegrasi dengan Azure Active Directory (AAD) untuk menyediakan proses masuk dan otorisasi yang aman untuk aplikasi Power BI Anda. Untuk mengintegrasikan aplikasi Power BI dengan Azure Active Directory, Anda perlu mendaftarkan detail aplikasi dengan Azure AAD melalui Portal Manajemen Azure. Untuk mendaftar layanan Power BI, Azure Active Directory Anda harus memiliki setidaknya satu pengguna organisasi. Lihat tautan berikut untuk detail tentang membuat penyewa Azure Active Directory:

Setelah prasyarat untuk opsi Output Power BI telah dikonfigurasi (pengguna/kata sandi Azure Active Directory dan pendaftaran di situs Power BI), langkah selanjutnya adalah memulai pembuatan opsi output Power BI. Pembuatan opsi keluaran Power BI pada dasarnya merupakan proses dua langkah:

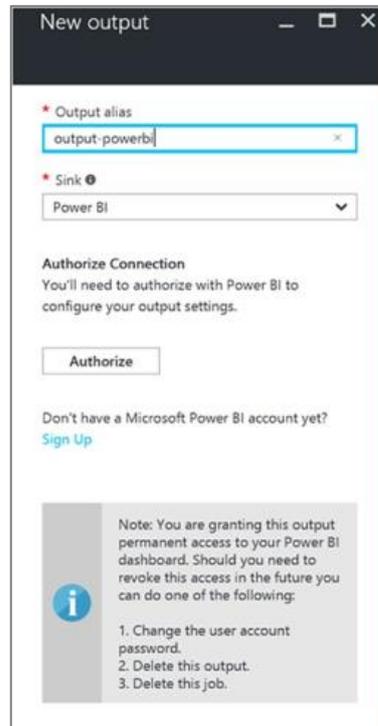
1. Otentikasi dengan Power BI. Saat melakukan langkah ini, Azure Streaming Analytics menghasilkan dan menyimpan kredensial tersebut ketika pekerjaan sedang berjalan dan memasukkan data real-time ke aplikasi dasbor Power BI Anda.
2. Berikan rincian yang diperlukan untuk definisi keluaran.

Untuk membuat definisi keluaran pekerjaan Azure Streaming Analytics untuk jalur data "panas", mulailah dengan menavigasi (melalui Portal Azure) ke definisi pekerjaan streaming yang telah ditentukan sebelumnya. Pilih Output.

Klik + Tambahkan untuk menambahkan definisi keluaran baru:

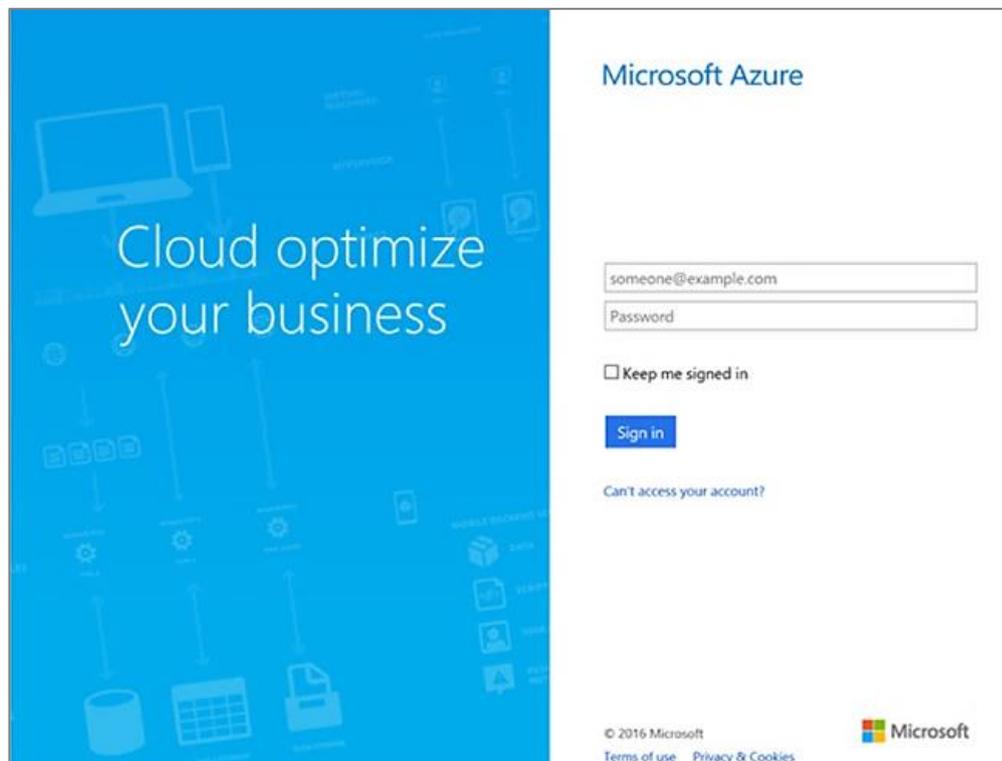
- Di bawah Parameter Sink: Pilih Power BI dari daftar drop-down. Tindakan ini akan mengatur dengan benar sisa kolom untuk Anda isi.
- Output Alias: Nama untuk definisi output ini yang akan digunakan dalam SQL Query kita.
- Sink: Mengacu pada tujuan keluaran yang dipilih; dalam hal ini, pilih Power BI.
- Otorisasi Koneksi: Dengan mengklik tombol Otorisasi, Anda akan disajikan layar otorisasi untuk mengautentikasi terhadap layanan Power BI.

Seperti disebutkan sebelumnya, kredensial ini di-cache dan digunakan untuk akses real-time untuk memasukkan data baru ke dasbor Power BI Anda. Gambar 5-18 menampilkan tangkapan layar parameter yang diperlukan untuk langkah pertama.



Gambar 5.18. parameter definisi keluaran jalur panas, autentikasi Power BI

Setelah mengklik tombol Otorisasi, Anda akan melihat layar seperti pada Gambar 5-19 untuk memasukkan kredensial Power BI Anda.

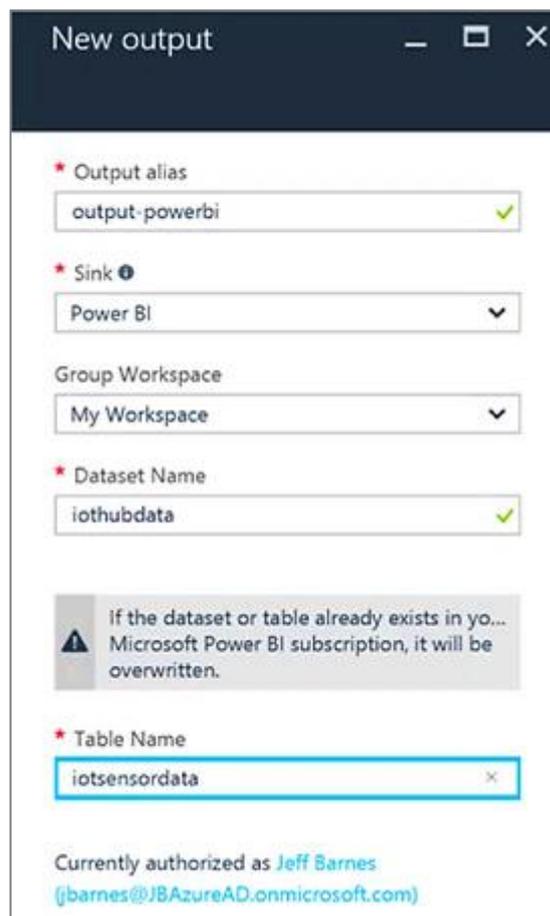


Gambar 5.19. Pekerjaan Stream Analytics: parameter definisi keluaran jalur panas Layar Otentikasi Power BI AAD

Setelah Anda mengautentikasi Power BI dengan kredensial AAD Anda, Anda kemudian akan mendapatkan akses ke parameter konfigurasi lainnya yang diperlukan untuk menyelesaikan definisi output.

- *Output Alias*: Nama untuk definisi output ini yang akan digunakan dalam query SQL kita.
- *Sink*: Mengacu pada tujuan keluaran yang dipilih; dalam hal ini, pilih Power BI.
- *Ruang Kerja Grup*: Merujuk pada ruang kerja di penyewa Azure Power BI tempat himpunan data output akan dibuat.
- *Nama Himpunan Data*: Nama himpunan data deskriptif untuk output Power BI yang akan digunakan sebagai referensi saat membuat dasbor dan visualisasi Power BI.
- *Nama Tabel*: Nama tabel deskriptif dalam himpunan data untuk output Power BI yang akan digunakan sebagai referensi saat membuat dasbor dan visualisasi Power BI.

Gambar 5-20 menggambarkan parameter konfigurasi tambahan yang terlihat setelah koneksi Power BI diautentikasi.



The screenshot shows a 'New output' configuration window with the following fields and values:

- Output alias**: output-powerbi (with a green checkmark)
- Sink**: Power BI (with a dropdown arrow)
- Group Workspace**: My Workspace (with a dropdown arrow)
- Dataset Name**: iothubdata (with a green checkmark)
- Table Name**: iotsensordata (with a red 'x' icon)

A warning message is displayed: "If the dataset or table already exists in your Microsoft Power BI subscription, it will be overwritten." At the bottom, it says "Currently authorized as Jeff Barnes (jbarnes@JBAzureAD.onmicrosoft.com)".

Gambar 5-20. Pekerjaan Stream Analytics: parameter definisi keluaran jalur panas
Parameter tambahan Power BI setelah autentikasi

Setelah Anda memasukkan semua parameter yang diperlukan, klik tombol **Buat** di bagian bawah layar dan definisi keluaran baru akan segera diuji dan ditambahkan ke pekerjaan

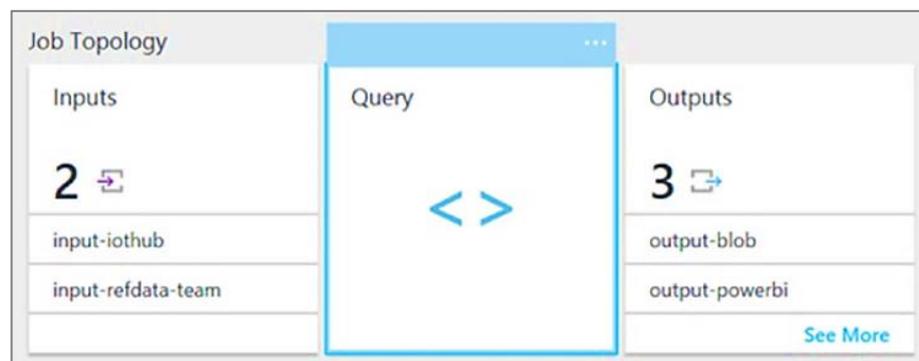
Azure Streaming Analytics. Kami akan mengeksplorasi penggunaan Power BI secara lebih rinci di Bab 9.

Pekerjaan Azure Streaming Analytics: Input dan Output

Pada titik ini, Anda telah membuat komponen pekerjaan Azure Streaming Analytics berikut ini:

- Masukan:
 - Pusat IoT
 - Data referensi
- Fungsi:
 - Layanan Web Azure ML – ChkTeamHealth()
- Keluaran:
 - Penyimpanan gumpalan
 - Basis data Azure SQL
 - Kekuatan BI

Pada titik ini, pekerjaan Azure Streaming Analytics Anda akan terlihat seperti tangkapan layar pada Gambar 5-21.



Gambar 5-21. Pekerjaan Stream Analytics: dua definisi masukan dan tiga definisi keluaran

Penyiapan Azure: Query Sql Pekerjaan Analisis Streaming

Sekarang kita telah membuat input, fungsi, dan output untuk pekerjaan Azure Streaming Analytics, sekarang saatnya untuk menggabungkan semuanya menggunakan bahasa ASA SQL Query. Dalam implementasi referensi, kami akan menggunakan sebagian besar opsi dan fitur yang tersedia dengan Azure Streaming Analytics untuk membantu mengilustrasikan betapa mudahnya menyusun mesin penyerapan berkinerja tinggi yang dapat dengan mudah ditingkatkan guna menambah kemampuan untuk mengubah skala besar volume data menjadi data real-time dan dapat ditindaklanjuti.

Kueri SQL yang akan Anda terapkan akan memungkinkan Anda menggabungkan dua definisi masukan bersama dengan definisi fungsi Azure ML dan kemudian mengisi tiga tujuan jalur data keluaran terpisah untuk data streaming kami. Ketiga jalur tersebut akan berkorelasi dengan panduan arsitektur Lambda kami untuk jalur data Panas, Hangat, dan Dingin:

- *Hot Path:* Kami menentukan definisi output menggunakan Power BI, yang memungkinkan kami memperbarui dasbor dan visualisasi real-time dengan data sensor ditambah hasil panggilan Layanan Web Azure Machine Learning melalui definisi fungsi.
- *Warm Path:* Kami akan menargetkan database Azure SQL yang memungkinkan kami menambah, memperbarui, dan mengkueri data dari banyak klien.
- *Cold Path:* Kita akan menggunakan penyimpanan blob Azure untuk mengarsipkan data dalam format aslinya ditambah hasil tambahan GABUNG pada INPUT dengan definisi data referensi untuk data kesehatan referensi anggota tim.

Untuk membuat definisi Kueri SQL pekerjaan Azure Streaming Analytics, mulailah dengan menavigasi (melalui Portal Azure) ke definisi pekerjaan streaming yang ditentukan sebelumnya. Pilih Kueri.

- Anda akan disajikan jendela SQL Query Editor.
- Masukkan teks Kueri SQL seperti yang ditunjukkan pada Gambar 5-14 untuk mengimplementasikan output “jalur panas” ke Power BI. Perhatikan panggilan ke ChkTeamHealth(). Ini memanggil Layanan Web Azure ML untuk memprediksi tingkat kelelahan anggota tim.

Hot Path Kueri SQL: Panggil Azure ML lalu Keluarkan ke Power BI

Daftar 5-13. Kueri ASA SQL yang Mengimplementasikan Panggilan Fungsi ke Layanan Web Azure Machine Learning

```

- *****
- * HOTPATH
- * Invoke Machine Learning As a Function "ChkTeamHealth()"
- * Via ASA SQL Subquery
- * then output to Power BI (Hot) & BLOB Storage (COLD) Storage
- *****
WITH [subquery] AS
(
  SELECT UserId, ChkTeamHealth(
    UserId,
    BreathingRate,
    Temperature,
    Ventilization,
    Activity,
    HeartRateBPM,
    Cadence,
    Velocity,
    Speed,
    HIB,
    HeartrateRedZone,
    HeartrateVariability,

```

```

        "N")
    as result from [input-iotHub]
    TIMESTAMP BY [Timestamp]
)
SELECT UserId,
    result.[BreathingRate],
    result.[Temperature],
    result.[Ventilization],
    result.[Activity],
    result.[HeartRateBPM],
    result.[Cadence],
    result.[Velocity],
    result.[Speed],
    result.[HIB],
    result.[HeartrateRedZone],
    result.[HeartrateVariability],
    result.[Scored labels],
    result.[Scored Probabilities],
    result.[Timestamp]
INTO [output-powerbi]
FROM subquery

```

Sangat disarankan agar Anda mengatasi kesalahan tata bahasa apa pun dalam skrip SQL Anda sebelum melanjutkan. Masalah paling umum adalah kesalahan ejaan antara definisi input/fungsi/output Anda dan definisi dalam skrip implementasi referensi yang disediakan sebagai kode contoh dalam buku ini.

Warm Path Kueri SQL: Output ke Azure SQL Database

Masukkan teks kueri SQL seperti yang ditunjukkan pada Daftar 5-14 untuk mengimplementasikan output "jalur hangat" ke database Azure SQL.

- Catatan Nama bidang dalam definisi tabel SQL dicocokkan dengan nama kolom JSON yang masuk untuk mempercepat upaya pengembangan.

Daftar 5-14. Kueri ASA SQL yang Dihasilkan ke Azure SQL Database

```

- *****
- * WARM Path
- * OUTPUT to Azure SQL DB
- *****
SELECT
    UserId,
    Age,
    Height,
    Weight,
    HeartRateBPM,

```

```

    BreathingRate,
    Temperature,
    Steps,
    Velocity,
    Altitude,
    Ventilization,
    Activity,
    Cadence,
    Speed,
    HIB,
    HeartRateRedZone,
    HeartrateVariability,
    Status,
    Id,
    DeviceId,
    MessageType,
    Longitude,
    Latitude,
    [Timestamp],
    EventProcessedUtcTime,
    PartitionId,
    EventEnqueuedUtcTime

INTO [output-sqlldb]

FROM [input-iothub]
TIMESTAMP BY [Timestamp]

```

Cold Path Kueri SQL: Output ke Azure SQL Database

Masukkan teks Kueri SQL yang diperlihatkan dalam Daftar 5-15 untuk mengimplementasikan output “Cold Path” ke penyimpanan blob Azure.

Daftar 5-15. ASA SQL Query yang JOIN dengan Data Referensi dan Output ke Azure Blob Storage

```

- *****
- * COLD Path
- * OUTPUT ALL incoming fields into to Azure BLOB Storage
- * JOIN on Reference Data
- *****
SELECT
    IH.UserId,
    IH.Age,
    IH.Height,
    IH.Weight,
    IH.HeartRateBPM,
    IH.BreathingRate,

```

```

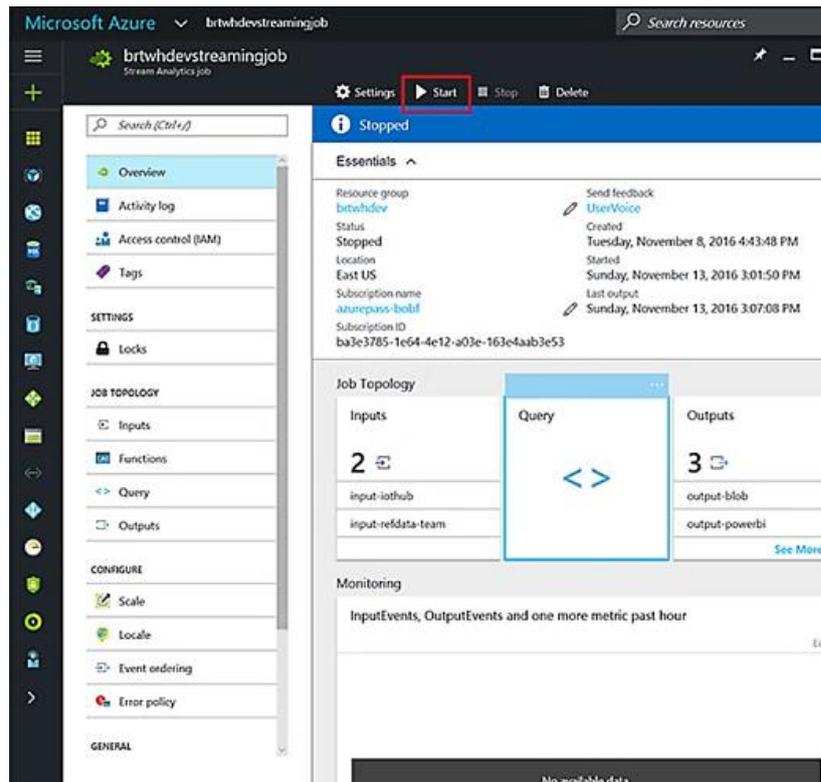
    IH.Temperature,
    IH.Steps,
IH.Velocity,
    IH.Altitude,
    IH.Ventilization,
    IH.Activity,
    IH.Cadence,
    IH.Speed,
    IH.HIB,
    IH.HeartRateRedZone,
    IH.HeartrateVariability,
    IH.Status,
    IH.Id,
    IH.DeviceId,
    IH.MessageType,
    IH.Longitude,
    IH.Latitude,
    IH.[Timestamp],
    IH.EventProcessedUtcTime,
    IH.PartitionId,
    IH.EventEnqueuedUtcTime,
    RF.healthInformation__age,
    RF.healthInformation__height,
    RF.healthInformation__weight,
    RF.healthInformation__gender,
    RF.healthInformation__race
INTO
[output-blob]
FROM
[input-iothub] IH
TIMESTAMP BY [Timestamp]
JOIN
    [input-refdata-team] RF
    ON IH.UserId = RF.id

```

Klik tombol Simpan di panel navigasi kiri atas di Portal Azure untuk menyimpan kueri SQL.

5.9 QUERY SQL STREAMING ANALYTICS

Sekarang langkah terakhir yang tersisa untuk menyelesaikan pekerjaan Azure Streaming Analytics telah selesai, sekarang saatnya untuk memulai pekerjaan streaming Analytics kami dan melihat hasil keluaran yang diharapkan. Navigasikan ke pekerjaan analitik streaming yang Anda buat sebelumnya dan klik tombol |> Mulai di bagian tengah atas Portal Azure, seperti yang ditunjukkan pada Gambar 5-22.



Gambar 5-22. Mulai pekerjaan Analisis Aliran

Menggunakan Simulator Perangkat untuk Menguji Aliran Pekerjaan Azure Streaming Analytics

Setelah pekerjaan streaming Azure berjalan, Anda dapat menjalankan kode C# Device Simulator dari repositori GitHub untuk menguji pekerjaan Azure Streaming Analytics. Setelah diunduh, Anda akan menemukan Solusi Visual Studio di folder `\brt\devices\Simulator`. Ini adalah aplikasi Konsol C# .NET sederhana yang membuat himpunan data informasi sensor yang komprehensif dan mengirimkannya ke koneksi Azure IoT Hub.

Setelah Anda menemukan solusi di repositori, Anda kemudian dapat memuat, membangun, dan menjalankan solusi (Simulator.sln). Anda akan melihat jendela konsol yang mirip dengan yang ada pada Gambar 5-23.

```

Simulating teammate1
Profile Id : 058d7760-e868-434e-aa1d-bbc438d5ebb6
Device Id : 339810bc-e868-4ff7-b648-72e1a161695
Simulating teammate2
Profile Id : c3ec2fe4-6789-4d3e-8446-852856ce025c
Device Id : 9369763c-39a8-4e6b-9795-5683b62938a1
Simulating teammate3
Profile Id : 663d1c24-9afd-4550-8d57-7d869b0918ca
Device Id : 94697898-627f-447f-93f7-b55a76edfae9
Simulating teammate4
Profile Id : 7809cfad-5cc0-4ce1-97ca-d4ed261a8e94
Device Id : 542f805c-e2fa-454e-9e50-33428ffd5a30
Simulating teammate5
Profile Id : 2d49b721-e157-4dcd-ba50-4de2f0323489
Device Id : d6a8aee0-0b95-43ba-a663-d0116dbc02f6
Simulating teammate6
Profile Id : 8b3ac2c9-807b-4470-9611-1ec362250654
Device Id : 59ba9390-ac3e-4512-a2b3-e4781abf763d
Simulating teammate7
Profile Id : 42105a24-00f7-4f5a-aca3-3001aa14eb75
Device Id : 8d9a836a-96f1-4960-a009-6e3c02adfee3
Simulating teammate8
Profile Id : aalc0209-d542-4535-81f5-ec90d0316935
Device Id : c48a7162-501d-4cde-bca9-f36062f4e701
Simulating teammate9
Profile Id : 698607de-a022-4ed7-9458-8ee36ad49144
Device Id : 67226a07-7490-4cf8-b13d-eec8d653ca18
Simulating teammate10
Profile Id : 82923b70-34cf-476c-a8b5-dc7bed2f17c6
Device Id : e8f3e93b-568e-468b-a7a4-937744f52fc2
Simulating teammate11
Profile Id : 0f15ad8d-272e-4981-9829-b34a284a526a
Device Id : 87995524-7c7f-420e-a7bf-f94c6fe06f4e
Simulating teammate12
Profile Id : 870ad05d-996d-4fd0-b074-f7ff4fb4cc67
Device Id : 130879df-8cae-40c2-940f-1bc682c806c9
Simulating teammate13
Profile Id : a87747ef-e6ab-4b3f-a9b1-0889299a92f9
Device Id : b2aac61e-3dd7-435d-ae79-65b76bc36f0e
Simulating teammate14
Profile Id : ce177590-df50-43f4-bfc0-7c54c53ba9e9
Device Id : 7002b8a9-5317-4d6e-950f-357495c72cad
Simulating teammate15
Profile Id : d7940971-ddce-4535-b738-3d94b00d2b24
Device Id : 563cc9bd-a061-499f-8362-78511fe959a3

```

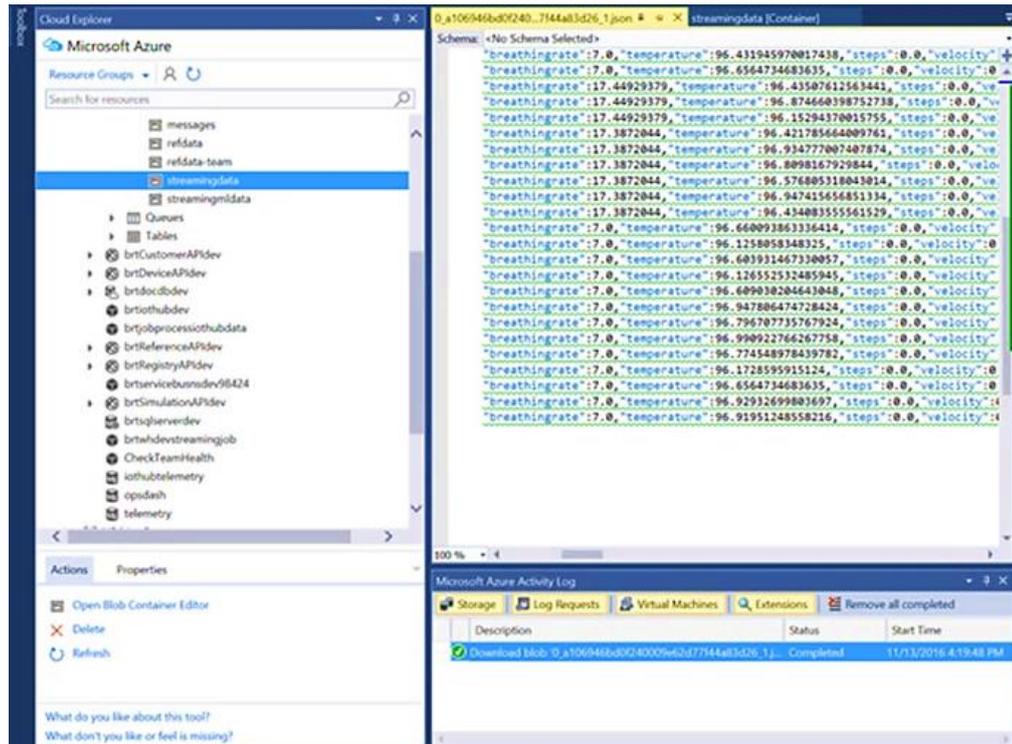
Gambar 5-23. Keluaran Simulator Perangkat Konsol C# .NET

Anda juga harus mulai melihat data mengalir ke berbagai tujuan output seperti yang ditentukan dalam pekerjaan Azure Streaming Analytics.

Verifikasi keluaran pekerjaan Azure Streaming Analytics: Azure Blob Storage

Anda dapat dengan mudah memeriksa dan memverifikasi tujuan keluaran blob Azure menggunakan alat seperti Cloud Explorer di Visual Studio 2015 atau dengan menggunakan “Azure Storage Explorer,” yang dapat diunduh dari <http://azurestorageexplorer.codeplex.com>.

Gambar 5-24 memberikan contoh tangkapan layar hasil keluaran blob Azure saat dilihat dengan alat Visual Studio Cloud Explorer.

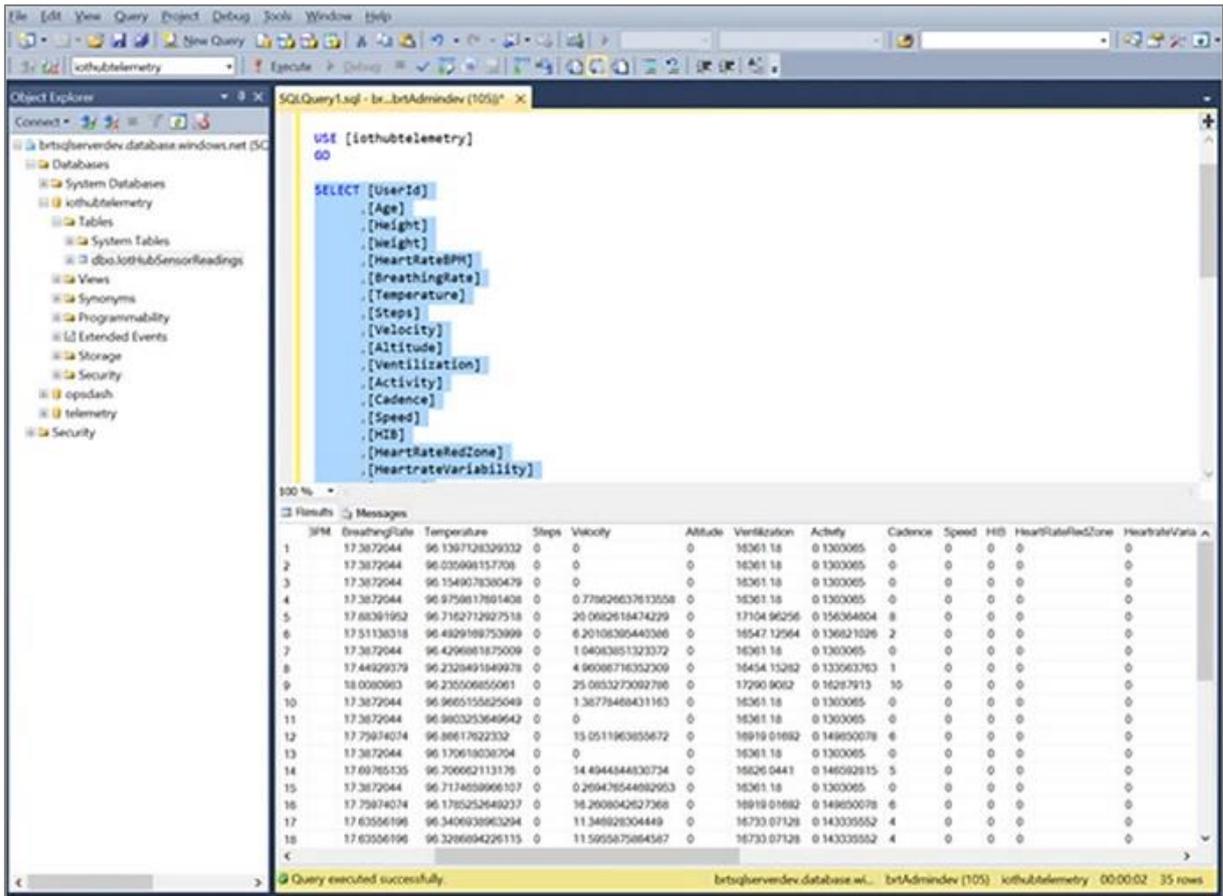


Gambar 5-24. Alat Cloud Explorer Visual Studio 2015 untuk memverifikasi hasil keluaran blob Azure dari pekerjaan THE Streaming Analytics

Verifikasi keluaran pekerjaan Azure Streaming Analytics: Azure SQL Database

Untuk memverifikasi hasil output database Azure SQL dari pekerjaan Azure Streaming Analytics, ANDA dapat menggunakan alat seperti SQL Server Management Studio (SSMS). Anda dapat mengunduhnya dari tautan berikut:

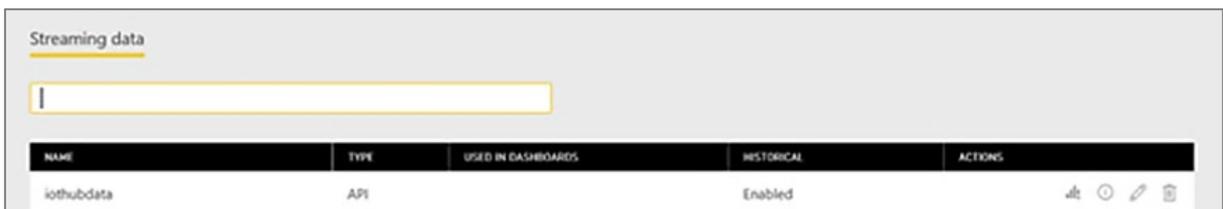
Gambar 5-25 memberikan contoh tangkapan layar hasil keluaran database Azure SQL ketika pernyataan SELECT dijalankan terhadap tabel keluaran SQL dan dilihat dalam jendela Kueri SQL Server Management Studio.



Gambar 5-25. Menggunakan SQL Server Management Studio untuk memverifikasi hasil output database Azure SQL dari pekerjaan Streaming Analytics

Verifikasi keluaran pekerjaan Azure Streaming Analytics: Power BI

Untuk memverifikasi output pekerjaan Streaming Analytics ke Power BI, Anda dapat masuk ke situs web Power BI di tautan berikut menggunakan ID organisasi Azure Active Directory Anda: <https://powerbi.microsoft.com/en-us>. Setelah Anda masuk ke Portal Web Power BI, Anda dapat menavigasi ke bawah panel navigasi sisi kiri ke area topik bernama Kumpulan Data dan memilih opsi untuk Kumpulan Data Streaming. Setelah mengklik link tersebut, Anda akan melihat daftar kumpulan data streaming yang tersedia. Carilah yang bernama iotHubdata, yang mewakili output dari pekerjaan THE Azure Streaming Analytics, seperti yang ditunjukkan pada Gambar 5-26.



Gambar 5-26. Bekerja dengan himpunan data streaming keluaran "jalur panas" untuk Power BI guna memverifikasi hasil keluaran dari pekerjaan Analisis Streaming

Untuk cakupan Power BI yang lebih mendalam sebagai jalur output untuk Azure Streaming Analytics, lihat Bab 9.

5.10 RINGKASAN

Bab ini membahas semua kemampuan dasar Azure Streaming Analytics. Anda mempelajari bagaimana Anda dapat dengan mudah membuat pekerjaan analisis streaming yang memungkinkan Anda memanfaatkan semua atribut positif yang harus dimiliki oleh arsitektur data Lambda modern, termasuk jalur data “panas”, “hangat”, dan “dingin” untuk memberikan hasil bisnis yang maksimal. Anda juga memeriksa manfaat menggunakan layanan PaaS yang terkelola sepenuhnya seperti Azure Streaming Analytics, dibandingkan membangun lingkungan virtual Anda sendiri di Azure menggunakan gambar Mesin Virtual Linux dan kombinasi banyak alat dan utilitas sumber terbuka.

Terakhir, Anda menerapkan pengetahuan tentang Azure Streaming Analytics ke referensi Arsitektur IoT dan membuat dua definisi input satu untuk peristiwa IoT Hub dan yang kedua untuk data referensi untuk informasi terkait kesehatan anggota tim. Selanjutnya, kami membuat definisi FUNCTION untuk mewakili panggilan Azure Machine Learning Web Service yang kami gunakan dalam ASA SQL Query kami. Kami kemudian membuat tiga definisi keluaran yang mewakili jalur data Panas, Hangat, dan Dingin menggunakan parameter definisi keluaran untuk memperbarui platform data Azure yang sesuai Power BI untuk “Panas”, Azure SQL Database untuk “Hangat”, dan Azure blob STORAGE untuk penyimpanan “Dingin” .

Seperti yang dapat dilihat dengan mudah dari bab ini, Azure Streaming Analytics dapat memainkan peran penting dalam penyerapan, pengorganisasian, dan orkestrasi transaksi sensor IoT. Lingkungan memudahkan untuk memulai namun sangat kuat dan fleksibel serta dapat dengan mudah diskalakan untuk menangani jutaan transaksi per detik. Mesin analisis aliran yang kuat sangat penting bagi keberhasilan perusahaan modern yang ingin menjalankan bisnis dengan kecepatan Internet.

BAB 6

PEMROSESAN BATCH DENGAN DATA FACTORY DAN DATA LAKE STORE

Bab ini membahas penggunaan Azure Data Factory dan Azure Data Lake, termasuk di mana, mengapa, dan bagaimana teknologi ini sesuai dengan kemampuan bisnis modern yang berjalan dengan kecepatan Internet. Ini pertama-tama mencakup aspek teknis dasar dan kemampuan Azure Data Factory dan Azure Data Lake. Setelah itu, bab ini mengimplementasikan tiga fungsi utama untuk implementasi referensi:

- Memperbarui data referensi yang kami gunakan untuk pekerjaan Azure Stream Analytics. Seperti yang Anda ingat, kami menggunakan data referensi dalam kueri ASA SQL JOIN untuk mengumpulkan data kesehatan anggota tim tambahan.
- Melatih kembali Model Azure Machine Learning untuk memprediksi tingkat kesehatan dan kelelahan anggota tim. Data ini akan didasarkan pada tes stres medis terkini yang diberikan kepada anggota tim secara berkala.
- Memindahkan data dari penyimpanan blob Azure ke Azure Data Lake. Langkah ini mempersiapkan implementasi referensi analitik Data Lake, yang merupakan topik Bab 7.

6.1 IKHTISAR PABRIK DATA AZURE

Azure Data Factory memenuhi kebutuhan penting dalam lingkungan pemrosesan Big Data modern apa pun. Hal ini dapat dilihat sebagai tulang punggung operasi data apa pun, karena Data Factory menyediakan kemampuan inti penting yang diperlukan untuk menjalankan fungsi transformasi data perusahaan. Ini termasuk:

- *Penyerapan dan Persiapan Data:* Dari berbagai sumber; kombinasi sumber data lokal dan berbasis cloud.
- *Transformasi dan Analisis:* Menjadwalkan, mengatur, dan mengelola proses transformasi dan analisis data.
- *Publikasikan dan Konsumsi:* Kemampuan untuk mengubah data mentah menjadi data jadi yang siap dikonsumsi oleh alat BI atau aplikasi seluler.
- *Pemantauan dan Manajemen:* Visualisasikan, pantau, dan kelola pergerakan data dan jalur pemrosesan untuk mengidentifikasi masalah dengan cepat dan mengambil tindakan cerdas. Kemampuan peringatan untuk memantau kesehatan layanan pemrosesan data secara keseluruhan.
- *Manajemen Sumber Daya yang Efisien:* Menghemat waktu dan uang Anda dengan mengotomatiskan jalur transformasi data dengan sumber daya dan manajemen cloud sesuai permintaan.

Azure Data Factory adalah layanan integrasi data berbasis cloud yang mengatur dan mengotomatiskan pergerakan dan transformasi data. Anda dapat membuat solusi integrasi data menggunakan Azure Data Factory yang dapat menyerap data dari berbagai penyimpanan

data (menangani lokal dan berbasis cloud), mengubah dan memproses data, lalu mempublikasikan hasil pemrosesan ke berbagai penyimpanan data keluaran.

Layanan Azure Data Factory adalah layanan berbasis cloud yang dikelola sepenuhnya yang memungkinkan Anda membuat “jalur” pemrosesan data yang dapat memindahkan dan mengubah data. Data Factory memiliki kemampuan untuk menjalankan fungsi ETL (Extract-Transform-Load) yang sangat canggih dan dapat disesuaikan pada data saat data bergerak melalui berbagai tahapan dalam alur pemrosesan. “Pipeline” pemrosesan data ini kemudian dapat dijalankan pada jadwal tertentu (seperti setiap jam, harian, mingguan, dll.) atau sesuai permintaan untuk memberikan kemampuan pemrosesan batch yang kaya untuk pergerakan data dan analitik pada skala perusahaan.

Azure Data Factory juga menyediakan visualisasi yang kaya untuk menampilkan riwayat, versi, dan dependensi antara alur data Anda, serta memantau semua alur data Anda dari satu tampilan terpadu. Hal ini memungkinkan Anda dengan mudah mendeteksi dan menentukan masalah pemrosesan apa pun dan mengatur peringatan pemantauan yang sesuai.

Gambar 6.1 memberikan ilustrasi berbagai operasi pemrosesan data yang dilakukan oleh Azure Data Factory, seperti penyerapan data, persiapan, transformasi, analisis, dan akhirnya publikasi. Data ini dapat dengan mudah digunakan oleh pengguna utama data tersebut.



Gambar 6.1. Azure Data Factory dapat menyerap data dari berbagai sumber data

Saluran Pipa dan Aktivitas

Dalam solusi Azure Data Factory normal, satu atau beberapa alur pemrosesan data biasanya digunakan. Alur Pabrik Data adalah pengelompokan aktivitas yang logis. Ini digunakan untuk mengelompokkan aktivitas menjadi sebuah unit yang bersama-sama melakukan satu tugas.

Aktivitas Azure Data Factory

Aktivitas Azure Data Factory menentukan tindakan yang akan dilakukan pada data Anda. Misalnya, Anda dapat menggunakan aktivitas penyalinan untuk menyalin data dari satu penyimpanan data ke penyimpanan data lainnya. Demikian pula, Anda dapat menggunakan aktivitas kumpulan, yang menjalankan kueri kumpulan pada kluster Azure HDInsight untuk mengubah atau menganalisis data Anda. Data Factory mendukung dua jenis aktivitas:

- ✱✱ Aktivitas Pergerakan Data: Ini mencakup aktivitas penyalinan, yang menyalin data dari penyimpanan data sumber ke penyimpanan data sink. Data Factory mendukung penyimpanan data berikut:
 - ✱✱ Cloud Blue:
 - Penyimpanan blob Azure
 - Penyimpanan Azure Data Lake
 - Basis data Azure SQL
 - Gudang data Azure SQL
 - Penyimpanan tabel Azure
 - Azure DocumentDB
 - Indeks Pencarian Azure
 - ✱✱ Basis Data:
 - SQLServer*
 - Peramal*
 - MySQL*
 - DB2*
 - Teradata*
 - PostgreSQL*
 - Basis data*
 - Cassandra*
 - MongoDB*
 - Pergeseran Merah Amazon
 - ✱✱ Sistem File:
 - Berkas sistem*
 - HDFS*
 - Amazon S3
 - FTP
 - ✱✱ Sistem Lainnya:
 - Tenaga penjualan
 - ODBC Generik*
 - OData Generik
 - Tabel Web (tabel dari HTML)
 - Sejarawan GE*
- ✱✱ Catatan: Penyimpanan data yang dilambangkan dengan * bisa ada di lokal atau di Azure Virtual Machine (IaaS). Opsi ini mengharuskan Anda menginstal Data Management Gateway di komputer lokal atau Azure Virtual Machine.
- ✱✱ Aktivitas Transformasi Data: Azure Data Factory mendukung aktivitas transformasi berikut yang dapat ditambahkan ke alur secara individual atau dirangkai bersama dengan aktivitas lain.

Aktivitas transformasi data lingkungan	Lingkungan komputasi
Hive	HDInsight [Hadoop]
Pig	HDInsight [Hadoop]
MapReduce	HDInsight [Hadoop]
Streaming Hadoop	HDInsight [Hadoop]
Aktivitas Pembelajaran Mesin	Mesin virtual Azure
Prosedur Tersimpan	Azure SQL, Azure SQL Data Warehouse, atau SQL Server di VM
U-SQL Analisis Data	Analisis Azure Data Lake
Dot Net	HD Insight [Hadoop] atau Azure Batch

Jika Anda perlu memindahkan data ke atau dari penyimpanan data yang tidak didukung oleh Aktivitas Penyalinan Azure Data Factory, atau Anda perlu mengubah data menggunakan logika kustom, Anda selalu dapat membuat aktivitas .NET kustom.

Layanan Tertaut

Layanan tertaut menentukan informasi yang diperlukan Azure Data Factory untuk menyambungkan ke sumber daya data eksternal (misalnya: SQL Server lokal, Azure Storage, dan HDInsight yang berjalan di Azure). Layanan tertaut digunakan untuk dua tujuan utama di Azure Data Factory:

- Untuk mewakili penyimpanan data: Seperti SQL Server lokal, database Oracle, berbagi file, atau akun penyimpanan blob Azure.
- Untuk mewakili sumber daya komputasi: Sumber daya yang dapat menampung eksekusi suatu aktivitas. Sebagai contoh, aktivitas kumpulan HDInsight berjalan pada kluster HDInsight Hadoop dan dapat digunakan untuk melakukan transformasi data.

Kumpulan data

Dalam skema yang lebih besar, layanan tertaut menghubungkan penyimpanan data ke pekerjaan Azure Data Factory. Kumpulan data mewakili struktur data dalam penyimpanan data tersebut. Sebagai contoh, "Layanan tertaut Azure SQL" mungkin memberikan informasi koneksi untuk database Azure SQL. Himpunan data Azure SQL kemudian akan menentukan tabel spesifik yang akan berisi data untuk diproses oleh Azure Data Factory.

Selain itu, "Layanan tertaut penyimpanan Azure" akan memberikan informasi koneksi untuk Azure Data Factory agar dapat tersambung ke akun Azure Storage. Dari sana, himpunan data blob Azure akan menentukan kontainer untuk blob dan folder di akun Azure Storage tempat alur harus membaca data yang masuk.

Saluran pipa

Alur Azure Data Factory adalah pengelompokan aktivitas yang terkait secara logis. Pipeline digunakan untuk mengelompokkan aktivitas ke dalam unit logis yang melakukan suatu tugas. Aktivitas menentukan tindakan spesifik yang harus dilakukan pada data. Setiap aktivitas alur dapat mengambil nol atau lebih kumpulan data sebagai masukan dan dapat menghasilkan satu atau lebih kumpulan data sebagai keluaran.

Misalnya, aktivitas penyalinan dapat digunakan untuk menyalin data dari satu penyimpanan data Azure ke penyimpanan data lainnya. Alternatifnya, seseorang dapat

menggunakan aktivitas kumpulan HDInsight untuk menjalankan kueri kumpulan pada kluster Azure HDInsight untuk mengubah aliran data. Azure Data Factory menyediakan berbagai aktivitas penyerapan, pergerakan, dan transformasi data. Pengembang juga memiliki kebebasan memilih untuk membuat aktivitas .NET kustom untuk menjalankan kode kustom mereka sendiri di alur Azure Data Factory.

Penjadwalan dan Eksekusi

Pada titik ini, kami telah memeriksa apa itu alur dan aktivitas Data Factory dan bagaimana hal tersebut disusun untuk membuat aliran kerja pemrosesan data holistik di Azure Data Factory. Kami sekarang akan memeriksa mesin penjadwalan dan eksekusi di Azure Data Factory. Penting untuk dicatat bahwa alur Azure Data Factory hanya aktif antara waktu mulai dan waktu berakhir. Akibatnya, ini tidak dijalankan sebelum waktu mulai atau setelah waktu berakhir. Jika pipeline berada dalam status “dijeda”, maka pipeline tidak akan dieksekusi sama sekali, tidak peduli bagaimana waktu mulai dan berakhirnya ditetapkan.

Perhatikan bahwa bukan pipeline yang benar-benar dieksekusi. Sebaliknya, ini adalah kumpulan aktivitas dalam alur Data Factory yang benar-benar dijalankan. Namun, mereka melakukannya dalam konteks keseluruhan alur Data Factory. Layanan Azure Data Factory memungkinkan Anda membuat alur data yang memindahkan dan mengubah data, lalu menjalankan alur tersebut pada jadwal operasional tertentu (setiap jam, harian, mingguan, dll.). Data Factory juga menyediakan visualisasi yang kaya untuk menampilkan riwayat, versi, dan dependensi antar alur data, dan memungkinkan Anda memantau semua alur data dari satu tampilan terpadu. Ini memberikan alat manajemen yang mudah untuk membantu menentukan masalah dan mengatur peringatan pemantauan.

Skenario End-to-End Aktivitas Penyalinan Pipeline

Di bagian ini, kami memeriksa contoh lengkap pembuatan alur Azure Data Factory untuk menyalin data dari penyimpanan blob Azure ke database Azure SQL. Sepanjang prosesnya, kami menekankan fitur dan kemampuan utama yang dapat Anda manfaatkan untuk memaksimalkan Azure Data Factory sesuai kebutuhan Anda.

Prasyarat Skenario

Sebelum Anda bisa membuat alur atau aktivitas Azure Data Factory, Anda memerlukan hal berikut:

- *Langganan Azure:* Jika Anda belum memiliki langganan, Anda dapat memulai secara gratis di <https://azure.microsoft.com/en-us/free/?b=16.46>.
- *Akun Azure Storage:* Anda menggunakan penyimpanan blob sebagai penyimpanan data "sumber" dalam skenario ini.
- *Azure SQL Database:* Anda menggunakan database Azure SQL sebagai penyimpanan data tujuan dalam tutorial ini.
- *SQL Server Management Studio atau Visual Studio:* Anda menggunakan alat ini untuk membuat database sampel dan tabel tujuan, dan untuk melihat data yang dihasilkan dalam tabel database.

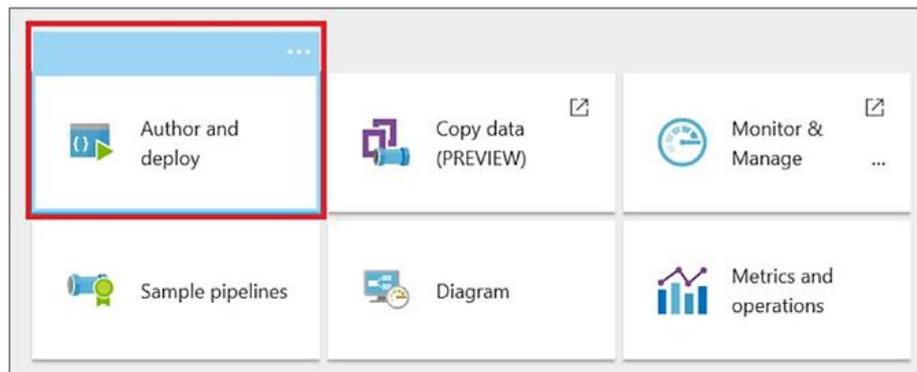
6.2 DEFINISI JSON

Jika Anda telah menelusuri tautan Azure Data Factory ke “Menyalin Data dari Penyimpanan Blob ke Database SQL Menggunakan Data Factory,” Anda mungkin telah memperhatikan bahwa ada berbagai alat yang dapat Anda gunakan untuk menentukan alur atau aktivitas Azure Data Factory:

- Salin Wisaya
- Portal Azure
- Studio visual
- PowerShell
- Templat Azure Resource Manager
- API Istirahat
- .NETAPI

Apa pun alat yang digunakan untuk membuat pekerjaan awal Azure Data Factory, pada akhirnya, Azure Data Factory menggunakan JavaScript Object Notation (JSON) untuk menentukan dan mempertahankan definisi yang Anda buat melalui alat tersebut.

JSON adalah format pertukaran data ringan yang memudahkan manusia untuk membaca dan menulis serta memudahkan mesin untuk mengurai dan menghasilkan. Salah satu keuntungan nyata dari pendekatan ini adalah parameter konfigurasi JSON tertentu dapat disesuaikan dan disesuaikan dengan skenario yang ada guna memberikan kontrol penuh atas konfigurasi dan menjalankan opsi untuk pekerjaan Data Factory. Untuk memulai, navigasikan (melalui Portal Azure) ke pekerjaan Azure Data Factory yang dibuat di tautan dan klik opsi Penulisan dan Penyebaran, seperti yang ditunjukkan pada Gambar 6.2.



Gambar 6.2. Pabrik Data Portal Azure: Opsi Penulisan dan Penyebaran

Setelah Anda mengklik opsi Penulis dan Penyebaran, Anda akan melihat layar yang serupa dengan yang ada di Gambar 6-3, tempat Anda dapat menavigasi setiap langkah pekerjaan alur Pabrik Data yang dibuat dan melihat JSON yang sesuai untuk setiap langkah dari proses tersebut.



Gambar 6.3. Pabrik Data: Opsi Penulisan dan Terapkan JSON

Mari kita lihat JSON yang dihasilkan untuk contoh alur kerja Aktivitas Salin untuk tautan Salin. Titik awalnya adalah menentukan sumber data masuk dan keluar untuk pekerjaan tersebut. Dalam hal ini, kami menggunakan penyimpanan blob Azure untuk masukan dalam bentuk file bernama EmpData.txt, yang merupakan file masukan berformat nilai yang dipisahkan koma (CSV). Perhatikan dua segmen kode JSON dalam Daftar 6-1 yang menjelaskan koneksi penyimpanan blob Azure dan definisi himpunan data yang sesuai untuk sumber input. Daftar 6-1. Deskripsi JSON tentang Sumber Data INPUT untuk Definisi Aktivitas Penyalinan

JSON - BLOB Storage Input Definition

```

{
  "name": "Source-BlobStorage-7an",
  "properties": {
    "hubName": "jbadftutorial_hub",
    "type": "AzureStorage",
    "typeProperties": {
      "connectionString":
"DefaultEndpointsProtocol=https;AccountName=brtblobstoragedev;
AccountKey=*****"
    }
  }
}

```

JSON - Data Definition:

```

{
  "name": "InputDataset-7an",
  "properties": {
    "structure": [
      {
        "name": "Column0",
        "type": "String"
      },
      {
        "name": "Column1",
        "type": "String"
      }
    ],
    "published": false,
    "type": "AzureBlob",
  }
}

```

```

"linkedServiceName": "Source-BlobStorage-7an",
"typeProperties": {
  "fileName": "EmpData.txt",
  "folderPath": "adftutorial",
  "format": {
    "type": "TextFormat",
    "columnDelimiter": ","
  }
},
"availability": {
  "frequency": "Day",
  "interval": 1
},
"external": true,
"policy": {}
}
}

```

Perhatikan dalam dua segmen kode JSON bahwa kedua definisi ini sepenuhnya menggambarkan sumber input data bahkan hingga definisi bidang dalam file teks CSV di penyimpanan blob Azure. Antarmuka di Portal Azure ini juga memungkinkan Anda mengganti parameter standar dengan mudah hanya dengan mengedit JSON secara langsung. Daftar 6.2 memperlihatkan contoh output JSON untuk Operasi Penyalinan Pabrik Data.

Daftar 6-2. Parameter JSON Operasi Penyalinan Pabrik Data

Data Factory - JSON Copy Pipeline Operations

```

{
  "name": "CopyPipelineBlob2SQLTable",
  "properties": {
    "description": "CopyPipelineBlob2SQLTable",
    "activities": [
      {
        "type": "Copy",
        "typeProperties": {
          "source": {
            "type": "BlobSource",
            "recursive": false
          },
          "sink": {
            "type": "SqlSink",
            "writeBatchSize": 0,
            "writeBatchTimeout": "00:00:00"
          },
          "translator": {
            "type": "TabularTranslator",

```

```

        "columnMappings":
"Column0:FirstName,Column1:LastName"
    }
  },
  "inputs": [
    {
      "name": "InputDataset-7an"
    }
  ],
  "outputs": [
    {
      "name": "OutputDataset-7an"
    }
  ],
  "policy": {
    "timeout": "1.00:00:00",
    "concurrency": 1,
    "executionPriorityOrder": "NewestFirst",
    "style": "StartOfInterval",
    "retry": 3,
    "longRetry": 0,
    "longRetryInterval": "00:00:00"
  },
  "scheduler": {
    "frequency": "Day",
    "interval": 1
  },
  "name": "Blobpathadftutorial->dbo_emp"
}
],
"start": "2016-11-22T15:06:22.806Z",
"end": "2099-12-31T05:00:00Z",
"isPaused": false,
"hubName": "jbadftutorial_hub",
"pipelineMode": "Scheduled"
}
}

```

Definisi JSON di Daftar 6-2 memungkinkan Anda memiliki kontrol penuh atas parameter, pemetaan antara file CSV dan tabel SQL, dan menjalankan perilaku pekerjaan Salin saluran ini. Selain itu, perhatikan bahwa dalam bagian penjadwal dari contoh kode JSON aktivitas, Anda dapat menentukan jadwal berulang untuk aktivitas alur. Misalnya, Anda dapat menjadwalkan aktivitas penyalinan alur Data Factory agar berjalan setiap jam dengan memodifikasi JSON sebagai berikut:

JSON Code Fragment - Scheduler

```
"scheduler": {
```

```
"frequency": "Hour",
"interval": 1
},
```

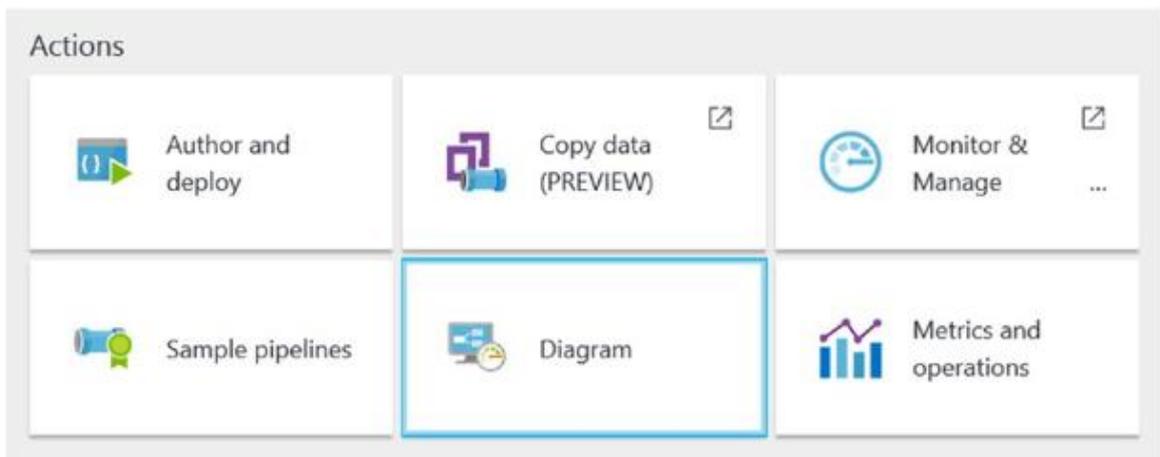
Seperti yang dapat dilihat dari arsitektur composable yang disediakan oleh definisi JSON, Azure Data Factory adalah alat yang sangat kuat dan fleksibel untuk membantu mengelola semua aspek penting dalam mengelola Big Data di cloud. Aspek seperti penyerapan data (baik di lokasi atau di Azure), persiapan, transformasi, pergerakan, dan penjadwalan merupakan fitur yang diperlukan untuk menjalankan platform manajemen data tingkat perusahaan.

Memantau Dan Mengelola Saluran Pabrik Data

Layanan Azure Data Factory menyediakan kemampuan dasbor pemantauan yang kaya yang membantu melakukan tugas-tugas berikut:

- Menilai data kesehatan saluran pipa dari ujung ke ujung
- Identifikasi dan perbaiki masalah pemrosesan jalur pipa
- Lacak riwayat dan asal usul data Anda
- Melihat hubungan antar sumber data
- Melihat riwayat akuntansi lengkap mengenai pelaksanaan pekerjaan, kesehatan sistem, dan ketergantungan pekerjaan

Anda dapat dengan mudah memantau status pekerjaan alur Azure Data Factory dengan menavigasi ke pekerjaan Data Factory Anda di Portal Azure, lalu mengklik opsi Diagram, seperti yang ditunjukkan pada Gambar 6.4.



Gambar 6-4. Portal Azure: pekerjaan Pabrik Data, tampilan diagram

Selanjutnya, Anda akan melihat diagram visual tugas alur Data Factory Anda, seperti yang ditunjukkan pada Gambar 6-5. Dengan mengklik salah satu definisi Input atau Output, Anda dapat melihat riwayat dan status setiap “potongan” data yang dibuat, beserta jadwal yang akan terjadi selanjutnya.

The screenshot displays the Azure Data Factory monitoring console. On the left, a pipeline activity named 'CopyPipelineBlob2SQLTable' is shown, connected to an input dataset 'InputDataset-7an' (Azure Blob Storage) and an output dataset 'OutputDataset-7an' (Azure SQL Database). The right-hand side features a 'Properties' pane for the selected activity, showing details like 'Table: OutputDataset-7an', 'Created: 11/23/2016 03:10:30 PM UTC', and 'Provisioning state: Succeeded'. Below the properties is a 'Monitoring' section with a table of 'Recently updated slices'.

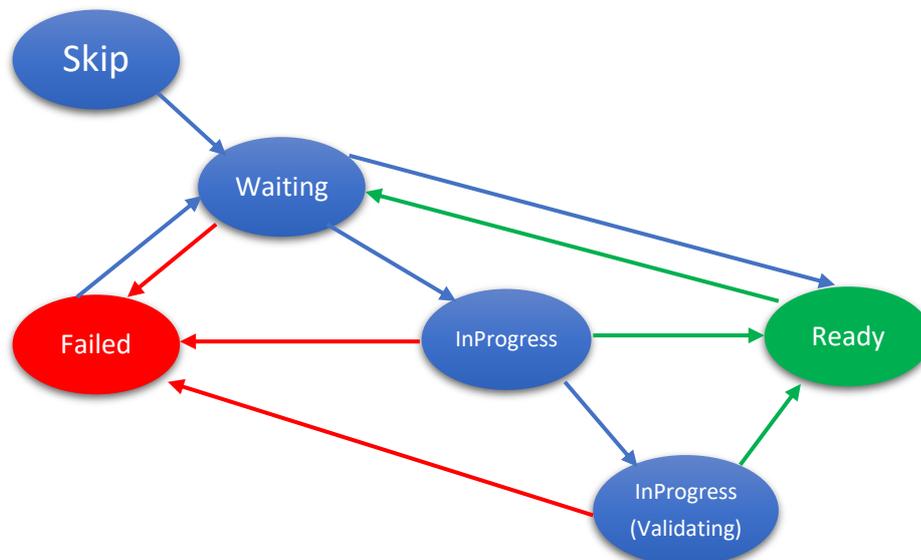
LAST UPDATE TIME	SLICE START TIME	SLICE END TIME	STATUS
11/25/2016 12:02:3...	11/24/2016 12:00 A...	11/25/2016 12:00 A...	Ready
11/24/2016 12:04:3...	11/23/2016 12:00 A...	11/24/2016 12:00 A...	Ready
11/23/2016 03:12:5...	11/22/2016 12:00 A...	11/23/2016 12:00 A...	Ready
11/23/2016 03:10:4...	12/31/2099 12:00 A...	01/01/2100 12:00 A...	Pending execution
11/23/2016 03:10:4...	12/30/2099 12:00 A...	12/31/2099 12:00 A...	Pending execution
11/23/2016 03:10:4...	12/29/2099 12:00 A...	12/30/2099 12:00 A...	Pending execution
11/23/2016 03:10:4...	12/28/2099 12:00 A...	12/29/2099 12:00 A...	Pending execution
11/23/2016 03:10:4...	12/27/2099 12:00 A...	12/28/2099 12:00 A...	Pending execution

Gambar 6-5. Memantau Azure Data Factory dengan melihat riwayat segmen output

Perhatikan bahwa status untuk setiap aktivitas alur di Azure Data Factory dapat berputar di antara banyak status eksekusi potensial, sebagai berikut:

- Skip
- Waiting
- In-Progress
- In-Progress (Validating)
- Ready
- Failed

Gambar 6.6 mewakili berbagai status eksekusi yang dapat terjadi ketika pekerjaan alur Azure Data Factory aktif.



Gambar 6.6. Alur transisi status pekerjaan alur Data Factory

"Irisan" Azure Data Factory adalah interval di mana pekerjaan alur dieksekusi dalam periode yang ditentukan di properti awal dan akhir alur. Misalnya, jika Anda menyetel waktu mulai dan waktu berakhir agar terjadi dalam satu hari, dan Anda menyetel frekuensinya menjadi satu jam, maka aktivitas akan dijalankan sebanyak 24 kali. Dalam hal ini, Anda akan memiliki 24 irisan, semuanya menggunakan sumber data yang sama.

Biasanya, di Azure Data Factory, potongan data dimulai dalam status Menunggu agar prasyarat dipenuhi sebelum dieksekusi. Kemudian, aktivitas mulai dijalankan dan potongan masuk ke status Sedang Berlangsung. Eksekusi aktivitas mungkin berhasil atau gagal. Potongan tersebut ditandai sebagai Siap atau Gagal berdasarkan hasil eksekusi. Anda dapat mengatur ulang irisan untuk kembali dari status Siap atau Gagal ke status Menunggu. Anda juga dapat menandai status irisan ke Lewati, yang mencegah aktivitas dijalankan dan tidak akan memproses potongan tersebut.

Aktivitas Pabrik Data dan Penyetelan Performa

Faktor penting lainnya yang perlu dipertimbangkan ketika memilih sistem pemrosesan analisis data cloud adalah kinerja dan skalabilitas. Azure Data Factory menyediakan platform penyerapan dan transformasi data yang aman, andal, dan berkinerja tinggi yang dapat berjalan dalam skala besar. Azure Data Factory dapat mengaktifkan skenario perusahaan di mana beberapa terabyte data dipindahkan dan diubah di berbagai macam penyimpanan data, baik di lokasi maupun di Azure. Aktivitas penyalinan Azure Data Factory menawarkan pengalaman pemuatan data yang sangat optimal yang mudah dipasang dan dikonfigurasi. Hanya dalam satu aktivitas penyalinan alur, Anda dapat mencapai kecepatan pemuatan yang serupa dengan berikut ini:

- Memuat data ke gudang data Azure SQL dengan kecepatan 1,2 GB per detik.
- Memuat data ke penyimpanan blob Azure dengan kecepatan 1,0 GB per detik.
- Memuat data ke Azure Data Lake Store dengan kecepatan 1,0 GB per detik.

Salinan Paralel

Azure Data Factory juga memiliki kemampuan untuk menjalankan aktivitas penyalinan dari sumber atau menulis data ke tujuan dalam operasi paralel yang dijalankan dalam aktivitas Penyalinan. Fitur ini dapat memberikan dampak dramatis pada throughput operasi penyalinan dan juga dapat mengurangi waktu yang diperlukan untuk melakukan transformasi data dan fungsi perpindahan.

Anda dapat menggunakan properti JSON "salinan paralel" untuk menunjukkan paralelisme yang ingin Anda gunakan untuk aktivitas penyalinan. Anda dapat menganggap properti ini sebagai jumlah maksimum thread dalam aktivitas penyalinan yang dapat membaca dari sumber Anda atau menulis ke penyimpanan data sink Anda secara paralel.

Daftar 6.3. Cuplikan JSON Aktivitas Penyalinan Alur Menampilkan Properti `parallelCopies`

JSON Pipeline Copy Activity - "parallelCopies" Property

```
"activities": [
  {
    "name": "Sample copy activity",
```

```

    "description": "",
    "type": "Copy",
    "inputs": [{ "name": "InputDataset" }],
    "outputs": [{ "name": "OutputDataset" }],
    "typeProperties": {
      "source": {
        "type": "BlobSource",
      },
    },
    "sink": {
      "type": "AzureDataLakeStoreSink"
    },
    "parallelCopies": 8
  }
}
]

```

Untuk setiap aktivitas penyalinan yang dijalankan, Azure Data Factory menentukan jumlah salinan paralel yang akan digunakan untuk menyalin data dari penyimpanan data sumber ke penyimpanan data tujuan. Jumlah default salinan paralel yang digunakan bergantung pada jenis sumber data dan sink data yang digunakan.

Unit Pergerakan Data Cloud (DMU)

Unit Pergerakan data Cloud (DMU) adalah pengukuran Pabrik Data yang mewakili daya relatif (kombinasi CPU, memori, dan alokasi sumber daya jaringan) dari satu unit di Azure Data Factory. DMU mungkin digunakan dalam operasi penyalinan cloud-ke-cloud, namun tidak dalam salinan hibrid dari penyimpanan data lokal. Secara default, Azure Data Factory menggunakan DMU cloud untuk melakukan eksekusi aktivitas penyalinan alur tunggal. Untuk mengganti nilai default, tentukan nilai untuk properti `cloudDataMovementUnits`, seperti yang ditunjukkan dalam segmen kode pada Daftar 6-4.

Daftar 6-4. Contoh cuplikan JSON menampilkan Properti `cloudDataMovementUnits`

Data Factory - JSON Property for "cloudDataMovementUnits"

```

"activities":[
{
  "name": "Sample copy activity",
  "description": "",
  "type": "Copy",
  "inputs": [{ "name": "InputDataset" }],
  "outputs": [{ "name": "OutputDataset" }],
  "typeProperties": {
    "source": {
      "type": "BlobSource",
    },
    "sink": {
      "type": "AzureDataLakeStoreSink"
    },
  },
}
]

```

```

        "cloudDataMovementUnits": 4
    }
}
]

```

Perhatikan bahwa Anda dapat mencapai throughput yang lebih tinggi dengan memanfaatkan lebih banyak unit pergerakan data (DMU) daripada DMU maksimum default, yaitu delapan untuk menjalankan aktivitas penyalinan cloud-ke-cloud. Sebagai contoh, Anda dapat menyalin data dari blob Azure ke Azure Data Lake Store dengan kecepatan 1 gigabyte per detik jika Anda diatur untuk menggunakan (100) DMU. Untuk meminta lebih banyak DMU daripada default delapan untuk langganan Anda, Anda perlu mengirimkan permintaan dukungan melalui Portal Microsoft Azure.

6.3 PENYIMPANAN AZURE DATA LAKE

Azure Data Lake Store adalah repositori dan lingkungan pemrosesan berskala besar untuk beban kerja analitis Big Data modern saat ini. Azure Data Lake memungkinkan Anda menyimpan data dengan ukuran, tipe data, dan kecepatan penyerapan apa pun, di satu lokasi, untuk digunakan dalam penelitian operasional dan analisis data.

Akses Hadoop

Azure Data Lake Store dapat diakses dari Hadoop dan Azure HDInsight menggunakan ReST API yang kompatibel dengan WebHDFS. Modul `hadoop-azure-datalake` menyediakan dukungan untuk integrasi dengan Azure Data Lake Store. File JAR diberi nama `azure-datalake-store.jar`.

Perhatikan bahwa ada perbedaan yang harus dibuat seputar arti istilah Azure Data Lake. Kemungkinan ada dua arti berbeda di Microsoft Azure. Ini biasanya digunakan untuk merujuk pada subsistem penyimpanan di Azure yang lebih sering disebut sebagai “Azure Data Lake Store” atau “ADLS”. Variasi lain dari istilah ini adalah “Azure Data Lake Analytics” atau “ADLA,” yang merupakan layanan analitik berbasis Azure tempat Anda dapat dengan mudah mengembangkan dan menjalankan program transformasi dan pemrosesan data paralel secara besar-besaran dalam berbagai bahasa seperti U-SQL, R, Python, dan .NET. Azure Data Lake Analytics dibahas secara mendetail di Bab 7. Untuk saat ini, kita akan membahas dasar-dasar Azure Data Lake Store.

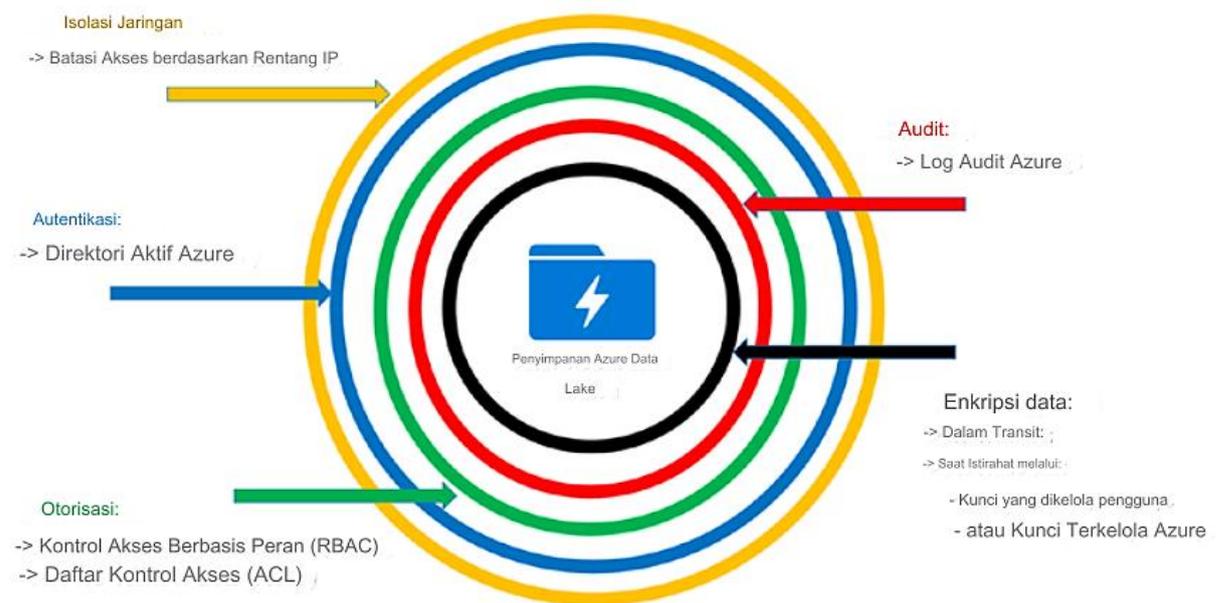
ADLS dirancang khusus untuk mengaktifkan analitik pada data yang disimpan di Azure Data Lake. Subsistem penyimpanan Data Lake disesuaikan secara khusus untuk kinerja tinggi dalam skenario analisis data. Sebagai penawaran layanan terkelola sepenuhnya dari Microsoft, Azure Data Lake Store mencakup semua kemampuan tingkat perusahaan yang diharapkan dari repositori berbasis cloud dengan skalabilitas besar. “Kemampuan” utama yang disediakan oleh Azure Data Lake Store meliputi: keamanan, pengelolaan, skalabilitas, keandalan, dan ketersediaan. Semua karakteristik tersebut penting untuk kasus penggunaan perusahaan di dunia nyata.

Dengan Azure Data Lake Store, kini Anda dapat menjelajahi dan memperoleh nilai dari semua data perusahaan Anda yang tidak terstruktur, semi-terstruktur, dan terstruktur dengan

menjalankan analitik paralel secara besar-besaran pada jumlah data berapa pun. Azure Data Lake Store tidak memiliki batasan buatan mengenai jumlah data, jumlah file, atau ukuran file individual yang dapat disimpan. Pada saat penulisan ini, ADLS dapat menyimpan file individual hingga berukuran petabyte, yang setidaknya 200x lebih besar daripada layanan penyimpanan cloud lain yang tersedia saat ini.

Lapisan Keamanan

Azure Data Lake Store memiliki fitur keamanan yang “bawaan” sejak awal. Seperti yang bisa dilihat di Gambar 6-7, Azure Data Lake Store memiliki sejumlah fitur dan kemampuan keamanan Azure yang berlapis untuk membantu memberikan keyakinan tertinggi pada keamanan data, baik data dalam keadaan diam atau dalam transit.



Gambar 6.7. Keamanan berlapis di Azure Data Lake Store

Gambar 6-7 mengilustrasikan berbagai lapisan keamanan yang terlibat dalam melindungi data Anda di Azure Data Lake Store. Berikut adalah ringkasan singkat dari fitur keamanan “bawaan” ini:

- **Isolasi Jaringan:** Azure Data Lake Store memungkinkan Anda membuat firewall dan menentukan rentang alamat IP untuk klien tepercaya Anda. Dengan rentang alamat IP, hanya klien yang memiliki IP dalam rentang yang ditentukan yang dapat tersambung ke Azure Data Lake Store.
- **Autentikasi:** Azure Data Lake Store memiliki Azure Active Directory (AAD) yang terintegrasi secara asli untuk membantu mengelola akses dan izin pengguna dan grup. AAD juga menyediakan manajemen siklus hidup penuh untuk jutaan identitas, integrasi dengan Direktori Aktif lokal, dukungan sistem masuk tunggal, autentikasi multifaktor, dan dukungan untuk protokol autentikasi terbuka standar industri seperti OAuth.
- **Otorisasi:** Azure Data Lake Store (ADLS) menyediakan kemampuan Kontrol Akses Berbasis Peran (RBAC) melalui Daftar Kontrol Akses (ACL) untuk mengelola akses ke file data di

penyimpanan Data Lake. Kemampuan ini memberikan kontrol menyeluruh atas akses dan izin file (dalam skala besar) ke semua data yang disimpan di Azure Data Lake.

- **Audit:** Azure Data Lake Store menyediakan kemampuan audit yang kaya untuk membantu memenuhi persyaratan keamanan modern dan kepatuhan terhadap peraturan. Audit diaktifkan secara default untuk semua aktivitas pengelolaan akun dan akses data. Log audit dari Azure Data Lake Store dapat dengan mudah diuraikan karena disimpan dalam format JSON. Selain itu, karena log audit dalam format yang mudah digunakan seperti JSON, Anda dapat menggunakan berbagai alat Business Intelligence (BI) untuk membantu menganalisis dan melaporkan aktivitas ADLS.
- **Enkripsi:** Azure Data Lake Store menyediakan enkripsi bawaan untuk skenario “saat istirahat” dan “dalam perjalanan”. Untuk skenario data tidak aktif, administrator Azure dapat menentukan apakah akan mengizinkan Azure mengelola Kunci Enkripsi Utama (MEK) atau Anda dapat menggunakan MEK yang dibawa sendiri. Dalam kedua kasus tersebut, MEK akan disimpan dan dikelola dengan aman di Azure Key Vault, yang dapat memanfaatkan HSM (Modul Keamanan Perangkat Keras) yang divalidasi FIPS 140-2 Level 2. Untuk skenario transit data, data Azure Data Lake Store selalu dienkripsi, dengan menggunakan protokol HTTPS (HTTP over Secure Sockets Layer).

Perhatikan bahwa di Azure Data Lake Store, Anda dapat memilih apakah data Anda dienkripsi atau tidak dienkripsi sama sekali. Jika Anda memilih enkripsi, semua data yang disimpan di Azure Data Lake Store dienkripsi sebelum data disimpan di penyimpanan. Sebagai alternatif, ADLS akan mendekripsi data sebelum diambil oleh klien. Dari sudut pandang klien, enkripsinya transparan dan mulus. Akibatnya, tidak ada perubahan kode yang diperlukan di sisi klien untuk melihat atau mengenkripsi/mendekripsi data.

Manajemen Kunci Enkripsi ADLS

Untuk manajemen kunci enkripsi, Azure Data Lake Store menyediakan dua mode untuk mengelola Kunci Enkripsi Utama (MEK) Anda. Kunci ini diperlukan untuk mengenkripsi dan mendekripsi data apa pun yang disimpan di Azure Data Lake Store.

Anda dapat membiarkan Data Lake Store mengelola kunci enkripsi utama untuk Anda atau memilih untuk mempertahankan kepemilikan MEK menggunakan akun Azure Key Vault Anda. Anda dapat menentukan mode manajemen kunci saat membuat akun Azure Data Lake Store baru.

Mengimplementasikan Data Factory dan Data Lake Store dalam Implementasi Referensi

Sekarang setelah Anda memiliki latar belakang yang kuat tentang fitur dan kemampuan di Azure Data Factory dan Azure Data Lake Store, Anda akan menggunakan pengetahuan Anda dengan menerapkan beberapa bagian penting lagi dari implementasi referensi di sisa bab ini. Sebagai penyegaran cepat, Anda akan mengimplementasikan tiga fungsi berikut yang diperlukan untuk implementasi referensi:

- ❖ Perbarui data Referensi yang Anda gunakan untuk pekerjaan Azure Stream Analytics. Anda akan menggunakan pekerjaan penyalinan Azure Data Factory untuk menyalin data profil anggota tim dari Azure DocumentDB ke file CSV berbasis teks di penyimpanan Azure Blob. Seperti yang Anda ingat, Anda menggunakan data referensi

ini dalam kueri ASA SQL JOIN untuk mengumpulkan data kesehatan anggota tim tambahan. Kami ingin memastikan bahwa data referensi ini diperbarui secara berkala melalui pekerjaan penyalinan terjadwal.

- ❖ Melatih kembali model Azure Machine Learning untuk memprediksi tingkat kesehatan dan kelelahan anggota tim. Kami menerapkan fungsi di Bab 5 (StreamAnalytics) untuk memanggil Layanan Web Azure Machine Learning. Kami ingin memperbarui prediksi model yang berjalan di belakang layanan ini menggunakan data stres medis yang diperbarui dari tes yang diberikan kepada anggota tim secara berkala.
- ❖ Memindahkan data dari penyimpanan blob Azure ke Azure Data Lake. Pekerjaan ini akan menyalin data yang awalnya berasal dari IoT Hub dan disimpan ke penyimpanan blob Azure oleh pekerjaan Azure Stream Analytics di Bab 5. Kami ingin memindahkan data ini dari Azure blob Storage ke Azure Data Lake Store.

6.4 PERBARUI FILE INPUT DATA REFERENSI UNTUK AZURE STREAM ANALYTICS

Di bagian ini, kita akan membahas langkah-langkah yang diperlukan untuk membuat pekerjaan alur penyalinan Azure Data Factory yang akan menyalin data dari database Azure DocumentDB “NoSQL” ke file CSV berbasis teks yang disimpan di penyimpanan blob Azure.

Perhatikan bahwa file CSV akan menerapkan konvensi penamaan file tertentu sehingga pekerjaan Azure Stream Analytics mengetahui cara menggunakan versi file terbaru untuk digunakan dalam pekerjaan analisis aliran yang memerlukan data referensi ini. Gambar 6-8 mengilustrasikan parameter INPUT REFERENCE DATA untuk pekerjaan Azure Stream Analytics yang menjelaskan konvensi penamaan file.

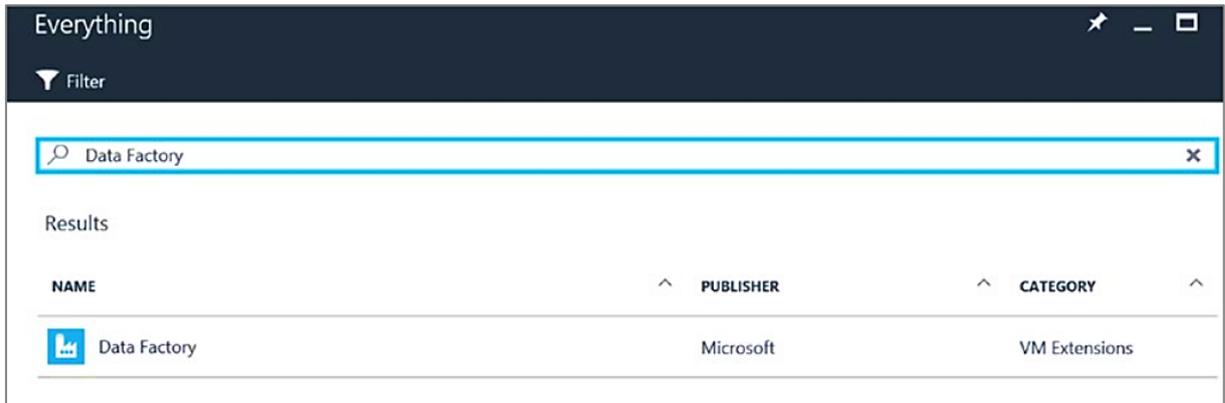


Gambar 6-8. Konvensi penamaan file data referensi Stream Analytics

Dengan memanfaatkan konvensi penamaan file ini, data referensi yang digunakan sebagai input dalam pekerjaan Azure Stream Analytics akan selalu mencerminkan versi data terbaru. Keuntungan lain menggunakan pendekatan ini adalah pembaruan data referensi di masa mendatang dapat dilakukan dengan mudah tanpa berdampak buruk pada pekerjaan analisis aliran apa pun yang sedang diproses. Pekerjaan berikutnya yang berjalan hanya menemukan dan menyerap file data referensi terbaru yang ada di penyimpanan blob Azure, berdasarkan konvensi penamaan file. Dengan cara yang sederhana namun sangat efektif ini, pekerjaan analisis aliran akan selalu menggunakan data referensi versi terbaru pada waktu proses.

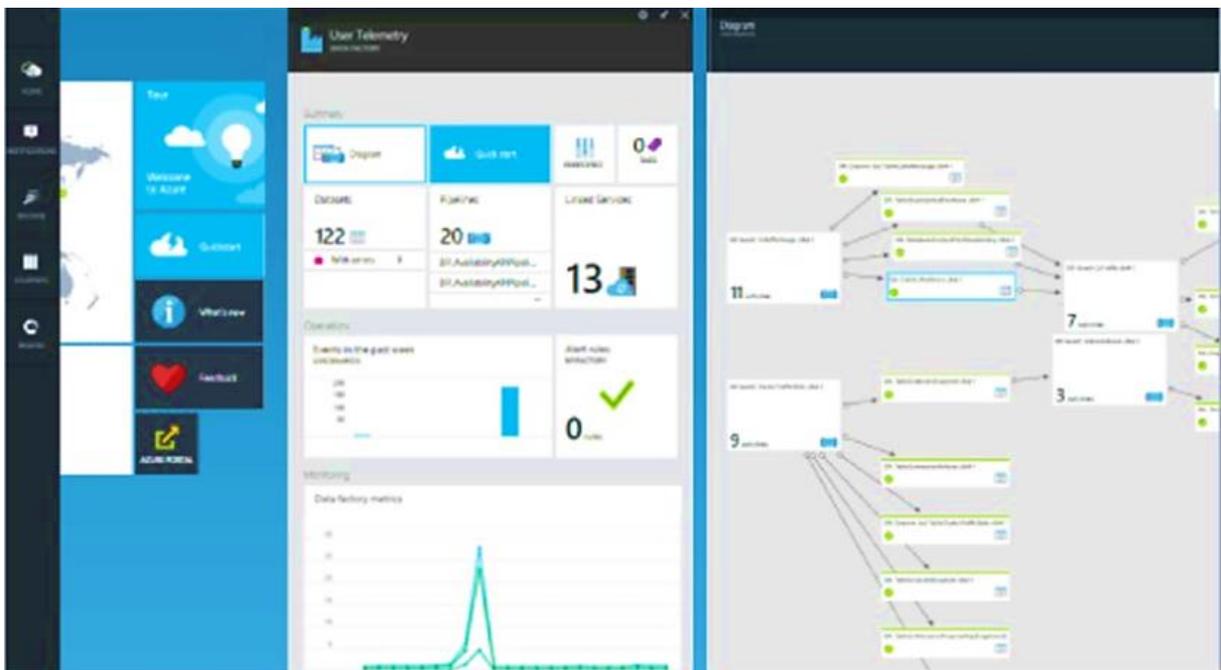
Buat Pekerjaan Azure Data Factory

Untuk memulai, navigasikan ke grup sumber daya untuk penerapan Anda melalui Portal Azure. Klik tombol + Tambah dan cari Data Factory, seperti yang ditunjukkan pada Gambar 6-9.



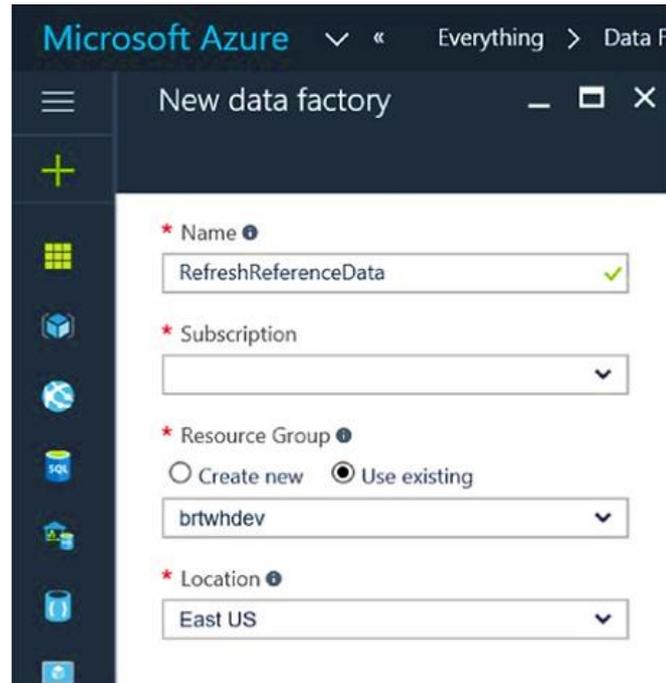
Gambar 6.9 Mencari dan menambahkan pekerjaan Data Factory ke grup sumber daya

Setelah memilih Azure Data Factory, klik pada layar berikutnya untuk membuat pekerjaan baru, seperti yang ditunjukkan pada Gambar 6-10.



Gambar 6.10. Buat pekerjaan Pabrik Data

Layar berikutnya memungkinkan Anda memasukkan parameter spesifik untuk membuat pekerjaan Azure Data Factory baru, seperti yang ditunjukkan pada Gambar 6-11.

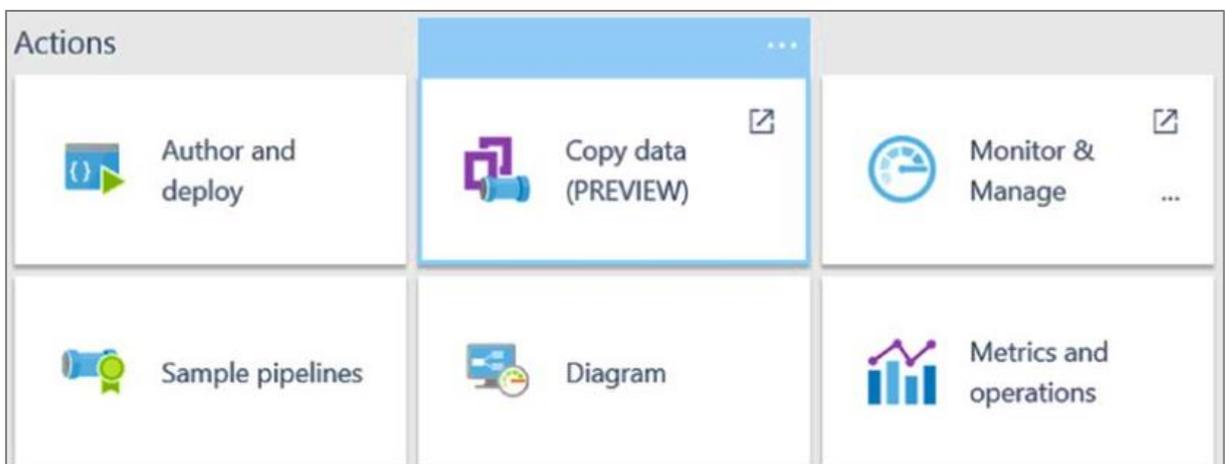


Gambar 6.11. Pabrik Data membuat parameter pekerjaan

Isi pilihan Anda untuk nilai parameter yang sesuai:

- Nama: Masukkan nama unik untuk Pabrik Data baru Anda. Perhatikan bahwa nama Azure Data Factory harus unik secara global.
- Berlangganan: Langganan Azure yang digunakan untuk pekerjaan ini.
- Grup Sumber Daya: Grup Sumber Daya Azure tempat layanan ini dibuat.
- Lokasi: Lokasi Pusat Data Azure.

Setelah selesai, klik tombol Buat di bagian bawah layar. Masukan Anda kemudian akan divalidasi dan pekerjaan Azure Data Factory baru akan dibuat setelah jangka waktu singkat. Diperlukan waktu kurang dari satu menit melalui Portal Azure. Setelah pekerjaan Anda disediakan, navigasikan ke Pabrik Data baru melalui Portal Azure dan pilih opsi Salin Data (PREVIEW), seperti yang ditunjukkan pada Gambar 6-12.



Gambar 6.12. Pabrik Data: Salin Panduan Data

Ini akan memanggil Azure Data Factory Copy Data Wizard untuk diluncurkan dan akan memandu Anda melalui langkah-langkah yang diperlukan untuk membuat alur salinan dasar. Di balik layar, Azure Data Factory menghasilkan file JSON untuk mencerminkan pilihan Anda di Copy Wizard. Gambar 6-13 menggambarkan layar pertama dari Copy Data Wizard dan memungkinkan Anda menentukan properti untuk pekerjaan penyalinan.

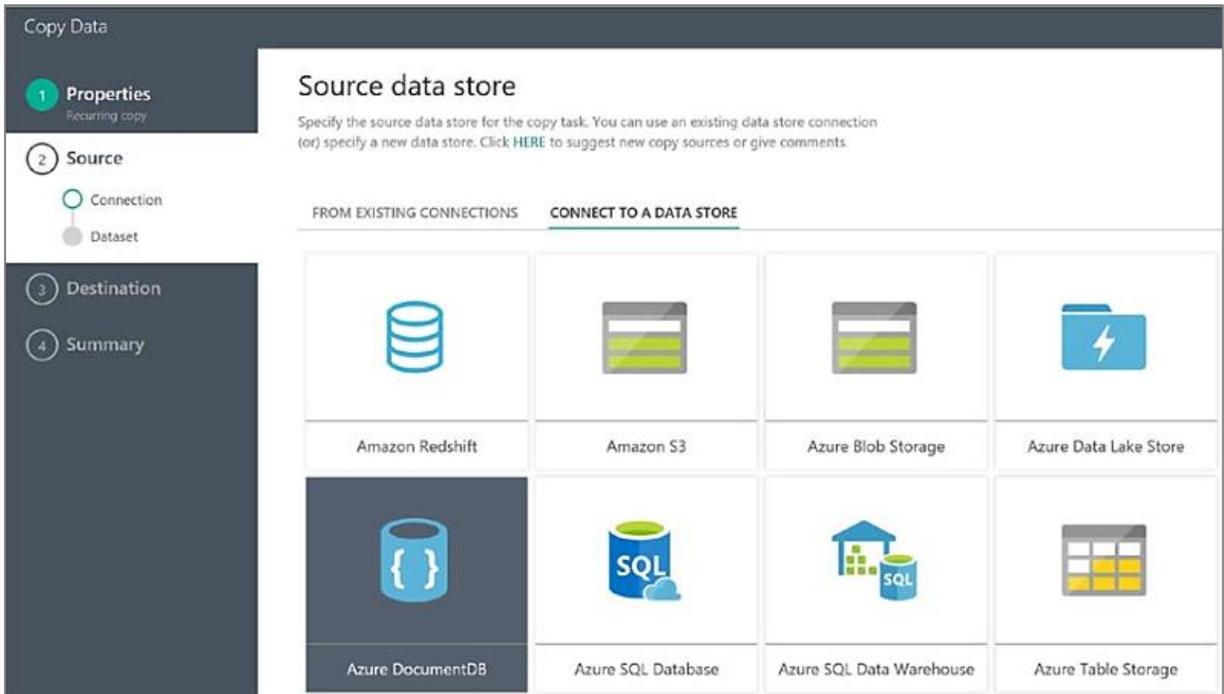
The screenshot shows the 'Copy Data' wizard interface. On the left, a sidebar lists four steps: 1 Properties (selected), 2 Source, 3 Destination, and 4 Summary. The main area is titled 'Properties' and contains the following fields and options:

- Task name (required):** A text input field containing 'CopyPipeline-cdw'.
- Task description:** A text input field with the placeholder text 'Enter description here'.
- Task cadence (or) Task schedule:** Two radio buttons: 'Run once now' (unselected) and 'Run regularly on schedule' (selected).
- Recurring pattern:** A dropdown menu set to 'Daily', followed by 'every' and a numeric input field set to '1', and 'day'.
- Start date time (UTC):** A text input field containing '11/28/2016 03:20 am'.
- End date time (UTC):** A text input field containing '12/31/2099 05:00 am'.

At the bottom of the form, there are two buttons: 'Previous' (disabled) and 'Next' (active).

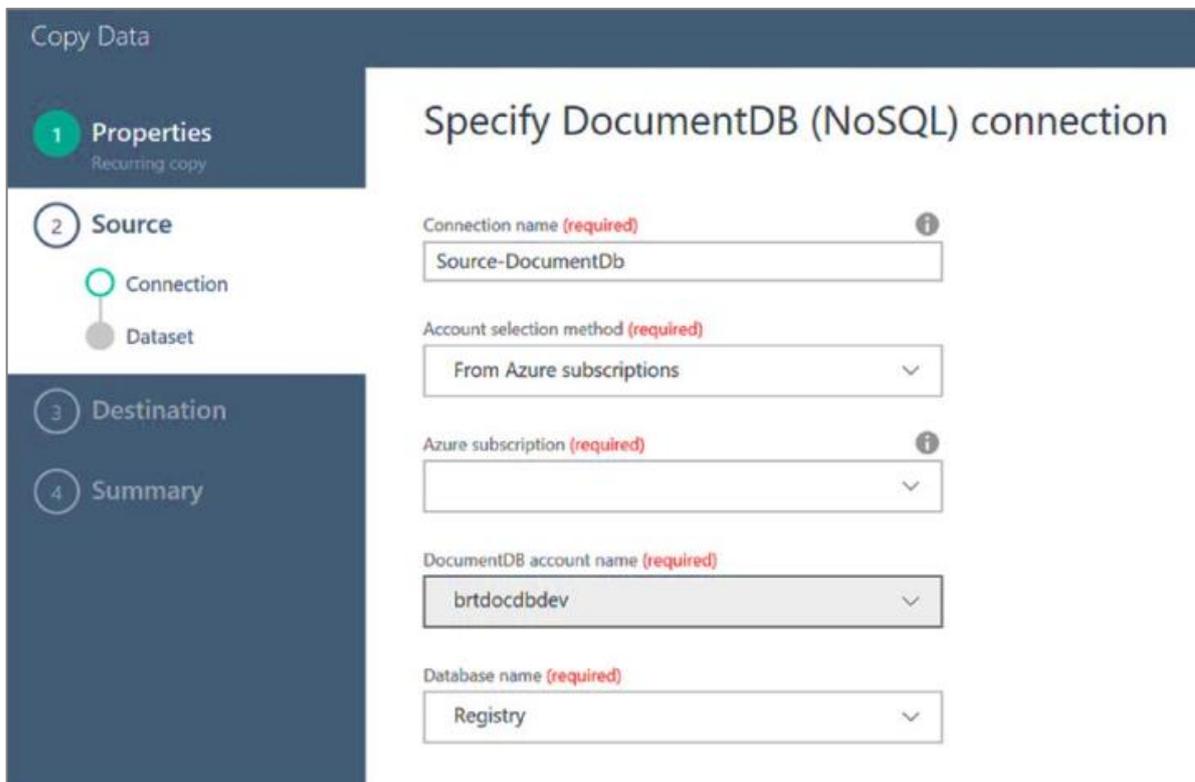
Gambar 6.13. Wizard Data Salin Pabrik Data: tentukan properti

Untuk contoh ini, masukkan nama tugas CopyReferenceData dan pertahankan sisa default untuk jadwal, status, dan tanggal akhir. Klik Berikutnya untuk melanjutkan ke langkah berikutnya. Gambar 6.14 menggambarkan layar Sumber Data, tempat Anda memilih Azure DocumentDB untuk skenario implementasi referensi.



Gambar 6.14. Wizard Data Salin Pabrik Data: tentukan penyimpanan data sumber Azure DocumentDB

Setelah memilih Azure DocumentDB, Anda kemudian akan melihat layar mendetail yang mirip dengan Gambar 6.15, tempat Anda dapat menentukan parameter instans DocumentDB untuk mengambil data referensi.



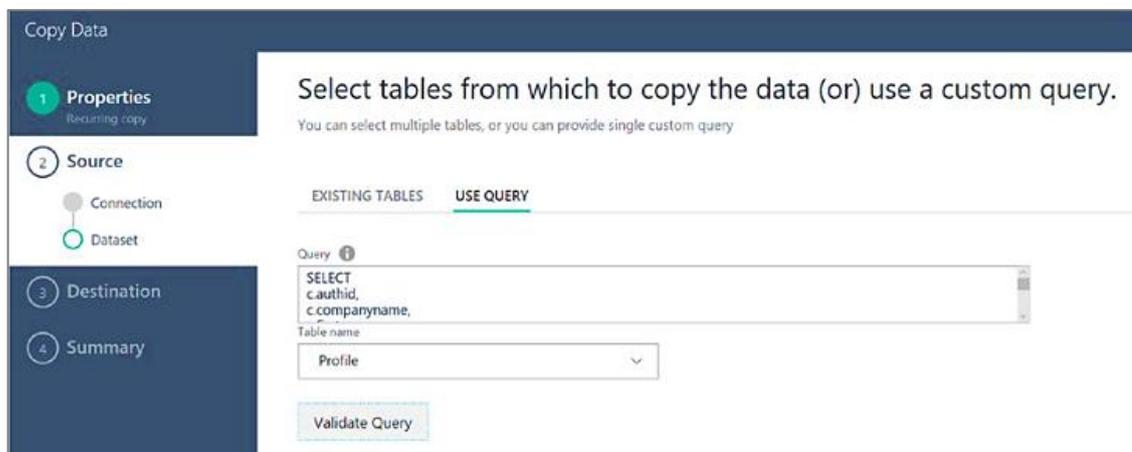
Gambar 6.15. Wizard Data Salin Pabrik Data: tentukan parameter Azure DocumentDB

Klik tombol Berikutnya setelah memasukkan parameter DocumentDB. Anda akan melihat layar seperti Gambar 6-16.



Gambar 6-16. Wizard Data Salin Pabrik Data: tentukan salinan dari tabel atau kueri

Klik pada opsi untuk Menggunakan Kueri alih-alih tabel yang ada. Anda ingin memilih secara dinamis bidang yang Anda perlukan dari tabel DocumentDB. Anda akan melihat layar seperti Gambar 6-17.



Gambar 6.17. Wizard Data Salin Pabrik Data: tentukan parameter kueri

Di jendela Kueri, ketikkan pernyataan SQL berikut:

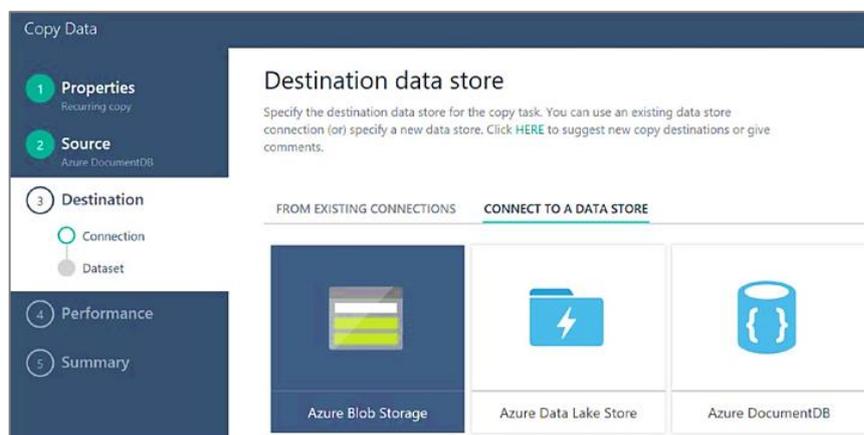
```
SELECT
  c.authid,
  c.companyname,
  c.firstname,
  c.lastname,
  c.username,
  c.imageUrl,
  c.type,
  c.address.address1,
  c.address.address2,
  c.address.address3,
```

```

c.address.city,
c.address.state,
c.address.zip,
c.address.country,
c.social.phone,
c.social.email,
c.social.linkedin,
c.social.facebook,
c.social.twitter,
c.social.blog,
c.healthInformation.age,
c.healthInformation.height,
c.healthInformation.weight,
c.healthInformation.gender,
c.healthInformation.race,
250
c.location.longitude,
c.location.latitude,
c.id,
c.cachettl,
c._rid,
c._self,
c._etag,
c._attachments,
c._ts
FROM
c
WHERE
c.type <> 1

```

Untuk nama tabel, pilih Profil. Kemudian klik Validasi Kueri untuk menguji sintaks SQL Anda. Jika tidak ada kesalahan, tombol akan beralih dari Validasi... kembali ke Validasi Kueri. Klik tombol Berikutnya untuk melanjutkan ke layar berikutnya di mana Anda akan menentukan tujuan penyimpanan data, seperti yang ditunjukkan pada Gambar 6-18.



Gambar 6.18. Wizard Data Salin Pabrik Data: tentukan penyimpanan data tujuan

Pilih Azure Blob Storage lalu klik tombol Berikutnya untuk melanjutkan ke layar berikutnya. Berikutnya akan menanyakan spesifik akun penyimpanan blob Azure Anda, seperti yang ditunjukkan pada Gambar 6.19.

The screenshot shows the 'Specify the Azure Blob storage account' step in the 'Copy Data' wizard. The left sidebar indicates the current step is '3 Destination', with sub-steps 'Connection' and 'Dataset'. The main area contains the following fields:

- Connection name (required):** Destination-BlobStorage
- Account selection method (required):** From Azure subscriptions
- Azure subscription (required):** azurepass-bobf (ba3e3785-1e64-4e12-a03e-16)
- Storage account name (required):** brtblobstoragedev

Gambar 6.19. Panduan Penyalinan Data Pabrik Data: tentukan properti akun penyimpanan blob Azure

Masukkan spesifikasi akun penyimpanan blob Azure Anda, lalu klik tombol Berikutnya untuk melanjutkan ke layar berikutnya, seperti yang ditunjukkan pada Gambar 6.20. Di sinilah Anda menentukan nama folder dan file untuk tujuan di penyimpanan blob Azure.

