

BELAJAR PEMROGRAMAN WEB DASAR

HTML, CSS & JAVA SCRIPT
Untuk Pemula

CSS



HTML



JS



Dendy Kurniawan, S.Kom, M.Kom



YAYASAN PRIMA AGUS TEKNIK

BELAJAR PEMROGRAMAN WEB DASAR

HTML, CSS & JAVA SCRIPT
Untuk Pemula

Dendy Kurniawan, S.Kom, M.Kom



YAYASAN PRIMA AGUS TEKNIK

PENERBIT :

YAYASAN PRIMA AGUS TEKNIK

JL. Majapahit No. 605 Semarang

Telp. (024) 6723456. Fax. 024-6710144

Email : penerbit_ypat@stekom.ac.id

Belajar Pemrograman Web Dasar
HTML, CSS & Java Script
Untuk Pemula

Penulis :

Dendy Kurniawan, S.Kom., M.Kom

ISBN : 978-623-8120-28-4 (PDF)

Editor :

Muhammad Sholikan, M.Kom

Penyunting :

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

Desain Sampul Dan Tata Letak :

Agus Priyadi, S.Ds., M.Kom.

Penebit :

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

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

Daftar Isi

Kata Pengantar	v
Daftar Isi	vii
BAB 1	1
1.1 WEBSITE	2
1.2 APA ITU HTML ?.....	3
1.3 SEJARAH SINGKAT HTML.....	3
1.4 HTML dan HTML5.....	4
1.5 CASCADING STYLE SHEETS (CSS)	5
1.6 JAVA SCRIPT (JS)	5
1.7 ANALOGI PERBEDAAN HTML CSS DAN JS.....	6
1.8 TEXT EDITOR.....	6
1.9 WEB BROWSER.....	8
1.10 STRUKTUR DASAR HTML	9
1.11 DEFINISI TAG	11
1.12 DEFINISI ELEMENT.....	12
1.13 DEFINISI ATRIBUT	14
BAB 2	15
2.1 HEADING	16
2.2 PARAGRAF	17
2.3 LIST HTML.....	21
2.4 HTML IMAGES	23
BAB 3	26
3.1 HYPERLINK HTML	27
3.2 FORM.....	30
BAB 4	39
4.1 TABEL.....	40
4.2 FRAME	45
BAB 5	54
5.1 OVERVIEW CSS	55
5.2 PENEMPATAN CSS.....	58

5.3	MEMULAI CSS.....	61
BAB 6	67
6.1	MENGATUR FONT DENGAN CSS	68
6.2	FORMAT TEXT DENGAN CSS.....	70
6.3	MODIFIKASI LIST DENGAN CSS.....	74
6.4	MEMODIFIKASI BORDER DENGAN CSS	76
BAB 7	83
7.1	MARGIN DAN PADDING DENGAN CSS	84
7.2	MENGATUR POSITION DENGAN CSS.....	86
7.3	TEKNIK FLOAT DENGAN CSS.....	90
7.4	DESAIN TABEL DENGAN CSS.....	91
BAB 8	97
8.1	MEMBERIKAN EFEK PADA GAMBAR DENGAN CSS	98
8.2	MEMBUAT TEMPLATE WEB	104
BAB 9	109
9.1	MENGISI KONTEN.....	110
9.2	MEMBUAT HALAMAN MENJADI RESPONSIVE	114
BAB 10	118
10.1	INTRODUCTION	119
10.2	CARA MEMBUAT JAVASCRIPT DI WEBSITE	121
BAB 11	136
11.1	VARIABEL.....	137
11.2	TIPE DATA.....	140
11.3	OPERATOR.....	141
BAB 12	153
12.1	PERCABANGAN.....	154
12.2	PERULANGAN	164
BAB 13	177
13.1	STRUKTUR DATA ARRAY.....	178
13.2	DOM (Document Object Model).....	188
BAB 14	200
14.1	MEMBUAT ANIMASI TRANSISI	201
BAB 15	214

15.1	PENGANTAR WEB ONLINE	215
15.2	MENGONLINEKAN WEB	215
	Penutup.....	227
	DAFTAR PUSTAKA.....	228

BAB 1

PENGENALAN HTML

TUJUAN :

- a. Mahasiswa mengetahui perbedaan Web Client dengan Web Server**
- b. Mahasiswa Mengetahui Pengertian dan Sejarah HTML**
- c. Mahasiswa Mengetahui Text Editor Pemrograman Web**
- d. Mahasiswa Mengetahui Web Browser**
- e. Mahasiswa Mengetahui Perbedaan HTML, CSS dan Java Script**
- f. Mahasiswa mengetahui struktur HTML, Tag, element dan Atribut dari HTML**
- g. Mahasiswa mampu membuat document HTML sederhana**

Overview :

Aplikasi web di buat dengan menggunakan tag HTML. Pada sesi BAB ini akan di bahas beberapa hal mendasar atau istilah yang berkaitan dnegan pemrograman web, sampai membuat file web beserta penulisan awal atau struktur dasar dari web tersebut yang sudah memuat Tag, element serta atribut.

1.1 WEBSITE

Website merupakan bagian yang tidak terpisahkan dari perkembangan internet, jumlah pengguna sampai saat ini mencapai lebih dari 1,9 miliar di seluruh dunia (internetlivestats, 2022). Kumlah tersebut di perkirakan akan terus mengalami peningkatan melihat saat ini website juga terus berkembang dan semakin bagus. Tujuan penggunaan website awalnya hanya digunakan untuk kalangan pribadi, namun sekarang hampir seluruh perusahaan, instansi, bahkan usaha kuliner juga memiliki website untuk media informasi mereka.

Website adalah kumpulan halaman yang berisi suatu informasi tertentu yang kemudian bisa di akses oleh siapapun, dimanapun dan kapanpun dengan mudah melalui koneksi internet. Proses pembuatan website salah satunya dapat dilakukan dengan pemrograman web atau menuliskan sebuah intruksi-intruksi yang di berikan kepada komputer untuk membuat suatu tugas atau fungsi tertentu. Dengan kata lain Pemrograman Web merupakan cara atau Proses untuk menjalankan intruksi ataupun perintah kepada komputer yang terhubung dengan internet yang dapat digunakan untuk membuat tugas maupun fungsi lainnya. Kemudian ketika menjalankan sebuah program tersebut di dalam web dapat melalui web browser misalnya: Opera, mozilla, Chrome, dll.

a) Jenis - jenis Website

- Website Statis

Website Statis merupakan web yang isi atau kontennya tidak pernah di perbaharui atau di update dalam durasi waktu tertentu, atau bisa di katakana isinya tidak pernah berubah sama sekali, contoh website statis seperti web company profile suatu instansi atau perusahaan dll. Tujuan web ini yaitu user atau masyarakat dapat melihat tentang profil dari instansi atau perusahaan tersebut. Website statis ini cenderung tidak responsive, hal ini karena ketika akan mengupdate fitur dari webnya harus mengubah scriptnya atau perintah program secara langsung, tidak dapat dilakukan melalui CMS yang tersedia.

- Website Dinamis

Situs web yang dinamis adalah situs web yang konten atau elemen lainnya sering berubah karena pengguna atau pengunjung lain dapat memperbarui informasi situs secara diam-diam. Website dinamis ini biasanya cocok di gunakan untuk web blog, jejaring sosial, toko online dll. Pembuatan website ini membutuhkan waktu yang cukup lama karena menggunakan database untuk menyimpan semua data dari website yang bersangkutan.

- Website Interaktif

Website Interaktif adalah website yang hampir mirip dengan website yang bersifat statis; namun, interaksi di situs web interaktif lebih sering atau sering terjadi karena ditujukan untuk pengguna yang juga akan menggunakannya untuk mengedit konten di situs web tersebut. Tidak seperti website dinamis yang pilihannya hanya share atau tinggalkan komentar. Seorang pengguna atau pengguna juga dapat melakukan tindakan saat mengedit konten di situs web interaktif seperti YouTube.

b) Development Web

Dalam development web, terdapat istilah yang dinamakan Client - side serta Server - side Scripting. Dua istilah tersebut digunakan untuk mengelompokkan dimana bahasa atau kode

pemrograman tersebut dijalankan, apakah berjalan di sisi client atau di sisi server. Jika diartikan dalam arti lain, Client-side memiliki arti sisi pengguna dan Server-side memiliki arti sisi server atau penyedia. Berikut penjelasan singkat mengenai web client dan web server.

- **Client Side Scripting**

Klien web, sering dikenal sebagai skrip sisi klien, adalah bagian dari bahasa pemrograman web di mana pemrosesan sintaksis terjadi di sisi klien. Dalam proses klien web, browser web berfungsi sebagai klien. Namun, ada juga perpustakaan yang tersedia di aplikasi browser web yang dapat digunakan untuk menulis ulang kode atau struktur sintaksis apa pun yang ditemukan di halaman web yang menggunakan skrip sisi klien. Akibatnya, dapat dikatakan bahwa skrip sisi klien mengacu pada aktivitas apa pun yang dilakukan di sisi pengguna atau klien (komputer pengguna). Contoh skrip sisi klien meliputi, antara lain, XHTML, HTML, JavaScript, CSS, JQUERY, dan XML.

- **Server Side Scripting**

Yang dimaksud dengan “server-side scripting” adalah bahasa pemrograman web yang proses pengembangannya dilakukan di sisi server, yaitu web server yang terintegrasi dengan web engine. Tujuan dari mesin pencari online adalah untuk meng-googole setiap kemungkinan frase nanti. Mesin web adalah komponen server web yang perlu diinstal di komputer sesegera mungkin.

Ada skrip sisi server di situs web dinamis. Dalam skrip sisi server, kecepatan dan efisiensi komputer server adalah faktor penentu seberapa cepat situs web memuat atau beroperasi. Karena pada dasarnya bersifat rahasia, sintaks tertulis di sisi server tidak dapat dilihat di antarmuka klien atau pengguna. PHP, Active Server Pages (ASP), Lasso, Java Server Pages (JSP), dan SSI adalah contoh program skrip sisi server.

1.2 APA ITU HTML ?

HTML, sering dikenal sebagai Hyper-Text Markup Language, adalah bahasa utama yang digunakan di seluruh dunia oleh situs web. Hampir semua website yang tersedia di internet dibuat dengan menggunakan variasi HTML. Pengembang dapat menambahkan teks, multimedia, dan tautan ke browser yang kompatibel dengan atribut atau elemen lain dengan menggunakan HTML. Dari elemen yang dapat terhubung ke dokumen Anda (hiperteks) hingga elemen yang membuat dokumen Anda lebih interaktif, semuanya terdapat dalam satu file HTML.

Dengan kata lain, HTML (Hypertext Markup Language) adalah bahasa yang biasa digunakan untuk menandai dokumen yang akan dipublikasikan secara online. Dalam bahasa sederhana, HTML mengilustrasikan struktur halaman Web, yang terdiri dari serangkaian elemen HTML, untuk menjelaskan kepada browser web cara menampilkan konten yang telah dipilih.

HTML memiliki standar yang dibuat oleh W3C (World Wide Web Consortium) pada tahun 1997 dan semakin lama semakin bermanfaat. Dalam HTML, ada tag yang digunakan untuk menunjukkan struktur. Tag dan elemen dapat dilindungi dengan menggunakan karakter < dan >.

1.3 SEJARAH SINGKAT HTML

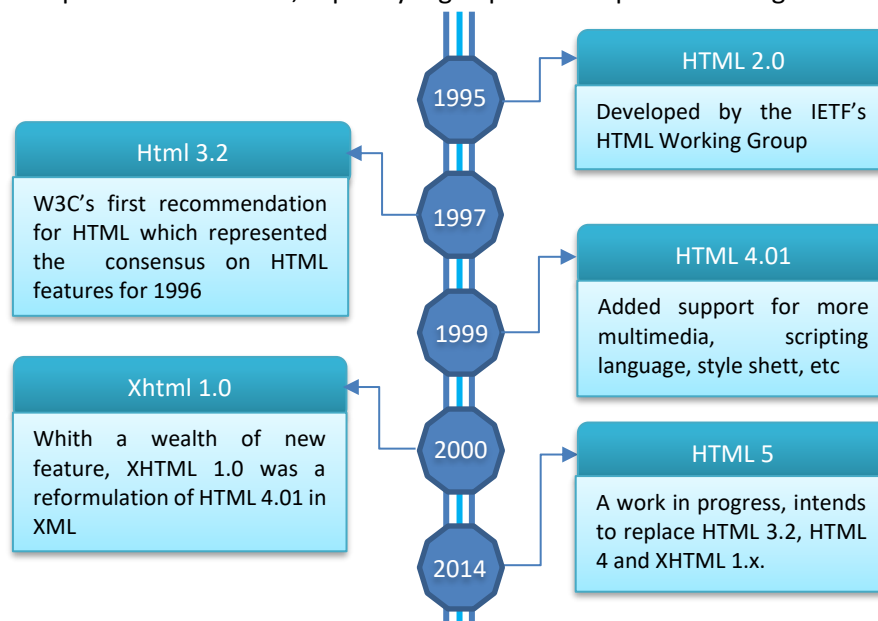
- Awal mula kemunculan HTML yaitu dimulai dari seorang ilmuwan yang bernama Tim Berners-Lee yang bekerja di CERN.