The screenshot shows the 'Choose the output file or folder' step in the 'Copy Data' wizard. The left sidebar indicates the current step is '3 Destination', with sub-steps 'Connection' and 'Dataset'. The main area contains the following fields:

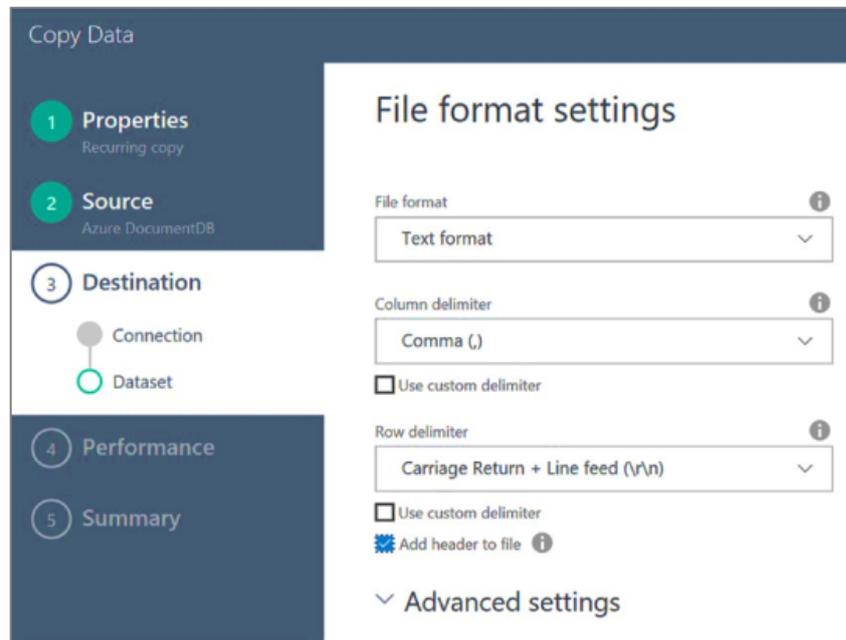
- Folder path:** refdata-team
- Filename:** TeamReferenceData.csv

Below the fields, there is a note: "You can use variables in the folder path to copy data from a folder that is determined at runtime. Make sure that you select a folder with that structure using the Browse button first. The supported variables are: {year}, {month}, {day}, {hour}, {minute} and {custom}. See [Data Movement Activities](#) article for details about these variables. Example: inputfolder/(year)/(month)/(day)."

Gambar 6.20. Wizard Data Salin Pabrik Data: tentukan properti file atau folder keluaran

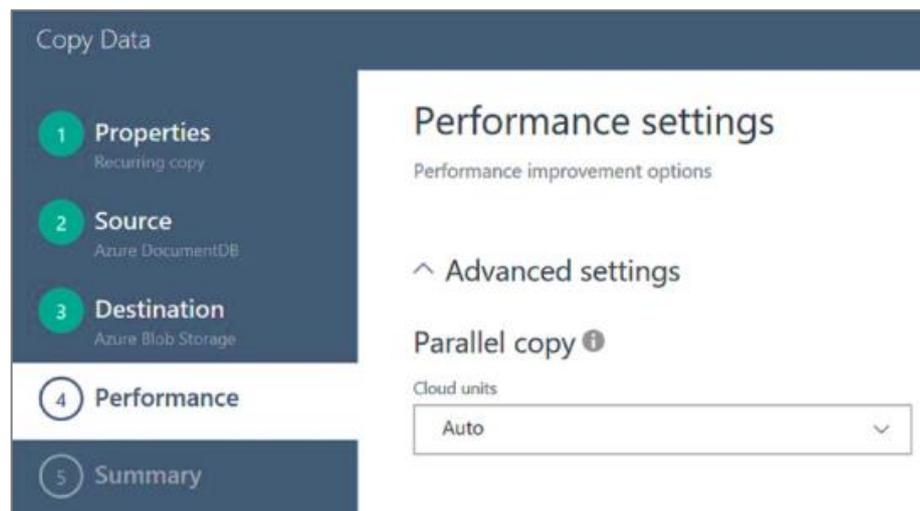
Masukkan refdata-team sebagai jalur folder dan TeamReferenceData.csv sebagai nama file untuk output tujuan. Kemudian klik tombol Berikutnya untuk melanjutkan ke layar

berikutnya, seperti yang ditunjukkan pada Gambar 6-21, di mana Anda akan menentukan pengaturan format file.



Gambar 6.21. Wizard Data Salin Pabrik Data: tentukan pengaturan format file

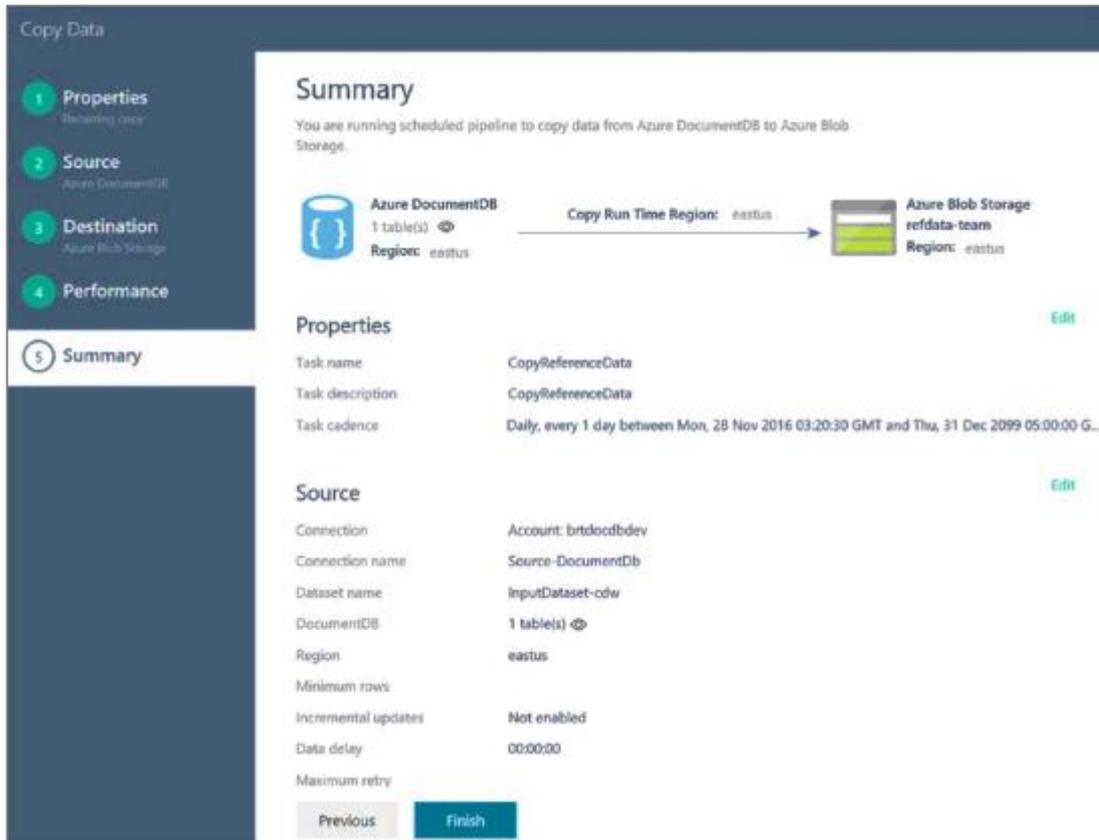
Pertahankan default untuk opsi pengaturan format file, tetapi pastikan untuk mencentang opsi Tambahkan Header ke File sehingga nama kolom dipertahankan. Klik tombol Berikutnya untuk melanjutkan ke layar berikutnya, pengaturan kinerja, seperti yang ditunjukkan pada Gambar 6.22.



Gambar 6.22. Wizard Data Salin Pabrik Data: tentukan pengaturan kinerja

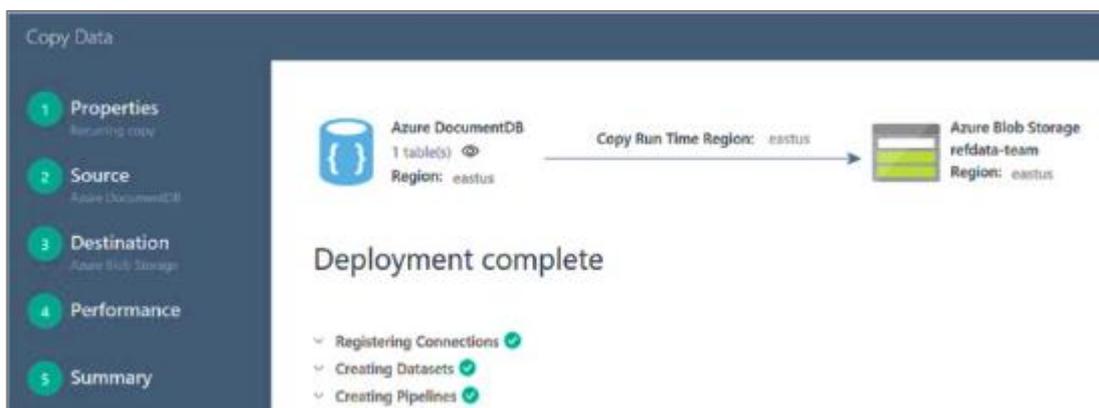
Untuk pengaturan kinerja, pertahankan default dan klik tombol Berikutnya. Pada titik ini, Anda akan melihat halaman ringkasan yang merangkum semua properti dan pengaturan

yang Anda tentukan untuk pekerjaan penyalinan baru ini, seperti yang ditunjukkan pada Gambar 6.23.



Gambar 6.23. Halaman ringkasan Panduan Salin Data Pabrik Data

Jika semua pengaturan terlihat bagus, klik tombol Selesai. Pekerjaan penyalinan Data Factory baru akan divalidasi dan disebar. Jika sudah selesai, Anda akan melihat layar seperti Gambar 6.24.



Gambar 6.24. Wizard Data Salin Pabrik Data, penerapan selesai

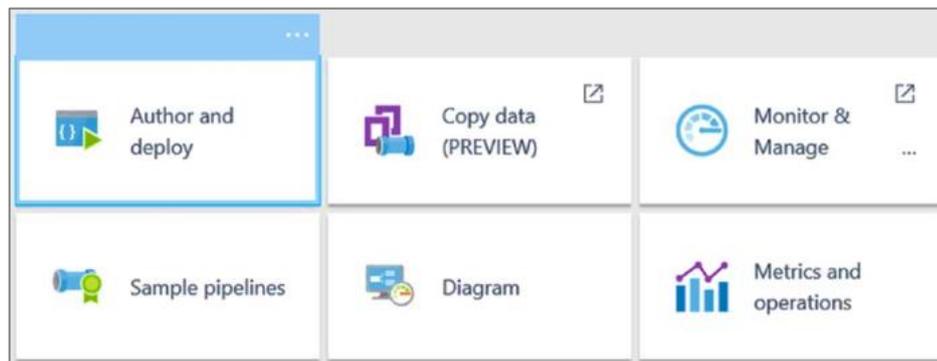
Ubah Parameter JSON untuk Pekerjaan Salin

Pada titik ini, Anda telah membuat pekerjaan pipa salinan Azure Data Factory dasar untuk implementasi referensi guna menyegarkan data referensi yang digunakan dalam pekerjaan Stream Analytics di Bab 5. Pipa salinan ini akan memilih data dari database Azure DocumentDB (melalui SQL pernyataan kueri) lalu tuliskan ke file CSV di penyimpanan blob Azure. Satu-satunya perubahan kecil yang tersisa adalah mengubah parameter keluaran JSON untuk membuat nama file keluaran di penyimpanan blob Azure menggunakan pola konvensi penamaan file seperti yang ditunjukkan: TeamReferenceDatayyyy-MM-ddhh-mm.csv

Perhatikan format penamaan tanggal/waktu berikut ini:

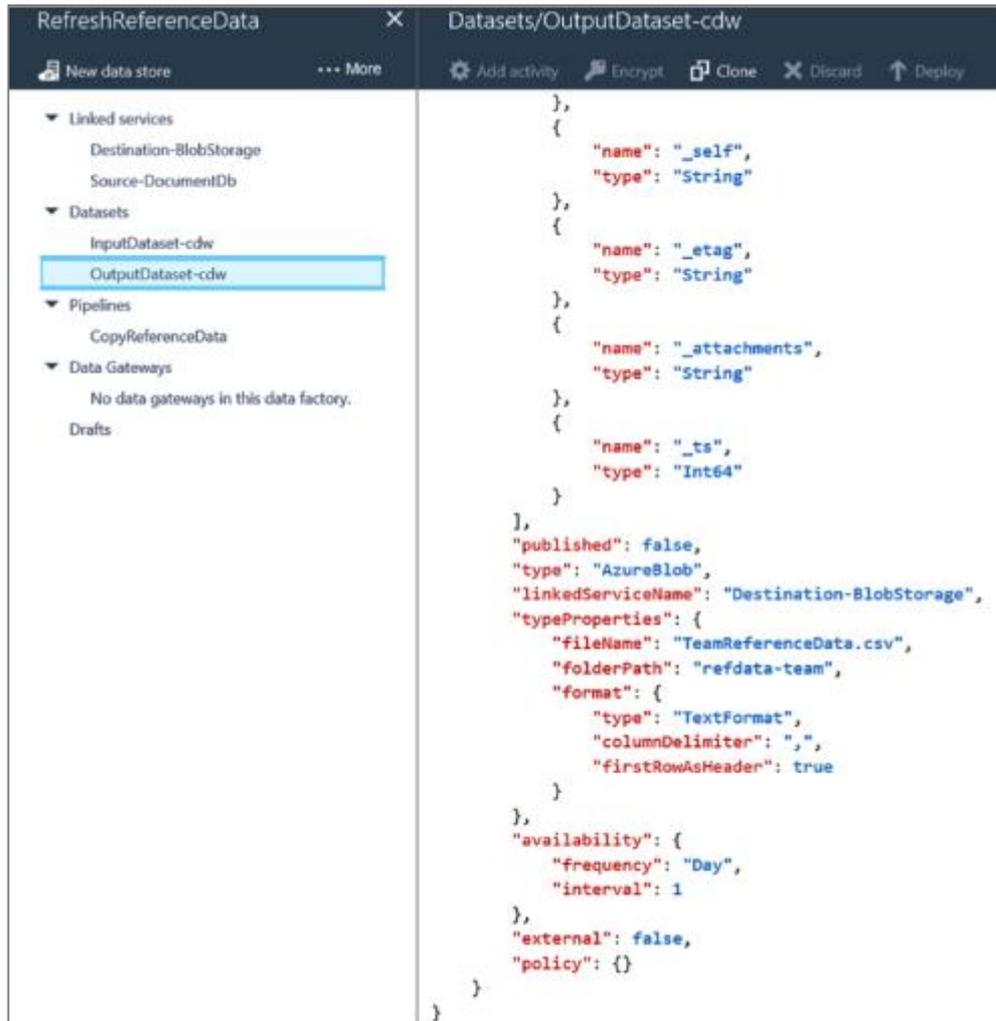
- yyyy = Tahun
- MM = Bulan
- dd = Hari
- hh = Jam
- mm = Menit

Perubahan konvensi penamaan ini sangat penting untuk memastikan bahwa Anda tidak mencoba memperbarui file saat sedang digunakan dan mungkin terkunci. Ini juga memungkinkan Anda membuat inventaris historis dari versi sebelumnya. Untuk melakukan perubahan, Anda perlu menggunakan JSON dari pekerjaan alur Azure Data Factory yang baru, menavigasi ke pekerjaan alur baru melalui Portal Azure, dan memilih opsi Penulisan dan Penyebaran, seperti yang ditunjukkan pada Gambar 6.25.



Gambar 6.25. Opsi Pabrik Data, Penulis, dan Penerapan

Setelah Anda memilih opsi ini, Anda akan melihat layar seperti Gambar 6.26. Di sini, Anda dapat mengklik setiap komponen alur dan memperluas parameter di bawah setiap bagian.



Gambar 6.26. Pabrik Data, melihat komponen pabrik data

Saat Anda mengklik setiap bagian, Anda akan melihat templat dan parameter JSON yang sesuai di sisi kanan halaman web Portal Azure. Klik pada OutputDataset-cdw di bawah bagian Datasets untuk melihat JSON yang sesuai. Gulir ke bawah dan Anda akan melihat bagian kode yang mirip dengan Gambar 6.27.

```

"typeProperties": {
  "fileName": "TeamReferenceData.csv",
  "folderPath": "refdata-team",
  "format": {
    "type": "TextFormat",
    "columnDelimiter": ",",
    "firstRowAsHeader": true
  }
},
},

```

Gambar 6.27. Pabrik Data: penamaan file keluaran default di JSON

Perhatikan parameter fileName, yang disetel ke nilai TeamReferenceData.csv. Ini akan menjadi bagian kode JSON yang akan Anda modifikasi agar memenuhi konvensi penamaan file

TeamReferenceData" + "yyyy-MM-ddhh-mm.csv". Untuk melakukannya, cari dan ganti kode JSON sebelumnya dengan kode JSON berikut:

```
"typeProperties": {
  "fileName": "TeamReferenceData{slice}.csv",
  "folderPath": "refdata-team",
  "format": {
    "type": "TextFormat",
    "columnDelimiter": ",",
    "firstRowAsHeader": true
  },
  "partitionedBy": [
    {
      "name": "slice",
      "value": {
        "type": "DateTime",
        "date": "SliceStart",
        "format": "yyyy-MM-ddhh-mm"
      }
    }
  ]
}
```

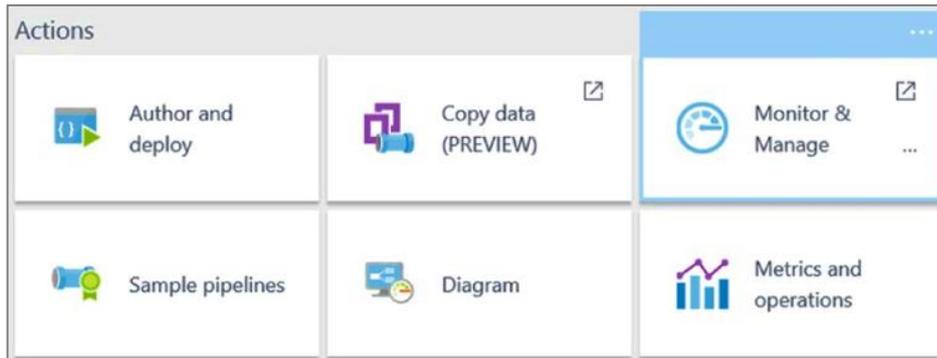
Perhatikan bahwa kode JSON yang Anda ganti akan menggunakan konvensi penamaan file dinamis berdasarkan atribut Tanggal dan Waktu untuk potongan pemrosesan tertentu yang dibuat dari pekerjaan alur Data Factory. Setelah Anda melakukan pembaruan pada JSON ini, opsi Deploy akan tersedia, seperti yang ditunjukkan pada Gambar 6.28.



Gambar 6.28. Pabrik Data: terapkan JSON yang diperbarui

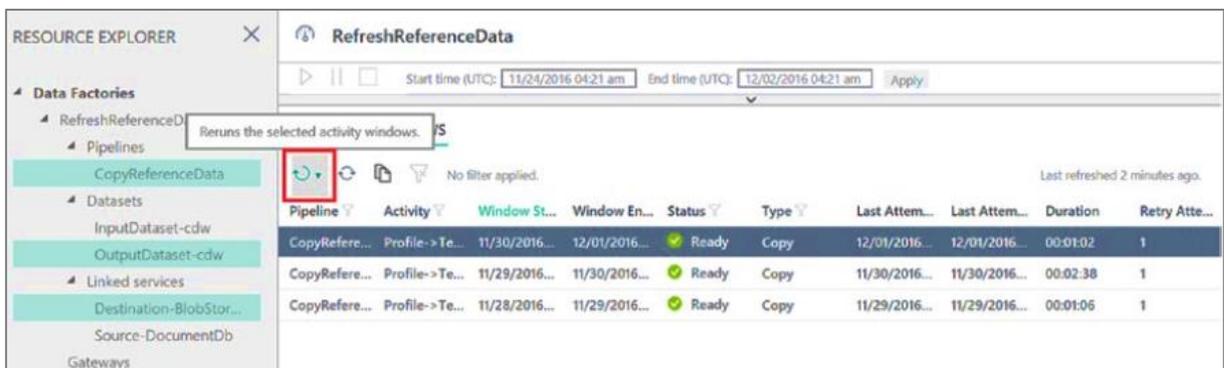
Setelah memilih opsi untuk menyebarkan data JSON alur yang diperbarui ke Azure, JSON akan disimpan, divalidasi, dan kemudian disebarkan ke layanan Azure Data Factory. Jalankan Pekerjaan Salin Data Factory Sesuai Permintaan dan Periksa Hasilnya

Untuk menjalankan pekerjaan alur Azure Data Factory baru ini, navigasikan ke pekerjaan melalui Portal Azure dan pilih opsi Pantau & Kelola, seperti yang ditunjukkan pada Gambar 6.29.



Gambar 6.29. Opsi Pantau & Kelola Pabrik Data

Setelah memilih opsi ini, tab baru akan terbuka di browser Anda. Aplikasi Penjelajah Sumber Daya Pabrik Data akan membuka dan menampilkan pekerjaan Pabrik Data Anda serta definisi dan aktivitas alur terkait, seperti yang ditunjukkan pada Gambar 6-30.

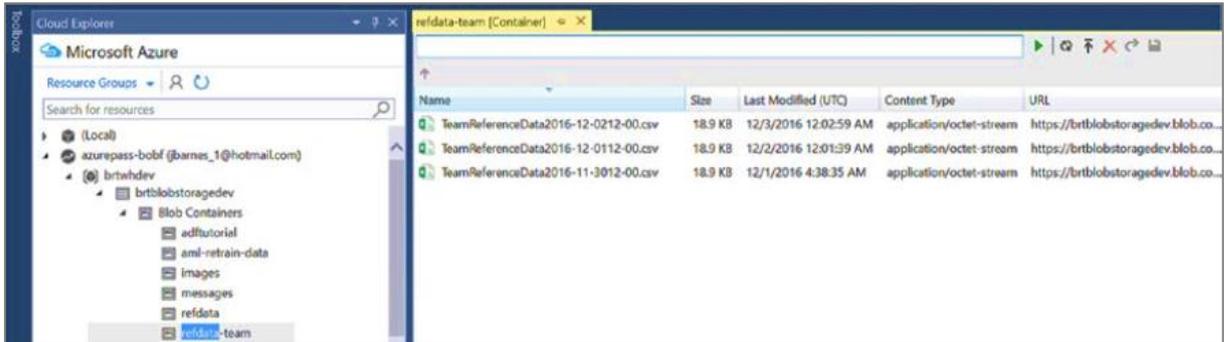


Gambar 6.30. Pabrik Data, jalankan kembali aktivitas pekerjaan

Untuk menjalankan pekerjaan saluran penyalinan yang diperbarui ini “sesuai permintaan”, lakukan hal berikut:

- Pilih aktivitas saluran terbaru.
- Klik pada ikon Jalankan Ulang, seperti yang disorot dengan warna merah pada Gambar 6-30.
- Pilih Jalankan Ulang atau Jalankan Ulang dengan Data Hulu.

Setelah beberapa menit, pekerjaan saluran penyalinan akan berjalan hingga selesai. Pada titik ini, Anda dapat memeriksa tim refdata kontainer tujuan keluaran blob Azure dan mencari file dengan konvensi penamaan yang mengikuti TeamReferenceDatayyyy-MM-ddhh-mm.csv. Tangkapan layar pada Gambar 6-31 menampilkan keluaran yang berhasil dari beberapa file CSV Data Referensi Tim selama jangka waktu tiga hari menggunakan alat Visual Studio Cloud Explorer untuk melihat kontainer blob keluaran.



Gambar 6.31. Pabrik Data: keluaran harian yang sukses dari file keluaran CSV data referensi tim

Untuk meringkas latihan ini, Anda baru saja menjalani semua langkah yang diperlukan untuk memperbarui data referensi yang Anda gunakan untuk pekerjaan Azure Stream Analytics di Bab 5.

6.5 AKTIVITAS SUMBER DAYA PEMBARUAN AZURE ML DATA FACTORY

Di bagian berikutnya, kami menangani tujuan kedua untuk tugas alur Azure Data Factory, yaitu melatih ulang model Azure Machine Learning melalui pekerjaan alur Azure Data Factory. Pembaruan pada model Azure Machine Learning akan berasal dari hasil uji stres medis yang diberikan kepada anggota tim secara berkala dan kemudian diunggah ke penyimpanan blob Azure untuk melatih ulang model tersebut. Kami membahas rincian spesifik penerapan model Azure Machine Learning dan Layanan Web terkait di Bab 9. Untuk latihan ini, kami berasumsi bahwa model prediktif Azure ML telah diperluas dengan titik akhir layanan web tambahan. Titik akhir tambahan akan memungkinkan pelatihan ulang model (dalam mode batch) berdasarkan data pelatihan yang baru saja diperbarui.

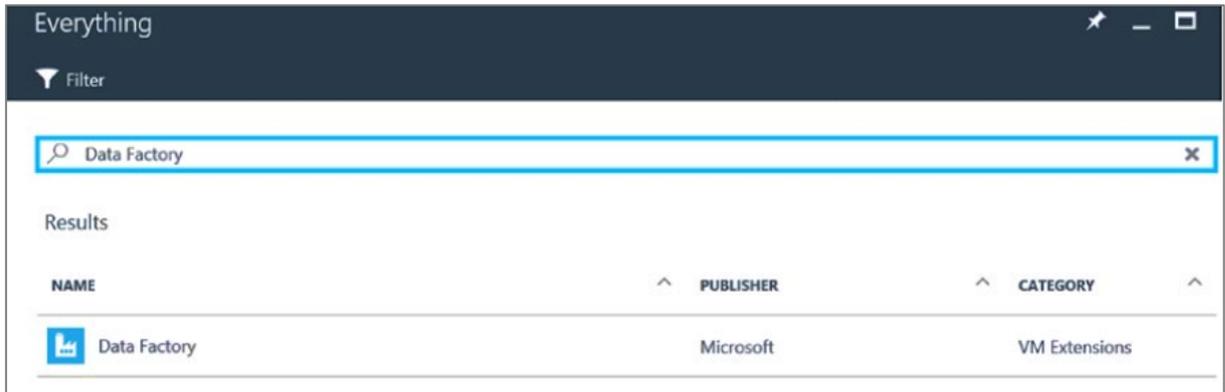
Pelatihan Ulang AML Pabrik Data: Desain Tingkat Tinggi

Pada tingkat tinggi, kita akan membuat pekerjaan alur Data Factory yang akan mencapai dua tujuan untuk sepenuhnya melatih kembali model Azure ML kita:

1. Memproses data pelatihan Machine Learning yang diperbarui dan menghasilkan File keluaran .iLearner ML. File ini kemudian menjadi masukan untuk aktivitas Update Resource pada langkah berikutnya.
2. Tambahkan Aktivitas Sumber Daya Pembaruan kedua ke alur untuk memperbarui Layanan Web Azure ML yang ada dengan model terlatih yang diperbarui melalui file keluaran .iLearner ML.

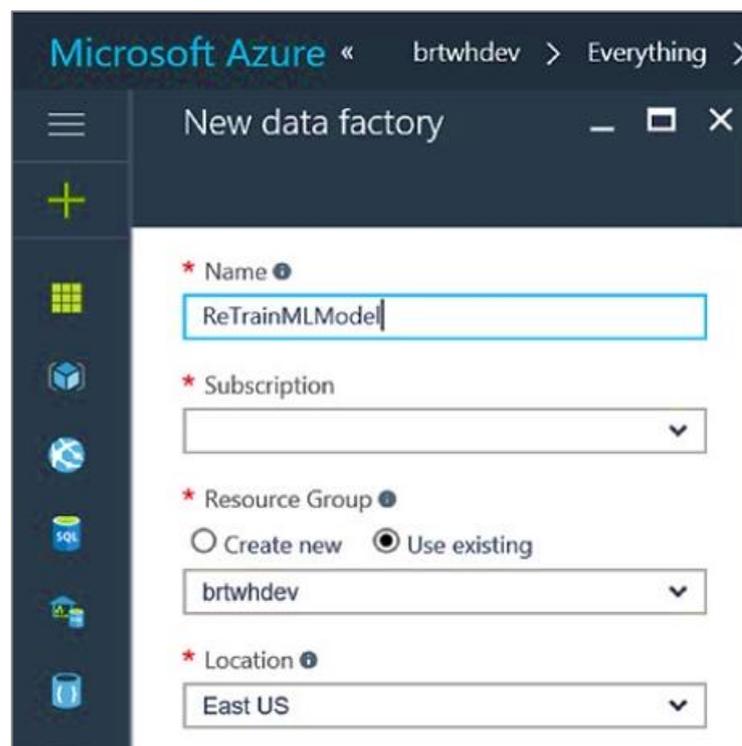
Buat Tugas Pabrik Data untuk Melatih Kembali Model Azure ML

Untuk memulai, navigasikan ke grup sumber daya untuk penerapan Anda melalui Portal Azure. Klik tombol + Tambah dan cari Data Factory, seperti yang ditunjukkan pada Gambar 6-32.



Gambar 6.32. Mencari dan menambahkan pekerjaan Data Factory ke grup sumber daya

Setelah memilih Azure Data Factory, Anda akan melihat layar gambaran umum Data Lake. Klik tombol Buat untuk membuat pekerjaan Data Factory baru. Layar berikutnya memungkinkan Anda memasukkan parameter spesifik untuk membuat pekerjaan Azure Data Factory baru, seperti yang ditunjukkan pada Gambar 6.33.

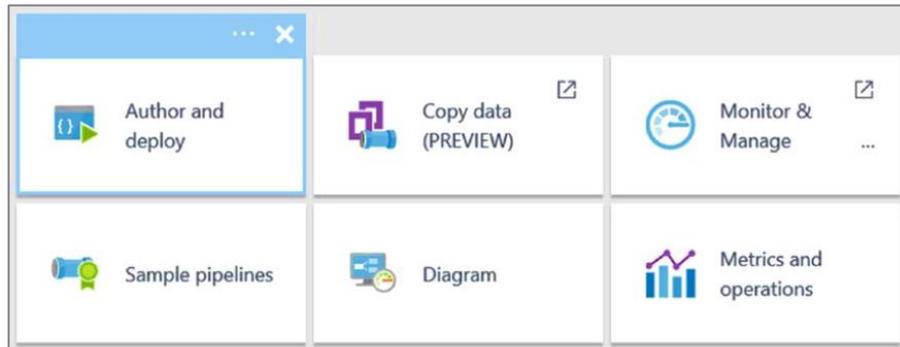


Gambar 6.33. Pabrik Data membuat parameter pekerjaan

Isi pilihan Anda untuk nilai parameter yang sesuai:

- *Nama:* Masukkan nama unik untuk Pabrik Data baru Anda. Perhatikan bahwa nama Azure Data Factory harus unik secara global. Kami menggunakan ReTrainMLModel.
- *Berlangganan:* Langganan Azure yang digunakan untuk pekerjaan ini.
- *Grup Sumber Daya:* Grup Sumber Daya Azure tempat layanan ini dibuat.
- *Lokasi:* Lokasi Pusat Data Azure.

Setelah selesai, klik tombol Buat di bagian bawah layar. Masukkan Anda kemudian akan divalidasi dan pekerjaan Azure Data Factory baru akan dibuat setelah jangka waktu singkat. Setelah pekerjaan Anda disediakan, navigasikan ke Pabrik Data baru melalui Portal Azure dan pilih opsi Penulisan dan Penyebaran, seperti yang ditunjukkan pada Gambar 6-34.

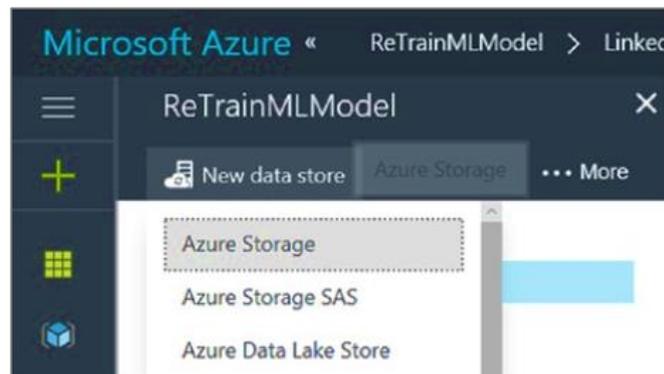


Gambar 6.34 Opsi Penulisan dan Penerapan Pabrik Data

Selanjutnya, Anda akan membuat komponen alur Data Factory hanya menggunakan JSON untuk menentukan elemen individual dari tugas alur pelatihan ulang Azure ML Data Factory.

Tentukan Layanan Tertaut: Azure Storage

Untuk menentukan Layanan Tertaut untuk Azure Storage, klik ikon Penyimpanan Data Baru di bilah navigasi atas, lalu pilih Azure Storage, seperti yang ditunjukkan pada Gambar 6-35.



Gambar 6.35 Pabrik Data: tambahkan penyimpanan data baru

Setelah layanan tertaut baru Anda dibuat, ganti JSON default dengan kode JSON yang ditampilkan di sini. Perhatikan bahwa Anda harus memiliki kredensial penyimpanan blob Azure spesifik untuk diisi.

```
{
}
"name": "AzureStorageLinkedService",
"properties": {
  "description": "",
  "hubName": "retrainmlmodel_hub",
```

```

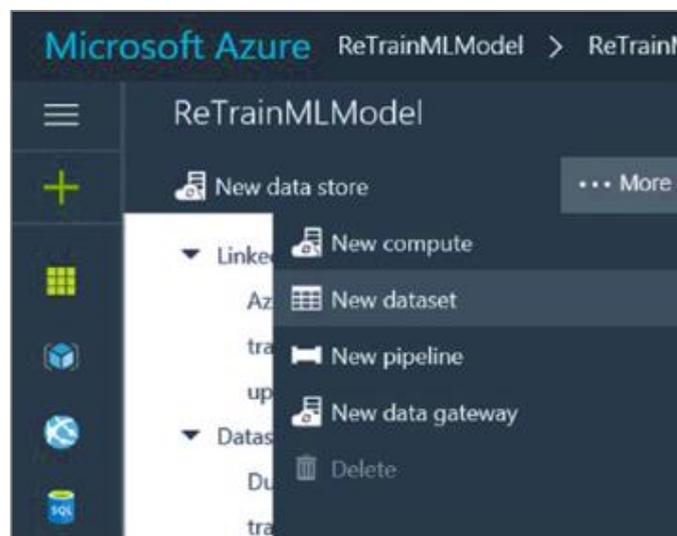
"type": "AzureStorage",
"typeProperties": {
  "connectionString":
"DefaultEndpointsProtocol=https;AccountName=<YourAccountName>;AccountKey=<YourAccountKey>"
}
}

```

Selanjutnya, klik ikon Deploy di bilah navigasi atas dan JSON Anda akan diunggah, divalidasi, dan disebar.

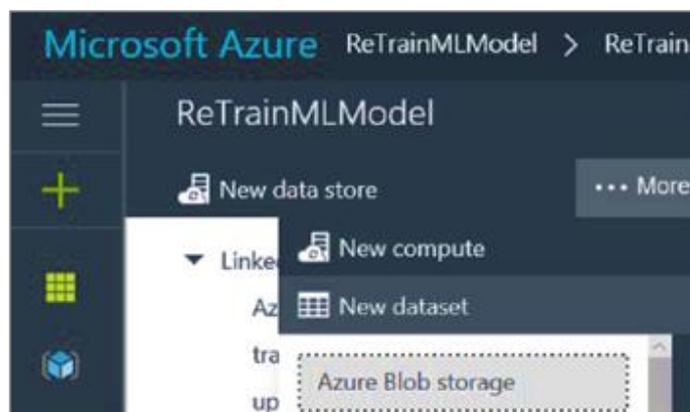
Tentukan Himpunan Data Masukan: Data Pelatihan Azure ML yang Diperbarui

Untuk menentukan dataset baru, klik ikon ...More lalu klik ikon New Dataset, seperti yang ditunjukkan pada Gambar 6-36.



Gambar 6.36. Pabrik Data: tambahkan kumpulan data baru, data pelatihan

Selanjutnya, pilih opsi untuk Azure Blob Storage, seperti yang ditunjukkan pada Gambar 6.37.



Gambar 6.37 Pabrik Data: tambahkan himpunan data baru, opsi penyimpanan blob Azure

Setelah dataset baru Anda dibuat, ganti JSON default dengan kode JSON yang ditampilkan di sini.

```
{
  "name": "trainingData",
  "properties": {
    "published": false,
    "type": "AzureBlob",
    "linkedServiceName": "AzureStorageLinkedService",
    "typeProperties": {
      "fileName": "RETRAIN_Teammates_AML_Training_Data.csv",
      "folderPath": "aml-retrain-data",
      "format": {
        "type": "TextFormat"
      }
    }
  },
  "availability": {
    "frequency": "Week",
    "interval": 1
  },
  "external": true,
  "policy": {
    "externalData": {
      "retryInterval": "00:01:00",
      "retryTimeout": "00:10:00",
      "maximumRetry": 3
    }
  }
}
```

Pastikan parameter `fileName` dan `folderPath` diatur untuk lingkungan Anda. Perhatikan bahwa frekuensinya adalah seminggu sekali. Selanjutnya, klik ikon Deploy di bilah navigasi atas dan JSON Anda akan diunggah, divalidasi, dan disebar ke definisi Data Factory.

Tentukan Himpunan Data Keluaran: Model Pelatihan Azure ML yang Diperbarui

Definisi himpunan data ini akan mewakili output file `.iLearner` dari layanan web pelatihan Azure ML. Aktivitas Eksekusi Batch Azure ML menghasilkan himpunan data ini. Himpunan data ini juga akan berfungsi sebagai file input untuk aktivitas Sumber Daya Pembaruan Azure ML.

Buat kumpulan data tambahan untuk pekerjaan Pabrik Data kami dengan mengikuti petunjuk yang sama untuk membuat kumpulan data baru dan dirangkum di sini:

- Klik ikon ...Lainnya lalu klik ikon Kumpulan Data Baru.
- Pilih opsi untuk Azure Blob Storage.

Setelah dataset baru Anda dibuat, ganti JSON default dengan kode JSON yang ditampilkan di sini.

```

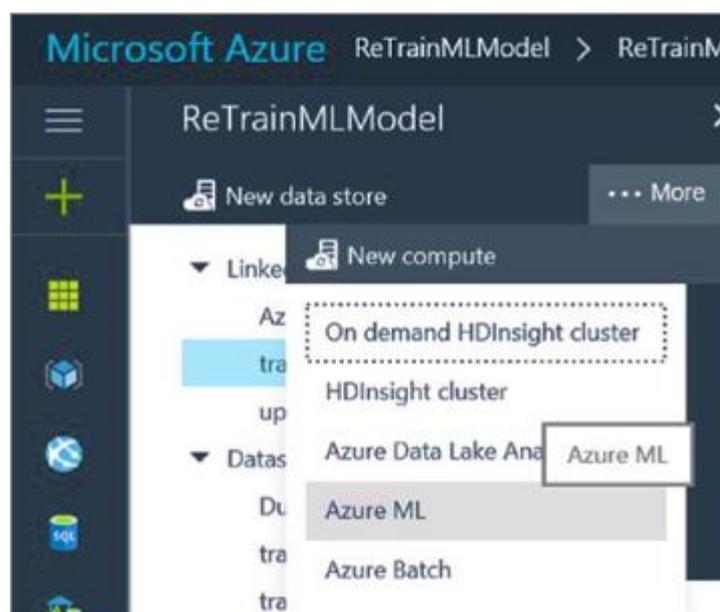
{
  "name": "trainedModelBlob",
  "properties": {
    "published": false,
    "type": "AzureBlob",
    "linkedServiceName": "AzureStorageLinkedService",
    "typeProperties": {
      "fileName": "model.ilearner",
      "folderPath": "aml-retrain-data",
      "format": {
        "type": "TextFormat"
      }
    }
  },
  "availability": {
    "frequency": "Week",
    "interval": 1
  }
}

```

Pastikan parameter `folderPath` diatur dengan benar untuk lingkungan Anda. Setelah selesai, klik ikon Deploy di bilah navigasi atas dan JSON Anda akan diunggah, divalidasi, dan disebarluaskan ke definisi Data Factory.

Tentukan Layanan Tertaut: Titik Akhir Pelatihan Azure ML

Selanjutnya, Anda akan membuat layanan tertaut yang mengarah ke titik akhir default layanan web pelatihan Azure ML. Untuk memulai, klik ikon ...Lainnya lalu klik ikon Komputasi Baru. Kemudian pilih opsi Azure ML, seperti yang ditunjukkan pada Gambar 6.38.



Gambar 6.38. Pabrik Data: tambahkan opsi penyimpanan blob Azure komputasi baru

Setelah layanan tertaut baru Anda dibuat, ganti JSON default dengan kode JSON yang ditampilkan di sini. Perhatikan bahwa Anda harus memiliki titik akhir URL Azure ML dan kunci API spesifik untuk mengisi parameter:

```
{
  "name": "trainingEndpoint",
  "properties": {
    "hubName": "retrainmlmodel_hub",
    "type": "AzureML",
    "typeProperties": {
      "mlEndpoint": "<YourEndPointURL>",
      "apiKey": "<YourAPIKey>"
    }
  }
}
```

Selanjutnya, klik ikon Deploy di bilah navigasi atas dan JSON Anda akan diunggah, divalidasi, dan disebar.

Tentukan Layanan Tertaut: Titik Akhir Penilaian Azure ML yang Dapat Diperbarui

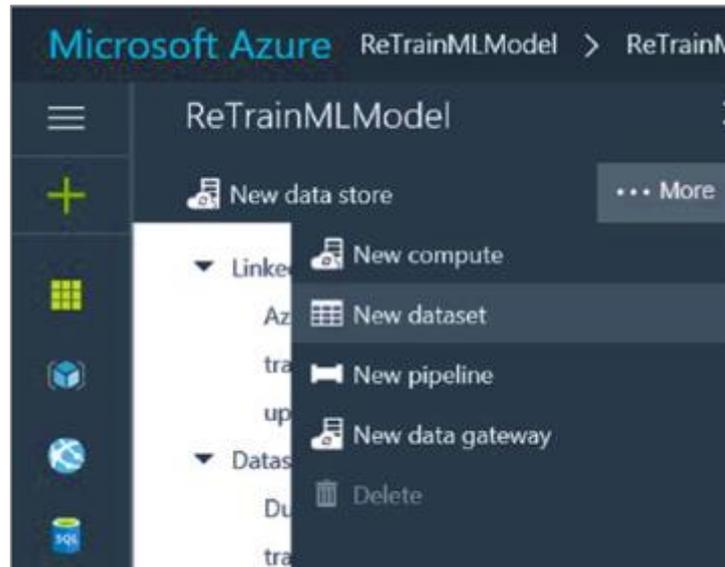
Selanjutnya, Anda akan membuat layanan tertaut yang mendefinisikan layanan tertaut Azure Machine Learning yang menunjuk ke titik akhir yang tidak dapat diperbarui secara default dari layanan Web Azure ML penskoran. Untuk memulai, klik ikon lainnya lalu klik ikon Komputasi Baru. Kemudian pilih opsi Azure ML, seperti yang ditunjukkan pada Gambar 6.38. Setelah layanan tertaut baru Anda dibuat, ganti JSON default dengan kode JSON yang ditampilkan di sini. Perhatikan bahwa Anda harus memiliki titik akhir URL Azure ML dan kunci API spesifik untuk mengisi parameter:

```
{
}
  "name": "updatableScoringEndpoint2",
  "properties": {
    "hubName": "retrainmlmodel_hub",
    "type": "AzureML",
    "typeProperties": {
      "mlEndpoint": "<YourMLRetrainingEndpoint>",
      "apiKey": "<YourMLRetrainingAPIKey>",
      "updateResourceEndpoint": "<YourMLRetrainingURLEndpoint>"
    }
  }
}
```

Selanjutnya, klik ikon Deploy di bilah navigasi atas dan JSON Anda akan diunggah, divalidasi, dan disebar.

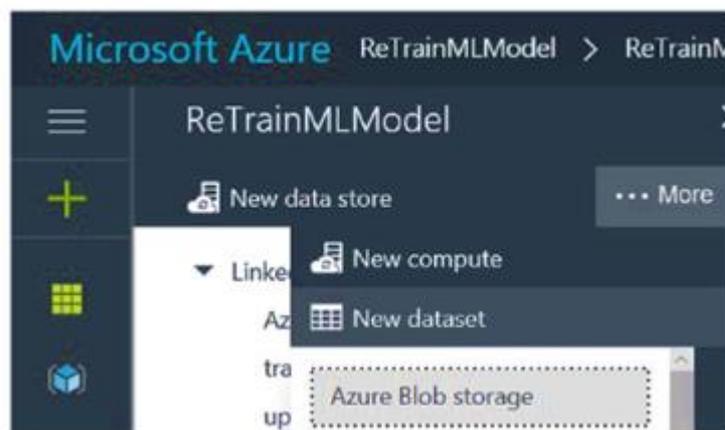
Tentukan Himpunan Data Output: Output Dummy Azure Blob

Pada saat penulisan ini, saat Anda menyertakan aktivitas sumber daya Pembaruan Azure ML dalam pekerjaan alur Data Factory, aktivitas tersebut tidak menghasilkan output apa pun. Namun, Azure Data Factory memerlukan himpunan data keluaran untuk mendorong jadwal alur. Oleh karena itu, kami akan mengimplementasikan himpunan data blob Azure dummy/placeholder untuk menangani kasus penggunaan ini. Untuk menentukan Dataset baru, klik ikon. More lalu klik ikon New Dataset, seperti yang ditunjukkan pada Gambar 6.39.



Gambar 6.39. Pabrik Data: tambahkan kumpulan data baru, data keluaran tiruan

Selanjutnya, pilih opsi untuk Azure Blob Storage, seperti yang ditunjukkan pada Gambar 6.40.



Gambar 6.40. Pabrik Data: tambahkan himpunan data baru, opsi penyimpanan blob Azure

Setelah dataset baru Anda dibuat, ganti JSON default dengan kode JSON yang ditampilkan di sini.

```

{
  "name": "DummyPlaceholderBlob",
  "properties": {
    "published": false,
    "type": "AzureBlob",
    "linkedServiceName": "AzureStorageLinkedService",
    "typeProperties": {
      "fileName": "dummyfile.csv",
      "folderPath": "aml-retrain-data",
      "format": {
        "type": "TextFormat"
      }
    },
    "availability": {
      "frequency": "Week",
      "interval": 1
    }
  }
}

```

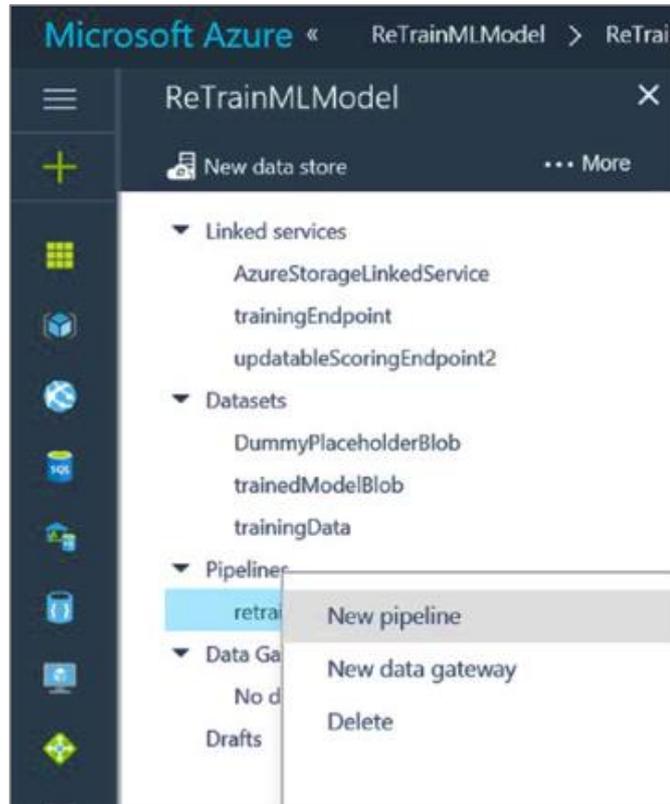
Pastikan parameter `fileName` dan `folderPath` diatur untuk lingkungan Anda. Perhatikan bahwa frekuensinya adalah seminggu sekali. Selanjutnya, klik ikon Deploy di bilah navigasi atas dan JSON Anda akan diunggah, divalidasi, dan disebar ke definisi Data Factory.

Tentukan Pekerjaan Pipa Pabrik Data dengan Dua Aktivitas

Sekarang, Anda akan menggabungkan semua layanan tertaut dan definisi himpunan data yang ditentukan sebelumnya saat Anda menentukan pekerjaan alur Data Factory yang baru. Pekerjaan alur Data Factory yang baru akan memiliki dua aktivitas yang ditentukan:

- *AzureMLBatchExecution*: Aktivitas Azure ML Batch Execution mengambil data pelatihan kesehatan tim yang diperbarui dari penyimpanan blob Azure sebagai input, lalu menghasilkan file `.iLearner` sebagai output.
- *AzureMLUpdateResource*: Aktivitas ini mengambil file `.iLearner` sebagai input dan kemudian mengirimkannya ke layanan web Pelatihan Azure ML untuk memperbarui model ML.

Untuk menentukan alur baru, klik kanan pada bagian Alur di bilah navigasi kiri panel Penulisan, lalu pilih Alur Baru, seperti yang ditunjukkan pada Gambar 6-41.



Gambar 6.41 Pabrik Data: tambahkan saluran baru

Setelah alur baru Anda dibuat, ganti JSON default dengan kode JSON yang ditampilkan di sini.

```
{
  "name": "retrainmlpipeline",
  "properties": {
    "activities": [
      {
        "type": "AzureMLBatchExecution",
        "typeProperties": {
          "webServiceInput": "trainingData",
          "webServiceOutputs": {
            "output1": "trainedModelBlob"
          },
          "webServiceInputs": {},
          "globalParameters": {}
        },
        "inputs": [
          {
            "name": "trainingData"
          }
        ],
        "outputs": [
          {
            "name": "trainedModelBlob"
          }
        ]
      }
    ]
  }
}
```

```

    ],
    "policy": {
      "timeout": "02:00:00",
      "concurrency": 1,
      "executionPriorityOrder": "NewestFirst",
      "retry": 1
    },
    "scheduler": {
      "frequency": "Week",
      "interval": 1
    },
    "name": "retraining",
    "linkedServiceName": "trainingEndpoint"
  },
  {
    "type": "AzureMLUpdateResource",
    "typeProperties": {
      "trainedModelDatasetName": "trainedModelBlob",
      "trainedModelName": "Training Exp for ADF ML
        [trained model]"
    },
    "inputs": [
      {
        "name": "trainedModelBlob"
      }
    ],
    "outputs": [
      {
        "name": "DummyplaceholderBlob"
      }
    ],
    "policy": {
      "timeout": "01:00:00",
      "concurrency": 1,
      "retry": 3
    },
    "scheduler": {
      "frequency": "Week",
      "interval": 1
    },
    "name": "AzureML Update Resource",
    "linkedServiceName": "updatableScoringEndpoint2"
  }
],
"start": "2016-02-13T00:00:00Z",
"end": "2016-02-14T00:00:00Z",
"isPaused": false,
"hubName": "retrainmlmodel_hub",

```

```

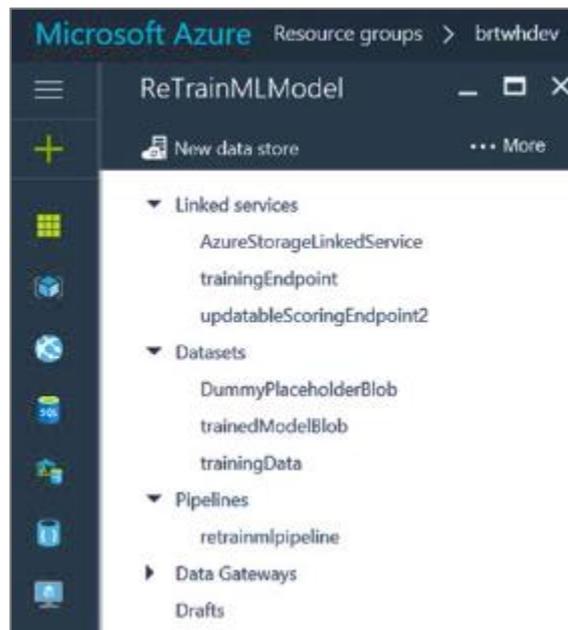
    "pipelineMode": "Scheduled"
  }
}

```

Pastikan parameter `fileName` dan `folderPath` diatur untuk lingkungan Anda. Perhatikan bahwa frekuensinya adalah seminggu sekali. Selanjutnya, klik ikon Deploy di bilah navigasi atas dan JSON Anda akan diunggah, divalidasi, dan disebar ke definisi Data Factory. Pada titik ini, Anda telah secara manual membuat pekerjaan alur Data Factory yang lengkap (dijelaskan melalui JSON) untuk memperbarui layanan pelatihan Azure ML.

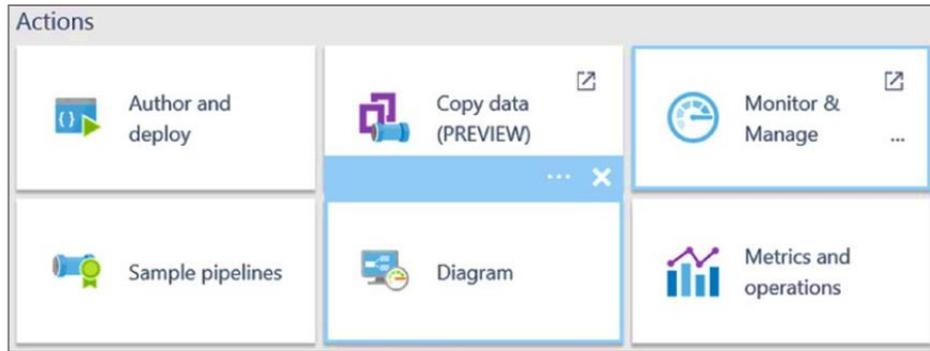
Alur Data Factory terdiri dari komponen-komponen berikut, seperti yang ditunjukkan pada Gambar 6.42.

- * * Tiga layanan terkait
- * * Tiga kumpulan data
- * * Satu saluran pipa



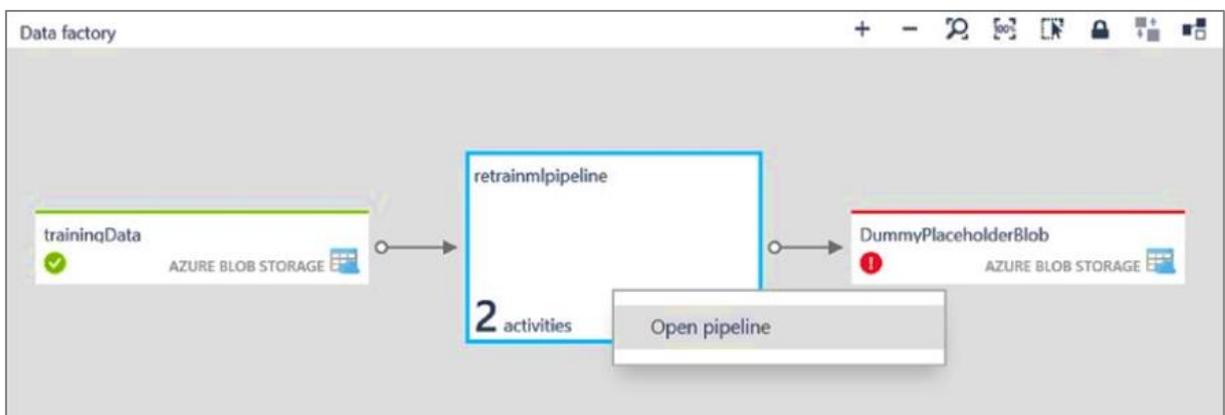
Gambar 6.42. Komponen alur Data Factory

Untuk mendapatkan representasi visual dari Azure Data Factory dan komponen yang telah Anda buat, navigasikan ke pekerjaan Data Factory Anda di Portal Azure dan pilih opsi Diagram, seperti yang ditunjukkan pada Gambar 6.43.



Gambar 6.43. Tampilan Diagram Pabrik Data

Setelah mengklik ikon Diagram, Anda akan melihat representasi alur Data Factory Anda, seperti yang ditunjukkan pada Gambar 6-44.



Gambar 6.44 Diagram Pabrik Data, opsi alur terbuka

Namun, Anda akan melihat bahwa tampilan ini masih dicitakan, karena ikon alur menyatakan bahwa tampilan tersebut memiliki dua aktivitas. Untuk memperluas pipeline, klik kanan pada ikon pipeline dan pilih Open Pipeline, seperti yang ditunjukkan pada Gambar 6.44. Saat Anda mengklik opsi Open Pipeline, layar akan diperluas untuk menyesuaikan jendela tampilan secara otomatis guna menampilkan keseluruhan pipeline. Gambar 6.45 mengilustrasikan pandangan baru.



Gambar 6.45 Diagram Pabrik Data, tampilan alur terbuka

Perhatikan bahwa tampilan ini sangat membantu saat menyusun pekerjaan alur multi-langkah menggunakan Azure Data Factory. Anda juga dapat menyesuaikan tingkat zoom dan menelusuri masing-masing komponen.

Ringkasnya, di bagian ini, Anda membuat pekerjaan alur Data Factory baru untuk memperbarui Layanan Web Machine Learning secara otomatis menggunakan definisi JSON berikut:

- Layanan Tertaut: Azure Storage
- Layanan Tertaut: Titik Akhir Pelatihan Azure ML
- Layanan Tertaut: Titik Akhir Penilaian Azure ML yang Dapat Diperbarui
- Input Dataset: Data Pelatihan Azure ML yang Diperbarui
- Himpunan Data Keluaran: Model Pelatihan Azure ML yang Diperbarui
- Himpunan Data Keluaran: Keluaran Dummy Azure Blob
- Pekerjaan Pipa Pabrik Data: Dengan dua Aktivitas

Bagian implementasi referensi ini menyediakan metode pelatihan ulang Azure Machine Learning Web Service yang sepenuhnya otomatis. Pembaruan tersebut didasarkan pada data kesehatan tim terkini yang dihasilkan dari hasil stress test berkala yang diberikan kepada anggota tim. Hal ini menciptakan solusi “siklus hidup penuh” untuk mempertahankan model pelatihan Azure ML yang diperbarui berdasarkan pembaruan data fisik.

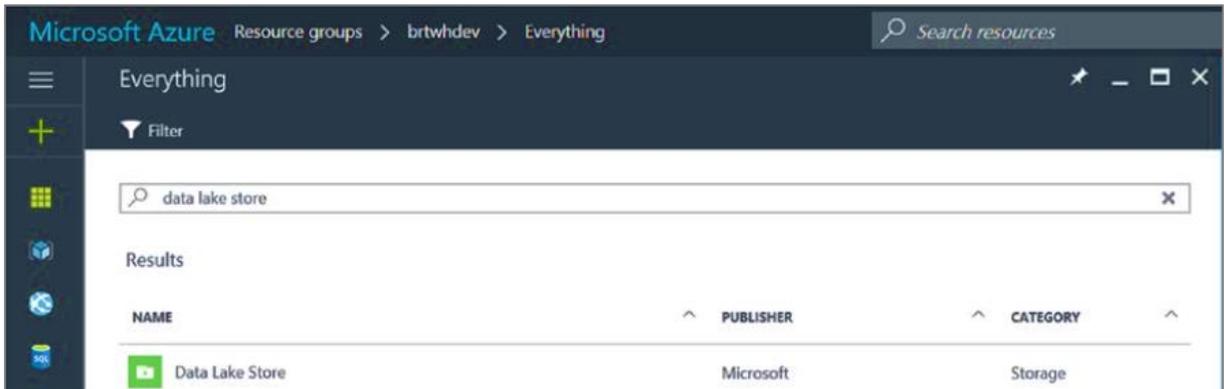
6.6 PINDAHKAN DATA DARI PENYIMPANAN BLOB KE DATA LAKE

Pekerjaan Data Factory ketiga dan terakhir yang akan kita terapkan dalam bab ini berfokus pada pemindahan data dari penyimpanan blob Azure ke Azure Data Lake. Pabrik Data ini akan menyiapkan data implementasi referensi untuk subjek berikutnya yang dibahas dalam Bab 7. Sebagai penyegaran, data dalam penyimpanan blob Azure yang akan Anda pindahkan ke Azure Data Lake dibuat sebagai hasil output dari pekerjaan analitik Azure Stream yang dibuat di Bab 5. Anda akan memindahkan data ini ke Azure Data Lake karena beberapa alasan, termasuk yang berikut:

- Arsip Data: Menyimpan data dalam bentuk aslinya seperti saat diterima dari IoT Hub.
- Analisis Mendalam: Azure Data Lake adalah platform penyimpanan dan analitik yang kuat, yang akan kita bahas lebih lanjut di Bab 7.
- Analisis Regresi: Seringkali, ada gunanya menjalankan kembali analisis Big Data atas data historis untuk mengevaluasi hasil alternatif atau membuat perbaikan (historis) lainnya.

Buat Akun Penyimpanan Azure Data Lake

Sebelum Anda dapat menyalin data dari penyimpanan blob Azure ke Azure Data Lake, Anda perlu membuat akun Azure Data Lake Store. Untuk memulai, navigasikan ke grup sumber daya untuk penerapan Anda melalui Portal Azure. Klik tombol + Tambah dan cari Data Lake, seperti yang ditunjukkan pada Gambar 6-46.



Gambar 6.46. Mencari dan menambahkan Penyimpanan Data Lake ke grup sumber daya

Selanjutnya, Anda akan melihat halaman gambaran umum tentang Azure Data Lake Store. Klik tombol Buat untuk melanjutkan ke layar berikutnya.

New Data Lake Store

Name: ✓
 brtadls.azuredatalakestore.net

* Subscription:

* Resource Group: Create new Use existing

* Location:

Pricing: 🔒

Encryption Settings: >

Gambar 6.47. Parameter untuk membuat Penyimpanan Data Lake baru

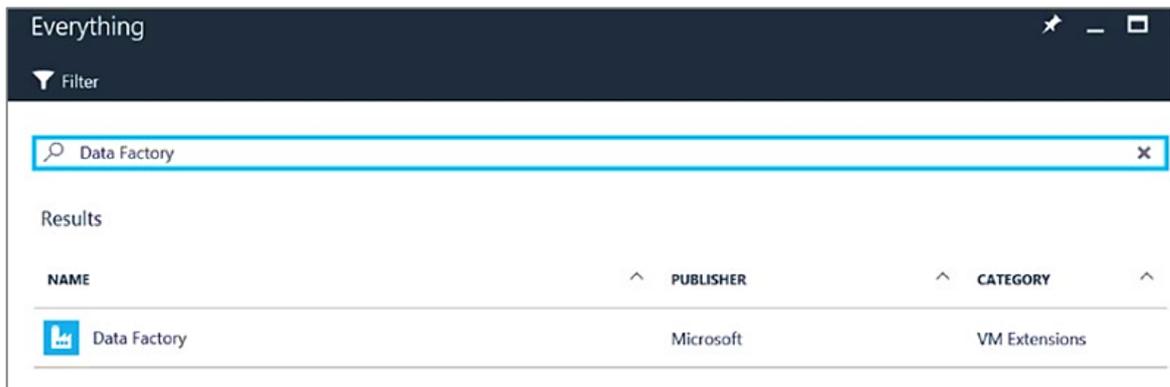
Isi pilihan Anda untuk nilai parameter yang sesuai:

- *Nama:* Masukkan nama unik untuk Penyimpanan Data Lake baru Anda.
- *Berlangganan:* Langganan Azure yang digunakan untuk pekerjaan ini.
- *Grup Sumber Daya:* Grup Sumber Daya Azure tempat provisi sumber daya ini.
- *Lokasi:* Lokasi Pusat Data Azure.

Setelah selesai, klik tombol Buat di bagian bawah layar. Masukan Anda kemudian akan divalidasi dan Azure Data Lake Store baru akan dibuat paling lama dalam beberapa menit.

Buat Pekerjaan Pabrik Data: Salin dari Azure Blob ke Data Lake

Untuk memulai, navigasikan ke grup sumber daya untuk penerapan Anda melalui Portal Azure. Klik tombol + Tambah dan cari Data Factory, seperti yang ditunjukkan pada Gambar 6-48.



Gambar 6.48. Mencari dan menambahkan pekerjaan Data Factory ke grup sumber daya

Setelah memilih Azure Data Factory, klik tombol Buat untuk membuat pekerjaan baru. Layar berikutnya akan memungkinkan Anda memasukkan parameter spesifik untuk membuat pekerjaan Azure Data Factory baru, seperti yang ditunjukkan pada Gambar 6.49.

 A screenshot of the 'New data factory' configuration form in the Azure Portal. The form contains several fields:

- Name:** A text input field containing 'CopyFromBlobToDataLake'.
- Subscription:** A dropdown menu.
- Resource Group:** Radio buttons for 'Create new' and 'Use existing' (selected), followed by a dropdown menu containing 'brtwhdev'.
- Location:** A dropdown menu containing 'East US'.

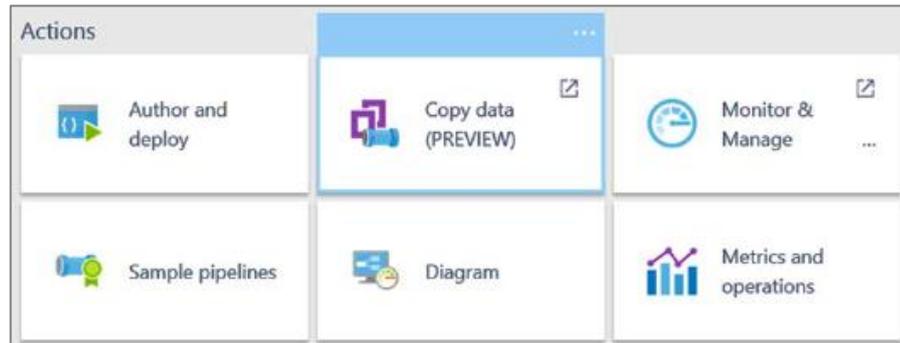
Gambar 6.49. Pabrik Data membuat parameter pekerjaan

Isi pilihan Anda untuk nilai parameter yang sesuai:

- *Nama:* Masukkan nama unik untuk tugas Pabrik Data baru Anda. Perhatikan bahwa nama Azure Data Factory harus unik secara global.
- *Berlangganan:* Langganan Azure yang digunakan untuk pekerjaan ini.

- *Grup Sumber Daya*: Grup Sumber Daya Azure tempat layanan ini dibuat.
- *Lokasi*: Lokasi Pusat Data Azure.

Setelah selesai, klik tombol Buat di bagian bawah layar. Masukan Anda kemudian akan divalidasi dan pekerjaan Azure Data Factory baru akan dibuat setelah jangka waktu singkat. Diperlukan waktu kurang dari satu menit melalui Portal Azure. Setelah pekerjaan Anda disediakan, navigasikan ke Pabrik Data baru melalui Portal Azure dan pilih opsi Salin Data (PREVIEW), seperti yang ditunjukkan pada Gambar 6.50.

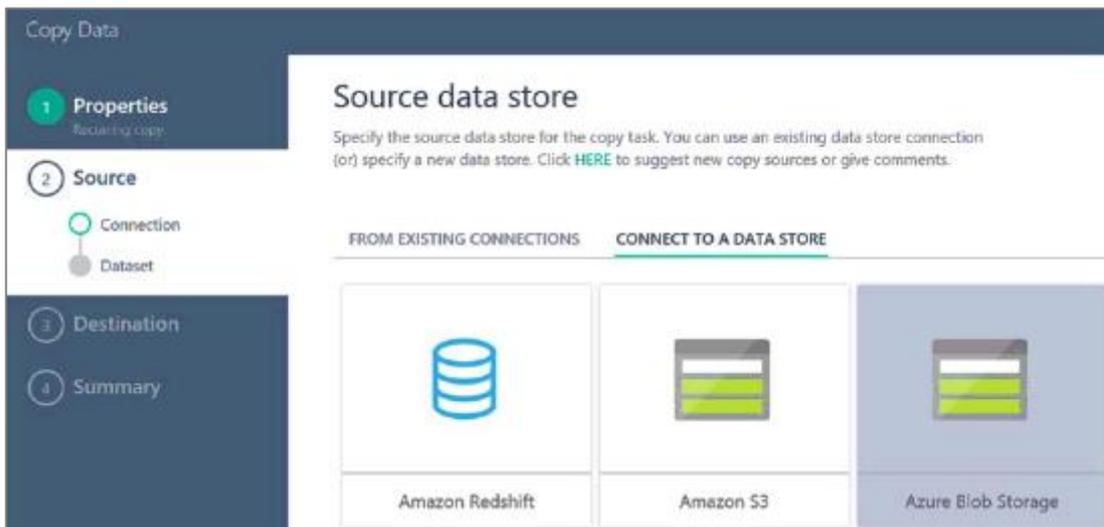


Gambar 6.50. Wisaya Penyalinan Data Pabrik Data

Ini akan memanggil Azure Data Factory Copy Data Wizard untuk diluncurkan dan akan memandu Anda melalui langkah-langkah yang diperlukan untuk membuat alur salinan dasar. Di balik layar, Azure Data Factory menghasilkan file JSON untuk mencerminkan pilihan Anda di Copy Wizard. Gambar 6.51 menggambarkan layar pertama dari Copy Data Wizard dan memungkinkan Anda menentukan properti untuk pekerjaan penyalinan.

Gambar 6.51. Wizard Data Salin Pabrik Data: tentukan properti

Untuk contoh ini, masukkan nama tugas CopyBlobToDataLake dan pertahankan sisa default untuk jadwal, status, dan tanggal akhir. Klik Berikutnya untuk melanjutkan ke langkah berikutnya. Gambar 6.52 menggambarkan layar Sumber Data, tempat Anda harus memilih Azure Blob Storage untuk skenario implementasi referensi.



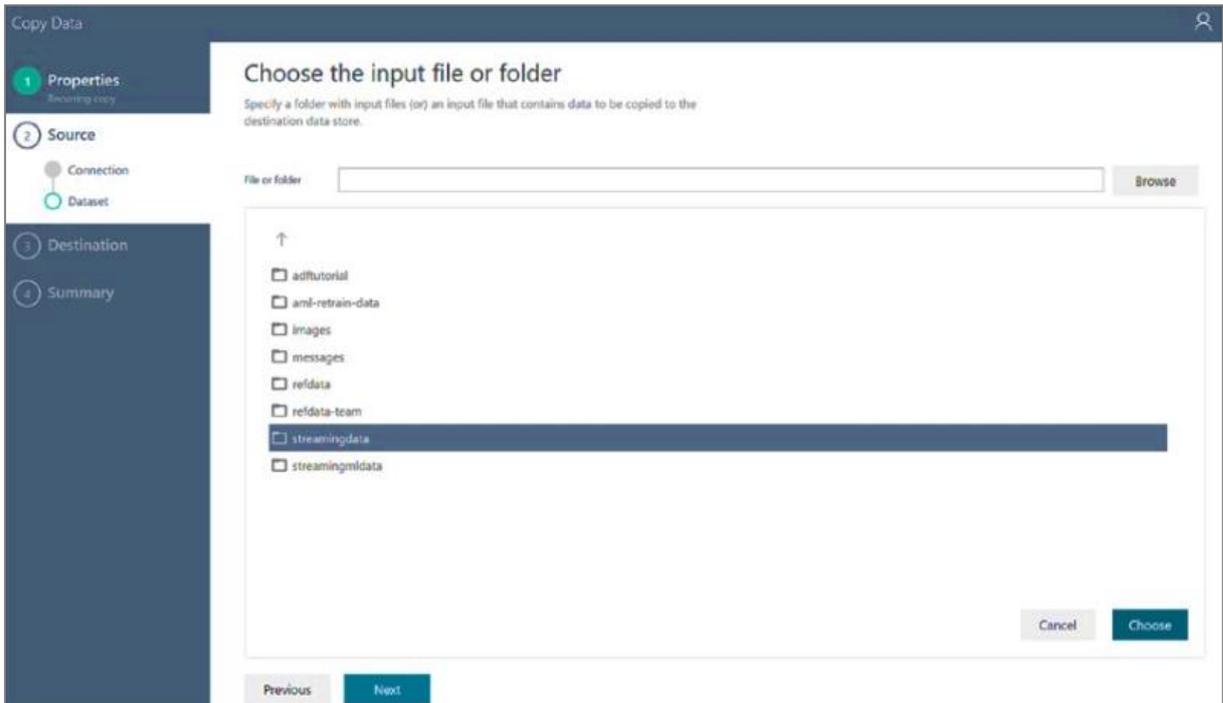
Gambar 6.52 Wizard Data Salin Pabrik Data: tentukan penyimpanan data sumber untuk penyimpanan blob Azure

Selanjutnya, tentukan properti Penyimpanan Data Sumber untuk penyimpanan blob Azure, seperti yang ditunjukkan pada Gambar 6.53.



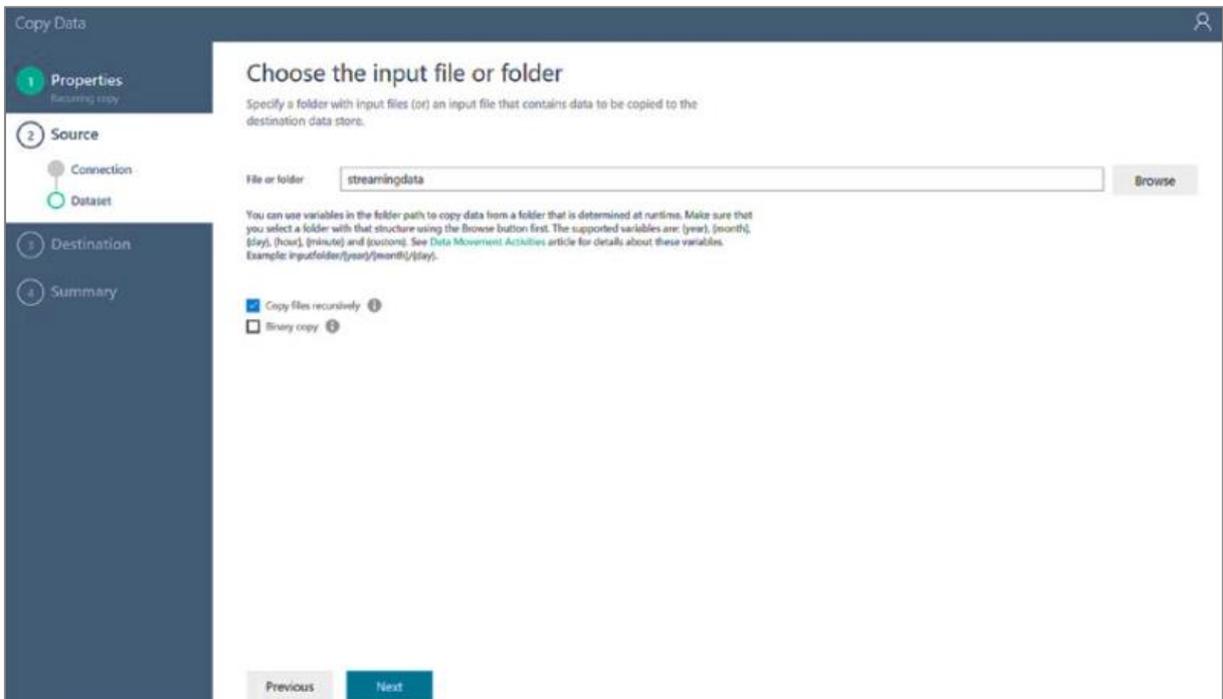
Gambar 6.53. Panduan Penyalinan Data Pabrik Data: tentukan penyimpanan data sumber untuk properti penyimpanan blob Azure

Selanjutnya, pilih folder di penyimpanan blob Azure yang akan berisi himpunan data sumber, seperti yang ditunjukkan pada Gambar 6.54.



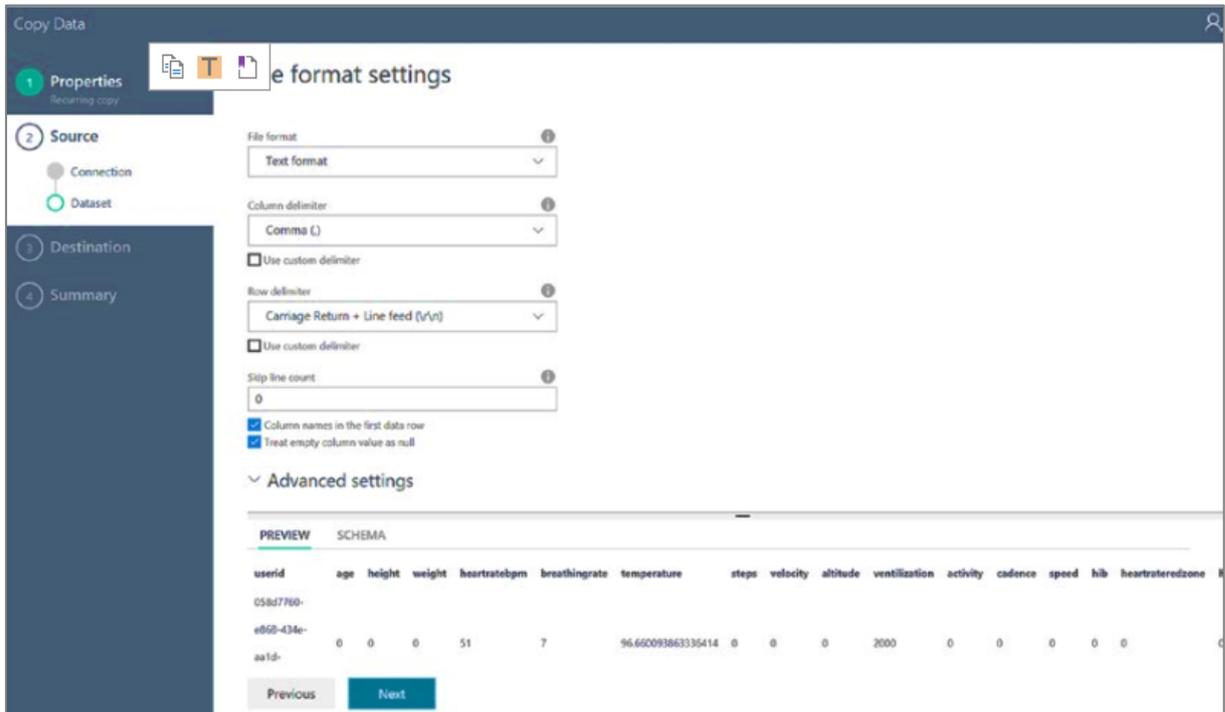
Gambar 6.54 Wizard Data Salin Pabrik Data: tentukan folder sumber di penyimpanan blob

Setelah memilih folder streamingdata dan mengklik Pilih, klik tombol Berikutnya. Anda akan melihat layar pada Gambar 6.55.



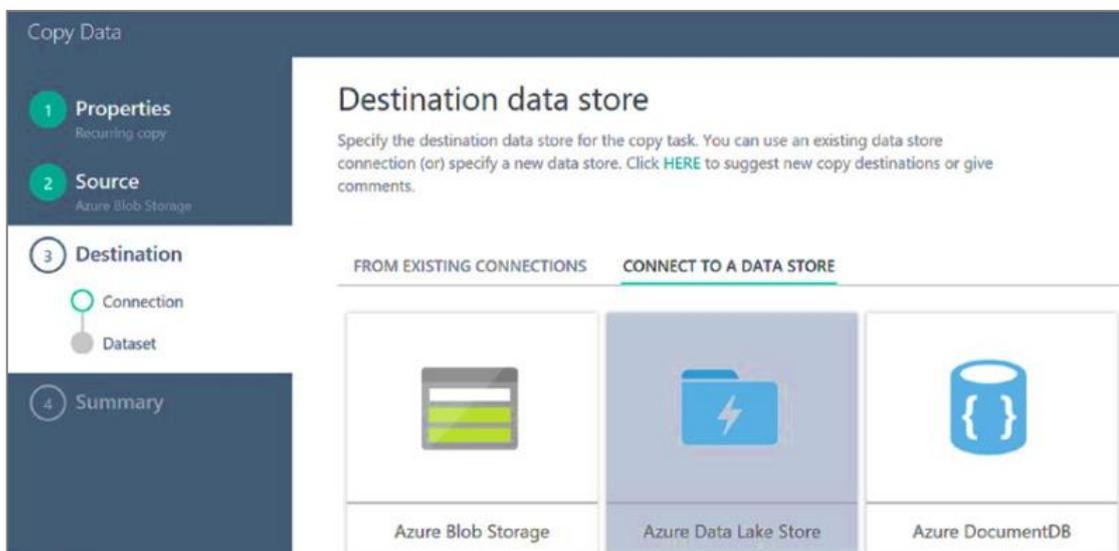
Gambar 6.55. Wizard Penyalinan Data Pabrik Data: tentukan folder sumber dan opsi untuk menyalin file secara rekursif

Pilih opsi untuk Menyalin File Secara Rekursif, seperti yang ditunjukkan pada Gambar 6.55, lalu klik tombol Berikutnya. Pada titik ini, Wizard Penyalinan Pabrik Data akan mencoba menyambungkan ke folder penyimpanan blob Azure dan secara otomatis mendeteksi format file dalam penyimpanan blob, seperti yang ditunjukkan pada Gambar 6.56.



Gambar 6.56. Wizard Data Salin Pabrik Data: pengaturan format file

Jika pengaturannya tidak cocok, Anda mungkin perlu meninjau output penyimpanan blob Azure dari pekerjaan Stream Analytics yang ditentukan di Bab 5. Jika pengaturan format file cocok dengan input yang Anda harapkan, klik tombol Berikutnya untuk memilih penyimpanan data tujuan, seperti yang ditunjukkan pada Gambar 6.57.



Gambar 6.57 Wizard Data Salin Pabrik Data: pilih tujuan Data Lake

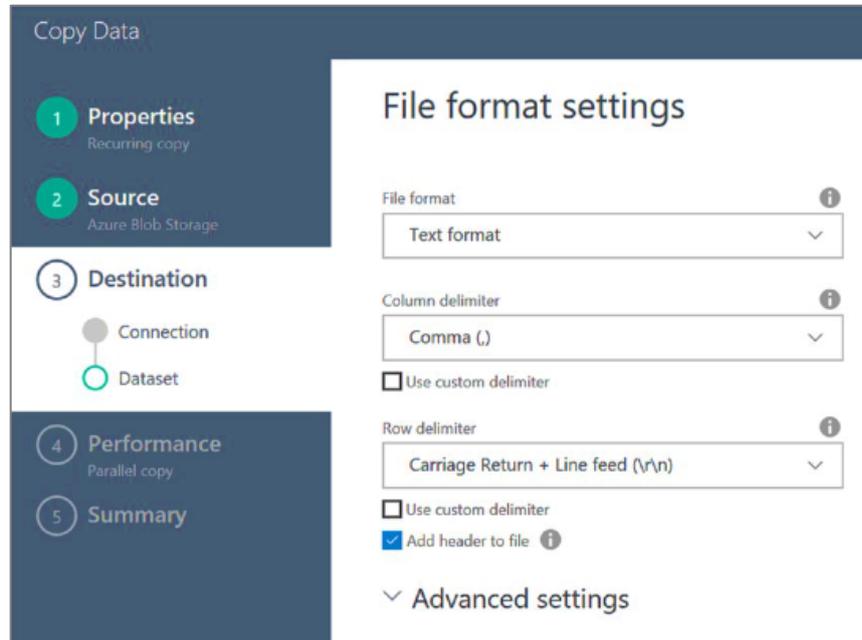
Klik ikon Azure Data Lake Store lalu klik tombol Berikutnya untuk memilih Data Lake sebagai tujuan keluaran. Opsi berikutnya yang akan Anda tentukan di Panduan Aktivitas Penyalinan adalah untuk akun Azure Data Lake Store tujuan, seperti yang ditunjukkan pada Gambar 6.58.

Gambar 6.58. Wizard Data Salin Pabrik Data: pilih tujuan Data Lake

Pilih langganan Azure Anda dan Nama Penyimpanan Azure Data Lake lalu klik tombol Berikutnya. Layar berikutnya di Copy Wizard akan memungkinkan Anda memilih folder tujuan output dan jalur file untuk data tujuan di Azure Data Lake. Atur parameter Nama file ke nilai `inputfolder/{year}/{month}/{day}`, seperti yang ditunjukkan pada Gambar 6.59.

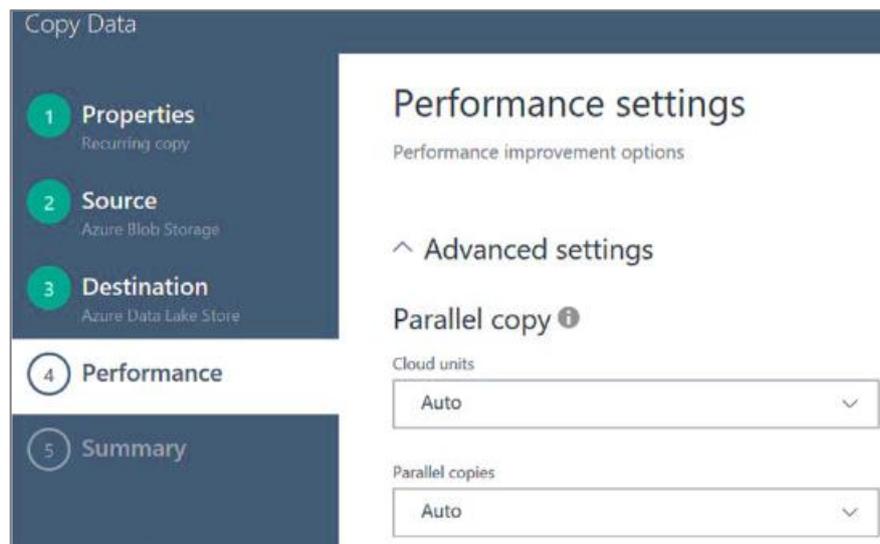
Gambar 6.59. Wizard Data Salin Pabrik Data: pilih folder Tujuan Data Lake dan nilai nama file Klik tombol Berikutnya untuk maju ke layar berikutnya di Copy Wizard.

Di sini, Anda akan melihat opsi untuk menentukan pengaturan format file. Pilih opsi Add Header to File, seperti yang ditunjukkan pada Gambar 6.60.



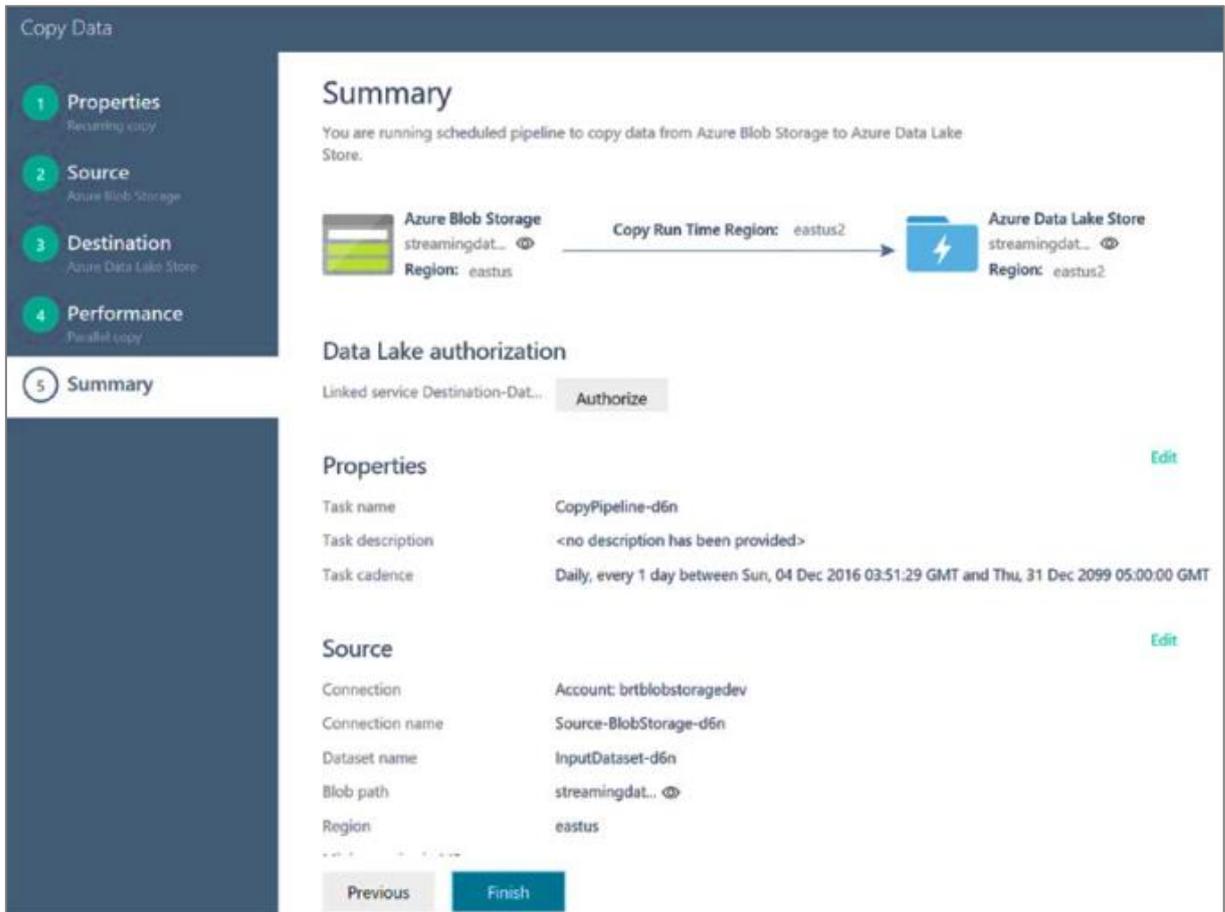
Gambar 6.60. Wizard Data Salin Pabrik Data: tentukan nilai format file tujuan Data Lake

Selanjutnya, Anda akan melihat layar di mana Anda dapat menyesuaikan pengaturan kinerja untuk pekerjaan penyalinan. Terima nilai default Auto for Cloud Units dan Parallel Copies dan klik ikon Next, seperti yang ditunjukkan pada Gambar 6-61.



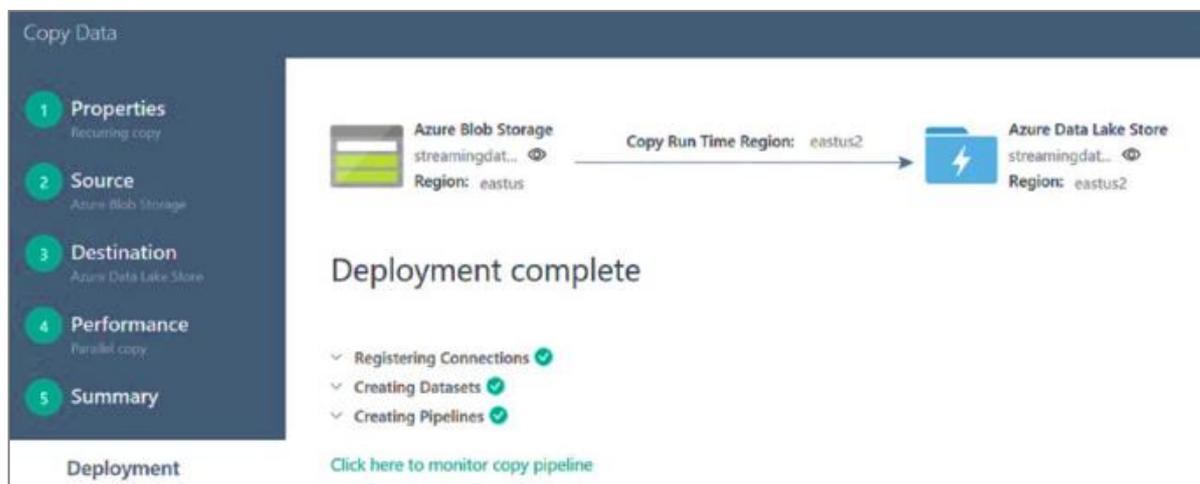
Gambar 6-61. Wizard Data Salin Pabrik Data: tentukan nilai format file tujuan Data Lake

Layar terakhir di Copy Wizard adalah layar ringkasan, seperti yang ditunjukkan pada Gambar 6-62.



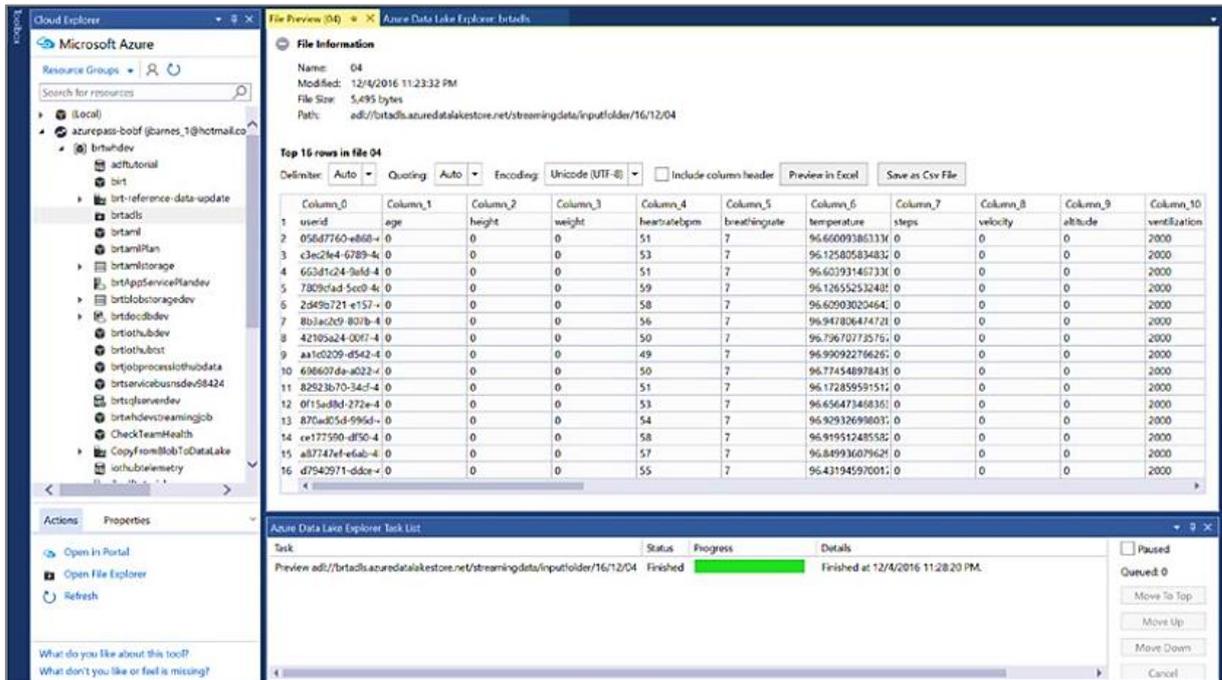
Gambar 6.62. Layar ringkasan Wizard Penyalinan Data Pabrik Data

Pertama, klik tombol Otorisasi, seperti yang ditunjukkan pada Gambar 6-62 sehingga Anda dapat masuk lagi ke langganan Azure dan pekerjaan penyalinan dapat mengambil kredensial. Terakhir, klik tombol Selesai untuk membuat pekerjaan alur Data Factory yang baru. Setelah jangka waktu singkat, pekerjaan penyalinan Data Factory baru akan didaftarkan dan disebar, seperti yang ditunjukkan pada Gambar 6-63.



Gambar 6.63. Penerapan selesai

Setelah pekerjaan alur baru Anda disebar, pekerjaan tersebut akan berjalan dan Anda dapat memeriksa output di Azure Data Lake Store tujuan Anda. Gambar 6-64 menggambarkan contoh hasil salinan Data Lake Store yang diperbarui menggunakan Visual Studio Cloud Explorer untuk melihat output tujuan di folder Data Lake Store.



Gambar 6.64 Visual Studio Cloud Explorer: lihat Penyimpanan Data Lake

Pada titik ini, Anda kini telah menyelesaikan skenario kasus penggunaan Data Factory ketiga dan terakhir untuk implementasi referensi.

6.7 RINGKASAN

Bab ini memberikan gambaran umum tingkat tinggi tentang Azure Data Factory dan Azure Data Lake Store. Anda menerapkan tiga tugas Data Factory untuk menyelesaikan skenario kasus penggunaan yang sesuai untuk implementasi referensi:

- Perbarui data referensi
- Melatih kembali model Azure Machine Learning
- Memindahkan data dari penyimpanan blob Azure ke Azure Data Lake Store

Sekarang akan terlihat jelas bahwa Azure Data Factory adalah alat utama untuk mencapai apa yang dikenal di bidang Intelijen Bisnis sebagai operasi ETL (Ekstrak, Transformasi, dan Muat) di Azure.

Anda juga melihat bagaimana pekerjaan Data Factory dapat diedit menggunakan JSON murni untuk mendapatkan kontrol yang baik atas aspek eksekusi pekerjaan Data Factory. Anda menggunakan parameter di Pabrik Data untuk membuat pekerjaan terjadwal guna mengotomatiskan operasi perpindahan data secara berulang setiap minggu. Data Lake Store adalah penyimpanan data yang kuat dan hampir tidak terbatas yang akan kita jelajahi lebih lengkap di bab berikutnya saat kita memeriksa analitik Data Lake.

BAB 7

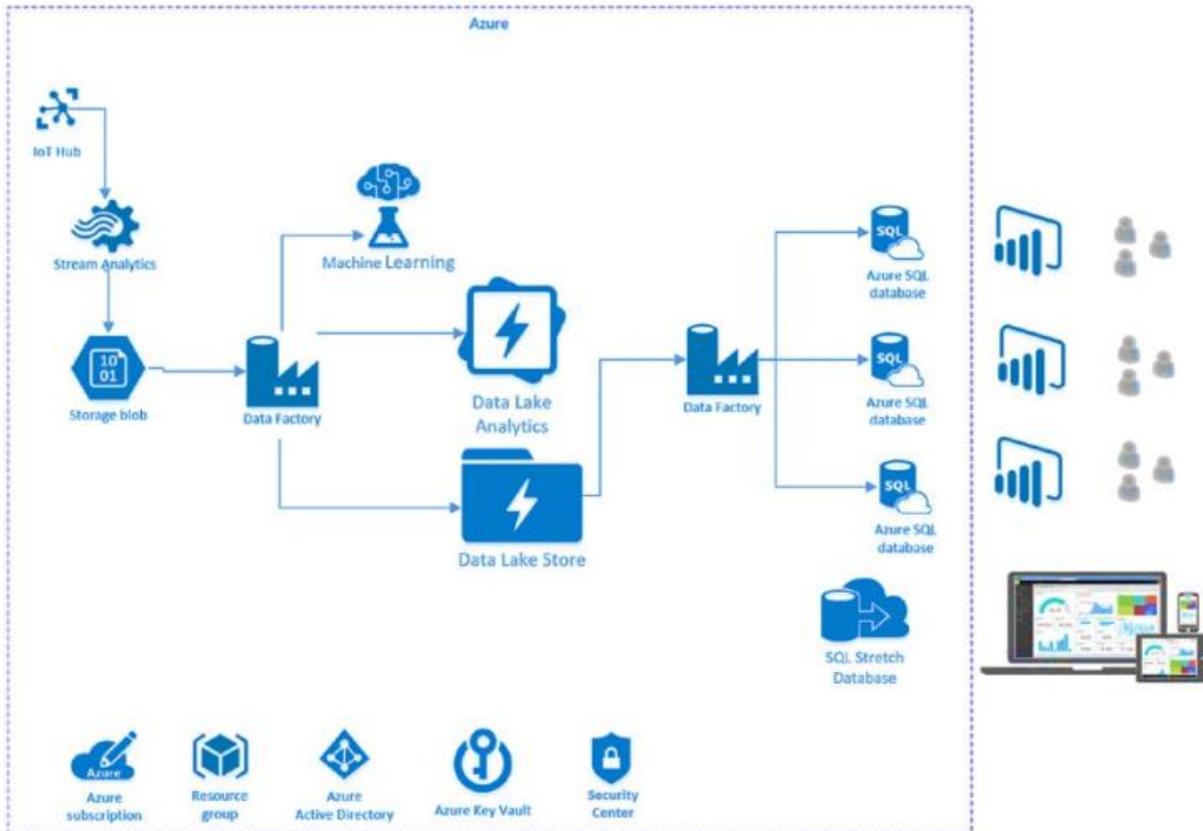
ANALISIS TINGKAT LANJUT DENGAN AZURE DATA LAKE ANALYTICS

Bab ini membahas penggunaan *Azure Data Lake Analytics* (ADLA), yang merupakan perangkat “Big Data” baru Microsoft yang berjalan di atas Azure Data Lake. Alat dan kemampuan ADLA membantu menyelesaikan masalah analisis dan pelaporan bisnis modern saat ini dengan lebih mudah dan efisien dibandingkan dengan solusi tradisional di lokasi. Ini lebih efisien karena menawarkan penyimpanan yang hampir tidak terbatas, dengan akses langsung ke penyimpanan tersebut untuk menjalankan operasi analitis langsung di atasnya. Tidak diperlukan penyediaan atau akuisisi tambahan; sumber daya segera tersedia sesuai permintaan.

Data Lake menawarkan kemampuan untuk mempertahankan data mentah dalam bentuk aslinya dan kemudian menjalankan tugas transformasional dan analitis untuk membuat analisis, ringkasan, dan prediksi baru—di seluruh data terstruktur dan tidak terstruktur semuanya berdasarkan data asli. Semua ini menghasilkan “waktu untuk mendapatkan nilai yang lebih cepat” bagi bisnis modern yang ingin memaksimalkan potensi sebenarnya.

Keuntungan utamanya adalah Anda tidak perlu melakukan operasi ETL (Extract-Transform-Load) apa pun pada data di Azure Data Lake untuk menjalankan operasi analitis. Hal ini memberikan keuntungan besar ketika menangani data dalam jumlah besar, terutama ketika persyaratan analisis historis dan regresi ikut berperan.

Layanan ADLA dapat menangani pekerjaan dalam skala apa pun secara instan; Anda cukup menentukan berapa banyak daya komputasi yang Anda perlukan saat mengirimkan tugas Anda. Anda dapat menentukan prioritas pekerjaan dan jumlah Unit Analitik (AU) untuk pekerjaan Anda. AU memungkinkan Anda menentukan berapa banyak sumber daya komputasi yang dapat digunakan oleh pekerjaan Anda. Satu AU kira-kira setara dengan dua inti CPU dan RAM 6 GB. Anda hanya membayar pekerjaan Anda saat pekerjaan tersebut sedang berjalan, sehingga lebih hemat biaya dibandingkan dengan pendekatan tradisional di lokasi, yang mengharuskan Anda membayar infrastruktur baik infrastruktur tersebut digunakan atau tidak. Dalam bab ini, kita mulai dengan memberikan pandangan sudut lebar tentang fitur dan kemampuan ADLA. Kami kemudian menerapkan pengetahuan ini pada implementasi referensi dan menyatukan semua bagian saat kami membangun solusi kerja yang lengkap. Gambar 7.1 mengilustrasikan peran ADLA dalam implementasi referensi.



Gambar 7.1 Azure Data Lake Analytics dalam arsitektur implementasi referensi

Kami mengambil pelajaran penting dari Bab 5 (Analisis Aliran) dan Bab 6 (Pabrik Data) untuk melanjutkan aktivitas pergerakan data di bab ini. Secara khusus, kami memproses data yang dipindahkan ke Azure Data Lake dari penyimpanan Azure Blob (melalui Data Factory); data tersebut mewakili data streaming IoT yang masuk. Kami kemudian menggabungkan data IoT mentah ini dengan hasil panggilan Layanan Web Azure Machine Learning yang dibuat "waktu nyata" selama pekerjaan penyerapan Stream Analytics dan kemudian menyimpan data tersebut ke penyimpanan Azure. Mari kita mulai dengan gambaran teknis singkat tentang kemampuan dan fitur Azure ADLA.

7.1 ANALISIS AZURE DATA LAKE

ADLA adalah layanan analitik terdistribusi Microsoft Azure baru yang dibangun di atas Apache YARN. YARN adalah singkatan dari "Yet Another Resource Negotiator" dan merupakan teknologi manajemen cluster untuk Apache Hadoop. ADLA dibangun dengan tujuan utama membuat analisis Big Data menjadi mudah dan efisien. ADLA memungkinkan Anda fokus pada menulis, menjalankan, dan mengelola pekerjaan analitis, daripada mengoperasikan infrastruktur komputasi terdistribusi. Beberapa kemampuan inti layanan ADLA antara lain sebagai berikut:

- **Penskalaan Dinamis:** ADLA dirancang pada intinya untuk skala dan kinerja cloud. ADLA dapat menyediakan sumber daya secara dinamis dan memungkinkan Anda melakukan analisis pada kumpulan data yang sangat besar, seperti data berukuran terabyte atau

bahkan exabyte. Ketika suatu pekerjaan selesai, sumber daya akan berkurang secara otomatis, dan Anda hanya membayar untuk daya pemrosesan yang digunakan untuk menjalankan pekerjaan Anda. Saat Anda menambah atau mengurangi ukuran data yang disimpan atau jumlah komputasi yang digunakan, Anda tidak perlu menulis ulang kode apa pun. Hal ini memungkinkan Anda fokus pada logika bisnis saja dan bukan pada cara Anda memproses dan menyimpan kumpulan data besar.

- *U-SQL*: ADLA menyertakan U-SQL, bahasa kueri yang menggabungkan bahasa deklaratif mirip SQL dengan ekstensibilitas dan kemampuan program yang disediakan oleh C#, untuk membuat prosesor dan reduksi khusus. U-SQL juga menyediakan kemampuan untuk mengkueri dan menggabungkan data dari berbagai sumber data, termasuk Azure Data Lake Storage, Azure Blob Storage, Azure SQL DB, Azure SQL Data Warehouse, dan instans SQL Server yang berjalan di Azure Virtual Machines.
- *Kembangkan, Debug, dan Optimalkan Lebih Cepat Menggunakan Visual Studio*: ADLA memiliki integrasi mendalam dengan Visual Studio, sehingga Anda dapat menggunakan alat yang sudah dikenal untuk menjalankan, men-debug, dan menyempurnakan kode pekerjaan analitik Anda. Selain itu, ADLA menyediakan visualisasi praktis dari pekerjaan U-SQL Anda. Visualisasi ini memungkinkan Anda memeriksa bagaimana kode Anda berjalan dalam skala besar, sehingga memudahkan Anda mengidentifikasi hambatan performa sejak dini sehingga mengoptimalkan performa (dan biaya).
- *Analisis Big Data untuk Banyak Orang*: ADLA menyediakan alat dan kerangka kerja sehingga pengembang baru pun dapat dengan mudah mengembangkan dan menjalankan program transformasi dan pemrosesan data paralel secara besar-besaran dalam U-SQL, R, Python, dan .Net melalui data berukuran petabyte. Semua data Anda dapat dianalisis dengan ADLA dan U-SQL termasuk data tidak terstruktur, semi terstruktur, dan terstruktur.
- *Integrasi dengan Investasi TI yang Ada*: ADLA dapat menggunakan investasi TI Anda yang ada untuk identitas, manajemen, keamanan, dan pergudangan data. ADLA terintegrasi dengan Azure Active Directory untuk manajemen dan izin pengguna. Itu juga dilengkapi dengan kemampuan pemantauan dan audit bawaan.
- *Hemat Biaya*: ADLA menjadi solusi yang sangat hemat biaya untuk menjalankan beban kerja Big Data jika Anda melihat detail penetapan harga dan skalanya. Dengan ADLA, Anda membayar berdasarkan “per pekerjaan” hanya ketika data Anda diproses. Sistem secara otomatis menaikkan atau menurunkan skala saat pekerjaan dimulai dan selesai, sehingga Anda tidak perlu membayar lebih dari yang Anda perlukan. Tidak diperlukan perangkat keras, lisensi, atau perjanjian dukungan khusus layanan tambahan.
- *Dioptimalkan untuk Data Lake*: Perlu diperhatikan bahwa ADLA dirancang dan dioptimalkan secara khusus untuk bekerja sama dengan Azure Data Lake Store guna memberikan tingkat performa, throughput, dan paralelisasi tertinggi untuk beban kerja Big Data Anda yang paling menuntut.
- *Manajemen dan Administrasi yang Disederhanakan*: ADLA dapat dikelola dengan mudah melalui Portal Azure. Selain itu, PowerShell dapat digunakan untuk

mengotomatiskan pekerjaan analitik dan melakukan tugas ADLA terkait. Bilah Portal Azure untuk ADLA juga menawarkan kemampuan untuk mengamankan lingkungan analitik Anda dengan alat Kontrol Akses Berbasis Peran (RBAC) yang terintegrasi dengan Azure Active Directory. Kemampuan pemantauan dan pemberitahuan juga dibangun ke dalam Portal Azure untuk memenuhi persyaratan operasi dan administrasi.

Memulai Azure Data Lake Analytics

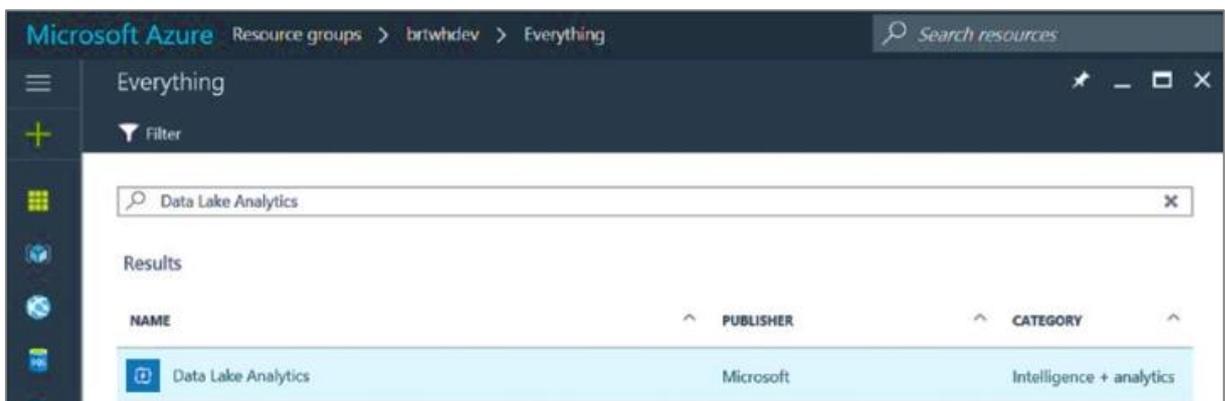
Sangat mudah untuk memulai dengan ADLA. Ada proses tiga langkah untuk memulai dan menjalankannya:

1. Buat akun ADLA di grup Langganan/Sumber Daya Azure Anda. Ini adalah penyiapan satu kali dan memungkinkan Anda mulai menjelajahi data serta menjalankan tugas analitik pada data tersebut.
2. Tulis dan kirimkan pekerjaan ADLA dengan U-SQL. Anda dapat membuat, mengirimkan, dan memantau pekerjaan dari berbagai sumber, seperti perintah Azure Portal, Visual Studio, dan PowerShell.
3. Periksa hasil pekerjaan. Pada intinya, pekerjaan ADLA pada dasarnya membaca dan menulis data dari penyimpanan dengan cara yang sangat terdistribusi dan paralel secara masif. Untuk memudahkan akses, sumber penyimpanan dapat berasal dari berbagai lokasi seperti Data Lake Store, penyimpanan blob Azure, atau data dari server SQL lain di platform atau layanan lain. Dan semua data dapat dianalisis “di tempat” tanpa penundaan apa pun untuk ekstraksi, persiapan, atau pemuatan.

Selanjutnya, kami akan menjalani langkah-langkah ini secara lebih rinci sehingga Anda dapat melihat betapa mudahnya memulai pemrosesan Big Data menggunakan ADLA.

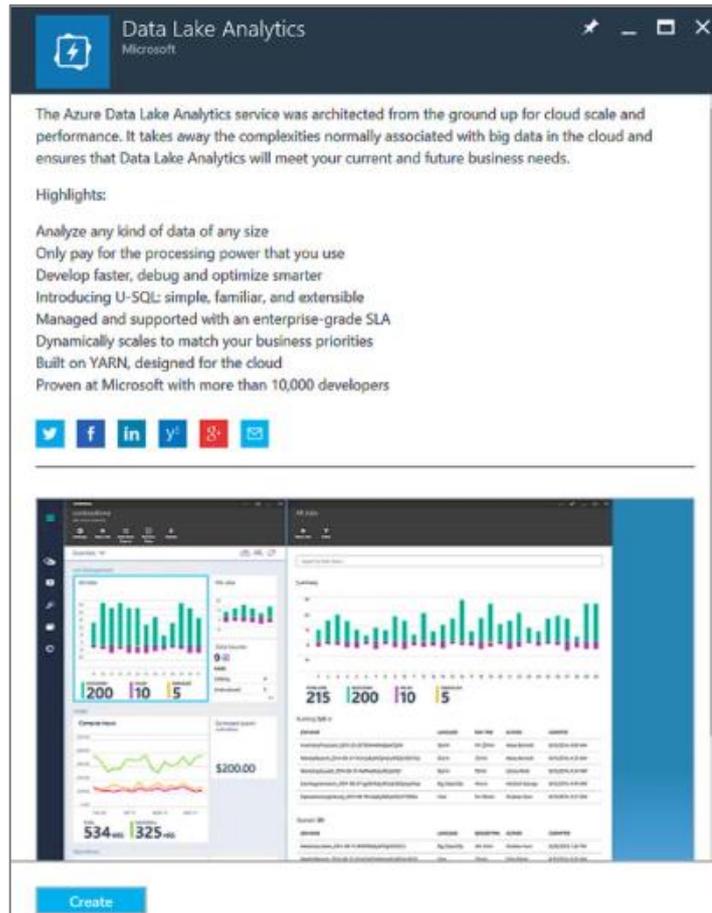
Buat Akun ADLA

Untuk memulai, di Portal Azure, navigasikan ke grup sumber daya untuk penerapan Anda. Klik tombol + Tambahkan dan cari Data Lake Analytics, seperti yang ditunjukkan pada Gambar 7.2.



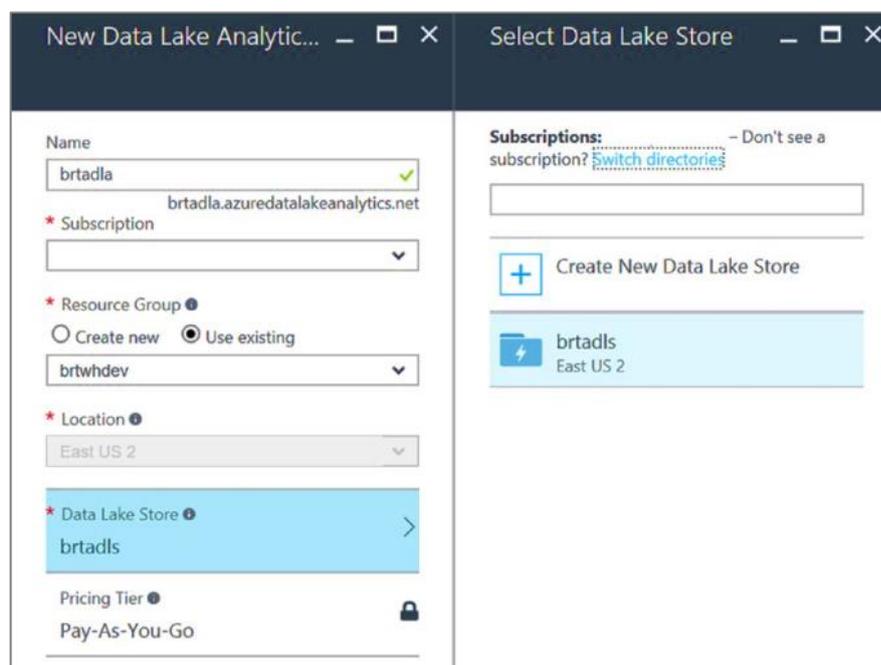
Gambar 7.2. Menambahkan Data Lake Analytics ke grup sumber daya

Setelah memilih opsi untuk Data Lake Analytics, klik pada layar berikutnya untuk membuat akun Data Lake Analytics baru, seperti yang ditunjukkan pada Gambar 7.3.



Gambar 7.3 Membuat akun Azure Data Lake Analytics baru

Layar berikutnya memungkinkan Anda memasukkan parameter spesifik untuk membuat akun ADLA baru, seperti yang ditunjukkan pada Gambar 7.4.

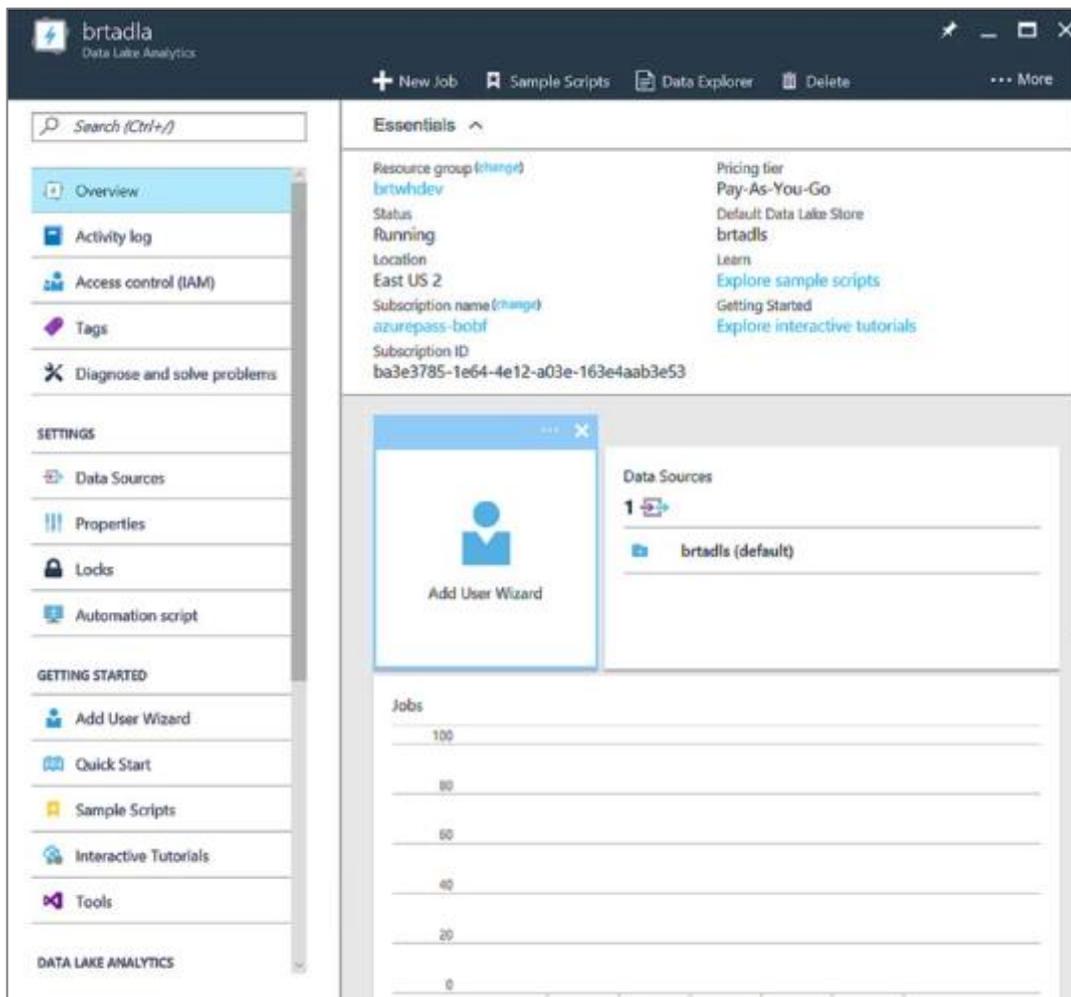


Gambar 7.4 Membuat akun Azure Data Lake Analytics baru

Isi pilihan Anda untuk nilai parameter yang sesuai:

- * Nama: Masukkan nama unik untuk akun ADLA baru Anda. Perhatikan bahwa nama ADLA harus unik secara global.
- * Berlangganan: Langganan Azure yang digunakan untuk pekerjaan ini.
- * Grup Sumber Daya: Grup Sumber Daya Azure tempat layanan ini dibuat.
- * Lokasi: Lokasi Pusat Data Azure.
- * Penyimpanan Data Lake: Penyimpanan Azure Data Lake yang akan menjadi lokasi utama untuk menganalisis data Anda. Perhatikan bahwa setiap akun ADLA memiliki akun Data Lake Store yang bergantung. Akun ADLA dan akun Data Lake Store dependen harus berlokasi di pusat data Azure yang sama.

Setelah selesai, klik tombol Buat di bagian bawah layar. Masukan Anda kemudian akan divalidasi dan akun Azure Data Lake baru akan dibuat setelah jangka waktu singkat (biasanya kurang dari satu menit). Setelah akun ADLA baru Anda diprovisikan, navigasikan ke akun tersebut melalui Portal Microsoft Azure. Layar ADLA Anda akan tampak seperti Gambar 7-5.



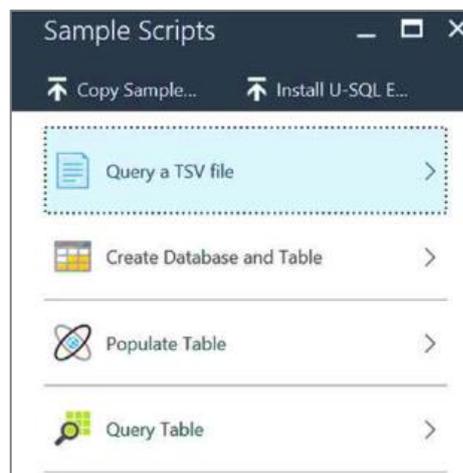
Gambar 7.5. Bilah halaman web Azure Data Lake Analytics

Saat bekerja dengan ADLA melalui Portal Microsoft Azure, perhatikan bahwa Anda terutama akan menggunakan dua opsi navigasi yang ditunjukkan di bagian atas layar pada Gambar 7-5:

- *Pekerjaan Baru*: Memungkinkan Anda membuat, mengirimkan, dan memantau pekerjaan ADLA Anda. Opsi ini memungkinkan Anda membuat, mengunggah, dan mengunduh skrip U-SQL. Jika Anda sudah siap, Anda dapat mengirimkannya untuk dijalankan dan memantau kemajuannya.
- *Data Explorer*: Memungkinkan Anda menavigasi data dengan cepat dan mudah di struktur database Data Lake Storage dan ADLA yang berisi tabel dan skema.

Contoh Skrip: Membuat, Mengirim, dan Memantau Pekerjaan

Salah satu cara terbaik untuk mengenal ADLA dengan cepat adalah dengan menjalankan contoh skrip yang Anda lihat di Portal Azure setelah Anda membuat akun analitik. Saat pertama kali Anda menyediakan Data Lake Store baru bersama dengan akun Data Lake Analytics baru, Anda akan diberikan opsi untuk menginstal data sampel. Sangat disarankan agar Anda melakukannya, terutama karena contoh skrip memiliki ketergantungan pada contoh data yang ada di sistem Anda. Setelah Anda mengklik ikon Sample Scripts di bilah navigasi atas, bilah baru akan muncul seperti Gambar 7.6.



Gambar 7.6. Contoh opsi skrip

Klik pada opsi Query a TSV File, seperti yang ditunjukkan pada Gambar 7.6, dan jendela New U-SQL Job akan muncul, seperti yang ditunjukkan pada Gambar 7.7.

```

* Job Name
Query a TSV file
Priority 1000
AUs 1

1 //Define schema of file, must map all columns
2 @searchlog =
3     EXTRACT UserId      int,
4             Start      DateTime,
5             Region     string,
6             Query       string,
7             Duration   int,
8             Urls        string,
9             ClickedUrls string
10    FROM @"/Samples/Data/SearchLog.tsv"
11    USING Extractors.Tsv();
12
13 OUTPUT @searchlog
14    TO @"/Samples/Output/SearchLog_output.tsv"
15    USING Outputters.Tsv();

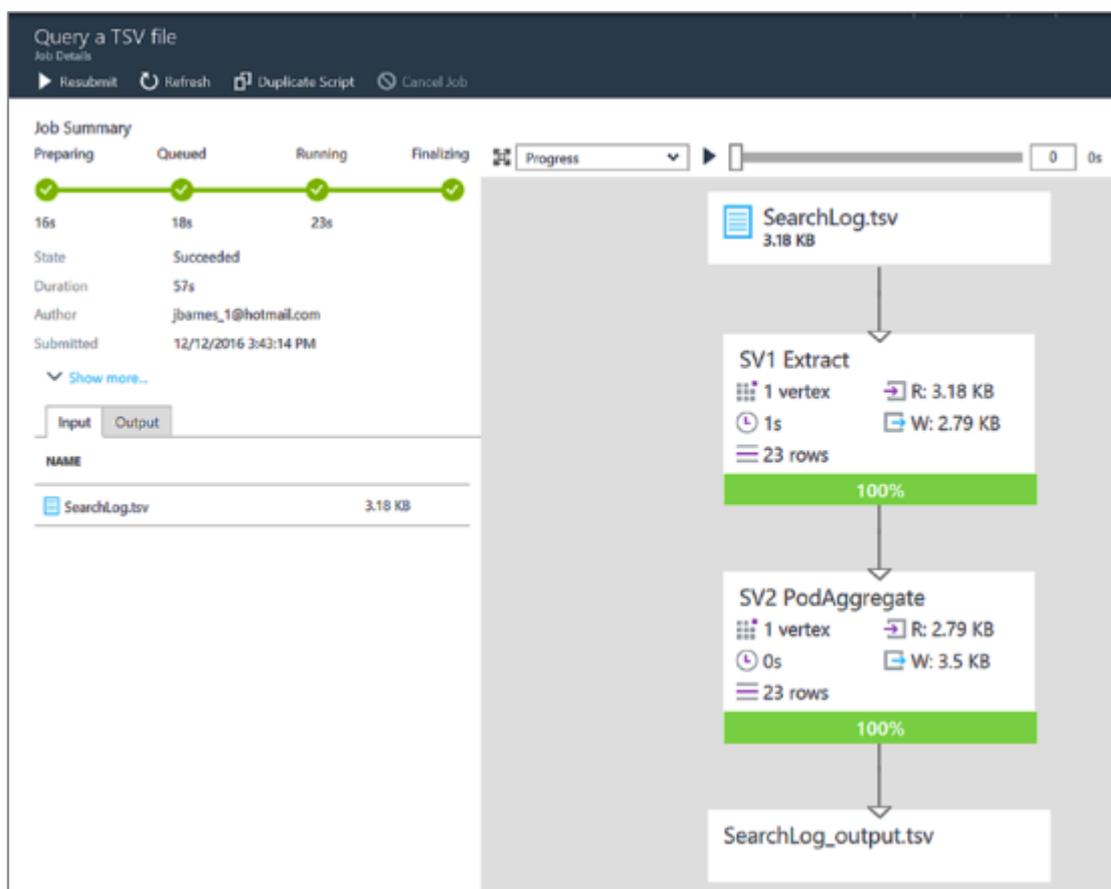
```

Gambar 7.7 Contoh skrip U-SQL untuk membaca dan menulis file TSV

Dari contoh skrip awal ini, Anda dapat mengidentifikasi beberapa aspek utama bahasa dan operasi U-SQL:

- Kueri U-SQL dapat diekspresikan dalam sintaksis SQL yang familiar.
- Ekstraktor dan keluaran adalah kunci untuk bekerja dengan data semi-terstruktur dan tidak terstruktur.
- U-SQL dapat dengan mudah mengekstrak data nilai yang dipisahkan tab dari satu file datar dan kemudian menuliskannya ke file datar lainnya.
- Nilai yang dibaca dari file datar dapat ditetapkan ke tipe variabel berbeda.

Klik ikon Kirim Pekerjaan, seperti yang ditunjukkan di sisi kiri atas Gambar 7.7. Selanjutnya, Anda akan melihat layar Ringkasan Pekerjaan seperti yang ditunjukkan pada Gambar 7.8.

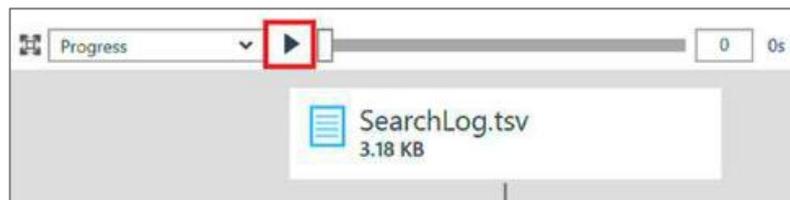


Gambar 7.8. Azure Data Lake Analytics: contoh hasil pekerjaan

Layar ini memberikan informasi saat pekerjaan melewati empat tahap pelaksanaan pekerjaan ADLA: Mempersiapkan, Mengantri, Menjalankan, dan Menyelesaikan. Perhatikan bahwa Anda dapat (dan sering kali akan) menggunakan perintah Refresh untuk menyegarkan layar secara manual guna melihat status terbaru pekerjaan Anda secara real-time di dasbor. Perhatikan bahwa Anda juga dapat menelusuri file input dan output yang terkait dengan eksekusi pekerjaan analitik, seperti yang ditunjukkan di sisi kiri Gambar 7.8.

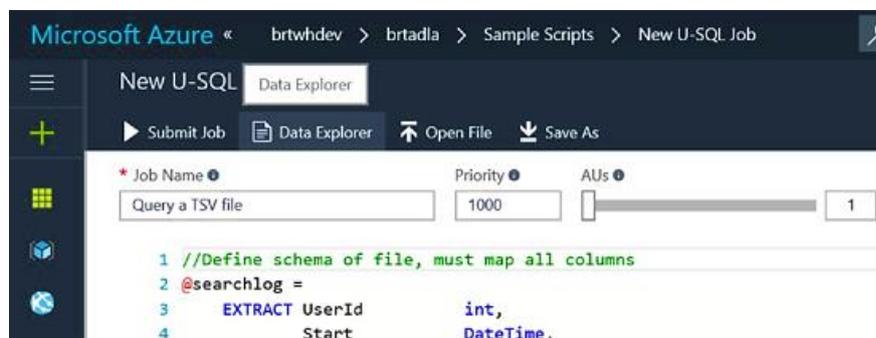
Sisi kanan layar memberikan informasi yang berkaitan dengan rincian pelaksanaan pekerjaan. Gambar 7.8 mengilustrasikan grafik yang disediakan untuk membantu

memvisualisasikan informasi pelaksanaan pekerjaan: Kemajuan, Pembacaan Data, Penulisan Data, Waktu Eksekusi, Waktu Eksekusi Rata-rata per Node, Throughput Input, dan Throughput Output. Dengan memanfaatkan informasi pelaksanaan pekerjaan yang kaya ini, Anda dapat menyesuaikan dan menyesuaikan kueri analitis Big Data untuk mengoptimalkan hasil dan biaya Anda. Perhatikan bahwa Anda juga dapat mengklik Putar Ulang (ikon panah) di bilah navigasi atas untuk memutar ulang pekerjaan Anda secara visual (secara logis tanpa benar-benar menjalankan pekerjaan) dan melihat bagaimana pekerjaan tersebut menggunakan sumber daya saat dijalankan. Lihat Gambar 7.9 untuk tombol Putar Ulang.



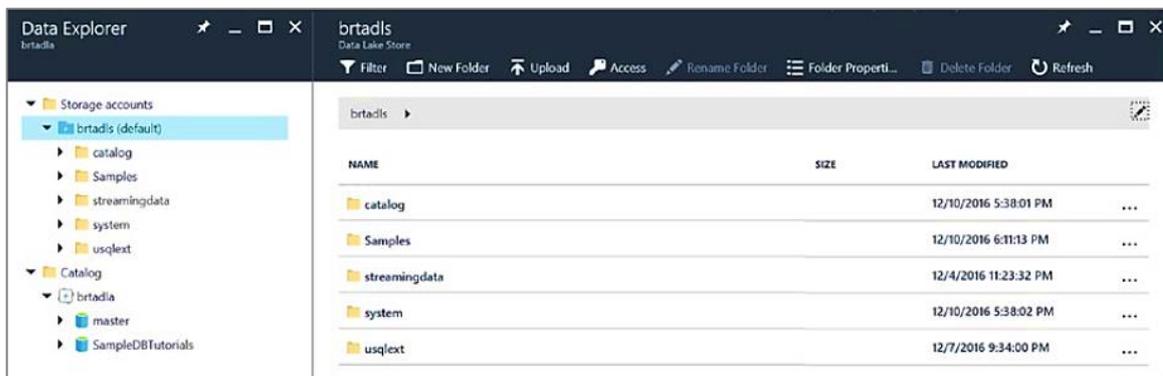
Gambar 7.9. Tombol Putar Ulang Azure Data Lake Analytics membantu Anda memvisualisasikan pekerjaan Anda setelah dijalankan

Tutup bilah ini dan Anda akan kembali ke bilah pengiriman pekerjaan, tempat Anda dapat mengklik opsi Data Explorer di bilah navigasi atas (lihat Gambar 7.10).



Gambar 7-10. Ikon Penjelajah Data

Selanjutnya, Anda akan melihat tampilan layar Data Explorer seperti yang ditunjukkan pada Gambar 7-11.



Gambar 7.11. Tampilan Penjelajah Data Data Lake Analytics

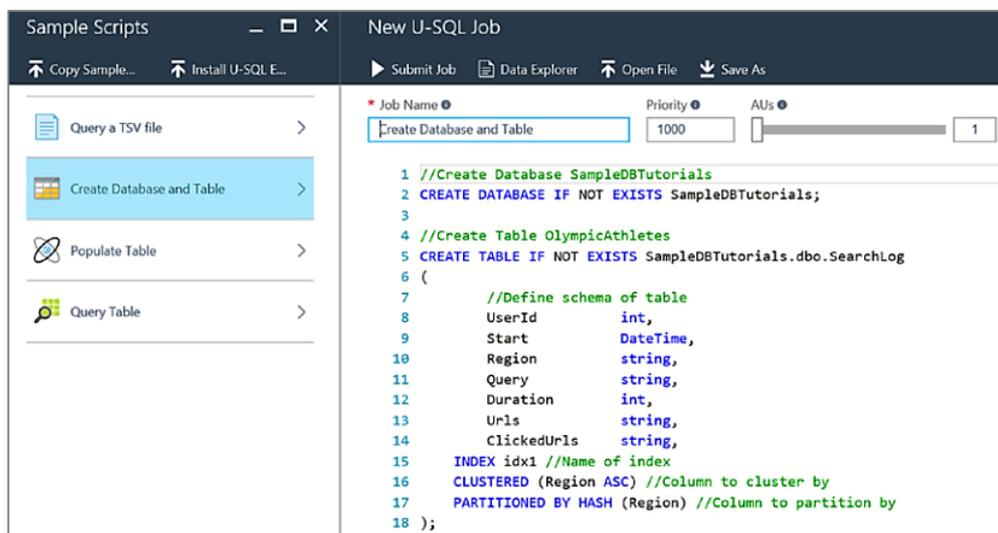
Beberapa item yang perlu diperhatikan pada Gambar 7-11:

- *Kiri atas layar:* Anda akan melihat bahwa Anda memiliki tampilan ke dalam konten Azure Data Lake Store Anda. Kontennya muncul sebagai sistem file dengan folder dan file. Ini adalah lokasi di mana Anda terutama akan bekerja dengan segala jenis data tidak terstruktur atau "file datar".
- *Kiri Bawah layar:* Termasuk "Katalog" yang mewakili tampilan "terstruktur" data Anda di ADLA, mirip dengan penyimpanan database SQL Server. Katalog ini digunakan untuk menyimpan data dan kode terstruktur sehingga keduanya dapat dibagikan oleh skrip U-SQL. Setiap katalog dapat berisi satu atau lebih database tambahan.

Basis data U-SQL berisi yang berikut ini:

- *Assemblies:* Berbagi kode .NET di antara skrip U-SQL.
- *Fungsi Tabel-Nilai:* Berbagi kode U-SQL di antara skrip U-SQL.
- *Tabel:* Berbagi data antar skrip U-SQL.
- *Skema:* Berbagi skema tabel di antara skrip U-SQL.

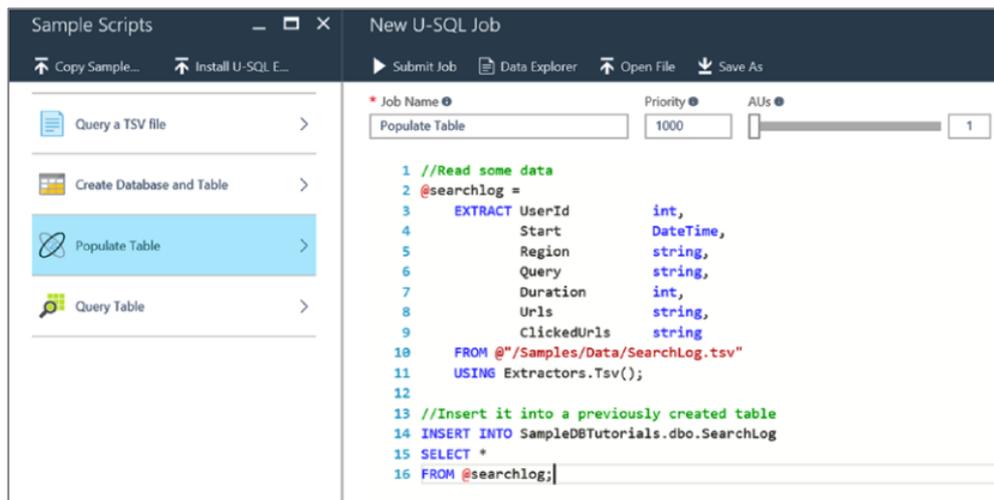
Contoh skrip berikutnya mengeksplorasi konsep ini lebih jauh, karena Anda akan membuat database dan tabel dalam skrip U-SQL. Navigasikan kembali ke bilah Skrip Sampel di Portal Azure dan pilih opsi kedua, yang disebut Buat Database dan Tabel, seperti yang ditunjukkan pada Gambar 7.12.



Gambar 7.12. Skrip sampel Data Lake membuat database dan tabel

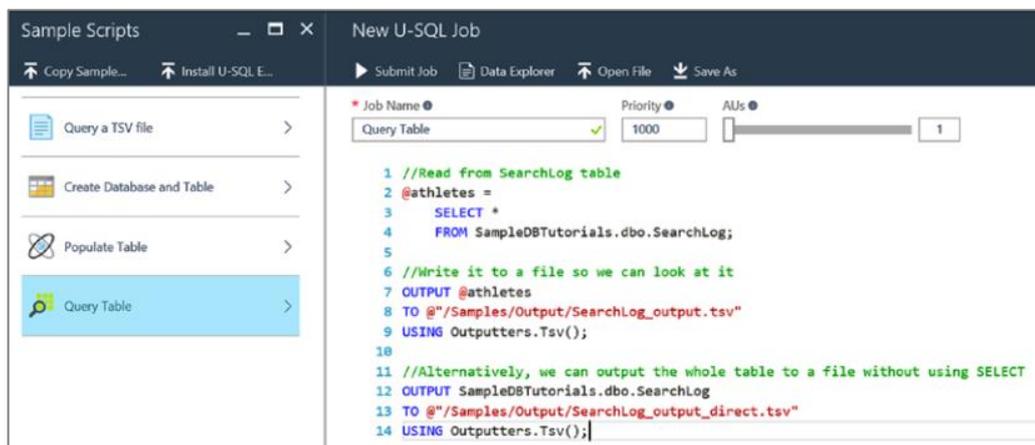
Di sini, Anda melihat pernyataan sintaksis mirip SQL yang memudahkan Anda mendeklarasikan database ADLA baru dan tabel terkait. Selain itu, kami telah mendeklarasikan indeks berkerumun yang dipartisi oleh hash di wilayah tersebut. Hal ini berpotensi memberikan peningkatan kinerja yang besar ketika menangani rekaman dalam jumlah besar. Pengamatan lain adalah bahwa lingkungan ADLA memungkinkan perpaduan optimal antara penyimpanan dan perintah SQL yang merangkum, menggabungkan, mengubah, mengekstrak, dan menghasilkan data bisnis yang bermakna dalam skala besar. Klik ikon Kirim Pekerjaan di bilah navigasi atas dan pekerjaan Anda akan ditempatkan dalam antrian untuk akhirnya

membuat database dan tabel baru. Saat ini, sudah jelas bahwa ada banyak kesamaan dalam ADLA dengan cara sistem komputer “warisan” generasi lama dirancang untuk dijalankan. Lingkungan batch selalu paling efisien dalam membagi beban kerja komputasi, menjalankan beban kerja pada node terdistribusi, memantau dan mengumpulkan hasilnya, dan pada akhirnya menghasilkan output. Bagian selanjutnya ini menunjukkan kemampuan U-SQL saat Anda mengisi database sampel menggunakan file input CSV sampel. Navigasikan kembali ke bilah Sample Scripts dan pilih opsi ketiga untuk mengisi tabel, seperti yang ditunjukkan pada Gambar 7-13.



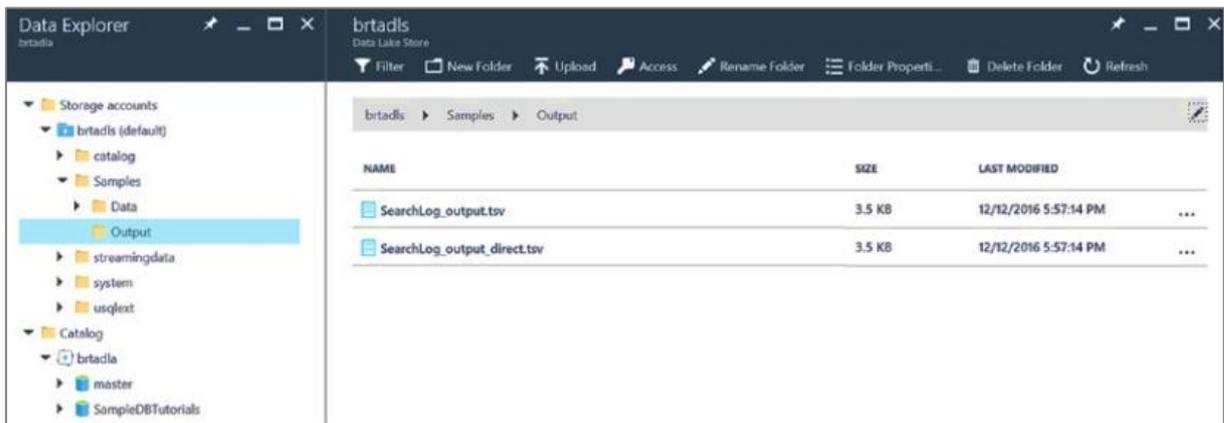
Gambar 7.13. Contoh skrip untuk mengisi tabel SQL

Skrip SQL ini akan membaca dari file .TSV (semi-terstruktur) bernama SearchLog.tsv sebagai sumber input. Kemudian akan menampilkan hasilnya ke dalam tabel SQL baru yang Anda buat dengan skrip sebelumnya. Klik Kirim Pekerjaan dan jalankan pekerjaan untuk mengisi tabel SQL baru. Langkah contoh skrip terakhir memungkinkan Anda memverifikasi hasil skrip sebelumnya dengan mengeluarkan data dari tabel SQL yang baru saja Anda isi kembali ke file .TSV. Navigasikan kembali ke bilah Sample Scripts dan pilih opsi keempat dan terakhir yang disebut Tabel Kueri, seperti yang ditunjukkan pada Gambar 7.14.



Gambar 7-14. Contoh skrip menanyakan tabel SQL dan menampilkan hasil ke file .TSV

Klik Kirim Pekerjaan untuk menjalankan pekerjaan ini, yang akan menanyakan tabel SQL baru dan kemudian menampilkan hasilnya ke file a.TSV. Seperti yang telah Anda lihat sebelumnya, setelah mengirimkan pekerjaan Anda dapat dengan mudah memantau dan melihat kemajuannya melalui tahapan pelaksanaan. Untuk memverifikasi hasil skrip Tabel Kueri dari sudut pandang Data Lake Storage, klik ikon Data Explorer di bilah navigasi atas dan telusuri folder file ke folder Samples/Output, seperti yang diilustrasikan dalam Gambar 7-15.



Gambar 7.15. Verifikasi file keluaran pekerjaan SQL-to-File menggunakan Data Explorer

Anda dapat mengklik setiap file untuk melihat kontennya dan memverifikasi bahwa konten tersebut berasal dari file yang Anda gunakan untuk mengisi tabel SQL Data Lake di Contoh Skrip #3 (Isi Tabel). Perhatikan bahwa saat ini tidak ada cara untuk melakukan kueri atau melihat hasil kueri ADLA SQL secara interaktif. Hal ini disebabkan oleh fakta bahwa ADLA terutama didasarkan pada arsitektur pemrosesan batch. Untuk memperhitungkan kenyataan ini, cara termudah untuk memverifikasi isi query SQL adalah dengan mendeklarasikan file output dan menulis ke tujuan tersebut, seperti yang ditunjukkan pada Gambar 7.14. Kemudian Anda dapat menggunakan Data Explorer untuk memverifikasi hasilnya. Dengan menjalankan contoh skrip U-SQL yang disertakan dengan ADLA, Anda telah menjelajahi beberapa kemampuan dan fitur layanan yang lebih umum digunakan. Meskipun semua yang Anda capai dalam tur singkat sampel ini dilakukan melalui Portal Azure, Microsoft juga menyediakan alat tambahan untuk bekerja dengan U-SQL di Visual Studio.

7.2 ALAT AZURE DATA LAKE UNTUK VISUAL STUDIO

Azure Data Lake Tools for Visual Studio adalah plug-in gratis yang berfungsi dengan Visual Studio (2015/2017) untuk memudahkan pembuatan, debugging, dan penyetelan skrip dan kueri ADLA U-SQL. Selain menangani semua operasi dasar yang tersedia melalui Portal Azure seperti pembuatan, pengiriman, dan pemantauan pekerjaan ADLA, ada banyak alat, utilitas, dan fungsionalitas tambahan yang dapat Anda manfaatkan dengan Azure Data Lake Tools untuk Visual Studio, seperti:

- *Pengujian Unit*: Alat Visual Studio menyertakan templat proyek baru untuk membuat skrip pengujian unit U-SQL. Ini sangat berharga karena dapat menjalankan skenario pengujian dan regresi otomatis.

- *IntelliSense*: Untuk bantuan dalam meminta entitas katalog Data Lake seperti database, skema, tabel, Objek Buatan Pengguna (UDO), dll. Karena Anda hanya dapat memiliki satu katalog master per akun ADLA, semua entitas terkait dengan komputasi spesifik Anda akun.
- *Pemformatan Otomatis*: Saat membuat pekerjaan ADLA, Visual Studio mempermudah visualisasi kode U-SQL Anda sehingga meningkatkan keterbacaan dan kemampuan pemeliharaan. Semua aturan pemformatan dapat dikonfigurasi di Alat ► Opsi ► Editor Teks ► SIP ► Pemformatan.
- *Definisi Masuk dan Temukan Semua Referensi*: Ini membantu Anda menentukan segmen kode dan menentukan jalur kode.
- *Peta Panas*: Alat VS Data Lake menyediakan hamparan warna yang dapat dipilih pengguna pada tampilan pekerjaan untuk menunjukkan: Kemajuan, I/O Data, Waktu eksekusi, dan throughput I/O setiap tahapan. Fitur ini memungkinkan pengguna untuk menentukan potensi masalah dan distribusi properti pekerjaan secara visual dan intuitif.
- *Jalankan U-SQL Secara Lokal*: Kemampuan ini jika digabungkan dengan Data Lake Tools untuk Visual Studio dapat memberikan peningkatan produktivitas yang besar bagi pengembang U-SQL. Alat-alat ini memungkinkan pengembang untuk memanfaatkan semua kemampuan pengembangan kaya berikut ini, semuanya secara lokal:
 - Jalankan skrip U-SQL, bersama dengan rakitan C#.
 - Skrip debug dan rakitan C#. Membuat/menghapus/melihat database lokal, rakitan, skema, dan tabel di Server Explorer, dengan cara yang persis sama seperti yang Anda lakukan untuk layanan ADLA.

Kombinasi alat dan alat Visual Studio di atas Azure Data Lake Store dan Analytics menyediakan lingkungan yang kaya dan kuat untuk mengembangkan, menjalankan, dan mengelola lingkungan analitik Big Data. Pembahasan lebih mendalam mengenai fitur dan kemampuan Data Lake Tools untuk Visual Studio berada di luar cakupan buku ini. Anda sangat disarankan untuk merujuk ke tautan dan sumber berikut untuk informasi tambahan.

Fitur dan Manfaat ADLA U-SQL

Terlepas dari apakah Anda membuat pekerjaan U-SQL menggunakan Portal Azure atau menggunakan Alat Data Lake untuk Visual Studio, ada beberapa manfaat besar dalam memanfaatkan layanan ini untuk kebutuhan pemrosesan Big Data Anda. U-SQL adalah bahasa baru dari Microsoft untuk memproses pekerjaan Big Data di Azure. U-SQL menggabungkan sintaksis SQL yang familiar dengan ekspresi kode kustom yang ditulis dalam C#, selain runtime perluasan skala yang dapat menangani hampir semua ukuran data. Beberapa dari banyak fitur dan manfaat memanfaatkan U-SQL adalah sebagai berikut:

- ❖ Menangani semua jenis data: tidak terstruktur, semi terstruktur, dan terstruktur.
- ❖ Memungkinkan Anda mendeklarasikan dan menggunakan tipe khusus domain yang ditentukan pengguna menggunakan C#.
- ❖ Anda dapat menjalankan kueri U-SQL melalui Data Lake Store dan blob Azure.

- ❖ Anda juga dapat menjalankan kueri gabungan melalui penyimpanan SQL operasional dan gudang data, sehingga mengurangi kompleksitas operasi ETL.
- ❖ Memungkinkan pengembang untuk memanfaatkan keterampilan mereka yang ada dengan SQL dan .NET. Pengembang U-SQL produktif sejak hari pertama.
- ❖ Mudah untuk diukur dan disesuaikan kinerjanya tanpa perlu mengkonfigurasi lingkungan secara manual.
- ❖ Konektor internal yang mudah digunakan untuk format data umum.
- ❖ Model ekstensibilitas yang sederhana dan kaya untuk menambahkan transformasi data khusus pelanggan.
- ❖ Tidak ada batasan skala. Menskalakan sesuai permintaan tanpa perlu mengubah kode.
- ❖ Secara otomatis memparalelkan kueri U-SQL dan kode khusus.
- ❖ Dirancang untuk memproses data berukuran petabyte.
- ❖ Fungsi agregasi bawaan yang dapat diperluas dengan fungsi agregasi C# khusus.
- ❖ Menggunakan ekstraktor bawaan untuk membaca file CSV dan TSV atau membuat ekstraktor khusus untuk berbagai format file data.
- ❖ Alat tingkat perusahaan dan lingkungan eksekusi.
- ❖ Termasuk alat untuk mengelola, mengamankan, berbagi, dan menemukan data dan objek kode yang sudah dikenal (tabel, fungsi, dll.)
- ❖ Memberikan otorisasi katalog dan akun penyimpanan berbasis peran menggunakan keamanan Azure Active Directory (AAD).
- ❖ Menyediakan audit untuk objek katalog seperti database, tabel, dll.
- ❖ Contoh perintah DDL Tabel SQL (Bahasa Definisi Data):
 - BUAT TABEL
 - BUAT INDEKS CLUSTERED
 - BUAT TABEL dengan INDEKS CLUSTERED
 - MENYISIPKAN
 - POTONG
 - MENJATUHKAN
 - Tabel didaftarkan dalam katalog metadata dan dapat ditemukan oleh orang lain melalui API katalog/metadata.

Jenis Operator Buatan Pengguna U-SQL

Bahasa U-SQL memungkinkan Operator Buatan Pengguna (UDO) berikut ini dapat diperluas:

- *Ekstraktor* (disebut dengan sintaks Ekstrak)
- *Prosesor* (disebut dengan sintaks Proses)
- *Applier* (disebut dengan sintaks Apply)
- *Penggabung* (disebut dengan sintaks Gabungan)
- *Reducer* (disebut dengan sintaks Reduce)
- *Outputter* (disebut dengan sintaks Output)

Lihat tautan berikut untuk informasi lebih lengkap:

Panduan programabilitas U-SQL: <https://docs.microsoft.com/en-us/azure/data-lake-analytics/data-lake-analytics-u-sql-programmability-guide#use-user-definisi-extractors>.

Saat tulisan ini dibuat, ADLA hadir dengan tiga ekstraktor:

- Teks yang dipisahkan koma-Nilai (CSV).
- Teks yang dibatasi Tab-Separated-Value (TSV).
- Ekstraktor serba guna untuk teks yang dibatasi

Fungsi Jendela U-SQL

U-SQL berisi konstruksi kuat yang disebut “fungsi windowing,” yang didefinisikan dengan penggunaan OVER klausa dan mewakili nilai-nilai yang dihitung dari beberapa baris, bukan hanya baris saat ini. U-SQL mengadopsi subset dari fungsi Jendela SQL Standar ANSI yang diperkenalkan ke dalam bahasa ini pada tahun 2003.

Fungsi jendela dikategorikan ke dalam area umum berikut:

- Fungsi Agregasi Pelaporan: Mencakup SUM dan AVG.
- Fungsi Pemeringkatan: Termasuk DENSE_RANK, ROW_NUMBER, NTILE, dan RANK.
- Fungsi Analitik: Menyertakan distribusi kumulatif atau persentil dan mengakses data dari baris sebelumnya (dalam kumpulan hasil yang sama) tanpa menggunakan gabungan mandiri.

Seperti yang Anda pelajari di Bab 5, jenis agregat ini sangat penting dalam analisis streaming data real-time selama periode waktu tertentu. Selain manfaat nyata dari pengumpulan agregat secara real-time melalui aliran data yang bergerak cepat, terdapat juga manfaat dalam melakukan jenis analisis agregat yang sama terhadap kumpulan data historis untuk analisis “bagaimana-jika” dan skenario pengujian regresi.

```
@employees =
SELECT * FROM ( VALUES
    (1, "Nick", "IT Department", 100, 10000),
    (2, "Sheryl", "IT Department", 100, 20000),
    (3, "Jim", "IT Department", 100, 30000),
    (4, "Claire", "Human Resources", 200, 10000),
    (5, "Jason", "Human Resources", 200, 10000),
    (6, "Becky", "Human Resources", 200, 10000),
    (7, "Scott", "Exec Management", 300, 50000),
    (8, "Anna", "Sales & Marketing", 400, 15000),
    (9, "Eric", "Sales & Marketing", 400, 10000) )
AS T(EmpID, EmpName, DeptName, DeptID, Salary);

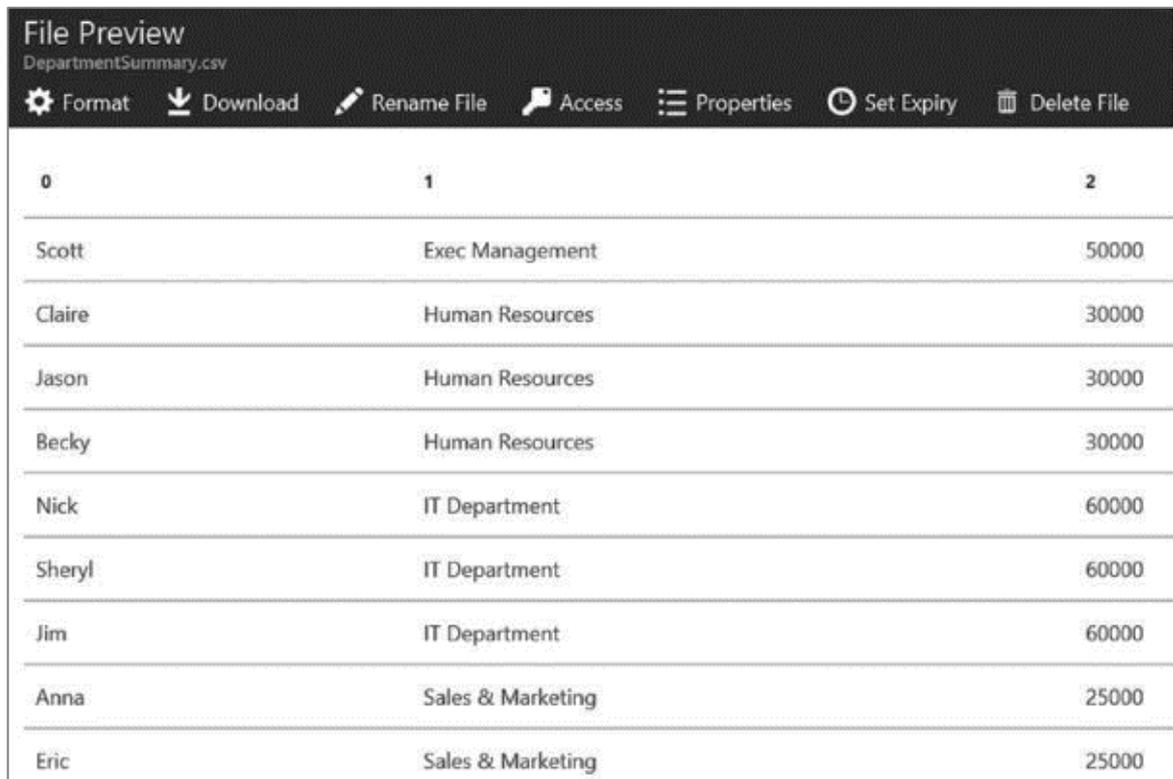
@result=
SELECT
    EmpName, DeptName,
    SUM(Salary) OVER( PARTITION BY DeptName ) AS SalaryByDept
FROM @employees;

OUTPUT @result TO "/DepartmentSummary.csv"
USING Outputters.Csv();
```

Gambar 7-16. Contoh skrip U-SQL yang menunjukkan kemampuan windowing ADLA menggunakan OVER dan PARTITION BY

Gambar 7-16 menyajikan contoh skrip U-SQL yang mendemonstrasikan penggunaan kata kunci OVER dengan klausa PARTITION BY untuk menyempurnakan “jendela” agar mencantumkan semua karyawan, departemen, dan total gaji untuk departemen tersebut.

Perhatikan bahwa klausa PARTITION BY ditambahkan ke klausa OVER untuk menciptakan efek ringkasan ini.



0	1	2
Scott	Exec Management	50000
Claire	Human Resources	30000
Jason	Human Resources	30000
Becky	Human Resources	30000
Nick	IT Department	60000
Sheryl	IT Department	60000
Jim	IT Department	60000
Anna	Sales & Marketing	25000
Eric	Sales & Marketing	25000

Gambar 7.17 Hasil contoh skrip U-SQL yang menunjukkan kemampuan windowing

Perhatikan bahwa pada Gambar 7.16 contoh aliran data masukan untuk karyawan dibuat seluruhnya dalam kode, bukan dibaca dari file atau tabel SQL. Karena kami bekerja dengan layanan batch, kami masih perlu menampilkan hasil kami ke file CSV (DepartSummary.csv) menggunakan opsi format keluaran CSV bawaan. Gambar 7.17 menggambarkan keluaran tersebut. Setiap baris hasil keluaran juga berisi total gaji untuk setiap departemen (jumlah gabungan seluruh gaji di departemen tersebut) dan dipecah berdasarkan departemen.

Fungsi Agregasi Pelaporan

Fungsi jendela ADLA mendukung agregat berikut sebagai bagian dari bahasa U-SQL:

- COUNT
- SUM
- MIN
- MAX
- AVG
- STDEV
- VAR

Fungsi Pemeringkatan

Fungsi jendela ADLA mendukung fungsi peringkat berikut sebagai bagian dari bahasa U-SQL:

- RANK

- DENSE_RANK
- NTILE
- ROW_NUMBER

Fungsi Analitik

Fungsi jendela ADLA mendukung fungsi analitik berikut sebagai bagian dari bahasa U-SQL:

- CUME_DIST
- PERCENT_RANK
- PERCENTILE_CONT
- PERCENTILE_DISC

7.3 FITUR ADLA, MENGKUERI DATA DI PENYIMPANAN

Salah satu fitur paling canggih dari ADLA adalah memungkinkan Anda dengan mudah mengkueri data yang berada di beberapa penyimpanan data Azure, dengan manfaat tambahan karena tidak perlu memindahkan data terlebih dahulu ke dalam satu penyimpanan data sebelum menjalankan kueri.

Ada banyak manfaat tambahan dari pendekatan menanyakan data dari “tempat tinggalnya”; berikut ini beberapa diantaranya:

- Menghindari perpindahan data dalam jumlah besar melintasi jaringan antar penyimpanan data. Hal ini dapat mengakibatkan berkurangnya masalah bandwidth dan latensi secara drastis pada jaringan.
- Menyediakan tampilan data tunggal tanpa memperhatikan lokasi fisik penyimpanan data jarak jauh. Referensi atau data master kini dapat hidup di “habitat aslinya” di mana ia biasanya berada.
- Meminimalkan masalah proliferasi data yang disebabkan oleh pemeliharaan banyak salinan. Ini adalah solusi IT kemarin untuk menangani masalah master data. Hal ini biasanya dicapai dengan mencoba memusatkan penyimpanan data, dan itu biasanya berarti membuat banyak salinan data.
- Menggunakan bahasa kueri tunggal (U-SQL) untuk semua kueri data. Selain menjadi bahasa pemrograman yang “menyatukan” dan “tunggal”, bahasa ini juga paling dikenal oleh para pengembang dan oleh karena itu paling cepat diadopsi dan dimanfaatkan berdasarkan keahlian yang ada. Hal ini dapat memberikan keuntungan besar dalam hal produktivitas pengembang, pemeliharaan kode, dan ketangkasan.
- Setiap penyimpanan data dapat mempertahankan kedaulatannya sendiri. Karena data dapat berada di lokasi aslinya, maka dapat dikueri “di tempat” dan hanya dengan subset (hasil kueri) dikembalikan melalui jaringan ke hasil kueri ADLA. Hal ini memungkinkan data untuk selalu berada di lokasi asalnya dan dengan demikian mematuhi semua aturan yang mengatur domain dan pergerakan data.

Ada satu hal penting yang perlu diperhatikan dalam penerapan fitur ini. Saat Anda menentukan kueri U-SQL untuk bekerja dengan data dari sumber eksternal lainnya di Azure, seperti database Azure SQL atau gudang data SQL, Anda perlu menentukan informasi berikut sebagai bagian dari pernyataan kueri:

- *SUMBER DATA*: Mewakili sumber data jarak jauh seperti Azure SQL Database. Mengharuskan Anda menentukan semua detail (string koneksi, kredensial, dll.) yang diperlukan untuk menghubungkan dan mengeluarkan pernyataan SQL.
- *TABEL EKSTERNAL*: Tabel ADLA SQL lokal, dengan kolom yang didefinisikan sebagai tipe data C#, yang mengalihkan kueri yang dikeluarkan terhadap tabel tersebut ke tabel jarak jauh yang menjadi dasarnya. U-SQL menangani konversi tipe data secara otomatis.
 - Implementasi tabel eksternal ADLA memungkinkan Anda menerapkan skema tertentu terhadap data jarak jauh, yang akan membantu melindungi Anda dari perubahan skema jarak jauh. Kemampuan ini memungkinkan Anda mengeluarkan kueri yang menggabungkan tabel eksternal dengan tabel lokal, memungkinkan beberapa skenario pemrosesan Big Data yang kreatif.
- *Kueri PASS THROUGH*: Kueri U-SQL dikeluarkan langsung terhadap sumber data jarak jauh dalam sintaksis sumber data jarak jauh. Contohnya adalah Transact SQL akan dikeluarkan terhadap sumber data jarak jauh untuk kueri database Azure SQL.
- *REMOPTABLE_TYPES*: Untuk setiap sumber data eksternal, Anda harus menentukan daftar “tipe jarak jauh”. Daftar ini membatasi jenis kueri yang akan dihapus. Contoh: `REMOPTABLE_TYPES = (bool, byte, short, ushort, int, desimal)`.
- *LAZY METADATA LOADING*: Di sini data jarak jauh dibuat skema hanya ketika kueri benar-benar dikirimkan ke sumber data jarak jauh. Program Anda harus mampu menangani perubahan skema jarak jauh.

Persyaratan implementasi ini menghasilkan beberapa pembelajaran penting yang layak untuk dikaji secara teknis lebih mendalam. Hal ini diperlukan untuk mengungkap sepenuhnya kepada pembaca bagaimana fitur ini diterapkan di Azure Data Lake Analytics.

Kueri Gabungan: Ikhtisar Langkah-Langkah yang Diperlukan untuk Kueri Tabel Eksternal

Penting untuk dicatat bahwa ada beberapa langkah yang diperlukan agar berhasil mengimplementasikan kueri Federasi ADLA. Kami menjalani setiap langkah yang diperlukan dimulai dengan prasyarat.

Konfirmasikan Kueri Federasi U-SQL: Prasyarat Dipasang dan Dikonfigurasi. Perlu diperhatikan bahwa sebagian besar prasyarat diperlukan dan akan menyebabkan masalah konektivitas dan akses signifikan yang mungkin sulit didiagnosis.

- Langganan Azure
- Akun Azure Data Lake Store (ADLS).
- Akun Azure Data Lake Analytics (ADLA).
- Azure SQL Database atau gudang Data Azure SQL dengan login SQL/kredensial kata sandi
- Visual Studio 2015 (Opsional). Untuk membuat dan menjalankan kueri U-SQL. Skrip U-SQL juga dapat dikembangkan dan disebarluaskan melalui Portal Azure.
- Azure Data Lake Tools untuk Visual Studio 2015. (Opsional) berfungsi dengan Visual Studio 2015.

Unduh: <https://www.microsoft.com/en-us/download/details.aspx?id=49504>

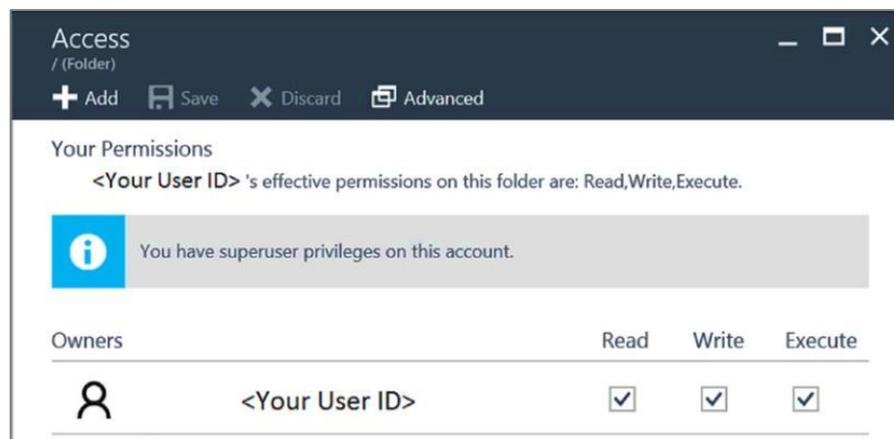
- Azure PowerShell (Opsional)
Unduh: <http://aka.ms/webpi-azps>

Verifikasi Izin “Baca/Jalankan” di Akun Azure Data Lake Store Anda

Ini diperlukan untuk membuat rahasia katalog melalui perintah PowerShell di langkah selanjutnya. Untuk memverifikasi izin Anda:

1. Navigasikan ke akun ADLS Anda di Portal Azure.
2. Klik ikon Data Explorer di bilah navigasi atas.
3. Klik ikon Akses di bilah navigasi atas.
4. Validasi Anda memiliki izin Baca/Eksekusi.

Tangkapan layar pada Gambar 7.18 mengilustrasikan izin yang tepat untuk pengguna yang berwenang.



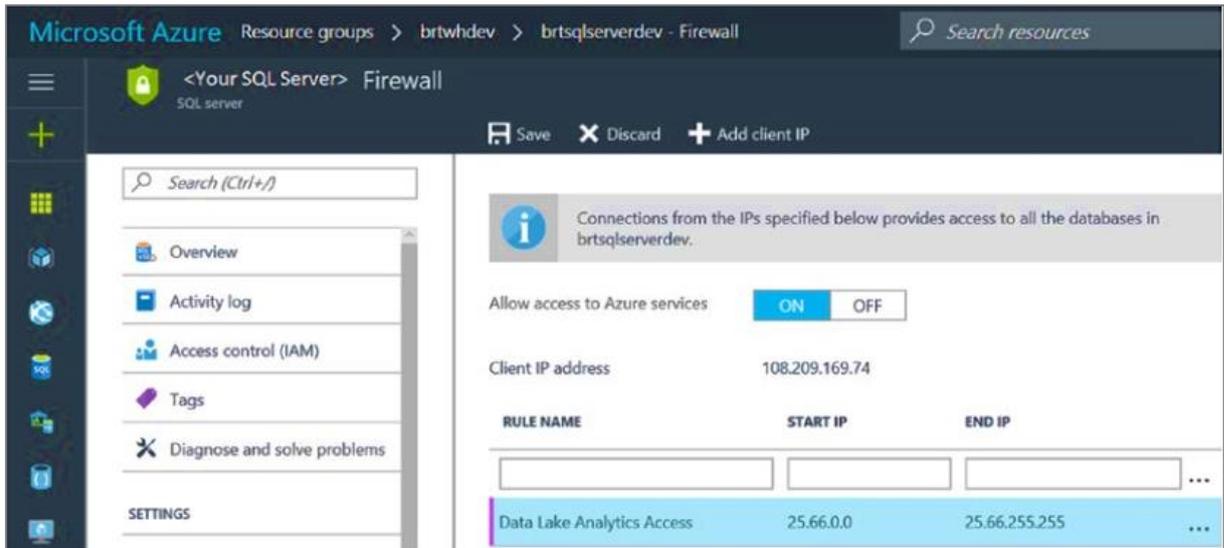
Gambar 7.18. Azure Data Lake Store: tampilan izin pengguna melalui Portal Azure

Konfigurasi Akses ke Database Azure SQL Jarak Jauh: Izinkan Rentang IP di Firewall SQL Server untuk Layanan ADLA yang Menjalankan Kueri U-SQL

Langkah ini memberikan akses ke layanan ADLA untuk mengakses Azure SQL Server Anda dan database Azure SQL terkait. Navigasikan ke instans Azure SQL Database Server yang ditargetkan melalui Portal Azure:

1. Klik ikon SQL Server untuk masuk ke halaman pengaturan.
2. Klik ikon Firewall di bilah navigasi kiri.
3. Buat aturan baru dengan rentang 25.66.0.0 hingga 25.66.255.255.
4. Klik ikon Simpan di bilah navigasi atas untuk menyimpan perubahan ini.

Tangkapan layar pada Gambar 7.19 menggambarkan layar konfigurasi Azure SQL Server Firewall dan aturan firewall baru yang ditambahkan untuk akses ADLA.



Gambar 7.19. Menambahkan aturan firewall SQL Server baru untuk mengizinkan akses jarak jauh melalui layanan Azure Data Lake Analytics

Buat Database ADLA SQL Baru dan tabel SQL melalui U-SQL Query

Langkah selanjutnya adalah membuat ADLA SQL Database dan tabel SQL. Untuk melakukannya, navigasikan ke akun ADLA Anda melalui Portal Azure lalu pilih ikon +Pekerjaan Baru untuk mengirimkan pekerjaan U-SQL baru. Masukkan kode U-SQL ini untuk membuat database dan tabel ADLA SQL:

```
//Create Table
CREATE DATABASE IF NOT EXISTS SearchMaster;

//Create Table
CREATE TABLE IF NOT EXISTS SearchMaster.dbo.SearchLog
(
    //Define schema of table
    UserId          int,
    Start           date time,
    Region          string ,
    Query           string,
    Duration        int,
    Urls            string,
    ClickedUrls    string,
    INDEX idx1 //Name of index
    CLUSTERED (Region ASC) //Column to cluster by
    PARTITIONED BY HASH (Region) //Column to partition by
);
```

Gunakan PowerShell untuk Membuat Rahasia Katalog Baru di Database ADLA

Rahasia ini berisi kata sandi untuk login SQL dan string koneksi untuk database Azure SQL.:

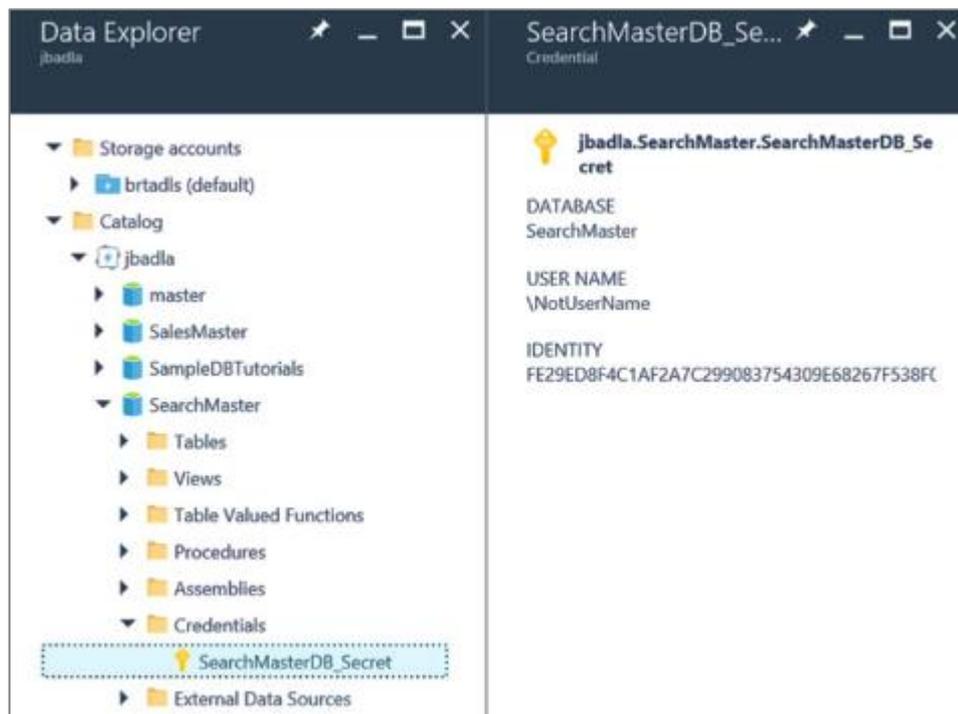
```
#Login (Microsoft Azure Login screen will appear):
Login-AzureRmAccount

#Show your available Azure Subscriptions:
Get-AzureRmSubscription

#Connect to Azure Subscription that contains the ADLA Database:
Set-AzureRMContext -SubscriptionId 00000000-0000-0000-0000-000000000000

New-AzureRmDataLakeAnalyticsCatalogCredential -AccountName "ContosoADLAccount" `
-DatabaseName "ContosoADLADB"
-CredentialName "ContosoAzureSQLDB_Secret"
-Credential (Get-Credential)
-DatabaseHost "ContosoSQLSVR.database.windows.net" -Port 1433
```

Anda dapat memverifikasi bahwa perintah PowerShell ini membuat objek kredensial baru di database ADLA SQL Anda dengan menavigasi ke database SQL Anda menggunakan ADLA Data Explorer. Mulailah dengan memperluas node untuk database Anda dan kemudian memperluas node Credentials, seperti yang ditunjukkan pada Gambar 7-20.



Gambar 7.20. Melihat kredensial database Azure Data Lake Analytics SQL yang dibuat oleh PowerShell

Buat kredensial dengan identitas yang cocok dengan nama AzureRm Data Lake Analytics Catalog Credential ini digunakan dalam skrip PowerShell (ContosoAzureSQLDB Secret) di Database ADLA menggunakan kueri U-SQL ini:

```
//Connect to ADLA Database
USE DATABASE YourADLADatabaseName;
//Create CREDENTIAL
//IDENTITY: ADLA SQL Catalog Secret, MUST MATCH name chosen in prior
  PowerShell script ->
  "ContosoAzureSQLDB_Secret"
CREATE CREDENTIAL IF NOT EXISTS [ContosoAzureSQLDB_Secret] WITH
USER_NAME = "
YourAzureSQLDB_Username ", IDENTITY = " YourAzureSQLDB_Secret";
```

Buat Sumber Data di Database ADLA dengan Referensi ke Azure SQL Database

Gunakan kueri U-SQL ini:

```
// Create External Data Source on ADLA SQL DB
CREATE DATA SOURCE IF NOT EXISTS [ASQL_YOURDB]
FROM AZURESQldb
WITH (PROVIDER_STRING = "Initial Catalog = YourASQLDB;
Trusted_Connection = False; Encrypt=True",
      CREDENTIAL = [ContosoAzureSQLDB_Secret],
      REMOTABLE_TYPES = (bool, byte, sbyte, short, ushort, int, uint,
                          long, ulong, decimal, float, double, string, DateTime)
);
```

Buat Tabel Eksternal di Database ADLA SQL Berdasarkan Sumber Data Jarak Jauh

Gunakan kueri U-SQL ini. Perhatikan bahwa skema tabel ADLA SQL harus mencocokkan skema tabel jarak jauh dengan tipe bidang yang sesuai.

```
//CREATE EXTERNAL TABLE in ADLA SQL database to represent the remote
database table.
```

```
CREATE EXTERNAL TABLE DailySales (
  OrderID      int?,
  SalesDate    DateTime?,
  Customer     string,
  Street       string,
  City         string,
  Region       string,
  State        string,
  Zip          string,
  SubTotal     decimal?,
  SalesTax     decimal?,
  SalesTotal   decimal?,
) FROM [ASQL_YOURDB] LOCATION "dbo.DailySales";
```

Kueri Tabel Database Azure SQL Eksternal Federasi Baru

Keluarkan hasilnya ke file teks (.CSV) menggunakan pernyataan kueri U-SQL ini:

```
@query = SELECT * FROM DailySales;

OUTPUT @query
TO "/Output/TestFederatedfile.csv"
USING Outputters.Csv();
```

Pada titik ini, Anda telah berhasil mengonfigurasi lingkungan Anda sesuai dengan prosedur pengaturan dan prasyarat. Anda sekarang seharusnya dapat mengeluarkan kueri pekerjaan U-SQL terhadap penyimpanan data eksternal jarak jauh (dalam hal ini database Azure SQL). Kemampuan Federated Query ADLA menawarkan keuntungan besar dalam produktivitas karena memungkinkan Anda melakukan kueri dan menghitung data jarak jauh yang ada di mana saja dan kemudian menggabungkan data tersebut dengan beberapa sumber cloud tambahan. Pada saat penulisan ini, sumber data eksternal berikut diperbolehkan dalam Kueri Federasi ADLA:

- *AZURESQLDB*: Menentukan bahwa sumber eksternal adalah instans Microsoft Azure SQL Database.
- *AZURESQLDW*: Menentukan bahwa sumber eksternal adalah instans gudang Data Microsoft Azure SQL.
- *SQLSERVER*: Menentukan bahwa sumber eksternal adalah instans Microsoft SQL Server yang berjalan di Microsoft Azure VM yang dapat diakses. Hanya SQL Server 2012 dan versi yang lebih baru yang didukung.

Menggabungkan Kumpulan Baris

Bahasa U-SQL Azure Data Lake menyediakan sejumlah operator untuk menggabungkan kumpulan baris dari berbagai sumber data. Berikut adalah operator saat ini yang didukung dalam bahasa pemrograman ADLA U-SQL:

- LEFT OUTER JOIN
- LEFT INNER JOIN
- RIGHT INNER JOIN
- RIGHT OUTER JOIN
- FULL OUTER JOIN
- CROSS JOIN
- LEFT SEMI JOIN
- RIGHT SEMI JOIN
- EXCEPT ALL
- EXCEPT DISTINCT
- INTERSECT ALL
- INTERSECT DISTINCT
- UNION ALL
- UNION DISTINCT

Integrasi Portal Azure

Seperti yang Anda lihat dalam ikhtisar singkat ADLA ini, ada banyak tugas yang dapat diselesaikan langsung di Portal Azure. Misalnya, Anda dapat menyelesaikan semua tugas ini langsung dari Portal Azure agar (segera) menjadi sangat produktif:

- Buat akun ADLA baru.
- Penulis skrip U-SQL. Buka Simpan.
- Kirimkan pekerjaan U-SQL.
- Membatalkan pekerjaan yang sedang berjalan.
- Penyediaan pengguna yang dapat mengirimkan pekerjaan ADLA.
- Visualisasikan statistik penggunaan (hitung jam).
- Visualisasikan bagan manajemen pekerjaan.

Tugas-tugas ini terdiri dari siklus hidup penuh pelaksanaan pekerjaan U-SQL mulai dari pembuatan, penyerahan, pemantauan, dan akhirnya menganalisis hasil pekerjaan. Tentu saja, untuk pengembang ADLA hard-core, Visual Studio dan Data Lake Tools untuk Visual Studio mungkin lebih cocok. Untuk mendapatkan gambaran tentang kemampuan Alat Data Lake untuk Visual Studio, lihat tautan berikut.

7.4 MANAJEMEN DAN ADMINISTRASI YANG DISEDERHANAKAN

Selain kemampuan manajemen berbasis web yang tersedia di Portal Azure, terdapat kemampuan manajemen dan pemantauan tambahan yang tersedia untuk Azure Data Lake Store dan Analytics.

- *Otomatisasi Tugas melalui Skrip PowerShell*: Skrip ini memungkinkan Anda menciptakan lingkungan pemrosesan data Big Data yang sangat otomatis.
- *Kontrol Akses Berbasis Peran (RBAC) dengan Azure Active Directory (AAD)*: Menyediakan layanan integrasi autentikasi dan otorisasi pengguna yang lancar. Pemantauan Operasional dan Aktivitas Pelayanan
- *Manajemen Pekerjaan*: Jumlah total pekerjaan yang diserahkan serta jumlah pekerjaan yang berhasil, gagal, atau dibatalkan
- *Penggunaan Komputasi Pekerjaan*: Jumlah jam komputasi yang digunakan oleh pekerjaan

U-SQL: Optimasi Sudah Terintegrasi

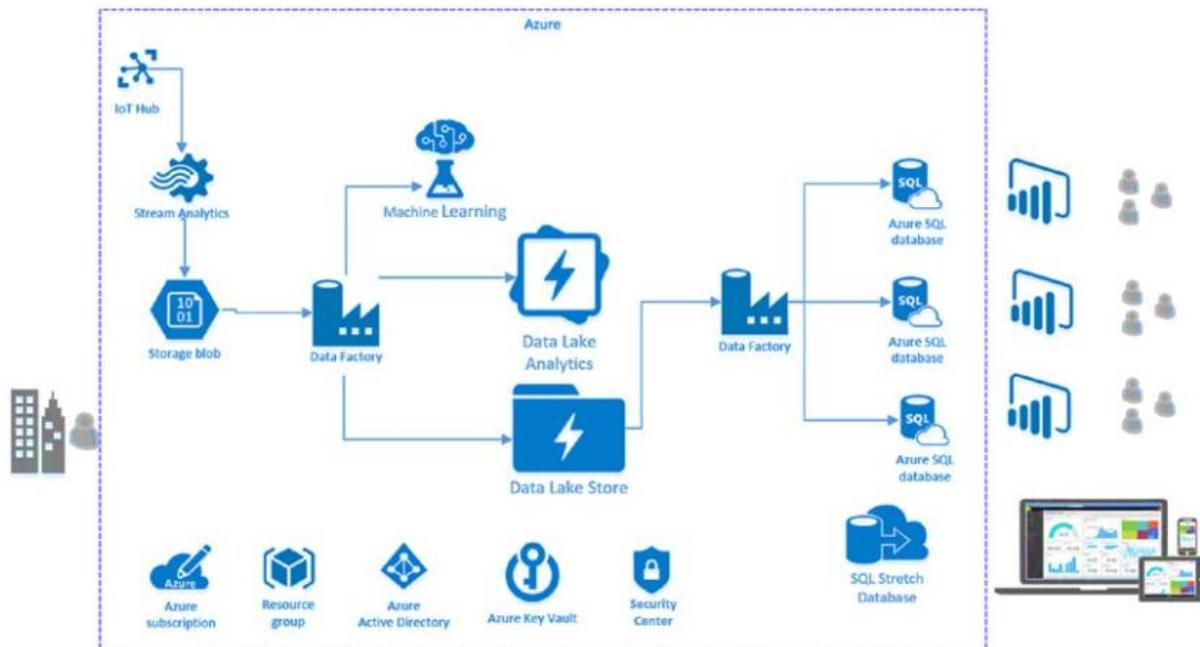
Bahasa ADLA U-SQL benar-benar unik karena memiliki kemampuan pengoptimalan canggih berikut ini:

- Otomatis “in-lining” ekspresi U-SQL, yang berarti bahwa seluruh skrip U-SQL mengarah ke model eksekusi tunggal.
- Rencana eksekusi yang dioptimalkan secara “out-of-the-box” dan tanpa memerlukan intervensi pengguna apa pun.
- Optimasi paralelisasi otomatis per pekerjaan dan berbasis pengguna.
- Visibilitas mendetail tentang langkah-langkah eksekusi tugas U-SQL, untuk tujuan debugging dan pengoptimalan.
- Fungsi peta panas untuk membantu mengidentifikasi hambatan kinerja.

Menerapkan ADLA dalam Implementasi Referensi

Sekarang setelah Anda memiliki latar belakang yang kuat tentang fitur dan kemampuan ADLA dan bahasa pemrograman U-SQL, Anda akan menggunakan pengetahuan ini dengan menerapkan beberapa bagian penting lagi dari implementasi referensi.

Sebagai penyegaran singkat, Gambar 7-21 mengilustrasikan “gambaran besar” implementasi referensi dan peran ADLA dalam implementasi arsitektur ini.



Gambar 7-21. Peran Data Lake Analytics dalam arsitektur implementasi referensi

Seperti dapat dilihat pada Gambar 7-21, Anda akan memanfaatkan pembelajaran utama kami dari Bab 6 (Pabrik Data) dan Bab 7 (Analisis Data Lake) untuk melanjutkan pergerakan data dan aktivitas transformasi untuk aplikasi referensi di bab ini. Secara khusus, di bagian berikutnya, Anda akan memproses data yang dipindahkan ke Azure Data Lake Store dari penyimpanan Azure Blob (melalui pekerjaan Azure Data Factory), yang mewakili data streaming IoT yang masuk. Anda kemudian akan menggabungkan data IoT mentah yang masuk ini dengan hasil panggilan layanan Web Azure Machine Learning yang dibuat secara real-time selama pekerjaan penyerapan Stream Analytics dan kemudian disimpan ke penyimpanan Azure Blob. Ini adalah salinan jalur data "panas" yang memeriksa kesehatan anggota tim. Tujuan jalur keluaran “panas” yang ditentukan dalam Bab 5 adalah untuk Power BI; kami juga menyimpan output ke Azure blob Storage untuk catatan yang lebih permanen dari hasil streaming real-time yang diproses dengan panggilan Machine Learning Web Service.

Sasaran di bagian ini adalah mengidentifikasi file input (.CSV) yang diperlukan untuk membuat tabel gabungan SQL Azure Data Lake Analytics (ADLA) yang menggabungkan sumber data berikut:

1. Data Streaming IoT Hub Historis.
 - A. JOIN Keys: userid, stempel waktu.
2. Hasil Panggilan Layanan Web Pembelajaran Mesin IoT Hub Historis.

- A. JOIN Keys: userid, stempel waktu.
- 3. Data Referensi Anggota Tim Afiliasi Perusahaan.
 - A. JOIN Keys: userid, id.

Dalam implementasi referensi yang ada, ketiga file ini dibuat di Bab 5 dan 6 saat kita membahas topik Azure Stream Analytics dan Azure Data Factory.

JOIN Keys di bawah setiap file mewakili kunci yang akan kita gunakan untuk membuat JOIN logis antara dua tabel data ini untuk menggabungkan catatan bersama-sama dan dengan demikian memperkaya dan menyempurnakan data dengan menggabungkan sumber data ini bersama-sama.

JOIN dengan data referensi sangat penting agar kita dapat mencocokkan record (berdasarkan ID pengguna) dengan badan usaha yang sesuai berdasarkan isi kolom “nama perusahaan”. Kami kemudian akan menggunakan data ini untuk membuat output individual yang kemudian akan digunakan untuk mengisi Database Azure SQL individual berdasarkan entitas bisnis. Dengan memaparkan data sebagai database Azure SQL ke setiap Badan Usaha, kami dapat sepenuhnya mengaktifkan pengalaman Business Intelligence (BI) “layanan mandiri”. Manfaat tambahannya adalah data database Azure SQL benar-benar terpisah dan dapat diamankan lebih lanjut untuk mengurangi risiko hukum atas paparan data yang tidak disengaja kepada Badan Usaha lain.

Perhatikan bahwa dalam skenario ini, tujuan kami adalah membuat catatan data yang sangat “lebar” yang sangat dinormalisasi untuk membantu meningkatkan kinerja baca. Hal ini juga memungkinkan pengalaman pelaporan yang lebih mudah dan lebih kaya, karena tidak diperlukan banyak penggabungan pada tabel tambahan.

Dua file pertama (Historical IOT Hub*.csv) dibuat sebagai hasil pekerjaan Analisis Aliran yang mempertahankan aliran data masuk ke jalur penyimpanan keluaran data panas, hangat, dan dingin. File Data Streaming IoT Hub Historis diambil sebagai bagian dari “jalur dingin” yang menampilkan semua pesan peristiwa sensor yang dikirim ke IoT Hub secara real-time. Pekerjaan Stream Analytics mempertahankan output ini langsung ke file dalam kontainer penyimpanan blob Azure.

File hasil Panggilan Layanan Web Pembelajaran Mesin IoT Hub Historis dihasilkan sebagai bagian dari “jalur panas” yang juga mengisi himpunan data Power BI untuk mengekspos pemantauan dasbor real-time terhadap kesehatan anggota tim saat bekerja. Data ini berisi semua parameter input dan hasil Panggilan Layanan Web Machine Learning (ML) terkait yang dikembalikan secara real-time, karena data sensor setiap anggota tim sedang diproses oleh pekerjaan Azure Stream Analytics.

Tujuan dari panggilan Layanan Web Pembelajaran Mesin adalah untuk memprediksi (secara real-time) apakah anggota tim berada pada titik kelelahan fisik atau kelelahan yang dapat mengakibatkan kemungkinan lebih tinggi terjadinya kecelakaan terkait pekerjaan. Kami membahas detail di balik pengembangan model Machine Learning dan memaparkannya sebagai layanan web yang mudah dikonsumsi di bab berikutnya.

File Data Referensi dibuat sebagai hasil dari Pekerjaan Azure Data Factory untuk memperbarui data Referensi, yang, pada gilirannya, digunakan oleh pekerjaan Azure Stream

Analytics sebagai "input" untuk data referensi yang dapat digabungkan ke streaming masuk data.

7.5 GAMBARAN UMUM TUGAS YANG HARUS DISELESAIKAN

Di bagian ini, kami menguraikan tugas-tugas yang perlu Anda selesaikan. Tujuannya adalah untuk mengisi Azure SQL Database individual (menurut Badan Usaha) dengan informasi gabungan dari file sebelumnya. Data ini mewakili tampilan historis dari semua data sensor yang direkam untuk masing-masing anggota tim entitas bisnis bersama dengan hasil panggilan Layanan Web Pembelajaran Mesin.

Data tersebut dapat digunakan untuk beberapa tujuan, antara lain:

- *Meningkatkan Umpan Balik Anggota Tim:* Memberikan analisis historis untuk membantu menciptakan putaran umpan balik tambahan (berdasarkan pembacaan sensor fisik) untuk membantu menyimpulkan dan menghubungkan “kausalitas” seputar peristiwa sehari-hari. Hasil positif termasuk mencatat rencana perbaikan atau perubahan perilaku yang berhubungan dengan kesehatan.
- *Audit Operasi:* Sebagai sarana untuk memberikan latar belakang dan analisis operasional, hukum, atau medis lebih lanjut. Data ini dapat dianggap sangat penting dan dapat dianalogikan dengan perekam “kotak hitam”: jika terjadi kejadian yang tidak terduga.
- *Penyempurnaan Model Prediksi:* Untuk membantu memberikan umpan balik guna lebih menyempurnakan model prediksi Machine Learning untuk penyempurnaan dan penyempurnaan model di masa mendatang berdasarkan pembaruan data dunia nyata.

Tugas yang akan Anda selesaikan di bagian selanjutnya meliputi:

1. Ubah tugas Data Factory yang disebut CopyFromBlobToDataLake untuk menyertakan tiga file yang akan dipindahkan ke Data Lake.
2. Buat dan uji Skrip Ekstrak CSV Azure Data Lake baru. Uji logika ekstraksi untuk ketiga file untuk penguraian bidang dan penetapan jenis bidang yang tepat.
 - Uji Ekstrak 1: data streaming IoT Hub
 - Ekstrak Uji 2: Data hasil Layanan Web Pembelajaran Mesin streaming IoT Hub
 - Ekstrak Tes 3: Data referensi tim
3. Buat database dan tabel Azure Data Lake Analytics (ADLA) baru. Isi tabel baru berdasarkan hasil JOIN dari ketiga file ekstrak.
4. Buat file ekstrak terpisah (satu untuk setiap entitas bisnis) untuk menampung data yang diekstrak dari tabel Database ADLA yang baru diisi.
5. Buat database Azure SQL terpisah (satu untuk setiap entitas bisnis) untuk menampung data yang diekstraksi dari database ADLA.
6. Buat pekerjaan Azure Data Factory untuk secara otomatis mengisi database Azure SQL berdasarkan ekstrak SQL (menurut entitas bisnis) dari database ADLA.

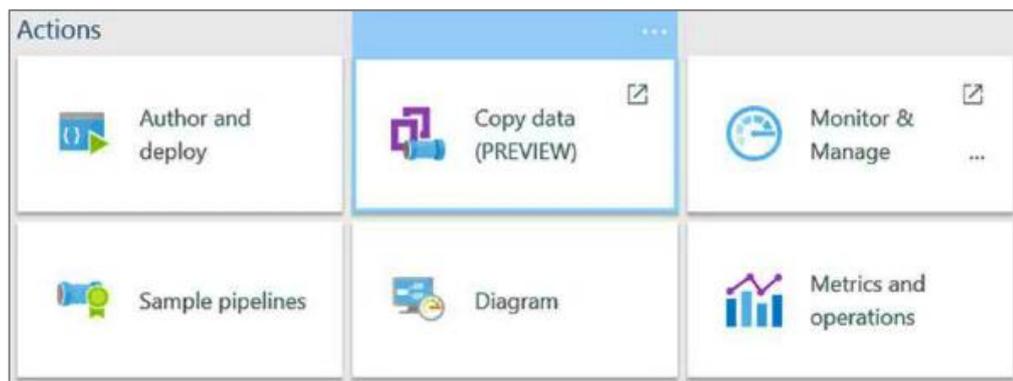
Sekarang kita telah menguraikan rencananya, sekarang saatnya untuk memulai.

MEMODIFIKASI PEKERJAAN PABRIK DATA AZURE: “COPYFROMBLOBTODATALAKE”

Di bagian ini, kita akan membahas cara memodifikasi alur Data Factory yang sudah ada bernama CopyFromBlobToDataLake. Pada dasarnya, kita perlu menambahkan dua aktivitas penyalinan lagi ke alur untuk memindahkan kedua file ini ke Azure Data Lake:

- Hasil Panggilan Layanan Web Pembelajaran Mesin Data Streaming IoT Hub Historis dari jalur "Hot".
- Data Referensi Anggota Tim Afiliasi Perusahaan Anggota Tim.

Cara termudah untuk melakukannya adalah dengan menavigasi ke Pekerjaan Azure Data Factory yang ada, yang disebut CopyFromBlobToDataLake, lalu pilih opsi untuk Salin Data (PREVIEW) seperti yang ditunjukkan pada Gambar 7.22.



Gambar 7-22. Opsi Salin Data Pabrik Data Azure

Memilih opsi ini akan memungkinkan Anda dengan mudah menambahkan fungsionalitas penyalinan ke pekerjaan Penyalinan Alur Pabrik Data yang ada. Ikuti langkah-langkah di Wisaya PREVIEW Penyalinan Data Pabrik data Azure untuk membuat pekerjaan penyalinan untuk file tambahan pertama yang kita perlukan—Hasil Panggilan Layanan Web Pembelajaran Mesin Data Streaming Hub IoT Historis. Berikut adalah garis besar input dan output tingkat tinggi yang diperlukan untuk membuat jalur penyalinan baru ini.

Pekerjaan Salin Baru: Hasil Panggilan Layanan Web Pembelajaran Mesin Data Streaming IoT Hub Historis:

- ❖ Kumpulan Data INPUT (Gumpalan):
- ❖ File-file ini akan ditemukan dalam kontainer Blob bernama streamingmldata dan akan dibuat dalam struktur folder Tahun/Bulan/Hari/Jam.
- ❖ Kumpulan Data KELUARAN (Data Lake):
- ❖ Atur kumpulan data keluaran agar mengarah ke nama file dalam bentuk StreamingMLResults DataFile{tahun}{bulan}{hari}{jam}.csv.

Selanjutnya, tambahkan pekerjaan penyalinan lain ke pekerjaan Azure Data Factory Pipeline yang ada untuk menyalin file data kedua, file .CSV Data Referensi Anggota Tim. Sekali lagi, kita akan menggunakan Copy Data Wizard untuk membuatnya cepat dan mudah. Lihat panduan himpunan data masukan dan keluaran berikut untuk menemukan dan mereplikasi data Anda sesuai dengan implementasi referensi. Berikut adalah garis besar masukan dan keluaran yang diperlukan untuk membuat saluran “salinan” baru ini.

Pekerjaan Salin Baru: Data Referensi Anggota Tim:

- ❖ Kumpulan Data INPUT (Gumpalan):
- ❖ File ini akan ditemukan dalam kontainer Blob bernama refdata-team dan akan dibuat dalam file bernama "TeamReferenceData" + "{year}{month}{day} {hour}" + ".csv".
- ❖ Perhatikan bahwa pembuatan file akan menyertakan ekstensi {tahun}{bulan}{hari} {hour} ke nama file.
- ❖ Kumpulan Data KELUARAN (Data Lake):
- ❖ Atur kumpulan data keluaran agar mengarah ke folder bernama streamingmldata. Nama filenya adalah "TeamReferenceData" + "{year}{month}{day}{jam}" + ".csv".

Pada titik ini, kami telah berhasil memodifikasi pekerjaan Azure Data Factory Pipeline untuk menyalin dua file tambahan ke Azure Data Lake untuk membuat tabel ADLA SQL Analisis Pembelajaran Mesin historis utama. Langkah terakhir adalah menjalankan pekerjaan Data Factory Copy yang diperbarui dan benar-benar memindahkan file data dari Azure Blob Storage ke Azure Data Lake.

7.6 PENGUJIAN CSV AZURE DATA LAKE BARU UNTUK TIGA FILE INPUT

Bagian ini berfokus pada membaca tiga file .CSV yang kami salin ke Azure Data Lake dari Azure Blob Storage. Untuk menyederhanakan pengalaman pengembangan, kami akan membuat pekerjaan ADLA terpisah untuk menguji ekstraksi masing-masing dari tiga file secara individual. Untuk setiap file, kami hanya akan menguji logika parsing kami hanya dengan membaca file dan kemudian mengeluarkannya. Untuk memulai, navigasikan ke Akun Azure Data Lake Anda di Portal Azure, lalu klik ikon + Pekerjaan Baru.

Pada titik ini, Anda dapat memberi nama pekerjaan baru Anda "TEST EXTRACT of STREAMING DATA CSV File" dan kemudian menempelkan konten kode ini ke dalam file itu.

```
@streamingdata =
    EXTRACT
        userid                string,
        age                   int,
        height                int,
        weight                int,
        heartratebpm         int,
        breathingrate        double,
        temperature          double,
        steps                 int,
        velocity              double,
        altitude              int,
        ventilization         double,
        activity              double,
        cadence               double,
        speed                 int,
        hib                   double,
        heartrateredzone     double,
        heartratevariability double,
```

```

        status                int,
        id                    string,
        deviceid              string,
        messagetype           int,
        longitude             double,
        latitude              double,
        timestamp             string,
        eventprocessedutctime string,
        partitionid          int,
        eventenqueuedutctime string,
        firstname             string,
        lastname              string,
        username              string,
        utype                 int,
        phone                 string,
        email                 string,
        gender                int,
        race                  int
FROM "/streamingdata/2016/StreamingResultsDailyFile{*}.csv"
USING Extractors.Csv();

OUTPUT @streamingdata
TO "/output/TESTStreamingResultsDailyFile20161222.csv"
USING Outputters.Csv();

```

Perhatikan bahwa tujuan skrip ini adalah untuk tujuan pengujian saja; tidak apa-apa jika beberapa aspek di-hardcode. Pilih Simpan Sebagai untuk mengunduh salinan secara lokal dan menyimpan skrip. Kirimkan pekerjaan dan periksa hasil yang diharapkan. Perbaiki kesalahan apa pun dan kirimkan kembali pekerjaan sampai Anda berhasil. Saat bekerja dengan skrip ADLA U-SQL, seringkali lebih baik memecahnya menjadi potongan-potongan kode yang lebih kecil, daripada mencoba men-debug skrip yang lebih besar dan seringkali lebih rumit. Pastikan ekstrak file pertama berfungsi dengan benar sebelum melanjutkan ke pengkodean dua ekstraksi file berikutnya.

■ Tip Tergantung pada konfigurasi lingkungan Anda, Anda mungkin melihat beberapa file .CSV yang dibuat secara unik dan disimpan dalam struktur folder rekursif.

Kunci untuk memproses skenario jenis ini adalah dengan menggunakan ekstensi “wildcard” saat memberi nama file masukan. Dalam skrip ADLA U-SQL yang diselesaikan dengan menggunakan notasi {*}. Misalnya, Anda dapat menggunakan nama file DailySales{*}.csv untuk menyertakan semua file yang dimulai dengan nama DailySales dan diakhiri dengan ekstensi file .CSV.

Buat Ekstrak #2: BACA File .CSV Input Data ML Streaming

Pada titik ini, Anda dapat mengganti nama pekerjaan Anda yang ada “TEST EXTRACT of STREAMING ML DATA CSV File” dan kemudian menempelkan kode ini ke dalam file.

```
// @streamingMLdata
@streamingmldata =
    EXTRACT
        userid                string
        timestamp             string,
        eventprocessedutctime string
        partitionid           int,
        eventenqueuedutctime  string,
        breathingrate         double,
        temperature           double,
        ventilization         string,
        activity              double,
        cadence               double,
        heartratebpm          int,
        velocity              double,
        speed                 int
        hib                   double
        heartrateredzone     double
        heartratevariability  double
        scored_labels        string
        scored_prob          double

    FROM "/streamingmldata/2016/StreamingMLResultsDataFile{*}.csv"
    USING Extractors.Csv();
OUTPUT @streamingmldata
TO "/output/TESTStreamingMLResultsDataFile20161222.csv"
    USING Outputters.Csv();
```

Pilih Simpan Sebagai untuk mengunduh salinan secara lokal dan menyimpan skrip. Jika Anda sudah siap, kirimkan pekerjaan dan periksa hasil keluaran yang diharapkan. Perbaiki kesalahan apa pun dan kirimkan kembali pekerjaan hingga Anda dapat menghasilkan file keluaran yang lengkap.

Buat Ekstrak #3: BACA Data Referensi Anggota Tim

Untuk pekerjaan ini, Anda dapat mengganti nama pekerjaan yang ada menjadi “TEST EXTRACT of TEAM REFERENCE DATA CSV File” dan kemudian menempelkan kode ini ke file baru.

```
// @ refdatateamdata
@refdatateamdata =
    EXTRACT
        authid                string,
        companyname           string,
```

```

    firstname          string,
    lastname           string,
    username           string,
    imageUrl           string,
    utype              string,
    address1           string,
    address2           string,
    address3           string,
    city               string,
    state              string,
    zip                string,
    country            string,
    phone              string,
    email              string,
    linkedin           string,
    facebook           string,
    twitter            string,
    blog               string,
    age                int,
    height             double,
    weight             double,
    gender             int,
    race               int,
    longitude          double,
    latitude           double,
    id                 string,
    cachettl           string,
    _rid               string,
    _self              string,
    _etag              string,
    _attachments       string,
    _ts                string
FROM "/refdata-team/TeamReferenceData{*}.csv"
USING Extractors.Csv();
OUTPUT @refdatateamdata
TO "/output/TESTTeamReferenceData20161222.csv"
USING Outputters.Csv()

```

Pilih Simpan Sebagai untuk mengunduh salinan secara lokal dan menyimpan skrip. Jika Anda sudah siap, kirimkan pekerjaan dan periksa hasil keluaran yang diharapkan. Perbaiki kesalahan apa pun dan kirimkan kembali pekerjaan hingga Anda dapat menghasilkan file keluaran yang lengkap. Pada titik ini, Anda telah berhasil memverifikasi tiga file input, dan Anda sekarang siap untuk BERGABUNG dengan mereka untuk mengisi database dan tabel ADLA.

Buat Database Azure Data Lake Analytics (Adla) Baru Dan Isi Tabel Baru Berdasarkan Hasil Gabung Dari Tiga File Ekstraksi

Langkah ini melibatkan pembuatan tugas U-SQL baru yang akan menggabungkan tiga skrip ekstrak pengujian yang Anda buat pada langkah sebelumnya untuk membuat serangkaian JOIN di antara kumpulan data. Pekerjaan tersebut kemudian akan membuat database Azure Data Lake, dan kemudian Anda akan secara otomatis mengisi tabel ADLA SQL baru dengan hasil operasi JOIN.

Perhatikan bahwa pada saat penulisan ini, tidak ada dukungan UPDATE atau MERGE yang tersedia dalam bahasa ADLA U-SQL untuk memperbarui tabel SQL. Oleh karena itu, Anda biasanya akan mengeluarkan perintah DROP dan CREATE U-SQL terhadap tabel ADLA SQL Anda untuk menyelesaikan operasi UPDATE dan MERGE.

Perhatikan juga bahwa U-SQL menyediakan operator gabungan umum seperti INNER JOIN, LEFT/RIGHT/FULL OUTER JOIN, dan SEMI JOIN untuk menggabungkan tidak hanya tabel ADLA SQL tetapi juga kumpulan baris apa pun (bahkan yang dihasilkan dari file) atau sumber data gabungan eksternal .

Untuk tujuan referensi, berikut adalah jenis operasi JOIN yang paling umum dan logika yang mendasari pencocokan data yang digunakannya:

- LEFT JOIN: Mengembalikan semua baris dari tabel kiri dan baris yang cocok dari tabel kanan.
- JOIN KANAN: Mengembalikan semua baris dari tabel kanan dan baris yang cocok dari tabel kiri.
- INNER JOIN: Mengembalikan semua baris ketika ada setidaknya satu kecocokan di kedua tabel.
- FULL JOIN: Mengembalikan semua baris ketika ada kecocokan di salah satu tabel.

Untuk pekerjaan ini, Anda dapat mengganti nama pekerjaan yang ada menjadi “EKSTRAK Semua 3 File dan JOIN ke Tabel ADLA”. Tugas ini menghasilkan skrip yang agak panjang. Anda akan mulai dengan menggabungkan tiga skrip ekstrak U-SQL sebelumnya ke dalam satu skrip. Anda kemudian akan menempelkan kode berikut untuk mengimplementasikan operasi GABUNG, membuat database ADLA, dan kemudian menampilkan GABUNG ke dalam tabel ADLA SQL baru.

```
//(1) JOIN the streaming data with the ML Results data
@joindata1 =
SELECT
    s.userid,
    s.age,
    s.height,
    s.weight,
    s.heartratebpm,
    s.breathingrate,
    s.temperature,
    s.steps,
    s.velocity,
```

```

s.altitude,
s.ventilization,
s.activity,
s.cadence,
s.speed,
s.hib,
s.heartrateredzone,
s.heartratevariability,
s.status,
s.id,
s.deviceid,
s.messagetype,
s.longitude,
s.latitude,
s.timestamp,
s.eventprocessedutctime,
s.partitionid,
s.eventenqueuedutctime,
s.firstname,
s.lastname,
s.username,
s.utype,
s.phone,
s.email,
s.gender,
s.race,
m.mluserid,
m.mltimestamp,
m.scoredlabels,
m.scoredprob
FROM @streamingdata AS s
LEFT JOIN @streamingmldata AS m
ON s.timestamp == m.mltimestamp
AND s.userid == m.mluserid;

```

//(2) JOIN the NEW streaming data + ML data by USER to GET the Associated COMPANY

```

@joindata2 =
SELECT
    j.userid,
    j.age,
    j.height,
    j.weight,
    j.heartratebpm,
    j.breathingrate,
    j.temperature,
    j.steps,
    j.velocity,

```

```

        j.altitude,
        j.ventilization,
        j.activity,
        j.cadence,
        j.speed,
        j.hib,
        j.heartrateredzone,
        j.heartratevariability,
        j.status,
        j.id,
        j.deviceid,
        j.messages,
        j.longitude,
        j.latitude,
        j.timestamp,
        j.eventprocessedutctime,
        j.partitionid,
        j.eventenqueuedutctime,
        j.firstname,
        j.lastname,
        j.username,
        j.utype,
        j.phone,
        j.email,
        j.gender,
        j.mluserid,
        j.mltimestamp,
        j.scoredlabls,
        j.scoredprob,
        j.race,
        r.companyname,
        r.rid
FROM @joindata1 AS j
    LEFT JOIN @refdatateamdata AS r
        ON j.userid == r.rid;
OUTPUT @joindata2
    TO "/output/SQLJoinTestResults.csv"
    USING Outputters.Csv();

DROP DATABASE IF EXISTS IOTStreamingDataMLHistoryDB;
CREATE DATABASE IOTStreamingDataMLHistoryDB;
USE DATABASE IOTStreamingDataMLHistoryDB;

CREATE TABLE StreamingDataMLHistory(
    INDEX sl_idx CLUSTERED (userid ASC)
    PARTITIONED BY HASH (userid) )
    AS SELECT * FROM @joindata2 AS S; // Note: Automatic Table Schema
    generation

```

Seperti sebelumnya, pilih Simpan Sebagai untuk mengunduh salinan skrip U-SQL secara lokal dan kemudian menyimpan skrip tersebut. Jika Anda sudah siap, kirimkan pekerjaan dan periksa hasil keluaran yang diharapkan di database ADLA SQL. Perbaiki kesalahan apa pun dan kirimkan kembali pekerjaan hingga Anda dapat membuat tabel ADLA SQL yang berhasil diisi.