- CERN bukanlah bisnis yang berhubungan dengan teknologi atau bahkan internet. CERN adalah singkatan dari Dewan Riset Nuklir Eropa, atau singkatnya CERN. Ini mengacu pada Komisi Eropa untuk Fisika Nuklir.



Gambar 1.1 Tim Berners-Lee

- Sejak awal, HTML telah mengalami banyak bug, dan versi HTML terbaru saat tulisan ini dibuat adalah HTML5. Dalam hal bahasa markup, HTML5 telah memimpin akhir-akhir ini dan sudah menyertakan beberapa fitur yang ada di versi HTML sebelumnya, serta fitur yang dihilangkan tetapi ternyata ada di versi XHTML sebelumnya. Meskipun tidak ada pembaruan versi khusus untuk setiap rilis baru, fitur baru ditambahkan ke HTML5 hampir setiap hari. Baik audio maupun video pada dasarnya berbeda antara HTML dan HTML5, tetapi di setiap fitur HTML, masing-masing adalah produk dari HTML5, seperti yang dapat dilihat pada contoh gambar di bawah ini.

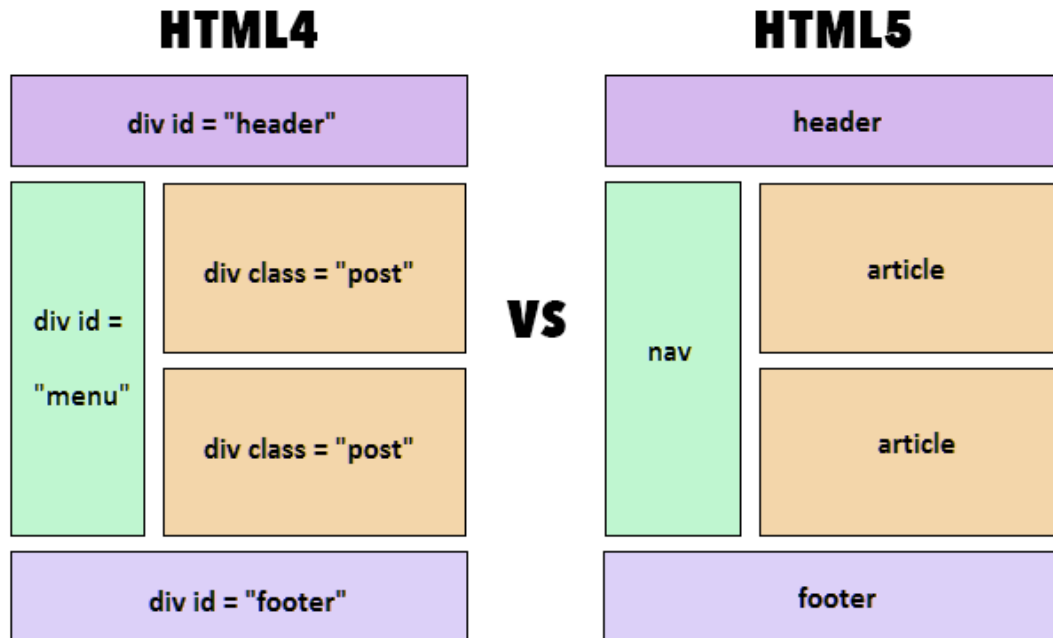


Gambar 1.2 Perubahan Versi HTML

1.4 HTML dan HTML5

- HTML5 mendukung SVG, canvas, dan grafik vektor lainnya, sedangkan dalam HTML, grafik vektor hanya dapat digunakan dengan teknologi lain seperti Flash, VML (Vector Markup Language), Silver-light, dan lain-lain. SVG (Scalable Vector Graphic) adalah format gambar yang menggunakan XML (eXtensible Markup Language) sebagai dasar pembuatan gambar vektor dua dimensi. HTML5 menggunakan database web SQL, sedangkan HTML hanya memiliki cache browser yang dapat digunakan untuk tujuan ini. HTML5 juga menggunakan aplikasi untuk caching data.

- Perbedaan lain antara HTML dan HTML5 adalah bahwa HTML5 tidak mencegah JavaScript berjalan di browser itu sendiri (berjalan di atas terpisah di bagian luar browser), sedangkan HTML5 melakukannya.
- HTML5 mendukung kontrol formulir baru, termasuk waktu dan tanggal, email, nomor telepon, jangkauan, telegraf, url, pencarian, dan lain-lain.



Gambar 1.3 Perbandingan HTML 4 dengan HTML 5

1.5 CASCADING STYLE SHEETS (CSS)

Singkatan CSS adalah singkatan dari "Cascading Style Sheets". Sesuai dengan namanya, CSS memiliki pengertian "style sheet language" dalam bahasa pemrograman yang digunakan untuk pengembangan web. CSS adalah bahasa pemrograman yang digunakan untuk membuat website header. CSS menggunakan id dan penanda yang dapat dikenali kelas saat mendesain header situs web.

Di bidang pemrograman dan teknologi yang berkembang pesat, CSS tidak lagi hanya digunakan dengan HTML dan XHTML, namun sudah bisa digunakan untuk mengembangkan tampilan aplikasi Android. CSS dijalankan menggunakan id atau kelas.

Fungsi CSS

- Mempertahankan variasi tampilan
- Membuat website tampil lebih rapi di berbagai ukuran layar
- Mempercepat Loading Halaman Web
- Memudahkan pengelolaan kode

1.6 JAVA SCRIPT (JS)

Javascript adalah satu-satunya bahasa web yang dapat membuat beranda situs web Anda lebih menarik dan dinamis. Eksekusi bahasa Javascript terjadi di ujung pengguna situs web atau lokasi lain daripada di server, yang berbeda dari PHP. Javascript digambarkan berbeda dari HTML dan CSS. HTML digunakan untuk mengubah format tabular konten, dan CSS mengubah tata letak, oleh karena itu desain keseluruhannya sangat berbeda. Ada beberapa contoh orang yang menggunakan HTML, CSS,

dan javascript secara bersamaan, termasuk konstruksi rumah , dekorasi interior , dan lampu serta pintu - pintu.

Akibatnya , HTML digambarkan sebagai sebuah blok bangunan yang dapat digunakan untuk merenungkan segala sesuatu, bahkan tempat - tempat dengan alamat tetap seperti tembok, pintu, atap, dan lain - lain. Sebaliknya, CSS adalah alat dekorasi rumah yang dapat mengubah tampilan wallpaper, karpet , dan hal - hal lain di rumah Anda .Javascript berguna untuk hal - hal yang perlu responsif dan menghidupkan , seperti menyalakan lampu atau membuat pintu.

Javascript makes websites more responsive since, even in the absence of a problem, their contents will still change significantly. You can find your house number. The absence of a light or a pintu may not be a problem, but the large house may become dim and begin to leak a little bit of moisture. Selain itu, ketiadaan javascript membuat website menjadi kurang menarik.

1.7 ANALOGI PERBEDAAN HTML CSS DAN JS

Agar lebih mudah memahami hubungan antara HTML CSS dan Java Script, silahkan lihat gambar 1.4 di bawah ini.



Gambar 1.4 Analogi Perbedaan HTML CSS dan Java Script

Dari gambar di atas dapat di pahami jika HTML di analogikan sebagai badan dari seseorang manusia tanpa pakaian. Karena HTML merupakan kerangka dasar yang digunakan untuk membangun sebuah web.

Sedangkan CSS dianalogikan sebagai pakaian seperti baju, celana, dasi, ikat pinggang dan lain sebagainya. Yang di maksud dalam analogi ini adalah agar membuat tubuh manusia terlihat lebih cantik atau keren dengan berbagai macam jenis pakaian yang ada seperti di dunia nyata ini. Sehingga dapat membuat tampilan website menjadi indah.

Java script dianalogikan sebagai otak manusia, hal ini di maksud karena otak manusia memiliki peran inti dalam menggerakkan atau menjalankan tindakan dari anggota tubuh manusia menjadi interaktif. Manusia dapat berbicara sesuai topik yang di bahas, dapat bergerak dan lain sebagainya itu didasari dari perintah yang ada dalam otak. Jadi dengan menggunakan java script website akan menjadi interaktif, dapat di berikan banyak animasi dan juga lebih user friendly.

1.8 TEXT EDITOR

Editor teks adalah perangkat lunak atau program komputer yang memungkinkan pengguna, dalam hal ini Anda, untuk membuat, mengedit, atau memodifikasi file teks biasa. Editor teks ini secara alami dapat digunakan untuk membuat program komputer dan mengekstrak kode sumber dari bahasa pemrograman. Selain itu, editor teks dapat digunakan untuk membuat situs web, templat situs web, dan aplikasi terkait lainnya.

Beberapa aplikasi text editor diantaranya :

- a) NotePad / NotePad ++



Gambar 1.5 Icon Notepad dan Notepad ++

Microsoft Notepad merupakan software pemrosesan kata yang disediakan oleh Windows. Sehingga software ini sudah tersedia jika anda menggunakan system operasi windows. Notepad dapat digunakan untuk membuat file tipe log ataupun segala bentuk file termasuk HTML, CSS dan Java Script.

Namun demikian notepad bawaan windows memiliki belum memiliki fitur – fitur dalam mendukung penulisan perintah – perintah pemrograman. Sehingga dikembangkanlah aplikasi tersebut yang terdapat fitur – fitur dalam mempermudah dalam digunakan untuk menulis intruksi – intruksi Bahasa pemrograman, aplikasi tersebut dinamakan notepad plus – plus yang bisa di unduh pada <https://notepad-plus-plus.org/downloads/>.

- b) Adobe Dreamweaver



Gambar 1.6 Icon Adobe Dreamweaver

Program editor web yang dikenal dengan nama Adobe Dreamweaver dikembangkan oleh Adobe Systems setelah sebelumnya dikenal dengan nama Macromedia Dreamweaver oleh Macromedia. Karena fiturnya yang mudah digunakan dan digunakan secara luas oleh pengembang web, aplikasi ini sering digunakan. Versi 8 Macromedia Dreamweaver dirilis sebelum Adobe Systems mengakuisisi Macromedia. Versi terbaru Dreamweaver dari Adobe Systems adalah versi 12, yang tersedia di Adobe Creative Cloud (sering disingkat Adobe Cc). Versi gratis Aalikasi tersedia untuk diunduh di https://www.adobe.com/id_id/products/dreamweaver/free-trial-download.html

- c) Sublime Text



Gambar 1.7 Icon Sublime Text

Perangkat lunak pengedit teks yang disebut Sublime Text sering digunakan oleh programmer, terutama pengembang web. Aplikasi ini sangat penting untuk pengembang web saat membangun aplikasi apa pun. Banyak para proqramer menggunakan aplikasi sublime text ini untuk

menyempurnakan situs web yang mereka buat. Aplikasi ini memberikan warna dari tulisan koding yang bervariasi, interaktif, multiple tabs, tab multi select, dan tentunya sangat menarik. Anda juga dapat mengubah warna tema pada sublime, selain itu variasi warna pada coding membuat program anda menjadi lebih beragam dan interaktif, sehingga membuat anda untuk lebih santai dalam proses coding.

Text editor ini berfungsi dalam menulis sejumlah code script serta dapat mendukung berbagai macam jenis file. Selain itu, sublime Text juga mendukung berbagai macam programming languages diantaranya CSS, C, C++, C#, html, ASP dan lain sebagainya. Python API pada sublime telah diperbarui ke Python 3.8. Aplikasi ini dapat di unduh pada <https://www.sublimetext.com/download>

d) Visual Studio Code



Gambar 1.8 Icon Visual Studio Code

Visual Code Studio adalah editor kode gratis yang dapat digunakan pada sistem operasi desktop berbasis Windows, Linux, dan Macintosh. Kode untuk editor ini dibuat oleh Microsoft, penyedia teknologi terdepan di dunia. Kode Visual adalah editor perangkat lunak yang tangguh, namun terkadang mengalami kegagalan fungsi saat digunakan. Dia mampu membuat dan menulis kode ringkasan dalam berbagai bahasa pemrograman. Yakni, JavaScript, TypeScript, dan Node.js.