Perhatikan bahwa pernyataan terakhir dalam skrip U-SQL mengeluarkan perintah CREATE TABLE yang diisi dari pernyataan SELECT. Manfaat besarnya di sini adalah Skema Tabel ADLA secara implisit disimpulkan (nama dan jenis kolom) dari kode ekstraksi file data .csv U-SQL sebelumnya. Hal ini meniadakan kebutuhan untuk menentukan tata letak tabel ADLA SQL sebelum diisi.

Buat File Ekstrak Terpisah (Satu Untuk Setiap Badan Bisnis) Untuk Menyediakan Data Yang Diekstrak Dari Tabel Database Azure Data Lake Analytics Yang Baru Dipopulasi

Pada langkah ini, Anda cukup membuat ekspor file .CSV terpisah (satu per entitas bisnis) dari tabel database ADLA yang baru diisi. Untuk memulai, navigasikan ke akun Azure Data Lake Anda di Portal Azure, klik ikon + new sheet, lalu tempelkan kode ini ke jendela pekerjaan baru:

```
//EXPORT data from ADLA Database Table to .CSV files by Business Entity

// Export #1 - WigiTech
@table1 = SELECT * FROM
[IOTStreamingDataMLHistoryDB].[dbo].[StreamingDataMLHistory]
WHERE companyname == "WigiTech";

OUTPUT @table1
    TO "/output/StreamingDataMLHistory-WigiTech.Csv"
    USING Outputters.Csv();

// Export #2 - Tall Towers
@table2 = SELECT * FROM
[IOTStreamingDataMLHistoryDB].[dbo].[StreamingDataMLHistory]
WHERE companyname == "Tall Towers";

OUTPUT @table2
    TO "/output/StreamingDataMLHistory-Tall-Towers.Csv"
    USING Outputters.Csv();

// Export #3 - The Complicated Badger
@table3 = SELECT * FROM
[IOTStreamingDataMLHistoryDB].[dbo].[StreamingDataMLHistory]
WHERE companyname == "The Complicated Badger";

OUTPUT @table3
    TO "/output/StreamingDataMLHistory-The-Complicated-Badger.Csv"
    USING Outputters.Csv();
```

Klik pada ikon Simpan Sebagai untuk menyimpan skrip U-SQL. Kemudian kirimkan pekerjaan dan periksa pembuatan file hasil keluaran .CSV.

Buat Database Dan Tabel Azure SQL (Satu Untuk Setiap Badan Bisnis) Untuk Menahan Data Yang Diekstrak Dari Tabel Database Adla

Langkah selanjutnya adalah membuat database Azure SQL dan tabel SQL baru untuk menyimpan informasi yang akan kita ekstrak dari tabel ADLA SQL yang kita isi dari tiga file teks CSV berikut:

- Data Streaming IoT Hub Historis
- Hasil Panggilan Layanan Web Pembelajaran Mesin IoT Hub Historis
- Data Referensi Anggota Tim Afiliasi Perusahaan

Dalam latihan ini, Anda akan melakukan pembuatan satu kali database Azure SQL baru dan tabel SQL baru untuk masing-masing dari tiga entitas bisnis dalam implementasi referensi. Masing-masing database ini akan berisi hasil penggabungan dataset (untuk salah satu entitas bisnis), tempat Anda menggabungkan ketiga file tersebut ke dalam tabel ADLA SQL. Untuk menyelesaikan tugas ini, Anda memerlukan prasyarat berikut:

- Informasi Azure SQL Server Database, seperti kredensial server, pengguna, dan kata sandi.
- Port firewall Azure SQL Server dibuka untuk memungkinkan alamat IP klien Anda mendapatkan informasi.
- SQL Server Management Studio (SSMS) atau Visual Studio Server Explorer untuk menjalankan skrip Transact-SQL untuk membuat tiga database dan tabel Azure SQL.

Setelah Anda membuat konektivitas ke database Azure SQL Server menggunakan SQL Server Management Studio atau Visual Studio Server Explorer, buka jendela Kueri baru dan masukkan skrip SQL ini:

```
USE [master]
GO
```

```
DROP DATABASE IF EXISTS [IOTDataMLHistory-WigiTech]
GO
```

```
CREATE DATABASE [IOTDataMLHistory-WigiTech]
GO
```

```
USE [IOTDataMLHistory-WigiTech]
GO
```

```
DROP TABLE IF EXISTS [dbo].[IOTDataMLHistory]
GO
```

```
CREATE TABLE [dbo].[IOTDataMLHistory] (
    [UserId] [char] (256) NOT NULL,
    [Age] [float] NOT NULL,
    [Height] [float] NOT NULL,
```

```

[Weight] [float] NOT NULL,
[HeartRateBPM] [float] NOT NULL,
[BreathingRate] [float] NOT NULL,
[Temperature] [float] NOT NULL,
[Steps] [float] NOT NULL,
[Velocity] [float] NOT NULL,
[Altitude] [float] NOT NULL,
[Ventilization] [float] NOT NULL,
[Activity] [float] NOT NULL,
[Cadence] [float] NOT NULL,
[Speed] [float] NOT NULL,
[HIB] [float] NOT NULL,
[HeartRateRedZone] [float] NOT NULL,
[HeartrateVariability] [float] NOT NULL,
[Status] [int] NOT NULL,
[Id] [char](256) NOT NULL,
[DeviceId] [char](256) NOT NULL,
[MessageType] [int] NOT NULL,
[Longitude] [float] NOT NULL,
[Latitude] [float] NOT NULL,
[Timestamp] [datetime2](7) NOT NULL,
[EventProcessedUtcTime] [datetime2](7) NOT NULL,
[PartitionId] [int] NOT NULL,
[EventEnqueuedUtcTime] [datetime2](7) NOT NULL,
[FirstName] [char](256) NOT NULL,
[LastName] [char](256) NOT NULL,
[UserName] [char](256) NOT NULL,
[UType] [char](256) NOT NULL,
[Phone] [char](256) NOT NULL,
[Email] [char](256) NOT NULL,
[Gender] [char](256) NOT NULL,
[MLUserid] [char](256) NOT NULL,
[MLTimestamp] [char](256) NOT NULL,
[ScoredLabels] [char](256) NOT NULL,
[ScoredProb] [float] NOT NULL,
[Race] [int] NOT NULL,
[CompanyName] [char](256) NOT NULL,
[Rid] [char](256) NOT NULL
)
GO

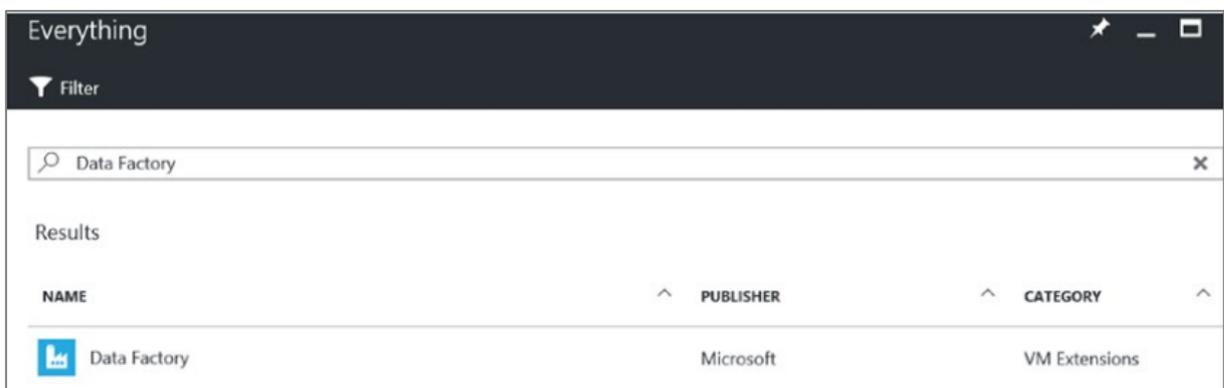
```

Setelah menjalankan skrip ini tiga kali (memodifikasinya untuk mencerminkan setiap entitas bisnis), Anda akan siap untuk mengisi tabel database Azure SQL dengan menggunakan pekerjaan Azure Data Factory.

Buat Pekerjaan Azure Data Factory Untuk Mengisi Database Azure Sql Secara Otomatis (Menurut Entitas Bisnis) Menggunakan Ekstraksi Query Sql Dari Database Azure Data Lake

Langkah terakhir ini akan menyelesaikan perjalanan analisis dan pelaporan data dengan memaparkan data sensor IoT historis dan hasil panggilan Layanan Web Pembelajaran Mesin terkait untuk dikonsumsi oleh entitas bisnis dalam skenario implementasi referensi. Pada titik ini, semua kerja keras telah selesai, dan yang harus Anda lakukan adalah membuat pekerjaan Azure Data Factory untuk memilih data dari database ADLA SQL, memfilter hasilnya berdasarkan entitas bisnis, lalu memasukkan data ke dalam database Azure SQL individual yang Anda buat pada langkah terakhir.

Untuk memulai melalui Portal Azure, pertama-tama navigasikan ke grup sumber daya Azure Anda yang sudah ada. Cari Data Factory lalu tambahkan pekerjaan Data Factory baru, seperti yang ditunjukkan pada Gambar 7-23.



Gambar 7.23. Menambahkan pekerjaan Azure Data Factory baru ke grup sumber daya

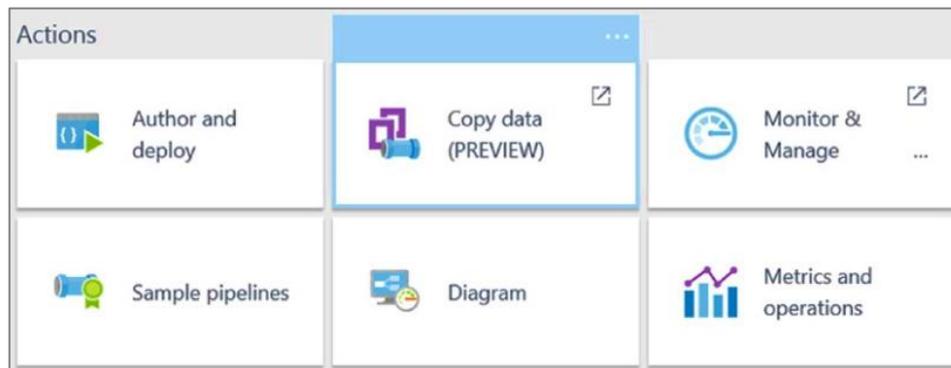
Selanjutnya, selesaikan parameter konfigurasi Azure Data Factory Job, seperti yang ditunjukkan pada Gambar 7-24.

The screenshot shows the 'New data factory' configuration form. It has a title bar 'New data factory' with standard window controls. The form contains several fields:

- Name:** A text input field containing 'PopulateAzureSQLDBs' with a green checkmark on the right.
- Subscription:** A dropdown menu.
- Resource Group:** A radio button group with 'Create new' and 'Use existing' options. The 'Use existing' option is selected. Below it is a dropdown menu containing 'brtwhdev'.
- Location:** A dropdown menu containing 'East US'.

Gambar 7.24. Parameter pekerjaan Azure Data Factory

Klik ikon Buat, lalu pekerjaan Data Factory baru akan diprovisikan ke Azure Resource Group Anda setelah beberapa menit. Setelah tugas Data Factory baru disediakan ke lingkungan Anda, navigasikan ke tugas Data Factory baru dan pilih ikon Salin Data (PREVIEW) untuk meluncurkan Copy Wizard.



Gambar 7.25. Wisaya Penyalinan Data Pabrik Data

Selanjutnya, telusuri Wizard Penyalinan Data untuk membuat pekerjaan alur Data Factory yang akan menyalin data dari setiap file .CSV individual (satu untuk setiap entitas) ke database Azure SQL yang sesuai (yang bervariasi menurut entitas bisnis). Nama tabel SQL dan skema di setiap database sama di semua database. Berikut adalah fragmen kode dari contoh pekerjaan Azure Data Factory untuk mengisi Azure SQL Database for WigiTech, berdasarkan file .CSV yang diekspor dari Azure Data Lake Store.

Layanan Tertaut: Sumber Azure Data Lake Store

```
{
  "name": "Source-DataLakeStore-f0g",
  "properties": {
    "hubName": "populateazuresqldbs_hub",
    "type": "AzureDataLakeStore",
    "typeProperties": {
      "dataLakeStoreUri":
        "https://brtadls.azuredatalakestore.net/webhdfs/v1",
      "authorization": "*****",
      "sessionId": "*****",
      "subscriptionId": "<Your Subscription ID>",
      "resourceGroupName": "brtwhdev"
    }
  }
}
```

Layanan Tertaut: Tujuan: Azure SQL Database

```
{
  "name": "Destination-SQLAzure-f0g",
  "properties": {
    "hubName": "populateazuresqldbs_hub",
    "type": "AzureSqlDatabase",
    "typeProperties": {
      "connectionString": "Data Source=brtsqlserverdev.database.
        windows.net;Initial Catalog=IoTDataMLHistory-WigiTech;Integrated
        Security=False;User ID=<Your User ID>;Password=<Your Password>;
        Connect Timeout=30;Encrypt=True"
    }
  }
}
```

Masukkan Himpunan Data: File .CSV di Azure Data Lake Store

```
{
  "name": "InputDataset-f0g",
  "properties": {
    "structure": [
      {
        "name": "Column0",
        "type": "String"
      },
      {
        "name": "Column1",
        "type": "Int64"
      },
      {
        "name": "Column2",
        "type": "Int64"
      },
      {
        "name": "Column3",
        "type": "Int64"
      },
      {
        "name": "Column4",
        "type": "Int64"
      },
      {
        "name": "Column5",
        "type": "Double"
      },
      {
        "name": "Column6",
```

```

        "type": "Double"
    },
    {
        "name": "Column7",
        "type": "Int64"
    },
    {
        "name": "Column8",
        "type": "Double"
    },
    {
        "name": "Column9",
        "type": "Int64"
    },
    {
        "name": "Column10",
        "type": "Double"
    },
    {
        "name": "Column11",
        "type": "Double"
    },
    {
        "name": "Column12",
        "type": "Int64"
    },
    {
        "name": "Column13",
        "type": "Int64"
    },
    {
        "name": "Column14",
        "type": "Int64"
    },
    {
        "name": "Column15",
        "type": "Int64"
    },
    {
        "name": "Column16",
        "type": "Int64"
    },
    {
        "name": "Column17",
        "type": "Int64"
    },
    {
        "name": "Column18",

```

```

        "type": "String"
    },
    {
        "name": "Column19",
        "type": "String"
    },
    {
        "name": "Column20",
        "type": "Int64"
    },
    {
        "name": "Column21",
        "type": "Double"
    },
    {
        "name": "Column22",
        "type": "Double"
    },
    {
        "name": "Column23",
        "type": "Datetime"
    },
    {
        "name": "Column24",
        "type": "Datetime"
    },
    {
        "name": "Column25",
        "type": "Int64"
    },
    {
        "name": "Column26",
        "type": "Datetime"
    },
    {
        "name": "Column27",
        "type": "String"
    },
    {
        "name": "Column28",
        "type": "String"
    },
    {
        "name": "Column29",
        "type": "String"
    },
    {
        "name": "Column30",

```

```

        "type": "Int64"
    },
    {
        "name": "Column31",
        "type": "Int64"
    },
    {
        "name": "Column32",
        "type": "String"
    },
    {
        "name": "Column33",
        "type": "Int64"
    },
    {
        "name": "Column34",
        "type": "String"
    },
    {
        "name": "Column35",
        "type": "Datetime"
    },
    {
        "name": "Column36",
        "type": "String"
    },
    {
        "name": "Column37",
        "type": "Double"
    },
    {
        "name": "Column38",
        "type": "Int64"
    },
    {
        "name": "Column39",
        "type": "String"
    },
    {
        "name": "Column40",
        "type": "String"
    }
],
"published": false,
"type": "AzureDataLakeStore",
"linkedServiceName": "Source-DataLakeStore-f0g",
"typeProperties": {
    "fileName": "StreamingDataMLHistory-WigiTech.Csv",

```

```

    "folderPath": "output/",
    "format": {
      "type": "TextFormat",
      "columnDelimiter": ",",
      "quoteChar": "\""
    }
  },
  "availability": {
    "frequency": "Day",
    "interval": 1
  },
  "external": true,
  "policy": {}
}

```

Himpunan Data Keluaran: Tabel Database Azure SQL

```

{
  "name": "OutputDataset-f0g",
  "properties": {
    "structure": [
      {
        "name": "UserId",
        "type": "String"
      },
      {
        "name": "Age",
        "type": "Double"
      },
      {
        "name": "Height",
        "type": "Double"
      },
      {
        "name": "Weight",
        "type": "Double"
      },
      {
        "name": "HeartRateBPM",
        "type": "Double"
      },
      {
        "name": "BreathingRate",
        "type": "Double"
      },
      {
        "name": "Temperature",

```

```

        "type": "Double"
    },
    {
        "name": "Steps",
        "type": "Double"
    },
    {
        "name": "Velocity",
        "type": "Double"
    },
    {
        "name": "Altitude",
        "type": "Double"
    },
    {
        "name": "Ventilization",
        "type": "Double"
    },
    {
        "name": "Activity",
        "type": "Double"
    },
    {
        "name": "Cadence",
        "type": "Double"
    },
    {
        "name": "Speed",
        "type": "Double"
    },
    {
        "name": "HIB",
        "type": "Double"
    },
    {
        "name": "HeartRateRedZone",
        "type": "Double"
    },
    {
        "name": "HeartrateVariability",
        "type": "Double"
    },
    {
        "name": "Status",
        "type": "Int32"
    },
    {
        "name": "Id",

```

```

    "type": "String"
  },
  {
    "name": "DeviceId",
    "type": "String"
  },
  {
    "name": "MessageType",
    "type": "Int32"
  },
  {
    "name": "Longitude",
    "type": "Double"
  },
  {
    "name": "Latitude",
    "type": "Double"
  },
  {
    "name": "Timestamp",
    "type": "Datetime"
  },
  {
    "name": "EventProcessedUtcTime",
    "type": "Datetime"
  },
  {
    "name": "PartitionId",
    "type": "Int32"
  },
  {
    "name": "EventEnqueuedUtcTime",
    "type": "Datetime"
  },
  {
    "name": "FirstName",
    "type": "String"
  },
  {
    "name": "LastName",
    "type": "String"
  },
  {
    "name": "UserName",
    "type": "String"
  },
  {
    "name": "UType",

```

```

        "type": "String"
    },
    {
        "name": "Phone",
        "type": "String"
    },
    {
        "name": "Email",
        "type": "String"
    },
    {
        "name": "Gender",
        "type": "String"
    },
    {
        "name": "MLUserid",
        "type": "String"
    },
    {
        "name": "MLTimestamp",
        "type": "String"
    },
    {
        "name": "ScoredLabels",
        "type": "String"
    },
    {
        "name": "ScoredProb",
        "type": "Double"
    },
    {
        "name": "Race",
        "type": "Int32"
    },
    {
        "name": "CompanyName",
        "type": "String"
    },
    {
        "name": "Rid",
        "type": "String"
    }
},
"published": false,
"type": "AzureSqlTable",
"linkedServiceName": "Destination-SQLAzure-f0g",
"typeProperties": {
    "tableName": "[dbo].[IOTDataMLHistory]"
}

```

```

    },
    "availability": {
      "frequency": "Day",
      "interval": 1
    },
    "external": false,
    "policy": {}
  }
}

```

Alur: Salin CSV ke Azure SQL Database

```

{
  "name": "CopyPipeline-f0g",
  "properties": {
    "activities": [
      {
        "type": "Copy",
        "typeProperties": {
          "source": {
            "type": "AzureDataLakeStoreSource",
            "recursive": true
          },
          "sink": {
            "type": "SqlSink",
            "writeBatchSize": 0,
            "writeBatchTimeout": "00:00:00"
          },
          "translator": {
            "type": "TabularTranslator",
            "columnMappings":
              "Column0:UserId,Column1:Age,Column2:Height,
              Column3:Weight,Column4:HeartRateBPM,Column5:
              BreathingRate,Column6:Temperature,Column7:St
              eps,Column8:Velocity,Column9:Altitude,Column
              10:Ventilization,Column11:Activity,Column12:
              Cadence,Column13:Speed,Column14:HIB,Column15
              :HeartRateRedZone,Column16:HeartrateVariabil
              ity,Column17:Status,Column18:Id,Column19:Dev
              iceId,Column20:MessageType,Column21:Longitud
              e,Column22:Latitude,Column23:Timestamp,Column
              n24:EventProcessedUtcTime,Column25:Partition
              Id,Column26:EventEnqueuedUtcTime,Column27:Fi
              rstName,Column28:LastName,Column29:UserName,
              Column30:UType,Column31:Phone,Column32:Email
              ,Column33:Gender,Column34:MLUserid,Column35:
              MLTimestamp,Column36:ScoredLabels,Column37:S

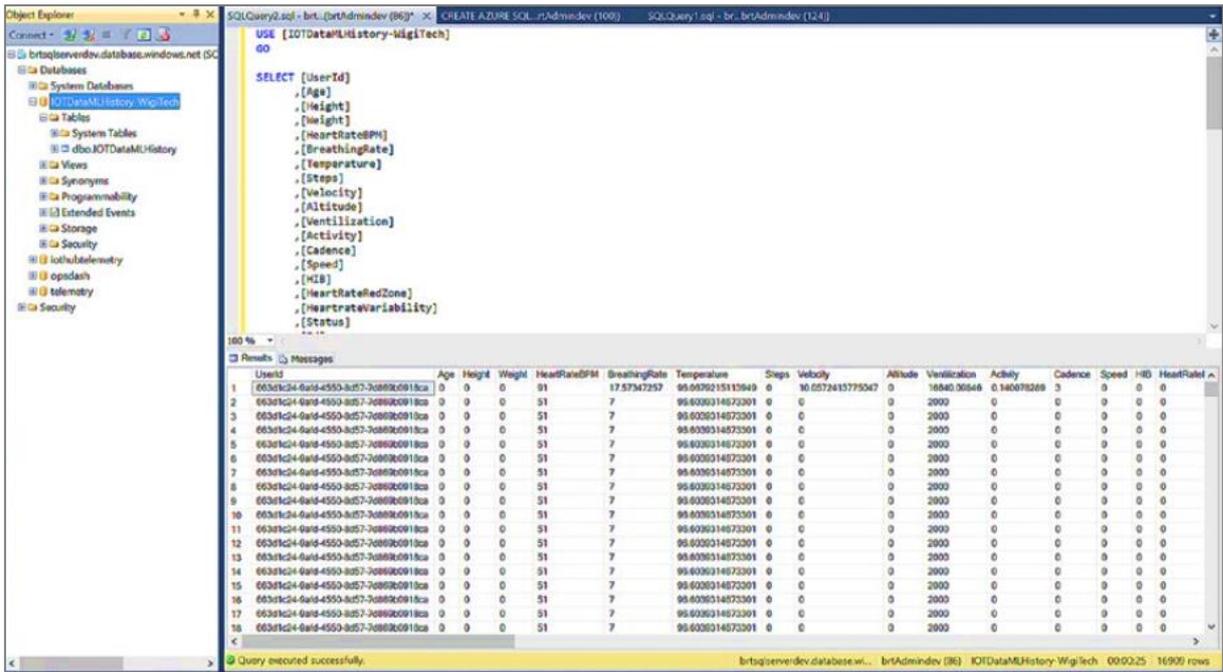
```

```

        coredProb,Column38:Race,Column39:CompanyName
        ,Column40:Rid"
    }
},
"inputs": [
    {
        "name": "InputDataset-f0g"
    }
],
"outputs": [
    {
        "name": "OutputDataset-f0g"
    }
],
"policy": {
    "timeout": "1.00:00:00",
    "concurrency": 1,
    "executionPriorityOrder": "NewestFirst",
    "style": "StartOfInterval",
    "retry": 3,
    "longRetry": 0,
    "longRetryInterval": "00:00:00"
},
"scheduler": {
    "frequency": "Day",
    "interval": 1
},
"name":"Activity-0-Datalakepath_output_[dbo]_[IOTDataMLHistory]"
}
],
"start": "2016-12-23T03:32:59.874Z",
"end": "2099-12-31T05:00:00Z",
"isPaused": false,
"hubName": "populateazuresqldbhub",
"pipelineMode": "Scheduled"
}
}

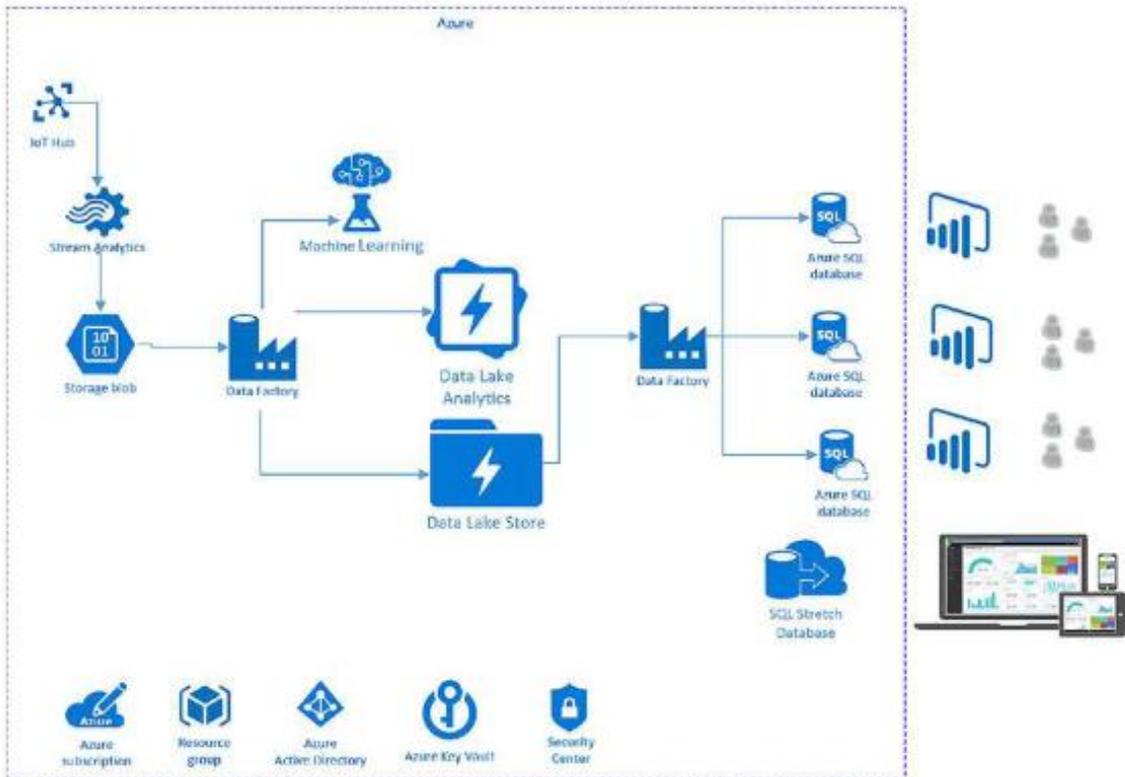
```

Setelah Anda selesai membuat pekerjaan Azure Data Factory baru untuk menyalin data file CSV ke database Azure SQL, Anda dapat dengan mudah memverifikasi hasilnya. Buka SQL Server Management Studio atau Visual Studio Server Explorer dan jalankan kueri SQL pada tabel SQL yang baru diisi. Gambar 7-26 mengilustrasikan hasil query SQL terhadap tabel SQL IOTDataMLHistory.



Gambar 7.26. Verifikasi impor database Azure SQL menggunakan kueri SQL di SQL Server Management Studio

Pada titik ini, Anda telah menyelesaikan semua tugas yang diperlukan untuk mengimplementasikan arsitektur Azure untuk implementasi referensi, seperti yang diilustrasikan pada Gambar 7.27.



Gambar 7.27. Arsitektur Azure untuk implementasi referensi

Langkah terakhir dalam proses implementasi produksi penuh ini adalah “mengunci” aspek keamanan solusi ini dengan membatasi pengguna dan akses pengguna ke Azure Data Lake Store dan database Azure SQL melalui Portal Azure.

7.7 RINGKASAN

Pada tahap konstruksi implementasi referensi ini, kami telah berhasil memperkenalkan penggunaan Azure Data Lake dan ADLA untuk menyediakan tambahan kemampuan pemrosesan dan analisis data batch berskala besar.

Salah satu aspek yang lebih unik dan memungkinkan dari aplikasi ADLA, yang dibangun sebagai solusi berbasis cloud, adalah Anda hanya membayar untuk sumber daya yang Anda gunakan. Ini murni pendekatan “model konsumsi”. Dalam kasus implementasi referensi, biaya penagihan Azure tambahan hanya berjumlah biaya untuk Data Lake Storage (penyimpanan yang digunakan) dan biaya ADLA (per pekerjaan yang dijalankan). Ini berarti Anda bisa saja fokus pada membangun dan menyempurnakan aplikasi data yang ada dan menyerahkan semua tugas penyediaan infrastruktur, penjadwalan pekerjaan, penerapan, penskalaan, dan pemantauan ke Microsoft Azure.

Untuk meringkas kemajuan sejauh ini, kami menyediakan kemampuan pengambilan data, analisis, dan pelaporan oleh entitas bisnis untuk membantu mencapai tujuan berikut:

- Menyediakan solusi pengarsipan dan analisis pemantauan kesehatan tim historis menggunakan Azure Data Lake.
- Membuat kumpulan data analisis Machine Learning yang komprehensif dan historis untuk membuat dan menyempurnakan prediksi kesehatan tim berdasarkan data sensor real-time aktual atau simulasi.
- Mengaktifkan skenario analisis regresi penuh untuk menganalisis dan “memutar ulang” peristiwa bersejarah.
- Menciptakan kemampuan untuk mengulang dan menyempurnakan algoritme Machine Learning pada data historis.
- Memberikan kemampuan “intelijen bisnis swalayan” yang aman dan terindividualisasi kepada setiap entitas bisnis untuk menghubungkan, melihat, dan mengekstrak data mereka. Power BI memudahkan penerbitan otomatis ke web atau perangkat seluler apa pun. Bab 9 mengeksplorasi topik ini lebih jauh.

Bab ini memberikan gambaran umum tingkat tinggi tentang Azure Data Lake Store (ADLS) dan Azure Data Lake Analytics (ADLA) dan mengeksplorasi beragam kemampuan pelengkap yang ditawarkan oleh kedua layanan. Fokus utama kedua layanan cloud ini jelas menangani Big Data dalam skala besar. Salah satu aspek yang lebih menarik adalah arsitektur fleksibel dan terbuka yang memungkinkan penyerapan semua jenis data melalui ekstraktor data dan pemformat keluaran khusus atau siap pakai. Beberapa proposisi nilai utama dibahas pada bagian berikut.

Menangani Hampir Semua Jenis Data

U-SQL adalah bahasa pilihan untuk membuat ADLA dan dapat menangani hampir semua skenario penyerapan data.

- Data tidak terstruktur, semi terstruktur, atau terstruktur.
- Tipe yang ditentukan pengguna khusus domain menggunakan C#.
- Kueri U-SQL melalui blob Data Lake dan Azure.
- Kueri Gabungan atas penyimpanan SQL Operasional dan SQL DW, menghilangkan kompleksitas dan persyaratan waktu pemrosesan dari operasi Ekstrak-Transform-Load (ETL) tradisional.

Produktif sejak Hari Pertama

Kombinasi Azure Data Lake Store, ADLA, dan U-SQL menjadikannya sangat mudah untuk menjadi produktif secara instan di lingkungan.

- Memberikan skala dan kinerja yang mudah tanpa perlu menyesuaikan atau mengonfigurasi lingkungan secara manual.
- Memberikan salah satu pengalaman pengembang terbaik sepanjang siklus pengembangan baik bagi pemula maupun ahli.
- Memungkinkan pengembang dengan mudah memanfaatkan keahlian yang ada dengan SQL dan .NET.
- ADLA menyediakan kemampuan ekstraksi, persiapan, dan pelaporan data yang mudah dan canggih.
- Konektor berbasis teks bawaan yang mudah digunakan untuk menyerap format data paling umum seperti tab dan file teks yang dibatasi koma.
- Model ekstensibilitas yang sederhana dan kaya untuk menambahkan transformasi data khusus pelanggan.

Tidak Ada Batasan untuk Skala

Azure Data Lake Store menawarkan kapasitas penyimpanan yang hampir tak terbatas dan lingkungan ADLA dibangun dari awal untuk menangani beberapa skenario pemrosesan data terdistribusi terbesar dan kompleks di industri saat ini.

- Menskalakan sesuai permintaan tanpa perlu mengubah kode.
- Secara otomatis memparalelkan pekerjaan U-SQL dan kode khusus.
- Dirancang untuk memproses data berukuran petabyte.

BAB 8

ANALISIS TINGKAT LANJUT MENGGUNAKAN PEMBELAJARAN MESIN DAN R

Bab ini mengeksplorasi dunia baru Machine Learning dan analisis prediktif yang menarik. Pembelajaran Mesin saat ini menjadi salah satu topik teknologi paling menarik di industri TI saat ini. Dengan alasan yang sangat bagus, karena digunakan di hampir setiap segmen industri besar dan vertikal saat ini. Kita benar-benar hidup di masa-masa yang menyenangkan karena tiga tren utama mulai menyatu dalam industri TI:

- Big Data dan Internet of Things (IoT)
- Komputasi awan dan penyimpanan berbasis awan yang murah
- Kemampuan intelijen bisnis

Ada yang berpendapat bahwa kombinasi kekuatan-kekuatan ini membantu mengantarkan Revolusi Industri keempat. Telah diprediksi bahwa kecerdasan buatan dan kemampuan Machine Learning (ML) akan dimasukkan ke dalam platform, aplikasi, dan layanan perangkat lunak yang jumlahnya semakin meningkat seiring dengan semakin dekatnya beberapa tahun mendatang. Kemampuan-kemampuan baru ini akan memungkinkan generasi baru profesional bisnis dan TI untuk memanfaatkan kecerdasan buatan dan kemampuan Pembelajaran Mesin, semuanya tanpa harus memahami secara pasti cara kerjanya.

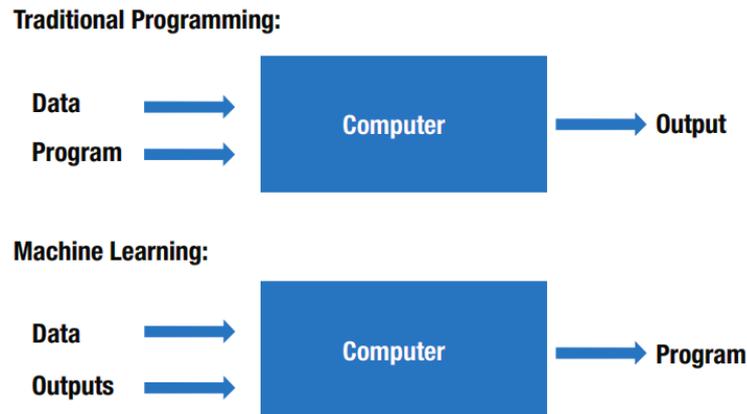
Hal ini menciptakan peluang besar bagi para pengembang dan ilmuwan data saat ini untuk membantu mewujudkan skenario dan kemampuan integrasi mendalam ini dengan penawaran aplikasi perangkat lunak saat ini dan masa depan.

8.1 APA ITU PEMBELAJARAN MESIN?

Pembelajaran Mesin dapat didefinisikan secara sederhana sebagai, “Sistem komputasi yang meningkat seiring dengan pengalaman”. Ada pepatah lama yang menangkap esensinya dengan cukup fasih dan ringkas “Masa Lalu Memprediksi Masa Depan”. Saat kita menjelajahi empat area utama algoritme Pembelajaran Mesin, Anda akan melihat bahwa tiga di antaranya didasarkan pada data historis, sehingga memberikan kepercayaan pada kutipan ini.

Memahami Pembelajaran Mesin

Untuk memahami cara kerja Machine Learning, ada baiknya membandingkan dan membedakannya dengan pendekatan tradisional dan menyoroti perbedaannya. Gambar 8.1 mengilustrasikan perbedaan antara pendekatan tradisional dan Pembelajaran Mesin.



Gambar 8.1 Pemrograman tradisional versus model pemrograman Pembelajaran Mesin

Dalam dunia pemrograman tradisional, pengembang menyediakan data dan mengembangkan program komputer untuk menghasilkan keluaran yang diinginkan. Di dunia baru Pembelajaran Mesin ini, fokusnya tidak hanya pada penyediaan data, tetapi juga pada pengayaan data dengan keluaran yang diketahui dan sudah ditentukan. Komputer kemudian menghitung program atau “model” Pembelajaran Mesin yang dapat memprediksi hasilnya, dalam rentang kepastian, berdasarkan data masukan.

Salah satu poin penting yang perlu ditekankan di sini adalah pentingnya persyaratan input data untuk keberhasilan pengembangan algoritma Machine Learning. Ungkapan “Data Sama dengan Pengalaman” adalah mantra pengoperasian baru di dunia Pembelajaran Mesin. Lebih banyak data selalu lebih baik, dan kebangkitan cloud, penyimpanan murah, Big Data, dan kekuatan komputasi yang ada di mana-mana, menjadikan periode dalam sejarah ini benar-benar memberdayakan teknologi Machine Learning dalam skala besar.

Hal ini membawa kita pada diskusi tentang dua jenis utama algoritma Machine Learning:

- *Pembelajaran yang Diawasi:* Dalam hal ini, hasil yang diketahui adalah bagian dari kumpulan data pelatihan komputer. Saat mengembangkan model, algoritme mengetahui secara pasti prediksi yang tepat, karena hasil yang diketahui tersebut merupakan bagian dari masukan yang diberikan.
- *Pembelajaran Tanpa Pengawasan:* Dalam hal ini, komputer hanya diberikan data untuk dianalisis, dan algoritme Pembelajaran Mesin kemudian berupaya menemukan pola, asosiasi, pengelompokan logis, dan hubungan dalam data.

Konsep-konsep ini diperkuat oleh empat kategori utama algoritma Machine Learning:

- * *Klasifikasi:* Berkaitan dengan prediksi apakah suatu elemen data masukan cocok dengan kategori atau jenis benda tertentu. Biasanya string atau label diprediksi. Penerapan umum dari teknik ini adalah klasifikasi biner, dimana hasilnya adalah salah satu dari dua nilai, misalnya, panas atau dingin, hidup atau mati, dll.
 - Ini mewakili metode pengajaran klasik “belajar melalui teladan”.
 - Contohnya adalah memprediksi musim apa yang akan terjadi pada tanggal tertentu dalam satu tahun (musim semi, musim panas, musim gugur, musim dingin).

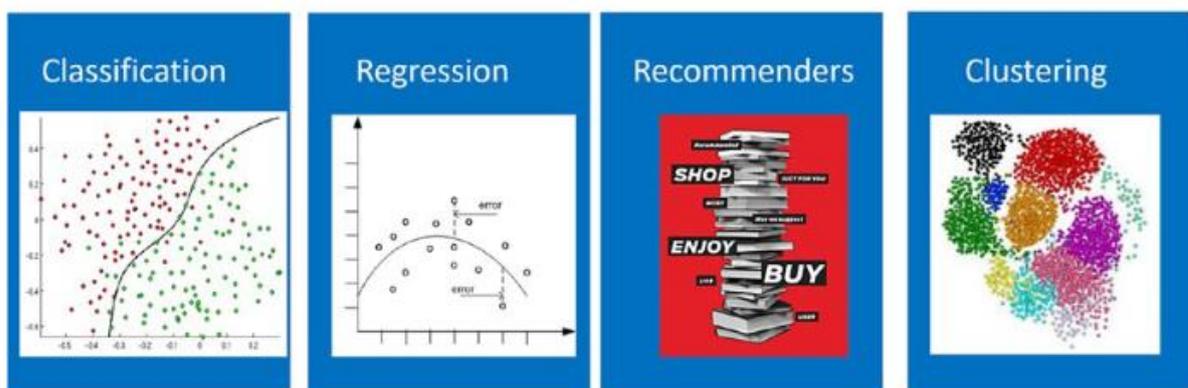
- Klasifikasi biner merujuk pada prediksi hasil yang hanya memiliki dua kemungkinan hasil.
 - Contohnya adalah memprediksi apakah sesuatu sedang Panas/Dingin atau Hidup/Mati.
 - Nantikan terus, karena kita akan menggunakan variasi algoritma ini dalam implementasi referensi di bab selanjutnya.
- ❖ *Regresi*: Mengacu pada prediksi hasil yang bernilai terus-menerus (“bilangan nyata”) berdasarkan berbagai atribut (dikenal sebagai “fitur”) yang diberikan sebagai masukan.
 - Contoh yang bagus dari algoritma ini adalah model yang memprediksi harga mobil baru berdasarkan merek, model, dan fiturnya.
- ❖ *Rekomendasi*: Ini adalah salah satu jenis algoritma Machine Learning yang paling umum digunakan saat ini. Algoritme tersebut memberikan rekomendasi berdasarkan riwayat yang diketahui, atribut, atau kesamaan fitur.
 - Beberapa contoh umum dari algoritma jenis ini adalah:
 - Situs web Amazon.com dan rekomendasi hasil personalisasi yang diberikan saat Anda berbelanja dan memasukkan barang ke dalam keranjang. Item yang ditampilkan di bagian berjudul “Orang yang menjelajahi item ini juga menjelajahi item ini...” adalah alat penjualan yang sangat hebat dan menyumbang 5-15% dari keseluruhan penjualan web mereka.
 - Netflix adalah contoh bagus lainnya dari penerapan algoritma Machine Learning yang direkomendasikan dan populer. Dengan menganalisis riwayat penayangan video sebelumnya, preferensi genre, rekomendasi anggota, layanan video online seperti Netflix dapat memberikan daftar rekomendasi film hasil personalisasi yang sangat akurat.
- ❖ *Clustering*: Algoritme jenis ini dapat menganalisis data dalam jumlah besar dan secara ringkas memberikan pengelompokan logis, analisis segmentasi, menyimpulkan pola, dan menandai penggambaran dalam data.
 - Ini adalah contoh algoritma Machine Learning tanpa pengawasan. Kategori algoritme ini memungkinkan pengguna untuk “melihat hutan dari balik pepohonan” jika Anda mau, sehingga pola dan korelasi dapat ditentukan pada data. Ini sangat berharga untuk kumpulan data yang sangat besar sehingga manusia tidak mungkin dapat memproses semuanya.
 - Salah satu contoh algoritma jenis ini adalah analisis sentimen “word cloud”, seperti yang diilustrasikan pada Gambar 8-2.
 - Dalam contoh ini, tag hash #microsoft diambil sampelnya dari feed Twitter langsung dan kemudian diproses oleh beberapa paket R ML untuk mengekstrak dan menganalisis kata kunci di setiap tweet.
 - Ini memberikan kata-kata yang muncul. Ukuran font menunjukkan seberapa sering setiap kata digunakan.

- Analisis sentimen juga dilakukan pada setiap tweet untuk menentukan apakah tweet tersebut positif atau negatif. Warnanya menunjukkan emosi yang diprediksi.



Gambar 8.2. Contoh algoritma pengelompokan awan kata Analisis sentimen Twitter #microsoft"

Bagi mereka yang mungkin baru mengenal bidang ilmu data yang menarik, kami telah membahas beberapa konsep utama seperti pembelajaran “yang diawasi” dan “tanpa pengawasan”, serta empat kategori utama algoritme Pembelajaran Mesin, seperti yang ditunjukkan pada Gambar 8 -3.



Gambar 8.3. Empat jenis utama algoritma Machine Learning

Memahami konsep dasar ini memberikan landasan untuk eksplorasi lebih dalam bidang ilmu data, Pembelajaran Mesin, kecerdasan buatan, dan analisis prediktif. Memahami cara kerja algoritme ini dan cara memanfaatkannya dengan sebaik-baiknya adalah langkah penting pertama bagi mereka yang memulai “transformasi digital” dan/atau perjalanan Big Data.

Sejarah Singkat Pembelajaran Mesin

Implementasi Machine Learning modern saat ini lebih selaras dengan bidang kecerdasan buatan. Penelitian awal yang dimulai pada tahun 1940-an sebagian besar berfokus pada kemampuan meniru perilaku manusia. Pada tahun 1943, Warren McCulloch dan Walter Pitts menciptakan model matematika awal untuk algoritma jaringan saraf yang disebut “logika ambang batas”. Penelitian awal ini mengarah pada fokus pada penerapan jaringan saraf untuk membantu menciptakan kemampuan “kecerdasan buatan”.

Pada tahun 1950an, Arthur Samuel dari IBM menciptakan salah satu program komputer pertama di dunia yang mampu memainkan catur melawan manusia. Salah satu bagian yang sangat menarik dari terobosan awal ini adalah Samuel menemukan bahwa algoritma komputer mampu “belajar” dari permainan sebelumnya dan dengan demikian meningkat seiring waktu dengan belajar dan beradaptasi dengan strategi pemain lawan.

Contoh yang bagus dapat ditemukan di Layanan Pos AS. Pada akhir tahun 1990-an, hanya sekitar 10% surat pos AS yang dapat disortir secara otomatis. Artinya, sisanya harus diproses secara manual oleh ribuan petugas pos. Kini mereka dapat memproses secara otomatis sekitar 98% surat di AS setiap hari menggunakan teknologi Optical Character Recognition (OCR) untuk memahami alamat tulisan tangan dan cetak dalam segala bentuk dan ukuran. Diperkirakan Layanan Pos AS kini memproses 500 juta surat unik setiap hari.

Dengan munculnya Internet dan World Wide Web, perusahaan e-commerce segera mulai memanfaatkan dan menyempurnakan Pembelajaran Mesin dan analisis prediktif untuk memengaruhi perilaku penelusuran dan pembelian web. Misalnya, analisis log web dapat memberikan banyak informasi tentang pola pencarian pelanggan, tingkat pengabaian pesanan, efektivitas periklanan, dan banyak lagi++. Seperti yang telah disebutkan sebelumnya, Amazon dan Netflix merupakan implementasi teknologi ini yang terkenal dan sangat sukses.

Aplikasi Industri Pembelajaran Mesin

Seperti yang dapat dilihat dari contohnya, keuntungan yang sangat besar muncul ketika memanfaatkan teknologi Machine Learning dan analisis prediktif. Potensi imbalan finansial, dikombinasikan dengan kebangkitan cloud, merupakan motivasi besar untuk mensponsori penelitian dan pengembangan berkelanjutan terhadap teknologi ini guna memaksimalkan efisiensi di semua jenis industri.

Berikut adalah daftar contoh dan kasus penggunaan umum teknologi Machine Learning yang diterapkan di beberapa industri utama saat ini, seperti ritel, jasa keuangan, layanan kesehatan, dan manufaktur:

- ❖ *Pengecer*
 - ✓ Perkiraan permintaan
 - ✓ Program loyalitas

- ✓ Cross-sell dan upsell
- ✓ Akuisisi pelanggan
- ✓ Deteksi penipuan
- ✓ Strategi penetapan harga
- ✓ Personalisasi
- ✓ Nilai pelanggan seumur hidup
- ✓ Segmentasi produk
- ✓ Demografi lokasi toko
- ✓ Manajemen rantai persediaan
- ✓ Manajemen persediaan
- ❖ *Jasa Keuangan*
 - ✓ Pergantian pelanggan
 - ✓ Program loyalitas
 - ✓ Cross-sell dan upsell
 - ✓ Akuisisi pelanggan
 - ✓ Deteksi penipuan
 - ✓ Risiko dan kepatuhan
 - ✓ Gagal bayar pinjaman
 - ✓ Personalisasi
 - ✓ Nilai pelanggan seumur hidup
 - ✓ Optimalisasi pusat panggilan
 - ✓ Bayar untuk kinerja
- ❖ *Kesehatan*
 - ✓ Optimasi bauran pemasaran
 - ✓ Akuisisi pasien
 - ✓ Deteksi penipuan
 - ✓ Pengumpulan tagihan
 - ✓ Kesehatan penduduk
 - ✓ Demografi pasien
 - ✓ Efisiensi operasional
 - ✓ Bayar untuk kinerja
- ❖ *Manufaktur*
 - ✓ Perkiraan permintaan
 - ✓ Optimasi bauran pemasaran
 - ✓ Strategi penetapan harga
 - ✓ Manajemen risiko kinerja
 - ✓ Optimalisasi rantai pasokan
 - ✓ Personalisasi
 - ✓ Pemantauan jarak jauh
 - ✓ Pemeliharaan prediktif
 - ✓ Manajemen aset

Pola Horizontal di Seluruh Industri Vertikal

Dalam daftar tersebut, Anda akan mencatat beberapa tema umum, pola, dan kasus penggunaan yang muncul. Daftar berikut menunjukkan beberapa skenario kasus penggunaan umum yang memiliki banyak penerapan praktis di sebagian besar industri vertikal:

- Perkiraan permintaan
- Program loyalitas
- Deteksi penipuan
- Personalisasi
- Nilai pelanggan seumur hidup

Dalam lingkungan bisnis modern saat ini, kini tersedia banyak paket perangkat lunak sumber terbuka, akademis, dan komersial yang memanfaatkan berbagai algoritme dan model Pembelajaran Mesin untuk membantu perusahaan meraih kesuksesan. Untuk itu, Microsoft telah merilis serangkaian templat Azure Machine Learning yang dapat dimanfaatkan di banyak skenario vertikal dan horizontal berikut:

- *Pemeliharaan Prediktif*: Memprediksi kegagalan mesin fisik berdasarkan riwayat masa lalu.
- *Customer Churn Prediction*: Memprediksi kapan terjadinya churn (kerugian) pelanggan.
- *Deteksi Penipuan Pembelian Online*: Memprediksi apakah suatu transaksi pembelian online mencurigakan atau curang.
- *Peramalan Ritel*: Menyediakan perkiraan penjualan produk untuk masing-masing toko ritel.
- *Klasifikasi Teks*: Klasifikasikan rekaman teks ke dalam kategori yang berbeda, misalnya untuk analisis sentimen. Templat Pembelajaran Mesin dengan Azure ML Studio: <https://gallery.cortanaintelligence.com/Collection/Machine-Learning-Templates-with-Azure-ML-Studio-1>.

Kita akan mengeksplorasi lebih banyak template, paket, pustaka, dan contoh ini di bab selanjutnya.

8.2 PEMBELAJARAN MESIN AZURE

Pada bulan Juli 2014, Microsoft pertama kali mempratinjau layanan Azure Machine Learning, layanan cloud terkelola sepenuhnya yang memungkinkan Anda dengan mudah membangun, menerapkan, dan berbagi solusi analitik prediktif. Pada saat itu, tujuannya sederhana namun visioner:

“Jadikan Machine Learning dapat diakses oleh setiap perusahaan, ilmuwan data, pengembang, pekerja informasi, konsumen, dan perangkat di mana pun di dunia.”

Dua tahun kemudian, Microsoft benar-benar membantu mendemokratisasi lanskap Machine Learning dengan menjadikan teknologi lebih mudah didekati dan diakses oleh semua orang. Yang Anda perlukan untuk memulai hanyalah browser web modern dan masalah yang

harus dipecahkan. Microsoft bahkan telah menyediakan versi gratis Azure Machine Learning Studio bagi pendatang baru untuk mencobanya.

Lanskap Ilmu Data Tradisional

Untuk lebih memahami dunia baru Pembelajaran Mesin berbasis cloud yang kini telah diaktifkan oleh Microsoft, ada gunanya mengatur konteksnya dengan memeriksa keadaan lanskap ilmu data sebelum penawaran cloud modern seperti Azure Machine Learning:

- *Mahal*: Biaya modal yang besar untuk peralatan, keahlian, serta kapasitas komputasi dan penyimpanan menciptakan hambatan masuk yang tidak perlu.
- *Silo Data*: Alat pengelolaan data yang rumit, data bisnis yang tertutup, dan akses terbatas ke data tersebut membatasi jumlah pembagian model dan kumpulan data prediktif.
- *Alat yang tidak terhubung*: Alat yang kompleks dan terfragmentasi membatasi partisipasi dalam mengeksplorasi data dan membangun model.
- *Kompleksitas penerapan*: Lingkungan khusus yang mendukung aplikasi ilmu data cukup kompleks dan memerlukan sumber daya yang sangat terlatih untuk mengelolanya.

Demokratisasi Pembelajaran Mesin

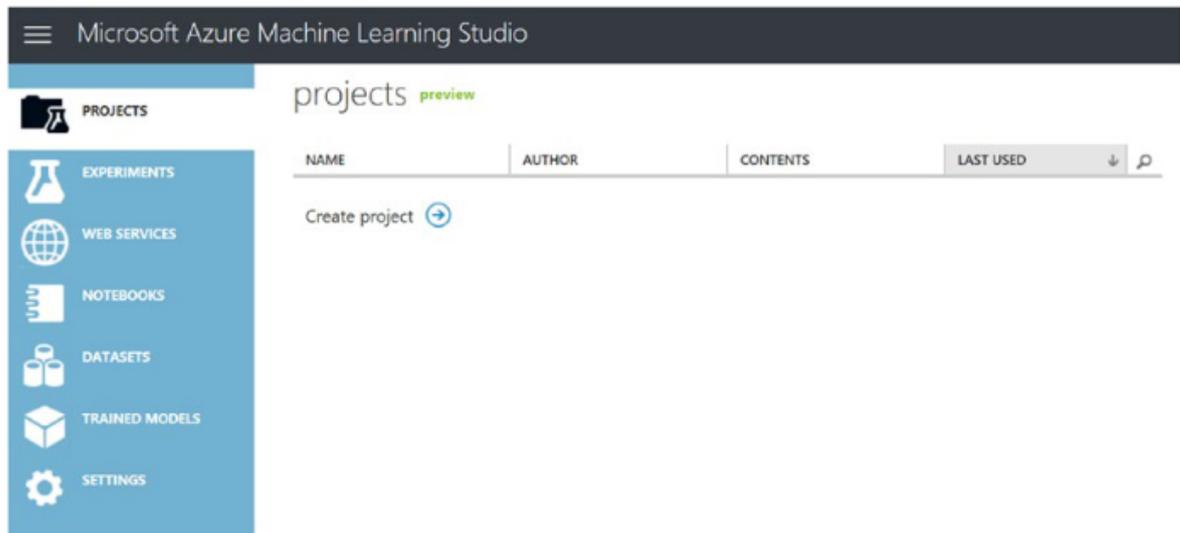
Berbeda dengan lanskap ilmu data tradisional yang diuraikan sebelumnya, tujuan Microsoft adalah menjadikan Machine Learning lebih sederhana, lebih baik, lebih cepat, dan lebih kolaboratif. Ruang kerja Machine Learning Studio Microsoft memungkinkan Anda mengundang dan berbagi eksperimen prediktif Anda dengan hampir semua orang di dunia.

- ❖ Panduan yang Diterbitkan: Melalui ikhtisar, dokumentasi, tutorial, dan panduan Azure Machine Learning online/offline.
- ❖ Galeri Cortana Analytics: Repositori sampel Azure Machine Learning dan templat solusi yang terus bertambah yang diproduksi oleh grup produk Microsoft dan anggota komunitas. Mereka dapat langsung disebarkan ke ruang kerja Azure Machine Learning untuk membantu Anda memulai dengan cepat.
- ❖ Lembar contekan algoritme Pembelajaran Mesin: Untuk membantu Anda memilih algoritme yang tepat untuk membuat model analitik prediktif, Microsoft menyediakan lembar contekan analitik prediktif yang bagus, seperti diagram alur solusi ML.
- ❖ Tingkat Gratis: Untuk memulai tanpa kewajiban. Sekarang, Anda tidak punya alasan untuk memulai.
- ❖ Proses Ilmu Data Tim (TDSP): Memberikan pendekatan dan kerangka kerja sistematis untuk membangun aplikasi cerdas yang memungkinkan tim ilmuwan data berkolaborasi secara efektif sepanjang siklus hidup solusi Pembelajaran Mesin.

Studio Pembelajaran Mesin Azure

Microsoft Azure Machine Learning Studio adalah lingkungan online, kolaboratif, seret dan lepas yang dapat Anda manfaatkan untuk membangun, menguji, dan menyebarkan solusi analitik prediktif. Meskipun Azure Machine Learning memudahkan pemula dan pendatang baru untuk mulai bekerja dengan analitik prediktif, Azure Machine Learning juga menyediakan serangkaian algoritme dan alat terintegrasi yang sangat canggih dan terkini yang bahkan akan dihargai oleh para profesional ilmu data paling berpengalaman sekalipun.

Setelah Anda memprovisikan ruang kerja Azure Machine Learning baru melalui Portal Azure, Anda dapat meluncurkan Azure Machine Learning Studio di sesi browser, seperti yang ditunjukkan pada Gambar 8.4.



Gambar 8.4 Studio Pembelajaran Mesin Azure

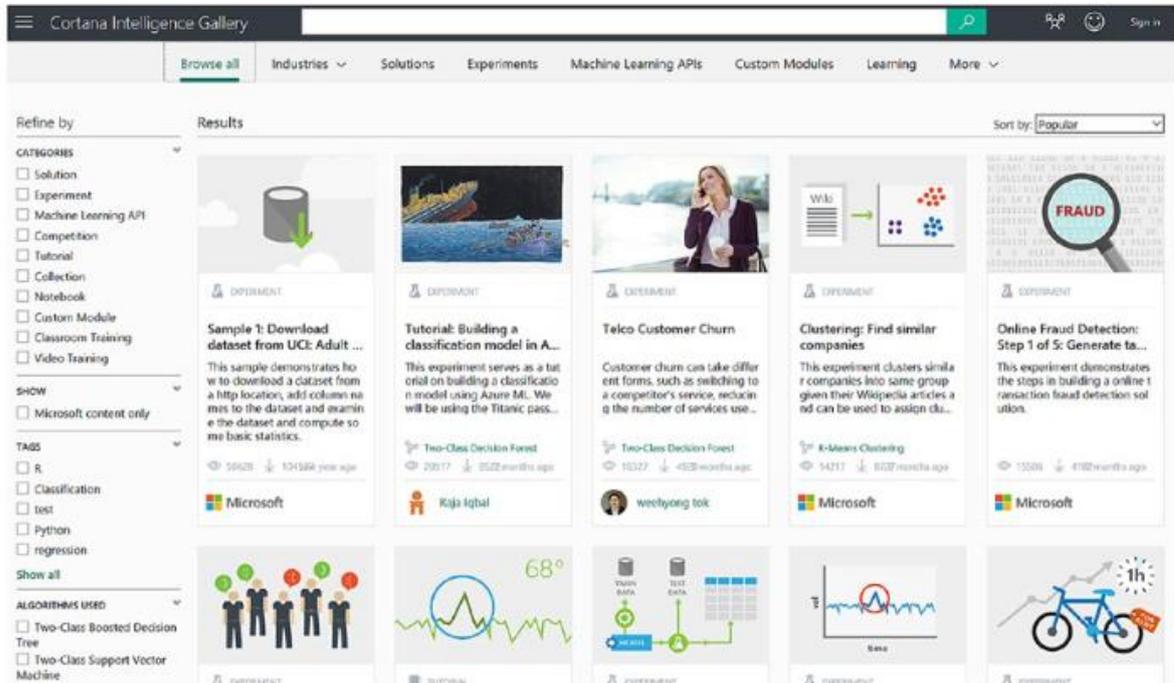
Setelah masuk ke Azure Machine Learning Studio, Anda akan melihat tab berikut di sebelah kiri:

1. **PROYEK:** Di sinilah Anda menyimpan kumpulan eksperimen, kumpulan data, buku catatan, dan sumber daya lain yang mewakili satu proyek Pembelajaran Mesin.
2. **EKSPERIMEN:** Di sinilah eksperimen yang telah dibuat, dijalankan, dan disimpan sebagai draf berada di ruang kerja Anda. Ini adalah opsi yang paling sering digunakan.
3. **LAYANAN WEB:** Layanan web Azure Machine Learning yang telah disebarkan dari eksperimen Anda dapat dilihat di sini.
4. **BUKU CATATAN:** Tautan ini menampilkan buku catatan Jupyter apa pun yang telah Anda buat. Jupyter Notebook adalah platform komputasi interaktif berbasis web yang menggabungkan kode langsung, persamaan, teks naratif, visualisasi, dan dasbor interaktif. Microsoft menyediakan lingkungan notebook Jupyter berbasis cloud di <https://notebooks.azure.com/>.
5. **SET DATA:** Tautan ini menunjuk ke himpunan data apa pun yang Anda unggah ke Azure Machine Learning Studio.
6. **MODEL TERLATIH:** Tautan ini menampilkan daftar model Machine Learning yang Anda latih dalam eksperimen dan disimpan di Azure Machine Learning Studio.
7. **PENGATURAN:** Kumpulan pengaturan yang dapat Anda gunakan untuk mengonfigurasi akun Azure Machine Learning dan sumber daya terkait.

Galeri Intelijen Cortana

Jika Anda mengklik ikon menu Hamburger di kiri atas bilah navigasi Azure Machine Learning Studio, Anda akan melihat tautan untuk Cortana Intelligence Gallery. Galeri (lihat

Gambar 8.5) adalah situs web tempat Microsoft membina komunitas ilmuwan data dan pengembang untuk berbagi solusi yang dibuat menggunakan Azure Machine Learning dan komponen Cortana Intelligence Suite.

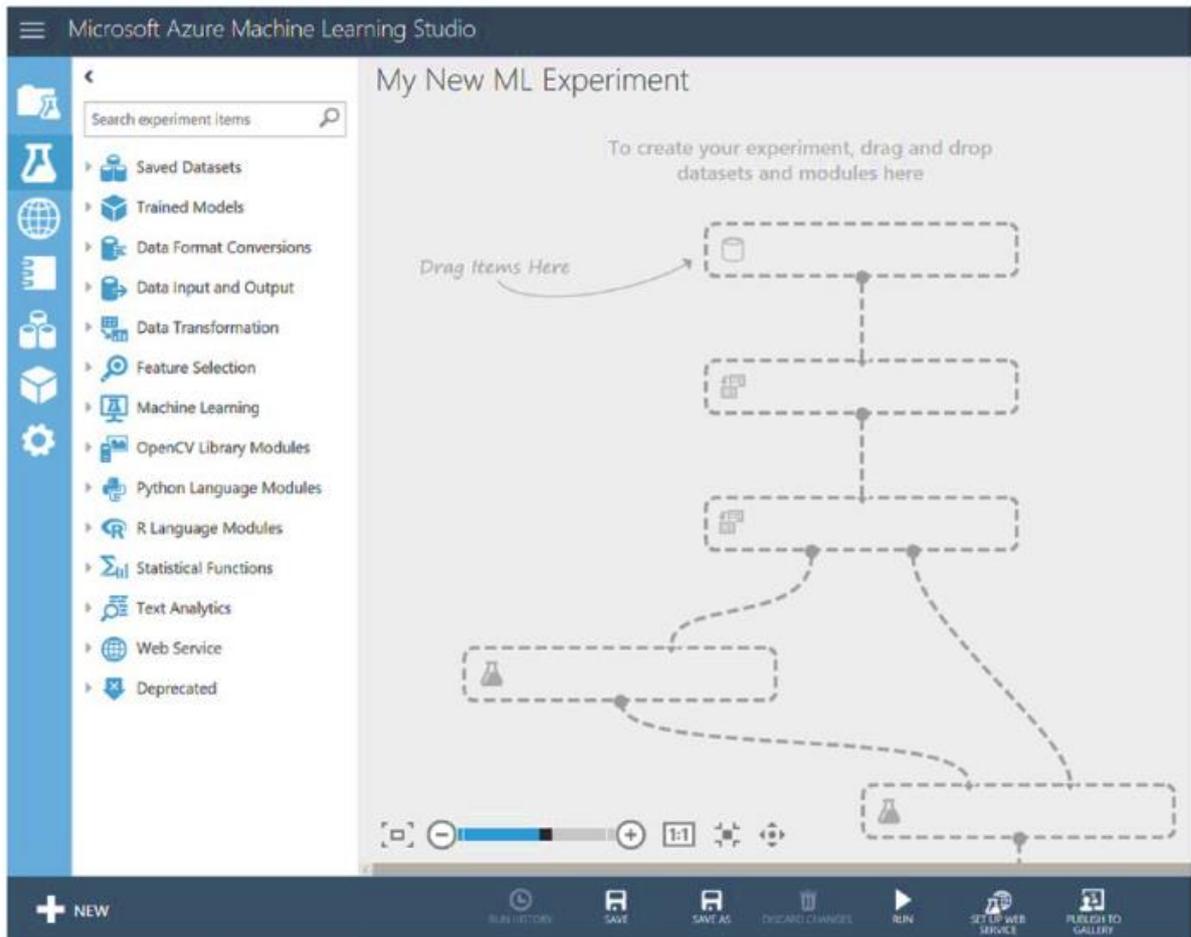


Gambar 8.5. Galeri Intelijen Cortana

Perhatikan bahwa ada tautan di bilah navigasi atas untuk sumber daya berharga lainnya seperti solusi industri, contoh eksperimen Pembelajaran Mesin, API Pembelajaran Mesin, modul khusus, dan banyak sumber daya pelatihan.

Eksperimen Pembelajaran Mesin Azure

Bagian Eksperimen di Azure Machine Learning Studio adalah tempat Anda akan menghabiskan sebagian besar waktu Anda mengembangkan dan menyempurnakan eksperimen Pembelajaran Mesin. Permukaan desainnya terasa seperti menggunakan Microsoft Visio, dan antarmukanya sepenuhnya digerakkan oleh drag-and-drop. Gambar 8.6 menggambarkan eksperimen Machine Learning baru dan kanvas Azure Machine Learning Studio Designer.



Gambar 8.6 Eksperimen Machine Learning baru dan kosong di Azure Machine Learning Studio

Modul Eksperimen Pembelajaran Mesin Azure

Ada serangkaian modul dan alat yang muncul di sisi kiri layar Azure ML Studio, seperti yang ditunjukkan pada Gambar 8-6. Modul-modul ini menyediakan banyak fungsi dan kemampuan yang dapat dengan mudah diseret ke permukaan desainer. Setelah menambahkan modul baru, Anda dapat mengatur properti konfigurasi individual melalui bagian Properti untuk setiap modul, yang muncul di sisi kanan layar.

Modul Azure Machine Learning dikelompokkan dalam kategori berikut:

- * * *Konversi Format Data:* Modul ini dapat membantu mengonversi data ke salah satu dari sejumlah format umum (seperti TSV, CSV, dan ARFF) yang digunakan oleh alat Machine Learning lainnya.
- * * *Input dan Output Data:* Gunakan modul ini untuk membaca data dari sumber data cloud dan web lainnya. Dimungkinkan juga untuk membaca kumpulan data “zip” dan mengekspor data menggunakan modul ini.
- * * *Transformasi Data:* Modul ini digunakan untuk membantu mempersiapkan data untuk analisis Machine Learning. Dengan modul ini, Anda dapat mengubah tipe data, mengidentifikasi dan/atau menghasilkan fitur, menormalkan data, dan banyak lagi.

- ✱✱ *Pemilihan Fitur*: Modul ini digunakan untuk membantu mengidentifikasi atribut atau fitur terbaik dalam data Anda, menggunakan metode statistik.
- ✱✱ *Machine Learning*: Grup ini berisi sebagian besar algoritma yang didukung oleh Azure Machine Learning.
 - Evaluasi: Setelah Anda melatih model, Anda dapat menggunakan modul Evaluasi untuk mengukur keakuratan model.
 - Inisialisasi: Modul ini menyediakan algoritma Machine Learning. Algoritme pada bagian ini dikelompokkan berdasarkan jenis berikut:
 - Deteksi anomali
 - Klasifikasi
 - Pengelompokan
 - Regresi
 - Skor: Modul ini digunakan untuk menghasilkan serangkaian hasil untuk mengevaluasi keakuratan model.
 - Latih: Modul ini digunakan untuk melatih model Machine Learning berdasarkan kumpulan data yang Anda berikan sebagai masukan.
- ✱✱ *Modul Perpustakaan OpenCV*: Modul ini menyediakan akses mudah ke perpustakaan sumber terbuka untuk pemrosesan gambar dan klasifikasi gambar. Istilah OCV adalah singkatan dari Open Computer Vision.
- ✱✱ *Modul Bahasa R*: Modul ini digunakan untuk memperluas eksperimen Pembelajaran Mesin dengan memanfaatkan kode R khusus.
- ✱✱ *Modul Pembelajaran Mesin Bahasa Python*: Modul ini digunakan untuk menambahkan kode Python khusus ke eksperimen Pembelajaran Mesin Anda.
- ✱✱ *Fungsi Statistik*: Modul ini digunakan untuk mengimplementasikan berbagai rutinitas statistik untuk menyelesaikan berbagai tugas analisis numerik.
- ✱✱ *Analisis Teks*: Modul ini dapat digunakan untuk memproses teks dan mengimplementasikan fungsi berbasis teks seperti hashing fitur dan pengenalan entitas bernama.
- ✱✱ *Layanan Web*: Modul ini dapat digunakan untuk menambahkan port input atau output ke layanan web Azure Machine Learning yang ada. Ini paling sering digunakan untuk membuat implementasi layanan web “pelatihan ulang” Machine Learning.

Eksperimen Azure Machine Learning terdiri dari himpunan data yang menyediakan data ke modul analitik, yang Anda sambungkan untuk membuat model analisis prediktif. Eksperimen Azure Machine Learning yang valid secara fungsional memiliki karakteristik berikut:

- Eksperimen memiliki setidaknya satu set data dan satu modul.
- Kumpulan data hanya dapat dihubungkan ke modul.
- Modul dapat dihubungkan ke kumpulan data atau modul lainnya.
- Semua port input untuk modul harus memiliki koneksi ke aliran data.
- Semua parameter yang diperlukan untuk setiap modul harus diatur melalui panel Properties di sisi kanan antarmuka desainer.

Kami memeriksa proses pembuatan eksperimen Azure Machine Learning secara mendetail di bagian akhir bab ini ketika kami membahas secara spesifik implementasi referensi.

Alur Ilmu Data Azure Machine Learning

Microsoft memberikan panduan komprehensif dan pendekatan sistematis untuk membangun aplikasi Pembelajaran Mesin yang mengikuti metodologi dan prinsip ilmu data yang sudah ada. Ini dapat diringkas sebagai proses lima langkah:

1. Pemahaman Bisnis:
 - a. Menetapkan kebutuhan/persyaratan teknis
 - b. Mengidentifikasi skenario Anda untuk mencocokkan dengan algoritma ML terbaik
2. Akuisisi dan Eksplorasi Data:
 - a. Memuat data ke lingkungan penyimpanan Azure
 - b. Mengimpor data ke Azure Machine Learning Studio
3. Mempersiapkan Data:
 - a. Menjelajahi data menggunakan alat dan utilitas bawaan
 - b. Memanfaatkan data sampel
4. Pemodelan:
 - a. Fitur teknik
 - b. Memilih fitur
 - c. Belajar dengan berhitung
 - d. Melatih model
 - e. Mengevaluasi model
 - f. Menyetel modelnya
5. Penerapan:
 - a. Menerbitkan model sebagai layanan web
 - b. Mengonsumsi model secara terprogram
 - c. Mengonsumsi model di Excel

8.3 IKHTISAR SERVER MICROSOFT R

Pada bulan April 2015, Microsoft mengakuisisi Revolution Analytics dan sejak itu merilis produk andalan mereka sebagai R Server. Revolution Analytics adalah penyedia perangkat lunak dan layanan komersial terkemuka untuk R, bahasa pemrograman yang paling banyak digunakan di dunia untuk komputasi statistik dan analisis prediktif.

Saat ini, bahasa pemrograman R adalah salah satu bahasa pemrograman statistik dan Pembelajaran Mesin terpopuler yang tersedia bagi pengembang dan ilmuwan data di seluruh dunia. Bahasa R tersedia secara gratis sebagai sumber terbuka dan populer sebagai alat visualisasi dan pelaporan data. Terdapat komunitas R yang kuat dan dinamis dengan sekitar 2,5 juta pengguna. Bahasa R umumnya diajarkan di sebagian besar universitas, dan minat terhadap penggunaan R ditunjukkan oleh banyak kelompok pengguna yang berkembang pesat di seluruh dunia.

Kemampuan R diperluas melalui paket yang dikembangkan pengguna, yang dapat menambahkan kemampuan dan ekstensi statistik atau grafis khusus. Paket-paket ini

dikembangkan terutama dalam R tetapi juga dalam bahasa lain seperti Java, C, dan C++. Hal ini telah menciptakan ekosistem yang sangat besar dan sehat dengan lebih dari 9.000 paket R yang dikontribusikan ke repositori open source hingga saat ini. Banyak dari paket dan implementasi ini mengatasi masalah bisnis umum dan dapat membantu memecahkan banyak kasus penggunaan yang terkait dengan analisis statistik dan prediktif.

Batasan pemrosesan Open Source R

Perlu diperhatikan bahwa ada dua jenis R yang tersedia saat ini untuk memenuhi kebutuhan spesifik dalam industri ilmu data.

- *Open Source R*: Distribusi gratis dan tersedia secara luas dari sumber seperti CRAN (Comprehensive R Archive Network) dan r-project.org. Perangkat lunak ini umumnya ditujukan untuk penggunaan akademis dan didukung oleh komunitas.
- *Paket R Komersial*: Biasanya dikembangkan, dilisensikan, dan didukung secara profesional untuk penggunaan perusahaan dalam skenario produksi.

Ada beberapa perbedaan signifikan antara distribusi R ini dalam hal kecepatan dan kinerja.

Open Source R memiliki beberapa keterbatasan seperti:

- ✱✱ Pengoperasian terbatas pada memori saja.
- ✱✱ Perpindahan/duplikasi data itu mahal.
- ✱✱ Kurangnya paralelisme; mereka hanya berulir tunggal.
- ✱✱ Bukan tingkat perusahaan:
 - Tidak ada SLA atau model pendukung produksi.
 - Kurangnya jaminan ketepatan waktu dukungan.
 - Dukungan masyarakat tidak memadai untuk perusahaan komersial.

Masuk ke Microsoft R Server

Microsoft R Server adalah server tingkat perusahaan untuk menghosting dan mengelola beban kerja proses R paralel dan terdistribusi di server (Linux dan Windows) dan cluster (Hadoop dan Apache Spark). Ini menyediakan mesin eksekusi untuk solusi yang dibangun menggunakan paket Microsoft R dan memperluas R sumber terbuka dengan dukungan untuk analitik kinerja tinggi, analisis statistik, skenario Pembelajaran Mesin, dan kumpulan data yang sangat besar.

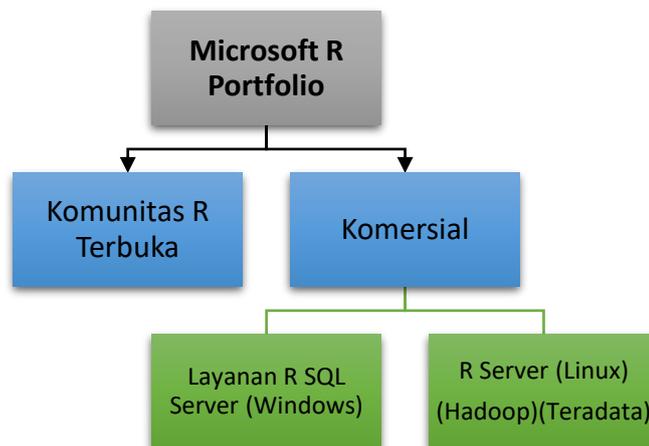
Untuk mengimbangi beberapa keterbatasan open source R yang diuraikan sebelumnya, Microsoft telah mengisi kesenjangan tersebut dengan penawaran Microsoft R Server untuk menyediakan penawaran produk perangkat lunak R tingkat perusahaan, tersedia secara komersial, yang menawarkan fitur-fitur berikut:

- Kecepatan, skalabilitas, fleksibilitas, dan efisiensi
- Dukungan produk dan SLA
- Bekerja pada data di memori atau pada disk (ScaleR)
- Tersedia beragam fungsi R yang dapat diskalakan dan terdistribusi
- Bekerja dalam beberapa konteks komputasi (termasuk Hadoop, Spark, dan SQL Server), dan sumber data (termasuk disk, HDFS, dan SQL)
- Bahasa R bersifat portabel dan mencakup jaminan investasi

Selain memperluas Azure Machine Learning melalui modul bahasa R, keluarga Microsoft R Server memungkinkan Anda mengembangkan dan menjalankan model R pada platform pilihan Anda:

- ❖ Windows: R Server untuk Windows dikirimkan sebagai Layanan R di SQL Server 2016.
- ❖ Linux: Manfaatkan investasi sumber terbuka Anda untuk mengaktifkan skenario kasus penggunaan analisis prediktif dan preskriptif tingkat lanjut.
- ❖ Hadoop/Apache Spark: Memungkinkan analisis Anda untuk diskalakan secara transparan dengan mendistribusikan tugas analitik untuk dijalankan di seluruh node tanpa pemrograman yang rumit.
- ❖ Teradata Database: Jalankan analitik tingkat lanjut dalam database untuk analisis data R yang lancar.

Singkatnya, Microsoft memiliki portofolio komprehensif penawaran pemrograman R (komunitas dan komersial) yang dapat berjalan di hampir semua sistem operasi atau platform. Gambar 8-7 mengilustrasikan berbagai versi R yang kini didukung Microsoft.

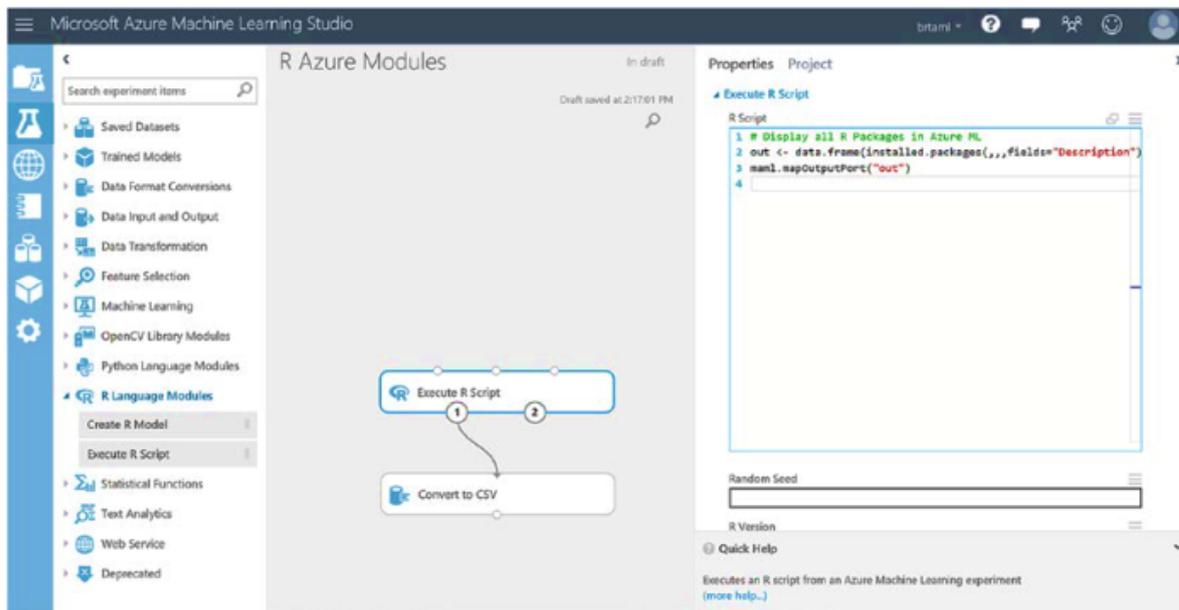


Gambar 8.7. Portofolio produk Microsoft R

Dengan memanfaatkan lingkungan R yang didukung Microsoft yang diilustrasikan pada Gambar 8.7, kini Anda dapat dengan mudah mengembangkan dan menjalankan model R pada platform pilihan Anda di Azure.

8.4 PERLUAS EKSPERIMEN PEMBELAJARAN MESIN DENGAN MODUL BAHASA R

Dengan memanfaatkan modul Bahasa R di perancang Eksperimen Azure Machine Learning Studio, Anda dapat langsung memperluas kemampuan pembuatan dan pelatihan model Machine Learning dengan memasukkan kode R langsung ke dalam eksperimen Azure Machine Learning. Anda cukup menarik dan melepas modul dari palet alat kiri di ML Studio ke permukaan desain. Kemudian Anda dapat mengklik jendela Properties di sebelah kanan, edit kode R Anda di Designer Editor, dan simpan setelah selesai. Gambar 8.8 menggambarkan tangkapan layar Azure Machine Learning Studio Designer dengan modul Execute R Script. Perhatikan bahwa kode R dapat diedit di panel Properties di sebelah kanan.



Gambar 8.8 Modul bahasa Azure Machine Learning R

Dalam contoh Jalankan Skrip R, kode R hanya akan mencantumkan paket R yang saat ini diinstal melalui pernyataan R ini:

```
out <- data.frame(installed.packages(,,,fields="Description"))
maml.mapOutputPort("out")
```

Modul percobaan berikutnya, modul Konversi ke CSV, menyediakan kemampuan untuk mengunduh dan menyimpan salinan daftar paket R tersebut. Perhatikan bahwa untuk upaya pengembangan R yang signifikan, kami menyarankan Anda menggunakan alat Pengembang R lainnya, seperti Alat Visual Studio untuk R atau R Studio untuk membuat, men-debug, membuat versi, dan menyempurnakan skrip R yang lebih kompleks. Kemudian Anda cukup menempelkan kode R ke modul bahasa R Azure Machine Learning.

Modul bahasa Azure Machine Learning R membuka dunia baru kode R, paket, dan fungsionalitas tambahan yang kini dapat diakses melalui Azure Machine Learning Studio dan disebarkan sebagai layanan web agar mudah dikonsumsi. Azure Machine Learning Studio saat ini mencakup lebih dari 400 paket R paling populer yang mewakili titik temu ekosistem R dengan aspek ketangkasan, kolaborasi, konsumsi, dan integrasi Azure Machine Learning.

Penggunaan R menjadi sangat populer sehingga Microsoft kini memasukkannya sebagai kontrol visualisasi di Power BI. Kami mengeksplorasi kemampuan baru ini dan banyak lagi di Bab 9.

Alat R untuk Visual Studio

Jika Anda seorang pengembang dan memiliki akses ke versi Visual Studio, Anda dapat dengan mudah mengubah Visual Studio menjadi lingkungan pengembangan R yang kuat dengan menginstal R Tools untuk Visual Studio. Ada juga unduhan gratis dan penawaran uji coba untuk Visual Studio yang tersedia di tautan ini:

R Tools for Visual Studio: <https://www.visualstudio.com/vs/rtvs/>

Menerapkan Azure Machine Learning dan R dalam Implementasi Referensi. Hingga saat ini, kami telah menjelajahi beberapa fitur dan kemampuan dasar Azure Machine Learning dan memberikan latar belakang singkat tentang bahasa pemrograman R dan penawaran R terbaru dari Microsoft. Selanjutnya, kami memanfaatkan beberapa pengetahuan yang baru kami peroleh dengan menerapkan beberapa implementasi referensi lagi.

Kasus Bisnis untuk Pembelajaran Mesin

Sebagai penyegaran singkat, kita perlu menerapkan layanan web Machine Learning yang dapat membantu memprediksi kapan anggota tim tertentu mungkin telah mencapai titik kelelahan fisik. Masukan akan didasarkan pada pembacaan sensor real-time yang diteruskan ke cloud Azure saat anggota tim sedang mengerjakan pekerjaan.

Tujuannya adalah untuk menghindari segala jenis kecelakaan, cedera, atau insiden yang mungkin disebabkan oleh anggota tim yang mencapai batas kelelahan fisik masing-masing. Dengan cara ini, badan usaha dapat secara efektif memantau operasi sehari-hari mereka dan secara proaktif menyesuaikan beban kerja dan tugas anggota tim untuk memitigasi risiko.

Referensi Implementasi: Asumsi

Kami membuat beberapa asumsi praktis untuk memenuhi syarat model prediksi yang berfungsi, sebagai berikut:

- Anggota tim mungkin enggan mengakui tanda-tanda kelelahan fisik: Ini adalah asumsi kerja yang mendasar dan dapat dianggap sebagai reaksi umum berdasarkan sifat manusia.
- Anggota tim mungkin mengambil risiko imbalan finansial jika mereka menunjukkan tanda-tanda kelelahan fisik: Karena mereka diberi kompensasi finansial berdasarkan jumlah jam kerja mereka, hanya sedikit orang yang mau mengakui bahwa mereka sudah merasa lelah dan ingin berhenti.

Untuk mengatasi kecenderungan alami manusia ini, kami berasumsi bahwa stress test akan diberikan kepada setiap anggota tim setiap bulan.

- ❖ Stress test akan dapat dengan cepat mensimulasikan dan mempercepat kondisi kerja fisik anggota tim, hingga mencapai titik kelelahan.
- ❖ Kelelahan fisik ditandai ketika anggota tim secara sukarela menekan tombol Stop darurat untuk menghentikan stress test.
- ❖ Data sensor terperinci akan diambil selama seluruh sesi stress test. Data ini kemudian akan digunakan untuk membuat data pelatihan untuk model prediktif Machine Learning kami yang baru.
- ❖ Model Machine Learning yang dihasilkan juga dapat dilatih ulang secara berkala dengan menggunakan hasil stress test yang diperbarui.
- ❖ Format data yang masuk untuk stress test akan berupa spreadsheet Excel dengan tab terpisah yang berisi hasil stress test masing-masing anggota tim.

- ❖ Selain kolom yang menyimpan pembacaan data sensor, kolom data tambahan akan ditambahkan di bagian akhir untuk menunjukkan kapan anggota tim memicu titik kelelahan fisik. Gambar 8.9 menggambarkan tangkapan layar hasil spreadsheet stress test anggota tim.

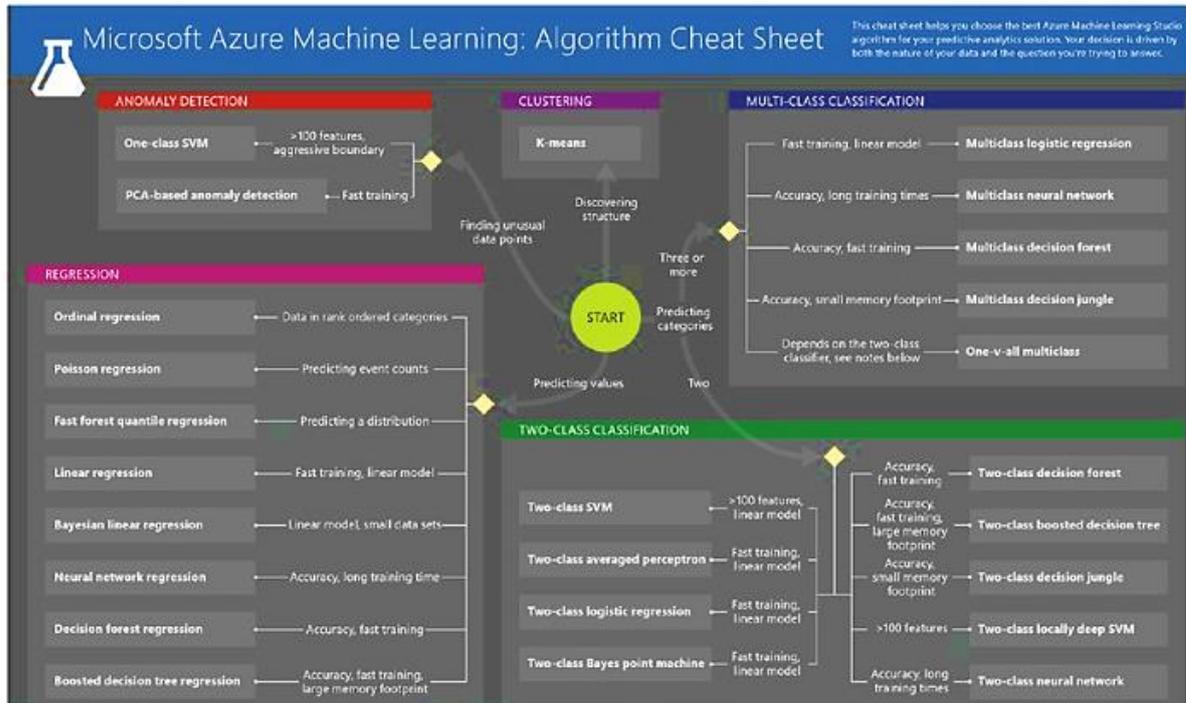
1	UserId	BreathingRate	Temperature	Ventilation	Activity	HeartRateBPM	Cadence	Velocity	Speed	HIB	HeartRateRedZone	HeartRateVariability	Exhaustion
2	c3ec2fe4-6789-4d3e-8446-852856ce025c	17.44929379	96.29215605	16454.15282	0.133564	90	1	2.895897	0	0	0	0	N
3	c3ec2fe4-6789-4d3e-8446-852856ce025c	17.51138318	96.88186507	16547.12564	0.136821	91	2	5.797794	0	0	0	0	N
4	c3ec2fe4-6789-4d3e-8446-852856ce025c	17.57347257	96.5117878	16640.09846	0.140078	91	3	8.696691	0	0	0	0	N
5	c3ec2fe4-6789-4d3e-8446-852856ce025c	17.63556196	96.31613514	16733.07128	0.143336	92	4	11.59559	0	0	0	0	N
6	c3ec2fe4-6789-4d3e-8446-852856ce025c	17.69765135	96.69531139	16826.0441	0.146593	92	5	14.49448	0	0	0	0	N
7	c3ec2fe4-6789-4d3e-8446-852856ce025c	17.75974074	96.77480951	16919.01692	0.14985	93	6	17.39338	0	0	0	0	N
8	c3ec2fe4-6789-4d3e-8446-852856ce025c	17.82183013	96.71956696	17011.98974	0.153107	93	7	20.29228	0	0	0	0	N
9	c3ec2fe4-6789-4d3e-8446-852856ce025c	17.88391952	96.35834845	17104.96256	0.156365	94	8	23.19118	0	0	0	0	N

Gambar 8.9. Contoh spreadsheet uji stres anggota tim yang digunakan untuk melatih model Azure Machine Learning

Memilih Algoritma Pembelajaran Mesin

Salah satu tugas pertama dalam upaya ini adalah menentukan algoritme Pembelajaran Mesin yang tepat untuk digunakan guna membantu memecahkan masalah ini. Untungnya, Microsoft menyediakan lembar contekan yang berguna (dalam bentuk unduhan PDF gratis) untuk membantu Anda menentukan algoritma terbaik untuk digunakan berdasarkan jenis masalah yang ingin Anda selesaikan.

Gambar 8.10 menggambarkan tangkapan layar lembar contekan Azure Machine Learning. Mengingat masalah sebenarnya yang kami coba selesaikan dalam implementasi referensi kami, kami akan mencoba memprediksi kategori berdasarkan lembar contekan.



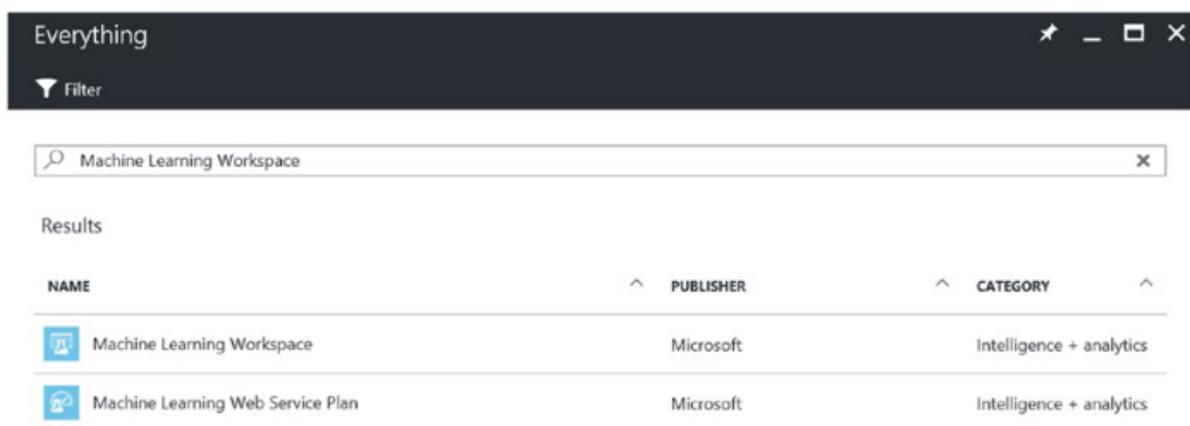
Gambar 8.10. Lembar contekan algoritma Azure Machine Learning

Dalam hal ini, hasil spesifik yang ingin kita prediksi adalah kelelahan, dan terdapat dua kondisi potensial: habis dan tidak habis. Anda akan mengingat kembali bab sebelumnya bahwa algoritme klasifikasi biner berupaya memprediksi salah satu dari dua kemungkinan hasil seperti hidup/mati, merah/biru, atau panas/dingin. Memang benar, berdasarkan panduan contekan, kami disarankan untuk menggunakan algoritma klasifikasi biner untuk menyediakan model Pembelajaran Mesin ini.

Hal lain yang perlu disampaikan di sini adalah bahwa model klasifikasi biner adalah contoh pembelajaran yang diawasi. Artinya, kami menyediakan data pelatihan anggota tim beserta hasil yang diamati (habis atau tidak) untuk setiap baris data pelatihan. Dengan cara ini, setelah model Machine Learning baru kami dilatih, model tersebut akan dapat memprediksi hasil (bersama dengan persentase probabilitas) dari serangkaian atribut (atau fitur) masukan.

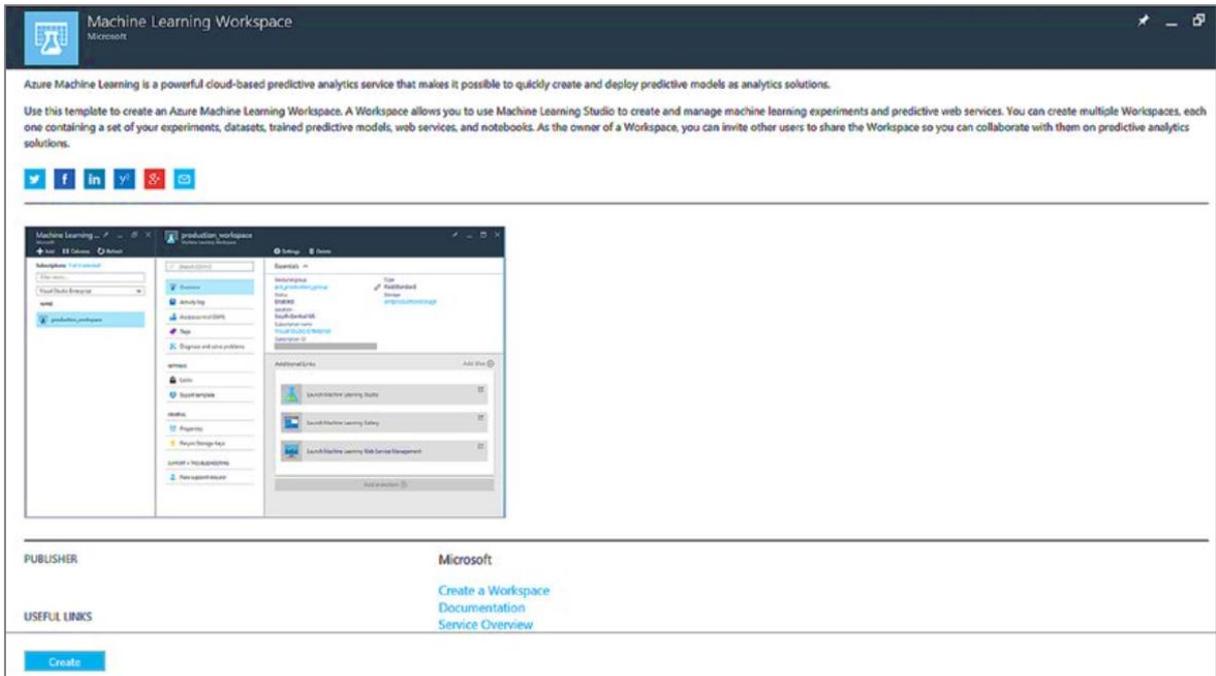
Buat Ruang Kerja Pembelajaran Mesin Azure

Langkah pertama untuk mewujudkan implementasi referensi kami adalah membuat ruang kerja Azure Machine Learning melalui Portal Azure. Ini akan menjadi titik awal untuk membuat eksperimen Machine Learning dan pada akhirnya menerapkannya sebagai layanan web. Untuk memulai, navigasikan ke grup sumber daya Azure yang Anda tentukan sebelumnya dan klik ikon + Tambahkan untuk menambahkan layanan Azure lainnya. Ketik Machine Learning Workspace di bilah pencarian; Anda akan melihat hasil serupa seperti yang ditunjukkan pada Gambar 8.11.



Gambar 8.11. Tambahkan ruang kerja Machine Learning baru

Klik pada Ruang Kerja Machine Learning. Anda kemudian akan melihat layar konfirmasi seperti pada Gambar 8.12.

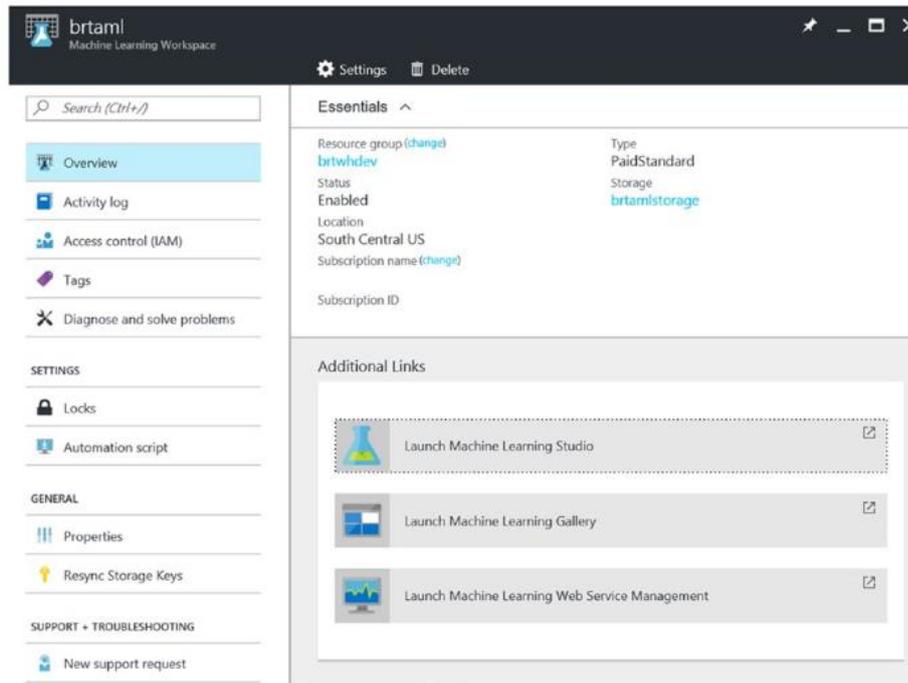


Gambar 8.12. Buat Ruang Kerja Azure Machine Learning

Klik pada ikon Buat di kiri bawah. Anda kemudian akan melihat bilah muncul yang meminta Anda memasukkan parameter yang diperlukan untuk membuat Ruang Kerja Azure Machine Learning baru Anda, seperti yang diilustrasikan dalam Gambar 8-13.

Gambar 8.13. Azure Machine Learning: Membuat parameter Ruang Kerja Baru

Isi parameter yang diperlukan dan kemudian klik ikon Buat. Ruang Kerja Azure Machine Learning yang baru akan disediakan dalam hitungan menit. Setelah ruang kerja baru Anda disediakan, Anda dapat menavigasi ke blade ruang kerja Machine Learning baru dan melihat opsi menu, seperti yang ditunjukkan pada Gambar 8-14.

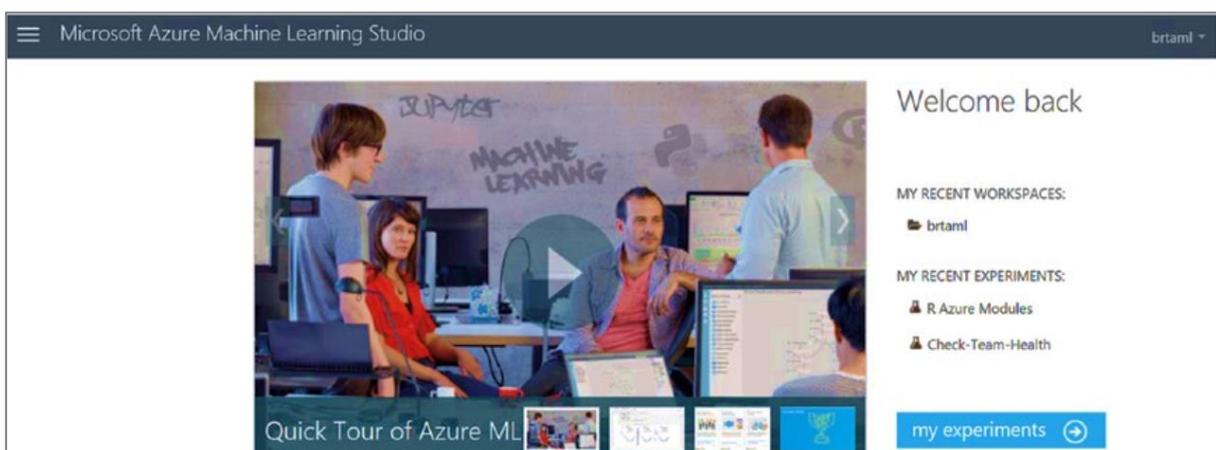


Gambar 8.14. Bilah Ruang Kerja Azure Machine Learning

Pilih opsi untuk Meluncurkan Machine Learning Studio, seperti yang ditampilkan pada Gambar 8-14. Selanjutnya kami akan mulai membuat eksperimen Machine Learning baru.

Buat Eksperimen Pembelajaran Mesin Azure Baru

Setelah Anda mengklik ikon Luncurkan Machine Learning Studio, sesi baru akan terbuka di tab browser baru, seperti yang diilustrasikan pada Gambar 8-15.

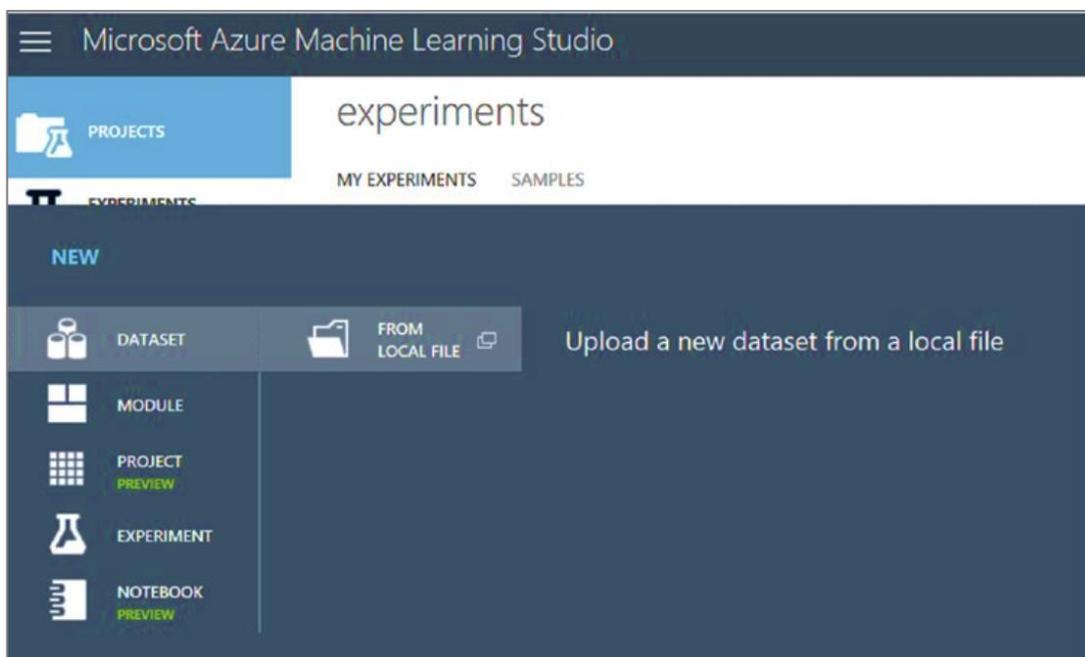


Gambar 8-15. Halaman arahan Azure Machine Learning Studio

Klik tombol Eksperimen Saya dan Anda akan diarahkan ke lingkungan Azure Machine Learning Studio. Sekarang Anda siap membuat eksperimen Machine Learning baru. Pertama, Anda perlu memuat data pelatihan untuk model baru ini. Dari basis kode buku ini (diterbitkan di GitHub), unduh file Excel bernama *Teammates_AML_Training_Data.xlsx*. Setelah file diunduh, buka spreadsheet dan klik tab *Teammates_AML_Training_Data*. Selanjutnya simpan spreadsheet sebagai file dalam format CSV dengan cara memilih *File/Save/Computer* lalu ubah jenis *Save As* menjadi CSV (comma delimited) (*.csv).

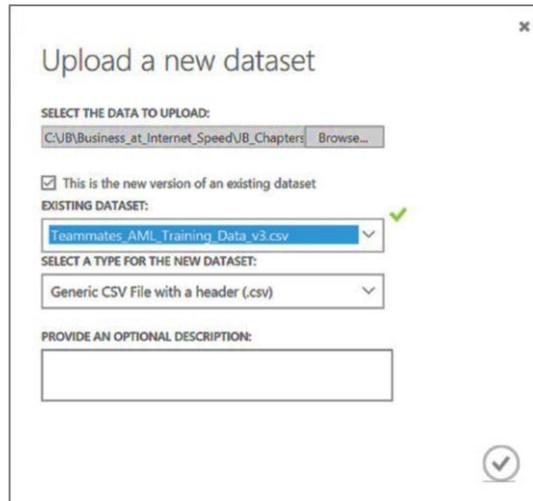
Unggah File Data Pelatihan ke Azure Machine Learning Studio

Selanjutnya, di sesi browser Azure Machine Learning Studio Anda, klik ikon + Baru di sudut kiri bawah, lalu klik opsi Kumpulan Data dan Dari File Lokal, seperti yang ditunjukkan pada Gambar 8.16.



Gambar 8.16. Azure Machine Learning Studio: Kumpulan data baru dari opsi file lokal

Selanjutnya, Anda mempunyai kesempatan untuk memasukkan lokasi file data sumber yang akan diunggah dari folder disk lokal Anda. Setelah menelusuri dan memilih file lokal yang baru saja Anda simpan dari spreadsheet Excel, tetapkan jenis file sumber sebagai file CSV generik dengan header (.csv), seperti yang ditunjukkan pada Gambar 8.17.

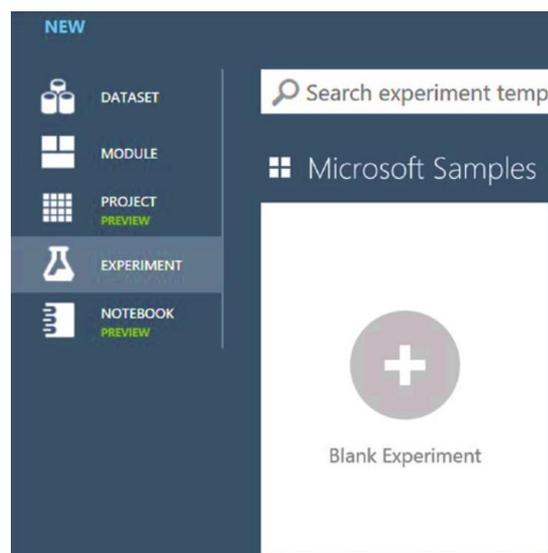


Gambar 8.17. Unggah himpunan data Azure Machine Learning baru dari file CSV lokal

Klik pada ikon Tanda Centang untuk memulai pengunggahan file. Bilah status akan muncul di bagian bawah layar untuk menampilkan kemajuan saat file Anda diunggah ke akun penyimpanan Azure ML. Sekarang setelah Anda mengunggah data pelatihan, Anda dapat melanjutkan membuat eksperimen Machine Learning.

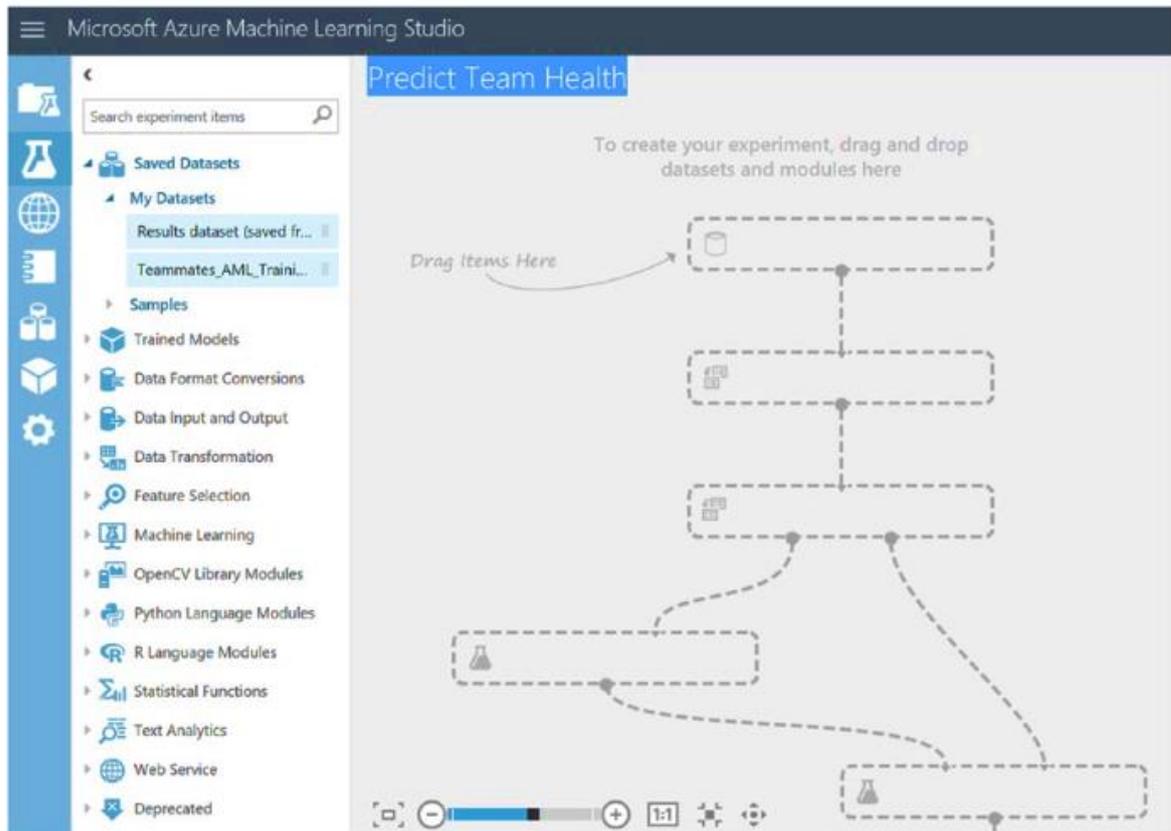
Buat Eksperimen Baru di Azure Machine Learning Studio

Selanjutnya, Anda membuat dan menguji eksperimen Machine Learning baru dengan menggunakan Azure Machine Learning Studio untuk membantu memprediksi tingkat kelelahan anggota tim. Mulai klik ikon + Baru di sudut kiri bawah Azure Machine Learning Studio, lalu klik ikon Eksperimen. Kemudian pilih templat Eksperimen Kosong untuk membuat eksperimen Azure Machine Learning baru dari awal, seperti yang ditunjukkan pada Gambar 8-18.



Gambar 8.18. Membuat eksperimen Machine Learning kosong baru di Azure Machine Learning Studio

Tindakan ini akan membuat eksperimen Machine Learning kosong baru di ruang kerja Anda. Mulailah dari yang sederhana dengan mengklik kolom nama teratas dan mengganti nama eksperimen menjadi Predict Team Health atau semacamnya, seperti yang ditunjukkan pada Gambar 8-19.

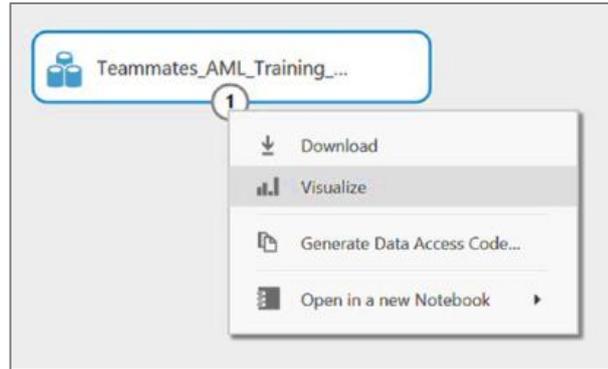


Gambar 8.19. Ganti nama eksperimen Azure Machine Learning kosong yang baru

Selanjutnya, klik opsi menu navigasi Kumpulan Data Tersimpan di bilah navigasi kiri untuk memperluas pilihan. Klik My Datasets dan Anda akan melihat file data CSV yang Anda upload tadi. Seret file ini ke tengah permukaan desainer dan letakkan di sana.

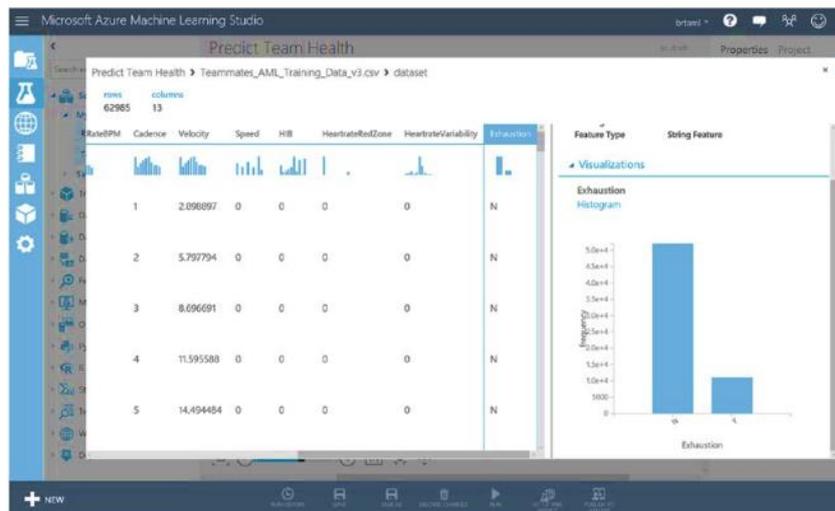
Visualisasikan Data Pelatihan

Anda dapat memanfaatkan beberapa alat bawaan di Azure Machine Learning Studio untuk membantu memvisualisasikan dan memahami data dengan lebih baik. Mulailah dengan mengklik kanan pada konektor bawah data pelatihan masukan yang baru saja Anda masukkan ke permukaan desainer dan kemudian pada ikon Visualisasikan, seperti yang ditunjukkan pada Gambar 8-20.



Gambar 8.20. Menggunakan alat bawaan untuk membantu memvisualisasikan data pelatihan

Selanjutnya, setelah mengklik ikon Visualize, Anda dapat menggunakan bilah penggeser untuk memilih kolom bernama Exhaustion di posisi terakhir. Dengan mengklik kolom Exhaustion, Anda akan melihat bahwa bagian Visualizations di sisi kanan layar secara otomatis akan terisi dengan tampilan visual sebaran bacaan. Kredibilitas data prediksi akan ditentukan berdasarkan data pelatihan stress test yang diunggah baru-baru ini, seperti yang ditunjukkan pada Gambar 8.21.



Gambar 8.21. Menggunakan alat visualisasi studio Azure Machine Learning bawaan untuk lebih memahami distribusi tingkat kelelahan di seluruh data pelatihan anggota tim

Perhatikan format histogram (diagram batang) yang digunakan untuk memvisualisasikan data dengan cepat. Hal ini membantu Anda dengan cepat memahami faktor-faktor utama dalam prediksi ML seperti campuran relatif data pemicu stres dalam set data pelatihan input Machine Learning. Dalam diagram batang, batang mewakili distribusi relatif tingkat kelelahan (Y atau N) dan pembacaan sensor terkait yang (Y) atau tidak (N) memicu tercapainya tingkat kelelahan.

Tambahkan Modul Jalankan Skrip R

Sekarang Anda telah menambahkan file data input ke Azure Machine Learning Studio Designer dan memvisualisasikan kontennya, sekarang saatnya memproses file dan

menyiapkannya untuk digunakan sebagai data pelatihan untuk eksperimen Machine Learning ini. Ingatlah bahwa ada kolom bernama Kelelahan dalam kumpulan data pelatihan masukan dan diisi dengan nilai Y atau N. Untuk melatih model ini menggunakan pustaka R, Anda perlu mengonversi nilai string ini menjadi nilai numerik yang sesuai. Untuk melakukan ini, Anda akan menggunakan modul R Script untuk mengisi kolom baru (ExhaustionLabel) dengan representasi numerik dari nilai di kolom Exhaustion. Singkatnya, Anda akan mengganti Y dengan 1 dan N dengan 0. Untuk melakukannya, perluas modul bahasa R di sisi kiri Machine Learning Studio dan seret modul Jalankan Skrip R ke permukaan desainer. Hubungkan file input data ke konektor kiri atas modul Execute R Script, seperti yang ditunjukkan pada Gambar 8.22.



Gambar 8.22. Jalankan Modul Skrip R

Klik pada modul Execute R Script dan kemudian klik pada R Script Editor di sisi kanan layar studio desainer. Masukkan kode R ini ke jendela R Script Editor:

```
# Map 1-based optional input ports to variables
dataset1 <- maml.mapInputPort(1) # class: data.frame

# Add/Change Label to numeric 0/1 for this R model
dataset1$ExhaustionLabel <- ifelse(dataset1$Exhaustion=="N", 0, 1)

# Select data.frame to be sent to the output Dataset port
maml.mapOutputPort("dataset1");
```

Kode R ini akan membuat kolom baru bernama ExhaustionLabel dan akan diisi dengan 0 atau 1 - berdasarkan nilai kolom bernama Exhaustion dan jika berisi N atau Y.

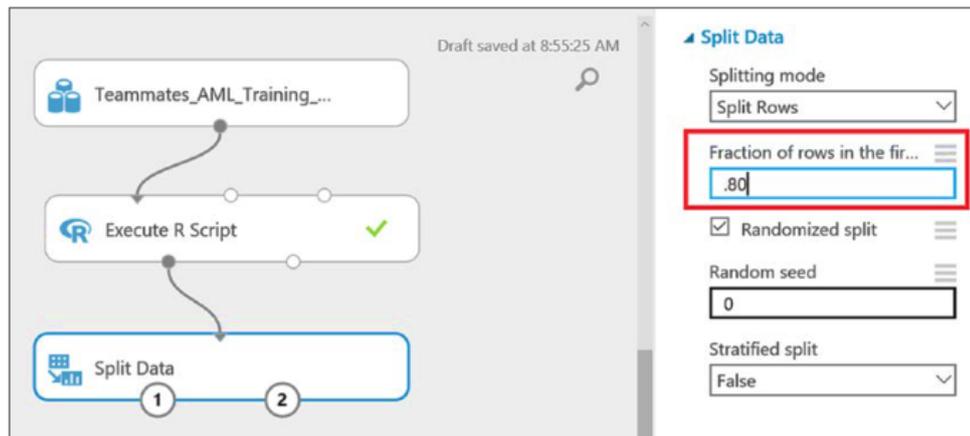
Tambahkan Modul Data Terpisah ke Eksperimen

Langkah selanjutnya adalah menambahkan modul lain yang disebut modul Split Data. Modul Split Data akan memisahkan data pelatihan masukan menjadi dua aliran berbeda. Aliran pertama akan digunakan untuk melatih model baru, dan aliran kedua akan digunakan untuk mengevaluasi keakuratan model baru.

Anda dapat dengan mudah menemukan modul Split Data dengan mengetikkan Split Data di jendela Pencarian di sisi kiri layar Azure Machine Learning Studio Designer. Anda juga

dapat menemukan modul ini dengan memperluas modul Transformasi Data di bawah subjudul Sampel dan Pemisahan.

Seret modul Split Data ke permukaan desainer langsung di bawah modul Execute R Script. Selanjutnya, buat sambungan antara konektor kiri bawah modul Execute R Script dan konektor atas modul Split Data. Klik pada modul Split Data untuk mengakses properti modul di sisi kanan. Ubah properti untuk Pecahan Baris dalam Kumpulan Data Keluaran Pertama ke nilai 0,80, seperti yang ditunjukkan pada Gambar 8.23.

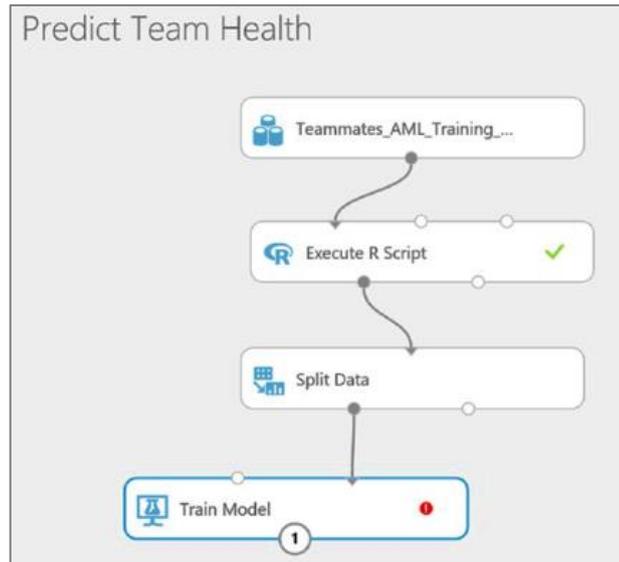


Gambar 8.23. Mengatur properti modul Azure Machine Learning Split Data

Tetapkan nilainya menjadi 0,80 untuk menunjukkan bahwa Anda ingin 80% dari kumpulan data pelatihan input digunakan untuk benar-benar melatih model baru ini. Sisanya sebesar 20% dari dataset pelatihan masukan akan digunakan untuk memeriksa keakuratan model Machine Learning yang baru.

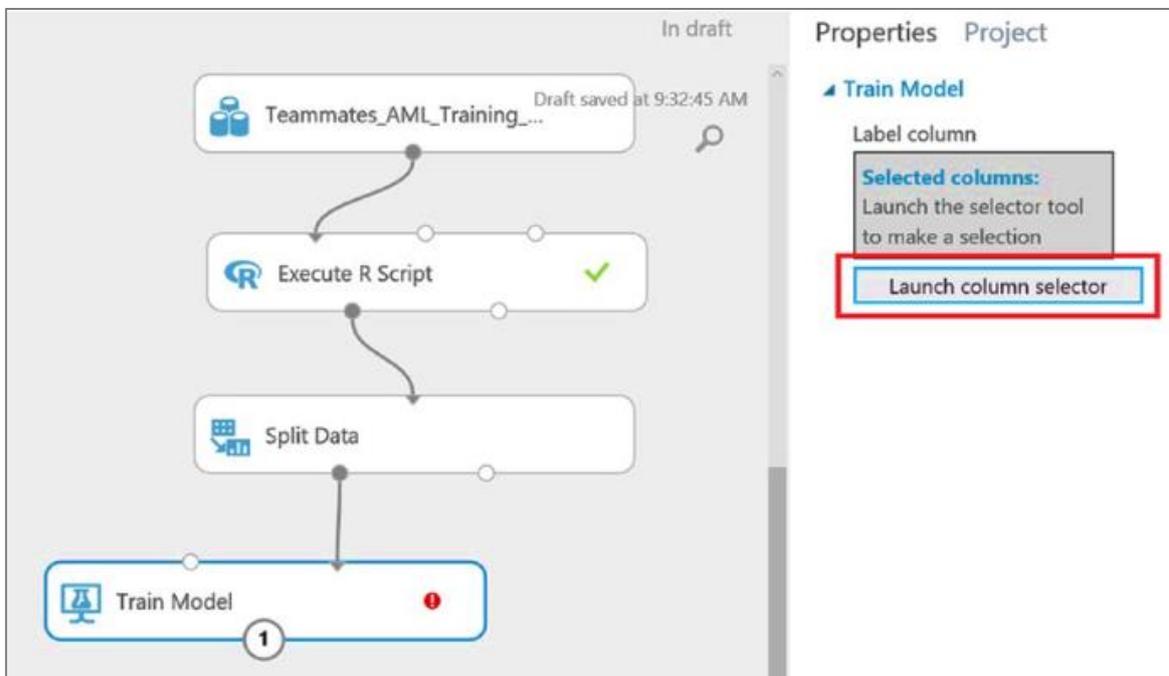
Tambahkan Modul Model Kereta ke Eksperimen

Langkah selanjutnya adalah menambahkan modul Train Model ke eksperimen ini. Temukan modul dengan mengetikkan Train Model di jendela Pencarian di sisi kiri layar Azure Machine Learning Studio Designer. Anda juga dapat menemukan modul ini dengan memperluas modul Machine Learning di bawah sub-judul Train. Seret modul Train Model ke permukaan desainer di bawah modul Split Data dan posisikan di kiri bawah modul Split Data. Selanjutnya, sambungkan bagian kiri bawah modul Split Data ke konektor kanan atas modul Train Model, seperti yang diilustrasikan pada Gambar 8.24.



Gambar 8.24. Menambahkan modul Train Model ke eksperimen

Anda mungkin memperhatikan ada tanda seru berwarna merah yang ditampilkan di modul Train Model. Metode Azure Machine Learning ini memberi tahu Anda bahwa Anda perlu mengatur properti dalam modul. Untuk memperbaiki peringatan merah, klik modul Train Model. Anda akan melihat jendela Properties muncul di panel di sisi kanan layar Studio. Klik pada tombol Launch Column Selector, seperti yang diilustrasikan pada Gambar 8.25.



Gambar 8.25 Luncurkan pemilih kolom untuk modul Train Model

Pemilih Kolom adalah alat yang banyak digunakan di Azure Machine Learning Studio yang dengan cepat memilih semua atau kolom tertentu untuk digunakan saat alur eksekusi diteruskan ke setiap modul. Untuk modul Train Model, Anda perlu menunjukkan kolom data

masuk mana yang Anda inginkan untuk model Machine Learning baru untuk memprediksi. Mulailah dengan mengetik di kolom bernama ExhaustionLabel. Anda akan melihat bahwa AML studio menyertakan fitur bagus untuk IntelliSense/type-ahead di kolom pencarian nama kolom. Gambar 8.26 menggambarkan pemilih kolom yang diisi dengan nama kolom tunggal ExhaustionLabel.

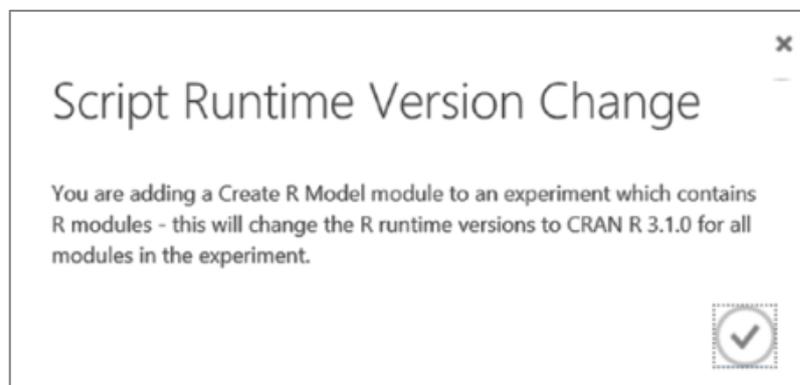


Gambar 8.26. Atur modul Kereta Model, bidang Pemilih Kolom untuk menunjukkan kolom mana yang akan diprediksi

Ingatlah untuk mengklik ikon tanda centang di kanan bawah layar pemilih kolom peluncuran untuk menyimpan perubahan Anda. Ini akan menyiapkan eksperimen dengan baik untuk langkah berikutnya, di mana Anda akan memasukkan kode R untuk menjalankan algoritma klasifikasi biner.

Masukkan Modul Buat Model R ke dalam Eksperimen

Pada langkah ini, Anda menambahkan modul Buat Model R ke eksperimen. Latihan ini menggambarkan kemampuan integrasi R penuh di Azure Machine Learning Studio. Mulailah dengan menemukan modul dengan mengetikkan Buat Model R di jendela Pencarian di sisi kiri layar Azure Machine Learning Studio Designer. Ini harus menemukan modul yang tepat. Anda juga dapat menemukan modul ini secara manual dengan memperluas modul Modul Bahasa R di panel navigasi sebelah kiri di Azure Machine Learning Studio. Tarik modul Create R Model ke permukaan desainer di atas modul Train Model dan di seberang modul Split Data. Perhatikan bahwa segera setelah Anda melepaskan modul Buat Model R, Anda mungkin menerima pesan bahwa versi skrip runtime akan berubah menjadi CRAN R 3.1.0 untuk semua modul dalam percobaan. Lihat tangkapan layar pada Gambar 8.27.



Gambar 8.27. Buat Model R: pemberitahuan perubahan versi runtime skrip

Selanjutnya, sambungkan bagian bawah modul Create R Model ke konektor kiri atas modul Train Model. Sekarang saatnya memodifikasi modul Create R Model dan memberikan

kode R. Klik pada modul Create R Model dan kemudian klik pada R Script Editor di sisi kanan layar. Anda dapat menggunakan modul Buat Model R untuk membuat model tidak terlatih dari skrip R yang Anda berikan.

Anda dapat mendasarkan model Anda pada pelajar mana pun yang disertakan sebagai paket R di lingkungan Azure Machine Learning.

Perhatikan bahwa ada dua lokasi untuk memasukkan kode R dalam modul Buat Model R:

- *Skrip R Pelatih*: Skrip R yang mengambil kumpulan data sebagai masukan dan keluaran model yang tidak terlatih.
- *Skrip Scorer R*: Skrip R yang mengambil model dan kumpulan data sebagai input dan output skor yang ditentukan dalam skrip.

Anda akan memasukkan blok kode R terpisah ke dalam masing-masing area ini. Setelah membuat model, Anda bisa menggunakan Train Model untuk melatih model pada himpunan data, seperti pelajar lainnya di lingkungan Azure Machine Learning. Model yang dilatih kemudian dapat diteruskan ke modul Score Model untuk menggunakan model tersebut dalam membuat prediksi. Model yang dilatih kemudian dapat disimpan, dan alur kerja penilaian dapat dipublikasikan sebagai layanan web. Untuk mengimplementasikan model ini, mulailah dengan memasukkan kode R ini ke jendela Trainer R Script Editor:

```
# Trainer R Script
# Input: dataset
# Output: model

# e1071 = pre-installed R package in the Azure Machine Learning
environment for
Binary Classification
library(e1071)

# The next three lines get the feature columns and the label column
from the dataset
# and combine them into a new R data frame that is named train.data:
features <- get.feature.columns(dataset)
labels   <- as.factor(get.label.column(dataset))
train.data <- data.frame(features, labels)

# The predefined function, get.feature.columns(), selects the columns
that were
designated as features in the metadata for dataset.
feature.names <- get.feature.column.names(dataset)

# the predefined function, get.feature.column.names(dataset), is used
to get feature
column names from the dataset. Those names are designated
# as the names for columns in train.data.
# a temporary name Class is created for the label column.
names(train.data) <- c(feature.names, "Class")
```

```
# Train the "Naïve Bayes" classifier algorithm by using the labels and
features in
the train.data data frame.
model <- naiveBayes(Class ~ ., train.data)
```

next, enter this r code into the scorer r script editor window:

```
# Scorer R Script
# Input: model, dataset
# Output: scores
# Loads the preinstalled R package.
library(e1071)

# Computes the predicted probabilities for the scoring dataset by
using the trained
model from the training script.
probabilities <- predict(model, dataset, type="raw")[,2]

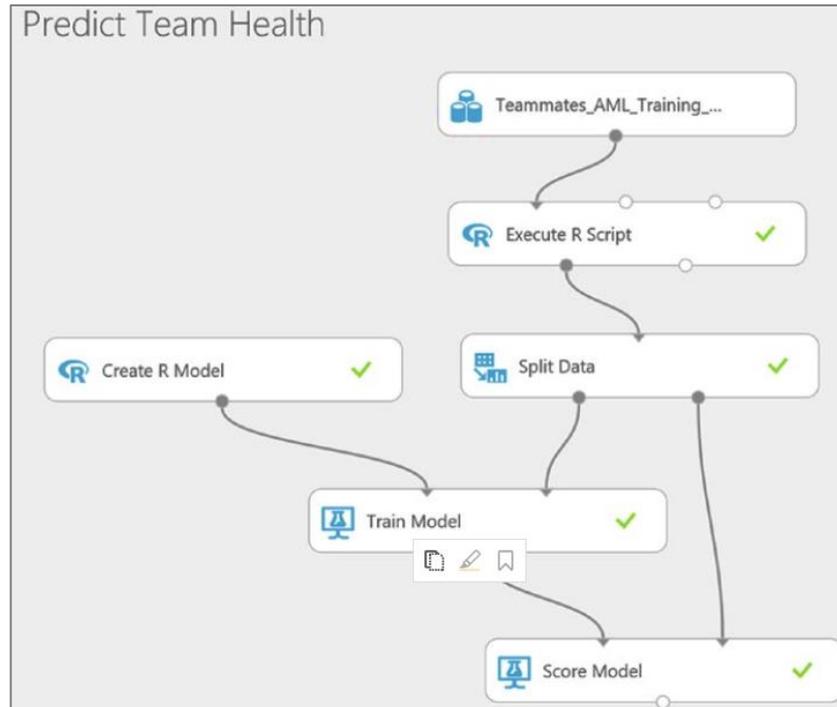
# Apply a default threshold of 0.5 to probabilities when assigning the
predicted
class labels.
classes <- as.factor(as.numeric(probabilities >= 0.5))

# Combine the class labels and probabilities into the output data
frame, named
"scores".
scores <- data.frame(classes, probabilities)
```

Setelah Anda memasukkan dua blok kode R, Anda dapat menyimpan perubahan hanya dengan mengklik di mana saja di Azure Machine Learning Studio Designer yang berada di luar modul Buat Model R.

Masukkan Modul Model Skor ke dalam Eksperimen

Langkah terakhir dalam membuat eksperimen Azure Machine Learning adalah menambahkan modul Model Skor ke dalamnya. Anda dapat menggunakan Model Skor untuk menghasilkan prediksi menggunakan model Pembelajaran Mesin klasifikasi atau regresi terlatih. Nilai prediksi dapat memiliki berbagai format, bergantung pada model dan data masukan Anda.



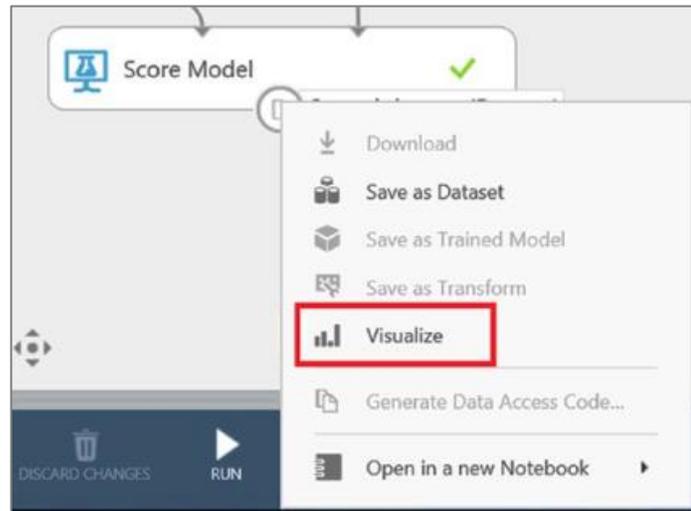
Gambar 8.28. Menyelesaikan eksperimen Azure Machine Learning untuk memprediksi kesehatan anggota tim

Dalam hal ini, karena Anda menggunakan model klasifikasi (biner) untuk membuat skor, modul Model Skor mengeluarkan nilai prediksi untuk kelas tersebut, bersama dengan probabilitas nilai prediksi. Inilah yang Anda perlukan untuk implementasi referensi Anda. Anda perlu membuat prediksi, berdasarkan data pembacaan sensor saat ini dan data uji stres historis, tentang apakah anggota tim berisiko mengalami kelelahan fisik. Mulailah dengan menemukan modul dengan mengetikkan Score Model di jendela Pencarian di sisi kiri layar Azure Machine Learning Studio Designer. Ini harus menemukan modul yang tepat. Anda juga dapat menemukan modul ini dengan memperluas kategori Machine Learning dan kemudian subkategori Skor di panel navigasi sebelah kiri di Azure Machine Learning Studio. Seret modul Score Model ke permukaan desainer di bawah dan di sebelah kanan modul Train Model. Hubungkan bagian bawah modul Train Model ke konektor kiri atas modul Score Model. Hubungkan bagian kanan bawah modul Split Data ke kanan atas modul Model Skor. Klik ikon Simpan di bagian bawah layar, lalu klik ikon Jalankan untuk memproses eksperimen Pembelajaran Mesin baru. Tanda centang hijau akan segera muncul di samping setiap modul dalam eksperimen Machine Learning saat diproses, seperti yang ditunjukkan pada Gambar 8.28.

Pada titik ini, Anda telah berhasil menyelesaikan eksperimen Azure Machine Learning untuk memprediksi kesehatan anggota tim berdasarkan riwayat pembacaan sensor dari uji tekanan berkala. Eksperimen ini menggunakan paket R klasifikasi biner bawaan untuk membuat model dan menilai hasilnya. Langkah selanjutnya mengkaji bagaimana mengeksplorasi keakuratan prediksi model.

Lihat Hasil Modul Model Skor untuk melihat Prediksi R

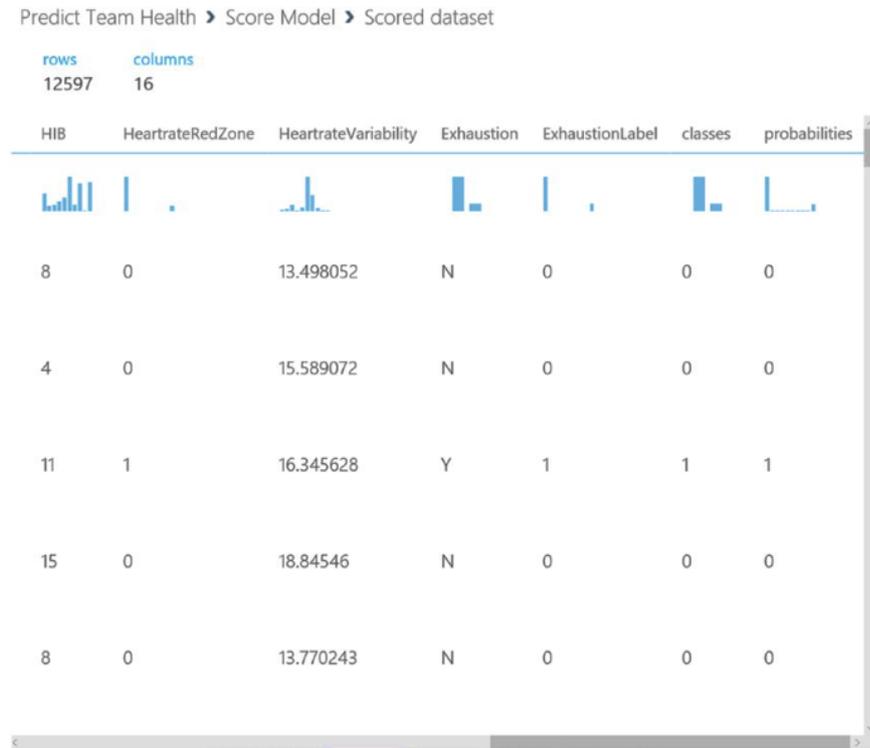
Pada langkah ini, Anda memeriksa prediksi yang dibuat oleh eksperimen pelatihan model R Machine Learning yang baru ini. Untuk memulai, cukup klik kanan pada konektor bawah modul Score Model dan kemudian klik opsi Visualize di menu konteks, seperti yang ditunjukkan pada Gambar 8.29.



Gambar 8-29. Visualisasikan hasil modul Score Model

Setelah Anda mengklik opsi Visualisasikan, Anda akan melihat daftar kolom di set pelatihan input. Gulir ke kanan untuk melihat tiga kolom terakhir dalam kumpulan data modul Model Skor:

- *ExhaustionLabel*: Ini adalah kolom yang dihasilkan dengan modul skrip R dalam percobaan. Anda mengganti Y atau N dengan 1 atau 0.
- *Kelas*: Ini adalah prediksi biner yang dibuat oleh model Pembelajaran Mesin R yang baru.
- *Probabilitas*: Ini adalah probabilitas yang dihitung bahwa prediksi model Machine Learning akan benar. Gambar 8.30 menggambarkan visualisasi kumpulan hasil Model Skor.



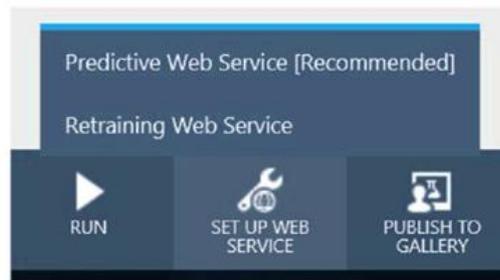
Gambar 8.30. Visualisasi modul Score Model

Anda dapat memeriksa hasil prediksi dan melihat bahwa algoritme tersebut cukup akurat, seperti Gambar 8-30 mengilustrasikan dengan baik di mana kolom Kelelahan = Y pada baris ketiga ke bawah. Perhatikan bahwa prediksi itu akurat dan probabilitasnya sangat tinggi, karena nilainya 1.

Buat Layanan Web dari Eksperimen Pembelajaran Mesin R

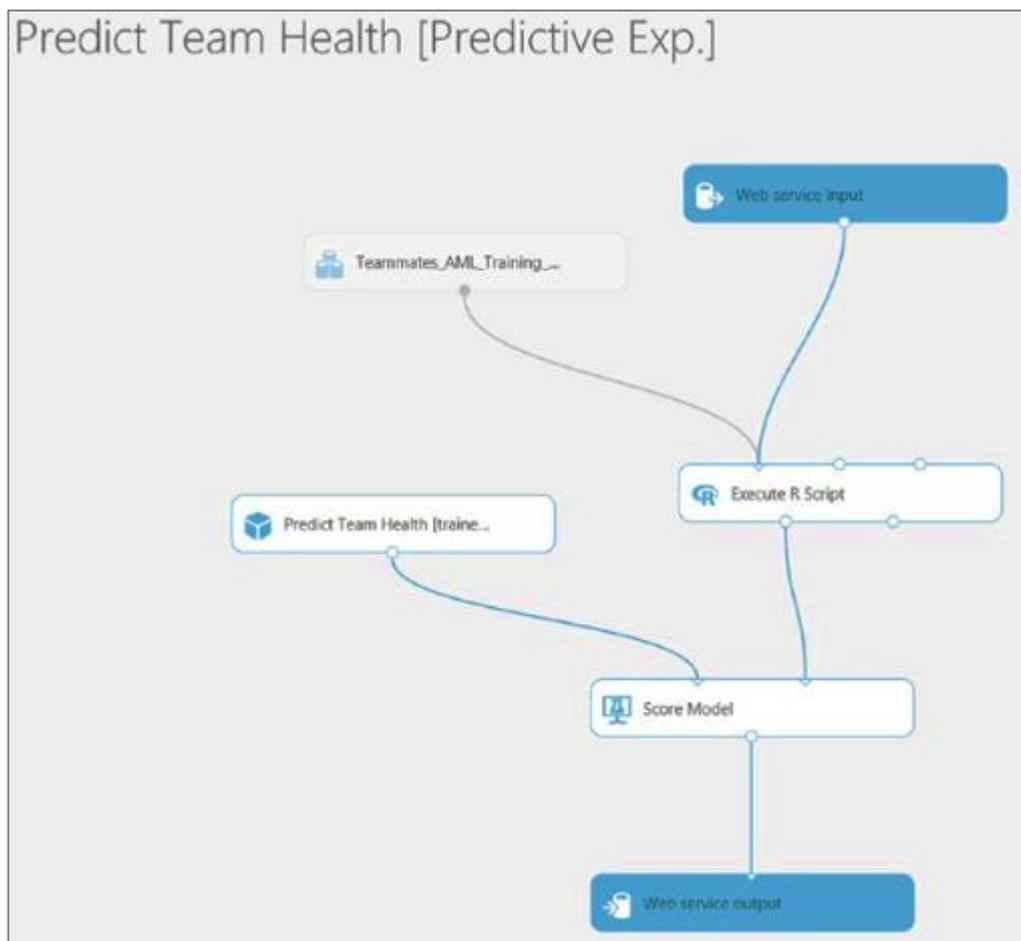
Sekarang setelah Anda memiliki model Azure Machine Learning yang berfungsi, sekarang saatnya untuk memproduksi eksperimen ini dengan mengeksposnya sebagai Layanan Web Azure Machine Learning. Membuat Layanan Web Azure Machine Learning memungkinkan Anda menyematkan analitik prediktif ke dalam aplikasi. Anda dapat meneruskan data fitur input baru ke Layanan Web Azure Machine Learning. Data fitur baru akan dijalankan melalui model Machine Learning dan layanan web akan mengirimkan kembali prediksinya.

Untuk memulai, pastikan Anda telah menjalankan eksperimen setidaknya sekali. Kemudian klik ikon Siapkan Layanan Web/Layanan Web Prediktif [Disarankan] di bagian bawah layar Azure Machine Learning Studio Designer, seperti yang ditunjukkan pada Gambar 8.31.



Gambar 8.31. Siapkan layanan web: layanan web prediktif

Setelah beberapa saat, eksperimen Anda akan diubah menjadi eksperimen Machine Learning yang baru dan efisien. Dalam versi layanan web baru ini, tidak perlu lagi melatih model atau membagi data yang masuk menjadi set pelatihan dan pengujian. Anda juga akan melihat bahwa dua port koneksi tambahan telah ditambahkan secara otomatis untuk layanan web—titik akhir Input dan Output—seperti yang diilustrasikan pada Gambar 8.32.



Gambar 8.32. Layanan web prediktif yang dibuat dari eksperimen Azure Machine Learning

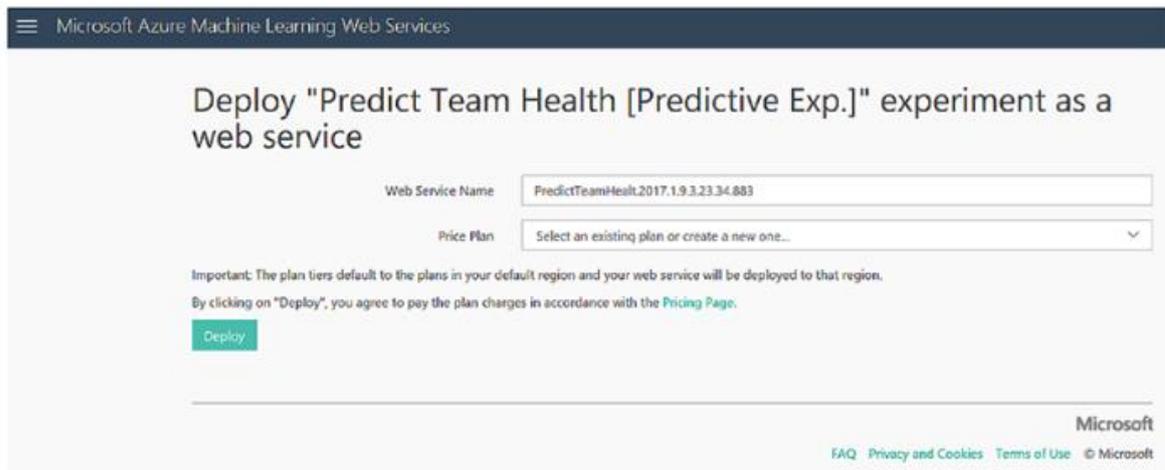
Klik ikon Jalankan di bagian bawah layar perancang untuk mengkompilasi layanan web baru. Anda akan melihat tanda centang hijau muncul di samping setiap modul saat modul

telah diproses. Setelah selesai, ikon Deploy Web Service akan diaktifkan di bagian bawah layar Azure Machine Learning Studio Designer, seperti yang ditunjukkan pada Gambar 8-33.



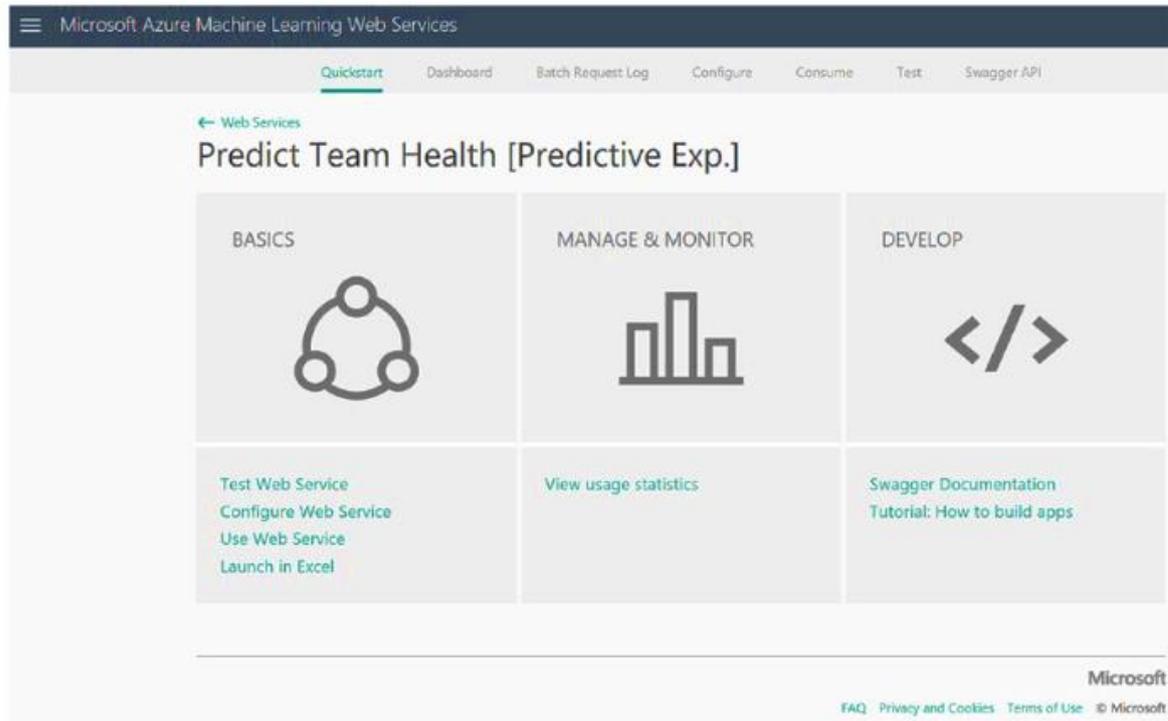
Gambar 8-33. Opsi penerapan layanan web

Pilih opsi Pratinjau Deploy Web Service [Baru], dan Anda akan diarahkan ke layar seperti yang ditunjukkan pada Gambar 8.34.



Gambar 8.34. Menyebarkan opsi pratinjau layanan web [baru].

Layar ini memungkinkan Anda mengubah nama layanan web dan paket harga Azure yang akan menjalankannya. Tentukan pilihan Anda dan klik tombol Deploy untuk menyebarkan layanan web. Tab browser baru akan terbuka dan Anda akan melihat layar seperti pada Gambar 8.35.



Gambar 8.35. Dasbor Layanan Web Pembelajaran Mesin Azure

Halaman web ini adalah titik peluncuran untuk semua hal yang terkait dengan pengelolaan, pengujian, dan pemantauan Layanan Web Azure Machine Learning Anda.

- Uji Layanan Web: Opsi ini memungkinkan Anda memanggil Layanan Web Azure Machine Learning baru secara interaktif dengan parameter Anda sendiri dan kemudian melihat hasil yang diprediksi.
- Konfigurasi Layanan Web: Memungkinkan Anda melihat kunci keamanan primer dan sekunder, mengaktifkan pembuatan log, dan mengaktifkan data sampel.
- Gunakan Layanan Web: Memberikan panduan dan kode sampel tentang cara memanggil layanan Web Azure Machine Learning dengan aman melalui respons permintaan atau mode Batch. Contoh kode disediakan dalam C#, Python, Python3, dan R.
- **Bonus**: Opsi ini juga menyediakan spreadsheet Excel dengan makro tertanam untuk memanggil Layanan Web AML Anda secara interaktif dan menguji hasilnya.
- Peluncuran di Excel: Memungkinkan Anda meluncurkan lembar bentang Excel pengujian dengan makro tertanam untuk memanggil Layanan Web AML Anda secara interaktif dan menguji hasilnya.

Ini mungkin salah satu alat dan fitur terbaik yang disertakan dengan Azure Machine Learning (gratis). Fitur ini memungkinkan Anda mengisi spreadsheet dengan data sampel untuk menguji Layanan Web Azure Machine Learning baru Anda tanpa harus membuat aplikasi klien. Gambar 8.36 menggambarkan spreadsheet Excel dengan input ditampilkan di bagian atas, panel Add-in Excel Office ditampilkan di sebelah kanan, dan hasil prediksi Layanan Web ditampilkan di bagian bawah spreadsheet.

The screenshot shows an Excel spreadsheet with data from an Azure Machine Learning web service. The spreadsheet has columns for various health metrics. The right-hand pane shows the 'Check Team Health' tool interface with input and output sections.

Userid	BreathingRate	Temperature	Ventilation	Activity	HeartRate	Cadence	Velocity	Speed	HIB	HeartRateRed	HeartRateVar	Exhaustion	Scored	Lat	Scored	Probal
c3ec2fe4-67	37.449294	96.292156	16454.153	0.1335638	90	1	2.8998969	0	0	0	0	0	N			
c3ec2fe4-67	37.511383	96.881865	16547.126	0.1346221	91	2	5.7977938	0	0	0	0	0	N			
c3ec2fe4-67	37.573473	96.511788	16640.090	0.1400789	91	3	8.6966907	0	0	0	0	0	N			
c3ec2fe4-67	37.635562	96.316135	16733.071	0.1433556	92	4	11.5935588	0	0	0	0	0	N			
c3ec2fe4-67	37.697651	96.895811	16826.044	0.1465928	92	5	14.4944494	0	0	0	0	0	N			
2c49b721-e	23.720322	96.748524	25844.408	0.4625473	141	102	284.118	4	2	0	8.1185469	Y				

Gambar 8.36. Lembar kerja pengujian web AML Excel 2013

Dalam latihan ini, Anda berhasil membuat Layanan Web baru dari Eksperimen Azure Machine Learning. Anda kemudian menerapkannya dan mengujinya menggunakan spreadsheet Excel. Ada alat Layanan Web Azure Machine Learning tambahan untuk menyediakan dukungan untuk diagnostik, pembuatan log, dan pembuatan dokumentasi Swagger API.

Buat Pelatihan Ulang Layanan Web AML

Salah satu fitur paling menarik di Azure Machine Learning adalah kemampuan untuk melatih ulang model Machine Learning secara terprogram berdasarkan data pelatihan baru atau yang diperbarui. Fitur tunggal ini memberikan kemampuan unik bagi komputer untuk diprogram agar secara otomatis beradaptasi dan belajar dari pengalaman.

Benar hal yang sama dengan buku fiksi ilmiah dan serial film Terminator. Kabar baiknya adalah kita masih berada di tahap awal dan belum ada kekhawatiran bahwa komputer akan menjadi sadar diri setidaknya untuk beberapa tahun ke depan. Ingatlah bahwa Anda membuat pekerjaan Azure Data Factory di Bab 6 untuk melatih kembali model Azure Machine Learning melalui Layanan Web Azure Machine Learning yang terbuka. Anda akan membuat layanan tersebut dalam latihan ini.

Untuk menerapkan eksperimen pelatihan sebagai layanan web pelatihan ulang, Anda harus menambahkan input dan output layanan web ke model yang ada.

Dengan menghubungkan modul Output Layanan Web ke modul Model Kereta eksperimen, Anda mengaktifkan eksperimen pelatihan untuk menghasilkan model terlatih baru yang kemudian dapat Anda gunakan dalam eksperimen prediktif Anda.

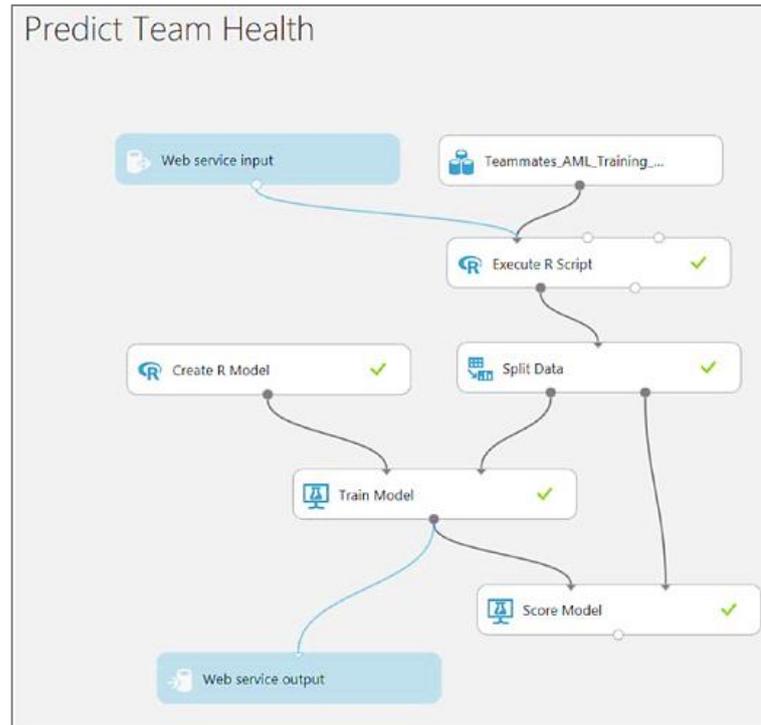
Berikut adalah alur kerja tingkat tinggi dari proses untuk melatih kembali Layanan Web Azure Machine Learning:

- ❖ Perbarui eksperimen pelatihan Azure Machine Learning untuk memungkinkan pelatihan ulang dengan menambahkan port Input dan Output Layanan Web.
- ❖ Menyebarkan Layanan Web Azure Machine Learning baru.
- ❖ Gunakan kode contoh Layanan Eksekusi Batch untuk melatih ulang model dengan memanggil Layanan Web Pembelajaran Mesin pelatihan ulang yang baru.
- ❖ Perhatikan bahwa tidak ada antarmuka permintaan-respons untuk melatih ulang Layanan Web Azure Machine Learning.
- ❖ Anda hanya dapat melatih ulang model Azure Machine Learning menggunakan layanan eksekusi batch.

Untuk memperbarui implementasi referensi eksperimen pelatihan Azure Machine Learning, navigasikan ke eksperimen pelatihan asli di Azure Machine Learning Studio, lalu ikuti langkah-langkah berikut:

1. Hubungkan modul Input Layanan Web ke input data Anda:
 - a. Petunjuk: Pastikan data masukan diproses dengan cara yang sama seperti data pelatihan asli.
 - b. Temukan modul dengan mengetikkan Input Layanan Web di jendela Pencarian di sisi kiri layar Azure Machine Learning Studio Designer. Anda juga dapat menemukan modul ini dengan memperluas modul Layanan Web.
 - c. Seret modul Input Layanan Web ke permukaan desainer tepat di atas modul Jalankan Skrip R.
 - d. Hubungkan modul Input Layanan Web ke konektor kiri atas modul Jalankan Skrip R.
2. Hubungkan modul Output Layanan Web ke output modul Train Model.
 - a. Temukan modul dengan mengetikkan Output Layanan Web di jendela Pencarian di sisi kiri layar Azure Machine Learning Studio Designer. Anda juga dapat menemukan modul ini dengan memperluas modul Layanan Web.
 - b. Seret modul Output Layanan Web ke permukaan desainer tepat di bawah modul Train Model.
 - c. Hubungkan modul Train Model ke bagian atas modul Output Layanan Web.
3. Jalankan percobaan.

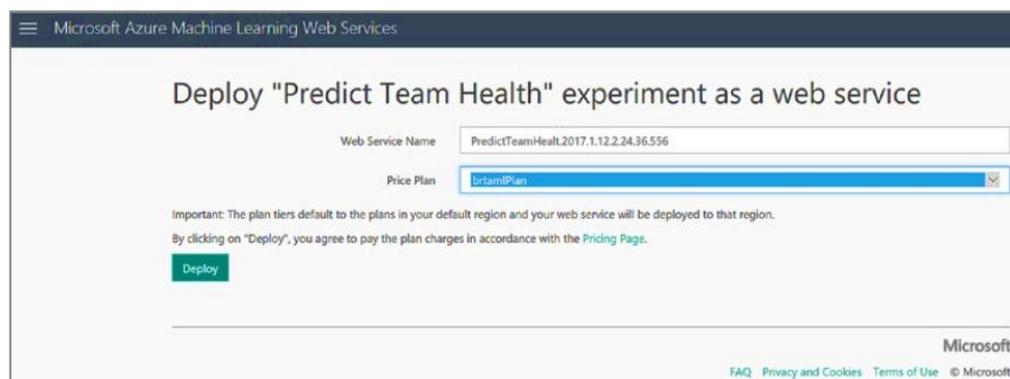
Gambar 8.37 memperlihatkan eksperimen Azure Machine Learning yang diperbarui setelah menambahkan modul Input dan Output Layanan Web.



Gambar 8.37. Memperbarui eksperimen Azure Machine Learning setelah menambahkan modul input dan output layanan web

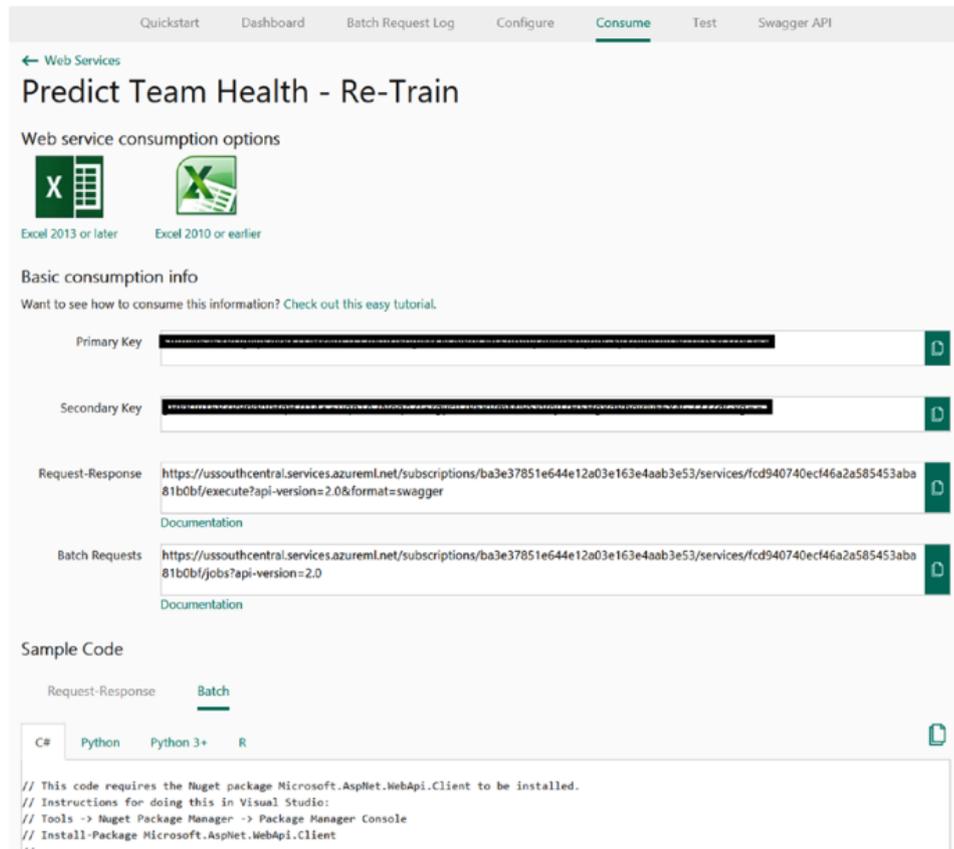
Selanjutnya, Anda harus menyebarkan eksperimen pelatihan Azure Machine Learning yang diperbarui sebagai layanan web yang menghasilkan model terlatih bersama dengan hasil evaluasi model. Untuk melakukannya, ikuti langkah-langkah berikut:

- Di bagian bawah kanvas eksperimen, klik Siapkan Layanan Web, lalu pilih Terapkan Layanan Web [Baru].
- Portal Azure Machine Learning Web Services kemudian akan membuka tab browser baru ke halaman Deploy Web Service.
- Ketikkan nama untuk layanan web Anda, pilih paket pembayaran, lalu klik Terapkan.
- Anda hanya dapat menggunakan metode Eksekusi Batch saat membuat model terlatih. Anda akan melihat layar seperti Gambar 8-38.



Gambar 8.38. Menyebarkan layanan web pelatihan ulang Machine Learning

Setelah layanan web baru diterapkan, klik ikon Konsumsi di bilah navigasi atas. Ini akan membawa Anda ke halaman tempat Anda dapat mengambil informasi penting tentang layanan web seperti kunci akses keamanan primer/sekunder bersama dengan URL layanan web untuk layanan respons permintaan dan eksekusi batch. Gambar 8-39 menunjukkan halaman Penggunaan Layanan Web. Perhatikan bahwa ada tautan untuk contoh kode untuk memanggil layanan web yang terletak di bagian bawah halaman. Selain itu, ada kode contoh untuk memanggil versi batch layanan web, yang merupakan hal yang Anda perlukan.



Gambar 8.39. Tab Konsumsi Layanan Web digunakan untuk melatih kembali Layanan Web Pembelajaran Mesin

Kode sampel yang diberikan akan membantu Anda membuat aplikasi konsol C# untuk melatih kembali model Azure Machine Learning. Selain membuat aplikasi Konsol .NET C# kustom untuk memanggil Layanan Eksekusi Batch untuk layanan web pelatihan ulang Machine Learning, Anda juga cukup menggunakan Azure Data Factory untuk menyiapkan dan menjalankan pekerjaan Azure Machine Learning. Yang diperlukan hanyalah Anda meneruskan URL layanan web terkait untuk pelatihan ulang dan menyediakan lokasi penyimpanan Azure untuk data pelatihan ulang.

8.5 RINGKASAN

Kita membahas cukup banyak hal dalam bab ini, dimulai dengan gambaran umum dan latar belakang tingkat tinggi tentang Pembelajaran Mesin dan analisis prediktif. Teknologi-teknologi ini akan terus meluas penggunaannya dan terus meresap ke dalam masyarakat modern kita.

Bisnis modern yang sukses saat ini tahu cara memanfaatkan data mereka secara maksimal dan, dalam banyak kasus, bahkan dapat mengubah data tersebut menjadi sumber pendapatan tambahan. Microsoft telah mendemokratisasi analitik prediktif dengan penawaran Machine Learning-nya. Kami menjelajahi bahasa R dan bagaimana Microsoft mengintegrasikan R ke dalam produk intelijen bisnis intinya seperti SQL Server 2016 dan Azure Machine Learning. Kami menunjukkan bahwa kombinasi R dan Azure Machine Learning membuka dunia kode dan paket ilmu data yang luas yang kini dapat dengan mudah dimanfaatkan dan diekspos melalui antarmuka Azure Machine Learning Web Service. Analogi yang bagus adalah integrasi R seperti akselerator untuk Machine Learning.

Kami menyimpulkan dengan melihat kemampuan yang kuat dalam melatih ulang model Azure Machine Learning. Fitur ini benar-benar memberikan kemampuan bagi mesin untuk terus belajar dan terus beradaptasi seiring perubahan lingkungan. Sama seperti dunia hewan. Kecuali bahwa mesin tidak pernah tidur. Kami juga melihat bagaimana Azure telah mengintegrasikan layanan cloud-nya untuk memaksimalkan proposisi nilai. Salah satu contoh integrasi ini adalah antara Azure Data Factory (ADF) dan Azure Machine Learning (AML).

Dengan ADF, Anda dapat membuat alur prediktif untuk memproses aliran data input besar dengan memanggil Azure Machine Learning Batch Execution Services (BES). Selain itu, Anda dapat membuat alur pelatihan ulang untuk memanggil titik akhir Layanan Eksekusi Batch Web pelatihan ulang AML dengan data pelatihan yang diperbarui untuk secara otomatis melatih ulang model Machine Learning Anda secara terjadwal.

BAB 9

VISUALISASI DATA, PERINGATAN, DAN NOTIFIKASI DENGAN POWER BI

Bab ini mengeksplorasi penggunaan visualisasi data, peringatan, dan pemberitahuan untuk membantu bisnis saat ini memberikan komunikasi yang berguna kepada karyawan dan pelanggan agar berhasil mengelola operasi mereka secara real-time. Kita akan memulai bab ini dengan gambaran singkat tentang lanskap pelaporan saat ini, lalu melihat bagaimana teknologi Microsoft seperti fungsi Power BI dan Azure dapat membantu memberikan solusi yang cepat dan mudah. Kami kemudian akan membahas cara mengaktifkan teknologi ini sebagai bagian dari skenario penerapan referensi kami. Kami akan mengakhiri bab ini dengan mempelajari penggunaan aplikasi "Simulator" C# .NET untuk secara otomatis menyalurkan ribuan contoh transaksi data pengujian melalui implementasi cloud Azure kami. Data simulasi akan diproses secara real-time menggunakan Azure Stream Analytics, dan kami akan menerapkan dasbor Power BI untuk melihat arsitektur cloud Lambda kami. Arsitektur Lambda, seperti yang Anda ingat, dirancang untuk menangani data dalam jumlah besar dengan memanfaatkan metode pemrosesan batch dan aliran. Implementasi referensi kami Dasbor Power BI akan menampilkan output untuk ketiga "suhu" model pemrosesan arsitektur Lambda melalui visualisasi jalur data dingin, hangat, dan panas.

9.1 LANSKAP PELAPORAN MODERN

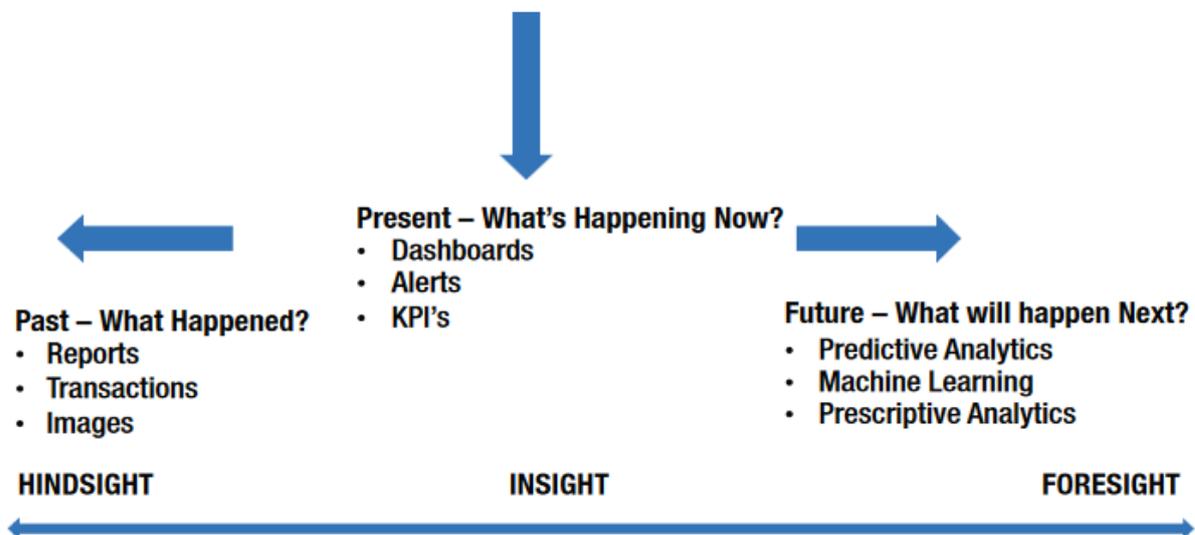
Cetakan garis hijau kemarin, informasi basi, dan laporan "TPS" kuno telah digantikan dengan komunikasi real-time yang ada di mana-mana, perangkat seluler modern, dan komunikasi instan, umpan balik, dan pelaporan. Peringatan teks instan, pembaruan email otomatis, portal web, dan dasbor Indikator Kinerja Utama (KPI) menguasai lanskap bisnis saat ini. Fitur dan kemampuan ini sekarang mewakili sebagian besar spesifikasi perangkat lunak konsumen, komersial, atau perusahaan untuk Produk yang Layak Minimum (MVP).

Saat ini, penekanannya adalah pada penggunaan istilah "komunikasi visual" untuk menyampaikan makna dengan cepat dan elegan kepada pengguna. Keberhasilan terletak pada kemampuan organisasi untuk memberikan informasi bisnis utama kepada penggunanya dengan cara yang jelas dan ringkas. Terkadang, manajemen dengan pengecualian dapat menjadi prinsip panduan saat ini, dan laporan, visualisasi, serta alat yang membantu mengisolasi, mengidentifikasi, dan memperbesar tren dan pengecualian yang tidak normal sangat bermanfaat saat ini. Telah dikatakan bahwa visualisasi data yang hebat adalah seni sekaligus sains. Kami tentu saja setuju dan juga mengusulkan untuk memiliki alat "metafora" kontrol visual yang ekstensif, fleksibel, dan dapat disesuaikan untuk membantu memberikan pelaporan dan pemberitahuan.

Selain melaporkan apa yang terjadi saat ini, dengan dasbor dan visualisasi, memprediksi apa yang akan terjadi selanjutnya dengan cepat menjadi tren besar saat ini

dalam dunia ilmu data. Kemajuan teknologi seperti Machine Learning, analisis prediktif, dan kecerdasan buatan membantu membuka jalan tersebut.

Lanskap intelijen bisnis modern saat ini mengharuskan perusahaan yang sukses mengandalkan perpaduan yang tepat antara tinjauan ke belakang, wawasan, dan tinjauan ke masa depan untuk membuat keputusan bisnis yang efektif guna mendorong kesuksesan. Saat ini, tidak cukup hanya mengandalkan strategi bisnis di masa lalu; Saat ini, ada begitu banyak nilai yang dapat diperoleh dari data historis, tren pasar, dan hasil KPI yang dapat dengan mudah diperoleh untuk membentuk strategi prediktif secara real-time. Semua ini menjadi mungkin dengan munculnya komputasi awan bayar sesuai penggunaan, penyimpanan awan yang murah dan tersedia di mana-mana, serta mesin analisis prediktif yang demokratis seperti Azure Machine Learning. Untuk bersaing di pasar global saat ini, dunia usaha harus belajar bagaimana memanfaatkan aset data utama mereka dan mengubahnya menjadi aliran yang menghasilkan hasil positif. Hasil tersebut dapat diwujudkan dalam bentuk peningkatan pendapatan, layanan pelanggan yang lebih baik, dan loyalitas pelanggan yang luar biasa. Gambar 9-1 mengilustrasikan lanskap intelijen bisnis saat ini. Perhatikan bahwa terdapat analisis 360 derajat yang digambarkan, dengan penekanan pada penggunaan analisis prediktif untuk membantu mendorong hasil di masa depan.



Gambar 9.1. Lanskap intelijen bisnis saat ini

Ada yang berpendapat bahwa di dunia Internet of Things yang modern saat ini, terdapat arti baru untuk istilah AI, atau Kecerdasan Buatan. Kenyataannya adalah AI dengan cepat digantikan dengan makna baru; yaitu kecerdasan yang "dapat ditindaklanjuti". Dan tidak ada konsep intelijen yang dapat ditindaklanjuti yang lebih penting (dan perlu) selain dengan munculnya Internet of Things (IoT) dan semua data yang dihasilkan yang dapat dihasilkan oleh sistem ini.

Karena banyaknya data sensor, manusia harus semakin bergantung pada sistem komputer canggih untuk melacak, melaporkan, mengelola, dan memprediksi pengecualian. Untungnya, seperti yang telah kita lihat di bab sebelumnya, teknologi seperti *Azure Stream Analytics* (ASA) dan *Azure Machine Learning* (AML) menyediakan fitur dan kemampuan

berharga untuk membantu membuat dan mengelola skenario ini. Alat-alat ini benar-benar dapat memungkinkan banyak skenario kasus penggunaan “kecerdasan yang dapat ditindaklanjuti”, dan kita akan mengeksplorasi lebih jauh aspek integrasi teknologi ini ketika kita memeriksa topik peringatan dan pemberitahuan nanti di bab ini.

Saat ini, terdapat data, lebih banyak data, kelebihan data, dan kemudian ada intelijen yang dapat ditindaklanjuti. Mengungkapkan hal-hal yang paling penting tepat pada saat dibutuhkan adalah salah satu kunci keberhasilan dalam mengoperasikan platform intelijen bisnis berbasis sensor.

9.2 IKHTISAR POWER BI

Power BI adalah layanan analitik berbasis cloud dari Microsoft yang memiliki tujuan memberikan “waktu yang lebih cepat untuk mendapatkan wawasan”. Power BI memiliki kemampuan untuk menyatukan data dari berbagai sumber untuk menghadirkan visualisasi yang kaya dan pandangan komprehensif tentang operasi bisnis di desktop, web, dan perangkat seluler.

Meskipun Power BI jelas menarik bagi pengguna yang mahir, Power BI juga ditujukan bagi pengguna bisnis dan analis yang kurang teknis untuk membantu mereka terhubung dengan data mereka dan memberikan kemampuan intelijen bisnis layanan mandiri. Fitur utama Power BI adalah memungkinkan Anda melihat semua data melalui “satu panel kaca” dan kemudian membuat lingkungan analitik lengkap untuk memantau data dan berbagi laporan.

Layanan Power BI

Layanan Power BI, yang terletak di www.powerbi.com, adalah layanan berbasis cloud Microsoft yang memungkinkan Anda untuk:

- Ciptakan Visualisasi yang Indah: Ceritakan kisah data yang menarik melalui kontrol visualisasi yang kaya.
- Membangun Dasbor yang Kaya dan Langsung: Hal ini dapat membantu mengubah intelijen bisnis menjadi wawasan bisnis dengan menyoroti pengecualian dalam data Anda.
- Membuat Laporan dan Kumpulan Data: Dapat digunakan untuk membuat visualisasi data dan dasbor pelaporan.
- Menyediakan Data Terkini: Melalui penyegaran data secara real-time, otomatis, dan terjadwal.
- Membuat dan Berbagi Dasbor Power BI: Mudah dengan orang lain di organisasi Anda.
- Ajukan Pertanyaan tentang Data Anda dalam Bahasa Inggris Biasa: Melalui kueri bahasa alami.
- Memungkinkan Anda Tetap Terhubung dengan Data Anda: Aplikasi Power BI Mobile tersedia di setiap toko aplikasi seluler besar.

Desktop Power BI

Power BI Desktop adalah alat eksplorasi data visual dan pelaporan interaktif yang disediakan gratis oleh Microsoft. Ini menyediakan kanvas yang kaya dan berbentuk bebas

untuk eksplorasi data Anda secara mendalam, bersama dengan perpustakaan visualisasi interaktif yang luas. Power BI Desktop menawarkan pengalaman penulisan yang sangat produktif untuk membuat laporan untuk layanan Power BI. Fitur-fitur baru terus diintegrasikan dan alat ini diperbarui setiap bulan. Bagi pengguna yang mungkin tidak memiliki akses ke Excel 2013, Power BI Desktop dapat digunakan untuk mengimpor data, membuat model data, dan menulis serta berbagi Laporan Power BI melalui layanan Power BI.

Meskipun Power BI Desktop mungkin tidak menyertakan semua fitur analitis Excel, Power BI Desktop memberikan solusi sederhana dan elegan untuk membuat laporan, visualisasi, dan dasbor Power BI.

Power BI Desktop adalah mesin eksplorasi data visual canggih yang memungkinkan Anda menyambungkan, mengkueri, dan menganalisis data dengan cepat, lalu dengan cepat membuat laporan dan visualisasi menakjubkan. Dengan Power BI Desktop, Anda dapat:

- ❖ *Memperoleh dan Mempersiapkan Data:* Menggunakan kemampuan kueri dan pemfilteran yang ekstensif.
- ❖ *Memanipulasi dan Mengkonsolidasikan:* Berbagai sumber data memungkinkan pengguna memanfaatkan data dari berbagai sumber dalam satu laporan.
- ❖ *Menetapkan Struktur Data:* Kemudian mengubah dan menganalisis data melalui alat bawaan dan komponen visual.
- ❖ *Visualisasikan dan Jelajahi Data:* Dengan cepat dan mudah melalui kanvas pembuat laporan desainer bentuk bebas, drag-and-drop.
- ❖ *Laporan Penulis:* Dengan beragam alat visualisasi data modern untuk menyempurnakan dan menyempurnakan dasbor dan laporan Anda.
- ❖ *Publikasikan Laporan yang Indah dan Interaktif:* Langsung ke app.powerbi.com, yang memudahkan berbagi dan berkolaborasi dalam proyek pelaporan yang kompleks.
- ❖ *Berbagi Laporan dengan Aman:* Hal ini dilakukan melalui pendekatan pustaka konten perusahaan yang unik dan terkurasi.

Membuka Kunci Analisis Data

Power BI Desktop juga memungkinkan kemampuan analisis data yang luas, memungkinkan penulis menghasilkan model data kaya yang berisi rumus dan hubungan. Misalnya, Anda dapat:

- ❖ Secara otomatis membuat model data hanya dengan mengimpor data:
 - Power BI Desktop dapat secara otomatis mendeteksi hubungan dalam data dan menerapkan ringkasan default.
- ❖ Menyempurnakan model data untuk memungkinkan penghitungan yang rumit:
 - Mengidentifikasi hubungan-hubungan penting di antara kumpulan data dari berbagai sumber.
 - Membuat hubungan antar tabel secara manual atau dengan menggunakan fitur Deteksi Otomatis.
 - Menyesuaikan jenis hubungan (satu-ke-satu, banyak-ke-banyak, atau banyak-ke-satu) untuk memperoleh wawasan data tertentu.

- *✱ Tentukan perhitungan: Anda dapat menentukan “tindakan” untuk menghasilkan bidang baru untuk digunakan dalam laporan dan dasbor Power BI:
 - Pengukuran memungkinkan Anda membuat data baru dari data yang sudah ada di model data Anda.
 - Salah satu manfaat langsung dari hal ini adalah hasil penghitungan pengukuran selalu berubah sebagai respons terhadap interaksi dengan laporan. Hal ini memungkinkan eksplorasi data ad hoc yang cepat dan dinamis.
 - Anda juga dapat menggunakan pustaka Data Analysis Expression (DAX) untuk membuat tabel terhitung. Misalnya, daripada membuat kueri dan memuat nilai ke dalam kolom tabel baru dari sumber data, Anda juga bisa membuat rumus Ekspresi Analisis Data (DAX) yang menentukan nilai tabel.
 - Tentukan sinonim: Memungkinkan pengalaman tanya jawab yang lebih baik di layanan Power BI untuk kemampuan kueri bahasa alami.
 - Tentukan keamanan tingkat baris: Memungkinkan Anda mengamankan data di tingkat tabel dan baris dengan menentukan filter keamanan. Anda dapat menentukan peran dan filter keamanan terkait di desktop Power BI lalu menetapkan anggota ke peran tersebut di layanan Power BI di app.powerbi.com.

Peran Excel dan Power BI

Excel 2013 adalah alat utama Microsoft untuk analisis bisnis. Ini mencakup fitur intelijen bisnis tambahan seperti Power Query, Power Pivot, Power View, dan Power Map. Dengan edisi Excel 2013, analisis bisnis kini dapat menerbitkan Buku Kerja Excel ke aplikasi situs web. powerbi.com untuk berbagi data, analisis, dan laporan dengan pengguna Power BI lainnya.

Dukungan untuk Visualisasi R

Dengan Power BI Desktop, Anda dapat memanfaatkan kekuatan bahasa pemrograman R untuk membantu memvisualisasikan data Anda. Untuk menggunakan R dengan Power BI Desktop, Anda harus menginstal mesin R terlebih dahulu. Untuk menjalankan Skrip R di Power BI Desktop, Anda harus menginstal R secara terpisah di komputer lokal Anda. Anda dapat mengunduh dan menginstal R secara gratis dari banyak lokasi, termasuk halaman unduh Revolution Open dan Repositori CRAN.

Microsoft R Terbuka

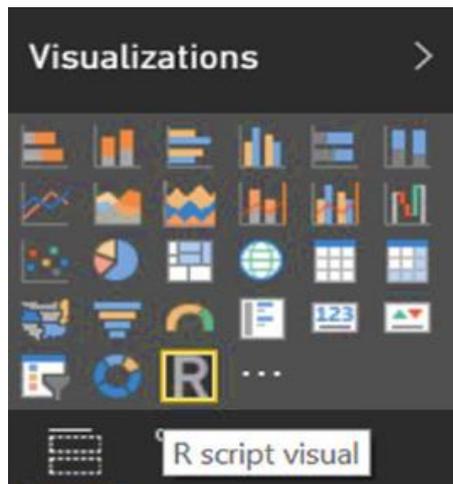
Microsoft R Open, sebelumnya dikenal sebagai Revolution R Open, adalah distribusi R yang disempurnakan dari Microsoft. Ini adalah platform R sumber terbuka lengkap untuk analisis statistik dan ilmu data. Microsoft R Open 3.3.2 (versi terkini pada saat penulisan ini) didasarkan pada (dan 100% kompatibel dengan) R-3.3.2, dan sepenuhnya kompatibel dengan semua paket, skrip, dan aplikasi yang bekerja dengan spesifik tersebut versi R. Microsoft R Open juga menyertakan kemampuan tambahan untuk meningkatkan kinerja pada platform berbasis Windows dan Linux.

Power BI Desktop: Visualisasi Skrip R

Setelah Anda menginstal dan mengonfigurasi Microsoft R Open agar berfungsi dengan Power BI Desktop, Anda dapat melanjutkan untuk membuat beberapa visual menakjubkan

dengan memanfaatkan kekuatan beberapa paket visualisasi R yang lebih populer seperti `ggplot2`.

“`ggplot2`” adalah paket visualisasi data yang kuat untuk digunakan dalam program R. Ini adalah implementasi dari “tata bahasa grafik”, yang merupakan pedoman umum untuk visualisasi data. Istilah ini berasal dari buku penting karya Leland Wilkinson dan ditulis pada awal tahun 2000-an. Skema tata bahasa grafis berupaya menguraikan grafik menjadi komponen semantik seperti skala dan lapisan. Setelah Anda berada di lingkungan Power BI Desktop, Anda dapat menyeret kontrol R Script Visual ke kanvas desainer, seperti yang diilustrasikan dalam Gambar 9.2.



Gambar 9.2. Kontrol visual Skrip R, di kotak alat Visualisasi Power BI

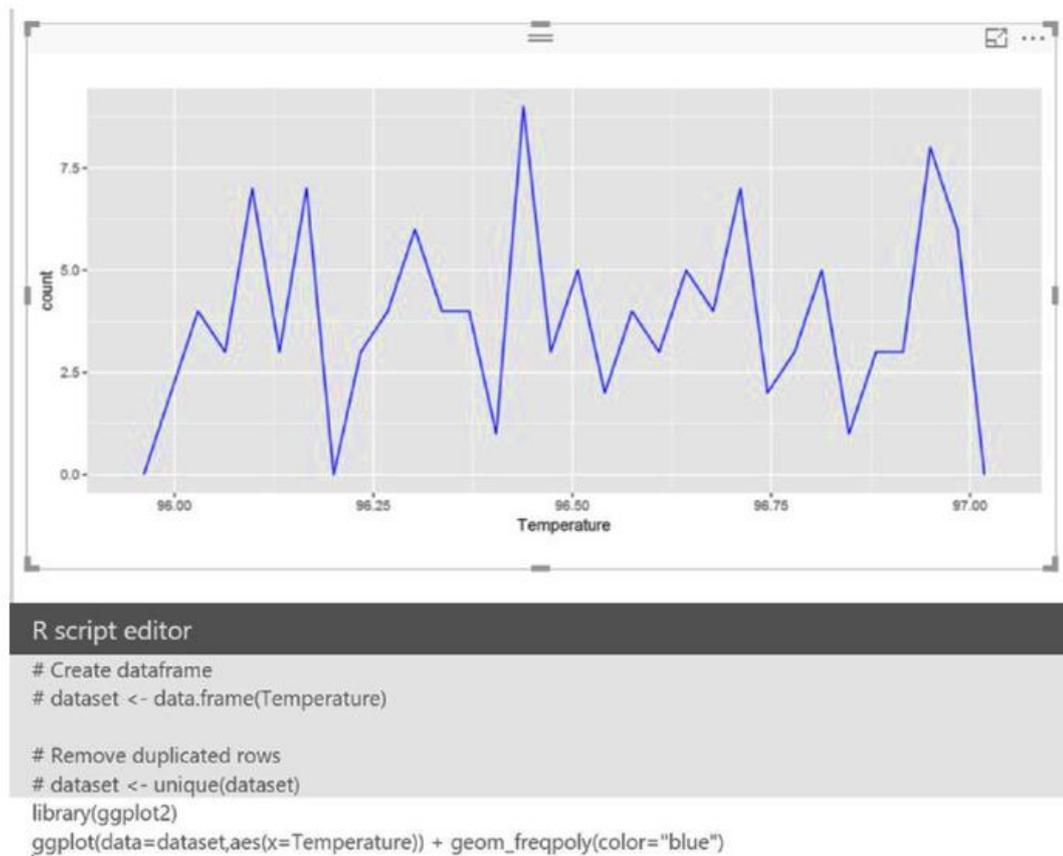
Setelah Anda menyeret kontrol visual R ke permukaan desainer Power BI, yang perlu Anda lakukan hanyalah:

1. Pilih kolom data yang akan diplot. Dalam contoh ini, kami menggunakan kolom Suhu.
2. Tambahkan baris kode R berikut ke kontrol visual Power BI R Script:

```
# LOAD the ggplot2 package
library(ggplot2)
```

```
# RENDER the ggplot2 Visual
ggplot(data=dataset, aes(x=Temperature)) + geom_freqpoly(color="blue")
```

Setelah mengisi kontrol dengan kode R dan mengklik panah Jalankan di R Script Editor, Anda akan melihat visualisasi grafis yang sangat kaya muncul. Dalam hal ini, poligon frekuensi geometris digambarkan, menunjukkan berbagai fluktuasi suhu dalam data sampel. Gambar 9.3 mengilustrasikan visualisasi kontrol Power BI R.



Gambar 9.3 Modul Power BI R menggunakan paket ggplot2 R untuk merender visualisasi pembacaan sensor suhu

Seperti yang dapat Anda lihat pada Gambar 9-3, Power BI membuatnya sangat mudah untuk memanfaatkan bahasa R dan pustaka paket yang luas untuk membuat visualisasi grafis yang sangat hebat. Perhatikan bahwa selain menyediakan visualisasi yang kaya, R juga dapat digunakan sebagai sumber input di desktop Power BI.

9.3 SUMBER DATA POWER BI

Power BI dapat terhubung ke berbagai sumber data cloud dan lokal, termasuk:

- * Solusi SaaS Populer: Seperti Salesforce, GitHub, dan Dynamics CRM.
- * Database Lokal: Power BI menawarkan konektivitas langsung ke layanan analisis SQL Server. Menggunakan solusi gateway, Power BI juga dapat menyambung ke solusi database lainnya.
- * Sumber Data Khusus: Power BI juga memiliki kemampuan untuk terhubung ke hampir semua sumber data melalui API ReST yang kuat. Ini adalah poin utama ekstensibilitas dan berarti Anda kini dapat menyediakan antarmuka khusus ke dalam sumber data perusahaan yang dipatenkan selain memanfaatkan layanan data eksternal tambahan.
- * Integrasi dengan Layanan Azure Lainnya: Power BI menyediakan integrasi erat dengan IoT Hub, Event Hub, SQL Azure, dan analitik aliran.

❄️ *File Power BI Desktop, Excel, dan CSV:* Buku kerja Excel dapat tersambung langsung ke Power BI.com dan juga dapat digunakan dengan Power BI Desktop.

- Power BI Desktop harus dianggap sebagai aplikasi pendamping layanan Power BI berbasis web.
- File Power BI Desktop dapat diunggah ke layanan Power BI, sama seperti file Excel atau file CSV.

Kekuatan BI Ponsel

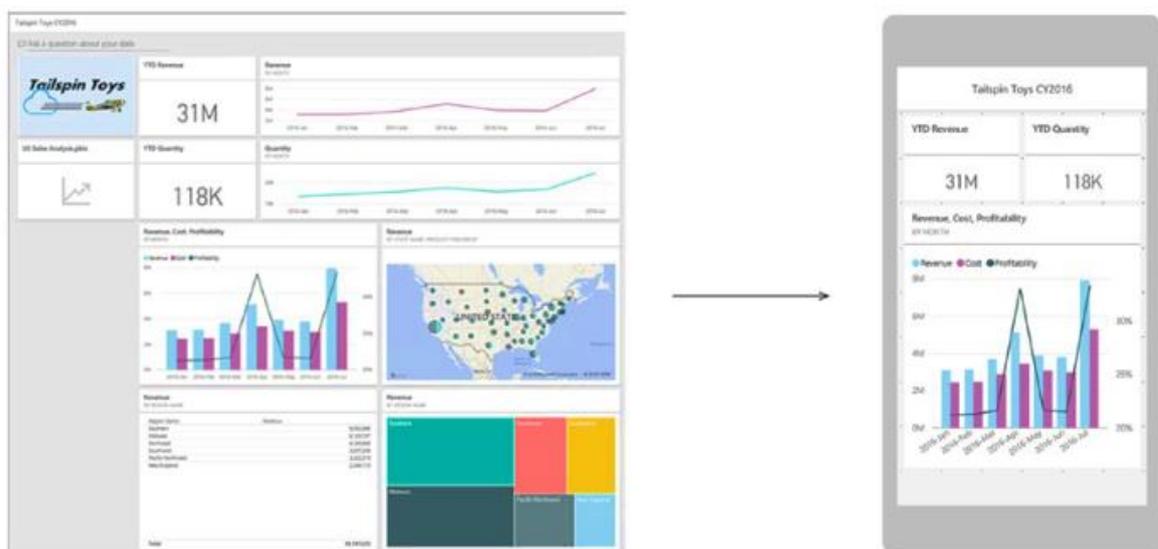
Salah satu fitur Power BI yang lebih canggih adalah kemampuan menginstal aplikasi seluler asli dengan cepat dan mudah di berbagai perangkat seluler, untuk memberdayakan pengguna organisasi agar dapat langsung mengakses dasbor Power BI. Aplikasi seluler Power BI saat ini tersedia di pasar seluler berikut:

- Apple iOS (iPhone dan iPad)
- Android
- Windows 10

Selain melihat dasbor Power BI saat bepergian, beberapa kemampuan tambahan Power BI yang telah diterapkan di semua platform seluler meliputi:

- Memungkinkan Anda menyetel “favorit” untuk visualisasi data penting.
- Menyediakan kemampuan untuk memperbesar/memperkecil visualisasi.
- Memungkinkan Anda membuat anotasi pada visualisasi dan berbagi cuplikan dengan orang lain di organisasi Anda.
- Memungkinkan Anda mengonfigurasi peringatan dengan mudah untuk menerima pemberitahuan ketika metrik bisnis penting mencapai ambang batas yang ditentukan.

Gambar 9-4 menggambarkan dasbor Power BI Desktop yang secara otomatis dirender ke dalam tampilan perangkat seluler.



Gambar 9.4 Tampilan Dasbor Power BI dari desktop/browser web dan di perangkat seluler

Laporan aplikasi seluler Power BI mendukung tampilan dasbor, laporan, laporan layanan pelaporan SQL Server, dan lembar bentang Excel. Aplikasi juga dapat mengakses data yang berada di lokasi dan di cloud dengan menyambungkan ke laporan layanan pelaporan SQL Server (on-premises) serta dasbor dan laporan Power BI (di cloud) melalui satu pengalaman aplikasi.

Power BI Tertanam

Power BI Embedded adalah layanan Azure yang memungkinkan pengembang memanfaatkan fitur Power BI dalam aplikasi kustom mereka sendiri. Artefak yang mungkin disematkan mencakup bagan, visualisasi, dan laporan. Salah satu keunggulan utama Power BI Embedded dibandingkan penawaran Power BI standar adalah dapat digunakan secara anonim, tanpa memerlukan informasi login akun apa pun untuk mengaksesnya.

Dengan Power BI Embedded, Anda dapat membuat dan menyebarkan visualisasi data yang menakjubkan dan laporan menarik langsung di aplikasi Anda menggunakan ReST API sederhana dan Power BI Embedded SDK. Fitur tambahan Power BI Embedded mencakup hal berikut:

- ✓ Power BI Embedded memberikan pengalaman pengguna autentikasi dan otorisasi yang bebas hambatan untuk skenario pelaporan yang disematkan.
- ✓ Pengembang aplikasi dapat dengan mudah menyematkan ubin dan laporan Power BI sebagai kontrol Iframe HTML di aplikasi seluler atau web.

API JavaScript Tertanam Power BI memungkinkan pengembang menyematkan laporan ke dalam aplikasi dan kemudian berinteraksi secara terprogram dengannya. Skenario umum untuk menggunakan Power BI Tertanam di aplikasi Anda meliputi:

- ✓ Vendor perangkat lunak independen (ISV) dan pengembang yang membangun aplikasi yang dapat diakses oleh pelanggan dapat menggunakan layanan Power BI Embedded, dan Power BI SDK, untuk menyematkan laporan interaktif.
- ✓ Sebagai pengembang, Anda dapat menggunakan kerangka visualisasi Power BI untuk membuat visualisasi khusus yang dapat digunakan di aplikasi Anda sendiri.
- ✓ Dengan memanfaatkan model autentikasi token aplikasi Azure Active Directory (AAD), Anda dapat menyematkan laporan interaktif yang dibuat di alat Power BI Desktop ke dalam aplikasi Anda sendiri.

Model Lisensi untuk Microsoft Power BI Tertanam

Lisensi untuk Power BI yang disematkan merupakan tanggung jawab pengembang aplikasi yang menggunakan visual Power BI, dan biaya dibebankan pada langganan yang memiliki sumber daya tersebut.

API Power BI ReST

API Power BI ReST memungkinkan pengembang memperluas aplikasi mereka secara terprogram untuk mengirimkan muatan data khusus dan real-time yang mendorong skenario dasbor utama. Berikut adalah beberapa kasus penggunaan utama untuk Power BI ReST API.

- Pengembang dapat menggunakan Power BI ReST API untuk secara terprogram mendorong data statis dan real-time langsung dari aplikasi ke Power BI.

- Power BI ReST API dapat memberikan akses terprogram ke sumber daya Power BI lainnya seperti himpunan data, tabel, dan skema.

Visual Kustom Power BI

Power BI adalah platform terbuka dan dapat diperluas. Anda dapat mengunduh atau mengembangkan visual khusus sesuai kebutuhan untuk mendukung kebutuhan bisnis tertentu. Untuk membuat visual kustom, pengembang dapat memulai dengan cepat menggunakan basis kode visualisasi Power BI sampel sumber terbuka dan berkualitas produksi dari Microsoft.

Selain rangkaian kontrol visual Power BI yang kaya (25+), pengembang juga memiliki opsi berikut untuk mendapatkan visualisasi Power BI tambahan:

- Galeri Visual Kustom Power BI:
 - ✓ Ini adalah galeri visual (gratis) yang dibuat oleh komunitas Power BI.
 - ✓ Anda juga dapat menginstal visual Power BI yang didukung R. Lihat tautan ini untuk informasi lebih lanjut: <https://app.powerbi.com/visuals/>.
- Buat Visual Kustom Power BI Anda Sendiri:
 - ✓ Pengembang juga memiliki opsi untuk membuat visualisasi Power BI kustom mereka sendiri untuk digunakan di dasbor, laporan, dan paket konten.
 - ✓ Proyek visual Microsoft Power BI menyediakan visualisasi data berkualitas tinggi yang dapat Anda gunakan untuk memperluas Power BI.
 - ✓ Proyek ini berisi lebih dari 20 jenis visualisasi, kerangka kerja untuk menjalankannya, dan infrastruktur pengujian yang memungkinkan Anda membuat visualisasi berkualitas tinggi. Lihat <https://powerbi.microsoft.com/custom-visuals>.
 - ✓ Microsoft telah menerbitkan kode sumber untuk semua visualisasi Power BI ke GitHub di <https://github.com/Microsoft/PowerBI-Visuals/>.
 - ✓ Microsoft memungkinkan pengembang membuat visual kustom mereka sendiri dari visual sumber terbuka yang dipublikasikan bersama dengan kerangka kerja yang disediakan, lalu mengirimkannya kembali ke komunitas.

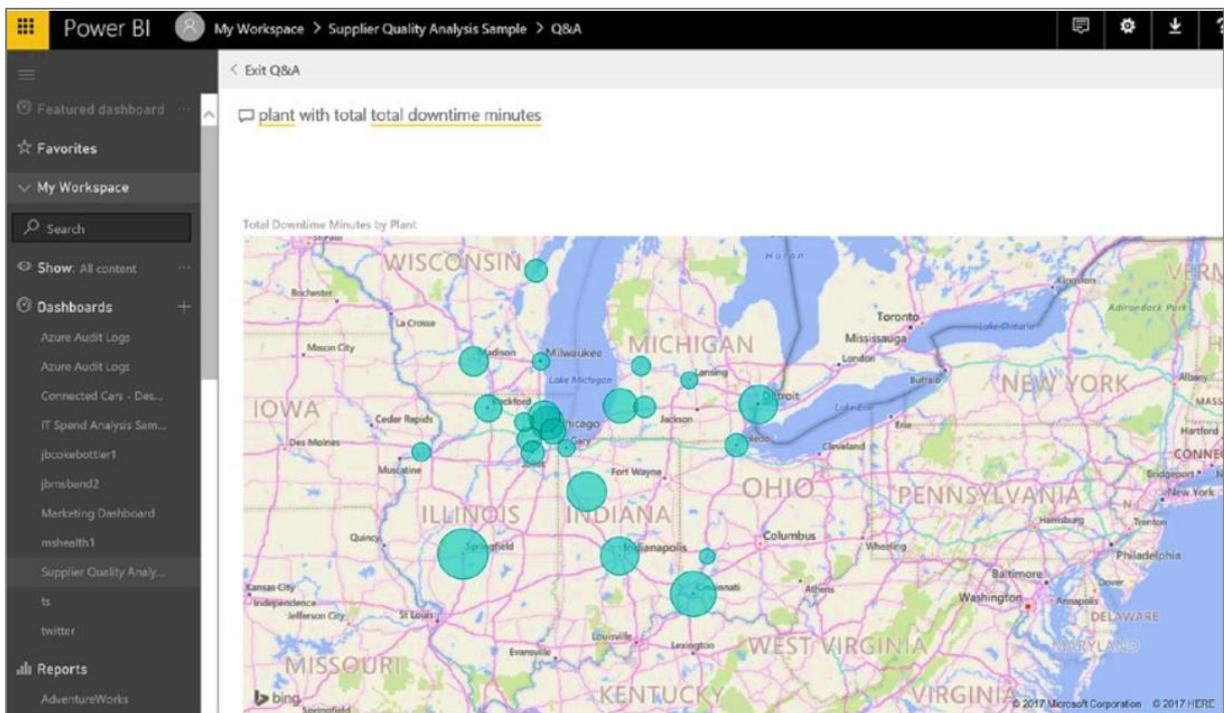
Kueri Bahasa Alami Power BI

Salah satu fitur unik Power BI adalah kemampuan untuk mengajukan pertanyaan tentang data Anda menggunakan bahasa kueri alami yang dikenal sebagai Tanya Jawab. Tanya Jawab Power BI berbeda dengan mesin pencari karena Tanya Jawab hanya memberikan hasil tentang data yang dihosting di Power BI. Anda dapat menggunakan Tanya Jawab Power BI untuk menjelajahi data Anda menggunakan pernyataan seperti "Tampilkan Toko dengan penjualan lebih dari 50.000" lalu menerima jawaban dalam bentuk bagan dan grafik. Tanya Jawab Power BI juga memilih visualisasi terbaik berdasarkan tipe data dasar yang ditampilkan.

Kotak pertanyaan Tanya Jawab Power BI adalah tempat Anda mengetik pertanyaan menggunakan bahasa alami. Tanya Jawab Power BI mengenali kata-kata yang Anda ketikkan dan kemudian secara otomatis menentukan himpunan data mana yang akan dikueri untuk menemukan jawabannya. Tanya Jawab Power BI juga membantu Anda menyusun pertanyaan dengan benar dengan pelengkapan otomatis, pernyataan ulang, dan bantuan tekstual dan visual lainnya. Jawaban atas pertanyaan Anda ditampilkan sebagai visualisasi interaktif dan

diperbarui secara otomatis saat Anda mengubah pertanyaan. Tanya Jawab Power BI dapat dengan cerdas memfilter, mengurutkan, meringkas, dan menampilkan data berdasarkan pertanyaan.

Bergantung pada himpunan data yang mendasarinya, layanan Tanya Jawab Power BI dapat menentukan cara terbaik untuk menampilkannya. Misalnya, jika data didefinisikan sebagai kota atau negara bagian, kemungkinan besar akan ditampilkan sebagai visualisasi peta. Setelah jawaban Tanya Jawab Power BI ditampilkan, Anda dapat menyematkan jawaban ke dasbor untuk referensi di masa mendatang. Bahkan dimungkinkan untuk menambahkan pertanyaan yang Anda sarankan sendiri untuk setiap kumpulan data. Sebagai contoh, Anda dapat membuat serangkaian pertanyaan umum untuk membantu meminta pengguna kumpulan data Power BI Anda. Gambar 9-5 menggambarkan tangkapan layar layar Tanya Jawab Power BI untuk dasbor sampel Analisis Kualitas Pemasok, yang tersedia dari layanan app.powerbi.com.