Juga kompatibel dengan bahasa lain dan lingkungan runtime, seperti PHP, Python, Java, and.NET, adalah Visual Code Studio. Ini terkait dengan ekonomi yang kuat dan perluasan pengetahuan yang mengancam. Anda bisa mendapatkan Visual Studio Code dari situs ini :

<https://code.visualstudio.com/download>

Selain itu masih banyak lagi aplikasi text editor lainnya. Dalam praktikum pemrograman web 1 ini nantinya bebas ingin menggunakan aplikasi text editor yang mana saja karena pada dasarnya perbedaan text editor hanya terletak pada vasilitas yang di milikinya. Intinya anda harus menyimpan file tersebut dengan ekstensi penyimpanan file tersebut.

1.9 WEB BROWSER

Tujuan dari web browser, juga dikenal sebagai web browser, web browser, atau penelusur web (Bahasa Inggris: web browser), adalah untuk mengambil dan mengirimkan informasi dari Internet. Sepotong informasi diidentifikasi dengan menggunakan metode untuk mengenalinya sebagai informasi, yang dapat berupa situs web, gambar, video, atau jenis konten lainnya. Browser Web, biasa disebut sebagai browser, adalah alat yang digunakan untuk mengakses, menyimpan, dan mengirimkan informasi di internet. Dan satu-satunya format yang digunakan untuk informasi adalah HTML.

Kode HTML yang dibuat akan dievaluasi oleh browser web untuk memastikan bahwa kode tersebut ditampilkan sebagaimana mestinya. Menurut teorinya, setiap browser web dapat menampilkan kode HTML dengan benar, namun jika Anda terbiasa dengan desain dokumen, setiap browser memiliki beberapa perbedaan. Google Chrome, Firefox, Internet Explorer, Opera, dan Safari adalah beberapa browser online yang umum.

HTML dibuat dan diadopsi oleh W3C, sebuah organisasi standar global dengan fokus pada web (World Wide Web Consortium). Hal ini disebabkan oleh fakta bahwa setiap perangkat lunak browser web menggunakan standar pengkodean HTML yang berbeda, sehingga diperlukan pembuatan standar yang seragam untuk semua browser. Namun, dalam praktiknya, standar ini hanya berfungsi sebagai saran. Beberapa browser web dapat membuat gambar sendiri.

Berikut adalah referensi web browser yang bisa di gunakan

a) Mozilla Firefox



Gambar 1.9 Icon Mozilla Firefox

Dapat didownload pada <https://www.mozilla.org/id/firefox/new/>

b) Google Chrome



Gambar 1.10 Icon Google Chrome

Dapat didownload pada https://www.google.com/intl/id_id/chrome/

c) Microsoft Edge



Gambar 1.11 Icon Microsoft Edge

Dapat didownload pada <https://www.microsoft.com/id-id/edge/home?form=MA13FJ>

1.10 STRUKTUR DASAR HTML

Untuk membuat dokumen HTML standar, gunakan tag html, head, dan body serta tag yang berjarak sama. Sebagai contoh, dokumen HTML berikut dihasilkan melalui sistem tag-by-tag:

```

<!doctype html>
<html>
  <head>
    <title> ... </title>
  </head>
  <body>

  </body>
</html>

```

Gambar 1.12 Struktur Dasar HTML

- Semua dokumen HTML harus diawali dengan deklarasi "<!DOCTYPE html>."
- Dokumen HTML itu sendiri diawali dengan <html> dan diakhiri dengan </html>.
- Kode HTML yang dapat dilihat di web browser berada di antara tag "<body>" dan "</body>".

Contoh :

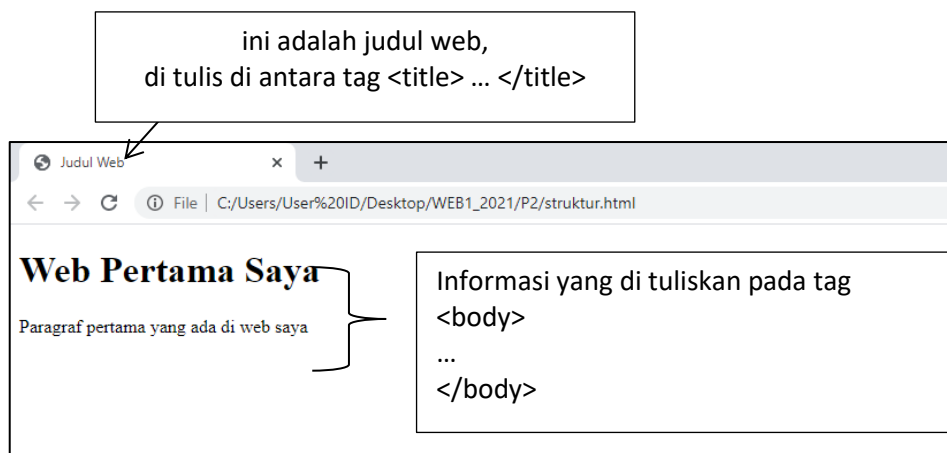
```

struktur.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Judul Web</title>
5 </head>
6 <body>
7
8 <h1>Web Pertama Saya</h1>
9 <p>Paragraf pertama yang ada di web saya</p>
10
11 </body>
12 </html>

```

Gambar 1.13 Contoh Perintah HTML

Kemudian simpan file dengan ekstensi .html dan bukalah file tersebut dengan web browser.



Gambar 1.14 Contoh Hasil

Deklarasi <! DOCTYPE>

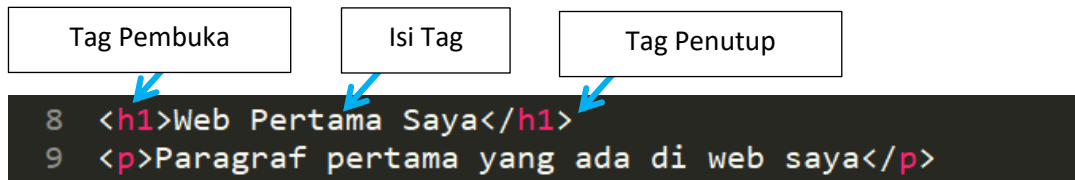
- <!DOCTYPE> deklarasi ini akan menjelaskan jenis dokumen untuk membantu browser supaya tampilan halaman web dapat di tampilkan dengan benar. Tag ini cukup di tuliskan sekali saja, pada bagian paling atas halaman (sebelum tag HTML).
- <!DOCTYPE> deklarasi ini tidak case sensitive.
- <!DOCTYPE> deklarasi HTML5 adalah: `<!DOCTYPE html>`

1.11 DEFINISI TAG

Tag adalah nama yang diberikan pada penanda awalan dan eksekusi elemen HTML saat ini. Tag dibuat dengan tanda kurung siku (<...>), kemudian diberi nama dan kadang-kadang disertai atribut.

Konteks: "<p>", "<a>", "<body>", "<head>", dan istilah sejenis lainnya.

Tag secara konsisten menggunakan kata "pasangan". Terdapat tag untuk pembuka dan penutup. Namun ada juga beberapa tag yang memiliki pasangan penutup yang tidak beraturan. Tag ditulis dengan jelas dengan menggunakan karakter garis miring (/) di awal nama tag.



Gambar 1.15 Contoh Tag Pembuka Isi dan Penutup

Setiap tag memiliki fungsi yang berbeda. Ada alat yang tersedia untuk membuat judul, tautologi, paragraf, judul, dan elemen lainnya.

Tag yang sering di gunakan dalam HTML

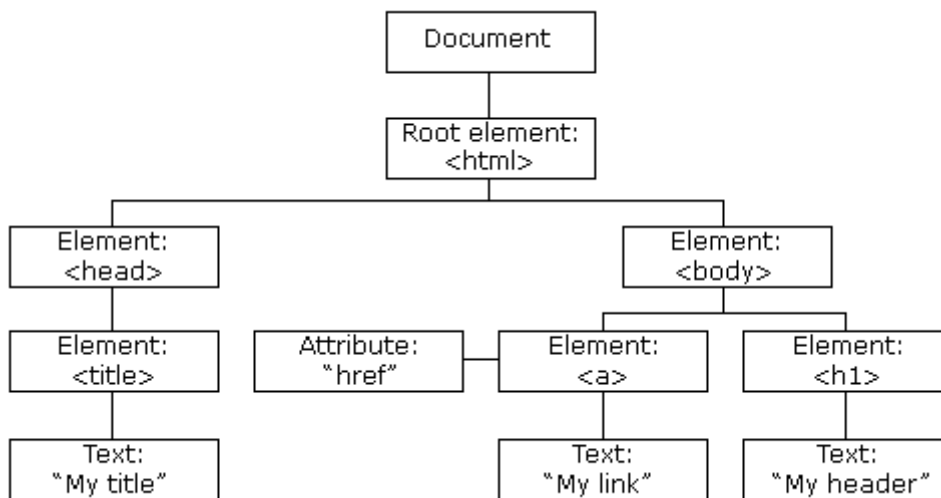
Tabel 1.1 Tag dan Fungsinya

Tag	Fungsi
<!---->	Digunakan untuk memberi sebuah komentar atau keterangan
<!DOCTYPE html>	digunakan untuk Mendefinisikan tipe document
<a>	Mendefinisikan sebuah anchor, digunakan untuk saling menautkan antara satu dokumen HTML ke dokumen HTML yang lain
	membuat teks menjadi tebal
<p>	membuat paragraf
<h1>	membuat heading satu
<h2>	membuat heading dua

<body>	mendefinisikan body/isi dokument html
<head>	mendefinisikan bagian kepala dokumen html
<title>	memdefinisikan judul halaman
<div>	mendefinisikan halaman
<link>	mendefinisikan hubungan antar dokumen
<script>	Mendefinisikan client-side script
<table>	mendefinisikan tabel
<th>	Mendefinisikan sel header di dalam sebuah tabel
<td>	Mendefinisikan sel di dalam sebuah tabel
<tr>	Membuat baris di dalam sebuah tabel
	Mendefinisikan daftar dalam format bullet
	mendefinisikan list

1.12 DEFINISI ELEMEN

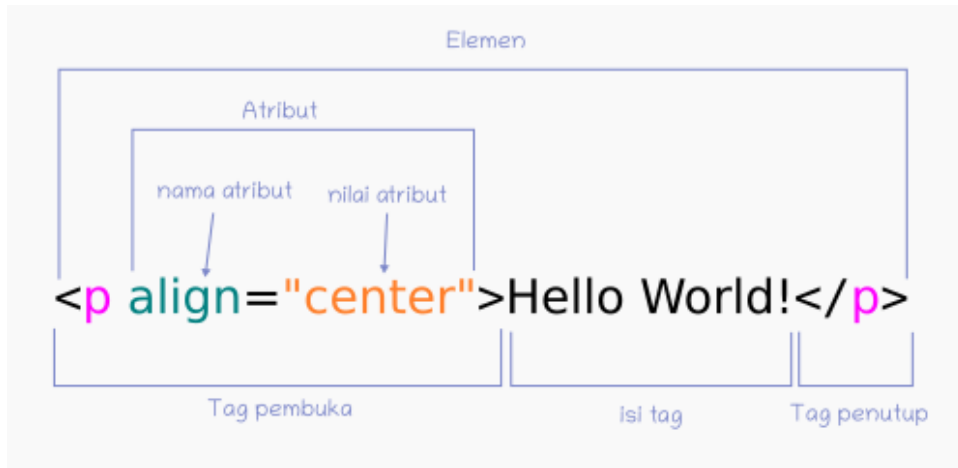
Elemen HTML adalah komponen yang mendukung dokumen HTML. Elemen juga disebut sebagai simpul karena merupakan satu-satunya jenis simpul dalam diagram HTML yang dapat digunakan untuk menyisipkan kode HTML dalam teks.



Gambar 1.16 Element dalam HTML

Ada tiga jenis simpul pada diagram yang dimaksud, elemen Node, atribut Node, dan teks Node. Elemen terdiri dari penutup, isi, dan tag pembuka. Kadang juga menyebutkan beberapa atribut.

Contoh:



Gambar 1.17 Element Tag

- Pada contoh di atas, hanya ada satu elemen dengan tag <p> yang memiliki atribut align="center" dan bertuliskan Hello World!
- Elemen tidak hanya berisi teks, terkadang juga mengandung jenis elemen lainnya. biasanya ini disebut sebagai elemen bersarang atau elemen di dalam elemen.

Element <head>

Selain berisi element title, Element ini berisi informasi meta tentang halaman HTML. Metadata adalah data tentang dokumen HTML. Metadata umumnya akan menentukan judul dari dokumen, kumpulan karakter, skrip, gaya dan informasi meta lainnya serta Metadata tidak akan ditampilkan di halaman, tetapi dapat diurai oleh mesin.