Gambar 9.5 Tanya Jawab Power BI menunjukkan jumlah total cacat yang ditolak

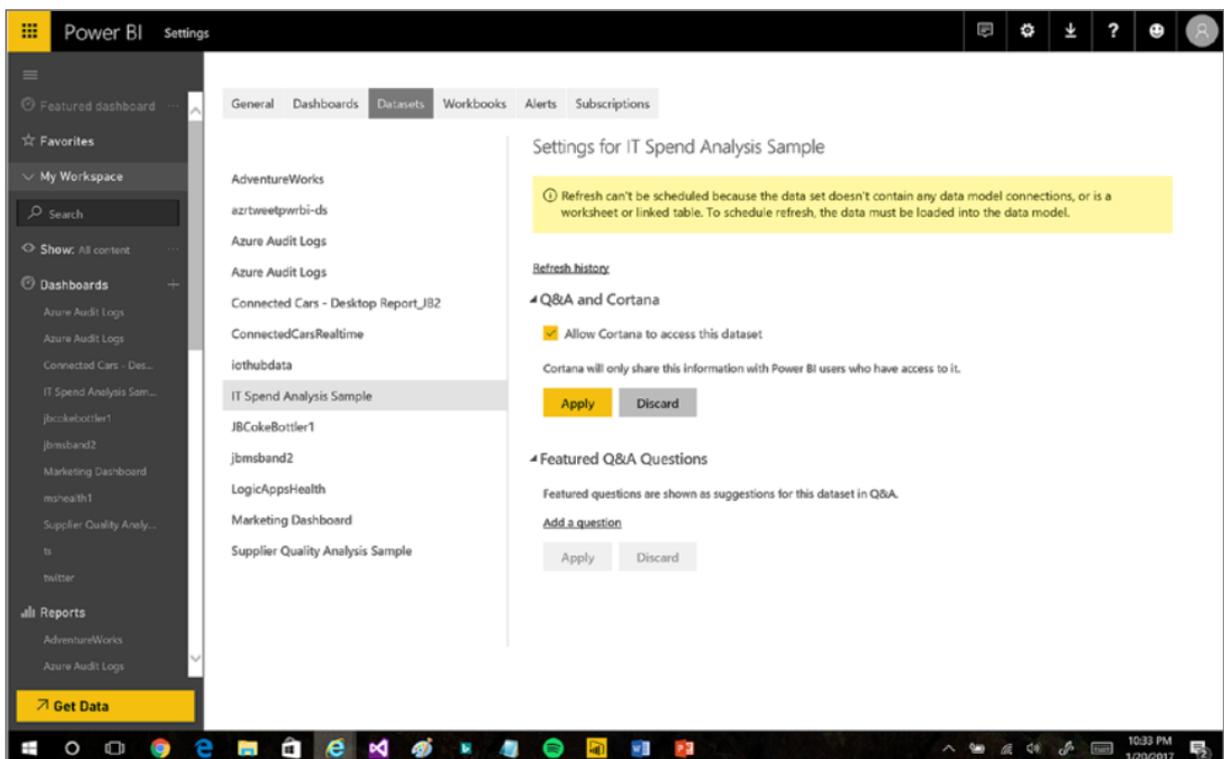
Dalam contoh ini, kami meminta “pabrik dengan total waktu henti total” dan dengan cepat menerima peta visual pada Gambar 9-5. Perhatikan petunjuk visual yang diberikan oleh ukuran lingkaran pada peta, yang menunjukkan lamanya waktu henti relatif untuk masing-masing pabrik.

Integrasi Power BI Cortana

Microsoft baru-baru ini mengintegrasikan Cortana (asisten digital pribadi) ke dalam Power BI untuk membantu manusia mencari jawaban atas data mereka. Integrasi suara Cortana bekerja secara lancar dengan dasbor Power BI dan memungkinkan pengguna mengajukan pertanyaan tentang data mereka—mirip dengan fitur Tanya Jawab. Pengguna

dapat mengajukan pertanyaan kepada Cortana baik secara lisan atau dengan mengetikkannya seperti “pabrik apa yang memiliki laporan kerusakan paling banyak”. Cortana kemudian dapat menemukan jawaban langsung dari kumpulan data atau dari halaman laporan Power BI yang dirancang khusus untuk Cortana (ini disebut Halaman Jawaban). Perhatikan bahwa ada beberapa prasyarat yang diperlukan sebelum Anda bisa mulai menggunakan Cortana dengan Power BI:

- Pengguna memiliki Windows 10 versi 1511 atau lebih baru (pembaruan November 2015).
- Pengguna harus menambahkan akun Power BI mereka ke Windows 10.
- Kumpulan data mendukung Tanya Jawab dan diaktifkan untuk diakses oleh Cortana. Lihat Gambar 9-6.



Gambar 9.6 Pengaturan Power BI untuk Mengizinkan Tanya Jawab dan Integrasi Cortana

Perhatikan bahwa Anda juga dapat dengan mudah menambahkan saran pertanyaan Tanya Jawab. Cortana akan mengembalikan dan memberi peringkat pada jawaban dari Power BI, memberikan satu atau lebih hasil yang paling cocok. Anda dapat terus berinteraksi dengan visualisasi yang dikembalikan seperti yang Anda lakukan di Power BI.

9.4 ARSITEKTUR BIAYA PELAPORAN CLOUD

Dalam banyak kasus, topik model lisensi untuk pelaporan, dasbor, dan visualisasi dapat berdampak signifikan pada detail implementasi akhir arsitektur cloud modern. Memperhatikan ungkapan “jarak tempuh Anda mungkin berbeda-beda”, perlu diingat bahwa

eksplorasi dan analisis lebih dalam sering kali disarankan untuk menentukan pendekatan terbaik dan paling hemat biaya untuk organisasi Anda.

Berbagai strategi untuk menerapkan dasbor, visualisasi, dan layanan pelaporan berbasis cloud untuk organisasi dapat terdiri dari opsi berikut:

- *Power BI Desktop/Seluler:*
 - Memungkinkan Anda dengan cepat mengonfigurasi dan mengaktifkan kemampuan BI layanan mandiri di suatu organisasi.
 - Model lisensi per pengguna.
- *Power BI Tertanam:*
 - Memberikan kemampuan untuk menyematkan visualisasi dan pelaporan Power BI langsung ke aplikasi kustom Anda.
 - Terbatas pada desain statis dan interaktivitas.
 - Dilisensikan berdasarkan penggunaan dalam aplikasi.
- *Aplikasi Khusus:*
 - Aplikasi web/seluler.
 - Dibutuhkan staf pengembangan.
 - Implementasi yang sepenuhnya disesuaikan.
- *Alat Pelaporan Pihak Ketiga:*
 - Seperti QlikView, Tableau, dan Spotfire.
 - Biaya perizinan tambahan.

Terkadang jawaban yang benar bahkan bisa berupa jawaban ganda. Misalnya, arsitektur pelaporan suatu organisasi dapat mencakup campuran opsi-opsi ini dengan kemampuan untuk sepenuhnya memanfaatkan setiap opsi untuk kasus penggunaan terbaiknya. Misalnya:

- ✓ Power BI: Dapat dengan cepat memberdayakan pengguna “power” yang paling menuntut organisasi Anda. Dengan mengaktifkan kemampuan BI layanan mandiri, organisasi dapat meringankan tumpukan permintaan pelaporan untuk staf TI.
- ✓ Power BI Tertanam: Menerapkannya jika skenario pelaporan berbasis web statis atau anonim diperlukan.
- ✓ Pengembangan Aplikasi Khusus: Memberikan kontrol, rendering, dan memformat. Pertimbangkan opsi ini ketika paket pelaporan yang tersedia tidak dapat digunakan.

Peringatan dan Pemberitahuan

Diskusi tentang lanskap pelaporan modern saat ini tidak akan lengkap tanpa menyebutkan peringatan dan pemberitahuan. Salah satu aspek terpenting dalam penerapan dan pengelolaan aplikasi cloud Big Data adalah kemampuan untuk menyediakan “manajemen dengan pengecualian” dan dengan demikian fokus pada aktivitas bernilai lebih tinggi seperti menyempurnakan algoritme Pembelajaran Mesin.

Seperti yang telah kita lihat, memiliki dasbor operasi yang mengagumkan dan menakjubkan secara grafis di Power BI adalah hal yang bagus, selama ada yang menontonnya! Peringatan dan notifikasi memungkinkan Anda mendapat informasi dan dihubungi secara otomatis ketika ambang batas atau metrik KPI tertentu telah tercapai. Di sinilah konsep actionable Intelligence menjadi pembeda bisnis.

Dalam lingkungan bisnis yang serba cepat saat ini, berkomunikasi dengan banyak orang pada saat yang sama sangatlah menantang. Ditambah lagi dengan banyaknya perangkat dan protokol komunikasi seperti email, SMS, dan pesan suara. Aspek lain yang perlu dipertimbangkan adalah “jadwal eskalasi” terkait yang dapat dikaitkan dengan metrik apa pun yang mungkin memicu alarm. Ini mengacu pada jadwal untuk menghubungi individu menggunakan protokol dan perangkat berbeda, bergantung pada tingkat keparahan alarm dan jumlah waktu yang berlalu sejak pemberitahuan terakhir.

Dalam hal konsistensi, beberapa karyawan sering memeriksa email mereka, beberapa beberapa kali sehari, dan beberapa hanya secara tidak teratur. Sebagian besar karyawan saat ini memiliki ponsel pintar dengan kemampuan pesan teks SMS. Dalam hal protokol komunikasi, persamaannya adalah teknologi yang paling banyak digunakan, tersebar luas, dan diadopsi secara luas. Hasilnya, email, pesan teks SMS, dan peringatan web dengan cepat menduduki posisi teratas sebagai pilihan terbaik untuk pemberitahuan yang akan menjangkau basis terbesar pengguna yang ada.

Fitur menarik dari sistem notifikasi yang hebat adalah sistem ini menyederhanakan komunikasi dengan memperingatkan semua karyawan tentang keadaan darurat melalui berbagai saluran komunikasi pilihan mereka. Proses ini akan terjadi dalam beberapa detik sejak pesan peringatan awal dikirim. Artinya, apa pun perangkat yang dimiliki karyawan Anda, dia dapat menerima pesan dan peringatan penting dan mendesak dari Anda secara tepat waktu.

Microsoft Azure memiliki beberapa kemampuan layanan terkelola hebat yang membuat penanganan jenis pemberitahuan dan peringatan ini lebih mudah dari sebelumnya. Nanti di bab ini, kita akan menjelajahi penggunaan Azure Event Hubs untuk menerima pesan peringatan dari implementasi referensi kami dan kemudian mengonfigurasi Azure Functions agar secara otomatis “dipicu” setiap kali pesan baru tiba di Event Hub. Membangun sistem peringatan dan pemberitahuan yang kuat memungkinkan penyiaran berita penting dan mendesak mengenai (potensi) keadaan darurat di tempat kerja secara instan. Hasilnya, sistem peringatan yang dirancang dengan baik dapat menjadi aset yang sangat berharga bagi bisnis, pelanggan, dan karyawannya. Di bagian ini, kami memeriksa teknologi Azure berikut untuk mencapai tujuan implementasi referensi kami:

- ✓ *Azure Event Hubs*: Layanan pesan terkelola dan berskala besar yang dapat menyediakan aliran data besar-besaran dari berbagai sumber termasuk, aplikasi, perangkat, dan situs web.
- ✓ *Azure Stream Analytics*: Kami akan mengunjungi kembali Azure Stream Analytics secara singkat dan mengeksplorasi penambahan opsi output tambahan untuk Event Hubs.
 - Selain memiliki “jalur panas” untuk metrik dasbor Power BI real-time untuk implementasi referensi kami, kami juga akan menerapkan “jalur panas” ke Azure Event Hub untuk menghapus pesan peringatan dan pemberitahuan yang memerlukan tindakan segera.

- ✓ *Azure Functions*: Penawaran PaaS ini adalah platform komputasi awan yang sangat ringan, berbasis peristiwa, tanpa server.
 - Azure Functions dapat dikonfigurasi dan diberi kode dengan cepat dan mudah untuk mempercepat pengembangan solusi pemberitahuan dan pemberitahuan.
 - Mereka menyediakan “pengait” terprogram ke dalam peristiwa populer berbasis Azure untuk menerapkan skenario integrasi instan.

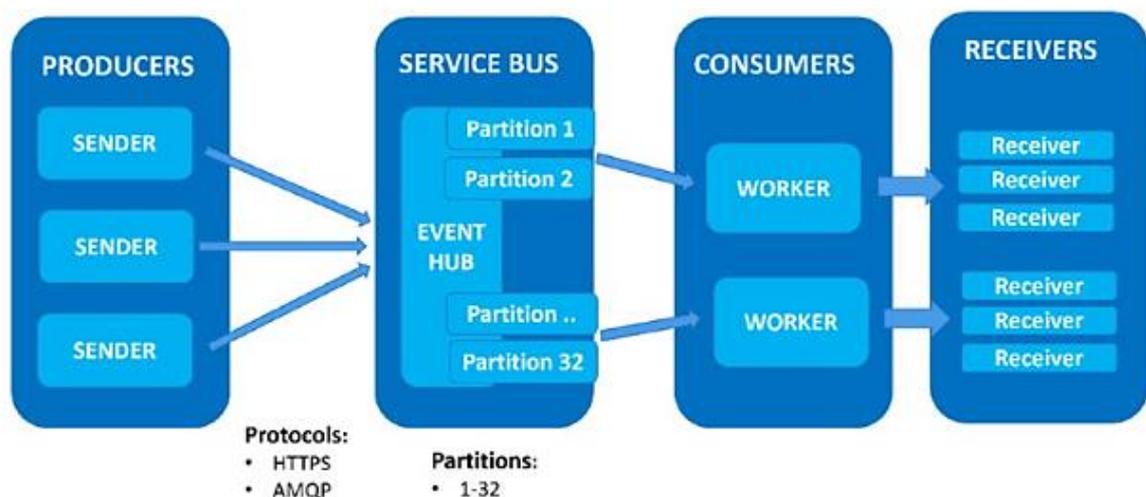
Pusat Acara Azure

Azure Event Hub adalah layanan penyerapan data yang sangat skalabel dan dapat menskalakan hingga jutaan peristiwa per detik, yang berasal dari beragam perangkat dan layanan, dengan latensi rendah dan keandalan tinggi. Kami akan mempertimbangkan penggunaan layanan Azure Event Hub sebagai antrian pesan utama untuk melakukan peringatan keluar dan pemberitahuan untuk implementasi referensi kami.

Tujuan desain Event Hubs mencakup pengelolaan apa yang terkadang disebut sebagai “tiga V” pemrosesan Big Data:

- *Volume*: Kemampuan untuk menangani data dalam jumlah besar.
- *Kecepatan*: Kemampuan untuk menangani penyerapan data dan keluarnya data dalam skala besar.
- *Variasi*: Kemampuan untuk menangani berbagai jenis data dan sumber data.

Microsoft Azure Event Hubs dapat mendukung hingga 1 juta pelanggan dan mendukung ribuan gigabyte data masuk. Gambar 9-7 mengilustrasikan arsitektur Azure Event Hub.



Gambar 9.7 Arsitektur Azure Event Hub

Penerbit Acara

Aplikasi apa pun yang mengirimkan data ke Azure Event Hub disebut sebagai penerbit peristiwa. Penerbit peristiwa dapat memancarkan peristiwa menggunakan AMQP, AMQP melalui protokol WebSockets dan HTTP. Saat mengirim pesan, penerbit dapat menggunakan token Shared Access Signature (SAS) untuk mengautentikasi ke Event Hub.

Partisi Hub Acara

Azure Event Hubs menyediakan fungsionalitas streaming pesan melalui pola konsumen yang dipartisi. Partisi adalah urutan kejadian yang diselenggarakan di Event Hub. Saat peristiwa baru tiba, peristiwa tersebut ditambahkan ke akhir rangkaian ini. Pola ini mengharuskan setiap konsumen hanya membaca subset tertentu dari aliran pesan. Pola partisi ini juga memungkinkan skalabilitas horizontal untuk pemrosesan peristiwa. Jumlah partisi Azure Event Hub ditentukan pada saat provisi awal di Azure dan dapat memiliki nilai antara 2 dan 32. Jumlah partisi yang ditentukan di Azure Event Hub secara langsung berkaitan dengan jumlah “pekerja” bersamaan yang akan ditugaskan untuk memproses pesan di latar belakang.

Konsumen Acara dan Kelompok Konsumen

Aplikasi apa pun yang membaca data dari Event Hub disebut sebagai konsumen peristiwa. Semua konsumen Event Hub terhubung melalui protokol sesi dan acara dikirimkan melalui sesi saat tersedia. Klien tidak perlu melakukan polling untuk ketersediaan data. Kemampuan terbitkan/berlangganan di Azure Event Hubs diaktifkan melalui gagasan grup konsumen. Grup konsumen didefinisikan sebagai tampilan (penanda, posisi, atau offset) konten data pesan di Event Hub. Grup konsumen memungkinkan beberapa aplikasi yang memakan waktu untuk masing-masing memiliki potongan atau tampilan terpisah dari aliran data hub peristiwa, dan kemudian dapat membaca dan memproses data aliran peristiwa dengan cara yang sepenuhnya terisolasi dan independen.

Unit Throughput

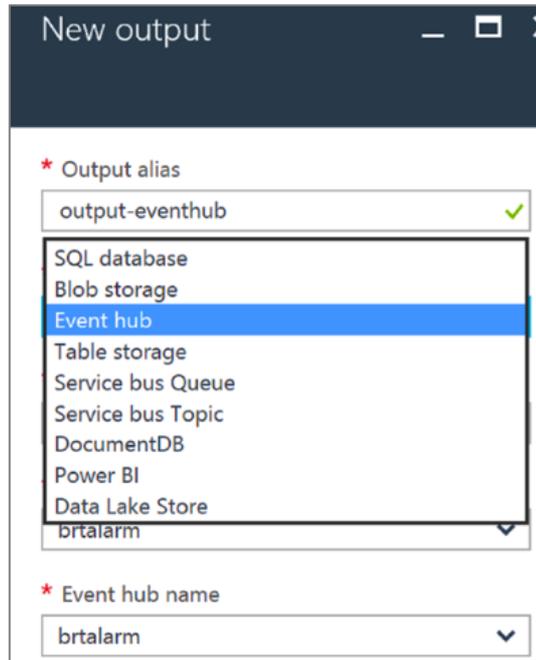
Kapasitas pemrosesan Azure Event Hubs diukur dalam satuan throughput. Unit throughput sama dengan unit kapasitas pemrosesan pesan Azure Event Hub yang dibeli sebelumnya dan setiap unit mencakup komponen pemrosesan berikut:

- ❖ Masuknya Data: Hingga 1MB per detik atau 1000 peristiwa per detik (mana saja yang lebih dulu) untuk operasi masuk.
- ❖ Data Egress: Hingga 2MB per detik untuk operasi keluar.

Praktik terbaik untuk mengoptimalkan performa Event Hubs adalah dengan menyeimbangkan unit throughput dan partisi untuk mencapai arsitektur penskalaan optimal di Azure. Sebagai aturan umum, jumlah unit throughput harus kurang dari atau sama dengan jumlah partisi di Event Hub.

Analisis Streaming: Output ke Event Hub

Sekarang setelah kita memiliki pemahaman yang baik tentang Azure Event Hubs, kita akan memeriksa kembali Azure Stream Analytics dan bagaimana hal itu dapat berintegrasi dengan Azure Event Hubs untuk memicu alur kerja pemberitahuan atau pemberitahuan. Gambar 9-8 mengilustrasikan berbagai opsi output yang tersedia untuk pekerjaan Azure Stream Analytics. Perhatikan bahwa salah satu opsi output ini adalah untuk Event Hub.



Gambar 9.8. Opsi keluaran Azure Streaming Analytics untuk pusat peristiwa

Anda sekarang memiliki kemampuan untuk menentukan jalur aktif lainnya untuk pekerjaan Analisis Streaming kami (selain Power BI) untuk peringatan waktu nyata. Jalur tambahan ini memungkinkan Anda untuk:

- 1) Pantau aliran data sensor yang masuk.
- 2) Panggil Layanan Web Pembelajaran Mesin untuk memprediksi apakah anggota tim mungkin kelelahan secara fisik.
- 3) Letakkan pesan peringatan atau pemberitahuan ke Azure Event Hub jika diperkirakan akan habis.
- 4) Melakukan aktivitas sebagai respons terhadap peringatan atau pemberitahuan, seperti mengirim email atau SMS. Azure Functions, topik pada bagian selanjutnya, adalah pendekatan yang direkomendasikan untuk mengimplementasikan tindakan tersebut.

9.5 FUNGSI AZURE

Azure Functions memberikan pengalaman komputasi yang sangat ringan, tanpa server, dan berbasis peristiwa yang dapat membantu mempercepat produktivitas pengembangan dan penerapan Anda. Pengembangan aplikasi dipercepat karena tidak ada penyediaan, konfigurasi, atau manajemen server yang terlibat. Fokusnya secara eksklusif adalah menyempurnakan alur kerja dan logika bisnis untuk menghubungkan peristiwa ke tindakan di Azure versus bagaimana peristiwa tersebut akan dihosting dan diskalakan secara fungsional.

Azure Functions menyediakan kaitan ke banyak peristiwa dan sumber berguna sebagai input dan output. Ini termasuk pemicu peristiwa yang telah ditentukan sebelumnya seperti permintaan HTTP, pembaruan penyimpanan, pengiriman antrian, dan aliran peristiwa, memungkinkan Anda dengan cepat membangun solusi dengan lebih sedikit kode.

Ikatan Masukan

Berikut adalah daftar berbagai jenis peristiwa INPUT yang dapat dipicu di Azure Functions (saat tulisan ini dibuat):

- 1) *Blob Trigger*: Fungsi yang akan dijalankan setiap kali blob ditambahkan ke kontainer tertentu.
- 2) *Pemicu Event Hub*: Fungsi yang akan dijalankan setiap kali hub acara menerima acara baru.
- 3) *Pemicu File Eksternal*: Fungsi yang akan dijalankan setiap kali file ditambahkan ke penyedia File Eksternal.
- 4) *Tabel Eksternal*: Fungsi yang mengambil entitas dari Tabel Eksternal ketika menerima permintaan HTTP.
- 5) *Pencari Lokasi Wajah*: Fungsi yang memproses gambar dan menghasilkan keluaran persegi panjang pembatas wajah.
- 6) *Generic Web Hook*: Fungsi yang akan dijalankan setiap kali menerima permintaan web hook.
- 7) *GitHub Commenter*: Fungsi yang akan dijalankan setiap kali menerima permintaan web hook GitHub Commenter.
- 8) *GitHub Web Hook*: Fungsi yang akan dijalankan setiap kali menerima permintaan web hook GitHub.
- 9) *Http GET (CRUD)*: Fungsi yang mengambil entitas dari Tabel Penyimpanan ketika menerima permintaan HTTP.
- 10) *Http POST (CRUD)*: Fungsi yang menambahkan entitas ke Tabel Penyimpanan ketika menerima permintaan HTTP.
- 11) *Http PUT (CRUD)*: Fungsi yang menambahkan entitas ke Tabel Penyimpanan ketika menerima permintaan HTTP.
- 12) *Pemicu Http*: Fungsi yang akan dijalankan setiap kali menerima permintaan HTTP.
- 13) *Image Resizer*: Fungsi yang membuat gambar diubah ukurannya setiap kali blob ditambahkan ke kontainer tertentu.
- 14) *Pemicu Manual*: Fungsi yang dipicu secara manual melalui tombol Portal Run.
- 15) *Pemicu Antrean*: Fungsi yang akan dijalankan setiap kali pesan ditambahkan ke Antrean Azure Storage yang ditentukan.
- 16) *Token SAS*: Fungsi yang menghasilkan token SAS untuk Azure Storage untuk nama kontainer dan blob tertentu.
- 17) *Surat Terjadwal*: Fungsi yang akan mengirim email secara berkala.
- 18) *Send Grid*: Fungsi yang mengirimkan email konfirmasi ketika item baru ditambahkan ke antrian tertentu.
- 19) *Pemicu Antrean Bus Layanan*: Fungsi yang akan dijalankan setiap kali pesan ditambahkan ke antrean Bus Layanan tertentu.
- 20) *Pemicu Topik Bus Layanan*: Fungsi yang akan dijalankan setiap kali pesan ditambahkan ke topik Bus Layanan yang ditentukan.

21) *Timer Trigger*: Fungsi yang akan dijalankan pada jadwal tertentu.

Ikatan Eksternal

Berikut adalah daftar berbagai jenis peristiwa OUTPUT yang dapat diintegrasikan ke dalam Fungsi Azure (saat tulisan ini dibuat):

- 1) Pusat Acara Azure
- 2) Penyimpanan Antrean Azure
- 3) Penyimpanan Azure Blob
- 4) File Eksternal (Pratinjau)
- 5) Tabel Eksternal (Eksperimental)
- 6) HTTP
- 7) Bus Layanan Azure
- 8) Penyimpanan Tabel Azure
- 9) Dokumen Azure DocumentDB
- 10) Catatan Tabel Azure Mobile
- 11) Pusat Pemberitahuan Azure
- 12) SendGrid (Pratinjau)
- 13) Twilio SMS (Pratinjau)
- 14) Kerangka Bot

Pengembang BYOL (Bawakan Bahasa Anda Sendiri). Azure Functions dapat ditulis dalam berbagai bahasa pemrograman, seperti C#, Node.js, Bash, F#, PHP, PowerShell, dan PHP. Hasilnya, pengembang aplikasi dapat memanfaatkan keahlian pengembangan yang ada saat mengimplementasikan Azure Functions.

Anda dapat mengembangkan Azure Functions di berbagai alat, termasuk Portal Azure, Visual Studio, dan editor teks favorit Anda. Integrasi kontrol sumber memungkinkan skenario penerapan berkelanjutan, dan Anda dapat menguji dan men-debug kode fungsi Anda secara lokal menggunakan Visual Studio atau alat baris perintah.

Penskalaan Hemat Biaya

Microsoft menawarkan dua opsi hosting untuk Azure Functions: paket konsumsi dan paket layanan aplikasi. Pilihan paket hosting memengaruhi biaya dan skalabilitas solusi khusus Anda.

- *Pendekatan Rencana Konsumsi*: Dengan rencana ini, Fungsi Azure berjalan secara paralel di beberapa instans aplikasi yang secara otomatis diskalakan berdasarkan pemanfaatan sumber daya. Anda tidak perlu mencadangkan sumber daya dan Anda hanya akan dikenakan biaya berdasarkan jumlah eksekusi dan sumber daya yang benar-benar digunakan.
- *Pendekatan paket Layanan Aplikasi*: Dengan paket ini, Fungsi Azure berjalan pada mesin virtual khusus, dan Fungsi Azure dapat berbagi server dengan aplikasi lain yang berjalan di akun pengguna.
 - VM khusus dialokasikan ke aplikasi App Service Anda.
 - VM selalu tersedia baik kode sedang dieksekusi secara aktif atau tidak.

- Ini mungkin pilihan yang baik jika skenario Anda mengharuskan fungsi berjalan terus-menerus.

Integrasi DevOps dengan Azure Functions

Organisasi dapat menerapkan integrasi DevOps mendalam untuk mengaktifkan fungsionalitas utama seperti skenario penerapan berkelanjutan dengan memanfaatkan Azure Functions. Beberapa Azure Functions dapat digabungkan untuk menghubungkan ke peristiwa yang terkait dengan banyak aplikasi kontrol kode sumber populer seperti Visual Studio Team Services, GitHub, Bit Bucket, dan alat DevOps populer lainnya. Perhatikan bahwa ada beberapa peristiwa khusus GitHub dengan Azure Functions seperti GitHubCommentator dan GitHubWebHook. Kait peristiwa ini memberikan titik integrasi spesifik dengan repositori, tindakan, dan aktivitas GitHub.

Fungsi Terjadwal

Salah satu fitur Azure Functions yang lebih canggih adalah kemampuan untuk menjadwalkan pemanggilan Azure Functions pada interval waktu yang ditentukan.

Anda dapat menggunakan sintaksis pekerjaan CRON untuk memicu Fungsi Azure untuk menjalankan tugas seperti pembersihan data atau sinkronisasi database dengan sistem eksternal. Ekspresi CRON biasanya mencakup enam bidang: {menit}{jam} {hari} {bulan} {hari dalam seminggu} {Tahun}.

Sebagai contoh, pengaturan CRON untuk Fungsi Azure terjadwal yang dipicu setiap lima menit sekali adalah:

```
"schedule": "0 */5 * * * *"
```

Figure 9-9 depicts an Azure Function that is triggered by a scheduled timer.

```
public static async Task Run(TimerInfo myTimer, TraceWriter log)
{
    var str = ConfigurationManager.ConnectionStrings["sqldb_connection"].ConnectionString;
    using (SqlConnection conn = new SqlConnection(str))
    {
        conn.Open();
        var text = "DELETE from dbo.TODOItems WHERE Complete='True'";
        using (SqlCommand cmd = new SqlCommand(text, conn))
        {
            // Execute the command and log the # rows deleted.
            var rows = await cmd.ExecuteNonQueryAsync();
            log.Info($"{rows} rows were deleted");
        }
    }
}
```

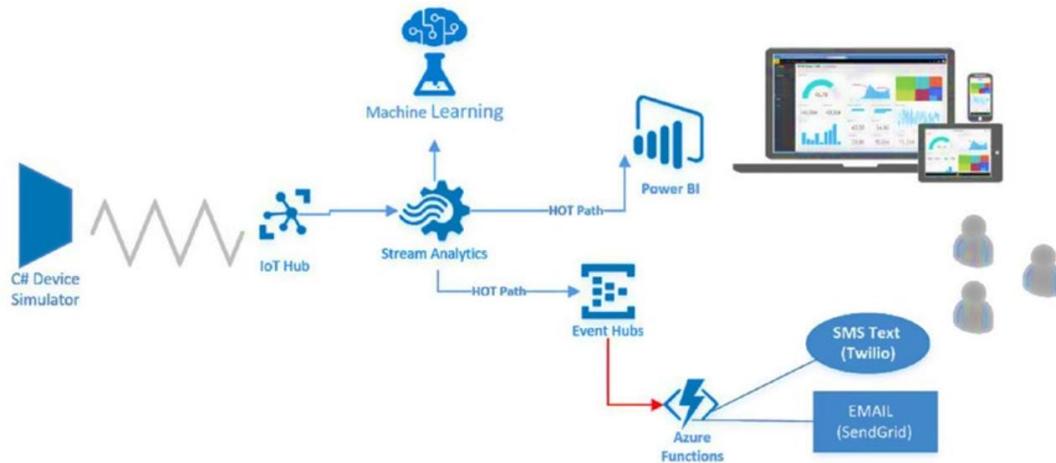
Gambar 9.9. Contoh Fungsi Azure untuk melakukan operasi pembersihan database

Referensi Implementasi

Hingga titik ini dalam bab ini, kami telah memberikan gambaran umum tentang fitur inti dan kemampuan Power BI, Azure Event Hubs, dan Azure Functions. Ini akan menjadi

informasi latar belakang yang berharga ketika kita meninjau kembali implementasi referensi dan memanfaatkan layanan Azure ini untuk solusinya.

Kami melanjutkan implementasi referensi teknis kami di bab ini dengan tahapan yang membahas tentang visualisasi, peringatan, dan notifikasi. Gambar 9.10 memberikan ikhtisar tingkat tinggi dari fungsionalitas yang akan kita terapkan pada bagian selanjutnya dari bab ini.



Gambar 9-10. Ikhtisar penerapan referensi untuk visualisasi, peringatan, dan notifikasi

Mari kita jelajahi komponen-komponen dalam solusi ini seperti yang digambarkan pada Gambar 9-10:

- ❖ C# Device Simulator: Ini adalah program utilitas yang dapat kita gunakan untuk menghasilkan pembacaan sensor acak dan mengirimkannya dengan aman ke Azure IoT Hub.
- ❖ IoT Hub: Ini adalah titik penyerapan utama untuk pembacaan sensor.
- ❖ Stream Analytics: Digunakan untuk memproses pembacaan sensor yang masuk dalam skala besar.
 - ✓ Layanan Web Machine Learning dipanggil dari Stream Analytics saat setiap rangkaian pembacaan anggota tim diproses, dan prediksi dibuat mengenai tingkat kelelahan fisik anggota tim.
 - ✓ Jalur Hot untuk output Power BI dan Event Hubs ditentukan; data dialirkan ke masing-masingnya secara real time.
- ❖ Power BI: Himpunan data diisi secara real-time dari layanan Azure Stream Analytics. Visualisasi grafis, laporan, dan dasbor kemudian dibuat dan dipaparkan kepada pengguna.
- ❖ Event Hubs: Digunakan untuk mengirimkan peringatan dan pemberitahuan keluar berprioritas tinggi.
 - ✓ Kueri Analisis Aliran akan memfilter output data ke Hub Peristiwa Alarm: berdasarkan hasil Panggilan Layanan Web Machine Learning.
 - Hanya hasil Positif untuk panggilan Layanan Web Machine Learning yang akan dimasukkan ke Hub Peristiwa Alarm.

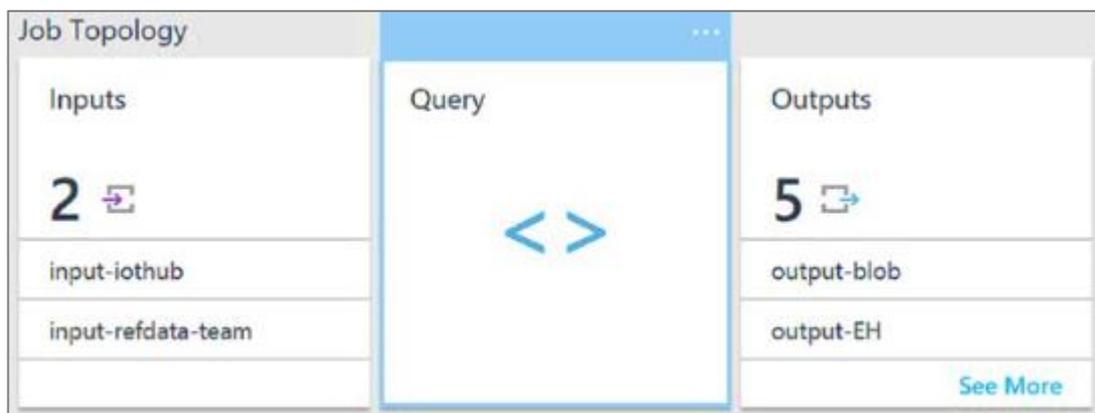
- Ini berarti bahwa seorang anggota tim berpotensi berada pada atau mendekati titik kelelahan fisik. Tindakan harus segera diambil!
- ❖ *Azure Functions*: Dipicu setiap kali item baru ditambahkan ke Alarm Event Hub. Peristiwa ini kemudian memicu Azure Functions terkait dan mengirim pesan sebagai SMS dan email melalui layanan pihak ketiga masing-masing sebagai Twilio dan SendGrid.
 - ✓ Kirim Pesan Uji SMS: Azure Functions dapat dikonfigurasi untuk mengirim pesan teks SMS secara otomatis melalui layanan komunikasi pihak ketiga (Twilio).
 - ✓ Pemberitahuan Email: Dapat dikonfigurasi di Azure Functions dan dikirim secara otomatis melalui layanan email pihak ketiga (SendGrid).

9.6 MODIFIKASI QUERY SQL STREAMING ANALYTICS

Di bagian ini, kita mengunjungi kembali kueri Azure Stream Analytics yang kita buat di Bab 5 dan membuat beberapa modifikasi.

Tinjau Output Kueri SQL Analisis Streaming yang Ada ke Power BI

Navigasikan ke pekerjaan Azure Stream Analytics Anda di Portal Azure dan klik pada panel Kueri, seperti yang ditunjukkan pada Gambar 9.11.



Gambar 9.11. Panel Kueri Azure Streaming Analytics

Anda akan melihat kueri ditampilkan di jendela editor. Bagian pertama dari kueri dicetak ulang di sini untuk referensi Anda.

```
-- *****
-- * HOTPATH - POWER BI
-- * Invoke Machine Learning As Function "ChkTeamHealth()"
-- * Via ASA SQL Subquery
-- * then output to PowerBI (Hot)/SQL (WARM)/BLOB (COLD)
-- *****
WITH [subquery] AS
(
    SELECT UserId,
    [Timestamp],
```

```

EventProcessedUtcTime,
PartitionId,
EventEnqueuedUtcTime,
    ChkTeamHealth(
    UserId,
    BreathingRate,
    Temperature,
    Ventilization,
    Activity,
    HeartRateBPM,
    Cadence,
    Velocity,
    Speed,
    HIB,
    HeartrateRedZone,
    HeartrateVariability,
    "N")
as result from [input-iothub]
TIMESTAMP BY [Timestamp]
)
SELECT  SQ.UserId,
        SQ.[Timestamp],
        SQ.EventProcessedUtcTime,
        SQ.PartitionId,
        SQ.EventEnqueuedUtcTime,
        CAST(SQ.result.[BreathingRate] as float) as [BreathingRate],
        CAST(SQ.result.[Temperature] as float) as [Temperature],
        CAST(SQ.result.[Ventilization] as float) as [Ventilization],
        CAST(SQ.result.[Activity] as float) as [Activity],
        CAST(SQ.result.[HeartRateBPM] as bigint) as [HeartRateBPM],
        CAST(SQ.result.[Cadence] as float) as [Cadence],
        CAST(SQ.result.[Velocity] as float) as [Velocity],
        CAST(SQ.result.[Speed] as float) as [Speed],
        SQ.result.[HIB],
        SQ.result.[HeartrateRedZone],
        SQ.result.[HeartrateVariability],
        SQ.result.[Scored labels],
        SQ.result.[Scored Probabilities],
        RF.id,
        RF.companynname,
        RF.imageUrl,
        RF.firstname,
        RF.lastname,
        RF.username,
        RF.[type],
        RF.[phone],
        RF.[email],
        AD.firstname as [adminfname],
        AD.lastname as [adminlname],
        AD.username as [adminunname],

```

```

        AD.type as [admintype],
        AD.phone as [adminphone],
        AD.email as [adminemail]

INTO [output-powerbi]

FROM subquery SQ

JOIN [input-refdata-team] RF
ON SQ.UserId = RF.id

JOIN [input-refdata-team] AD
ON RF.companyname = AD.companyname
WHERE AD.[type] = '2'

```

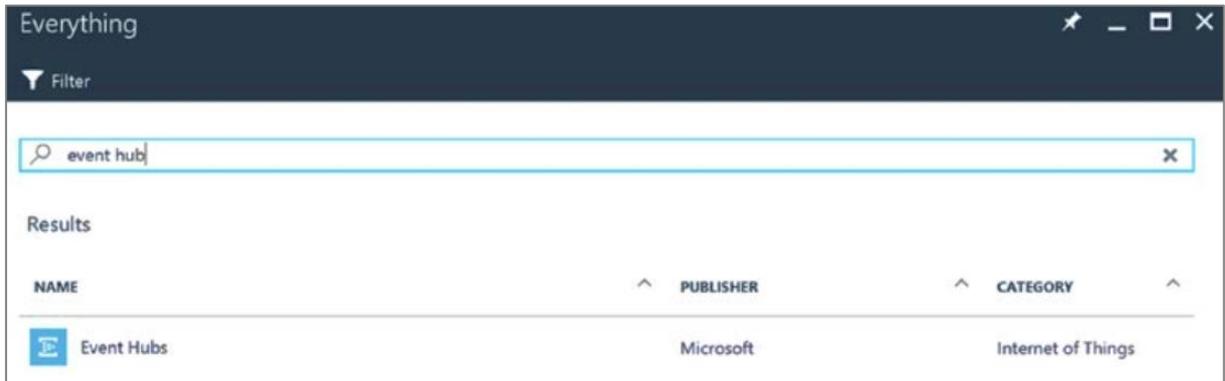
Berikut beberapa hal yang perlu diperhatikan dalam contoh kode:

1. Klausula WITH digunakan untuk membuat subkueri SQL untuk memanggil Layanan Web Machine Learning melalui panggilan fungsi inline ASA dan mengembalikan hasilnya.
 - Subquery digunakan kembali beberapa kali dalam pernyataan SQL yang lebih besar. Ini sebenarnya merupakan praktik terbaik untuk meningkatkan kinerja; hindari menanyakan ulang sumber data yang sama dengan menahan hasil sementara dari klausula WITH.
2. Layanan Web Pembelajaran Mesin dipanggil melalui Fungsi ASA yang disebut Kesehatan Tim Chk.
3. Pernyataan CAST telah ditambahkan untuk mengubah bidang tertentu menjadi keluaran numerik (bukan “string”) untuk meningkatkan visualisasi keluaran Power BI.
 - Contoh: CAST(SQ.result.[BreathingRate] sebagai float) sebagai [BreathingRate]
4. Ada klausula GABUNG di akhir pernyataan untuk input data referensi bernama [tim masukan-refdata].
 - Ini dianggap sebagai Kueri pengayaan sehingga informasi tambahan anggota tim dapat ditambahkan ke kumpulan data.
 - Hal ini memungkinkan Anda menentukan informasi tambahan untuk peringatan, seperti informasi kontak supervisor anggota tim.
5. Jika seorang anggota tim diperkirakan berada pada atau mendekati tingkat kelelahan fisik, Anda perlu segera memberi tahu supervisor.
 - Kami sekarang memiliki informasi supervisor yang tersedia di kumpulan data “jalur panas” melalui Klausula GABUNG tambahan pada kueri dasar.

Tambahkan Alias Output ASA Baru untuk Event Hub untuk Alarm

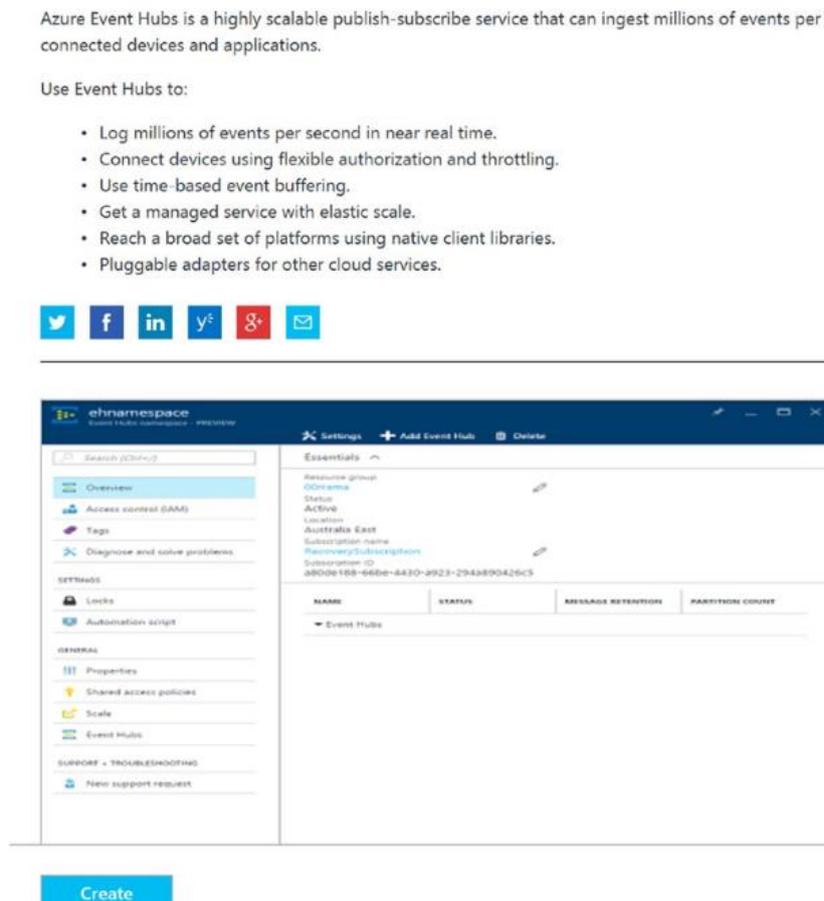
Langkah selanjutnya adalah membuat Azure Event Hub baru melalui portal, sehingga Anda dapat mengirim pesan ke sana melalui kueri Azure Stream Analytics. Pesan yang masuk ke hub peristiwa ini akan digunakan untuk memicu pemberitahuan melalui Fungsi Azure. Fungsi Azure, pada gilirannya, akan mengirimkan pesan teks SMS dan email ke informasi kontak supervisor anggota tim melalui penyedia komunikasi pihak ketiga seperti Twilio dan

SendGrid. Navigasikan ke grup sumber daya Azure untuk solusi Anda dan klik ikon + Tambahkan untuk menambahkan layanan Azure lainnya. Ketik Event Hub di bilah pencarian dan Anda akan melihat hasil serupa seperti Gambar 9.12.



Gambar 9.12. Tambahkan pusat peristiwa baru ke grup sumber daya Azure

Klik pada Event Hub dan Anda akan melihat layar konfirmasi seperti pada Gambar 9.13.



Gambar 9.13. Buat Azure Event Hub baru

Klik ikon Buat di kiri bawah, lalu Anda akan melihat bilah muncul yang meminta Anda memasukkan parameter yang diperlukan untuk membuat Azure Event Hub baru, seperti yang diilustrasikan pada Gambar 9.14.

Gambar 9.14. Buat parameter Azure Event Hub baru

Isi parameter yang diperlukan dan klik ikon Buat. Azure Event Hub baru akan disediakan dalam hitungan menit. Setelah pusat peristiwa baru Anda disediakan di grup sumber daya, Anda dapat menavigasi ke bilah Pusat Peristiwa baru untuk mengambil pengaturan yang diperlukan untuk menentukan alias keluaran di Azure Streaming Analytics. Untuk mengambil pengaturan koneksi:

1. Klik pada Pengaturan/Kebijakan Akses Bersama.
2. Klik RootManagerSharedAccessKey.
3. Ambil pengaturan kunci primer atau sekunder berikut:
 - a. Kunci.(Primer/Sekunder)
 - b. Rangkaian Koneksi

Simpan pengaturan ini untuk langkah berikutnya.

Tambahkan Alias Output ASA Baru untuk Event Hub untuk Peringatan

Sekarang kita telah menentukan Azure Event Hub baru dan mengambil kunci akses, langkah berikutnya adalah menentukan definisi Output Stream Analytics baru. Untuk memulai, navigasikan ke pekerjaan Azure Stream Analytics Anda melalui Portal Azure, lalu klik pada panel Output yang ditunjukkan pada Gambar 9.15.



Gambar 9.15. Definisi Output Analisis Streaming

Selanjutnya, klik ikon + Tambahkan di bilah navigasi atas. Anda akan melihat layar seperti Gambar 9.16.

 The image shows a configuration form for adding an output alias. It includes several fields with asterisks indicating they are required:

- Output alias:** An empty text input field.
- Sink:** A dropdown menu with 'Event hub' selected.
- Subscription:** A dropdown menu with 'Use event hub from current subscription' selected.
- Service bus namespace:** A dropdown menu with 'brtalarm' selected.
- Event hub name:** A dropdown menu with 'brtalarm' selected.
- Event hub policy name:** A dropdown menu with 'RootManageSharedAccessKey' selected.
- Partition key column:** An empty text input field.
- Event serialization format:** A dropdown menu with 'JSON' selected.
- Encoding:** A dropdown menu with 'UTF-8' selected.
- Format:** A dropdown menu with 'Line separated' selected.

 At the bottom of the form is a blue 'Create' button.

Gambar 9.16. Tambahkan parameter Output Alias untuk hub peristiwa

Isi parameter yang diperlukan dan klik ikon Buat. Alias Output baru untuk Azure Event Hub akan dibuat.

Ubah Kueri ASA untuk Menambahkan Output ke Event Hub untuk Peringatan

Sekarang setelah Anda menentukan Azure Event Hub dan alias Output baru, Anda akan menambahkan kueri SQL baru untuk mengeluarkan pemberitahuan ke hub peristiwa ini, dan pada gilirannya, akan memicu Fungsi Azure untuk perutean lebih lanjut.

Navigasikan ke pekerjaan Azure Stream Analytics Anda di Portal Azure dan klik ikon Kueri. Berikut adalah kode untuk menambahkan Azure Event Hub baru sebagai tujuan output tambahan. Tempelkan ke dalam kueri tepat setelah output ke Power BI.

```
-- *****
-- * HOTPATH - OUTPUT to Event Hub
-- * If Machine Learning Predicts Stress
-- *****
SELECT  SQ.UserId,
        SQ.[Timestamp],
        SQ.EventProcessedUtcTime,
        SQ.PartitionId,
        SQ.EventEnqueuedUtcTime,
        CAST(SQ.result.[BreathingRate] as float)
as [BreathingRate],
        CAST(SQ.result.[Temperature] as float)
as [Temperature],
        CAST(SQ.result.[Ventilization] as float)
as [Ventilization],
        CAST(SQ.result.[Activity] as float)
as [Activity],
        CAST(SQ.result.[HeartRateBPM] as bigint)
as [HeartRateBPM],
        CAST(SQ.result.[Cadence] as float)
as [Cadence],
        CAST(SQ.result.[Velocity] as float)
as [Velocity],
        CAST(SQ.result.[Speed] as float)
as [Speed],
        SQ.result.[HIB],
        SQ.result.[HeartrateRedZone],
        SQ.result.[HeartrateVariability],
        SQ.result.[Scored labels],
        SQ.result.[Scored Probabilities],
        RF.id,
        RF.companyname,
        RF.imageUrl,
        RF.firstname,
        RF.lastname,
        RF.username,
        RF.[type],
        RF.[phone],
        RF.[email],
        AD.firstname as [adminfname],
        AD.lastname as [adminlname],
        AD.username as [adminuname],
        AD.type as [admintype],
        AD.phone as [adminphone],
```

```

        AD.email as [adminemail]

INTO [output-EH]
FROM subquery SQ

JOIN [input-refdata-team] RF
    ON SQ.UserId = RF.id

JOIN [input-refdata-team] AD
    ON RF.companyname = AD.companyname

WHERE    SQ.result.[Scored labels] = 'Y'
        AND AD.type = '2'

```

Anda akan melihat bahwa klausa WHERE berisi ungkapan ini:

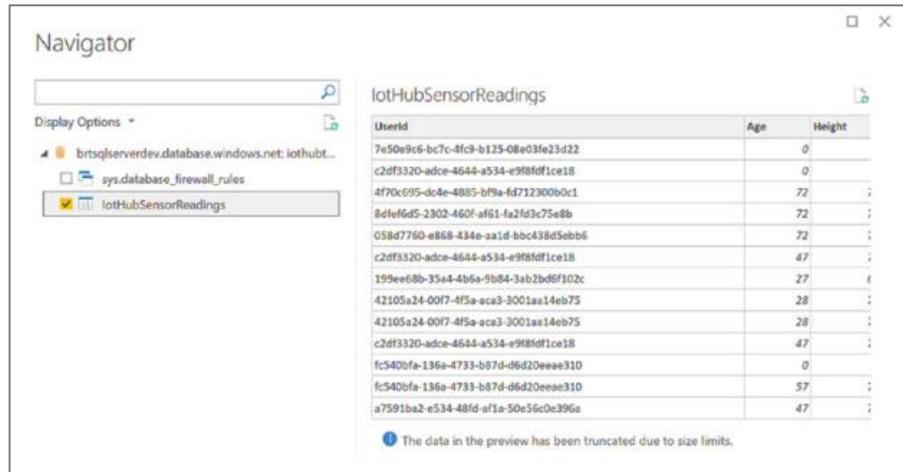
```
SQ.result.[Scored labels] = 'Y'
```

Ini adalah filter kunci untuk kueri SQL ini karena hanya akan mengirim pesan ke Event Hub jika Layanan Web Pembelajaran Mesin mengembalikan “Y” sebagai tingkat kelelahan yang diprediksi. Mudah-mudahan sangat sedikit pesan yang disampaikan ke pusat acara ini.

9.7 VISUALISASI PEMBACAAN SENSOR MENGGUNAKAN R DAN POWER BI DESKTOP

Langkah pertama adalah menggunakan Power BI Desktop untuk menyambungkan ke sumber data jalur hangat dalam skenario referensi, yaitu output yang diarahkan ke Azure SQL Database oleh kueri Stream Analytics. Setelah Anda menginstal Power BI, Anda akan menyambungkan ke database Azure SQL yang merupakan tujuan output yang ditentukan dalam pekerjaan Stream Analytics. Klik Dapatkan Data dan kemudian SQL Server. Setelah itu akan muncul jendela baru dan menanyakan informasi koneksi database SQL, seperti terlihat pada Gambar 9.17.

Gambar 9.17. Parameter informasi koneksi Power BI SQL Server



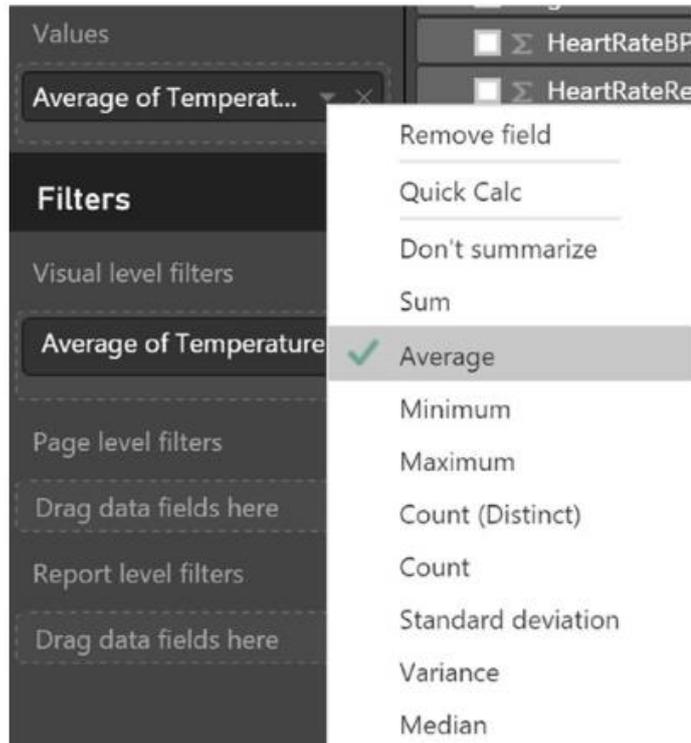
Gambar 9.18. Dapatkan Data dari database Azure SQL: pilih tabel

Masukkan nama server untuk database Azure SQL dalam format ini: <NAMA-SERVER-ANDA>.database.windows.net. Sebaiknya tentukan DirectQuery untuk mendapatkan informasi terbaru dari database, bukan Impor, yang membuat cuplikan data. Klik ikon OK, dan Anda akan melihat layar seperti Gambar 9.18, di mana Anda dapat memilih tabel yang akan digunakan dengan Power BI Desktop.

Pilih tabel [dbo].[lotHubSensorReadings] dan klik ikon Muat untuk menyiapkan data untuk digunakan dengan Power BI Desktop. Pada titik ini, Anda siap untuk mulai membuat kontrol visual dan laporan di Power BI Desktop untuk implementasi referensi. Kami akan membuat tiga laporan dasbor Power BI terpisah, satu untuk masing-masing perusahaan yang diwakili dalam sampel:

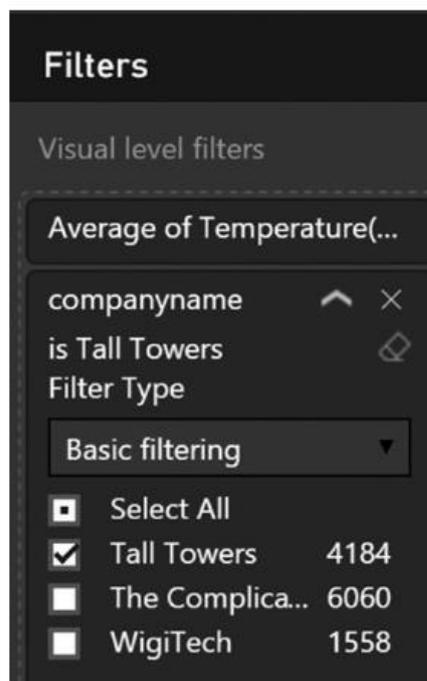
- ❖ Menara Tinggi
- ❖ Badger yang Rumit
- ❖ WigiTech

Kami akan menggunakan kontrol skrip R Power BI bersama dengan paket plot ggplot2 R untuk membuat kontrol poligon frekuensi. Klik pada kontrol R Script di palet Power BI Desktop. Kemudian klik pada bidang Suhu dan seret ke jendela Nilai untuk kontrol. Klik kanan pada nilai suhu dan pilih Rata-rata, seperti yang ditunjukkan pada Gambar 9.19.



Gambar 9.19 Mengubah jenis ringkasan untuk bidang suhu

Saat kontrol Skrip R masih disorot, klik kanan pada bidang nama perusahaan dan pilih Tambahkan Filter. Ini akan menambahkan kontrol filter baru ke laporan, sehingga Anda dapat memfilter masing-masing dari tiga perusahaan dalam penerapan referensi. Setelah filter nama perusahaan ditambahkan, klik pada filter dan pilih Tall Towers, seperti yang ditunjukkan pada Gambar 9.20.



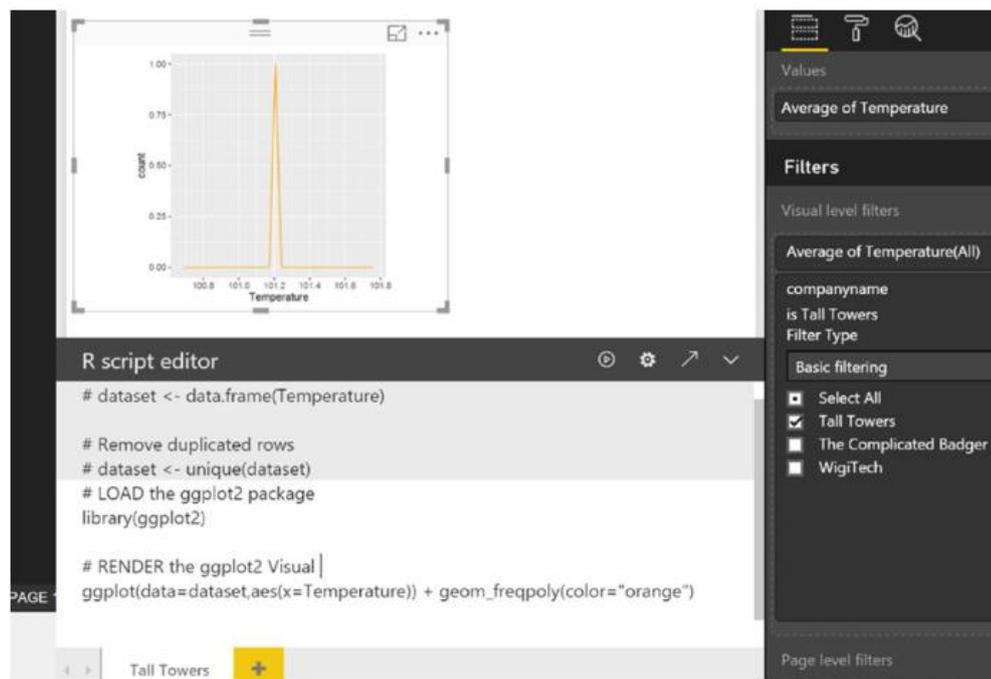
Gambar 9.20. Terapkan kriteria filter ke bidang nama perusahaan

Saat Anda melakukannya, klik kanan pada tab halaman di bagian bawah desainer Power BI dan pilih Ganti Nama Halaman. Atur nama halaman Laporan Power BI ke Menara Tinggi. Sekarang, Anda siap menambahkan kode R ke kontrol. Klik pada kontrol R Script dan tempelkan kode berikut ke jendela editor R:

```
# LOAD the ggplot2 package
library(ggplot2)

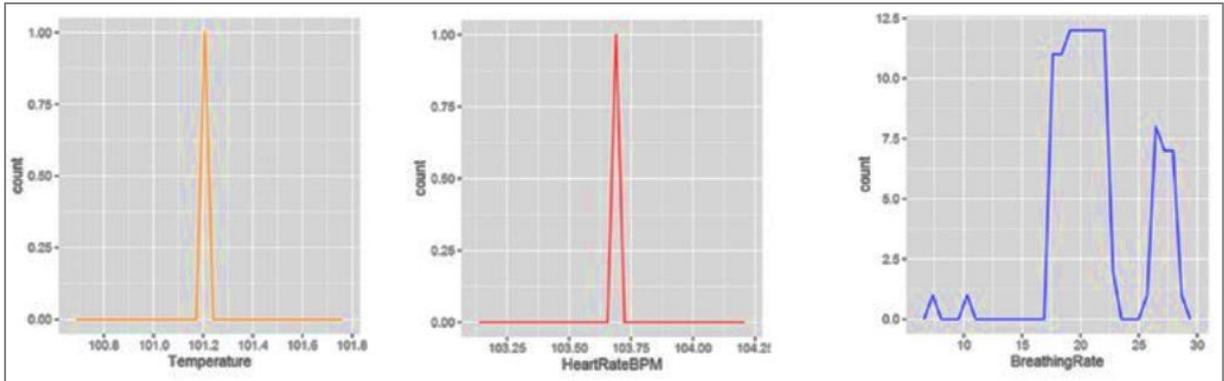
#RENDER the ggplot2 Visual
ggplot(data=dataset,aes(x=Temperature))
  + geom_freqpoly(color="orange")
```

Setelah Anda menempelkan kode R ke kontrol, klik panah ► untuk mengkompilasi dan menjalankan Skrip R. Kontrol Anda akan ditampilkan seperti Gambar 9-21.



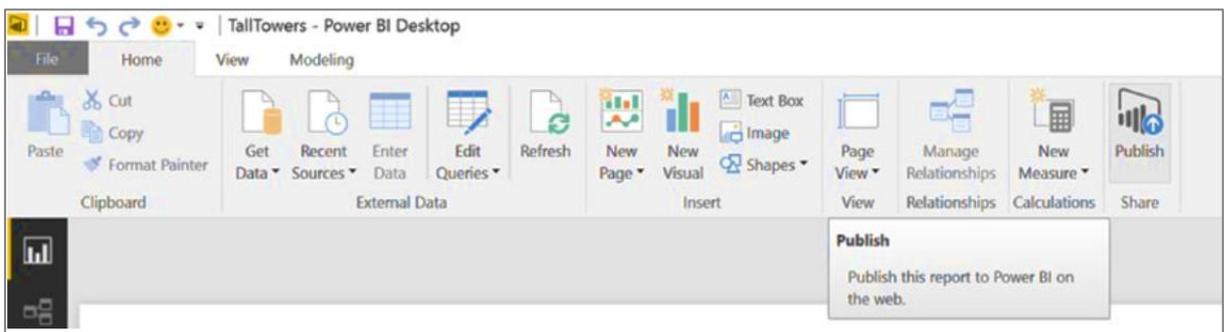
Gambar 9.21. Kontrol Skrip Power BI R beraksi, menampilkan suhu rata-rata untuk Menara Tinggi

Sekarang setelah Anda mengonfigurasi kontrol Skrip Power BI R untuk menampilkan pembacaan sensor suhu rata-rata untuk Menara Tinggi, Anda dapat dengan mudah mengulangi proses untuk membuat dua kontrol Skrip R lagi untuk HeartRateBPM dan BreathingRate. Kami akan menggunakan kode warna untuk membedakan nilainya. Gambar 9.22 menggambarkan kontrol Skrip R yang telah selesai di Power BI Desktop.



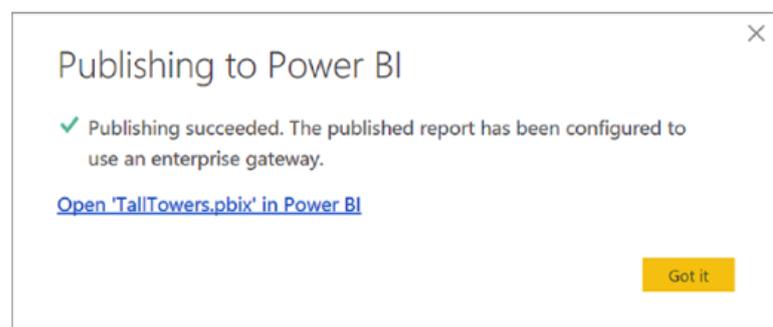
Gambar 9.22. Skrip R mengontrol plot frekuensi rendering untuk berbagai pembacaan sensor

Sebelum Anda menerbitkan laporan ke Layanan Power BI, simpan file Power BI Desktop sebagai Menara Tinggi.pbix. Alasan pemberian nama file deskriptif adalah karena langkah selanjutnya adalah mempublikasikan laporan ke layanan Power BI dan ini akan menjadi nama yang diberikan untuk laporan tersebut. Klik ikon Publikasikan di bilah navigasi atas, seperti yang ditunjukkan pada Gambar 9.23.



Gambar 9.23 Publikasikan Laporan Power BI dengan kontrol R kustom ke Layanan Power BI

Setelah beberapa saat, Anda akan melihat jendela penyelesaian muncul, seperti yang ditunjukkan pada Gambar 9.24.

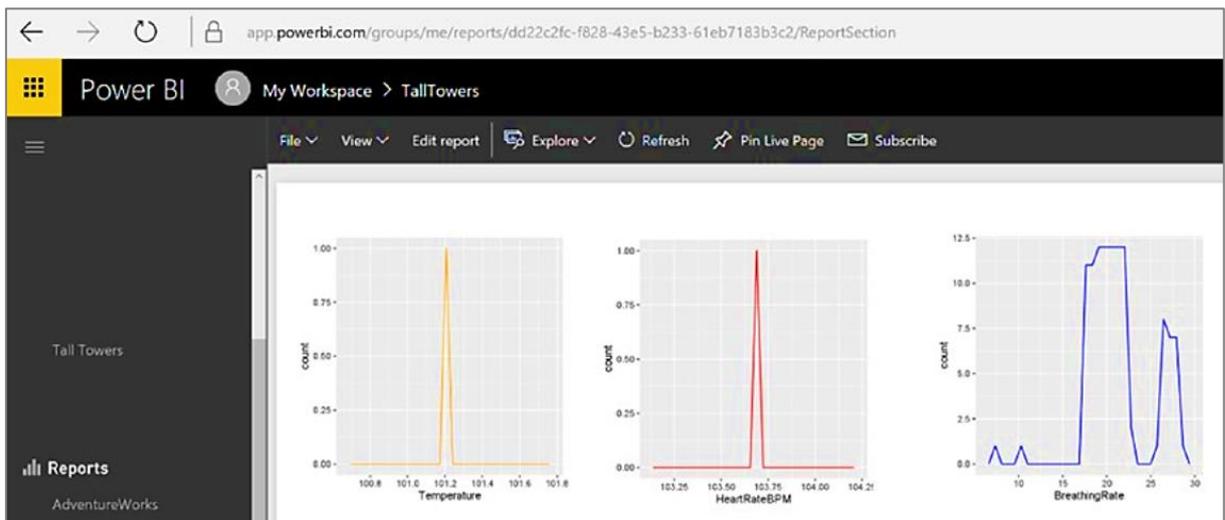


Gambar 9.24. Pesan penyelesaian untuk mempublikasikan laporan Power BI ke layanan Power BI

Pada titik ini, Anda dapat mengklik tautan untuk Membuka 'Tall Towers.pbix' di Power BI dan sesi browser di layanan Power BI yang terletak di <http://app.powerbi.com> akan terbuka.

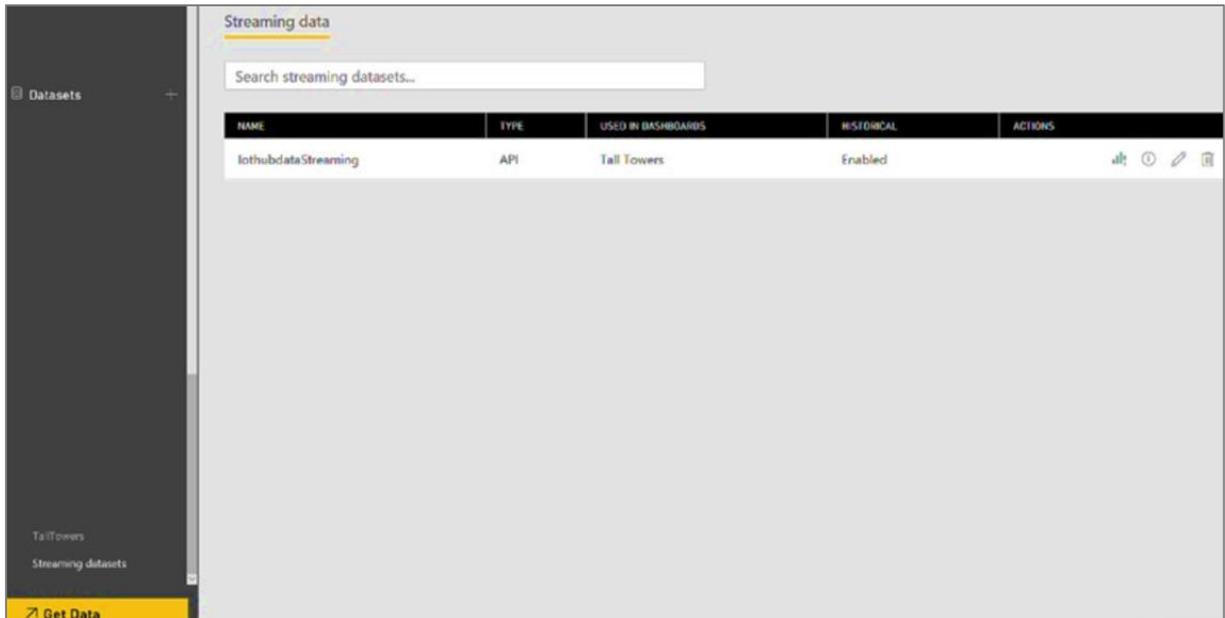
Membangun Visualisasi Menggunakan Layanan Power Bi Untuk Dashboard Pembacaan Sensor Real-Time

Setelah mengklik tautan untuk Membuka 'Tall Towers.pbix' di Power BI pada langkah terakhir, Anda kemudian akan dimintai kredensial Anda untuk mendapatkan akses ke layanan aplikasi Power BI.



Gambar 9.25. Laporan dengan kontrol Skrip R yang dipublikasikan ke layanan Power BI (app.powerbi.com) dari Power BI Desktop

Langkah selanjutnya adalah menambahkan kontrol visualisasi untuk menyempurnakan kontrol R Script yang baru saja Anda unggah. Perhatikan bahwa saat ini hanya mungkin untuk membuat kontrol Skrip R di Power BI Desktop. Kami ingin menggunakan sumber data berbeda untuk visualisasi kami, yaitu "*Kumpulan Data Streaming*" yang diisi oleh pekerjaan Azure Stream Analytics yang sebelumnya Anda modifikasi di bab ini. Klik pada bilah navigasi kiri bawah saat berada di layanan aplikasi Power BI dan klik opsi Kumpulan Data Streaming, seperti yang ditunjukkan pada Gambar 9.26.



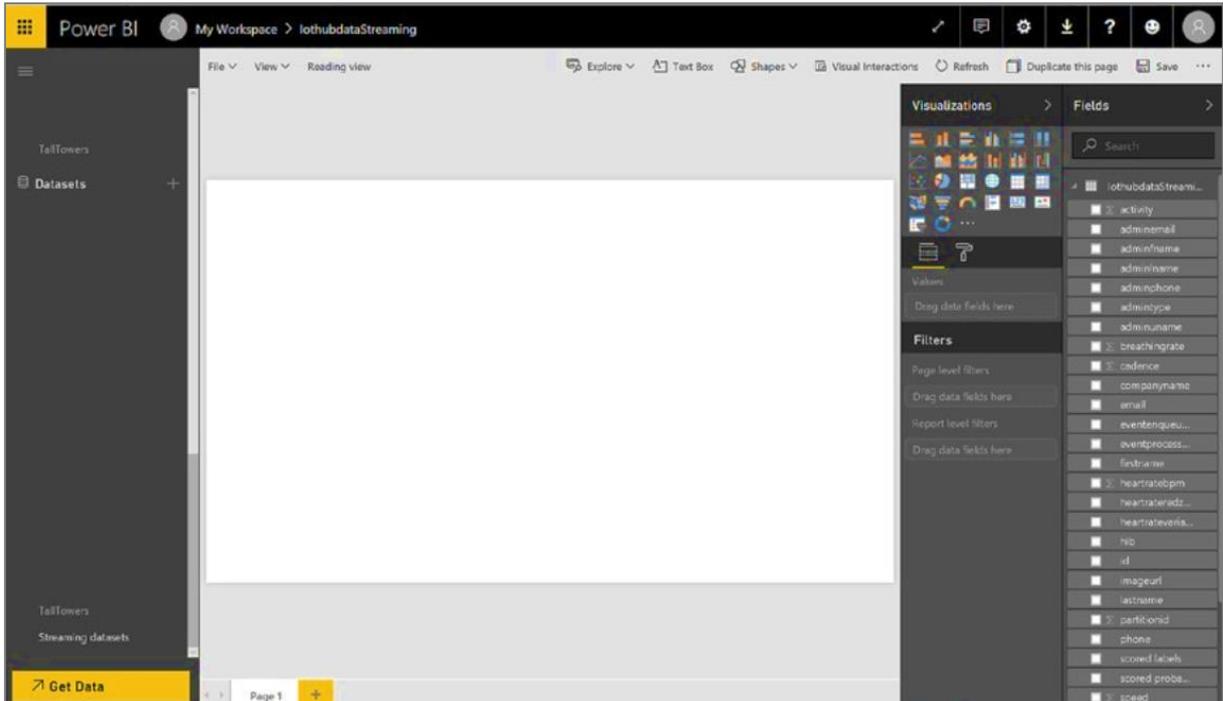
Gambar 9.26. Bilah navigasi kiri bawah untuk mengakses kumpulan data streaming dari layanan Power BI

Akan muncul daftar berisi dataset streaming yang telah dibuat. Jika Anda belum pernah memulai pekerjaan Azure Stream Analytics atau memproses data apa pun (misalnya menggunakan simulator), daftar ini mungkin kosong. Jika demikian, silakan langsung beralih ke bagian menjalankan simulator, lalu kembali ke bagian ini untuk melihat kumpulan data yang diisi di sini. Klik pada ikon Buat Laporan untuk kumpulan data streaming bernama `iothubdataStreaming`, seperti yang ditunjukkan pada Gambar 9.27.



Gambar 9.27. Kumpulan data streaming Layanan Power BI, ikon Buat Laporan

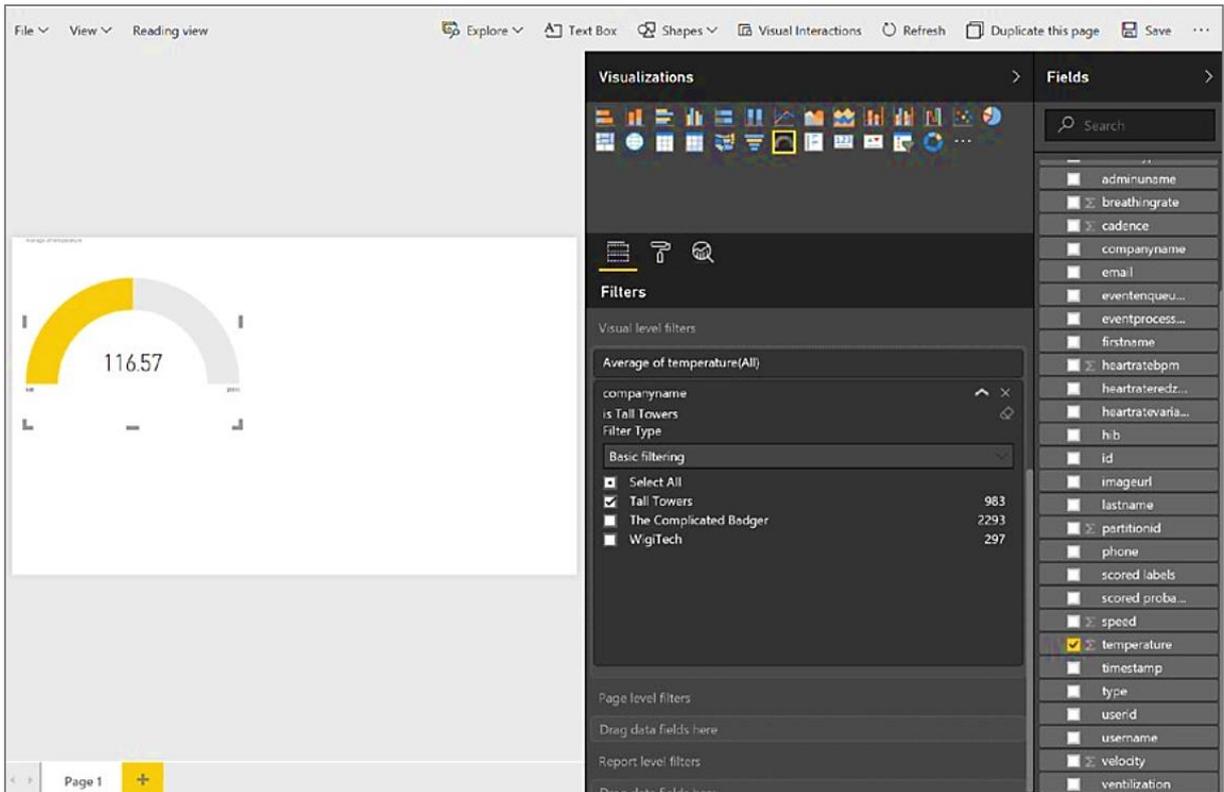
Selanjutnya, perancang Power BI memuat dan mengisi dengan himpunan data yang dihasilkan dari pekerjaan Stream Analytics secara real-time.



Gambar 9.28. Perancang Layanan Aplikasi Power BI dengan himpunan data iothubdataStreaming

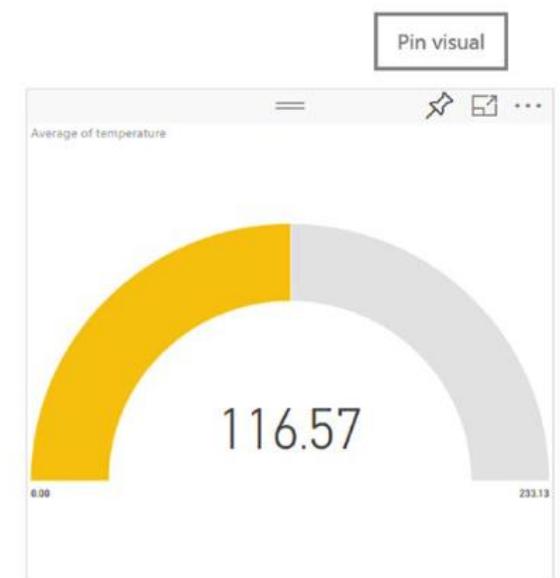
Kontrol Power BI pertama yang akan kami buat adalah pengukur sensor waktu nyata yang menunjukkan suhu rata-rata anggota tim dari organisasi Menara Tinggi. Klik pada kontrol pengukur di palet kontrol Power BI. Ini mengisi permukaan desainer dengan kontrol baru. Selanjutnya, klik pada bidang suhu, dan secara otomatis akan ditampilkan di pengukur. Namun perlu diperhatikan, bahwa secara default, ini menunjukkan jumlah bidang suhu. Untuk mengubahnya, klik kanan pada bidang suhu di area Nilai lalu pilih Rata-rata. Kontrol pengukur kemudian akan menampilkan rata-rata pembacaan sensor suhu yang diproses sejauh ini.

Hal berikutnya yang perlu Anda lakukan adalah menambahkan filter untuk perusahaan yang tepat. Klik kanan pada bidang nama perusahaan di daftar bidang kumpulan data dan pilih opsi Tambahkan Filter. Ini akan menambahkan kontrol filter di bawah properti untuk kontrol pengukur. Sesuaikan properti filter agar hanya menampilkan data Menara Tinggi. Selanjutnya, atur warna kontrol pengukur ini untuk sensor suhu agar sesuai dengan kontrol R Script sebelumnya, yang ditampilkan dalam warna oranye. Untuk melakukan ini, cukup klik pada ikon Paint Roller Brush di bawah properti untuk kontrol pengukur. Perluas properti Warna Data lalu pilih warna oranye untuk nilai isian. Pada titik ini, kontrol Power BI Gauge Anda akan terlihat serupa dengan yang ada di Gambar 9.29.



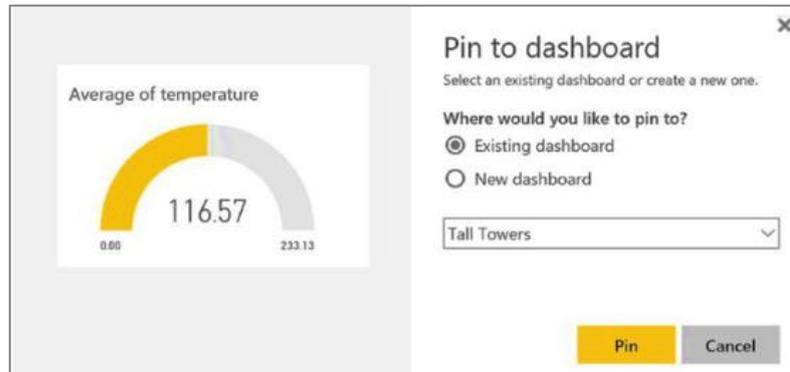
Gambar 9.29. Kontrol Power BI Gauge untuk pembacaan suhu rata-rata secara real-time

Selanjutnya, ganti nama halaman saat ini dengan mengklik kanan pada tab halaman, pilih opsi Ganti Nama Halaman, dan ubah menjadi Tall Towers Real-Time. Klik ikon Simpan di bilah navigasi kanan atas dan Anda akan diminta memasukkan nama untuk laporan Anda agar dapat disimpan. Masuki Menara Tinggi Secara Real-Time. Sekarang Anda akan menambahkan kontrol pengukur ini ke dasbor Menara Tinggi dengan mengklik ikon Pin Visual di sudut kanan atas kontrol pengukur, seperti yang diilustrasikan pada Gambar 9.30.



Gambar 9.30. Layanan aplikasi Power BI Pin ikon Visual

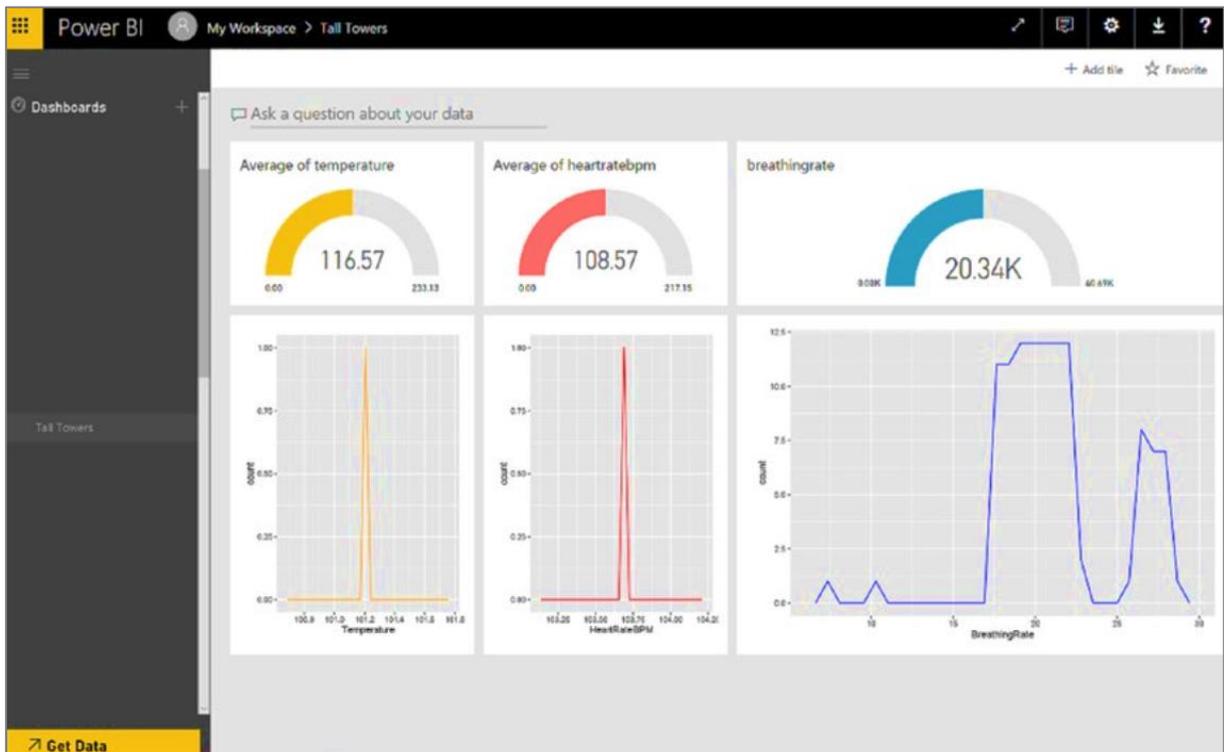
Setelah mengklik ikon Pin Visual, Anda kemudian akan disajikan opsi untuk memilih dasbor yang akan dipin kontrolnya, seperti yang ditunjukkan pada Gambar 9.31.



Gambar 9.31 Pilih dasbor Power BI untuk menyematkan kontrol

Dasbor Tall Towers seharusnya sudah ada sejak Anda menerbitkan dasbor dari Power BI Desktop. Pilih opsi ini (jika tersedia) atau Anda dapat membuat dasbor baru.

Ini akan menambahkan kontrol pengukur ke dasbor Menara Tinggi yang ada dengan tiga kontrol Skrip R untuk pembacaan sensor. Ulangi langkah sebelumnya untuk membuat dua kontrol pengukur tambahan untuk HeartRateBPM dan BreathingRate, terapkan tema warna, dan sematkan visual pengukur ke dasbor Menara Tinggi. Pada titik ini, dashboard Tall Towers akan terlihat seperti Gambar 9-32.



Gambar 9.32. Dasbor Power BI untuk Menara Tinggi

Perhatikan bahwa kontrol pengukur di baris atas dasbor Tall Towers mewakili data real-time, jalur data panas yang dialirkan langsung ke layanan Power BI dari layanan Stream Analytics. Kontrol Skrip R yang terletak di bawah kontrol pengukur pada Gambar 9.32 didasarkan pada data dari database Azure SQL yang juga diisi dari layanan Stream Analytics. Data ini dianggap hangat karena peningkatan latensi terkait dengan data yang diambil dari database relasional. Dengan cara ini, Anda telah membuat dasbor untuk Tall Towers yang menampilkan metrik KPI real-time dan batch.

9.8 MEMBANGUN VISUALISASI POWER BI UNTUK PREDIKSI PEMBELAJARAN MESIN

Kita perlu mengonfigurasi satu set visualisasi Power BI lagi untuk menyampaikan prediksi Layanan Web Pembelajaran Mesin untuk anggota tim yang kelelahan. Untuk memulai, navigasikan kembali ke laporan Tall Towers Real-Time dan klik ikon Edit di bilah navigasi atas. Anda akan menambahkan dua kontrol lagi ke dasbor:

- Kontrol diagram lingkaran untuk menampilkan visualisasi jumlah prediksi Machine Learning (Y atau N) yang mencerminkan kelelahan anggota tim.
- Kontrol tabel untuk menampilkan informasi masing-masing anggota tim saat Anda mengklik bagian diagram lingkaran untuk menelusuri data.

Kontrol Diagram Lingkaran

Klik pada kontrol Pie Chart dan kemudian pilih bidang Scored Labels. Selanjutnya seret bidang Label Skor dan letakkan di properti Nilai untuk kontrol Diagram Lingkaran. Selanjutnya, tambahkan filter hanya untuk data Menara Tinggi dengan mengklik kanan pada kolom nama perusahaan dan memilih opsi Tambahkan Filter. Perluas pengaturan filter di bawah properti Diagram Lingkaran dan filter hanya untuk data Menara Tinggi.

Tambahkan filter untuk Perusahaan dengan mengklik kanan kolom nama perusahaan di daftar kolom kumpulan data, lalu pilih opsi Tambahkan Filter. Atur properti Filter ke Menara Tinggi. Yang terakhir adalah mengubah warna pada Pie Chart. Untuk melakukan ini, cukup klik ikon Paint Roller Brush di bawah properti kontrol diagram lingkaran. Perluas properti Warna Data lalu pilih Hijau untuk nilai isian N data. Pilih Merah untuk nilai pengisian data Y.

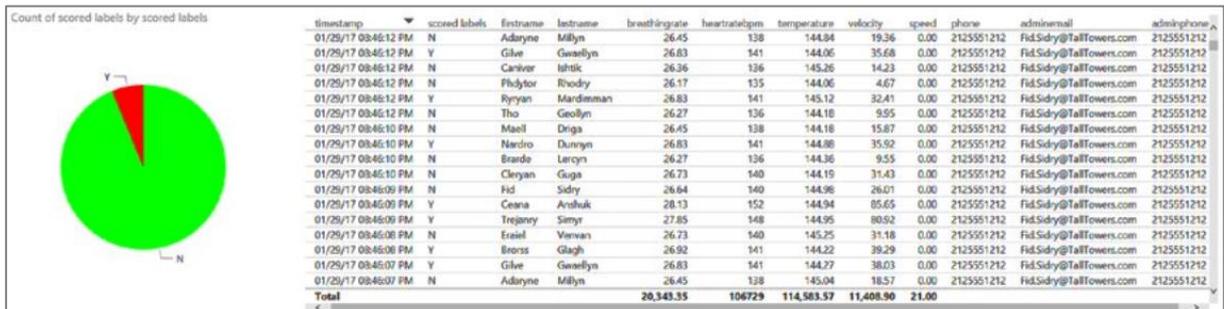
Kontrol Tabel

Mulailah dengan mengklik kontrol Tabel lalu pilih bidang berikut untuk ditampilkan di kontrol:

- ✓ stempel waktu
- ✓ label yang diberi skor
- ✓ nama depan
- ✓ nama keluarga
- ✓ laju pernapasan
- ✓ detak jantungBPM
- ✓ suhu
- ✓ kecepatan
- ✓ kecepatan
- ✓ telepon
- ✓ email admin

✓ telepon admin

Selanjutnya, tambahkan filter hanya untuk data Menara Tinggi dengan mengklik kanan kolom nama perusahaan di daftar kolom kumpulan data lalu memilih opsi Tambahkan Filter. Atur properti Filter ke Menara Tinggi. Pada titik ini, kontrol Diagram Lingkaran dan Tabel yang baru akan menyerupai Gambar 9.33.

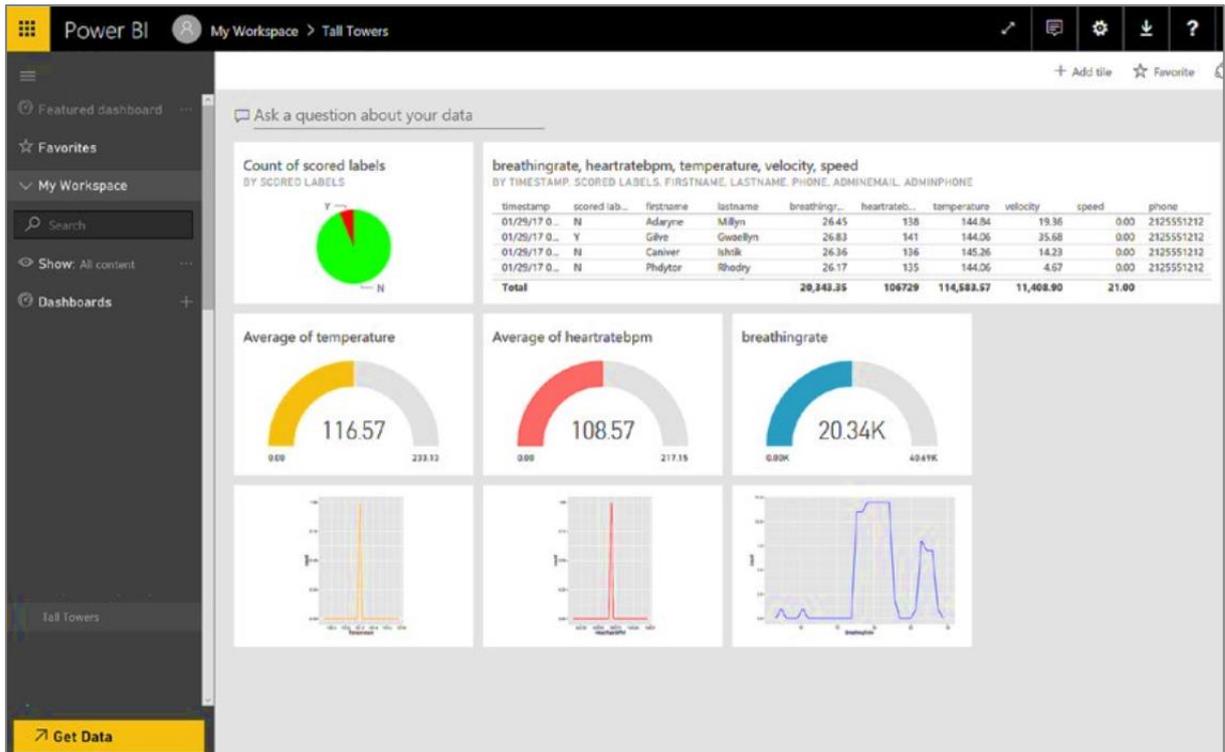


Gambar 9.33. Diagram lingkaran dan tampilan tabel Power BI untuk prediksi Machine Learning

Beberapa hal yang perlu diperhatikan tentang bagaimana kontrol ini berinteraksi secara otomatis:

- ❖ Kontrol Diagram Lingkaran dan kontrol Tabel bekerja sama melalui properti filter.
- ❖ Jika Anda mengklik bagian hijau pada diagram lingkaran, data di kontrol Tabel secara otomatis difilter menjadi hanya rekaman dengan prediksi N di bidang Label Skor.
- ❖ Jika Anda mengklik bagian merah pada diagram lingkaran, data di kontrol Tabel secara otomatis difilter menjadi hanya rekaman dengan prediksi Y di bidang Label Skor.
- ❖ Jika Anda mengklik di mana saja di luar kontrol Pie Chart, Anda akan melihat semua data ditampilkan di kontrol Tabel.

Selanjutnya, simpan laporan Power BI dan sematkan diagram lingkaran dan kontrol Tabel ke dasbor Tall Towers. Anda kemudian dapat mengatur ulang dan mengubah ukuran kontrol di dasbor Power BI untuk Menara Tinggi agar tampilannya menarik secara visual. Gambar 9.34 menggambarkan dasbor terbaru untuk Menara Tinggi.



Gambar 9.34. Tampilan dasbor Power BI yang lengkap untuk Menara Tinggi

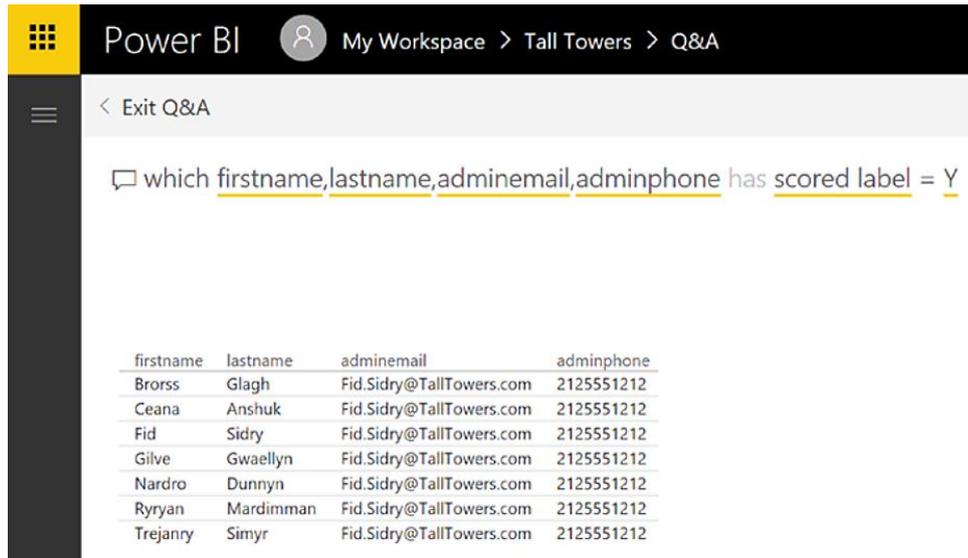
Tanya Jawab Power BI

Ingatlah bahwa Power BI memiliki fitur canggih yang disebut Tanya Jawab yang dapat Anda gunakan untuk “mengajukan” pertanyaan tentang data Anda. Untuk mulai menggunakan fitur kueri bahasa alami Power BI, cukup ketik di jendela Tanya Jawab di bagian atas dasbor Power BI. Anda dapat melihat jendela Tanya Jawab di bagian atas dasbor yang digambarkan pada Gambar 9.34. Tanya Jawab Power BI mengenali kata-kata saat Anda mengetiknya dan mengetahui nama bidang serta kumpulan data mana yang dapat menemukan jawabannya. T&J membantu Anda membentuk pertanyaan bahasa alami dengan pelengkapan otomatis, pernyataan ulang, dan bantuan tekstual lainnya. Untuk mendemonstrasikan cara kerja Tanya Jawab, misalkan Anda ingin melihat nama anggota tim yang diperkirakan berisiko mengalami kelelahan fisik.

Anda cukup mengetikkan pernyataan berikut di jendela Tanya Jawab Power BI di bagian atas dasbor:

`which firstname, lastname, adminemail, adminphone has scored label = Y`

Saat Anda berhenti mengetik, Tanya Jawab Power BI segera mengembalikan jawaban yang ditunjukkan pada Gambar 9.35.



Power BI My Workspace > Tall Towers > Q&A

< Exit Q&A

which firstname,lastname,adminemail,adminphone has scored label = Y

firstname	lastname	adminemail	adminphone
Brors	Glagh	Fid.Sidry@TallTowers.com	2125551212
Ceana	Anshuk	Fid.Sidry@TallTowers.com	2125551212
Fid	Sidry	Fid.Sidry@TallTowers.com	2125551212
Gilve	Gwaellyn	Fid.Sidry@TallTowers.com	2125551212
Nardro	Dunnyn	Fid.Sidry@TallTowers.com	2125551212
Ryryan	Mardimman	Fid.Sidry@TallTowers.com	2125551212
Trejanry	Simyr	Fid.Sidry@TallTowers.com	2125551212

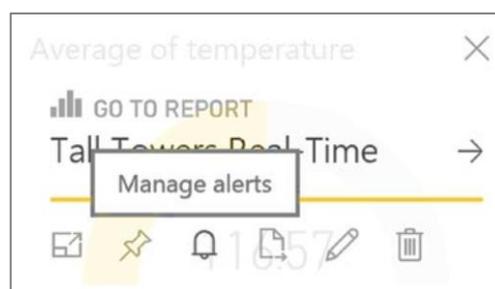
Gambar 9.35. Hasil kueri bahasa alami Tanya Jawab Power BI

Perhatikan bahwa selain memiliki informasi penting tentang anggota tim, Anda juga memiliki email administrator dan nomor telepon untuk setiap anggota tim dengan prediksi positif. Anda akan menggunakan informasi ini di bagian berikutnya untuk memicu peringatan otomatis ke admin menggunakan Azure Functions.

Peringatan Power BI

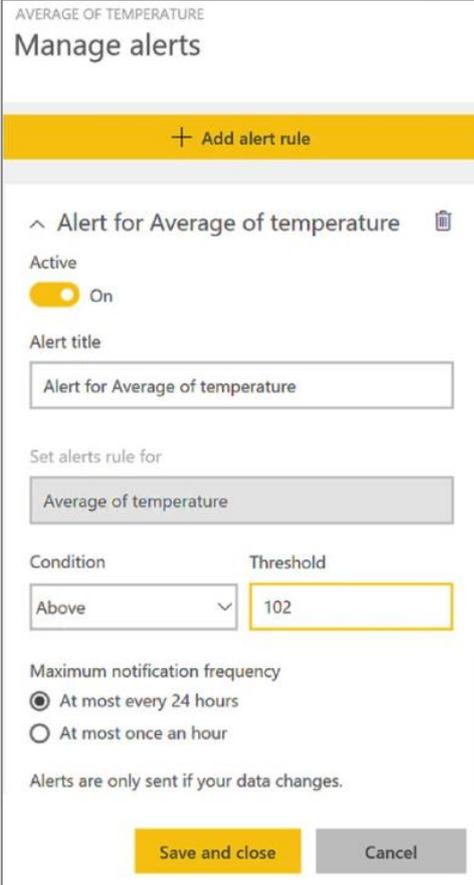
Power BI juga memiliki kemampuan untuk mengonfigurasi peringatan untuk menerima pemberitahuan ketika KPI bisnis penting dan metrik utama melebihi ambang batas yang Anda tetapkan. Hanya Anda yang dapat melihat peringatan data yang Anda atur, meskipun Anda berbagi dasbor dengan orang lain. Peringatan hanya berfungsi untuk kontrol Power BI yang menampilkan satu nomor, seperti kontrol “kartu” atau “ukuran”. Ini akan bekerja dengan sempurna untuk implementasi referensi ini, karena Anda mengonfigurasi tiga kontrol pengukur individual di dasbor Anda untuk memantau suhu rata-rata, detak jantung, dan laju pernapasan anggota tim.

Untuk memulai, klik pada tiga titik (atau elips) di sudut kanan atas kontrol pengukur yang Anda buat untuk suhu. Anda kemudian akan melihat ikon Peringatan, seperti yang ditunjukkan pada Gambar 9.36.



Gambar 9.36. Konfigurasi peringatan Power BI

Setelah Anda mengklik ikon Peringatan, Anda akan melihat jendela baru tempat Anda dapat menambahkan aturan peringatan baru. Gambar 9-37 menggambarkan properti yang dapat Anda atur untuk aturan peringatan baru.



AVERAGE OF TEMPERATURE
Manage alerts

+ Add alert rule

^ Alert for Average of temperature 

Active
 On

Alert title
Alert for Average of temperature

Set alerts rule for
Average of temperature

Condition	Threshold
Above 	102

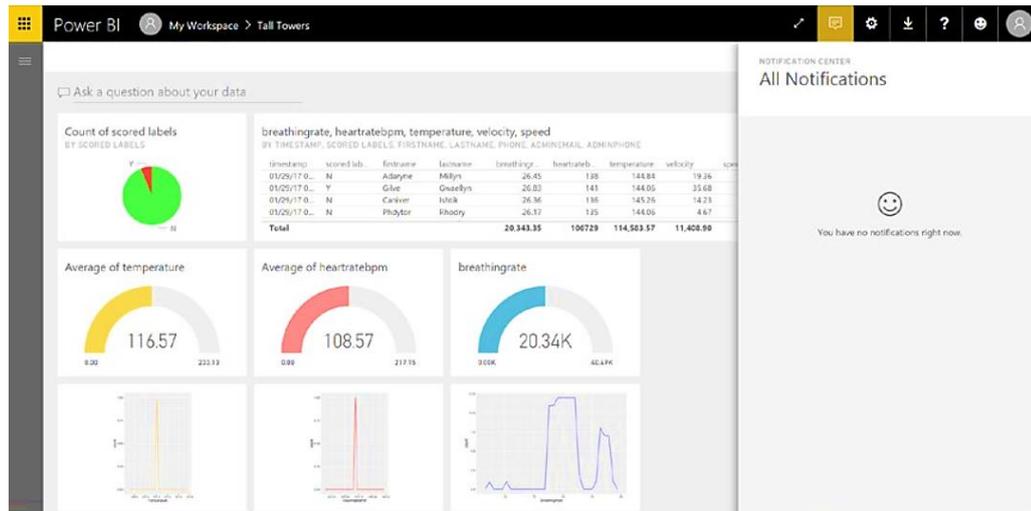
Maximum notification frequency
 At most every 24 hours
 At most once an hour

Alerts are only sent if your data changes.

Save and close Cancel

Gambar 9.37 Peringatan Power BI: tambahkan properti aturan pemberitahuan baru

Setelah Anda mengatur peringatan untuk kontrol dasbor, Anda akan menerima notifikasi melalui pusat notifikasi Power BI, seperti yang ditunjukkan di sebelah kanan pada Gambar 9-38.



Gambar 9.38 Jendela pusat notifikasi Power BI

Pada titik ini, Anda telah menyelesaikan dasbor Power BI untuk memvisualisasikan sensor dan data Pembelajaran Mesin. Di dasbor ini, Anda menggabungkan himpunan data untuk sumber data real-time (dari Stream Analytics) dan batch (dari Azure SQL Database).

9.9 FUNGSI AZURE UNTUK PERINGATAN

Di bagian ini, kami membahas implementasi Azure Functions untuk mengirim peringatan dan pemberitahuan untuk implementasi referensi. Ingatlah bahwa sebelumnya di bab ini, Anda menambahkan Azure Event Hub baru untuk menyimpan pemberitahuan peringatan yang dihasilkan oleh prediksi dari Layanan Web Machine Learning Anda. Anda kemudian memodifikasi kueri Stream Analytics untuk mengeluarkan pesan pemberitahuan ke pusat peristiwa jika prediksinya adalah Y.

Skenario Implementasi Referensi

Dalam implementasi referensi, Anda akan mengirimkan peringatan ke administrator yang bertanggung jawab atas setiap anggota tim untuk perusahaan tertentu. Saat Anda memodifikasi kueri Streaming Analytics, Anda memasukkan kode mirip SQL yang melakukan operasi JOIN untuk mendapatkan informasi administrator anggota tim:

```
JOIN [input-refdata-team] AD
ON RF.companyname = AD.companyname
WHERE SQ.result.[Scored labels] = 'Y'
AND AD.type = '2'
```

Sebagai hasil dari klausa JOIN Anda memiliki akses ke bidang berikut dalam kueri:

```
AD.firstname as [adminfname],
AD.lastname as [adminlname],
AD.username as [adminuname],
AD.type as [admintype],
AD.phone as [adminphone],
```

AD.email as [adminemail]

Gambar 9.39 adalah tampilan terbaru dari Gambar 9.35 yang menyoroti informasi administrator anggota tim yang dikembalikan sebagai hasil dari kueri Tanya Jawab.

The screenshot shows the Power BI Q&A interface. At the top, it says 'Power BI' and 'My Workspace > Tall Towers > Q&A'. Below that, there is a navigation bar with '< Exit Q&A'. The main area contains a query: 'which firstname,lastname,adminemail,adminphone has scored label = Y'. Below the query is a table with the following data:

firstname	lastname	adminemail	adminphone
Brorss	Glagh	Fid.Sidry@TallTowers.com	2125551212
Ceana	Anshuk	Fid.Sidry@TallTowers.com	2125551212
Fid	Sidry	Fid.Sidry@TallTowers.com	2125551212
Gilve	Gwaellyn	Fid.Sidry@TallTowers.com	2125551212
Nardro	Dunnyn	Fid.Sidry@TallTowers.com	2125551212
Ryryan	Mardimman	Fid.Sidry@TallTowers.com	2125551212
Trejanry	Simyr	Fid.Sidry@TallTowers.com	2125551212

Gambar 9.39. Tanya Jawab Power BI menampilkan informasi administrator dalam himpunan data yang dikembalikan

Informasi ini adalah dasar untuk mengirimkan pesan peringatan, dan Anda akan mengirimkan pemberitahuan teks SMS ke administrator untuk tindakan segera. Sebagai cadangan, Anda juga akan mengirimkan email prioritas tinggi. Masing-masing skenario pemberitahuan ini akan diimplementasikan menggunakan Azure Functions.

Buat Fungsi Azure untuk Mengirim Peringatan Teks melalui Twilio

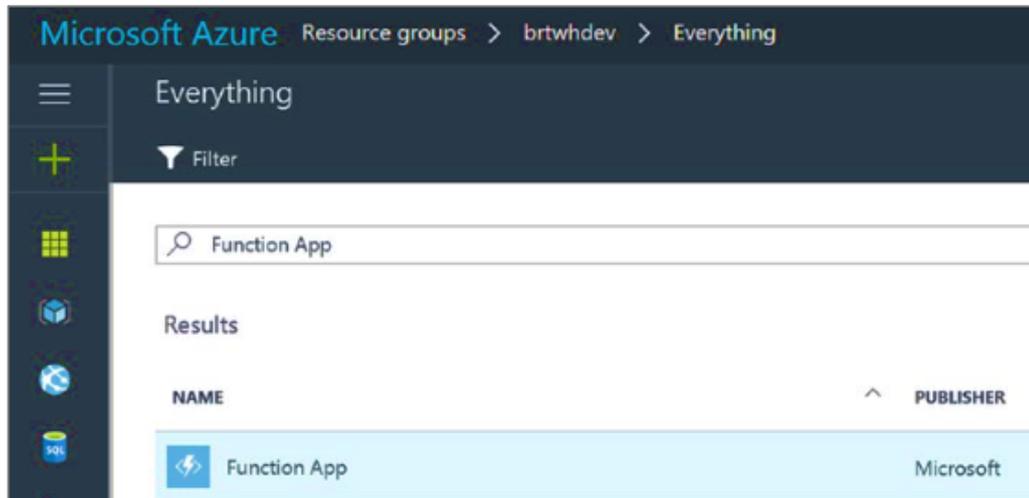
Fungsi Azure pertama yang Anda buat adalah untuk mengirim pesan teks SMS. Salah satu pengikatan keluaran yang tersedia di Azure Functions adalah melalui Twilio, platform komunikasi cloud pihak ketiga untuk membangun aplikasi suara dan pesan melalui ReST API. Twilio memiliki API yang kuat untuk mengirim pesan teks SMS.

Perhatikan bahwa ada dua set kredensial yang diberikan, satu untuk kredensial aktif dan satu lagi untuk pengujian. Anda perlu mencatat informasi Anda untuk kredensial pengujian berikut:

- SID Akun
- Token Akun

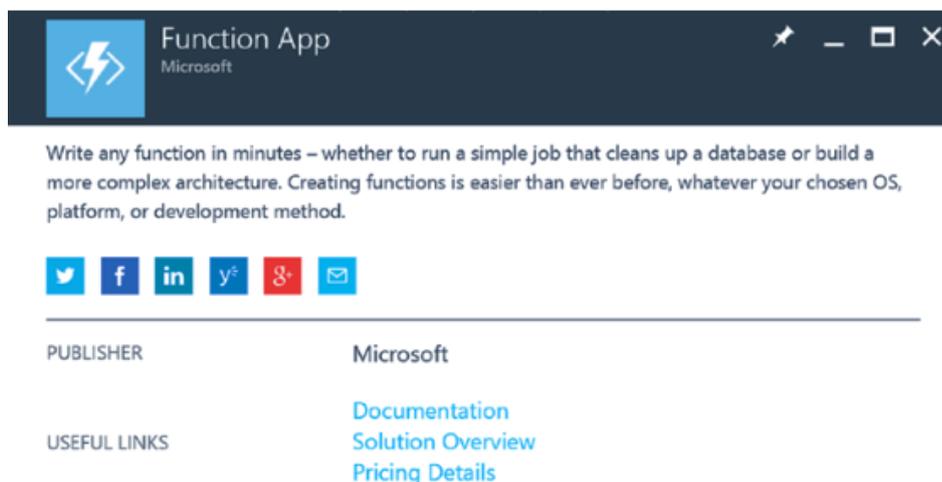
Selain itu, pastikan untuk membaca dokumentasi Twilio untuk menggunakan kredensial pengujian di <https://www.twilio.com/docs/api/rest/test-credentials>. Satu hal penting yang dapat diambil dari tautan sebelumnya adalah adanya nomor telepon “ajaib” khusus yang dapat Anda gunakan saat menguji dengan kredensial tes Twilio. Penting juga untuk dicatat

bahwa pesan teks SMS asli tidak akan dikirim ke nomor telepon asli mana pun saat menggunakan kredensial tes Twilio. Untuk memulai, navigasikan ke grup sumber daya Azure untuk solusi Anda dan klik ikon + Tambahkan untuk menambahkan layanan Azure lainnya. Ketik Aplikasi Fungsi di bilah pencarian. Anda akan melihat hasil yang serupa dengan tangkapan layar pada Gambar 9.40.



Gambar 9.40. Menambahkan aplikasi Azure Function baru ke grup sumber daya

Klik pada Function App dan Anda akan melihat layar konfirmasi seperti pada Gambar 9.41.



Gambar 9.41 Halaman konfirmasi aplikasi fungsi

Klik pada ikon Buat di kiri bawah dan Anda akan melihat bilah muncul. Bilah ini akan meminta Anda memasukkan properti yang diperlukan untuk membuat Fungsi Azure Anda, seperti yang diilustrasikan pada Gambar 9.42.

Function App

* App name
AlarmFunctions ✓
.azurewebsites.net

* Subscription
azurepass-bobf

* Resource Group
 Create new
 Use existing

* Hosting Plan
App Service Plan

* App Service plan/Location >

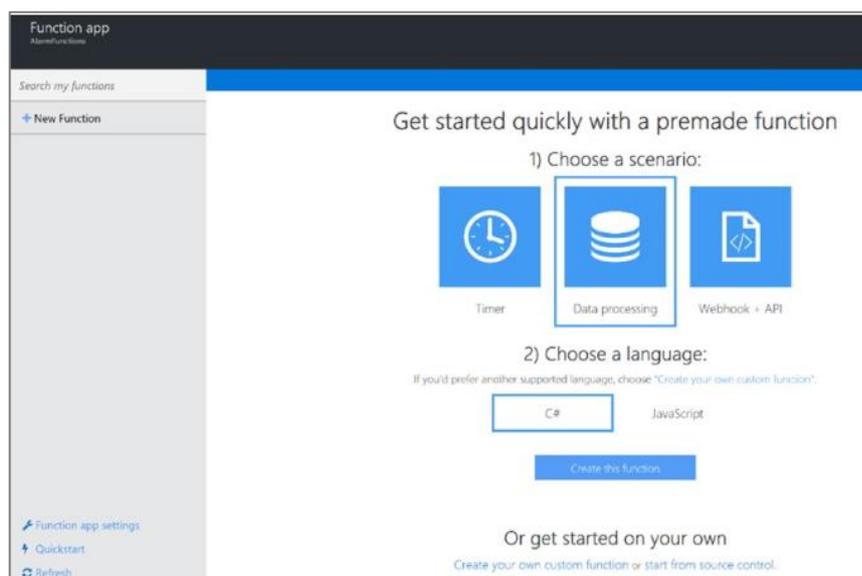
* Storage Account >

Pin to dashboard

Create Automation options

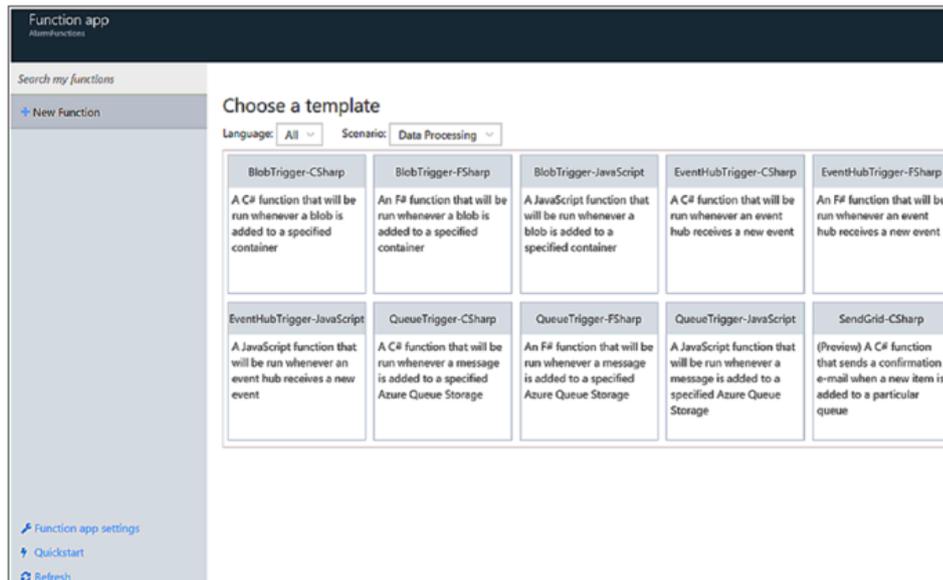
Gambar 9.42. Buat parameter Fungsi Azure baru

Setelah Azure Function Anda disediakan di grup sumber daya, Anda dapat menavigasi ke bilah Azure Function baru, seperti yang digambarkan pada Gambar 9.43.



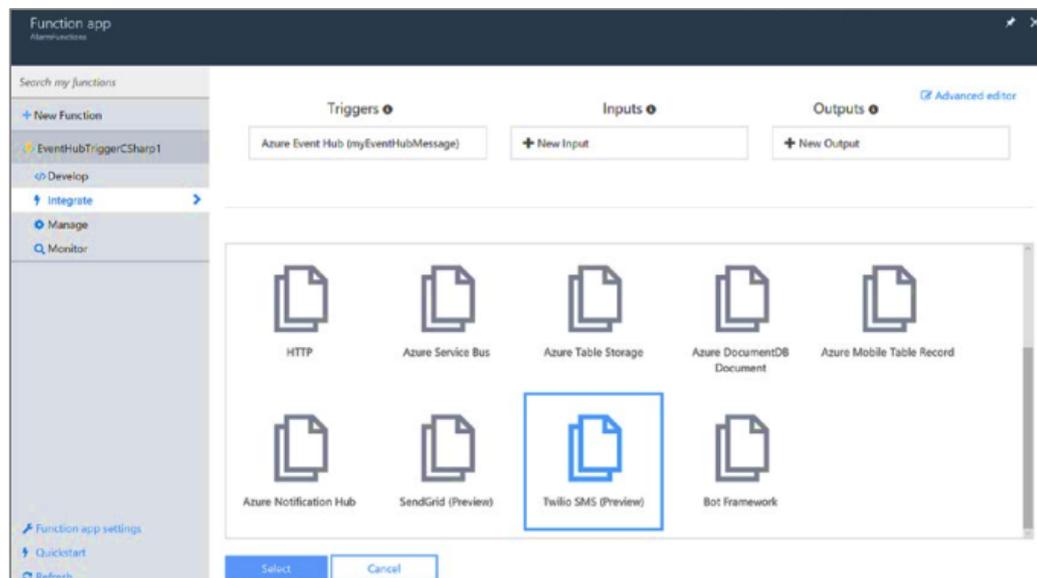
Gambar 9.43. Bilah Fungsi Azure

Perhatikan bahwa layar utama berisi wizard untuk membantu memandu Anda dalam membuat fungsi baru menggunakan templat fungsi untuk skenario umum. Klik ikon + Fungsi Baru untuk menambahkan Fungsi Azure baru. Selanjutnya, Anda akan melihat layar yang mirip dengan Gambar 9.44, di mana Pemrosesan Data telah dipilih dalam filter drop-down untuk template Skenario.



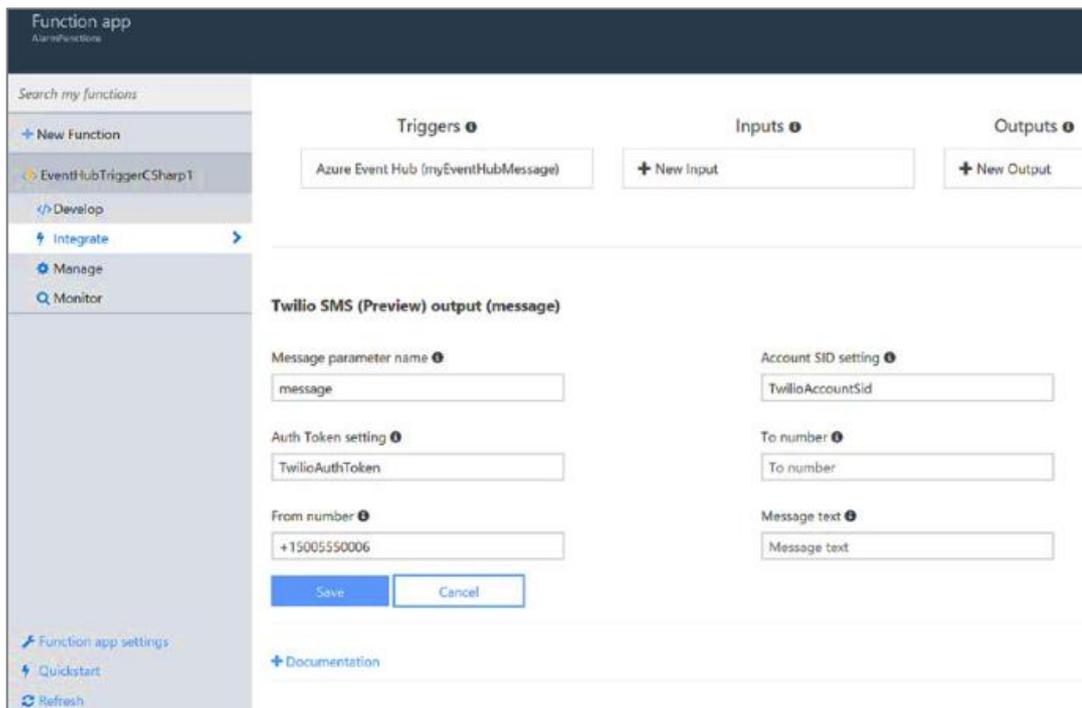
Gambar 9.44 Pilih templat Azure Function untuk CSharp pemacu hub peristiwa

Pilih skenario untuk EventHubTrigger-CSharp lalu gulir ke bawah pada layar untuk memasukkan nama untuk fungsi baru Anda. Perhatikan bahwa Anda juga perlu memasukkan nama hub peristiwa dan informasi koneksi hub peristiwa di bagian bawah layar. Setelah Anda memasukkan parameter yang diperlukan, klik tombol Buat. Anda kemudian akan melihat layar yang mirip dengan Gambar 9.45.



Gambar 9.45 Menambahkan pengikatan keluaran untuk pesan teks SMS Twilio

Hal pertama yang perlu Anda lakukan adalah menambahkan output baru untuk fungsi pemicu hub peristiwa. Untuk melakukan ini, klik opsi Integrasikan di bilah navigasi kiri. Kemudian klik ikon + Output Baru di kiri atas layar untuk menambahkan output baru. Gulir ke bawah sampai Anda melihat Twilio SMS (Pratinjau) dan pilih ubin itu, seperti yang ditunjukkan pada Gambar 9.45. Kemudian klik pada perintah Pilih di bagian bawah layar. Anda kemudian akan disajikan dengan layar yang mirip dengan Gambar 9.46. Di sini, Anda akan menetapkan parameter spesifik untuk fungsi pemicu hub peristiwa.

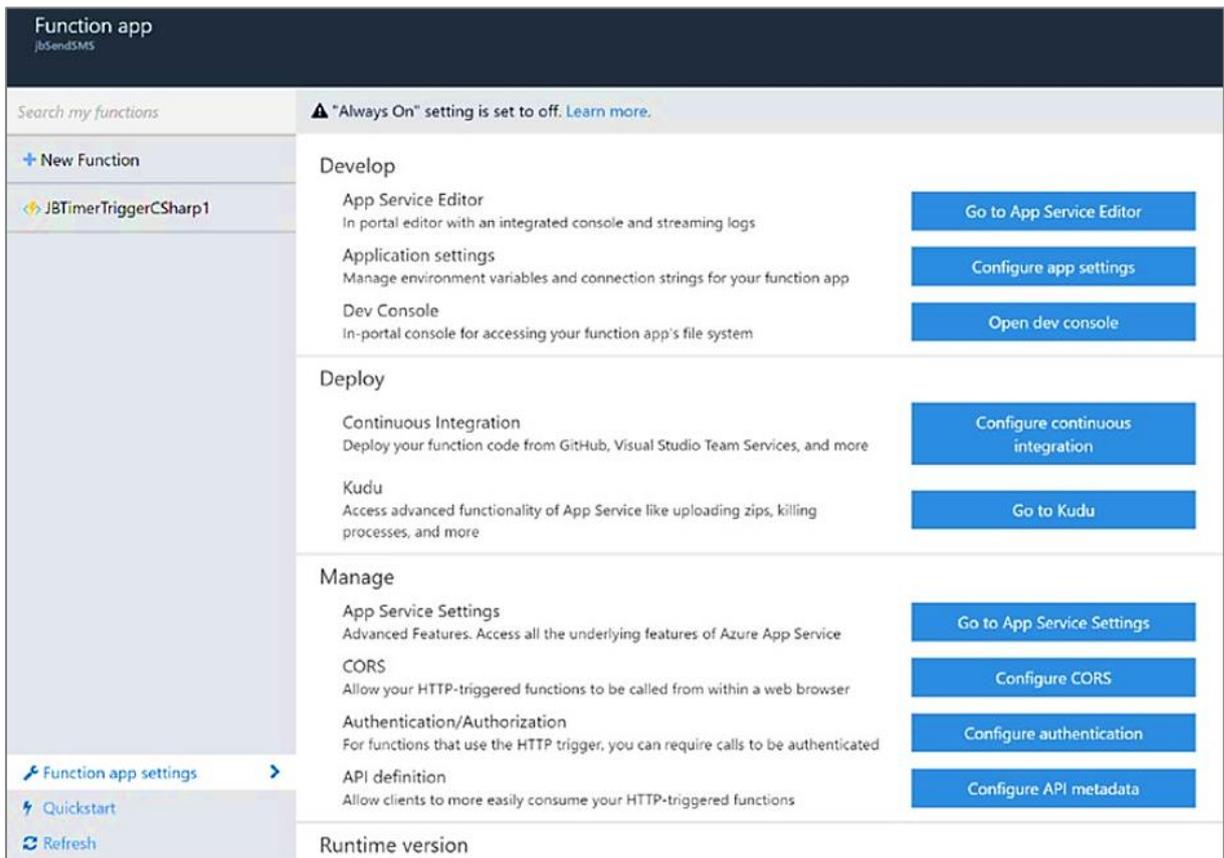


Gambar 9.46. Parameter integrasi fungsi untuk integrasi Twilio

Masukkan nilai parameter berikut:

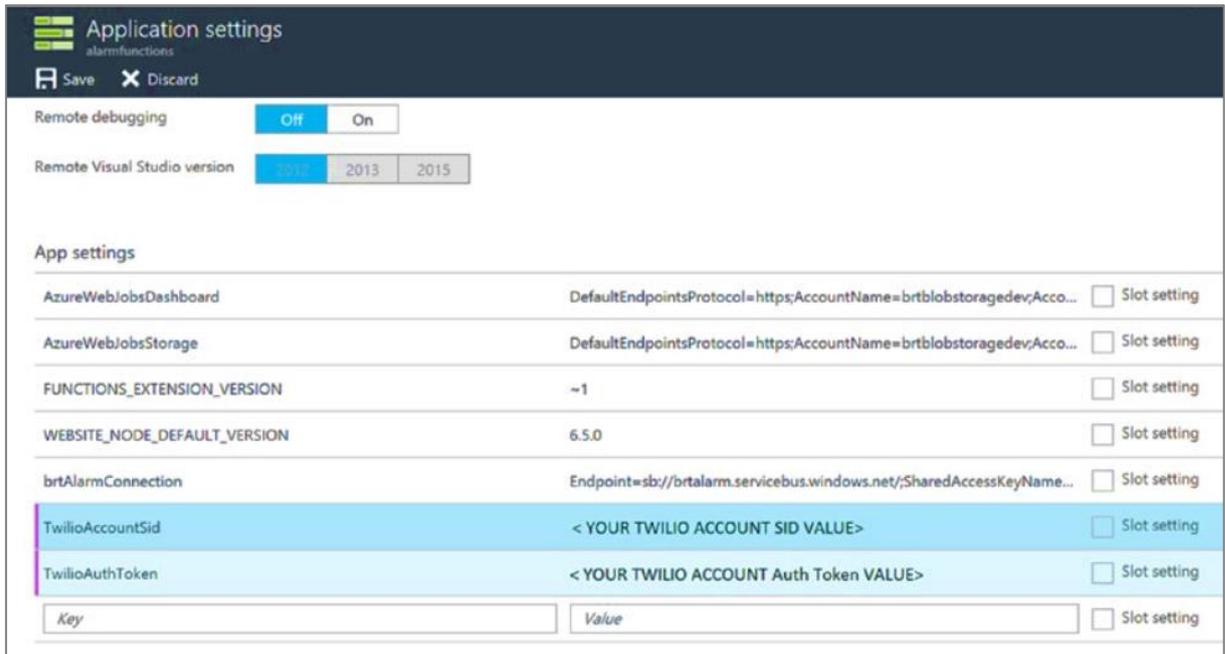
1. Nama Parameter Pesan : pesan
 - ❖ Catatan: Nilai ini harus sesuai dengan nilai pada fungsi.
2. Pengaturan Token Otentikasi : TwilioAuthToken
 - ❖ Catatan: Ini hanya nama variabel untuk variabel pengaturan aplikasi yang akan menyimpan nilai sebenarnya.
3. Dari Nomor : +15005550006
 - ❖ Catatan: Ini adalah nomor Twilio ajaib yang digunakan untuk mengirim teks SMS dengan kredensial tes.
4. Pengaturan SID Akun : TwilioAccountSid
 - ❖ Catatan: Ini hanya nama variabel untuk variabel pengaturan aplikasi yang akan menyimpan nilai sebenarnya.
5. Ke Nomor : Biarkan kosong, karena Anda akan mengisinya melalui kode.
6. Teks Pesan : Biarkan kosong, karena Anda akan mengisinya melalui kode.

Klik pada perintah Simpan di bagian bawah setelah Anda memasukkan parameter yang diperlukan. Langkah selanjutnya adalah membuat pengaturan aplikasi untuk dua variabel baru yang akan Anda gunakan untuk menyimpan kredensial Twilio. Untuk melakukan ini, klik tautan Pengaturan Aplikasi Fungsi di panel navigasi sisi kiri bawah. Selanjutnya, Anda akan melihat layar yang mirip dengan Gambar 9.47, tempat Anda dapat melihat dan memperbarui semua pengaturan yang relevan untuk Fungsi Azure ini.



Gambar 9.47. Konfigurasi pengaturan aplikasi

Karena Anda perlu menambahkan dua variabel pengaturan aplikasi untuk menampung Twilio SID dan token autentikasi, klik tautan untuk Konfigurasi Pengaturan Aplikasi. Ini akan membawa Anda ke panel Pengaturan Aplikasi untuk Fungsi Azure Anda. Layarnya akan tampak seperti Gambar 9.48.



Gambar 9.48. Tambahkan dua pengaturan aplikasi Twilio baru untuk SID dan token autentikasi

Di sini, Anda akan menambahkan dua pengaturan aplikasi baru untuk menyimpan kunci API Twilio. Masukkan nama Pengaturan Aplikasi baru dan nilai terkait yang diperoleh saat Anda mendaftar untuk kunci API dari Twilio:

- `TwilioAccountSid`: <SID Anda>
- `TwilioAuthToken`: <Token Auth Anda>

Ingatlah untuk mengklik ikon Simpan di kiri atas layar Pengaturan Aplikasi setelah Anda selesai.

Sekarang, Anda hanya perlu memperbarui dua file kode kecil dan Anda akan memiliki Fungsi Azure operasional:

- ❖ `Function.json`: Menyimpan informasi pengikatan input/output dasar untuk fungsi tersebut.
- ❖ `Run.csx`: Berisi kode C# yang akan dijalankan ketika fungsi dipicu, dalam hal ini, setiap kali item baru ditambahkan ke hub peristiwa untuk memberi sinyal kondisi alarm.

Untuk memperbarui kode file `Function.json`, klik ikon `</>`Kembangkan di bilah navigasi kiri. Agar file terlihat untuk diedit, Anda harus mengklik ikon Lihat File di bilah navigasi kanan atas. Anda kemudian akan dapat melihat file yang terkait dengan fungsi tersebut.

Klik pada file `Function.json` dan itu akan terbuka di jendela editor. Ganti kode yang ada dengan kode JSON berikut lalu klik perintah Simpan untuk menyimpan file.

```
{
  "bindings": [
    {
      "type": "eventHubTrigger",
```

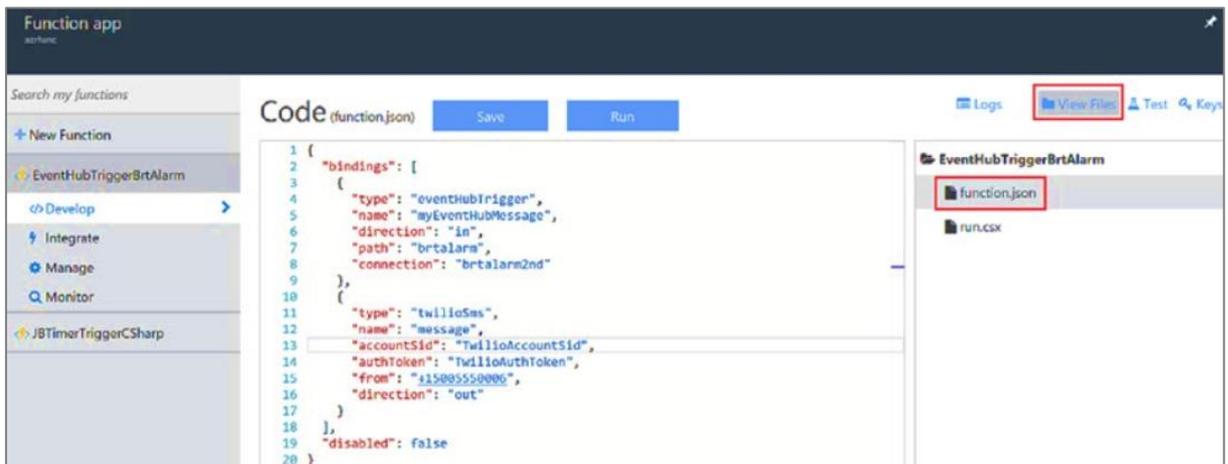
```

    "name": "myEventHubMessage",
    "direction": "in",
    "path": "brtalarm",
    "connection": "brtalarm2nd"
  },
  {
    "type": "twilioSms",
    "name": "message",
    "accountSid": "TwilioAccountSid",
    "authToken": "TwilioAuthToken",
    "from": "+15005550006",
    "direction": "out"
  }
],
"disabled": false
}

```

Code: Function.json

Gambar 9.49 menggambarkan keseluruhan kode Function.json di jendela editor.



Gambar 9.49. Kode Fungsi Azure baru untuk Function.json

Selanjutnya, klik pada file `run.csx`, dan itu akan membuka isi file yang ada ke jendela editor. Ganti kode yang ada dengan kode C# berikut lalu klik perintah Simpan di bagian atas layar untuk menyimpan file.

```

#r "Microsoft.ServiceBus"
#r "Newtonsoft.Json"
#r "Twilio.Api"

using System;
using Microsoft.ServiceBus.Messaging;
using Newtonsoft.Json;
using Twilio;

```

```

public static void Run(string myEventHubMessage, out SMSMessage
message, TraceWriter
log)
{
string jsonContent = myEventHubMessage.ToString();
var alert = JsonConvert.DeserializeObject<EventHubData>(jsonContent);

string eventtime = alert.timerstamp;
string uname = alert.username;
string adminph = alert.adminphone;

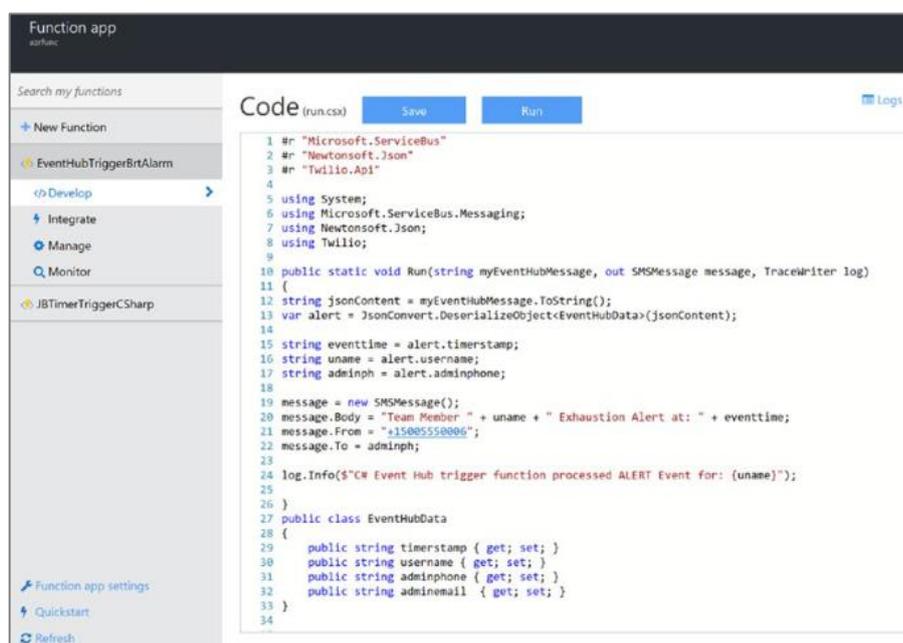
message = new SMSMessage();
message.Body = "Team Member " + uname + " Exhaustion Alert at: " +
eventtime;
message.From = "+15005550006";
message.To = adminph;

log.Info($"Processed ALERT Event for: {uname}");

}
public class EventHubData
{
    public string timerstamp { get; set; }
    public string username { get; set; }
    public string adminphone { get; set; }
    public string adminemail { get; set; }
}

```

Code: run.csx



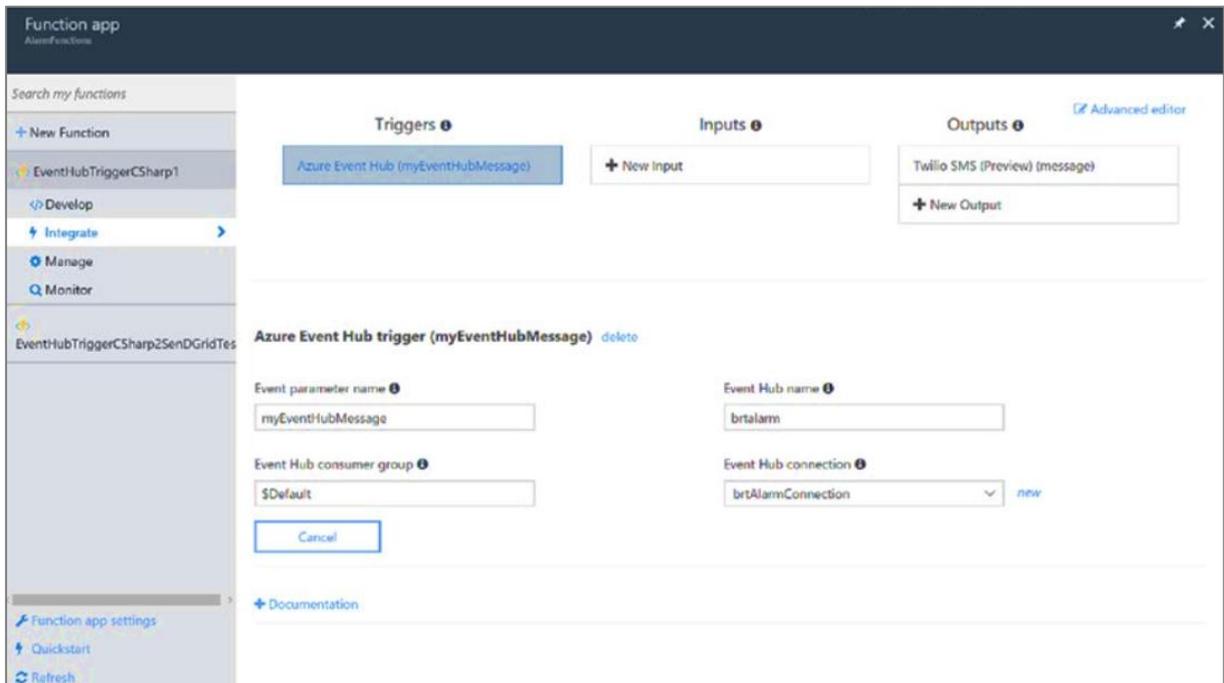
Gambar 9.50. Kode Azure Function C# baru untuk run.csx di jendela editor

Ingatlah bahwa pesan ke pusat peristiwa ini dihasilkan dari pekerjaan Kueri Azure Stream Analytics yang Anda tentukan sebelumnya di bab ini. Dalam pekerjaan itu, Anda menentukan logika mirip SQL untuk mengeluarkan pesan peringatan ke pusat peristiwa jika layanan web memperkirakan anggota tim berada pada tingkat kelelahan fisik. Perhatikan bahwa dalam fungsi tersebut, kami mendefinisikan kelas bernama `EventData` untuk mendeskripsikan tata letak pesan hub peristiwa yang masuk. Kami akan menggunakan kelas ini untuk melakukan deserialisasi string JSON yang diteruskan dari hub peristiwa. Ini agar Anda dapat mengambil bidang pesan tertentu yang diperlukan untuk mengirim pesan teks SMS ke administrator yang bertanggung jawab atas anggota tim tertentu tersebut.

Buat Fungsi Azure untuk Mengirim Peringatan Email melalui SendGrid

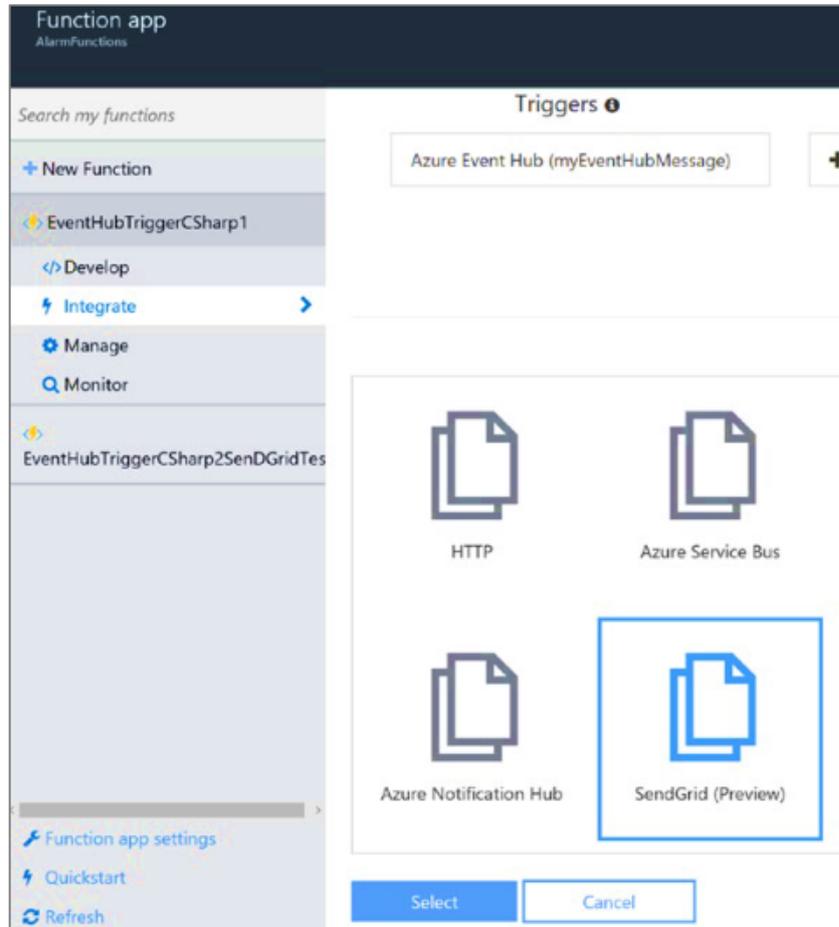
Langkah selanjutnya adalah memperluas fungsi yang ada untuk mengirim email ke administrator anggota tim. Untuk melakukannya, Anda akan menambahkan output untuk SendGrid dan kemudian mengubah kode dalam dua file kode Azure Functions (`Function.json` dan `run.csx`) untuk menangani koneksi keluar dan logika tambahan. Untuk memanfaatkan SendGrid untuk mengirim email, Anda harus terlebih dahulu mendaftar untuk mendapatkan kunci API pengembang gratis di www.SendGrid.com.

Setelah Anda memiliki kunci SendGrid, klik opsi Integrasikan di bilah navigasi kiri bilah Azure Function Editor. Kemudian klik ikon +Output Baru di sudut kanan atas layar, seperti yang digambarkan pada Gambar 9.51.



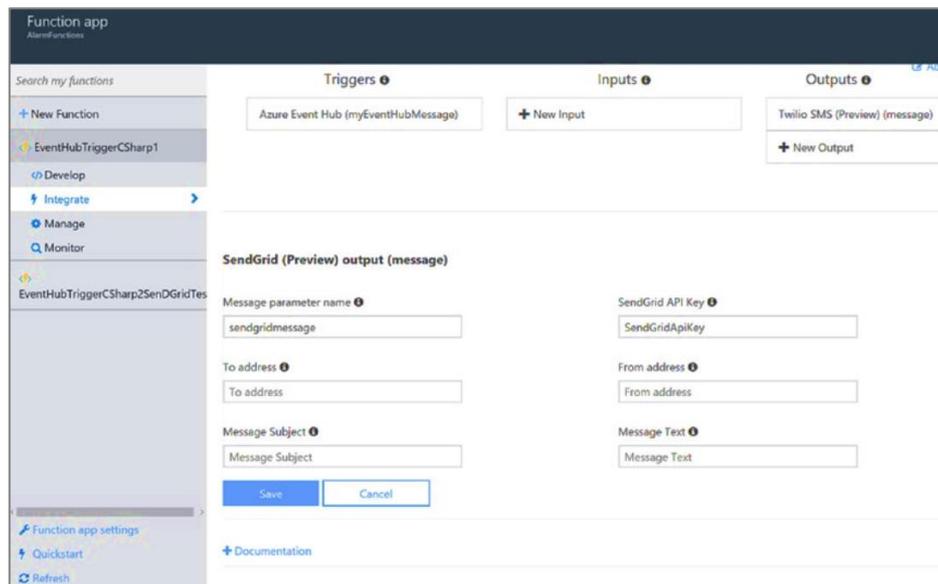
Gambar 9-51. Opsi Integrasi Fungsi Azure, tambahkan output baru

Gulir ke bawah dan pilih opsi SendGrid (Pratinjau) dan klik Pilih, seperti yang digambarkan pada Gambar 9-52.



Gambar 9.52. Memilih opsi keluaran SendGrid untuk Fungsi Azure

Setelah mengklik tombol Select pada Gambar 9.52, Anda selanjutnya akan melihat layar yang mirip dengan Gambar 9.53, di mana Anda dapat memasukkan informasi spesifik untuk mengkonfigurasi output binding untuk SendGrid.



Gambar 9.53. Tambahkan parameter SendGrid keluaran baru

Yang perlu Anda isi di sini hanyalah nama berbeda untuk "parameter pesan" dan kunci API SendGrid Anda. Kami akan mengisi kolom yang tersisa dari kode C# di fungsi. Klik Simpan setelah selesai. Untuk memperbarui kode file `Function.json` agar menyertakan definisi keluaran SendGrid, klik pada `</>` Kembangkan ikon di bilah navigasi kiri. Selanjutnya, klik ikon Lihat File di bilah navigasi kanan atas untuk membuat file terlihat untuk diedit. Klik pada file `Function.json` dan itu akan membuka konten file yang ada ke jendela editor. Ganti kode yang ada dengan kode JSON berikut lalu klik perintah Simpan untuk menyimpan file.

```
{
  "bindings": [
    {
      "type": "eventHubTrigger",
      "name": "myEventHubMessage",
      "direction": "in",
      "path": "brtalarm",
      "connection": "brtAlarmConnection"
    },
    {
      "type": "twilioSms",
      "name": "message",
      "accountSid": "TwilioAccountSid",
      "authToken": "TwilioAuthToken",
      "from": "+15005550006",
      "direction": "out"
    },
    {
      "type": "sendGrid",
      "name": "sendgridmessage",
      "apiKey": "SendGridApiKey",
      "from": "AzureAlerts@TallTowers.com",
      "direction": "out"
    }
  ],
  "disabled": false
}
```

Code: `Function.json`

Selanjutnya klik pada file `run.csx` dan akan terbuka isi file yang ada ke jendela editor. Ganti kode yang ada dengan kode C# berikut dan kemudian klik perintah Simpan di bagian atas layar untuk menyimpan file C# yang diperbarui.

```
#r "Microsoft.ServiceBus"
#r "Newtonsoft.Json"
#r "Twilio.Api"
#r "SendGrid"
#r "Microsoft.Azure.WebJobs.Extensions.SendGrid"
```

```

using System;
using Microsoft.ServiceBus.Messaging;
using Newtonsoft.Json;
using Twilio;
using SendGrid;
using SendGrid.Helpers.Mail;

public static void Run(string myEventHubMessage, out SMSMessage
message, out Mail
sendgridmessage, TraceWriter log)
{
string jsonContent = myEventHubMessage.ToString();
var alert = JsonConvert.DeserializeObject<EventData>(jsonContent);
string eventtime = alert.timerstamp;
string uname = alert.username;
string adminph = alert.adminphone;
string admemail = alert.adminemail;

//Send SMS TEXT ALERT via Twilio API
message = new SMSMessage();
message.Body = "Team Member " + uname + " Exhaustion Alert at: " +
eventtime;
message.From = "+15005550006";
message.To = adminph;

//Send EMAIL ALERT via SendGrid API
var personalization = new Personalization();
personalization.AddTo(new Email(admemail));
string subject = "ALERT NOTIFICATION - Team Member " + uname + "
Exhaustion Alert
received at: " + eventtime;
var messageContent = new Content("text/html", "ALERT NOTIFICATION -
Team Member " +
uname + " Exhaustion Alert received at: " + eventtime);

sendgridmessage = new Mail();
sendgridmessage.Subject = subject;
sendgridmessage.AddContent(messageContent);
sendgridmessage.AddPersonalization(personalization);

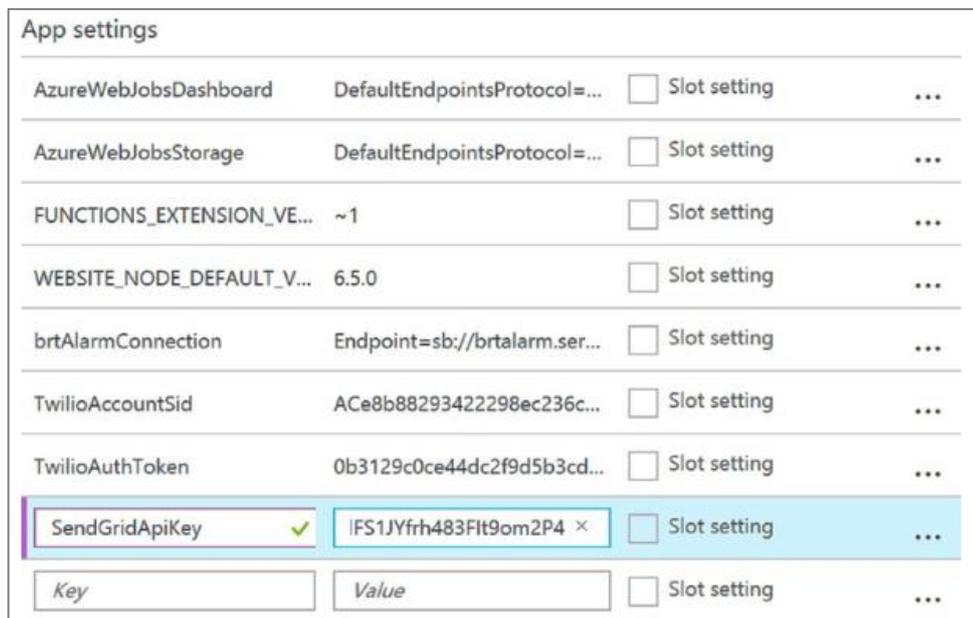
log.Info($"Processed ALERT Event for: {uname}");
}
public class EventHubData
{
public string timerstamp { get; set; }
public string username { get; set; }
public string adminphone { get; set; }
public string adminemail { get; set; }
}

```

}

Code: run.csx

Hal berikutnya yang perlu Anda lakukan adalah menambahkan pengaturan Konfigurasi Aplikasi baru untuk kunci SendGridAPI. Namanya harus sama dengan yang digunakan di file `Function.json`, yaitu `SendGridApiKey`. Mulailah dengan menavigasi ke Azure Functions Anda melalui portal, lalu klik tautan Pengaturan Aplikasi Fungsi di kiri bawah layar. Selanjutnya, klik pada perintah Pengaturan Aplikasi. Ini akan membawa Anda ke panel Pengaturan Aplikasi untuk Fungsi Azure Anda. Layarnya akan tampak seperti Gambar 9.54.



Gambar 9.54. Menambahkan pengaturan aplikasi baru untuk SendGridAPIKey

Di sini, Anda akan menambahkan pengaturan aplikasi baru untuk menyimpan kunci API SendGrid. Masukkan ID Pengaturan Aplikasi baru sebagai `SendGridApiKey` dan kunci API Anda yang sesuai sebagai nilainya. Ingatlah untuk mengklik ikon Simpan di kiri atas panel Pengaturan Aplikasi setelah Anda selesai.

Memantau Fungsi Azure

Setelah Anda selesai mengonfigurasi pengikatan dan logika Azure Function, ada beberapa cara untuk memantau fungsi Anda. Anda bisa mendapatkan umpan balik langsung tentang bagaimana fungsi Anda berjalan melalui tampilan Logs, yang terlihat di bawah jendela editor kode pada Gambar 9.55.

```

Code (run.csx) Save Run
1 #r "Microsoft.ServiceBus"
2 #r "Newtonsoft.Json"
3 #r "Twilio.Api"
4
5 using System;
6 using Microsoft.ServiceBus.Messaging;
7 using Newtonsoft.Json;
8 using Twilio;
9
10 public static void Run(string myEventHubMessage, out SMSMessage message, TraceWriter log)
11 {
12     string jsonContent = myEventHubMessage.ToString();
13     var alert = JsonConvert.DeserializeObject<EventData>(jsonContent);
14
15     string eventtime = alert.timestamp;
16     string uname = alert.username;
17     string adminph = alert.adminphone;
18
19     message = new SMSMessage();
20     message.Body = "Team Member " + uname + " Exhaustion Alert at: " + eventtime;
21     message.From = "+15005550006";
22     message.To = adminph;
23
24 }

```

Logs Start Clear Copy Logs Expand Close

```

2017-02-03T15:15:43.503 Function started (Id=06d33ce6-8a2b-4b30-bec8-465c7963ad4e)
2017-02-03T15:15:43.503 C# Event Hub trigger function processed ALERT Event for: Cleryan.Guga
2017-02-03T15:15:43.503 Function completed (Success, Id=06d33ce6-8a2b-4b30-bec8-465c7963ad4e)
2017-02-03T15:15:47.773 Function started (Id=2932dfc9-c976-4650-be53-6a32135112c1)
2017-02-03T15:15:47.773 C# Event Hub trigger function processed ALERT Event for: Maelll.Origa
2017-02-03T15:15:47.773 Function completed (Success, Id=2932dfc9-c976-4650-be53-6a32135112c1)
2017-02-03T15:15:49.640 Function started (Id=34715e7b-a3fb-4e1b-b78d-bbd03fc3094f)
2017-02-03T15:15:49.640 C# Event Hub trigger function processed ALERT Event for: Canliver.Ishtik

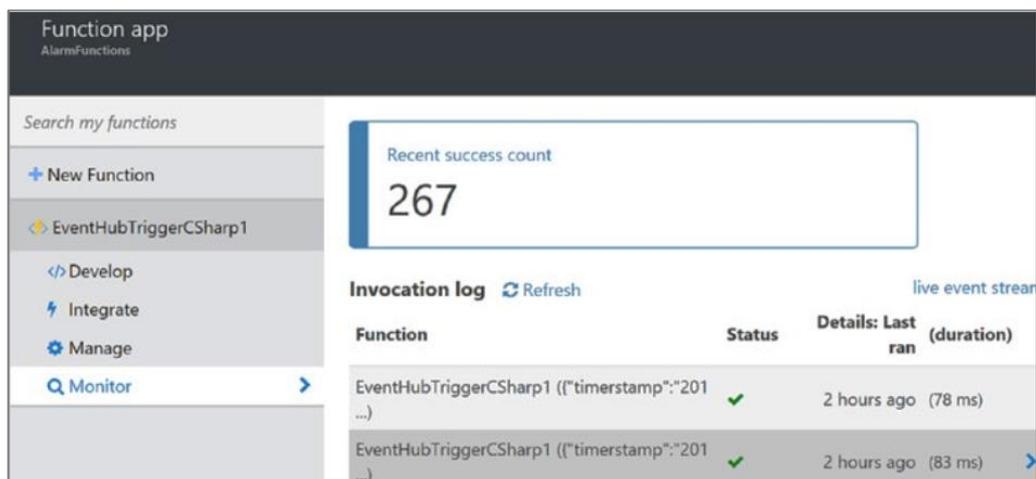
```

Gambar 9.55 Jendela Azure Function Logs menampilkan output pencatatan waktu nyata

Perhatikan bahwa Anda dapat melihat output dari perintah `log.Info()` seperti yang Anda gunakan dalam fungsi C#:

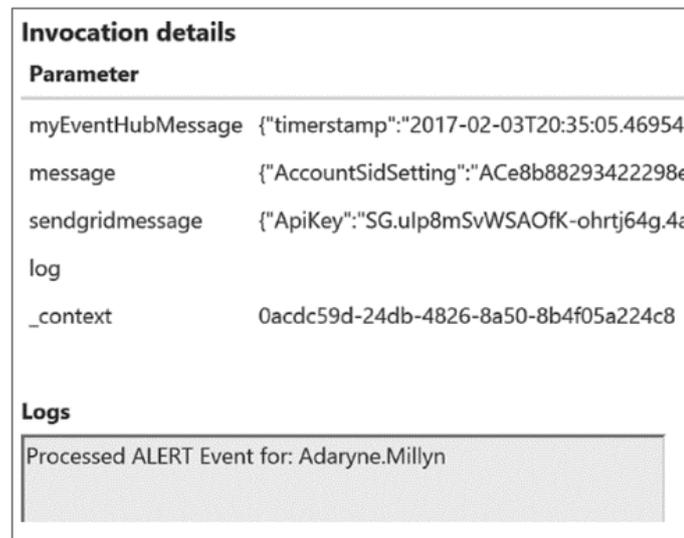
```
log.Info($"Processed ALERT Event for: {uname}");
```

Metode lainnya adalah dengan menggunakan alat pemantauan bawaan di Portal Azure untuk Azure Functions. Mulailah dengan mengklik ikon Monitor di bilah navigasi kiri pada bilah Azure Function. Selanjutnya, Anda akan melihat daftar semua peristiwa dan pesan keluaran pencatatan yang telah dihasilkan oleh panggilan Fungsi Azure ini. Gambar 9.56 memperlihatkan tangkapan layar jendela pemantauan Fungsi Azure.



Gambar 9.56. Azure Functions: memantau peristiwa fungsi

Jika Anda mengklik acara untuk suatu fungsi, Anda akan melihat panel yang menampilkan detail pemanggilan tentang pemanggilan fungsi tersebut. Gambar 9.57 menggambarkan contoh panel pemanggilan untuk panggilan peristiwa Azure Function yang dipilih.



Gambar 9.57. Melihat detail pemanggilan untuk suatu acara

Perhatikan bahwa data Parameter dan Log yang ditampilkan pada Gambar 9-57 bisa sangat berguna untuk menyelesaikan data konfigurasi atau men-debug pernyataan pemrograman C#.

JALANKAN SIMULATOR

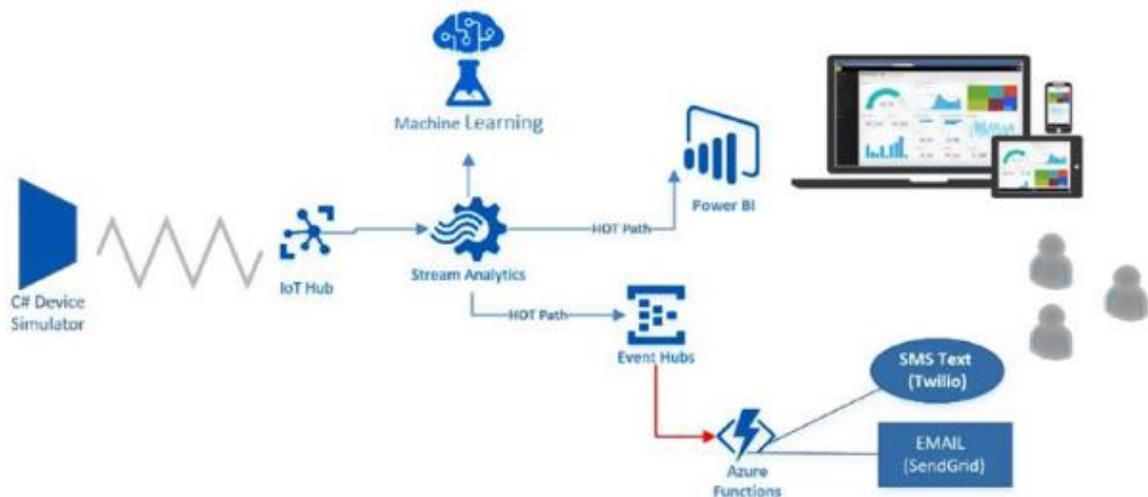
Tersembunyi di dalam, kedalaman gelap repositori GitHub untuk buku ini adalah aplikasi konsol C# yang sangat berguna yang dapat bertindak sebagai “simulator” perangkat yang efektif sebagai pengganti perangkat sensor nyata, anggota tim langsung, dan alat berat komersial sebenarnya. Simulator perangkat sangat berguna untuk menghasilkan data sensor yang realistis untuk (15) anggota tim untuk masing-masing dari tiga entitas bisnis dalam implementasi referensi. Badan usaha tersebut terkenal di kalangan terbatas dan tercantum di sini untuk referensi Anda:

- Badger yang Rumit
- Menara Tinggi
- WigiTech

Jalur folder Github ke kode C# simulator perangkat adalah

```
\devices\device-teamsim
```

Setelah titik ini di jalur folder, Anda akan melihat tiga jalur kode berbeda di repositori—satu untuk setiap entitas bisnis. Hal ini bertujuan agar simulator perangkat setiap entitas selanjutnya dapat disesuaikan lebih lanjut untuk mengakomodasi masing-masing entitas bisnis dalam referensi implementasinya.



Gambar 9.58 Peran simulator perangkat dalam arsitektur referensi

Manfaat tambahan dari pendekatan ini adalah Anda dapat menjalankan simulator perangkat untuk setiap perusahaan secara bersamaan. Menjalankan simulator perangkat secara bersamaan membantu memfasilitasi pemodelan aliran data lingkungan IoT multi-penyewa yang sebenarnya. Peran simulator perangkat tidak dapat diabaikan, karena sangat penting bagi keberhasilan setiap proyek IoT. Ini adalah cara terbaik untuk menghasilkan data sensor IoT yang representatif dan kemudian mengirimkan data tersebut sepenuhnya melalui solusi Azure yang Anda buat sebagai bagian dari implementasi referensi. Peran simulator perangkat tercermin pada Gambar 9.58.

Simulasikan Kondisi Stres Sensor dengan Mode BRZRKR

Salah satu manfaat utama solusi implementasi referensi adalah kemampuan untuk mendeteksi masalah dan mengirimkan pemberitahuan peringatan secara otomatis. Untuk itu, fokus dasbor Power BI serta peringatan dan notifikasi adalah menghasilkan visualisasi dan peringatan untuk kondisi abnormal ini. Untuk menirunya, ada fitur bernama mode BRZRKR yang bisa disesuaikan dengan lingkungan Anda. Gambar 9.59 mengilustrasikan potongan kode dalam fungsi SendTelemetry untuk menyesuaikan pengaturan mode BRZRKR.

```
private static void SendTelemetry(Teammate teammate)
{
    var rowindex = 0;

    const double brzrkr = 1.5; // inflate readings to quickly simulate alarm conditions
```

Gambar 9.59. Mode Perangkat Simulator BRZRKR

Untuk memulai, buka proyek simulator perangkat C# di Visual Studio 2015. Buka solusi Tall Towers, yang dapat ditemukan di lokasi folder berikut di repositori GitHub:
 devices\device-teamsim\ talltowers.

Periksa Pengaturan di file app.config untuk memastikannya cocok dengan lingkungan Anda. Kemudian Anda dapat membuat dan menjalankan simulator. Lihat Gambar 9-60.

```

Simulating teammate1
Profile Id : 8dfef6d5-2302-460f-af61-fa2fd3c75e8b
Device Id : 537cfd35-1d39-4aeb-bf45-0ef6faaf300c
Simulating teammate2
Profile Id : a7591ba2-e534-48fd-af1a-50e56c0e396a
Device Id : 2aadb103-fae8-4d84-a07b-15ed48e33a55
Simulating teammate3
Profile Id : a8569261-b13a-421f-938c-81adef612e25
Device Id : bfc37538-9800-4d56-b1da-6ba9a9453f23
Simulating teammate4
Profile Id : fd36c995-e599-4bc3-8c28-0d5b6fe54624
Device Id : bf8e2539-1a0d-4642-94ea-088a44ab0b3f
Simulating teammate5
Profile Id : 33e37484-31ac-4b34-b928-6520175b5d74
Device Id : e4867159-f070-409c-a86e-dcf848088eb3

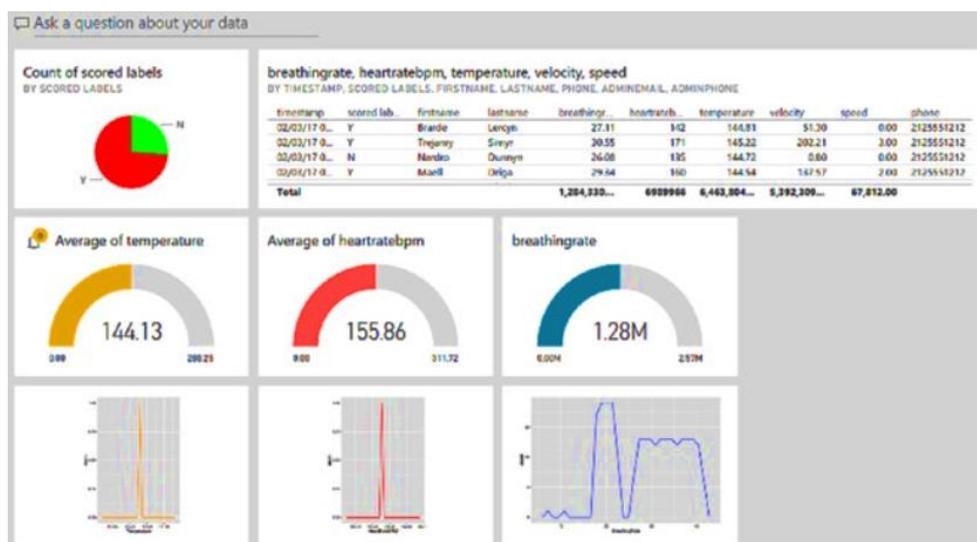
```

Gambar 9.60. Contoh keluaran simulator Perangkat

Setelah memulai aplikasi konsol simulator perangkat C#, Anda biasanya memulai pekerjaan Azure Stream Analytics. Setelah kedua aplikasi dimulai sepenuhnya, Anda mulai melihat data muncul di dasbor Power BI Anda. Bergantung pada pengaturan mode BRZKR dan cara Anda mengonfigurasi Azure Functions untuk pemberitahuan, Anda mungkin juga mulai melihat pesan peringatan muncul di ponsel atau di kotak masuk email Anda.

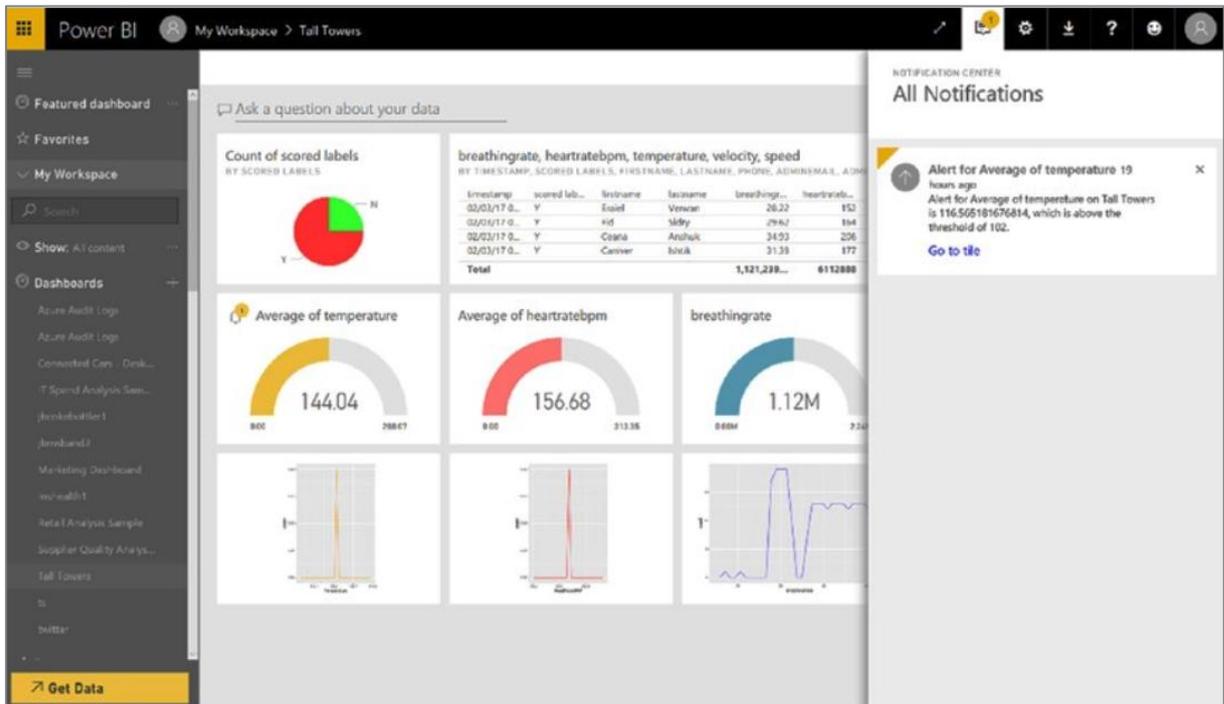
End-to-End: Simulator -> Dasbor dan Peringatan

Di bagian ini, Anda melihat bagaimana semuanya menyatu dalam implementasi referensi. Kami akan meninjau beberapa visualisasi dasbor dan artefak peringatan yang dihasilkan saat kami menjalankan data melalui solusi menggunakan simulator perangkat. Gambar 9.61 menggambarkan dasbor Power BI saat simulator perangkat dijalankan di latar belakang dalam mode BRZRKR.



Gambar 9.61. Dasbor Power BI real-time untuk implementasi referensi

Perhatikan bahwa pada Gambar 9.61, terdapat ikon alarm yang terlihat di sudut kiri atas pengukur suhu rata-rata. Ini terjadi setelah kami menyetel alarm secara manual di awal bab. Gambar 9.62 menampilkan pesan pemberitahuan dasbor Power BI spesifik yang dihasilkan untuk alarm ini. Anda dapat melihat bahwa hal ini dipicu oleh melebihi ambang batas yang Anda tetapkan sebelumnya.



Gambar 9.62. Contoh pemberitahuan peringatan dasbor Power BI

Dalam kasus ini, peringatan pemberitahuan Power BI berbunyi sebagai berikut (dari Gambar 9-62):

Alert for Average of temperature on Tall Towers is 116.565181676814, which is above the threshold of 102.

Perhatikan juga bahwa subjek peringatan Power BI ini adalah “Peringatan untuk Suhu rata-rata sehari”, yang berarti Anda akan menerima peringatan ini sekali sehari. Meskipun peringatan dashboard ini sangat berguna, kita masih memerlukan mekanisme peringatan yang lebih efektif dan langsung untuk menghubungi pihak yang tepat, pada perangkat yang tepat, dan pada waktu yang tepat. Kemampuan ini memungkinkan Anda mengirimkan pemberitahuan peringatan keluar yang penting kepada pihak yang paling berkepentingan. Dan semua ini dapat dengan mudah disampaikan melalui metode sarana komunikasi modern “*lowest-common-denominator*”, yaitu teks dan email.

Gambar 9.63 mewakili contoh email yang dihasilkan oleh kode C# dan API SendGrid yang dibahas sebelumnya dalam bab ini.



Gambar 9.63. Contoh pemberitahuan pemberitahuan email yang dihasilkan oleh pengikatan output SendGrid di Fungsi Azure

Dalam skenario ini, kemampuan untuk secara otomatis memicu, menghasilkan, dan mengirim pesan teks SMS dan email menawarkan keunggulan kompetitif bagi tiga entitas bisnis implementasi referensi. Manfaat ini sering kali diwujudkan dalam bentuk peningkatan kesehatan anggota tim, pengurangan kecelakaan, pengurangan tanggung jawab hukum, dan peningkatan kepuasan pelanggan.

9.10 RINGKASAN

Bab ini membahas topik visualisasi, pemberitahuan, dan pemberitahuan menggunakan Power BI dan Azure Functions. Kami memulai dengan melihat lanskap pelaporan modern dan kemudian memberikan gambaran umum tentang kemampuan platform Microsoft Power BI, termasuk versi dan komponen berikut:

- Layanan Power BI
- Desktop Power BI
- Power BI Tertanam
- Power BI Ponsel
- API ReST
- Visual Khusus
- Kueri Bahasa Alami
- Integrasi Cortana
- Desktop Power BI
- Visualisasi Skrip R

Kami kemudian menjelajahi penggunaan beberapa layanan Azure—termasuk hub peristiwa, Stream Analytics, dan Functions—untuk mengaktifkan solusi pemberitahuan dan pemberitahuan otomatis dengan pengkodean minimal. Kami memeriksa teknologi Azure berikut:

- *Azure Event Hubs*: Digunakan sebagai mekanisme pemicu untuk Azure Functions
- *Analisis Streaming*: Output ke kemampuan Event Hub
- *Fungsi Azure*: Mengirim teks SMS (melalui Twilio) dan mengirim email peringatan (melalui SendGrid)

Kami kemudian “mewujudkannya” dengan menggabungkan semua kemampuan Azure ini ke dalam implementasi referensi. Bagian terakhir adalah simulator perangkat dan kemampuan

untuk menghasilkan kondisi normal dan alarm untuk menguji sistem end-to-end. Dengan adanya simulator perangkat, kami kemudian dapat dengan cepat membuat data simulasi dan menghasilkan visualisasi dasbor yang kaya dan interaktif menggunakan Power BI. Dengan sedikit usaha tambahan, tampilan dasbor yang sama langsung tersedia di semua versi Power BI, termasuk versi desktop, web, dan seluler. Selain visualisasi dan dasbor yang kaya, kami juga dapat menerapkan peringatan dan pemberitahuan dengan cepat dan mudah dengan memanfaatkan beberapa layanan Azure seperti hub peristiwa dan Azure Functions. Dalam dunia bisnis yang serba cepat saat ini, yang terus-menerus beroperasi pada “kecepatan Internet”, terdapat banyak persyaratan penting untuk dasbor dan visualisasi operasional yang sangat efisien, adaptif, untuk membantu meningkatkan hasil bisnis dengan menghindari skenario yang tidak menguntungkan. Saat Anda menambahkan kemampuan berbasis cloud seperti Machine Learning dan analisis prediktif, Anda dapat mulai mengelola hasil bahkan sebelum hasil tersebut terjadi. Gabungkan kemampuan prediktif ini dengan peringatan dan notifikasi yang tepat waktu, dapat ditindaklanjuti, dan efektif, dan Anda akan mendapatkan semua fondasi arsitektur bisnis cloud modern yang hebat.

BAB 10

KEAMANAN DAN IDENTITAS

Bab ini mengulas beberapa topik keamanan utama termasuk pemodelan ancaman, protokol keamanan, enkripsi, dan manajemen kunci serta pengelolaan identitas pengguna dan mendukung multi-tenancy dalam solusi Anda.

10.1 PEMODELAN ANCAMAN

Kejahatan dunia maya dan keamanan IoT telah menjadi berita utama selama setahun terakhir. Mendefinisikan strategi keamanan untuk solusi IoT yang menggunakan platform cloud publik memerlukan pemahaman tentang area kerentanan dan vektor serangan yang mungkin dimanfaatkan oleh penjahat dunia maya. Keamanan bukanlah solusi; ini adalah proses berkelanjutan yang memerlukan disiplin dan analisis, peninjauan, dan tindakan yang konstan.

Vektor ancaman dalam konteks solusi IoT ditentukan oleh zona, di mana setiap zona menguraikan area permukaan serangan. Tim desain menggunakan definisi zona ini untuk mengartikulasikan strategi keamanan, tumpukan teknologi, pemantauan, peninjauan, dan proses respons. Zona dalam solusi IoT mencakup lingkungan lokal yang terdiri dari jaringan kabel, nirkabel, dan seluler, perangkat yang Anda sebarkan ke lingkungan tersebut, aplikasi yang Anda berikan kepada pengguna akhir, dan segala sesuatu di antaranya.

Pemodelan ancaman membantu Anda memahami bagaimana penyerang berusaha berkompromi dan mendapatkan akses ke sistem. Ini juga mendefinisikan proses dan alat yang akan Anda manfaatkan untuk memitigasi serangan. Pemodelan ancaman harus menjadi aktivitas yang dilakukan pada waktu desain dan berlanjut sepanjang masa aplikasi. Strategi mitigasi keamanan harus didefinisikan dan terus diperbarui seiring berkembangnya solusi. Kegiatan pemodelan ancaman terdiri dari empat langkah:

- ❖ Memodelkan aplikasi
- ❖ Menghitung ancaman
- ❖ Mengurangi ancaman
- ❖ Memvalidasi mitigasi

Proses pemodelan ini dimulai dengan pembuatan diagram arsitektur solusi. Semua elemen dalam diagram merupakan area serangan permukaan yang potensial. STRIDE adalah model klasifikasi ancaman yang dapat digunakan untuk mengatur potensi ancaman dan strategi mitigasi yang diketahui saat Anda melakukan analisis pada arsitektur Anda.

STRIDE adalah singkatan dari:

- ❖ *Spoofing*: Penyerang berpura-pura menjadi seseorang atau suatu layanan yang bukan dirinya
- ❖ *Tampering*: Penyerang memodifikasi data saat transit
- ❖ *Penolakan*: Pengguna melakukan suatu tindakan dan kemudian mengklaim bahwa mereka sebenarnya tidak melakukannya, misalnya transaksi kartu kredit

- ❖ *Pengungkapan Informasi*: Dapatkah penyerang atau pengguna mengakses data yang tidak boleh mereka lihat
- ❖ *Denial of Service*: Penyerang menurunkan layanan melalui kelebihan beban atau pengalihan
- ❖ *Peningkatan Hak Istimewa*: Penyerang dapat memperoleh hak istimewa yang biasanya tidak mereka miliki

Layanan dalam solusi Anda akan tunduk pada klasifikasi STRIDE yang berbeda. Misalnya:

- Proses dapat diklasifikasikan menggunakan STRIDE
- Aliran data diklasifikasikan menggunakan komponen TID STRIDE
- Penyimpanan data diklasifikasikan menggunakan TID, dan terkadang R, jika penyimpanan data berupa file log
- Entitas eksternal diklasifikasikan menggunakan SRD

Zona Pemodelan Ancaman dan IoT

Zona dalam solusi IoT dapat ditentukan sebagai berikut:

- * * Lokal
- * * Perangkat
- * * Gerbang Awan
- * * Layanan Cloud

Zona Lokal

Salah satu bagian terpenting dari solusi IoT sehubungan dengan keamanan adalah perangkat yang diterapkan ke seluruh dunia—ke dalam lingkungan fisik yang mungkin tidak dapat kita kendalikan. Zona lokal ini rentan terhadap vektor serangan fisik dan digital. Hubungan antara pengguna akhir atau sensor dan perangkat menentukan batas transisi data. Pada batas inilah penyerang akan memusatkan perhatiannya. Ruang fisik di mana data tersebut disebar, segmen jaringan yang dimilikinya, dan cara data diterima dan dikirim harus dipertimbangkan.

Singkat tapi tentu saja bukan daftar pertanyaan lengkap yang harus diajukan oleh tim desain selama analisis zona lokal:

- Apakah perangkat akan berada di ruangan terkunci dengan akses terbatas?
- Apakah pengguna akhir akan berinteraksi langsung dengan perangkat, seperti kunci atau termostat?
- Apakah perangkat terhubung ke jaringan nirkabel terbuka, terhubung ke operator seluler, atau menggunakan LAN berkabel?
- Apakah perangkat berada pada segmen jaringannya sendiri?
- Bagaimana cara pencabutan akses perangkat jika rusak atau diganti?

Zona Perangkat

Zona perangkat berfokus pada cara kerja bagian dalam perangkat. Sistem operasi dan firmware yang menjalankan logika aplikasi Anda, menyambung ke jaringan, menerima dan mengirim data, menyimpan data secara lokal, dan sebagainya dianalisis. Salah satu pertimbangan penting adalah bagaimana akun pengguna dan admin dibuat dan kata sandi mereka dipelihara. Perangkat IoT konsumen diketahui rentan dalam hal ini karena produsen

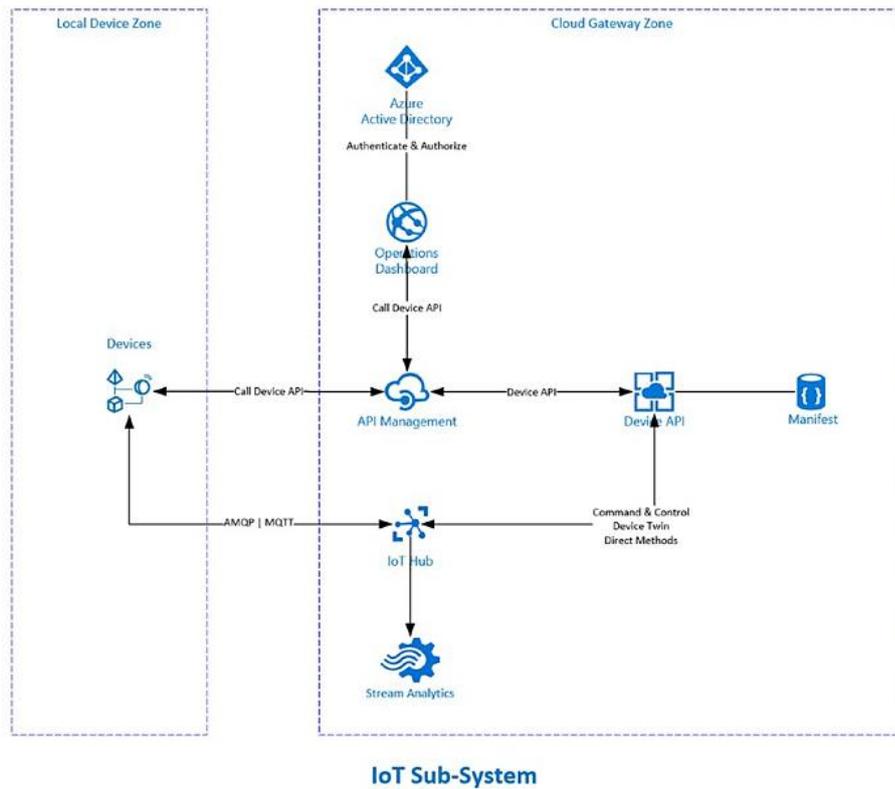
lemah dalam desainnya dan pengguna akhir mungkin tidak cukup paham secara teknis untuk mempertimbangkan implikasi dari kata sandi 12345. Berikut adalah daftar pertanyaan desain yang singkat namun tentu saja tidak lengkap. tim harus bertanya selama analisis zona perangkat:

- ❖ Sistem operasi apa yang tertanam pada perangkat?
- ❖ Bahasa apa yang digunakan untuk mengimplementasikan firmware/aplikasi?
- ❖ Apakah tim firmware membiarkan port terbuka untuk koneksi jarak jauh dan debugging?
- ❖ Bagaimana firmware ditingkatkan?
- ❖ Akun pengguna/admin apa yang dibuat dan bagaimana kredensialnya dipertahankan?
- ❖ Bagaimana cara perangkat diinstal pada jaringan lokal?
- ❖ Bagaimana cara perangkat mengautentikasi ke cloud gateway?
- ❖ Apakah data dienkripsi saat terbang dan diam?
- ❖ Protokol transmisi data apa (MQTT, AMQP, HTTPs, atau lainnya) yang digunakan untuk berkomunikasi secara lokal dan dengan cloud?

Seperti yang telah kita bahas di bab sebelumnya, kami menggunakan mekanisme autentikasi dua fase yang menggabungkan panggilan HTTP aman ke API Perangkat untuk mengambil manifes perangkat diikuti dengan panggilan aman melalui MQTT atau AMQP ke IoT Hub untuk menyambungkan dan menyiapkan keduanya. komunikasi -cara antara perangkat dan cloud. Kami juga mendemonstrasikan penggunaan metode kembar dan langsung perangkat untuk melakukan operasi manajemen perangkat yang aman (lihat Gambar 10.1).

Zona Gerbang Awan

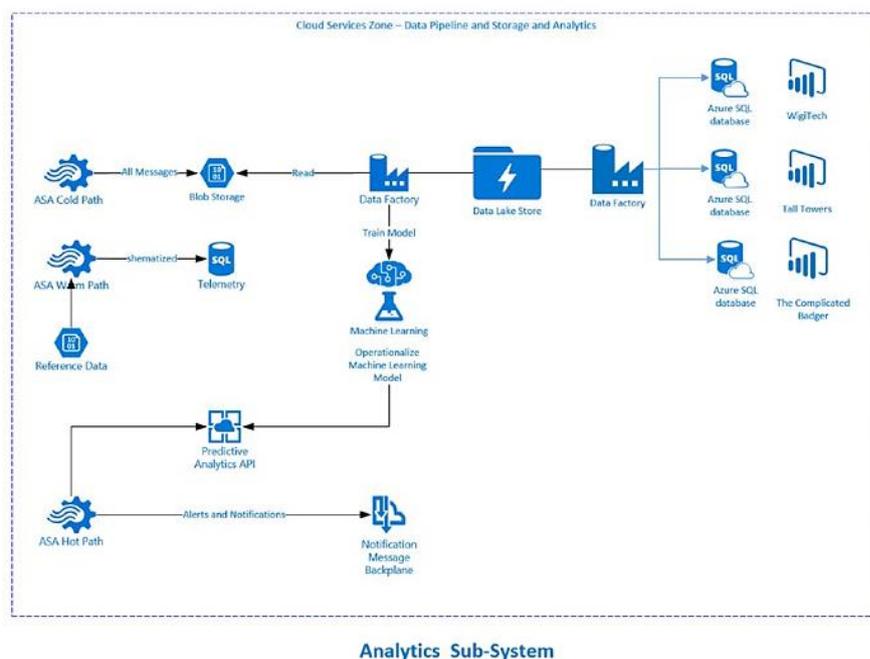
Gateway cloud adalah layanan yang dihosting di cloud atau layanan yang menyediakan titik akhir aman cloud publik bagi perangkat untuk mendaftar, mengautentikasi, dan mengirim serta menerima pesan. Gateway cloud harus mampu menyediakan layanan ini di beberapa zona perangkat. Dalam arsitektur solusi kami, IoT Hub dan Device API mewakili para pemain di zona gateway cloud. Mereka mengekspos titik akhir aman yang memungkinkan perangkat terhubung dan berkomunikasi. Gateway protokol dan/atau gateway SMS mungkin juga berada di zona ini jika Anda menyambung ke perangkat lama dalam solusi Anda (lihat Gambar 10.1).



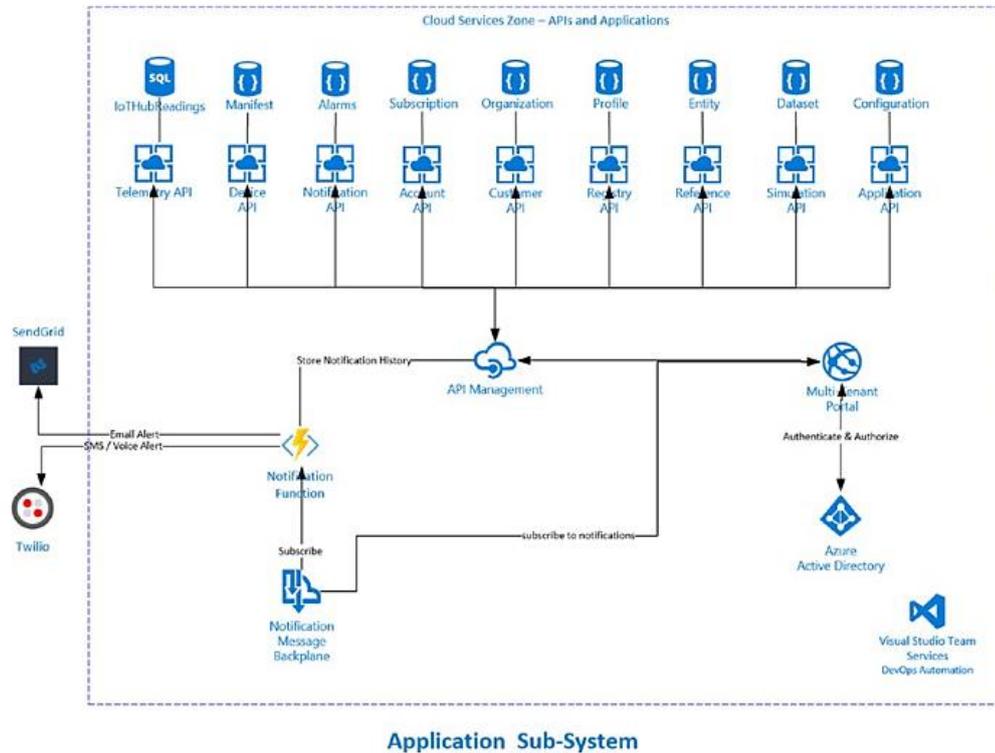
Gambar 10.1. Zona gateway lokal, perangkat, dan cloud

Zona Layanan Cloud

Gateway cloud menyediakan akses ke pesan masuk melalui layanan internal yang dihosting di cloud di Cloud Services Zone yang menyediakan penyerapan data, pemrosesan aliran, penyimpanan, analisis tingkat lanjut, dan integrasi aplikasi melalui API, antrean, dan pemrosesan batch (lihat Gambar 10 -2 dan 10-3).



Gambar 10.2. Zona layanan cloud, saluran data, penyimpanan, dan analitik



Gambar 10.3. Zona layanan cloud, API, dan aplikasi

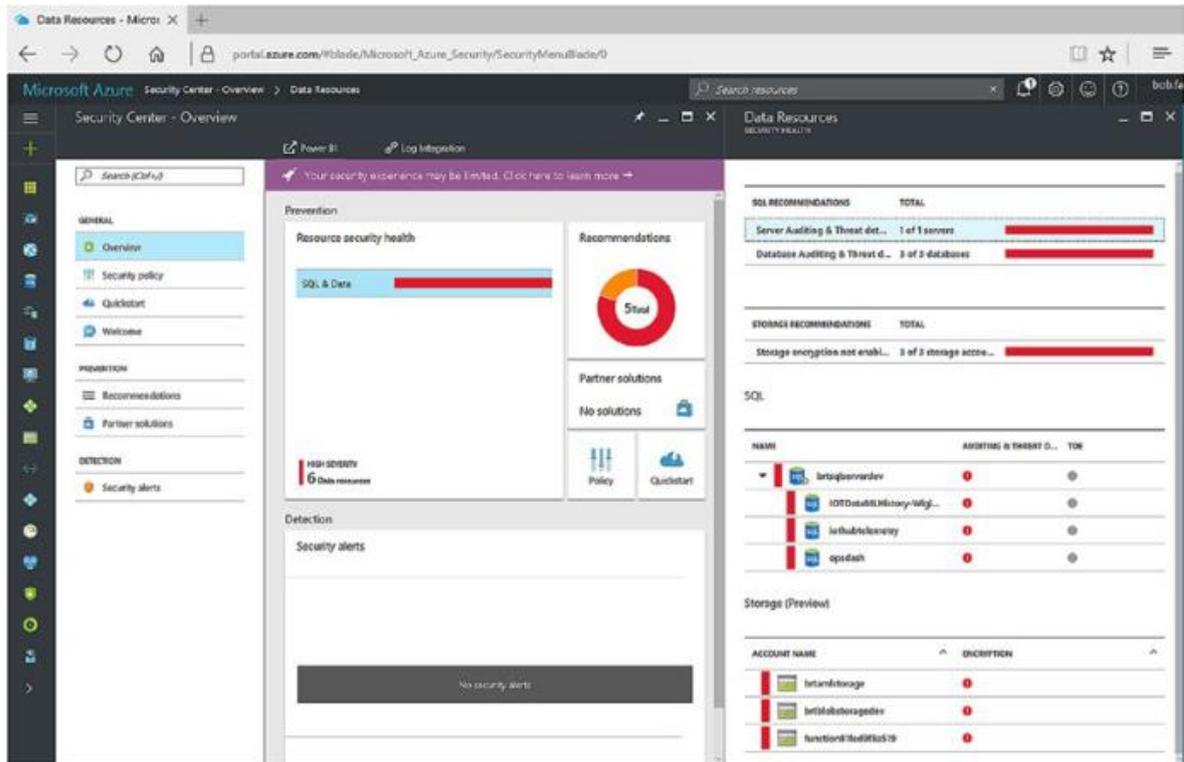
10.2 PROTOKOL KEAMANAN

Interaksi antara infrastruktur dan komponen Azure yang terdiri dari solusi IoT dapat diamankan menggunakan protokol standar industri termasuk sertifikat X.509, kontrol akses berbasis peran, aturan firewall, dan jaringan virtual.

Data yang berasal dari perangkat IoT lokal diamankan tanpa syarat dalam penerbangan melalui TLS melalui protokol seperti HTTP/S pada port 443 dan AMQP aman pada port 5671. Data yang tidak digunakan juga dapat diamankan melalui enkripsi berbasis layanan (seperti Enkripsi Data Transparan di Azure SQL Database dan Azure Storage Service Encryption) atau enkripsi tingkat aplikasi dengan Azure Key Vault yang menyediakan penyimpanan kunci akses yang aman. Persyaratan khusus untuk keamanan saat tidak digunakan akan bervariasi sesuai dengan sifat data dan mekanisme penyimpanan, yang ditangani pada lapisan arsitektur yang lebih tinggi.

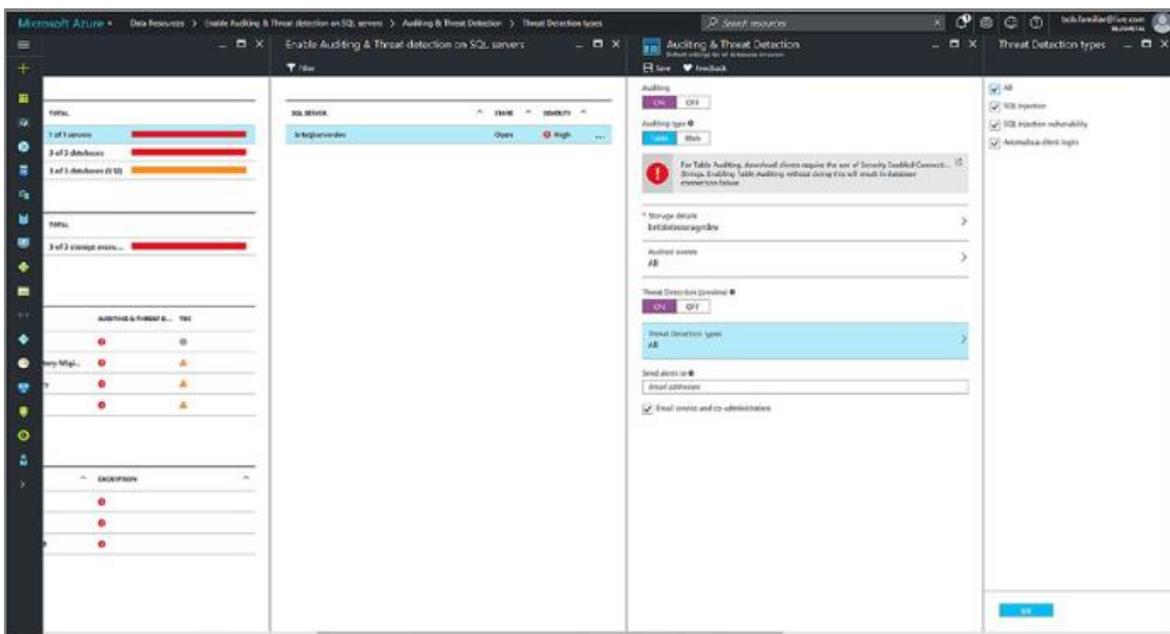
Pusat Keamanan Azure

Pusat Keamanan Azure memberi Anda gambaran umum tentang layanan Anda saat ini sehubungan dengan potensi ancaman dan memverifikasi bahwa kontrol keamanan yang sesuai sudah diterapkan dan dikonfigurasi dengan benar. Anda dapat mengatur peringatan dan pemberitahuan untuk setiap layanan Anda sehingga administrator dan anggota tim utama diperingatkan akan potensi ancaman (lihat Gambar 10.4).



Gambar 10.4. Pusat Keamanan melaporkan layanan yang rentan

Pusat Keamanan memberikan panduan remediasi dan kemampuan untuk mengaktifkan enkripsi, mengaktifkan audit, dan mengonfigurasi peringatan serta menyiapkan pemodelan ancaman jika diperlukan. Pada Gambar 10.5, Anda dapat melihat bahwa pemodelan ancaman pada Database SQL digunakan untuk mendeteksi injeksi SQL dan login klien yang anomali.



Gambar 10.5. Konfigurasi pemodelan ancaman

Enkripsi data

Semua layanan yang kami gunakan melalui solusi ini menyediakan enkripsi saat data sedang dikirim atau, seperti yang sering dijelaskan, dalam penerbangan. Seperti yang Anda lihat dari laporan Pusat Keamanan pada Gambar 10.5, enkripsi untuk data tidak aktif tidak diaktifkan secara default. Anda disarankan untuk mengaktifkan enkripsi untuk penyimpanan blob dan Database SQL. Anda dapat melakukan ini melalui Pusat Keamanan. Data lake menyediakan enkripsi untuk data tidak aktif secara default.

Manajemen Kunci

Azure Key Vault melindungi kunci dan rahasia kriptografi. Anda dapat mengenkripsi kunci dan rahasia seperti kunci autentikasi, kunci akun penyimpanan, kunci enkripsi data, file .PFX, dan kata sandi menggunakan kunci yang dilindungi oleh Modul Keamanan Perangkat Keras (HSM). Anda dapat membuat proses yang disederhanakan menggunakan brankas kunci sehingga Anda dapat mempertahankan kontrol atas kunci yang digunakan untuk mengakses dan mengenkripsi data Anda. Pengembang dapat membuat kunci untuk pengembangan dan pengujian menit lalu memigrasikannya dengan lancar ke kunci produksi. Administrator keamanan dapat memberikan dan mencabut izin pada kunci sesuai kebutuhan. Jika Anda memiliki langganan Azure, Anda dapat membuat dan menggunakan Key Vault. Dalam suatu organisasi, masuk akal untuk menyiapkan proses yang lebih formal di mana administrator mengawasi pembuatan, distribusi, dan pencabutan kunci. Orang atau beberapa orang ini akan mengizinkan pengguna dan aplikasi untuk mengakses brankas kunci secara terprogram untuk subkumpulan kunci sambil membiarkan pembuatan, impor, penghapusan, pencabutan, dan pemantauan menjadi masalah pribadi.

10.3 IDENTITAS

Identitas adalah tentang siapa yang dapat mengakses aplikasi Anda, API, dan data mendasar yang merupakan inti dari solusi IoT Anda. Anda akan menginginkan kemampuan untuk memberikan layanan mandiri pada tingkat tertentu kepada pengguna untuk pendaftaran, manajemen kata sandi, dan pembaruan profil dengan tetap menjaga protokol ketat untuk mengakses kemampuan aplikasi dan data.

Otentikasi dan Otorisasi

Otentikasi dan otorisasi untuk pelanggan, karyawan, dan mitra disediakan melalui penyedia identitas sebagai layanan (IaaS) yang dihosting di cloud. Azure Active Directory dengan cepat menjadi pilihan populer di ekosistem Azure untuk mengelola identitas multi-penyewa. Azure Active Directory (Azure AD) menyederhanakan autentikasi bagi pengembang dengan dukungan untuk protokol standar industri seperti OAuth 2.0 dan OpenID Connect serta pustaka sumber terbuka untuk berbagai platform untuk mengakses API aplikasi dan operasinya.

Azure AD B2C adalah layanan manajemen identitas cloud berfitur lengkap untuk aplikasi web dan seluler yang digunakan oleh konsumen dan mitra. Ini adalah layanan global dengan ketersediaan tinggi yang menjangkau ratusan juta identitas. Pengguna Anda dapat mendaftar, mengotentikasi, dan mengelola profil mereka menggunakan email perusahaan

atau salah satu akun sosial mereka seperti Google+, Facebook, atau LinkedIn. Formulir layanan mandiri disediakan oleh Azure AD B2C, dan aplikasi yang ingin diintegrasikan dengan layanan ini didaftarkan dengan direktori melalui Portal Microsoft Azure atau melalui API otomatisasi Azure Active Directory yang disebut GraphAPI.

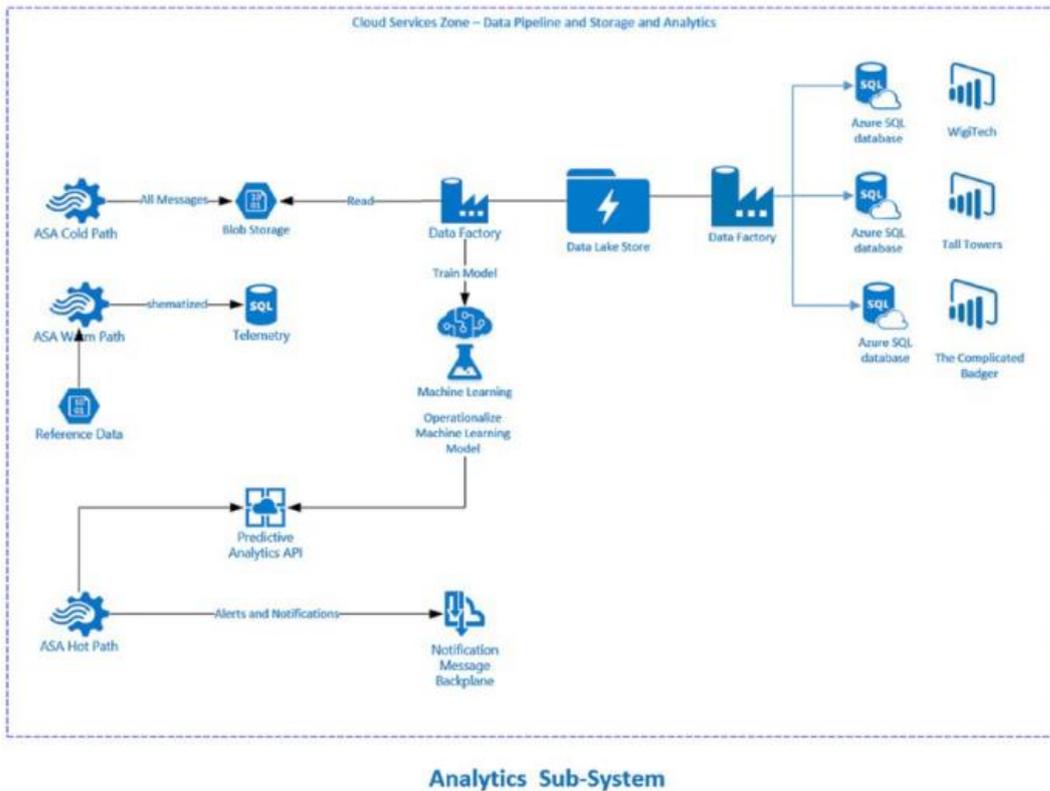
Multi-Sewa

Multi-tenancy mengacu pada kemampuan penerapan kode tunggal, sering kali berupa aplikasi web atau API, untuk melayani basis pengguna dari beberapa perusahaan dan mitra (penyewa) sekaligus memberikan potensi pengalaman berbeda untuk setiap penyewa. Setelah masuk, pengguna hanya akan dapat melihat area fungsional aplikasi yang boleh mereka lihat dan mengakses data yang boleh mereka lihat. Sehubungan dengan akses data, ada dua pola desain yang dapat diterapkan:

- *^{*} Layanan saluran data membuat database individual dan terisolasi untuk setiap organisasi dan otorisasi berbasis peran diterapkan untuk karyawan di perusahaan tersebut.
- *^{*} Layanan saluran data menempatkan semua data ke dalam satu penyimpanan elastis yang skemanya menyediakan lapisan keamanan yang menggunakan klaim autentikasi untuk membatasi akses berdasarkan perusahaan dan peran.

Tidak ada cara yang benar atau salah dalam melakukan hal ini; itu adalah keputusan bisnis mengenai mana yang sesuai untuk Anda. Anda mungkin memiliki pelanggan yang ingin diyakinkan bahwa data mereka diisolasi dari organisasi lain yang menggunakan aplikasi SaaS yang sama. Dalam hal ini, konsekuensinya adalah tim Anda perlu merancang proses pembuatan dan pengelolaan database untuk setiap organisasi. Jika semua organisasi mengakses penyimpanan elastis dan terukur yang sama, maka tugas Anda adalah tidak hanya mengelola penyimpanan ini namun juga memberikan jaminan bahwa akses data dibatasi dengan benar di seluruh organisasi dan pengguna dalam organisasi.

Solusi referensi menunjukkan kedua pola ini. Bab 7 dan 9 memperkenalkan pola pertama: bagaimana Anda bisa merancang proses saluran data menggunakan Data Factory untuk menghasilkan database individual untuk setiap organisasi (lihat Gambar 10.6).



Gambar 10.6. Penyimpanan data terpisah untuk setiap organisasi

Di sini kita akan memeriksa pola kedua: penyimpanan tunggal yang dapat diskalakan yang mendefinisikan skema untuk menyediakan akses berbasis peran dan perusahaan. Selain Blob Storage dan Data Lake Store yang digambarkan pada Gambar 10.6, pekerjaan Stream Analytics mengirimkan telemetry ke tabel Database SQL yang disebut sebagai jalur hangat. Saat pekerjaan Stream Analytics memproses setiap pesan yang masuk, pekerjaan tersebut mengubah pesan dengan menggabungkan data biometrik dengan detail tentang karyawan yang mengenakan rompi seperti namanya, email, usia, berat badan, dan tinggi badan beserta perusahaannya. nama dan jenis pengguna. Dalam skenario kami, ada dua jenis pengguna: administrator dan karyawan. Seorang karyawan hanya dapat melihat data pribadinya sendiri. Administrator hanya dapat melihat data karyawan di perusahaannya. Dengan menggunakan kolom tipe perusahaan dan pengguna dalam tabel Database SQL, kami akan mendemonstrasikan pola kedua: menyediakan overlay keamanan berdasarkan tipe pengguna dan organisasi untuk membatasi akses data (lihat Gambar 10.7).