Elemen meta saat ini ditujukan untuk "program robot" seperti mesin telusur daripada pengguna atau orang yang mengunjungi web (mesin pencari). <meta charset="utf-8">? Menurut definisi resmi, charset adalah sekumpulan instruksi yang digunakan untuk mengubah bit-bit dalam HTML menjadi karakter. Akibatnya, rangkaian karakter UTF-8 memberikan petunjuk ke browser web tentang cara mengonversi karakter UTF-8 rangkaian karakter HTML.

Elemen yang bisa masuk ke dalam elemen <head> adalah : <title> , <style>, <base>, <link>, <meta>, <script> <noscript>

```

Simple HTML Document
<head>
<meta charset="UTF-8"> // menentukan karakter yang digunakan
<meta name="description" content="Free Web tutorials">
<meta name="keywords" content="HTML, CSS, JavaScript">
<meta name="author" content="John Doe">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>

```

Gambar 1.18 Element di dalam Head

1.13 DEFINISI ATRIBUT

Kata tertentu yang ada di badan tag disebut atribut. Selain itu, atribut digambarkan sebagai pengubah yang akan menunjukkan persyaratan elemen. Atribut dapat diterapkan ke elemen apa pun. Ada elemen lain yang memerlukan penggunaan atribut, seperti elemen "a", "img", "video", dan "dll".

```
<a href="https://stekom.ac.id/">Klik di sini</a>
```

Gambar 1.19 Atribut Element <a>

- Tag "<a>" adalah tag yang digunakan untuk membuat link. Tag ini harus menggunakan atribut href untuk menampilkan tujuan dari tautan yang rusak.
- Jumlah elemen atribut mungkin lebih besar dari satu.

Contoh:

```

```

Gambar 1.20 Atribut Element

Atribut "src" khusus untuk tag "" yang berfungsi sebagai tempat penampung gambar yang akan ditampilkan. Atribut sebelumnya yang mengubah bentuk objek adalah lebar dan tinggi.

Contoh :

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title></title>
5 </head>
6 <body>
7 <h2 align="center">Pengenalan atribut HTML</h2> ←
8 </body>
9 </html>
```

Gambar 1.21 Atribut src dalam Element

"<h2>" berfungsi sebagai judul tag "h2".

Align adalah nama atributnya,

Center adalah nilai atributnya, dan

Pengenalan atribut HTML adalah elemen dari tag h2.

BAB 2

HEADING, PARAGRAF, LIST & IMAGE

TUJUAN :

- a. Mahasiswa mampu memahami penggunaan Heading
- b. Mahasiswa mampu membuat dokumen HTML dengan melakukan formatting paragraph
- c. Mahasiswa mampu melakukan listing pada dokumen HTML
- d. Mahasiswa dapat menampilkan gambar pada HTML

Overview :

Pada sesi ini akan di pelajari bagaimana fungsi dari tag h1, h2, h3, dst atau di sebut heading, membuat format text dan paragraph seperti membuat text tebal, garis bawah, miring, warna font, ukuran font dan sejenisnya dengan menggunakan fungsi HTML. Juga di bahas cara membuat list dan menampilkan gambar pada HTML.

2.1 HEADING

Heading adalah kumpulan kata-kata yang berfungsi sebagai judul atau subtitle dari dokumen HTML dan tersedia dalam berbagai ukuran yang berbeda. Menurut aturan, tag <Hx> digunakan untuk menunjuk judul apa pun, di mana "x" menunjukkan level apa pun dari 1 hingga 6. Tag Heading, sering dikenal sebagai <Hx>, adalah tag yang mencakup kedua tag untuk awal dan tag untuk akhir.

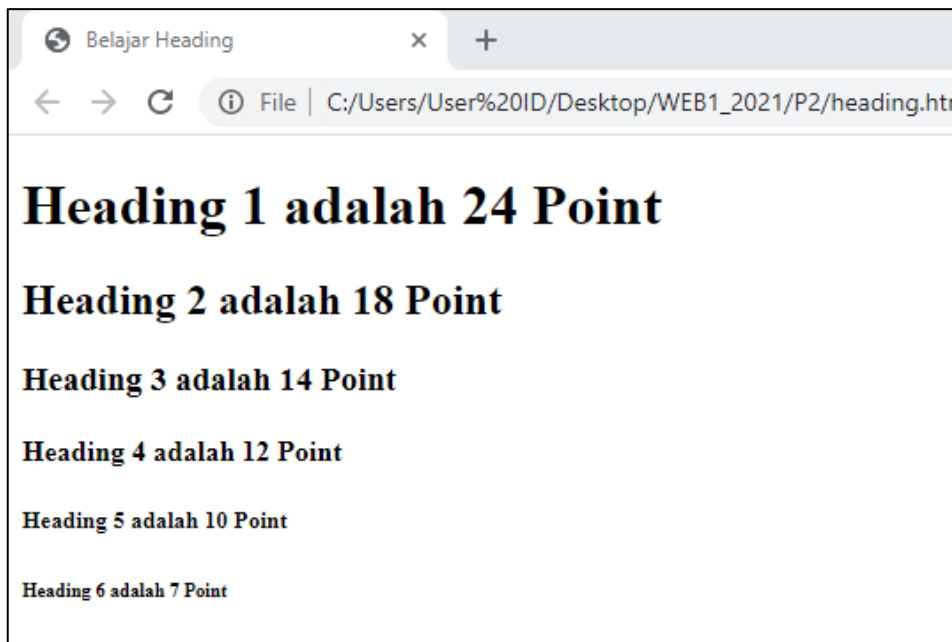
Format : <H1>.....</H1> sampai <H6>.....</H6>

Contoh :

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Belajar Heading</title>
5 </head>
6 <body>
7   <H1>Heading 1 adalah 24 Point</H1>
8   <H2>Heading 2 adalah 18 Point</H2>
9   <H3>Heading 3 adalah 14 Point</H3>
10  <H4>Heading 4 adalah 12 Point</H4>
11  <H5>Heading 5 adalah 10 Point</H5>
12  <H6>Heading 6 adalah 7 Point</H6>
13 </body>
14 </html>
```

Gambar 2.1 Contoh Sintak Heading

Hasilnya :



Gambar 2.2 Hasil Sintak Heading

2.2 PARAGRAF

a) Formatting Paragraf

Jika Anda membuat situs web, Anda juga harus mempertimbangkan untuk menyertakan informasi yang akan diperlukan. HTML juga menyertakan elemen khusus untuk menggunakan markup khusus untuk menentukan teks.

HTML dapat menggunakan tag `<p>` untuk membuat sebuah paragraph yang nantinya juga bisa di berikan atribut sesuai kehendak yang diinginkan.

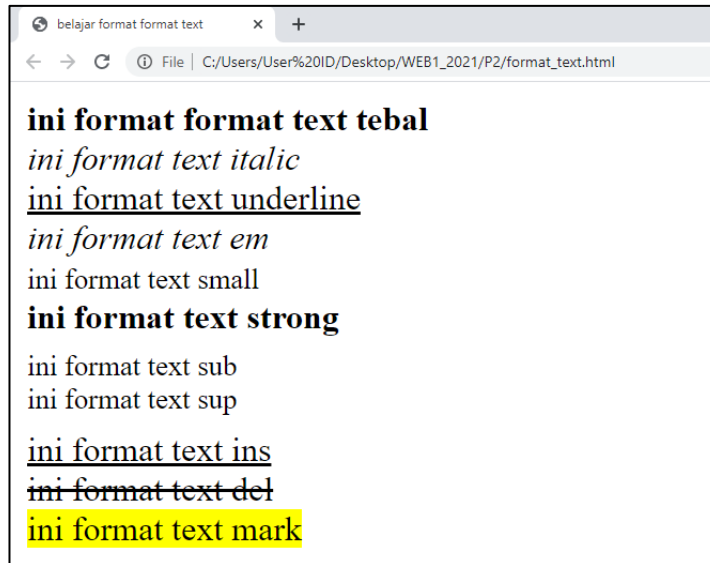
Elemen format yang dapat di gunakan untuk membuat atau menampilkan gaya khusus dari teks :

- Cetak Tebal : tag "``"
- Cetak Penting : tag "``"
- Cetak Miring : tag "`<i>`"
- Garis bawah : tag "`<u>`"
- Menekankan : tag "``"
- Penanda / Menandai : tag "`<mark>`"
- Menjadi Kecil : tag "`<small>`"
- Menghapus : tag "``"
- Memasukan : tag "`<ins>`"
- Pangkah dibawah / Subscript : tag "`<sub>`"
- Pangkat diatas / Superscripts : tag "`<sup>`"

```
format_text.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>belajar format format text</title>
5 </head>
6 <body>
7   <b>ini format format text tebal</b><br/>
8   <i>ini format text italic</i><br/>
9   <u>ini format text underline</u><br/>
10  <em>ini format text em</em><br/>
11  <small>ini format text small</small><br/>
12  <strong>ini format text strong</strong><br/>
13  <sub>ini format text sub</sub><br/>
14  <sup>ini format text sup</sup><br/>
15  <ins>ini format text ins</ins><br/>
16  <del>ini format text del</del><br/>
17  <mark>ini format text mark</mark>
18 </body>
19 </html>
```

Gambar 2.3 Contoh 1 Sintak Formatting Paragraf

Hasilnya dari sintak diatas seperti gambar berikut ini :



Gambar 2.4 Hasil Contoh 1 Sintak Formatting Paragraf

Sampai disini dapat di pahami proses penulisan dan apa masing-masing tag, elemen, dan atribut untuk memahami cara membuat paragraf dalam HTML. Paragraf di bawah ini berfungsi sebagai titik awal untuk membuat paragraf dalam format yang diinginkan. Ada tiga atribut / tag yang dapat digabungkan langsung dengan tag paragraf <p> dalam kalimat ini. Tag atribut yang dimaksud adalah:

- Menjadikan paragraf menjadi rata kiri dapat memakai <p align="left"> ... isi paragraf </p>
- Menjadikan paragraf menjadi rata kanan dapat memakai <p align="right">.. isi paragraf </p>
- Menjadikan paragraf menjadi rata tengah dapat memakai <p align="center">.. isi paragraf </p>
- Menjadikan paragraf rata kiri kanan dapat memakai <p align="justify">.. isi paragraf </p>

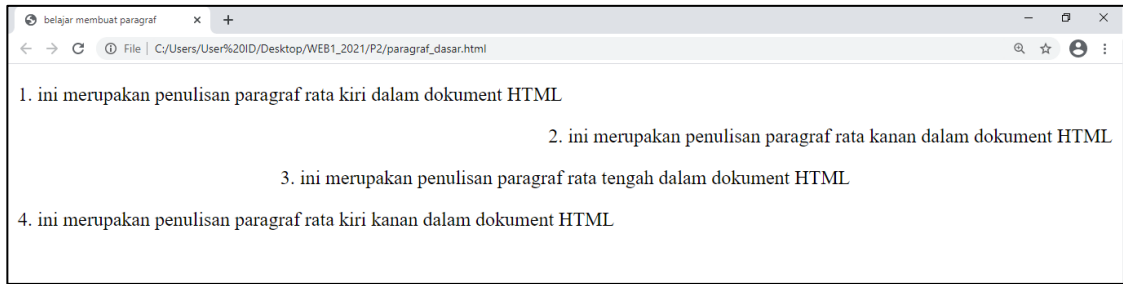
```

paragraf_dasar.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>belajar membuat paragraf</title>
5 </head>
6 <body>
7   <p align="left">1. ini merupakan penulisan paragraf rata kiri dalam
  dokument HTML</p>
8   <p align="right">2. ini merupakan penulisan paragraf rata kanan dalam
  dokument HTML</p>
9   <p align="center">3. ini merupakan penulisan paragraf rata tengah dalam
  dokument HTML</p>
10  <p align="justify">4. ini merupakan penulisan paragraf rata kiri kanan
  dalam dokument HTML</p>
11 </body>
12 </html>

```

Gambar 2.5 Contoh 2 Sintak Formatting Paragraf

Berikut hasil syntac diatas:



Gambar 2.6 Hasil Contoh 2 Sintak Formating Paragraf

b) Line Break

Jika kita terbiasa menekan Enter untuk memasukkan teks, ini tidak akan berfungsi saat menulis dokumen HTML. Mempertimbangkan seberapa sering kami menggunakan tombol Enter, kami tidak dapat mengatakan bahwa ini adalah pertama kalinya.