SQLQuery1.sql - tcp.sqlserverdev (113) - x

```

/***** Script for SelectTopNRows command from SSIS *****/
SELECT TOP 1000 [UserId]
,[Age],[Height],[Weight],[companyname],[firstame],[lastname],[username],[type]
,[HeartRateBPM],[BreathingRate],[Temperature],[Steps],[Velocity],[Altitude]
,[Ventilation],[Activity],[Cadence],[Speed],[HRB],[HeartRateRedZone]
,[HeartRateVariability],[Status],[Id],[DeviceId],[MessageType],[Longitude],[Latitude],[Timestamp],[imageURL],[phone],[email],[gender],[race]
FROM [dbo].[TchuoSensorReadings]

```

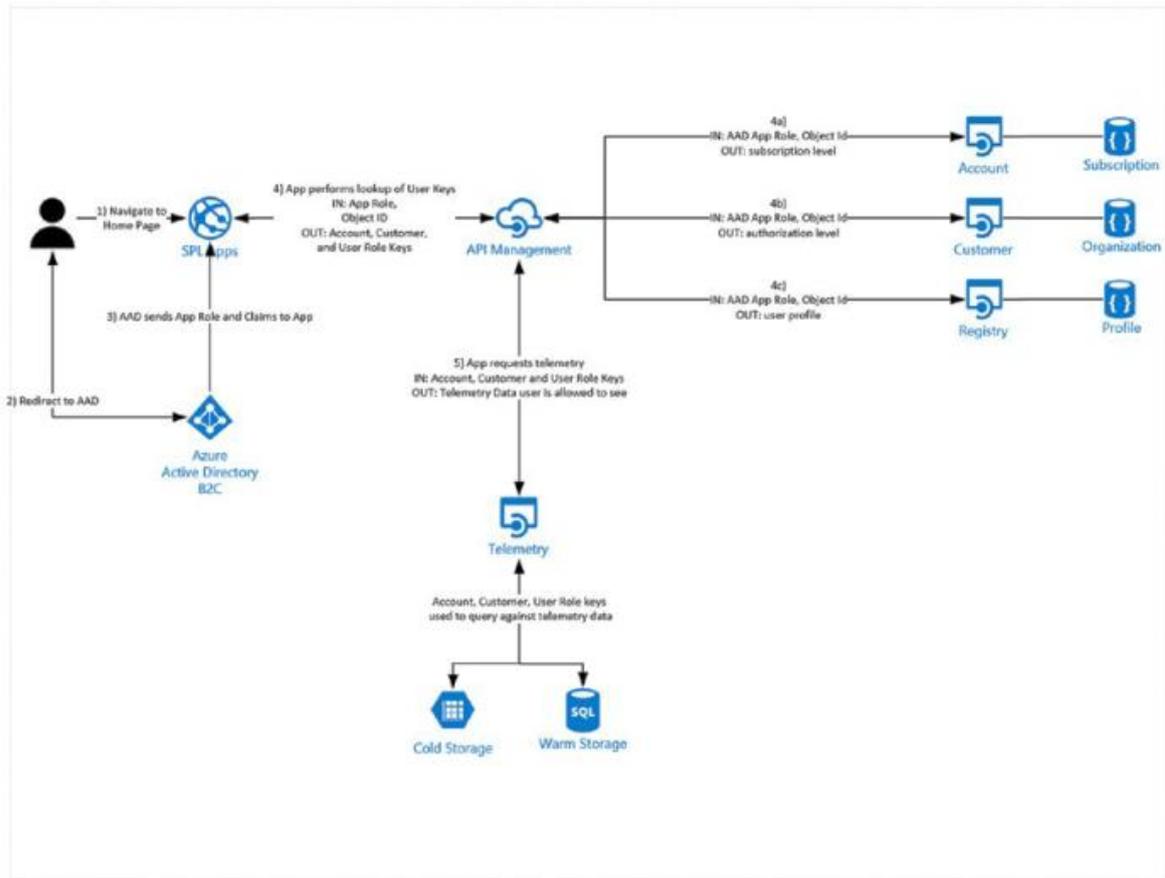
Userid	Age	Height	Weight	companyname	firstame	lastname	username	type	HeartRateBPM	BreathingRate	Temperature	Steps	Velocity	Altitude	Ventilation	Act
3a50e5c6bc7c-8c3b125-d8e23e23d22	0	0	0	The Complicated Badger	Jyngnen	Nergaua	Jyngnen Nergaua	3	90	17.44329379	96.1526437001576	0	3.35241379250156	0	16454.15202	0.1
c2df3320adce-4644-e534-e98df1ce18	0	0	0	The Complicated Badger	Caell	Anur	Caell Anur	3	91	17.51138318	96.2737178547417	0	7.2690688633901	0	16547.12564	0.1
470c695-dc4e-485f-9e4f-7125020c1	0	0	0	The Complicated Badger	Jyssa	Anuk	Jyssa Anuk	2	91	17.67347257	96.7012042983906	0	8.6965905894405	0	16640.03946	0.1
470c695-dc4e-485f-9e4f-7125020c1	72	70	270	The Complicated Badger	Jyssa	Anuk	Jyssa Anuk	2	51	7	96.6020930533364	0	0	0	2000	0
8bfef65-2302-469f-af11-fa253c75e9b	72	70	270	Tall Towers	Fid	Sidy	Fid.Sidy	2	51	7	96.6020930533364	0	0	0	2000	0
058f7760-e188-434e-aa1016c-13665eb66	72	70	270	WigTech	Togrik	Denyc	Togrik.Denyc	2	51	7	96.6020930533364	0	0	0	2000	0
c2df3320adce-4644-e534-e98df1ce18	47	75	220	The Complicated Badger	Caell	Anur	Caell Anur	3	93	7	96.1258098248325	0	0	0	2000	0
199e68b-35a4-4b6a-9b34-3ab2bd9f102b	27	63	100	Tall Towers	Canver	Ihctk	Canver.Ihctk	3	105	19.2498861	96.6379467276009	0	83.4848571307439	0	19150.3646	0.2
42105a24-607f-4f5e-eca3-3301aa14eb75	28	71	190	WigTech	Aog	Dean	Aog Dean	3	99	18.56950281	96.75711120856	0	55.6302877299839	0	18127.66358	0.1
42105a24-607f-4f5e-eca3-3301aa14eb75	28	71	190	WigTech	Aog	Dean	Aog Dean	3	91	17.51138318	96.2454475067768	0	6.80721252263795	0	16547.12564	0.1
c2df3320adce-4644-e534-e98df1ce18	47	75	220	The Complicated Badger	Caell	Anur	Caell Anur	3	90	17.3872044	96.5029058603118	0	0.231835423170355	0	16361.18	0.1
fc540fe-136a-4733-b87d-d5d20eeae310	0	0	0	The Complicated Badger	Jlonna	Animesh	Jlonna Animesh	3	90	17.3872044	96.0501257921803	0	0	0	16361.18	0.1
fc540fe-136a-4733-b87d-d5d20eeae310	57	73	235	The Complicated Badger	Jlonna	Animesh	Jlonna Animesh	3	96	18.19436647	96.9310698870994	0	37.6462526463955	0	17569.82666	0.1
a7591ba2-e534-48f4-f1a-50e56c0e396a	47	75	220	Tall Towers	Bross	Glah	Bross.Glah	3	96	18.19436647	96.9310698870994	0	37.6462526463955	0	17569.82666	0.1
fa76cd9-72a0-40e0-82cb-5482a13b860	34	72	195	Tall Towers	Cleyan	Guga	Cleyan.Guga	3	92	17.63556196	96.3916952532677	0	12.3758166360503	0	16733.07128	0.1
c2df3320adce-4644-e534-e98df1ce18	47	75	220	The Complicated Badger	Caell	Anur	Caell Anur	3	111	20.05704817	96.5979114331295	0	116.519145466531	0	20359.01126	0.2
199e68b-35a4-4b6a-9b34-3ab2bd9f102b	27	63	100	Tall Towers	Canver	Ihctk	Canver.Ihctk	3	117	20.80212085	96.1251467406401	0	151.160195679522	0	21474.6851	0.3
fa76cd9-72a0-40e0-82cb-5482a13b860	34	72	195	Tall Towers	Cleyan	Guga	Cleyan.Guga	3	101	18.75317098	96.2971493831367	0	60.2295612020569	0	18406.58204	0.2
fc540fe-136a-4733-b87d-d5d20eeae310	57	73	235	The Complicated Badger	Jlonna	Animesh	Jlonna Animesh	3	92	17.63556196	96.6754882918147	0	12.9441371256477	0	16733.07128	0.1
fc540fe-136a-4733-b87d-d5d20eeae310	57	73	235	The Complicated Badger	Jlonna	Animesh	Jlonna Animesh	3	96	18.13227708	96.0082225161643	0	35.8868796974008	0	17476.85364	0.1
c2df3320adce-4644-e534-e98df1ce18	47	75	220	The Complicated Badger	Caell	Anur	Caell Anur	3	113	20.24331634	96.4701525850548	0	124.480064267232	0	20637.92972	0.2
a6569261-b13a-42f9-935b-81ad6f12e25	68	65	186	Tall Towers	Tanjary	Smry	Tanjary.Smry	3	92	17.69765135	96.3370570992069	0	14.4322481511568	0	16826.0441	0.1
199e68b-35a4-4b6a-9b34-3ab2bd9f102b	27	63	100	Tall Towers	Canver	Ihctk	Canver.Ihctk	3	111	19.99495878	96.1105233976201	0	115.364888647862	0	20266.03844	0.2
a7591ba2-e534-48f4-f1a-50e56c0e396a	47	75	220	Tall Towers	Bross	Glah	Bross.Glah	3	102	18.87734976	96.629551836117	0	67.0840979852776	0	18592.52768	0.2

Query executed successfully. tcp:sqlserverdev.databas... btdadmindev@btsqlserv... iethubtelemetry 00:00:00 1000 rows

Gambar 10.7. Tabel SQL yang berisi telemetri pengguna

Agar rencana ini berhasil, kami memerlukan API aman yang berada di atas tabel Database SQL dan menerapkan persyaratan model keamanan kami. Telemetri API akan menyediakan titik akhir yang aman untuk aplikasi yang ingin mengakses data pokok. Aplikasi apa pun yang ingin memanggil API Telemetri harus memberikan detail pengguna yang diautentikasi, nama perusahaan, dan jenis pengguna sebagai parameter panggilan tersebut. Aplikasi ini perlu didaftarkan pada Active Directory untuk membangun hubungan kepercayaan antara penyedia identitas (Active Directory) dan aplikasi itu sendiri. Setelah pengguna diautentikasi, aplikasi memiliki akses ke “klaim” untuk pengguna tersebut dan dapat menggunakan informasi tersebut untuk memanggil API platform. Klaim adalah pasangan nama-nilai yang memberikan detail tentang pengguna seperti nama depan dan belakang mereka. Selain itu, Anda dapat memperluas kumpulan klaim default dengan menambahkan properti tambahan yang diperlukan di tingkat aplikasi.

Dengan menggunakan klaim ID pengguna, misalnya, aplikasi dapat mencari profil pengguna melalui API Registri, mengambil tingkat langganan pengguna dari API Akun, detail organisasi dari API Pelanggan, dan detail konfigurasi aplikasi dari API Aplikasi (lihat Gambar 10.8).



Gambar 10.8. Otentikasi dan otorisasi multi-penyewa

Gambar 10.8 menggambarkan alur kerja otentikasi, otorisasi, dan permintaan telemetri pengguna:

1. Pengguna menavigasi ke halaman beranda aplikasi web atau meluncurkan aplikasi seluler.
2. Pengguna dialihkan ke halaman masuk Azure Active Directory dan memberikan nama pengguna dan kata sandinya. Dalam kasus aplikasi seluler, halaman masuk Azure Active Directory disajikan dalam popup modal.
3. Azure AD mengautentikasi pengguna dan, jika berhasil, mengalihkan kembali ke aplikasi dengan token yang berisi serangkaian klaim standar seperti nama, ID unik, dll.
4. Aplikasi menggunakan klaim ini untuk memanggil sekumpulan API terkelola yang diamankan dari aplikasi ke proksi Manajemen API menggunakan token pembawa OAuth dan dari proksi ke layanan menggunakan sertifikat terkelola, autentikasi dasar, atau rahasia bersama.
 - ❖ Account API mengembalikan model berlangganan. Model berlangganan mungkin menentukan tingkat kemampuan fitur seperti emas, perak, dan perunggu, yang dipetakan ke model monetisasi.
 - ❖ API Pelanggan mengembalikan model organisasi yang terdiri dari peran yang dimiliki pengguna dalam organisasi pelanggan, misalnya Admin Super, Admin, Layanan Teknis, Hanya Baca, Manajer Distrik, Manajer Cabang, dan seterusnya.

- ❖ Registry API mengembalikan model profil pengguna yang berisi semua detail tentang pengguna termasuk lokasi, informasi kontak, dan preferensi.
5. Pengguna meminta untuk melihat telemetri.
- ❖ Aplikasi memanggil API Telemetri terkelola aman yang meneruskan kunci yang relevan untuk profil pengguna, peran pengguna, akun, dan pelanggan. Kunci ini digunakan dalam klausa WHERE bersama dengan parameter lain dalam kueri yang mendasarinya untuk membatasi data yang dikembalikan untuk pengguna tersebut.

Sisa bab ini akan membawa Anda melalui serangkaian latihan yang mengonfigurasi penyewa Azure AD B2C di langganan Anda, mengimpor pengguna dari tiga organisasi, mendaftarkan aplikasi, dan memanfaatkan klaim pengguna yang dikembalikan dari Azure AD B2C untuk melakukan panggilan ke API Telemetri, yang membatasi akses ke data berdasarkan peran pengguna.

10.4 PELATIHAN MEMBUAT PENYEWAWA AZURE B2C

Untuk mengimplementasikan aplikasi yang memanfaatkan Azure AD B2C untuk identitas, Anda harus memiliki instans Azure AD B2C terlebih dahulu, lalu mendaftarkan aplikasi Anda ke instans tersebut. Proses pendaftaran aplikasi membuat ID unik untuk aplikasi Anda, yang dikenal sebagai ID Aplikasi. Proses registrasi juga akan meminta Redirect URL yang digunakan untuk mengarahkan kembali ke aplikasi Anda setelah otentikasi selesai.

Azure Active Directory mendukung beberapa protokol autentikasi. Skenario ini memanfaatkan OpenID Connect (yang bergantung pada OAuth 2.0), sehingga aplikasi berkomunikasi dengan Microsoft Azure Active Directory menggunakan dua titik akhir berikut:

- <https://login.microsoftonline.com/{tenant}/oauth2/v2.0/authorize>
- <https://login.microsoftonline.com/{tenant}/oauth2/v2.0/token>

dengan {tenant} adalah nama penyewa Azure AD B2C Anda; misalnya, mytenant.onmicrosoft.com (atau GUID yang terkait dengan penyewa tersebut).

Untuk menyelesaikan proses pendaftaran aplikasi, Anda mengonfigurasi serangkaian kebijakan, masing-masing kebijakan untuk operasi yang ingin dilakukan Azure AD B2C atas nama Anda. Kebijakan meliputi:

- Mendaftar
- Masuk
- Memperbaharui profil
- Reset Kata Sandi

Ikuti langkah-langkah berikut untuk membuat dan mengonfigurasi instans Azure Active Directory Anda.

1. Navigasikan ke portal Manajemen Azure Klasik dan masuk. (<http://manage.windowsazure.com>)
2. Pilih New ► Directory ► Custom Create (lihat Gambar 10-9).

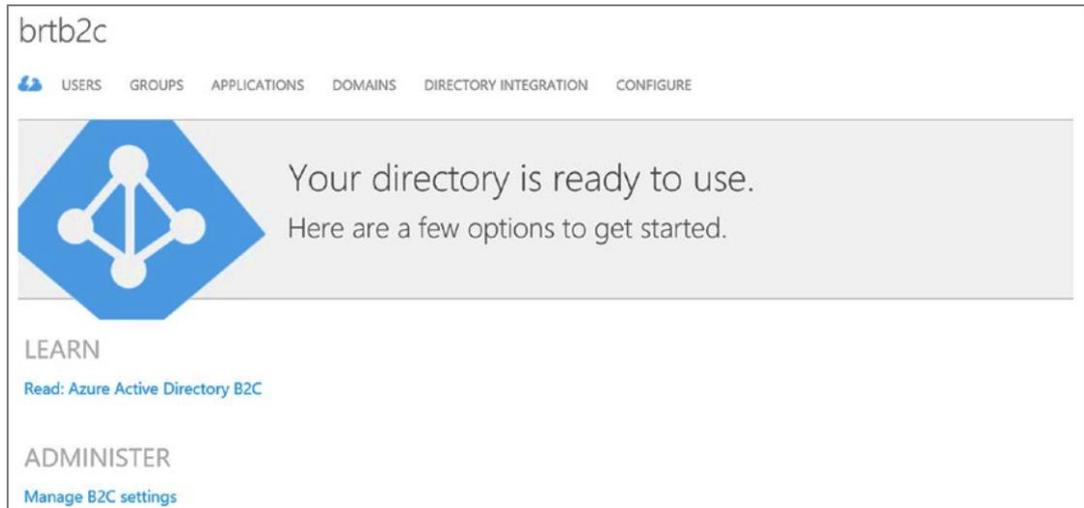


Gambar 10.9. Buat instans Azure Active Directory B2C

3. Isi formulir yang berisi nama organisasi Anda, nama unik penyewa Anda, dan wilayah. Terakhir pastikan untuk mencentang kotak berlabel “Ini adalah direktori B2C” (lihat Gambar 10.10).

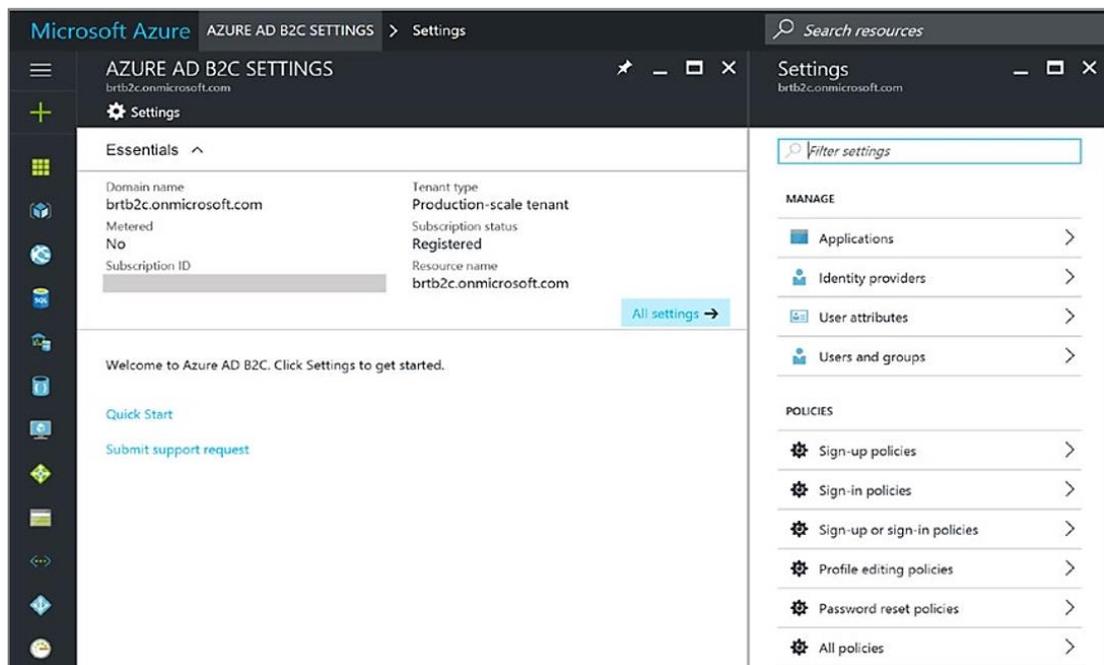
Gambar 10.10. Beri nama dan buat penyewa B2C Anda

4. Setelah proses pembuatan selesai, Anda akan dapat mengkonfigurasi direktori (lihat Gambar 10.11).



Gambar 10.11. Siap untuk mengelola Pengaturan B2C

5. Klik Kelola Pengaturan B2C. Ini akan membawa Anda ke portal Microsoft Azure baru (lihat Gambar 10.12).



Gambar 10.12 Bilah Pengaturan Azure AD B2C

Selanjutnya, Anda akan mengaitkan penyewa ke langganan Anda.

Hubungkan Penyewa Anda Dengan Berlangganan Anda

Untuk mengaitkan penyewa Direktori Aktif dengan langganan Anda, Anda harus menggunakan akun Microsoft. ID Organisasi tidak akan dapat menjalankan fungsi ini.

Apa bedanya? ID Organisasi adalah ID akun Azure yang hanya bisa masuk ke direktori miliknya. Email Office 365 Anda adalah contoh ID organisasi. Akun Microsoft, atau yang biasa

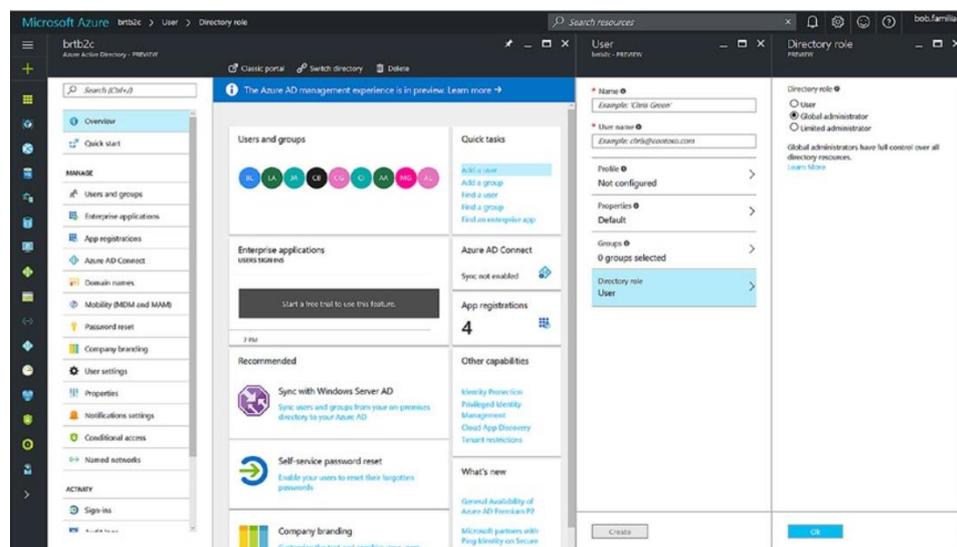
disebut Live ID, adalah akun yang Anda gunakan untuk masuk ke layanan seperti Skype, Outlook.com, OneDrive, Windows Phone, dan Xbox LIVE.

1. Jika perlu, buat ID Microsoft dan jadikan pengguna tersebut sebagai administrator bersama langganan Azure Anda.

Anda dapat melakukannya dari portal klasik di <https://manage.windowsazure.com>. Klik pengaturan di bilah navigasi sebelah kiri, klik nama langganan Anda, lalu pilih Administrator dari menu atas. Kemudian klik tombol Tambah di toolbar bawah (lihat Gambar 10-13).

Gambar 10.13. Tentukan co-administrator untuk langganan

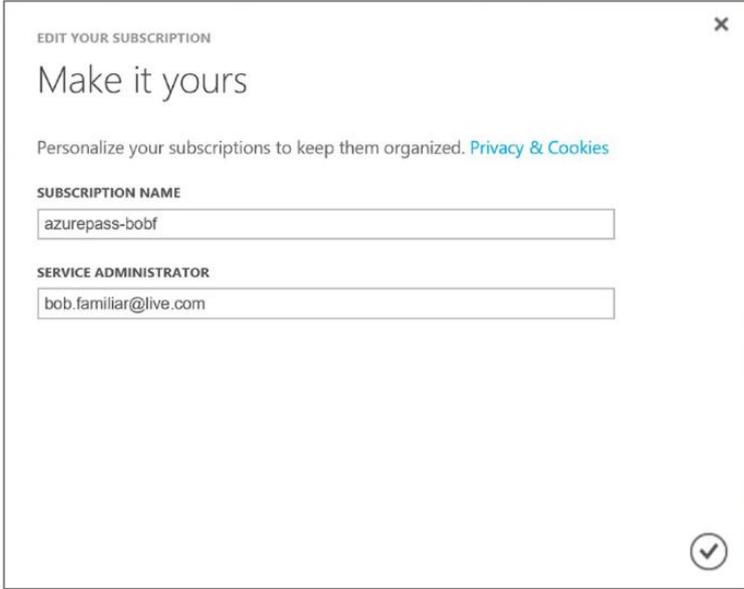
2. Tambahkan akun Microsoft ke Direktori Azure AD B2C sebagai Administrator Global (lihat Gambar 10-14).



Gambar 10.14. Tambahkan administrator global

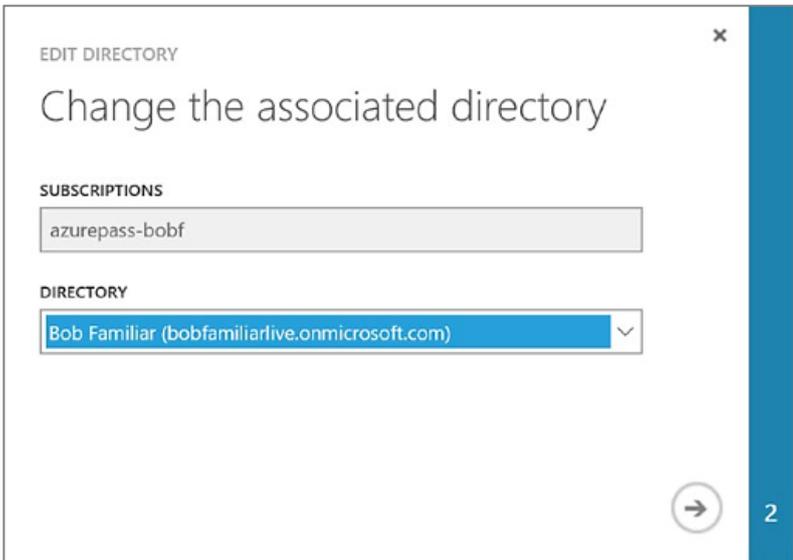
- Jadikan pengguna ini sebagai administrator layanan langganan (lihat Gambar 10.15).

Masuk ke <https://account.windowsazure.com> menggunakan ID yang Anda gunakan untuk membuat langganan Azure, lalu pilih langganan Anda untuk menelusuri halaman detail. Klik Edit Detail Langganan dan atur ID Microsoft sebagai administrator layanan.



Gambar 10.15. Mengubah administrator layanan

- Keluar dan navigasikan ke portal klasik. Masuk menggunakan akun Microsoft. Klik Pengaturan di bilah navigasi kiri, pilih langganan dalam daftar, lalu klik tombol Edit di bilah alat bawah. Pilih direktori di daftar drop-down, klik ke Langkah 2 dan simpan perubahan Anda (lihat Gambar 10.16).

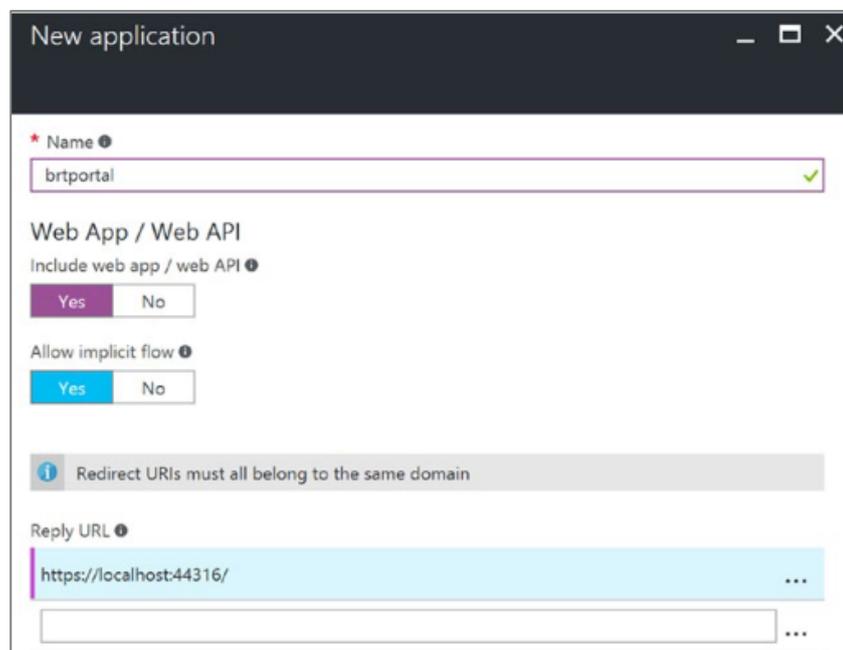


Gambar 10.16. Ubah direktori terkait

DAFTAR APLIKASI

Hanya aplikasi terdaftar yang dapat berintegrasi dengan Azure AD B2C. Dalam latihan ini, Anda akan mendaftarkan aplikasi web yang menyediakan akses ke Telemetri dan menentukan kebijakan untuk pendaftaran, proses masuk, dan manajemen profil. Perhatikan bahwa kami akan memperbarui aplikasi ini di latihan selanjutnya. Untuk saat ini, kami hanya akan mendaftarkannya ke Azure AD:

1. Dari Blade Manajemen Azure AD B2C, klik Aplikasi, lalu Tambahkan (lihat Gambar 10-17).
2. Masukkan brtportal untuk namanya
3. Klik Ya untuk Menyertakan Aplikasi Web/API Web.
4. Masukkan `https://localhost:44316/` sebagai URL pengalihan.
5. Klik tombol Buat.



Gambar 10.17 Mendaftarkan aplikasi

6. Setelah proses pembuatan selesai, klik nama aplikasi untuk menampilkan ID aplikasi yang dihasilkan.
7. Salin ID aplikasi untuk digunakan nanti.

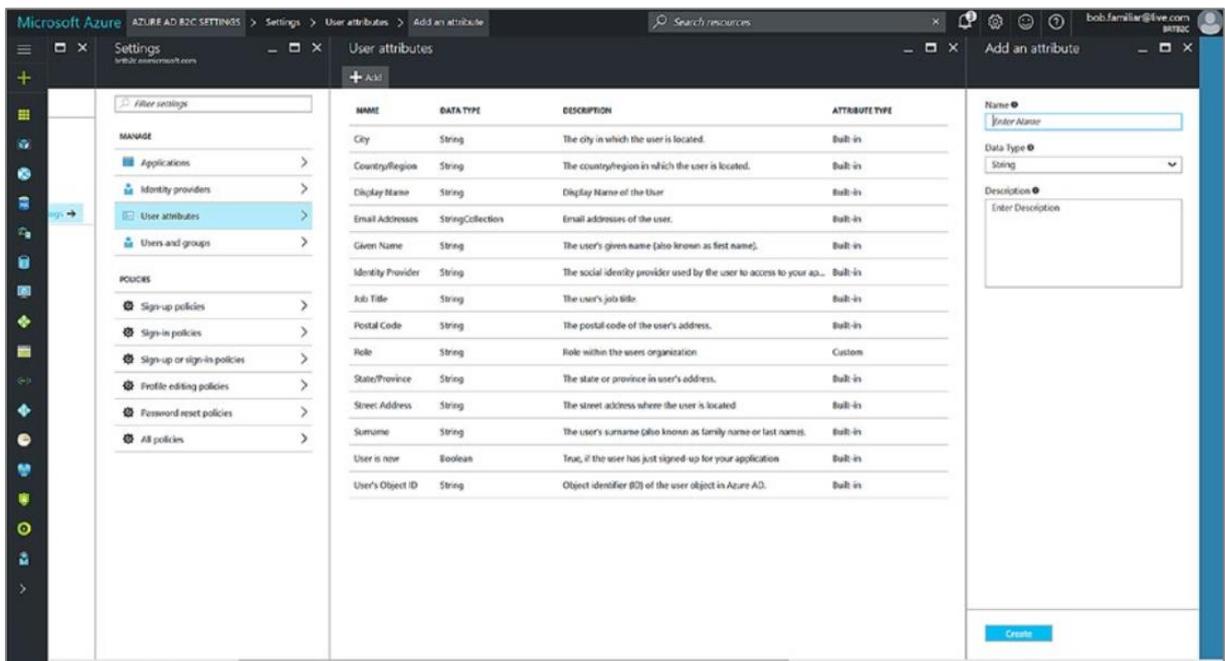
Pada langkah berikutnya, kami menentukan serangkaian kebijakan untuk aplikasi terdaftar kami. Sebagai bagian dari definisi ini, kita dapat menentukan atribut pengguna yang ingin kita minta saat mendaftar serta atribut mana yang akan dikembalikan sebagai bagian dari klaim setelah autentikasi. Anda dapat mengakses atribut pengguna dari Blade Manajemen Azure AD B2C (lihat Gambar 10-18). Dimungkinkan juga untuk menentukan atribut pengguna khusus.

8. Klik Tambah pada layar Atribut Pengguna dan buat atribut khusus yang disebut Peran dengan tipe data string.

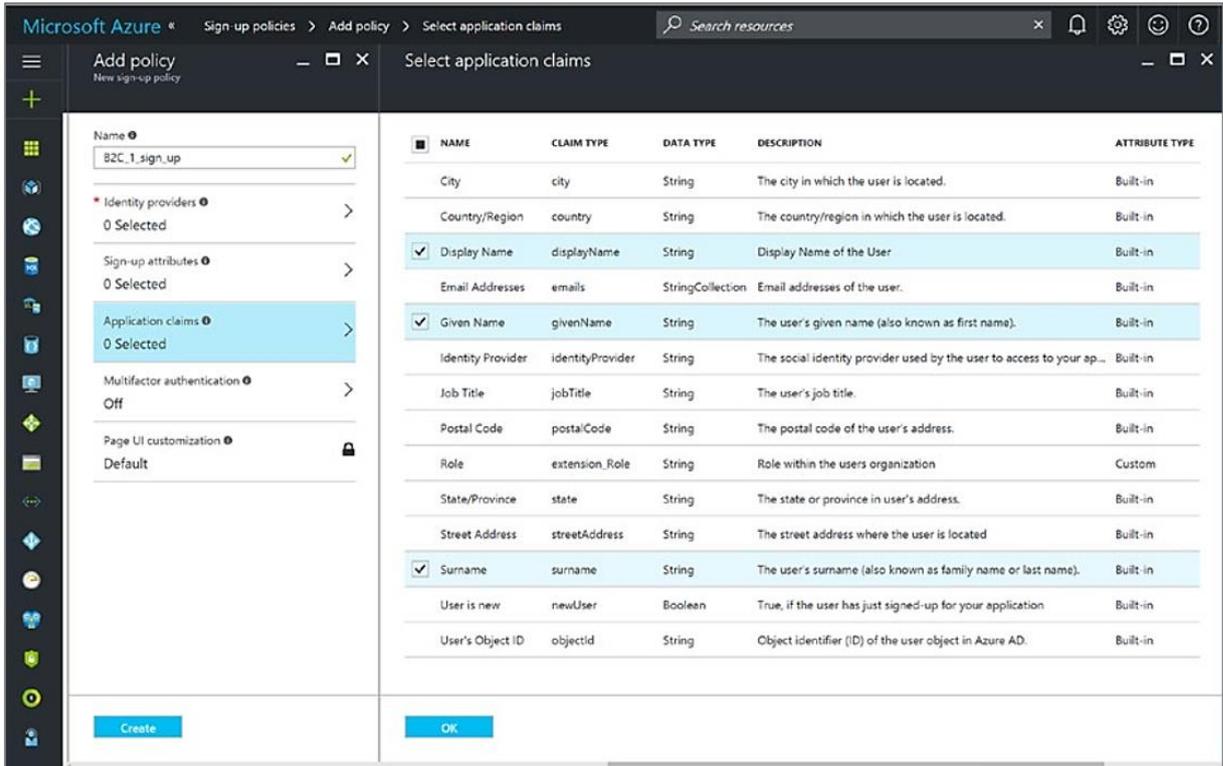
Sekarang setelah aplikasi terdaftar, kita dapat menentukan kebijakan untuk aplikasi ini. Kebijakan menentukan titik integrasi antara Azure AD B2C dan aplikasi. Azure AD B2C akan menyediakan layar untuk pendaftaran, masuk, manajemen profil, dan pengaturan ulang kata sandi.

9. Di bilah Manajemen Azure B2C, klik Kebijakan Pendaftaran.

Untuk Penyedia identitas, pilih Email. Atribut pendaftaran adalah bidang yang Anda ingin pengguna berikan pada formulir pendaftaran. Pilih Email dan Nama Tampilan. Untuk klaim, minimal pilih Nama Tampilan tetapi Anda dapat memilih atribut sebanyak yang ingin Anda kirimkan kembali setelah pendaftaran berhasil. Simpan hasil edit Anda dan klik Buat (lihat Gambar 10.19).

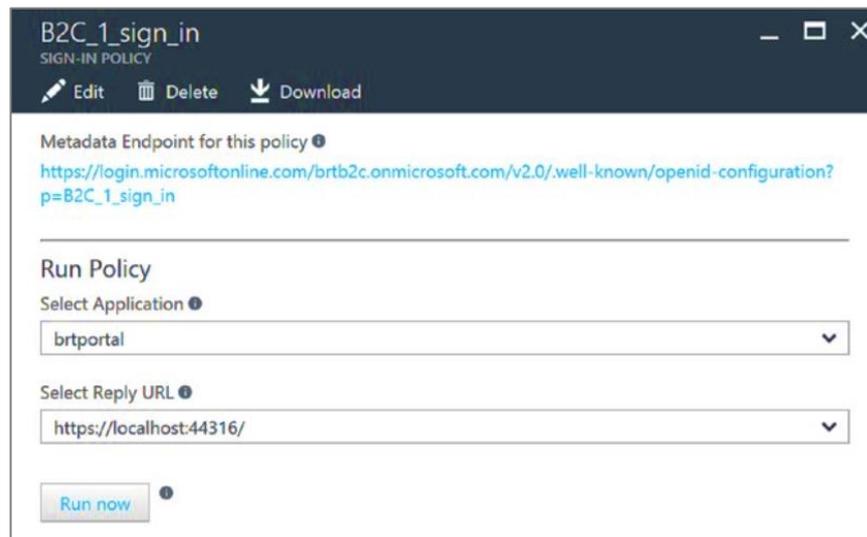


Gambar 10.18. Atribut pengguna



Gambar 10.19. Tambahkan kebijakan, pilih klaim

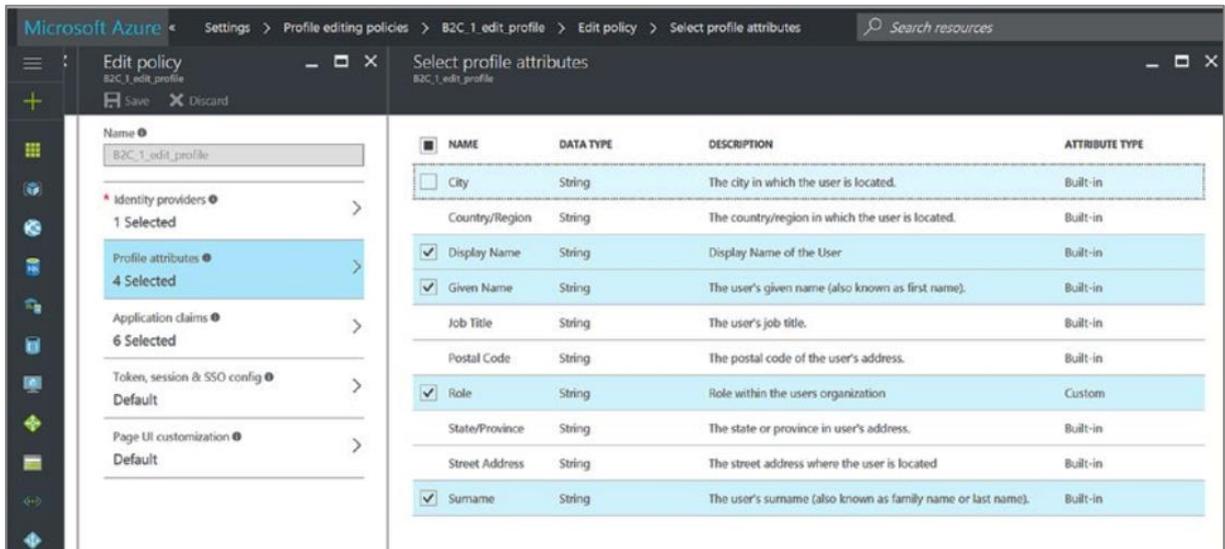
- Setelah kebijakan dibuat, klik namanya dan pilih aplikasi yang menerapkan kebijakan ini. Masukkan URL Balasan untuk aplikasi itu. Untuk contoh ini, nilainya adalah <https://localhost:4435644316/> (lihat Gambar 10-20).



Gambar 10.20 Kebijakan B2C_1_sign_in

- Ikuti prosedur yang sama untuk membuat kebijakan masuk.
- Buat kebijakan Edit Profil. Kebijakan ini akan memungkinkan pengguna memperbarui profilnya untuk memberikan nilai pada atribut khusus yang kami tetapkan. Pastikan

untuk memilih Peran sebagai salah satu atribut yang dapat diedit pengguna sebagai bagian dari profil mereka (lihat Gambar 10.21).



Gambar 10.21. Buat kebijakan Edit Profil

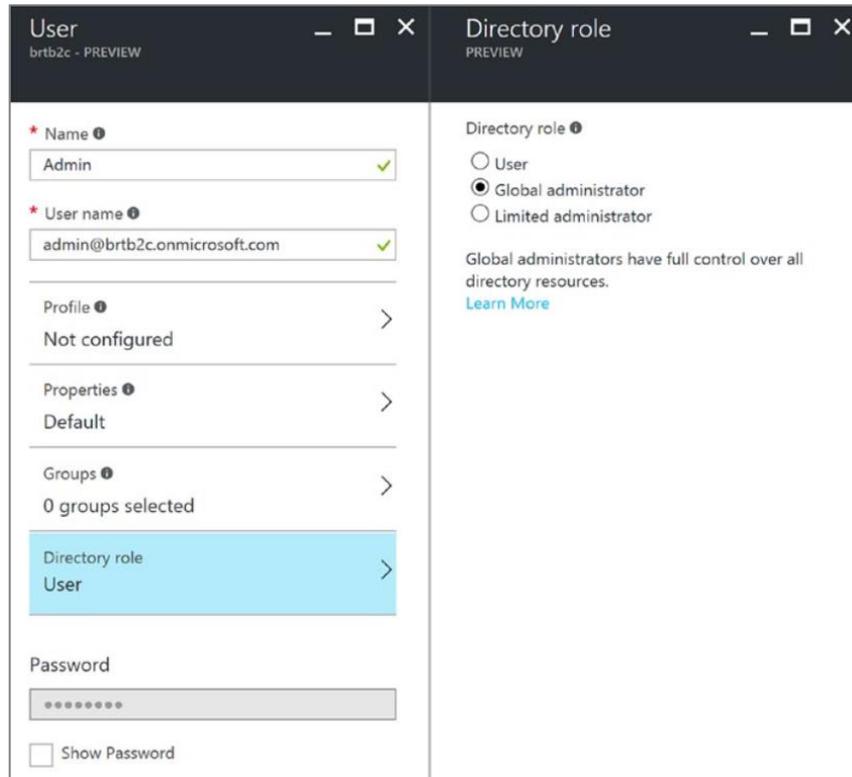
Impor Pengguna Ke Penyewa Baru

Dalam latihan ini, Anda mengimpor akun pengguna ke penyewa Azure AD B2C sehingga Anda bisa menguji fitur multi-penyewa. Microsoft menyediakan utilitas baris perintah yang mudah digunakan untuk melakukan operasi terhadap penyewa. API untuk Direktori Aktif disebut API Grafik, dan latihan ini akan memberikan wawasan tentang bagaimana Anda dapat mengotomatiskan operasi terhadap penyewa AD menggunakan API ini. Untuk memberikan akses aman ke Graph API, ada beberapa pengaturan sebelum Anda dapat melakukan operasi direktori berikut:

- Buat administrator global lokal di penyewa Anda.
- Instal Online Sign-In Assistant dan Modul Direktori Aktif untuk PowerShell.
- Daftarkan utilitas baris perintah dengan Azure AD B2C sehingga memiliki kredensial yang tepat untuk memanggil GraphAPI.

1. Di bilah manajemen, klik Pengguna dan Grup, Semua Pengguna, lalu klik Tambah.
2. Buat akun pengguna untuk admin lokal dengan nama `user.admin@[penyewa Anda].onmicrosoft.com`
3. Tetapkan kata sandi.
4. Tetapkan Peran Direktori menjadi Administrator Global.
5. Klik Buat.

Akun ini memiliki hak untuk memanggil GraphAPI terhadap penyewa Anda (lihat Gambar 10.22).



Gambar 10.22. Tambahkan akun administrator global lokal secara manual

6. Ikuti tautan berikut untuk mengunduh dan menginstal Asisten Masuk Online dan Modul Direktori Aktif untuk PowerShell:
<http://go.microsoft.com/fwlink/?LinkID=286152>
<http://go.microsoft.com/fwlink/p/?linkid=236297>
7. Setelah selesai, buka jendela perintah atau jendela konsol PowerShell, navigasikan ke folder identitas repo lokal Anda, dan kloning utilitas baris perintah menggunakan Git. git clone (<https://github.com/AzureADQuickStarts/B2C-GraphAPI-DotNet.git>)

Ada tiga skrip PowerShell di folder identitas yang akan membantu Anda melalui proses ini.

- * 01-RegisterAppInADTenant.ps1: Mendaftarkan utilitas baris perintah dengan Azure Active Directory dan menghasilkan ID Objek, ID Utama Aplikasi, dan Rahasia Klien.
- * 02- AssignAppPermissions.ps1: Memperluas izin utilitas baris perintah.
- * 03-ImportUsers.ps1: Menggunakan utilitas baris perintah untuk mengimpor pengguna. Pengguna telah ditentukan sebelumnya dalam sekumpulan file JSON di folder identitas\UserJson.

Utilitas baris perintah akan menggunakan rahasia klien untuk mengautentikasi ke Azure AD dan untuk memperoleh token akses. ID Objek dan ID Utama Aplikasi digunakan untuk menetapkan izin tambahan yang perlu dilakukan oleh utilitas, yaitu operasi CRUD berikut: pembaca direktori untuk membaca pengguna, penulis direktori untuk membuat dan memperbarui pengguna, dan administrator akun pengguna untuk menghapus pengguna.

8. Buka jendela konsol PowerShell dan navigasikan ke folder identitas repo. Jalankan skrip `01-RegisterAppInADTenant.ps1`. Skrip akan meminta nama langganan Azure Anda dan nama aplikasi yang ingin Anda daftarkan. Perhatikan nama aplikasinya adalah `B2CGraphClient` (lihat Gambar 10.23).

```

Administrator: Windows PowerShell

PS C:\users\bob\source\repos\brt\identity> ls

Directory: C:\users\bob\source\repos\brt\identity

Mode                LastWriteTime         Length Name
----                -
d-----            1/22/2017   9:04 PM          B2CGraphClient
d-----            1/22/2017   9:04 PM          UserJson
-a-----            1/22/2017   9:04 PM         830 01-RegisterAppInADTenant.ps1
-a-----            1/22/2017   9:04 PM         648 02-AssignAppPermissions.ps1
-a-----            1/22/2017   9:04 PM         454 03-ImportUsers.ps1
-a-----            1/22/2017   9:04 PM        1096 AppRegistration.txt
-a-----            1/22/2017   9:04 PM          94 ClientSecret.txt
-a-----            1/22/2017   9:04 PM         124 tenant.txt

PS C:\users\bob\source\repos\brt\identity> .\01-RegisterAppInADTenant.ps1

cmdlet 01-RegisterAppInADTenant.ps1 at command pipeline position 1
Supply values for the following parameters:
(Type !? for Help.)
Subscription: azurepass-bobf
AppName: B2CGraphClient

```

Gambar 10.23. Daftarkan utilitas baris perintah B2CGraphClient

Skrip pendaftaran menghasilkan dua file—`AppRegistration.txt` dan `ClientSecret.txt`. File Pendaftaran Aplikasi berisi ID Utama Aplikasi dan ID Objek. ID Objek digunakan untuk memperluas izin pendaftaran aplikasi. ID Utama Aplikasi dan Rahasia Klien ditambahkan ke kode sumber `B2CGraphClient` sehingga dapat mengautentikasi dan mengambil token dari Azure AD B2C.

9. Jalankan skrip `02-AssignAppPermission.ps1`, dengan meneruskan nama langganan dan ID Objek.
10. Buka solusi `B2CGraphClient` di Visual Studio lalu buka file `app.config`. Perbarui pengaturan aplikasi berikut:

```

key="b2c:Tenant" value="[your-tenant].onmicrosoft.com"
key="b2c:ClientId" value="[your-app-principal-id]"
key="b2c:ClientSecret" value="[your-client-secret]"

```

11. Bangun solusinya.
12. Untuk menguji aplikasi, navigasikan ke folder `B2C-GraphClient-DotNet\B2CGraphClient\Bin\Debug` dan ketik perintah berikut untuk mendapatkan bantuan untuk utilitas tersebut (lihat Gambar 10-24).

```
> .\b2c.exe help
```

```

Administrator: Windows PowerShell
-----
d----- 2/2/2017 6:05 PM B2C-GraphAPI-DotNet
d----- 1/22/2017 9:04 PM UserJson
-a----- 1/22/2017 9:04 PM 830 01-RegisterAppInADTenant.ps1
-a----- 1/22/2017 9:04 PM 648 02-AssignAppPermissions.ps1
-a----- 1/22/2017 9:04 PM 454 03-ImportUsers.ps1
-a----- 1/22/2017 9:04 PM 1096 AppRegistration.txt
-a----- 1/22/2017 9:04 PM 94 ClientSecret.txt
-a----- 1/22/2017 9:04 PM 124 tenant.txt

PS C:\users\bob\source\repos\brt\identity> cd b2c-graphapi-dotnet
PS C:\users\bob\source\repos\brt\identity\b2c-graphapi-dotnet> cd b2cgraphclient
PS C:\users\bob\source\repos\brt\identity\b2c-graphapi-dotnet\b2cgraphclient> cd bin
PS C:\users\bob\source\repos\brt\identity\b2c-graphapi-dotnet\b2cgraphclient\bin> cd debug
PS C:\users\bob\source\repos\brt\identity\b2c-graphapi-dotnet\b2cgraphclient\bin\debug> .\b2c.exe help
Get-User          : Read users from your B2C directory. Optionally accepts an ObjectId as a 2nd argument, and
Create-User       : Create a new user in your B2C directory. Requires a path to a .json file which contains r
2nd argument.
Update-User       : Update an existing user in your B2C directory. Requires an objectId as a 2nd arguemnt & a
nt.
Delete-User       : Delete an existing user in your B2C directory. Requires an objectId as a 2nd argument.
Get-Extension-Attribute : Lists all extension attributes in your B2C directory. Requires the b2c-extensions-app obj
Get-B2C-Application : Get the B2C Extensions Application in your B2C directory, so you can retrieve the objectId
Help              : Prints this help menu.
Syntax           : Gives syntax information for each command, along with examples.
PS C:\users\bob\source\repos\brt\identity\b2c-graphapi-dotnet\b2cgraphclient\bin\debug>

```

Gambar 10.24. Menguji utilitas baris perintah B2C Graph Client

13. Uji coba mendapatkan pengguna dari direktori dengan perintah ini:

```
> .\b2c.exe get-user
```

Anda akan melihat akun administrator dikembalikan sebagai aliran JSON.

14. Navigasikan kembali ke folder identitas dan jalankan skrip ketiga, 03-ImportUsers.ps1.

Skrip ini akan menggunakan konten folder identitas\UserJson untuk mengimpor pengguna untuk masing-masing dari tiga perusahaan semu yang kami tentukan.

15. Ketika skrip selesai, navigasikan kembali ke folder bin aplikasi dan jalankan kembali perintah get-user:

```
> .\b2c.exe get-user
```

Anda akan melihat 50+ pengguna kembali dari direktori (lihat Gambar 10-25).

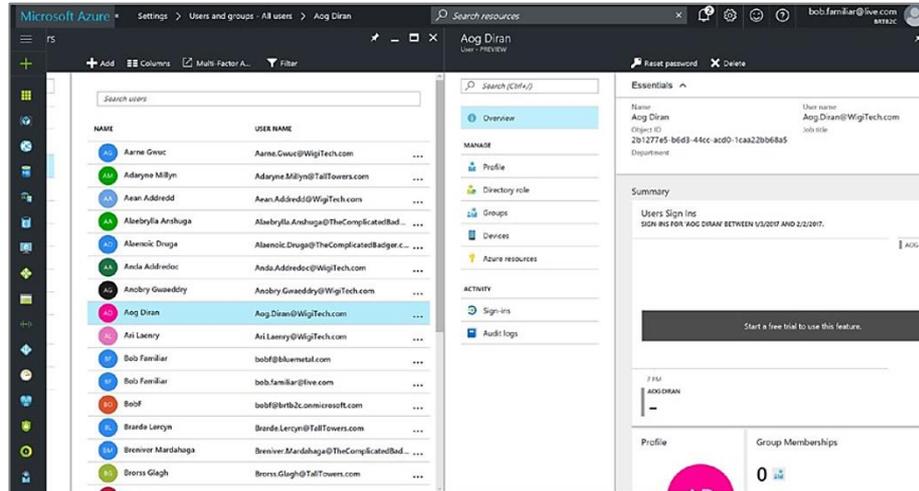
```

Administrator: Windows PowerShell
{
  "mailNickname": "Brorss.Glagh",
  "mobile": "2125551212",
  "onPremisesSecurityIdentifier": null,
  "otherMails": [],
  "passwordPolicies": "DisablePasswordExpiration",
  "passwordProfile": null,
  "physicalDeliveryOfficeName": null,
  "postalCode": "10018",
  "preferredLanguage": null,
  "provisionedPlans": [],
  "provisioningErrors": [],
  "proxyAddresses": [],
  "refreshTokensValidFromDateTime": "2017-01-23T00:31:30Z",
  "showInAddressList": null,
  "sipProxyAddress": null,
  "state": "NY",
  "streetAddress": "53 WEST 36TH STREET",
  "surname": "Glagh",
  "telephoneNumber": "2125551212",
  "thumbnailPhoto@odata.mediaEditLink": "directoryObjects/feffacce-f87c-44ec-97e9-4ed6ec5c5fdf/Microsoft.DirectorySer
  "usageLocation": null,
  "userPrincipalName": "07a4e71f-09ab-4828-8988-0e113118c96b@brtb2c.onmicrosoft.com",
  "userType": "Member"
}
PS C:\users\bob\source\repos\brt\identity\b2c-graphapi-dotnet\b2cgraphclient\bin\debug>

```

Gambar 10.25. Gunakan utilitas untuk membuat daftar semua pengguna di direktori

16. Kembali ke bilah manajemen Azure AD B2C dan klik Pengguna dan Grup lalu Semua Pengguna. Anda akan melihat bahwa direktori telah diisi dan Anda dapat melihat pengguna dan propertinya (lihat Gambar 10.26).

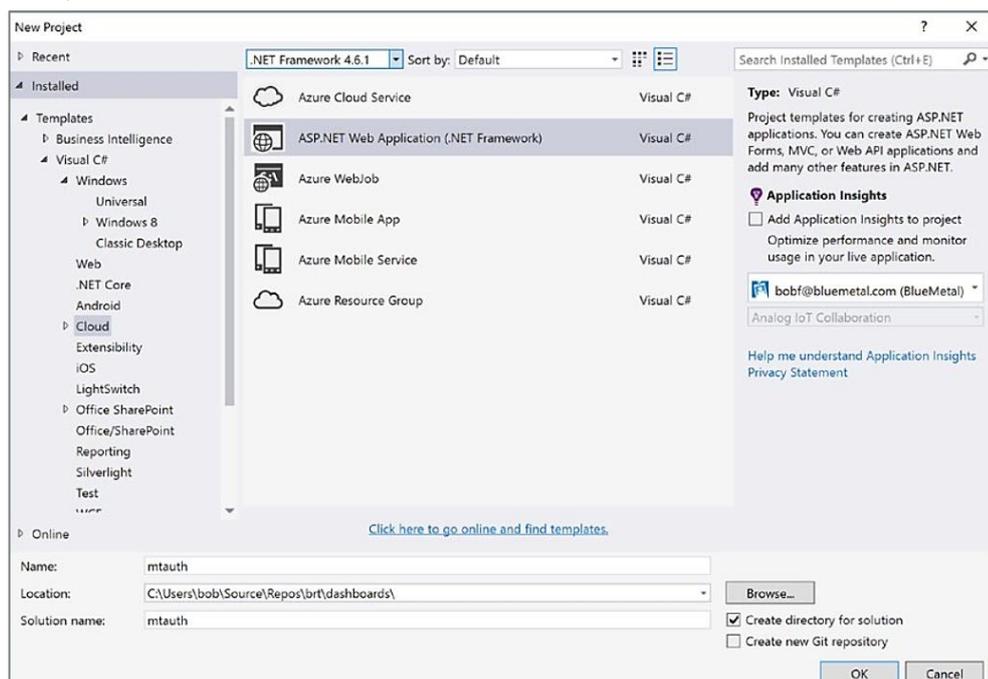


Gambar 10.26. Pengguna berhasil mengimpor ke penyewa

BUAT APLIKASI MULTI-TENANT

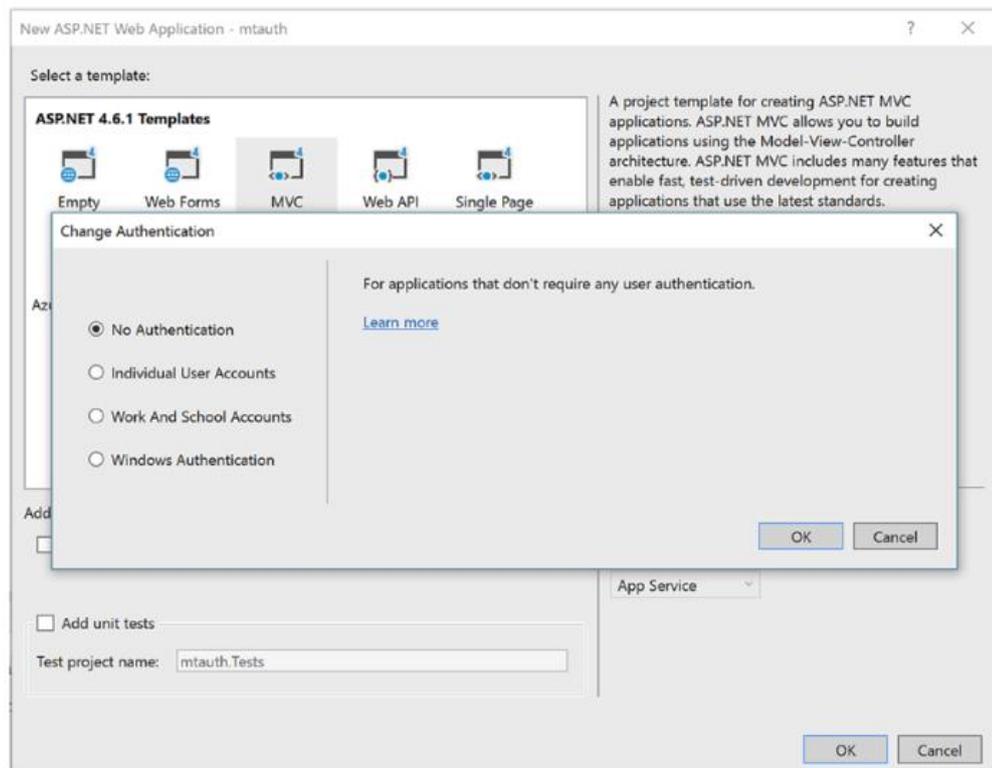
Dalam latihan ini, Anda akan membuat aplikasi web yang akan menggunakan Azure AD B2C untuk mengautentikasi pengguna, dan Anda akan menentukan Kebijakan Azure AD B2C untuk masuk, mendaftar, dan manajemen profil.

1. Buka Visual Studio dan buat proyek baru menggunakan template ASP.NET MVC. Pilih folder dashboard di repo dan atur versi framework ke 4.6.1. Hubungi solusi mtauth. Klik tombol OK untuk melanjutkan ke langkah berikutnya dalam wizard (lihat Gambar 10.27).



Gambar 10.27 Buat aplikasi web

- Pilih templat MVC dan ubah autentikasi menjadi Tanpa Otentikasi. Klik OK (lihat Gambar 10.28).

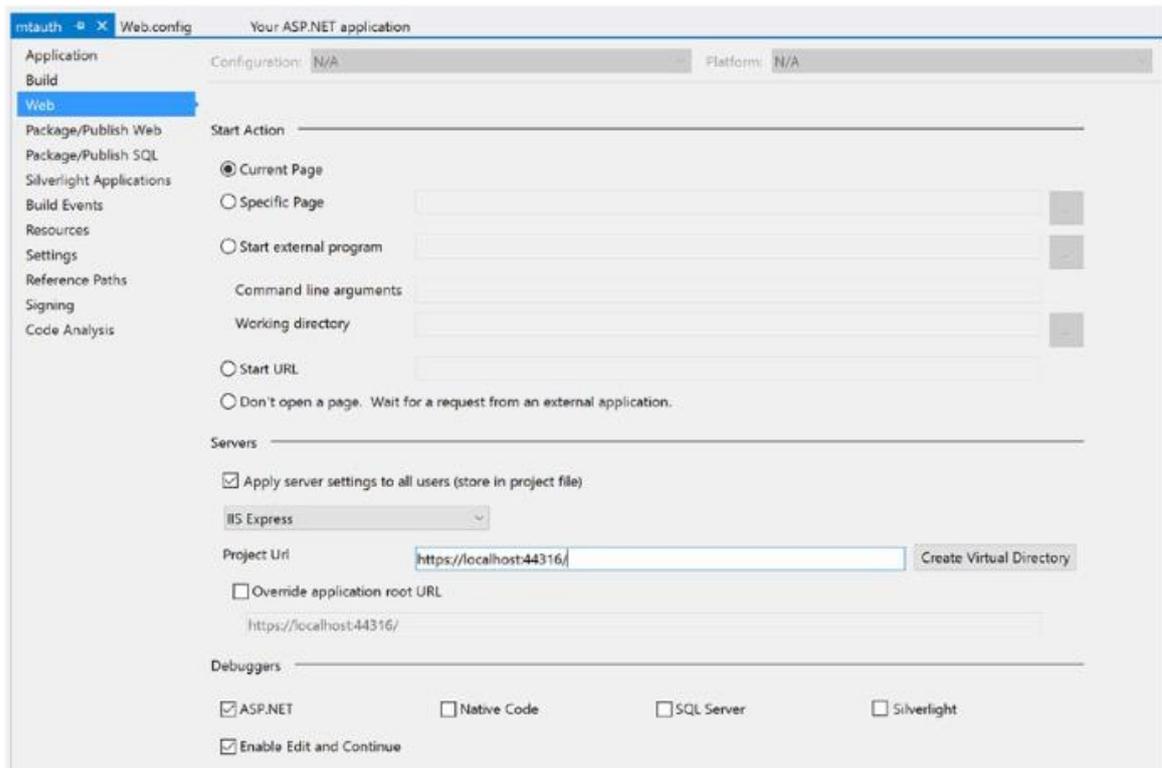


Gambar 10.28. Ubah otentikasi

- Dari menu, pilih Alat ► Manajer Paket NuGet ► Konsol Manajer Paket. Instal paket NuGet ini, yang menyediakan perancah untuk mengautentikasi ke Azure Active Directory menggunakan Open ID Connect:

```
Install-Package Microsoft.Owin.Security.OpenIdConnect
Install-Package Microsoft.Owin.Security.Cookies
Install-Package Microsoft.Owin.Host.SystemWeb
```

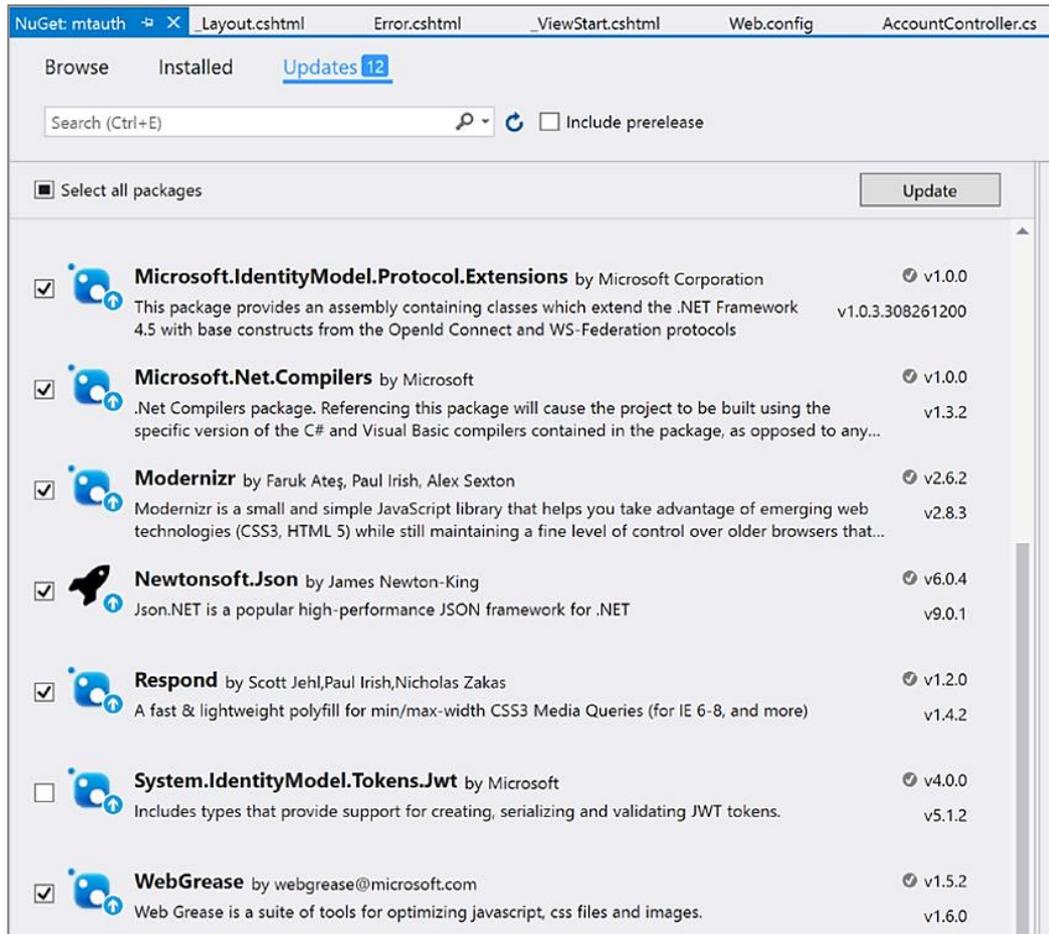
- Buka pengaturan proyek, pilih tab Web dan perbarui URL proyek menjadi `https://localhost:44316/`. Saat diminta untuk membuat direktori virtual, klik OK (lihat Gambar 10.29).



Gambar 10.29. Perbarui pengaturan proyek

5. Buka NuGet Packager Manager, pilih nuget.org di drop-down sumber, dan klik Updates.

Pilih semua paket yang akan diperbarui kecuali `System.IdentityModel`. Paket `Token.JWT`. Paket tersebut harus tetap pada versi 4.0 (lihat Gambar 10-30). Perhatikan bahwa Anda mungkin harus memulai ulang Visual Studio setelah pembaruan paket ini.



Gambar 10.30 Perbarui paket NuGet untuk solusinya

Ada beberapa file yang perlu Anda tambahkan ke proyek ini. Mereka telah disediakan dalam repo di folder `dashboards\mtauth-files`.

6. Tambahkan file `web.config` dari folder `mtauth-files` dan perbarui pengaturan aplikasi berikut. Perhatikan bahwa ID Aplikasi ini adalah yang dihasilkan saat Anda mendaftarkan aplikasi ini dengan Azure Active Directory di Latihan 2.

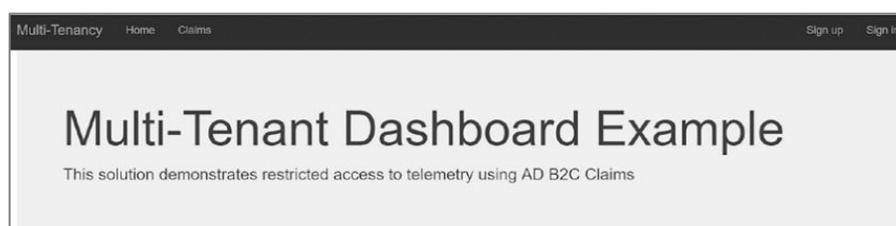
```
<add key="ida:Tenant"
  value="[your-tenant].onmicrosoft.com" />
<add key="ida:ClientId"
  value="[your-app-id]" />
<add key="ida:AadInstance"
  value="https://login.microsoftonline.com/{0}/v2.0/.well-known/
  openid-configuration?p={1}" />
<add key="ida:RedirectUri"
  value="https://localhost:44316/" />
<add key="ida:SignUpPolicyId"
  value="b2c_1_sign_up" />
<add key="ida:SignInPolicyId"
  value="b2c_1_sign_in" />
```

```
<add key="ida:UserProfilePolicyId"
      value="b2c_1_edit_profile" />
```

7. Tambahkan file bernama `Startup.cs` ke tingkat atas solusi. Kelas ini menyediakan titik masuk untuk startup aplikasi. Perhatikan bahwa implementasi `ConfigureAuth()` belum ditentukan.
8. Tambahkan file bernama `Startup.Auth.cs` di folder `App_Start`. Kelas ini memuat pengaturan dari `Web.config` dan kemudian memanggil Open ID Connect API untuk mengautentikasi pengguna dan menjalankan kebijakan. Kelas ini mengimplementasikan rutinitas `ConfigureAuth()`.
9. Tambahkan file `AccountController.cs` ke folder `Controllers`. Kelas ini menyediakan perutean ke metode pemanggilan kebijakan kelas `Startup` yang dipicu saat pengguna berinteraksi dengan antarmuka pengguna.
10. Buka file `HomeController.cs` dan tambahkan titik akhir untuk merutekan pengguna ke tampilan Klaim. Tampilan Klaim akan menampilkan data klaim yang dikembalikan dari Azure AD B2C.

```
[Authorize]
public ActionResult Claims()
{
    var displayName =
        ClaimsPrincipal.Current.FindFirst(
            ClaimsPrincipal.Current
                .Identities.First().NameClaimType);
    ViewBag.DisplayName = displayName != null ?
        displayName.Value : string.Empty;
    return View();
}
```

11. Ada file View yang diperbarui di folder `mtauth-files`. Salin file tersebut ke file yang ada dalam solusi Anda, lalu tambahkan file `Claims.cshtml` menggunakan perintah menu Tambahkan Item yang Ada di Visual Studio.
12. Tambahkan file `profile-dropdown.js` ke folder `Scripts` dalam solusi Anda menggunakan perintah menu Add Existing Item di Visual Studio.
13. Bangun dan jalankan solusinya. Akan muncul halaman beranda yang menampilkan pilihan menu Sign Up, Sign In, dan Claims (lihat Gambar 10.31).

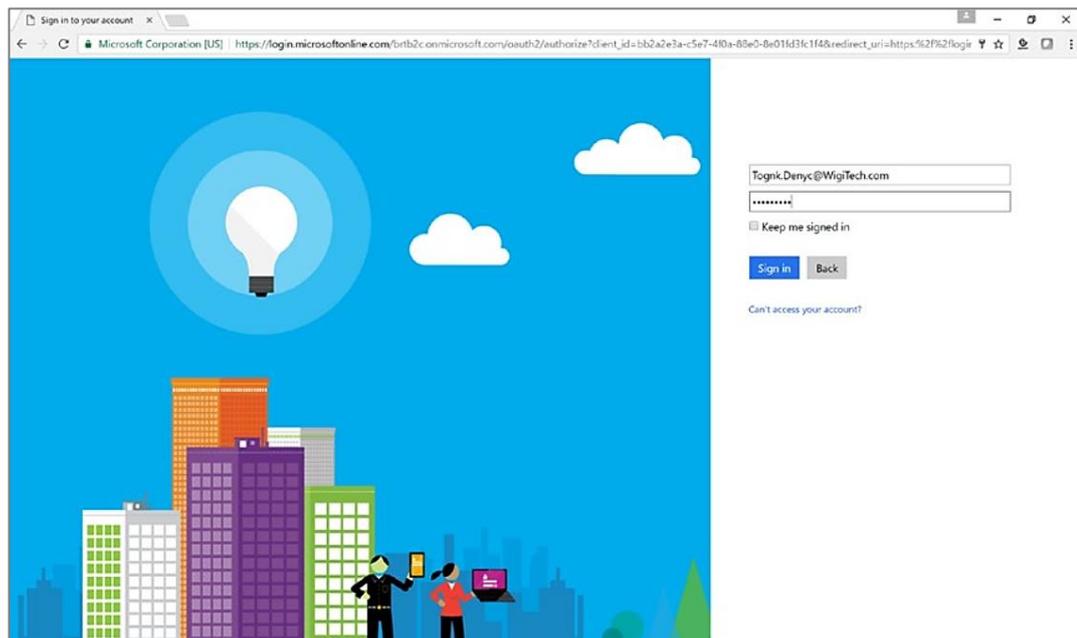


Gambar 10.31. Halaman beranda dasbor multi-penyewa

14. Klik Masuk. Anda akan diarahkan ke Halaman Kebijakan Masuk Azure AD B2C. Perhatikan bahwa ini adalah tampilan dan nuansa default. Dimungkinkan untuk menerapkan gaya Anda sendiri pada halaman ini.
15. Tognk adalah karyawan di WigiTech. Dia adalah administrator di sana sehingga dia telah meningkatkan hak istimewa dalam solusi tersebut, seperti yang akan Anda lihat pada latihan selanjutnya (lihat Gambar 10-32). Masuk menggunakan kredensialnya:

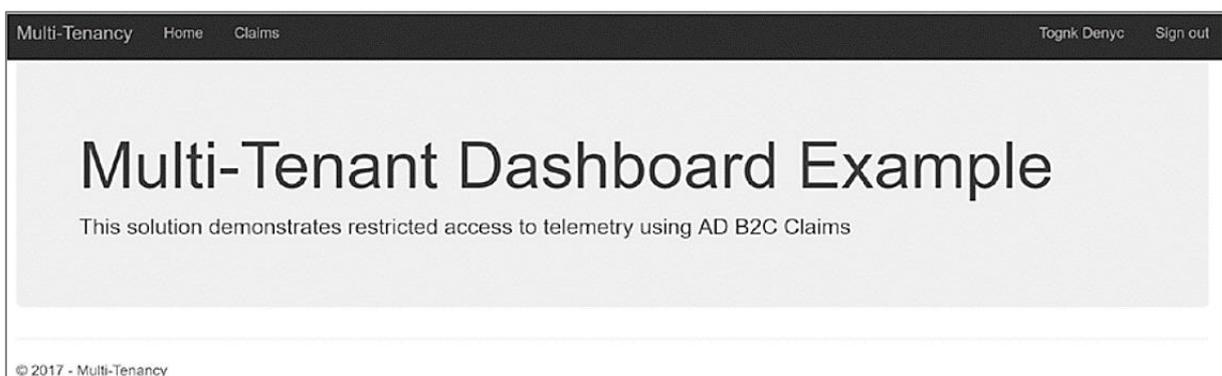
Username: Tognk.Denyc@WigiTech.com

Password: P@ssword



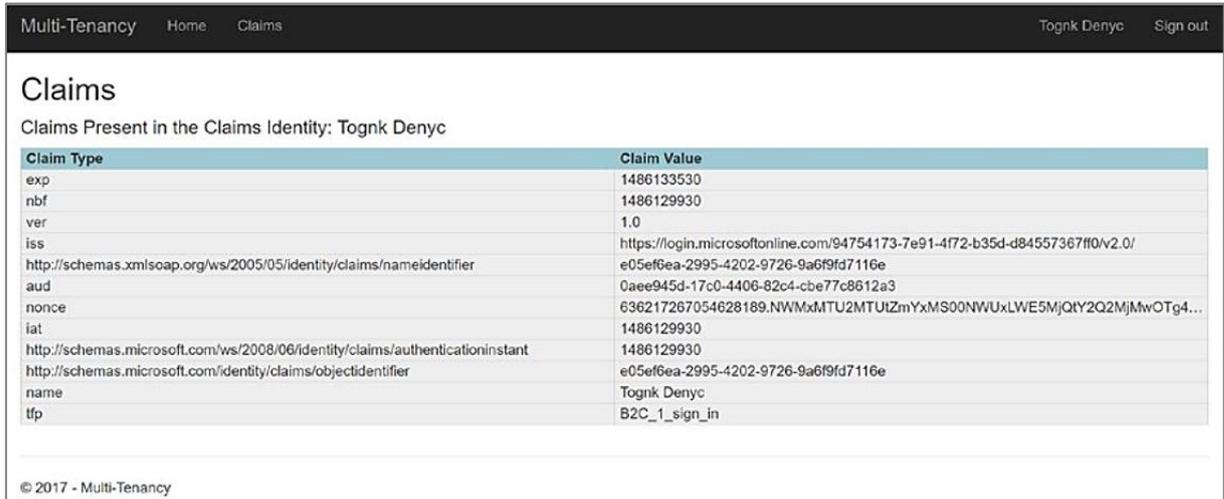
Gambar 10.32. Masuk ke dasbor

16. Setelah diautentikasi, Anda akan diarahkan kembali ke halaman beranda, dan Anda akan melihat nama Tognk muncul di tempat yang dulunya bertuliskan Daftar, dan Masuk telah diganti dengan Keluar (lihat Gambar 10.33).



Gambar 10.33. Pengguna diautentikasi

17. Klik pada pilihan menu Klaim. Anda akan dialihkan ke tampilan Klaim, yang akan menampilkan klaim yang berasal dari Azure AD B2C untuk pengguna ini (lihat Gambar 10.34).



Claim Type	Claim Value
exp	1486133530
nbf	1486129930
ver	1.0
iss	https://login.microsoftonline.com/94754173-7e91-4f72-b35d-d84557367f0/v2.0/
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier	e05ef6ea-2995-4202-9726-9a6f9fd7116e
aud	0aee945d-17c0-4406-82c4-cbe77c8612a3
nonce	636217267054628189.NWMxMTU2MTU2mYxMS00NWUxLWE5MjQY2Q2MjMwOTg4...
iat	1486129930
http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationinstant	1486129930
http://schemas.microsoft.com/identity/claims/objectidentifier	e05ef6ea-2995-4202-9726-9a6f9fd7116e
name	Tognk Denyc
tfp	B2C_1_sign_in

© 2017 - Multi-Tenancy

Gambar 10.34. Klaim pengguna

18. Uji coba aplikasi dengan pengguna lain dari perusahaan yang sama dan dari perusahaan lain. Perhatikan bahwa kata sandi untuk semua akun pengguna adalah P@ssword!. Berikut beberapa akun yang dapat Anda gunakan untuk menguji:

WigiTech

Admin - Tognk.Denyc@WigiTech.com
Employee - Aog.Diran@WigiTech.com

Tall Towers

Admin - Fid.Sidry@TallTowers.com
Employee - Maell.Driga@TallTowers.com

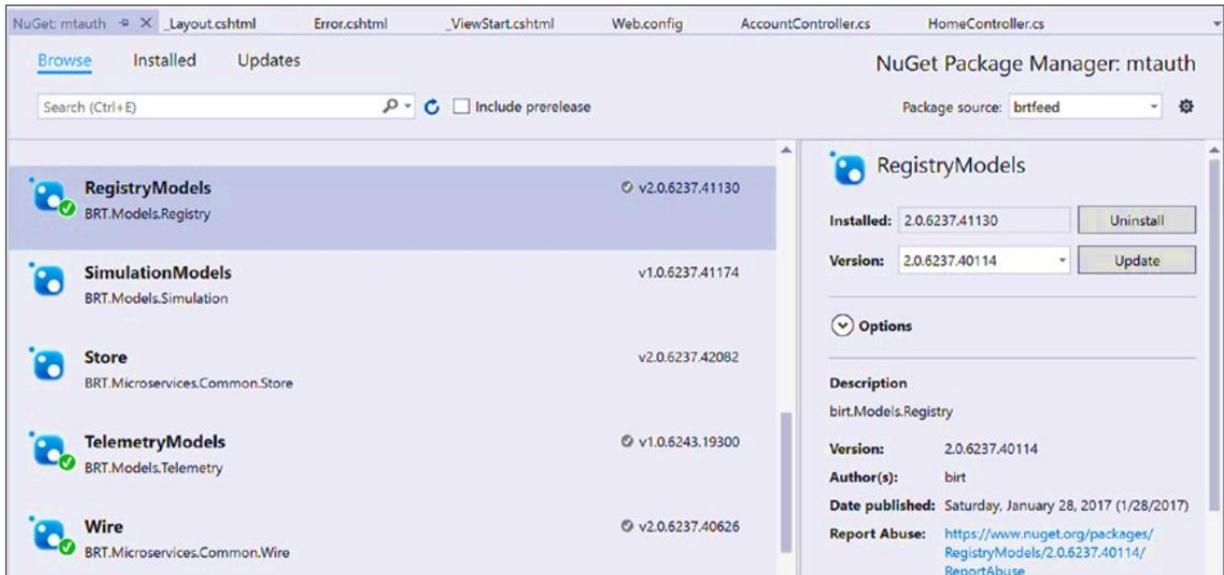
The Complicated Badger

Admin - Jyssa.Anuk@TheComplicatedBadger.com
Employee - Jill.Aniua@TheComplicatedBadger.com

10.5 PERBARUI APLIKASI UNTUK MENGHUBUNGI API PLATFORM

Aplikasinya sekarang multi-penyewa. Anda dapat masuk ke aplikasi menggunakan akun dari salah satu tiga organisasi yang didukung. Langkah selanjutnya adalah menggunakan informasi dalam klaim untuk memanggil API platform dan mengambil detail tentang pengguna.

1. Tambahkan referensi ke paket NuGet Wire, Registry Model, dan Telemetry Model dalam solusi mtauth (lihat Gambar 10.35).



Gambar 10.35. Tambahkan paket NuGet

2. Tambahkan pengaturan berikut di file Web.config:

```
<add key="RegistryAPI" value="https://[your-apim-host].azure-
api.net/dev/v1/registry/profiles"/>
```

```
<add key="TelemetryAPI" value="https:// ://[your-apim-
host].azure-api.net/
dev/v1/telemetry/events" />
```

```
<add key="DevKey" value="subscription-key=[your-key]" />
```

3. Buka file HomeController.cs dan ganti metode Claims dengan kode berikut. Kode ini mencari profil pengguna dan, menggunakan properti type dari profil, menyiapkan panggilan ke API Telemetri. API Telemetri mengambil parameter nama perusahaan, jenis pengguna, email pengguna, jumlah catatan data yang akan diambil, dan kunci langganan APIM. Koleksi yang dikembalikan dimuat ke objek ViewBag untuk diproses oleh tampilan Klaim.

```
[Authorize]
public ActionResult Claims()
{
    var displayName = ClaimsPrincipal.Current.FindFirst(
        ClaimsPrincipal
            .Current
            .Identities
            .First().NameClaimType);
    if (displayName != null)
    {
        ViewBag.DisplayName = displayName.Value;
    }
}
```

```

var registryApi = ConfigurationManager
.AppSettings["RegistryAPI"];
var telemetryApi = ConfigurationManager
.AppSettings["TelemetryAPI"];
var devKey = ConfigurationManager
.AppSettings["DevKey"];
// get the full name and split into first, last
var names = displayName.Value.Split(' ');
var firstname = names[0];
var lastname = names[1];
// set up the call to the registry api
var query = $"/firstname/{firstname}" +
$
$/lastname/{lastname}?{devKey}";
var api = registryApi + query;
var json = Rest.Get(new Uri(api));
var profile = ModelManager
.JsonToModel<Profile>(json);
    // NOTE: only type 2 (admin) and type 3
    // (employee) are able to access telemetry.
    // All other user types will return an empty
    // telemetry collection
var type = 0;
if (profile != null)
{
    switch (profile.type)
    {
        case ProfileTypeEnum.NotSet:
            type = 0;
            break;
        case ProfileTypeEnum.Organization:
            type = 1;
            break;
        case ProfileTypeEnum.Administrator:
            type = 2;
            break;
        case ProfileTypeEnum.Employee:
            type = 3;
            break;
        case ProfileTypeEnum.Contractor:
            type = 4;
            break;
        case ProfileTypeEnum.Temporary:
            type = 5;
            break;
        case ProfileTypeEnum.Partner:
            type = 6;
            break;
    }
}

```

```

        default:
            throw new
                ArgumentOutOfRangeException();
    }
    query = "/companyname/" +
        $"{profile.companyname}" +
        $"?usertype={type}" +
        $"&username={profile.username}" +
        $"&count=15&{devKey}";
    api = telemetryApi + query;
    json = Rest.Get(new Uri(api));
    var telemetryList = ModelManager
        .JsonToModel<TelemetryList>(json);
    ViewBag.UserProfile = profile;
    ViewBag.Telemetry = telemetryList.list;
    }
}
return View();
}

```

4. Buka file `Views\Home\Claims.cshtml` dan perbarui isi file dengan kode berikut.

Kode ini akan menampilkan tabel klaim pengguna serta menampilkan daftar 15 pembacaan telemetri terbaru yang dapat dilihat pengguna tersebut. Jika masuk sebagai karyawan, pengguna hanya akan melihat pembacaan telemetri miliknya sendiri. Jika login sebagai administrator, mereka akan melihat pembacaan dari karyawannya.

```

@using System.Security.Claims
@using BRT.Models.Telemetry
@{
    ViewBag.Title = "Claims";
}
<h2>@ViewBag.Title</h2>
<h4>Claims Present for @ViewBag.DisplayName</h4>
<table class="table-hover claim-table">
<tr>
<th class="claim-type claim-data claim-head">Claim Type</th>
<th class="claim-data claim-head">Claim Value</th>
</tr>
@foreach (Claim claim in
    ClaimsPrincipal.Current.Claims)
    {
    }
</table>
<br/>
<table>

```

```

<tr>
<tr>
<td class="claim-type claim-data">
@claim.Type
</td>
<td class="claim-data">
@claim.Value
</td>
</tr>
<td class="claim-data">Lastname</td>
<td class="claim-data">Age</td>
<td class="claim-data">Weight</td>
<td class="claim-data">Height</td>
<td class="claim-data">Activity</td>
<td class="claim-data">Altitude</td>
<td class="claim-data">Breathing Rate</td>
<td class="claim-data">Cadence</td>
<td class="claim-data">Heart Rate</td>
<td class="claim-data">Red Zone</td>
<td class="claim-data">Variability</td>
<td class="claim-data">Speed</td>
<td class="claim-data">Steps</td>
    <td class="claim-data">Temperature</td>
    <td class="claim-data">Velocity</td>
    <td class="claim-data">Ventilization</td>
</tr>
@foreach (UserTelemetry telemetry in
    ViewBag.Telemetry)
{
    <tr>
        <td class="claim-data">
            @telemetry.User.lastname
        </td>
        <td class="claim-data">
            @telemetry.Readings.age
        </td>
        <td class="claim-data">
            @telemetry.Readings.weight
        </td>
        <td class="claim-data">
            @telemetry.Readings.height
        </td>
        <td class="claim-data">
            @telemetry.Readings.activity
        </td>
        <td class="claim-data">
            @telemetry.Readings.altitude
        </td>

```

```

        <td class="claim-data">
            @telemetry.Readings.breathingRate
        </td>
        <td class="claim-data">
            @telemetry.Readings.cadence
        </td>
        <td class="claim-data">
            @telemetry.Readings.heartRateBPM
        </td>
        <td class="claim-data">
            @telemetry.Readings.heartRateRedZone
        </td>
        <td class="claim-data">
            @telemetry.Readings.heartrateVariability
        </td>
        <td class="claim-data">
            @telemetry.Readings.speed
        </td>
        <td class="claim-data">
            @telemetry.Readings.steps
        </td>
        <td class="claim-data">
            @telemetry.Readings.temperature
        </td>
        <td class="claim-data">
            @telemetry.Readings.velocity
        </td>
        <td class="claim-data">
            @telemetry.Readings.ventilization
        </td>
    </tr>
    }
</table>

```

5. Jalankan simulator telemetri (lihat Bab 4). Perhatikan bahwa pekerjaan Stream Analytics harus berjalan agar data dirutekan ke tabel Database SQL.
6. Uji aplikasi portal dengan masuk sebagai administrator dan karyawan dari perusahaan yang berbeda. Lihat Gambar 10.36 dan 10.37.

Claim Type	Claim Value
exp	1486230722
nbf	1486227122
ver	1.0
iss	https://login.microsoftonline.com/94754173-7e91-4f72-b35d-d84557367f0/v2.0/
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier	e05ef6ea-2995-4202-9726-9a6f9fd7116e
aud	0aae945d-17c0-4406-82c4-cbe77c8612a3
nonce	636218238730291984.NzBmMzE0OGUNjMzS00NTNmLWFJmZUUMTJINWUwMDBIOT...
iat	1486227122
http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationinstant	1486227122
http://schemas.microsoft.com/identity/claims/objectidentifier	e05ef6ea-2995-4202-9726-9a6f9fd7116e
name	Tognk Denyc
tfp	B2C_1_sign_in

Lastname	Age	Weight	Height	Activity	Altitude	Breathing Rate	Cadence	Heart Rate	Red Zone	Variability	Speed	Steps	Tempera...	Velocity	Ventiliza...
Rheoran	47	220	75	17290.9...	28.3878...	0	0.16287...	95	18.0080...	0	10	96.8734...	0	0	0
Aethocyn	68	186	65	17104.9...	23.1272...	0	0.15636...	94	17.8839...	0	8	96.1556...	0	0	0
Gwaeddry	64	150	64	16826.0...	13.1729...	0	0.14659...	92	17.6976...	0	5	96.9983...	0	0	0
Canullyn	25	165	69	16454.1...	3.13471...	0	0.13356...	90	17.4492...	0	1	96.1492...	0	0	0
Addressoc	42	210	64	16547.1...	7.60836...	0	0.13682...	91	17.5113...	0	2	96.7376...	0	0	0
Rheoran	47	220	75	17290.9...	28.7533...	0	0.16287...	95	18.0080...	0	10	96.4878...	0	0	0
Sur	31	210	70	16454.1...	3.57567...	0	0.13356...	90	17.4492...	0	1	96.0263...	0	0	0
Aethocyn	68	186	65	17011.9...	19.5525...	0	0.15310...	93	17.8218...	0	7	96.9306...	0	0	0
Addressd	34	195	72	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.0481...	0	0	0
Blaenyc	23	145	68	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.5872...	0	0	0
Diran	28	190	71	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.1024...	0	0	0
Laenry	57	235	73	16919.0...	19.7206...	0	0.14985...	93	17.7597...	0	6	96.3790...	0	0	0
Addressm...	27	100	63	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.9357...	0	0	0
Gubryn	36	223	73	16454.1...	2.92313...	0	0.13356...	90	17.4492...	0	1	96.0236...	0	0	0
Mun	54	235	70	18592.5...	60.5532...	0	0.20848...	102	18.8773...	0	24	96.5201...	0	0	0

Gambar 10.36. Masuk sebagai Tognk Denyc, administrator di WigiTech

Kami akan menggunakan klaim yang dikembalikan dari AD untuk mencari pengguna di sistem dan meneruskan informasi profil pengguna ke API Telemetri untuk menunjukkan pembatasan akses ke telemetri. Pengguna yang akan kami gunakan untuk pengujian berasal dari organisasi WigiTech. Tognk Denyc adalah seorang administrator, sehingga dia dapat melihat semua informasi karyawan dari WigiTech. Aog Diran adalah seorang karyawan, jadi Aog hanya bisa melihat informasinya sendiri.

Claim Type	Claim Value
exp	1486230940
nbf	1486227340
ver	1.0
iss	https://login.microsoftonline.com/94754173-7e91-4f72-b35d-d84557367f0/v2.0/
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier	2b1277e5-b6d3-44cc-acc0-1caa22bb68a5
aud	0aae945d-17c0-4406-82c4-cbe77c8612a3
nonce	636218241016037504.OwVhOGVvNvYiZDVhYS00MzA2LWFIQGYiOWJINdG3NzlwZm...
iat	1486227340
http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationinstant	1486227340
http://schemas.microsoft.com/identity/claims/objectidentifier	2b1277e5-b6d3-44cc-acc0-1caa22bb68a5
name	Aog Diran
tfp	B2C_1_sign_in

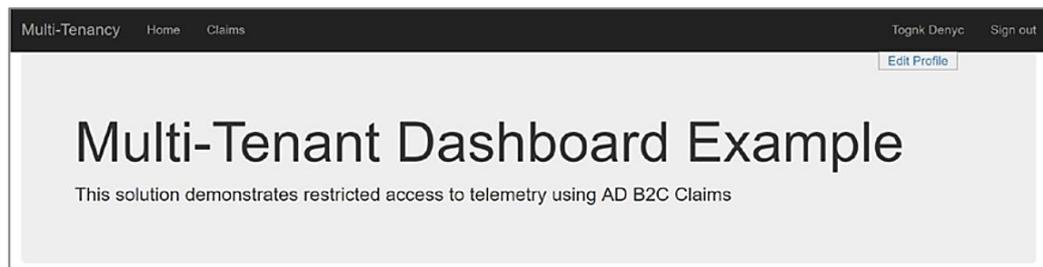
Lastname	Age	Weight	Height	Activity	Altitude	Breathing Rate	Cadence	Heart Rate	Red Zone	Variability	Speed	Steps	Tempera...	Velocity	Ventiliza...
Diran	28	190	71	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.1024...	0	0	0
Diran	28	190	71	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.6383...	0	0	0
Diran	28	190	71	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.1907...	0	0	0
Diran	28	190	71	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.1996...	0	0	0
Diran	28	190	71	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.6571...	0	0	0
Diran	28	190	71	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.4389...	0	0	0
Diran	28	190	71	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.0501...	0	0	0
Diran	28	190	71	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.8051...	0	0	0
Diran	28	190	71	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.4103...	0	0	0
Diran	28	190	71	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.0208...	0	0	0
Diran	28	190	71	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.0073...	0	0	0
Diran	28	190	71	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.1677...	0	0	0
Diran	28	190	71	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.0812...	0	0	0
Diran	28	190	71	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.1042...	0	0	0
Diran	28	190	71	16361.18	0	0	0.13030...	90	17.3872...	0	0	96.1397...	0	0	0

Gambar 10.37. Masuk sebagai Aog Diran, karyawan WigiTech

Hal terakhir yang ingin kami validasi dalam aplikasi kami adalah pengguna dapat memperbarui profil mereka. Pada latihan sebelumnya, kami menetapkan kebijakan edit profil yang memungkinkan pengguna memperbarui beberapa atribut, termasuk atribut khusus yang disebut Peran.

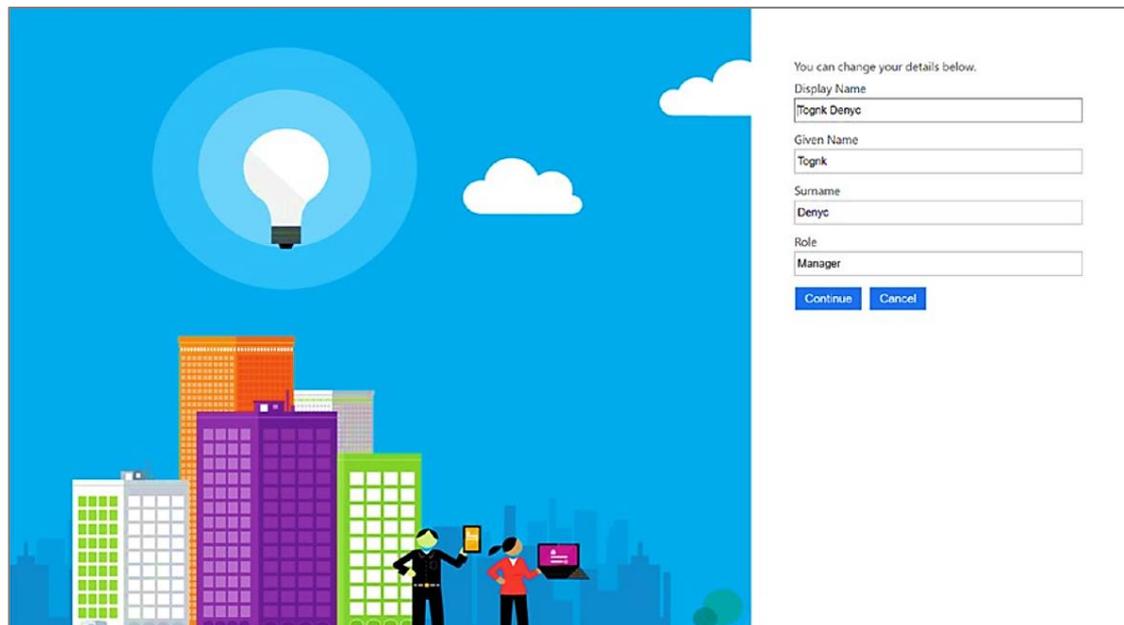
Ikuti langkah-langkah berikut untuk memvalidasi bahwa fitur ini berfungsi.

7. Klik nama pengguna di pojok kanan atas aplikasi. Tautan Edit Profil ditampilkan (lihat Gambar 10-38).



Gambar 10-38. Opsi menu Edit Profil

8. Klik tautannya. Anda akan diarahkan ke halaman yang disediakan oleh Azure AD B2C yang memungkinkan pengguna memperbarui profil berdasarkan definisi kebijakan. Perhatikan bahwa bidang Peran tersedia untuk diedit (lihat Gambar 10-39).



Gambar 10.39 Editor Profil Pengguna disediakan oleh Azure AD B2C

BAB 11

EPILOG

Setiap bisnis sedang atau akan segera menghadapi transformasi digital. Untuk mempercepat transformasi ini dan memberikan solusi yang paling berdampak kepada pelanggan, mitra, dan karyawan, bisnis-bisnis ini perlu melakukan hal tersebut menggabungkan data real-time. Meskipun mungkin ada layanan SaaS yang dirancang untuk memberikan solusi end-to-end yang lengkap, pengalaman kami menunjukkan bahwa saat ini tidak ada pilihan untuk membeli produk yang benar-benar memberikan apa yang dibutuhkan organisasi. Untuk mendorong transformasi digital, organisasi mempunyai dua pilihan:

- Tunggu hingga pasar ISV mengejar ketinggalan dan berisiko kehilangan peluang yang ada.
- Membangun solusi saat ini yang dapat berkembang dan beradaptasi di masa depan.

Pekerjaan kami dengan klien selama beberapa tahun terakhir telah melahirkan pendekatan rekayasa yang disoroti oleh praktik terbaik sehubungan dengan siklus hidup produk perangkat lunak, arsitektur berbasis layanan mikro, dan kekuatan serta kohesi layanan cloud yang disediakan oleh Microsoft Azure. Memanfaatkan layanan PaaS Azure menghilangkan kebutuhan untuk mengelola sumber daya pada lingkungan berbasis mesin virtual, mengonfigurasi dan melakukan patch sistem operasi, dan mengembangkan infrastruktur skala elastis Anda sendiri. Sebaliknya, Anda fokus pada otomatisasi, konfigurasi, dan hanya kode yang diperlukan untuk menentukan dan mengimplementasikan kebutuhan bisnis Anda.

Perjalanan untuk memberikan solusi transformatif bisnis dimulai dengan menumbuhkan dan mengintegrasikan budaya DevOps dalam organisasi Anda. DevOps adalah tentang manusia, proses, dan alat yang bekerja secara kolaboratif untuk mendukung proses pengembangan perangkat lunak berkecepatan tinggi yang penting untuk kesuksesan. Tentu saja, teknologinya luar biasa, namun ia tidak menyediakan, membangun, dan menyebarkan dirinya sendiri.

Manajemen perangkat sering kali diabaikan ketika organisasi pertama kali terjun ke IoT. Sangat mudah untuk membuat solusi awal berskala kecil dengan komputer papan tunggal dan kemudian secara manual mendukung penerapannya. Kemampuan untuk mengotomatiskan pemantauan dan komunikasi dengan perangkat Anda ketika Anda berpotensi memiliki puluhan ribu perangkat di alam liar harus dipertimbangkan dengan cermat.

Betapapun pentingnya perangkat, Anda tidak boleh lupa bahwa IoT tetap mengutamakan data. Data adalah inti mentah yang diharapkan oleh perusahaan untuk ditambang untuk mendapatkan emas, yaitu wawasan tajam yang benar-benar mendorong transformasi. Keberhasilan di sini memerlukan perancangan dan penerapan saluran data yang kuat yang menggabungkan pemrosesan waktu nyata, mengintegrasikan data referensi, dan memanggil API Pembelajaran Mesin yang dioperasikan. Pipeline seperti itu membahas tiga jalur untuk data:

- Panas (seperti peringatan yang memerlukan tindakan segera)
- Hangat (seperti telemetri)
- Dingin (untuk pengarsipan dan analisis)

Spektrum data dan analitik terkait ini sangat penting dalam menyediakan visualisasi, peringatan, dan notifikasi yang kaya yang pada akhirnya mendorong transformasi.

Berfokus pada keamanan sejak awal pengembangan produk Anda adalah hal yang terpenting. Mengadopsi proses pemodelan dan mitigasi ancaman serta memanfaatkan alat seperti Microsoft Defender untuk Cloud dapat membantu Anda mengidentifikasi ancaman dan memitigasi risiko. Kami sangat berharap Anda menikmati membaca buku ini dan mengerjakan latihannya sama seperti kami menikmati meneliti dan menyusunnya. Semoga semua usaha bisnis Anda berjalan dengan kecepatan Internet!

DAFTAR PUSTAKA

- Belli, L., Davoli, L., Medioli, A., Marchini, P. L., & Ferrari, G. (2019). Toward Industry 4.0 with IoT: Optimizing business processes in an evolving manufacturing factory. *Frontiers in ICT*, 6, 17.
- Biswas, A. R., & Giaffreda, R. (2014, March). IoT and cloud convergence: Opportunities and challenges. In *2014 IEEE World Forum on Internet of Things (WF-IoT)* (pp. 375-376). IEEE.
- Biswas, S., Sharif, K., Li, F., Maharjan, S., Mohanty, S. P., & Wang, Y. (2019). PoBT: A lightweight consensus algorithm for scalable IoT business blockchain. *IEEE Internet of Things Journal*, 7(3), 2343-2355.
- Brouns, N., Tata, S., Ludwig, H., Asensio, E. S., & Grefen, P. (2018). Modeling iot-aware business processes-a state of the art report. arXiv preprint arXiv:1811.00652.
- Bujari, A., Furini, M., Mandreoli, F., Martoglia, R., Montangelo, M., & Ronzani, D. (2018). Standards, security and business models: key challenges for the IoT scenario. *Mobile Networks and Applications*, 23, 147-154.
- Cheng, Y., Zhao, S., Cheng, B., Hou, S., Shi, Y., & Chen, J. (2018). Modeling and optimization for collaborative business process towards IoT applications. *Mobile Information Systems*, 2018.
- Chiu, H. H., & Wang, M. S. (2013). A study of iot-aware business process modeling. *International Journal of Modeling and Optimization*, 3(3), 238.
- Dar, K., Taherkordi, A., Baraki, H., Eliassen, F., & Geihs, K. (2015). A resource oriented integration architecture for the Internet of Things: A business process perspective. *Pervasive and Mobile Computing*, 20, 145-159.
- Fattouch, N., Lahmar, I. B., & Boukadi, K. (2020, September). IoT-aware Business Process: comprehensive survey, discussion and challenges. In *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)* (pp. 100-105). IEEE.
- Ilyashenko, O., Kovaleva, Y., Burnatcev, D., & Svetunkov, S. (2020, September). Automation of business processes of the logistics company in the implementation of the IoT. In *IOP Conference Series: Materials Science and Engineering* (Vol. 940, No. 1, p. 012006). IOP Publishing.
- Ingemarsdotter, E., Jamsin, E., & Balkenende, R. (2020). Opportunities and challenges in IoT-enabled circular business model implementation—A case study. *Resources, Conservation and Recycling*, 162, 105047.
- Krotov, V. (2017). The Internet of Things and new business opportunities. *Business Horizons*, 60(6), 831-841.

- Mahendra, S., Sathiyarayanan, M., & Vasu, R. B. (2018, August). Smart security system for businesses using internet of things (iot). In 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT) (pp. 424-429). IEEE.
- Moh'd Ali, M. A., Basahr, A., Rabbani, M. R., & Abdulla, Y. (2020, November). Transforming business decision making with internet of things (IoT) and machine learning (ML). In 2020 International Conference on Decision Aid Sciences and Application (DASA) (pp. 674-679). IEEE.
- Navani, D., Jain, S., & Nehra, M. S. (2017, December). The internet of things (IoT): A study of architectural elements. In 2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) (pp. 473-478). IEEE.
- Park, S., Cho, K., Kim, S., Yoon, G., Choi, M. I., Park, S., & Park, S. (2021). Distributed energy IoT-based real-time virtual energy prosumer business model for distributed power resource. *Sensors*, 21(13), 4533.
- Raturi, R. (2018). Machine learning implementation for business development in real time sector. *International Journal of Pure and Applied Mathematics*, 119(15), 1289-1300.
- Raza, U., Lomax, J., Ghafir, I., Kharel, R., & Whiteside, B. (2017, July). An IoT and business processes based approach for the monitoring and control of high value-added manufacturing processes. In *Proceedings of the International Conference on Future Networks and Distributed Systems* (pp. 1-8).
- Singh, S., & Singh, N. (2015, October). Internet of Things (IoT): Security challenges, business opportunities & reference architecture for E-commerce. In 2015 International conference on green computing and internet of things (ICGCIoT) (pp. 1577-1581). IEEE.
- Verma, S., Kawamoto, Y., Fadlullah, Z. M., Nishiyama, H., & Kato, N. (2017). A survey on network methodologies for real-time analytics of massive IoT data and open research issues. *IEEE Communications Surveys & Tutorials*, 19(3), 1457-1477.
- Weinberger, M., Bilgeri, D., & Fleisch, E. (2016). IoT business models in an industrial context. *at-Automatisierungstechnik*, 64(9), 699-706.
- Zancul, E. D. S., Takey, S. M., Barquet, A. P. B., Kuwabara, L. H., Cauchick Miguel, P. A., & Rozenfeld, H. (2016). Business process support for IoT based product-service systems (PSS). *Business Process Management Journal*, 22(2), 305-323.

Analisis Bisnis secara Real Time dalam Transformasi Digital

Dr. Budi Raharjo, S.Kom., M.Kom., MM.



BIODATA PENULIS



Dr. Budi Raharjo, S.Kom, M.Kom, MM lahir di Semarang, tanggal 22 Februari 1985. Beliau adalah Alumni dari Universitas Bina Nusantara (BINUS University) Jakarta dan juga alumni Universitas Kristen Satya wacana (UKSW) Salatiga. Dr. Budi Raharjo telah menjadi Dosen pada Universitas STEKOM pada mata kuliah Kepemimpinan (Leadership), mata kuliah Pengantar Akuntansi, Manajemen Proses, Manajemen Akuntansi dan Manajemen Resiko Bisnis. Selain sebagai dosen Universitas STEKOM, Dr. Budi Raharjo, M.Kom, MM juga mempunyai bisnis sendiri dalam bidang perhotelan dan juga sebagai wirausaha dalam bidang pemasok unggas (ayam) beku, ke berbagai kota besar, khususnya Jakarta dan sekitarnya.

Pengalaman beliau berwirausaha menjadi bekal utama dalam penulisan buku ajar yang diterbitkan oleh Yayasan Prima Agus Teknik (YPAT) Semarang. Oleh sebab itu bukunya berisi langkah langkah praktis yang mudah diikuti oleh para mahasiswa, saat mahasiswa mengikuti proses perkuliahan pada Universitas Sains dan Teknologi Komputer (Universitas STEKOM). Jabatan struktural yang di embannya saat ini adalah Wakil Rektor 1 (Akademik) Universitas STEKOM Semarang.



YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-623-8120-81-9 (PDF)