Untuk membuat baris baru berperilaku serupa dengan tombol Enter keyboard, gunakan tag HTML Line Break. Menggunakan tag `
` untuk jeda baris

```

1 <html>
2   <head>
3     <title> Penggunaan Line Break </title>
4   </head>
5   <body>
6     Menggunakan Line Break <br><br>
7     Dengan menggunakan line break, kita dapat memindahkan kalimat<br>
8     setelah elemen tersebut satu bari dibawahnya.<br>
9     seperti menggunakan ENTER pada word processing.
10  </body>
11 </html>
12

```

Gambar 2.7 Contoh Sintak Line Break

c) Preformatted Text

Hubungan antara setiap kata dalam teks atau kalimat yang relevan dalam dokumen web secara konsisten satu dan sama. Teks yang ditulis dalam bahasa Galia harus diberi tag "`<pre>`" di awal dan "`</pre>`" di akhir agar web browser menampilkan dokumen sesuai dengan alamat web yang dimasukkan.

Dengan tag "`<pre>`", teks yang sudah "diformat sebelumnya" akan ditampilkan di browser dengan ukuran font yang lebih mudah dibaca. Tag ini juga akan menimbulkan spasme, baris baru, dan tap yang sesuai dengan teks asli saat ditampilkan di browser.

```

1 <html>
2   <head>
3     <title> Penggunaan Preformatted </title>
4   </head>
5   <body>
6     <pre>
7       Pengetikan ini dicoba menggunakan Preformatted Text
8
9       Dimana format penulisan ditampilkan dalam web sesua dengan format yang dilakukan
10      Seperti contoh melakukan satu tab seperti ini.
11    </pre>
12  </body>
13 </html>

```

Gambar 2.8 Contoh Sintak Preformatted Text

d) Mengolah Properti Font

Selama proses pengkodean, mungkin ada saatnya Anda perlu melakukan beberapa pekerjaan persiapan, atau bahkan Anda mungkin perlu mencocokkan jenis huruf yang Anda gunakan dengan alasan saat ini, mungkin karena informasi penting yang perlu dipahami lebih baik, dll.

Tag **style="property:value"** dapat digunakan untuk merubah style property dari font. Berikut ini beberapa atribut style yang dapat digunakan untuk elemen styling HTML.

- **"Face** : mengubah jenis font
- **"color** : mengubah warna text
- **"font-family** : mengubah font text
- **"font-size** : mengubah ukuran text
- **"text-align** : mengubah perataan text

```
1 <!doctype html>
2 <html>
3 <head>
4 <title> Property Font </title>
5 </head>
6 <body>
7 <font size="1" face=tahoma color=#FF0000>
8     Tulisan ini dengan ukuran 1, font tahoma dan berwarna merah
9 </font>
10 <br>
11 <font size="2" face=arial color=blue>
12     Tulisan ini dengan ukuran 2, font arial dan berwarna biru
13 </font>
14 </body>
15 </html>
```

Gambar 2.9 Contoh Sintak Properti Font

e) Membuat Garis Horizontal

Gunakan bilah horizontal, sering disebut sebagai "garis horizontal", untuk memperbarui header situs web secara visual. Tag yang digunakan adalah "hr". Atribut tersebut termasuk ALIGN, SIZE, dan WIDTH. Bersama dengan tag br (
), tag hr adalah tag referensi langsung. Oleh karena itu, tag penutup tidak perlu digunakan.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Paragraf</title>
5 </head>
6 <body>
7 <!-- Judul -->
8 <h2>Selamat Datang di Web Saya</h2>
9
10 <!-- contoh tag p -->
11 <p>Ini merupakan pengalaman saya belajar pemrograman web dasar di
12 universitas Sains dan Teknologi Komputer. Ini sangat menyenangkan
13 sekali...
14 Pesan ini saya tulis menggunakan tag p</p>
15
16 <!-- contoh tag <br> -->
17 <br>
18 <!-- contoh tag <pre> -->
19 <pre>
20     Saya akan menjadi sebuah web programmer
21     Saya akan mempelajari pemrograman web dengan tekun
22     Ini merupakan awal yang bagus
```

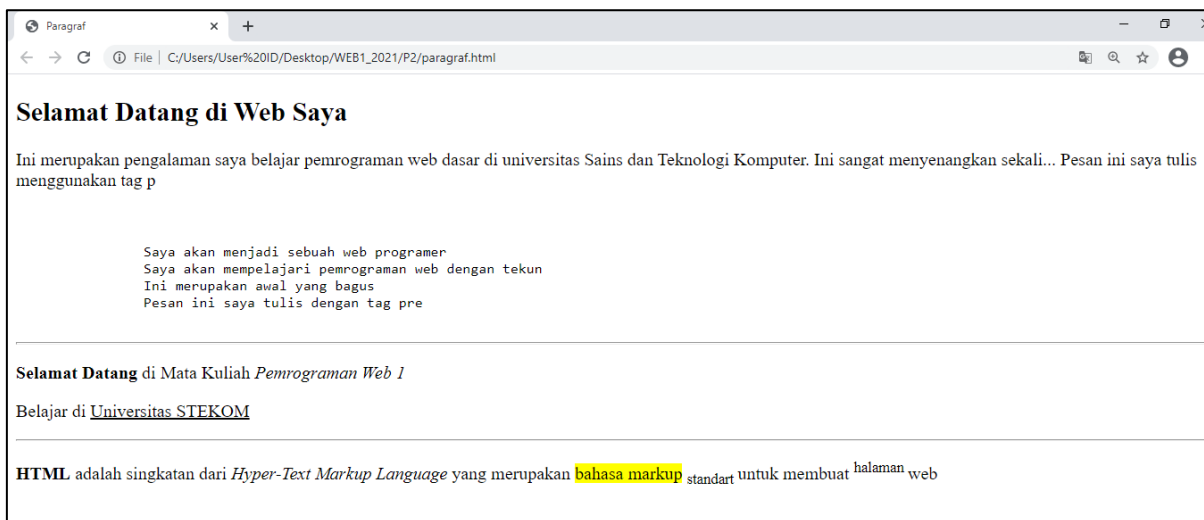
```

22     Pesan ini saya tulis dengan tag pre
23     </pre>
24
25     <!-- membuat garis lurus dengan tag <hr> -->
26     <hr>
27
28     <!-- contoh tag <b>, <i>, <u> -->
29     <p><b>Selamat Datang</b> di Mata Kuliah <i> Pemrograman Web 1</i></p>
30     <p>Belajar di <u>Universitas STEKOM</u></p>
31
32     <!-- membuat baris <hr> -->
33     <hr>
34
35     <!-- contoh tag <strong>, <em>, <sub> dan <sup> -->
36     <p><strong>HTML</strong> adalah singkatan dari <em> Hyper-Text Markup
37     Language</em> yang merupakan <mark> bahasa markup</mark> <sub>standart </
38     sub> untuk membuat <sup> halaman </sup> web </p>
39 </body>
40 </html>

```

Gambar 2.10 Contoh Sintak Membuat Horizontal Line

Hasilnya :



Gambar 2.11 Hasil Contoh Sintak Membuat Garis Horizontal

2.3 LIST HTML

Dalam HTML, terdapat 3 jenis tag list istilah, yaitu berurutan (ordered list) , tidak berurutan (unordered list) dan definition list.

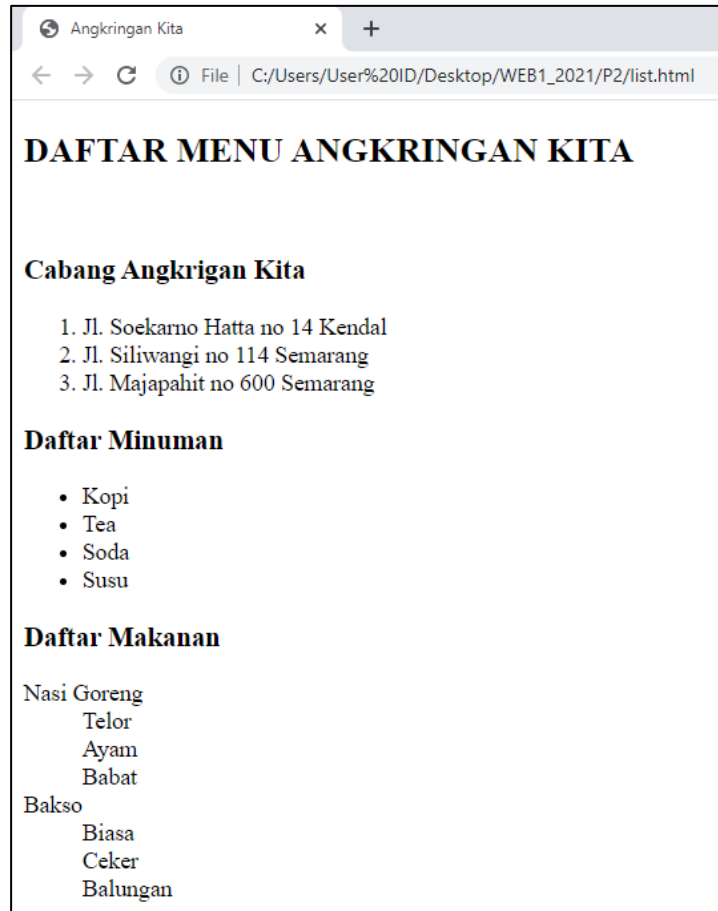
- Ordered list** akan ditampilkan menggunakan huruf atau angka dan menggunakan tag "", sedangkan tag "" akan digunakan untuk daftar itu sendiri. Ordered list mempunyai atribut type untuk mengganti tipe list yaitu 1, A, a, I, i. contoh : `<ol type="i">`
- Unordered list** ditampilkan dengan sebuah lingkaran atau persegi dan menggunakan tag , sedangkan untuk list sendiri menggunakan tag . Unordered list mempunyai atribut type untuk mengganti tipe list yaitu disc, circle, square, none. Contoh : `<ul type="square">`
- Definition list** menampilkan daftar formulir dengan deskripsi masing-masing formulir. Tag "<dl>" dan "<dt>" mendefinisikan deskripti dokumen, "<dt>" mendefinisikan nama institusi, dan "<dd>" mendeskripsikan masing-masing institusi; namun, tag "" digunakan untuk mendefinisikan daftar itu sendiri.

Perhatikan contoh di bawah ini agar lebih mudah memahaminya.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Angkringan Kita</title>
5 </head>
6 <body>
7   <!-- judul konten -->
8   <h2>DAFTAR MENU ANGKRINGAN KITA</h2>
9
10  <!-- enter -->
11  <br>
12
13  <!-- Contoh ordered list Number -->
14  <h3>Cabang Angkringan Kita</h3>
15  <ol>
16    <li>Jl. Soekarno Hatta no 14 Kendal</li>
17    <li>Jl. Siliwangi no 114 Semarang</li>
18    <li>Jl. Majapahit no 600 Semarang</li>
19  </ol>
20
21  <!-- contoh Unordered list Bullets -->
22  <h3>Daftar Minuman</h3>
23  <ul>
24    <li>Kopi</li>
25    <li>Tea</li>
26    <li>Soda</li>
27    <li>Susu</li>
28  </ul>
29
30  <!-- contoh Definition List -->
31  <h3>Daftar Makanan</h3>
32  <dl>
33    <dt>Nasi Goreng</dt>
34    <dd>Telor</dd>
35    <dd>Ayam</dd>
36    <dd>Babat</dd>
37    <dt>Bakso</dt>
38    <dd>Biasa</dd>
39    <dd>Ceker</dd>
40    <dd>Balungan</dd>
41  </dl>
42
43 </body>
44 </html>
```

Gambar 2.12 Contoh Sintak Membuat List

Hasilnya dapat dilihat pada gambar berikut ini:



Gambar 2.13 Hasil Contoh Sintak Membuat List

2.4 HTML IMAGES

Tag `img` digunakan untuk meletakkan gambar dalam dokumen HTML dan terlihat seperti ini: Menurut standar teknis, gambar tidak ditambahkan ke database situs web; sebaliknya, mereka hanya ditambahkan ke database situs web. Tag `img` membuat ruang kosong tempat gambar yang relevan dapat ditampilkan.

Tag `` memiliki beberapa atribut yang di gunakan :

- **Src** : Digunakan untuk memberi tahu browser tempat untuk menemukan gambar
- **Alt** : Memberikan informasi teks alternatif untuk gambar
- **Title** : Memberikan Nama pada gambar pada saat cursor berada di wilayah gambar
- **Width** : Menentukan lebar gambar, ukuran gambar default (px) pixcel dan persen (%)
- **Hight** : Menentukan tinggi gambar, ukuran gambar default (px) pixcel dan persen (%)

Penulisan tag `img` External :

```

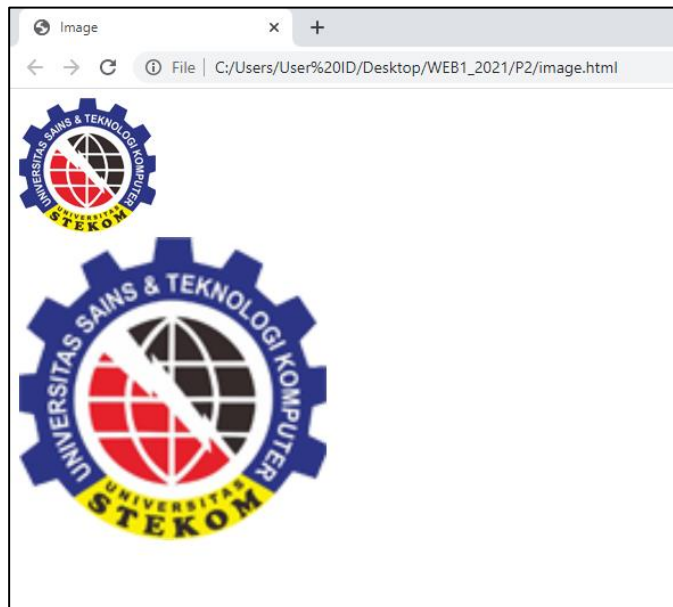
```

Penulisan tag `` Internal

```
<img src = "gambar/logostekom.png">
```

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Image</title>
5 </head>
6 <body>
7   <!-- gambar 1 -->
8   <img src = "gambar/logostekom.png" alt="Logo Universitas Stekom">
9
10  <br>
11
12  <!-- gambar 2 -->
13  
15 </body>
16 </html>
```

Gambar 2.14 Contoh Sintak Menyisipkan Gambar



Gambar 2.15 Hasil Contoh Sintak Menyisipkan Gambar

LATIHAN MANDIRI

1. Buatlah sintac HTML untuk menampilkan tulisan seperti berikut ini

CO₂

$X^2 + 6x + 6$

Windows merupakan salah satu contoh *Operating System*

2. Buatlah sintac HTML untuk menampilkan pesan seperti berikut ini

Daftar Nama Mahasiswa Kelas Pemrograman Web

- Pria
 1. Dendy Kurniawan
 2. Antariksa Wibawa
- Wanita
 1. Erinda Anasia Agustin
 2. Safira Berliana Putri

BAB 3

HYPERLINK & FORM PADA HTML

TUJUAN :

- a. Mahasiswa dapat menge-link-kan halaman antar dokumen HTML.
- b. Mahasiswa mampu membuat tautan ke tujuan tertentu dari sebuah dokumen HTML.
- c. Mahasiswa dapat membuat form pada dokumen HTML
- d. Pengguna memahami cara menggunakan fungsionalitas formulir di header dokumen HTML
- e. Mahasiswa dapat menyisipkan field sesuai dengan tujuan penggunaannya.

Overview :

Salah satu keunggulan aplikasi berbasis web adalah dapat menghubungkan satu dokumen dengan dokumen yang berada di server yang berbeda, baik yang berada di server yang sama maupun server yang berbeda. HTML memungkinkan Anda melakukan "interaksi yang lebih luas" dengan dokumen itu sendiri. Misalnya, kemampuan dokumen HTML dapat menyediakan fasilitas input atau pengiriman pengguna. Informasi ini dapat diolah lebih teliti menjadi informasi yang sangat dibutuhkan oleh pengunjung situs maupun organisasi pengelola situs tersebut. Teknik pengisian ini menggunakan bentuk elemen.

3.1 HYPERLINK HTML

Elemen utama HTML adalah tautan, sering dikenal sebagai tautan hypertext, hyperlink, atau lebih dikenal sebagai tautan. Dengan hyperlink ini, Anda dapat membuka lebih banyak dokumen HTML atau mengarahkan langsung ke dokumen HTML yang diinginkan. Hyperlink ini dapat diaktifkan pada teks tertentu atau bahkan gambar, namun jika diaktifkan pada teks, teks akan ditampilkan dengan cara yang berbeda dan peringatan akan berubah (secara default). Tag yang di gunakan adalah tag <a> yang mempunyai arti anchor atau jangkar.

Tabel 3.1 Atribut Dalam Hyperlink

Atribut	Kegunaan
href	Menunjukkan URL yang di tuju
name	Memberikan nama pada bagian tertentu
title	Menunjukkan judul dari link Ketika cursor di Gerakan di sekitar area hiperlink
target	Menunjukkan target yang ditampilkan link
	_self : membuka link di tab yang sama
	_blank : membuka link di tab baru
	_parent : membuka link di parent frame
	_top : membuka link dokumen di seluruh body

Penulisan External Link :

```
<a href="https://www.google.com/">Klik disini</a>
```

Penulisan Internal Link :

```
<a href="hyperlink2.html">Klik disini</a>
```

Penulisan Relatif Link atau berbeda Folder penyimpanan :

```
<a href="admin/hyperlink3.html">Selanjutnya</a>  
<a href="../hyperlink2.html">Halaman2</a>
```

Untuk menggunakan tautan yang akan mengarah ke badan dokumen, klik di sini.

"#nama_section" menjadi

```
<a href="#bagian1">Bagian 1</a>
```

Dengan ketentuan kita harus memberikan Id terlebih dahulu.

```
<h2 id="bagian1">Bagian 1</h2>
```

Contoh : buat 3 file html berikut ini

File name : index.html

```

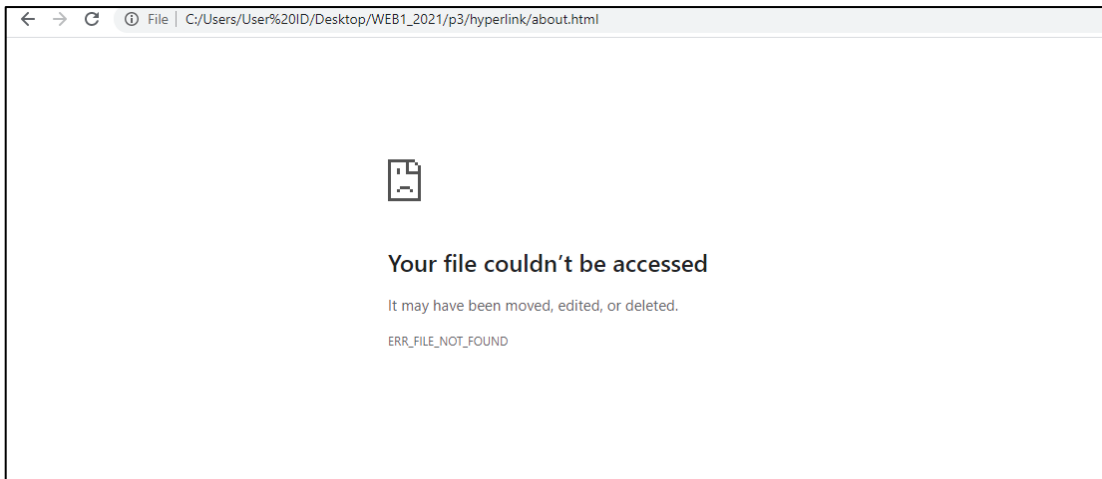
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Tutorial Link di HTML</title>
5   </head>
6   <body>
7     <h1 id="title">Selamat Datang</h1>
8     <p>
9       <a href="index.html" >Beranda</a> |
10      <a href="contact.html" >Contact</a> |
11      <a href="about.html" >About</a>
12    </p>
13    <hr>
14    <p>
15      Selamat datang di websiteku. Coba klik menu di atas,
16      maka kamu akan membuka halaman yang berbeda. Semua
17      link di atas adalah internal link.
18    </p>
19    <hr>
20    <a href="#footer"><font color="blue">scrall ke bawah</font></a>
21    <br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
22    <br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
23    <br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
24    <br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
25    <div id="footer">
26      Copyright &copy; <a href="https://stekom.ac.id/" target="_blank">
27        Universitas Stekom</a>
28    </div>
29    <a href="#title"><font color="blue">scrall ke atas</font></a>
30  </body>
</html>

```

Gambar 3.1 Sintak index.html untuk Link

Tabel 3.2 Penjelasan Sintak Pada Gambar 3.1

LINE	PENJELASAN
Line 7	id="title" adalah untuk memberikan nama id title pada tag h1 line 7 yang nantinya dapat di panggil oleh tag atau bagian document.
line 9 – 10	pemanggilan internal link file html untuk di tampilkan jika di klik
line 14 – 18	isi pesan yang di tampilkan dari halaman beranda (index.html)
line 19	membuat baris
line 20	memanggil dokumen bagian dengan nama ID footer (ada pada line 25)
line 21 – 24	berisi enter untuk membuat halaman menjadi panjang (agar bisa di scrall untuk menunjukan pemanggilan dokumen ke bagian) Anda dapat mengisi bagian ini dengan article / text apapun
line 25	seperti line 7, berguna untuk memebrikan nama ID footer pada tag div line 25
line 26	&copy membuat symbol ©, memanggil external link stekom.ac.id dengan nama Universitas Stekom , target=_blank untuk membuka link di new tab
line 28	Memanggil dokumen bagian dengan nama ID title (ada pada line 7) Selanjutnya Anda harus membuat file contact.html dan about.html agar ketika link yang ada di index.html di klik, akan mengarah ke file yang dituju. Jika nama file salah atau file belum di buat maka akan di tampilkan halaman berikut ini



Gambar 3.2 Contoh Saat Halaman Belum di Buat

File Name : contact.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Tutorial Link di HTML</title>
5   </head>
6   <body>
7     <h1>Contact</h1>
8     <p>
9       <a href="index.html" >Beranda</a> |
10      <a href="contact.html" >Contact</a> |
11      <a href="about.html" >About</a>
12    </p>
13    <hr>
14    <p>
15      Kamu bisa menghubungi saya melalui nomer WA: 081xxxxxxxxxxxx
16
17      <a href="https://api.whatsapp.com/send?phone=6281234567890&text=Saya
18      tertarik dengan web Anda">+6281234567890</a>
19
20      atau juga alamat email: universitas@stekom.ac.id
21    </p>
22    <hr>
23    <div>
24      Copyright &copy; <a href="https://stekom.ac.id/" target="_blank">
25      Universitas Stekom</a>
26    </div>
27  </body>
28 </html>
```

Gambar 3.3 Sintak contac.html untuk Link

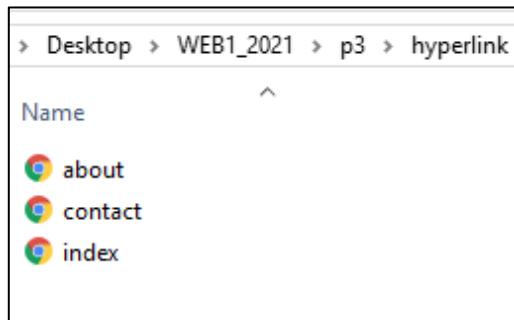
Penjelasan masing – masing line source code, sama dengan penjelasan file index.html. namun ada tambahan api pada line 17 yang fungsinya mindirect ke WA nomer tersebut untuk mengirim pesan “Saya tertarik dengan Web Anda”. setelah ini Anda membuat file about.html

File name : about.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Tutorial Link di HTML</title>
5   </head>
6   <body>
7     <h1>About</h1>
8     <p>
9       <a href="index.html" >Beranda</a> |
10      <a href="contact.html" >Contact</a> |
11      <a href="about.html" >About</a>
12    </p>
13    <hr>
14    <p>
15      Ini adalah halaman about dari website saya.
16      Jadi ini adalah contoh cara membuat link internal di HTML.
17    </p>
18    <hr>
19    <div>
20      Copyright &copy; <a href="https://stekom.ac.id/" target="_blank">
21        Universitas Stekom</a>
22    </div>
23 </body>
24 </html>
```

Gambar 3.4 Sintak about.html untuk Link

Penejelasan seperti file index.html. Silahkan jalankan file html tersebut dengan web browser yang ada di computer Anda. Coba persatu tombol Beranda, Contact, About yang ada di header serta tombol scroll ke atas, scroll ke bawah yang ada pada halaman beranda dan tombol universitas stekom yang ada pada footer masing – masing halaman web tersebut. Catatan : letakkan 3 file tersebut ke dalam 1 folder yang sama



Gambar 3.4 Peletakan ke Tiga File untuk Link

3.2 FORM

Dalam HTML, formulir adalah bagian dari konten yang berfungsi sebagai kolom input atau sebagai permintaan pengguna; itu akan segera dihapus atau diubah agar dapat digunakan sesuai kebutuhan. Pertimbangkan prosedur transfer data, browsing, hapus, penguncian data, dan yang serupa lainnya. Prosedur pengisian formulir HTML dijelaskan di sini.

```
<form action="..URL.." name="..Nama Form.." method="GET|POST"> Isi Form </form>
```

"Nama Formulir" mengacu pada nama formulir yang sekarang digunakan, "Metode" mengacu pada metode atau sarana yang digunakan untuk mentransfer data ke server, dan "Action" mengacu pada alamat atau file lain yang digunakan untuk menerima input dari form di server.

a) MEDIA INPUT DALAM FORM

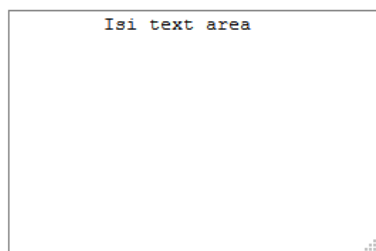
Salah satu kegunaan pembuatan media inputan adalah untuk membantu user agar lebih mudah dalam memasukan data kedalam halaman web. Media inputan terdiri dari Kontrol berupa text, radio button, checkbox, dropdown list dan button.

1) KONTROL BERUPA TEXT

Menggunakan dua textarea dan input tag anak, formulir HTML dapat digunakan untuk memasukkan informasi berbasis teks. Tag input digunakan untuk memasukkan teks yang hanya berisi satu bar, sedangkan tag textarea digunakan untuk memasukkan teks yang memiliki beberapa baris.

```
<textarea name="..nama textarea.." id="..nama Id textarea.." cols="30" rows="10" >
  Isi text area
</textarea>
```

Hasilnya :



Didalam text area terdapat beberapa atribut yaitu

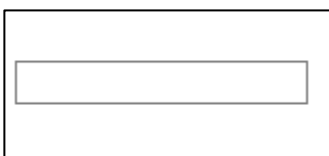
- **Name**, digunakan untuk memberi nama pada tag textarea
- **Id**, digunakan untuk memberi nama pada textarea
- **Cols**, digunakan untuk menentukan panjang dari textarea
- **Rosw**, digunakan untuk menentukan tinggi dari textarea

Dengan menggunakan properti height dan width CSS, Anda juga dapat mengubah margin kiri dan kanan textarea.

Meskipun dapat menampung jumlah kata yang banyak, textareas tidak dapat digunakan untuk semua situasi entri data. Biasanya menginginkan pengguna untuk memasukkan data satu digit dari sejumlah besar teks. Anda dapat menggunakan elemen masukan untuk jenis masukan seperti :

```
<input type="text">
```

Hasilnya :



Ketahui bahwa elemen ini akan menggunakan tipe atribut untuk menunjukkan tipe data yang akan didistribusikan. Ada banyak tipe data yang bisa diakses, dan browser akan mencocokkan tipe teks yang tersedia dengan atribut yang Anda tentukan. Mendefinisikan istilah untuk jenis atribut meliputi:

- | | |
|-------------------|--------------|
| 1. color | 9. time |
| 2. datetime-local | 10. datetime |
| 3. number | 11. month |
| 4. tel | 12. search |
| 5. week | 13. url |
| 6. Date | 14. password |
| 7. email | 15. text |
| 8. range | 16. fil |

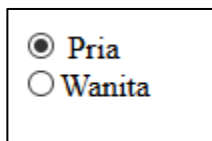
2) KONTROL BERUPA RADIO BUTTON

Tombol radio digunakan untuk menunjukkan pilihan dengan komponen emosional yang kuat. Biasanya digunakan di situs web untuk pertanyaan komprehensif seperti yang ditanyakan saat pendaftaran online, pilihan ini biasanya berkaitan dengan keyakinan seseorang dalam hal agama, politik, atau topik lainnya.

Tombol radio dibuat menggunakan input elemen HTML dan atribut ndengan tipe :
`radio`

```
<input type="radio" name="jeniskelamin" value="pria"> Pria  
<br>  
<input type="radio" name="jeniskelamin" value="wanita">Wanita
```

yang akan menghasilkan elemen masukan seperti berikut:



The image shows a rectangular box containing two radio button options. The first option is 'Pria' with a filled radio button, indicating it is selected. The second option is 'Wanita' with an empty radio button.

Dalam proses pembuatan tombol radio elemen, dua atribut lainnya—nama dan nilai—harus ditentukan. Atribut nama digunakan untuk memberi tahu browser bahwa tombol radio tertentu milik grup tombol radio tertentu, sehingga pengguna tidak dapat memilih dua opsi untuk tombol radio yang dipilih.

3) KONTROL BERUPA CHECKBOX

Berbeda dengan tombol radio yang hanya memungkinkan pengguna memilih satu opsi, kotak yang menyala memungkinkan pengguna memilih satu, beberapa, atau tidak ada opsi sama sekali. Ini biasanya digunakan untuk mengumpulkan data operasional berkualitas tinggi. Anda dapat menggunakan "Centang kotak" untuk masking atribut. Kotak centang input adalah alat untuk memilih opsi atau pilihan tertentu. Jika Anda mengisinya, pengguna akan memeriksanya (mencentang), dan jika tidak, Anda hanya akan mendapatkan peringatan sederhana.

Mirip dengan tombol radio, elemen ini dibuat menggunakan elemen input yang jenis atributnya menentukan kotak centang.

```

<input type="checkbox" name="jurusan" value="sistemkomputer">Sistem Komputer
<br>
<input type="checkbox" name="jurusan" value="teknikkomputer">Teknik Komputer
<br>
<input type="checkbox" name="jurusan" value="teknikinformatika">Teknik Informatika

```

Syntax diatas akan menghasilkan elemen masukan seperti berikut:

4) KONTROL BERUPA DROPDOWN LIST

Penciptaan dua komponen menu drop-down, pilih dan opsi, dilakukan. Elementselect menyertakan semua opsi elemen yang tersedia untuk membuat menu drop-down. Elemen sebenarnya dari opsi adalah pengidentifikasi dropdown yang diinginkan. Perhatikan kode di bawah ini:

```

<select name="satuan" id="satuan">
  <option value="lembar">Lembar</option>
  <option value="botol">Botol</option>
  <option value="sachcet">Sachcet</option>
  <option value="box">BOX</option>
  <option value="unit">Unit</option>
</select>

```

Yang menghasilkan

Jika kita menambahkan atribut multiple

```

<select name="satuan" id="satuan" multiple>

```

Maka akan menghasilkan tampilan

b) BUTTON

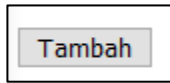
Kontrol input bertipe button pada HTML mewakili tombol yang dapat diklik, yang biasanya digunakan untuk mengirimkan formulir ke alamat tujuan atau action dalam form.

1) Submit

Tujuan tombol ini adalah menampilkan URL yang telah disetujui untuk tindakan atribut. Pembuatan tombol submit juga melibatkan penggunaan elemen input, yang jenis atributnya diidentifikasi oleh nomor submit, seperti yang terlihat di bawah ini:

```
<input type="submit" name="submit" value="Tambah">
```

Sintax diatas akan menghasilkan elemen masukan seperti berikut:



2) Reset

Tombol ini bertujuan untuk mengirim form ke tahap awal (mengosongkan nilai semua elemen yang ada pada form). Mirip dengan kirim tombol, pembuatan tombol dilakukan dengan kondisi yang sama, tetapi dengan nilai tipe atribut yang berbeda yang ditunjukkan oleh tombol reset pada tombol saat ini.

```
<input type="reset" name="reset" value="Batal">
```

Sintax diatas akan menghasilkan elemen masukan seperti berikut:



c) ORGANISASI ELEMEN FORM

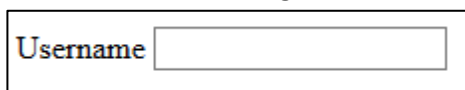
HTML juga menyediakan elemen yang dapat digunakan sebagai penanda elemen yang digunakan dalam formulir untuk memberikan informasi kepada pengguna. Label, fieldset, dan legend adalah satu-satunya elemen yang dapat digunakan untuk tujuan ini.

1) Label

Label elemen menawarkan kemampuan untuk melampirkan teks ke label dengan menggunakan formulir elemen. Unsur dalam kalimat ini ditentukan oleh deskripsi yang dimaksudkan dari atribut kalimat itu. Atribut untuk label elemen memiliki nilai numerik yang sama dengan id atribut untuk formulir pengiriman elemen. Nilai serupa akan menyertakan elemen input dan label yang disorot, memungkinkan pengguna mengklik teks yang telah disediakan oleh elemen label untuk menghasilkan nilai.

```
<label for="username">Username</label>  
<input type="text" name="username" id="username">
```

Sintax diatas akan menghasilkan elemen masukan seperti berikut:

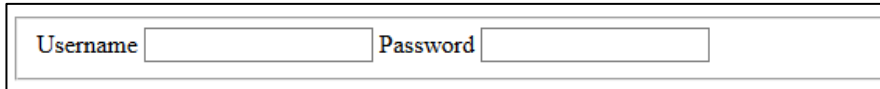


2) Fieldset

Element fieldset adalah alat yang digunakan untuk mengelompokkan sejumlah elemen yang berhubungan dengan bentuk untuk menunjukkan bahwa masing-masing elemen ini adalah bagian dari satu grup atau jika tidak terhubung.

```
<fieldset>  
  <label for="username">Username</label>  
  <input type="text" name="username" id="username">  
  <label for="password">Password</label>  
  <input type="text" name="password" id="password">  
</fieldset>
```

Sintax diatas akan menghasilkan elemen masukan seperti berikut:

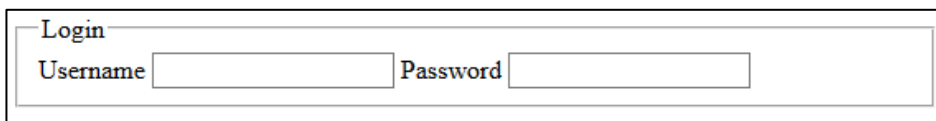


3) Legend

Legenda elemen digunakan untuk memberikan fieldset dengan judul. Penggunaan elemen legenda sangat mudah dan hanya perlu mengacu pada elemen sebagai anak pertama dari fieldset, seperti yang terlihat di bawah ini:

```
<fieldset>
  <legend>Login</legend>
  <label for="username">Username</label>
  <input type="text" name="username" id="username">
  <label for="password">Password</label>
  <input type="text" name="password" id="password">
</fieldset>
```

Sintax diatas akan menghasilkan elemen masukan seperti berikut:



d) VALIDASI MASUKAN FORM

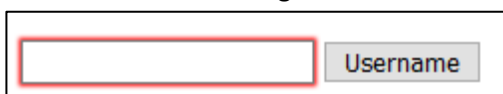
Validasi formulir digunakan untuk memeriksa keakuratan data yang dimasukkan ke dalam formulir. Misalnya, jika kami ingin memvalidasi bahwa bidang email dalam formulir memerlukan alamat email, kami memerlukan pengguna untuk memasukkan alamat email dalam format yang valid. Jika pengguna memasukkan alamat email dalam format yang tidak valid, pesan kesalahan akan muncul di lapangan dan data tidak dapat diproses. Validasi formulir memastikan bahwa browser dapat mengeksekusi data yang dimasukkan pengguna dalam format yang konsisten dengan yang ditentukan dalam standar HTML.

1) Atribut required

Atribut ini digunakan untuk mengidentifikasi bahwa form harus diisi. Jika pengguna tidak mengisi form yang sudah ada atribut ini maka akan tampil pesan peringatan.

```
<form action="" method="POST">
  <input type="text" name="username" required>
  <input type="button" name="username" value="Username">
</form>
```

Sintax diatas akan menghasilkan elemen masukan seperti berikut:



2) Atribut maxlength

Atribut ini digunakan untuk mengatur berapa jumlah karakter yang dapat di input form. Penggunaanya kita harus menambahkan atribut maxlength dan menentukan value yang merupakan jumlah max karakter inputan.

```
<form action="" method="POST">
<input type="text" name="username" maxlength="6">
<input type="button" name="username" value="Username">
</form>
```

Sintax diatas akan menghasilkan elemen masukan seperti berikut:

3) Atribut type : |number|url|email|

Penggunaan atribut type mempunyai fungsi bermacam macam, jika kita menginginkan form hanya bisa diisi dengan angka saja maka gunakan number pada typenya. Jika kita menginginkan yang diisikan tipe url atau email, ganti dengan url atau email. Sebagai contoh perhatikan sintaks di bawah ini:

```
<form action="" method="POST">
<input type="number" name="username">
<input type="button" name="username" value="Username">
</form>
```

Sintax diatas akan menghasilkan elemen masukan seperti berikut:

Contoh Membuat Form :

Form.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Registrasi</title>
5 </head>
6 <body>
7 <form action="contact.php" method="POST">
8 <fieldset>
9 <legend>Registrasi</legend>
10 <p>
11 <label>Nama:</label>
12 <input type="text" name="nama" placeholder="Nama lengkap..." />
13 </p>
14 <p>
15 <label>Username:</label>
16 <input type="text" name="username" placeholder="Username..." />
17 </p>
18 <p>
19 <label>Email:</label>
20 <input type="email" name="email" placeholder="Your email..." />
21 </p>
```

```

22 <p>
23 <label>Password:</label>
24 <input type="password" name="password" placeholder="Passowrd..." />
25 </p>
26 <p>
27 <label>Jenis kelamin:</label>
28 <label><input type="radio" name="jenis_kelamin" value="laki-laki" />
29 <input type="radio" name="jenis_kelamin" value="laki-laki" />
30 <input type="radio" name="jenis_kelamin" value="laki-laki" />
31 <label><input type="radio" name="jenis_kelamin" value="perempuan" />
32 <input type="radio" name="jenis_kelamin" value="perempuan" />
33 <input type="radio" name="jenis_kelamin" value="perempuan" />
34 </p>
35 <p>
36 <label>Hobi :</label>
37 <input type="checkbox" name="hobi1" value="Menonton"> Menonton
38 <input type="checkbox" name="hobi2" value="Olahraga"> Olahraga
39 <input type="checkbox" name="hobi3" value="Shopping"> Shopping
40 </p>
41 <p>
42 <label>Agama:</label>
43 <select name="agama">
44 <option value="islam">Islam</option>
45 <option value="kristen">Kristen</option>
46 <option value="hindu">Hindu</option>
47 <option value="budha">Budha</option>
48 <option value="katolik">Katolik</option>
49 </select>
50 </p>
51 <p>
52 <label>Biografi:</label>
53 <textarea name="biografi"></textarea>
54 </p>
55 <p>
56 <input type="submit" name="submit" value="Daftar" />
57 <input type="reset" name="reset" value="Hapus" />
58 </p>
59 </fieldset>
60 </form>
61 </body>
62 </html>

```

Hasilnya :

The image shows a web browser window with a single tab titled "Registrasi". The address bar shows the file path: "C:/Users/User%20ID/Desktop/20.2/pagi/form.html". The main content area contains a registration form with the following fields and controls:

- Nama:** Text input field with placeholder text "Nama lengkap..."
- Username:** Text input field with placeholder text "Username..."
- Email:** Text input field with placeholder text "Your email..."
- Password:** Text input field with placeholder text "Passowrd..."
- Jenis kelamin:** Radio button group with options "Laki-laki" and "Perempuan".
- Hobi :** Checkboxes for "Menonton", "Olahraga", and "Shoping".
- Agama:** Dropdown menu with "Islam" selected.
- Biografi:** Text area with a small diagonal icon in the bottom right corner.
- Buttons:** "Daftar" and "Hapus" buttons.

Gambar 3.5 Hasil dari form.html

BAB 4

MEMBUAT TABEL & LAYOUT WEB PAGE

TUJUAN :

- a. Pengguna dapat membuat dan memformat tabel dalam dokumen HTML.
- b. Pengguna dapat memahami cara membuat bingkai dalam dokumen HTML.
- c. Pengguna dapat memahami cara menggunakan frame yang dimaksud.
- d. Mahasiswa dapat menyesuaikan batasan frame.
- e. Mahasiswa mampu mengenali target sel frame dari hyperlink.

Overview :

Tabel adalah metode untuk menampilkan informasi di header situs web dalam format baris dan kolom. Hampir setiap situs web dengan kualitas dan profesionalisme tinggi perlu dianalisis menggunakan tabel. Mirip dengan spreadsheet yang memiliki kolom dan berformat angka-angka, tabel online memiliki kolom yang berformat link, gambar, dan teks.

Saat membuat situs web, penggunaan tabel dan bingkai dimungkinkan. Secara umum, tidak ada perbedaan yang sangat signifikan dalam situasi ini, tetapi selama pembuatan waktu akses situs web, kemungkinan hal ini akan menjadi sangat jelas dalam kaitannya dengan waktu respons yang diperlukan.

Frame akan dapat mengubah satu dokumen HTML menjadi beberapa dokumen HTML. Meskipun demikian, setiap bagian tertaut ke dokumen HTML yang valid.

4.1 TABEL

Tabel digunakan untuk menampilkan data dalam format kolom dan bar. Padahal, setiap kolom memiliki data yang sejenis, dan setiap baris yang ditempatkan pada sebuah kolom mencakup data yang telah disusun menjadi satu kesatuan. Setiap tabel memiliki judul, tempat bagi Anda untuk menunjukkan peringatan atau kode bar yang sedang digunakan, bar untuk informasi, dan sel untuk setiap item. Kolom pertama pada tabel di bawah berisi informasi tentang header; setiap baris mengubah tag HTML untuk tabel; dan setiap kolom mengubah fungsi atau tujuan dari sebuah tag. Elemen yang sering digunakan dalam membuat tabel HTML dibahas di bagian ini.

Elemen tabel HTML membuat data tabular dengan menyajikan informasi dalam tabel dengan dua dimensi yang terdiri dari sel berisi data dan kolom teks. Ada beberapa tag yang harus dipahami sebelum membuat tabel HTML:

- **<table>** pembungkus tabel
- **<thead>** pembungkus kepala tabel
- **<tbody>** pembungkus body tabel
- **<tfoot>** pembungkus kaki tabel
- **<tr>** "tabel row" membuat baris di tabel
- **<td>** "table data" membuat sel di tabel
- **<th>** "table head" membuat judul pada header tabel

Selain itu juga terdapat Tag Atribut yang dapat digunakan untuk mengatur tabel didalam html, berikut ini contoh Tag atribut yang dapat di gunakan :

- **Align=left | Center | right**, digunakan untuk mengatur posisi tabel
- **Width**, Atribut digunakan untuk menentukan lebar yang diinginkan dari seluruh table, jika atribut tidak ada , lebar table ditentukan oleh data
- **Colspan**, Atribut digunakan untuk menggabungkan kolom
- **Rowspan**, Atribut digunakan untuk menggabungkan baris
- **Bgcolor**, digunakan untuk memberikan warna pada table.
- **Border**, digunakan untuk menentukan lebar garis
- **Caption**, digunakan untuk membuat judul dari table.

Bentuk penulisan tag Table pada Html dapat dilihat pada contoh di bawah ini:

table.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tabel 1 </title>
</head>
```

```

<body>
  <table>
    <tr>
      <td>Baris 1 kolom 1</td>
      <td>Baris 1 kolom 2</td>
    </tr>
    <tr>
      <td>Baris 2 kolom 1</td>
      <td>Baris 2 kolom 2</td>
    </tr>
  </table>
</body>
</html>

```

Hasilnya jika di tampilkan pada browser

Baris 1 kolom 1	Baris 1 kolom 2
Baris 2 kolom 1	Baris 2 kolom 2

Gambar 4.1 Hasil Syntac table.html

a) Fungsi Tag Caption

Tujuan dari Tag Caption adalah untuk membuat judul tabel. Elemen tersebut terletak di dalam tabel elemen dan memiliki atribut yang disejajarkan dengan bagian atas dan tombol. Bentuk penulisannya:

```

<caption>
Daftar Harga
</caption>

```

b) Fungsi Tag Th

Baris pertama yang kita gunakan saat membuat tabel adalah sebagai judul baris berikutnya di bawahnya. Biasanya, tabel pertama dalam skenario ini memiliki perbedaan visual dari tabel kedua agar lebih menonjol, misalnya dengan menggunakan font atau warna yang berbeda.

HTML memiliki tag khusus yang dapat digunakan untuk keperluan ini, yang dikenal dengan tag "<th>" (singkatan dari table head). Untuk menggunakan tag th, yang harus dilakukan adalah mengganti tag td dengan th pada kolom pertama tabel.

Anda dapat melihat contoh penggunaan tag "<th>" dan atribut "align", "rowspan and colspan", dan "caption" pada gambar di bawah ini.

Tag.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Table 1</title>
</head>
<body>
  <table border="1" align="center">
    <caption align="top">Daftar Harga</caption>
    <tr>
      <th>Nama Barang</th>
      <th>Kategori</th>
      <th>Harga satuan</th>
      <th>Jumlah barang</th>
      <th>Total</th>
    </tr>
    <tr>
      <td>Keyboard</td>
      <td rowspan="2">Tools</td>
      <td>Rp.90.000</td>
      <td>4</td>
      <td>Rp.360.000</td>
    </tr>
    <tr>
      <td>Mouse</td>
      <td>Rp.50.000</td>
      <td>5</td>
      <td>Rp.250.000</td>
    </tr>
    <tr>
      <td>USB 8 GB</td>
      <td rowspan="2">Utility</td>
      <td>Rp.60.000</td>
      <td>2</td>
      <td>Rp.120.000</td>
    </tr>
    <tr>
      <td>CD Blank</td>
      <td>Rp.5.000</td>
```



```

        <td>10</td>
        <td>Rp.50.000</td>
    </tr>
    <tr>
        <td colspan="4">Total</td>
        <td>Rp.780.000</td>
    </tr>
</table>

</body>
</html>

```

Hasilnya akan seperti pada gambar di bawah ini

Daftar Harga				
Nama Barang	Kategori	Harga satuan	Jumlah barang	Total
Keyboard	Tools	Rp.90.000	4	Rp.360.000
Mouse		Rp.50.000	5	Rp.250.000
USB 8 GB	Utility	Rp.60.000	2	Rp.120.000
CD Blank		Rp.5.000	10	Rp.50.000
Total				Rp.780.000

Gambar 4.2 Hasil dari syntac tag.html

c) Tag Thead, Tbody dan Tfoot

Pada tag html juga di sediakan tag yang digunakan untuk membedakan antara bagian header table, body table dan footer table. Tag **<thead>** digunakan untk membungkus kepala table yang biasanya berisi judul dari masing masing field yang di buat, Tag **<tbody>** digunakan untk membungkus badan atau body table, yang berisi isi utama dari dari sebuah table. Tag **<tfoot>** Tag untk membungkus bagian footer dari table, yang berisi kesimpulan atau penjumlahan dari table yang telah dibuat. Sebagai contoh lihat Latihan berikut:

thead.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Tabel 1 </title>
</head>
<body>

    <p>Table with thead, tfoot, and tbody</p>
    <table border="1">

```

```

<caption>Daftar Harga</caption>
<thead>
  <tr>
    <th>Nama Barang</th>
    <th>Kategori</th>
    <th>Harga satuan</th>
    <th>Jumlah barang</th>
    <th>Total</th>
  </tr>
</thead>
<tbody>
  <tr>
    <td>Keyboard</td>
    <td rowspan="2">Tools</td>
    <td>Rp.90.000</td>
    <td>4</td>
    <td>Rp.360.000</td>
  </tr>
  <tr>
    <td>Mouse</td>
    <td>Rp.50.000</td>
    <td>5</td>
    <td>Rp.250.000</td>
  </tr>
  <tr>
    <td>USB 8 GB</td>
    <td rowspan="2">Utility</td>
    <td>Rp.60.000</td>
    <td>2</td>
    <td>Rp.120.000</td>
  </tr>
  <tr>
    <td>CD Blank</td>
    <td>Rp.5.000</td>
    <td>10</td>
    <td>Rp.50.000</td>
  </tr>

```

```

        </tr>
    </tbody>
    <tfoot>
        <tr>
            <td colspan="4">Total</td>
            <td>Rp.780.000</td>
        </tr>
    </tfoot>
</table>
</body>
</html>

```

Table with thead, tfoot, and tbody

Daftar Harga				
Nama Barang	Kategori	Harga satuan	Jumlah barang	Total
Keyboard	Tools	Rp.90.000	4	Rp.360.000
Mouse		Rp.50.000	5	Rp.250.000
USB 8 GB	Utility	Rp.60.000	2	Rp.120.000
CD Blank		Rp.5.000	10	Rp.50.000
Total				Rp.780.000

Gambar 4.3 hasil dari sintac thead.html

Elemen thead, tbody, dan tfoot digunakan untuk menyort konten atau elemen yang terlihat masing-masing di bagian atas (head), bawah (foot), dan tengah (body) dari tabel tertentu.

Saat digunakan, elemen <thead> digunakan bersamaan dengan elemen <tbody> dan <tfoot> yang masing-masing menentukan bagian atas (tubuh) dan bagian bawah (kaki) dari tabel tertentu.

Tidak ada atribut khusus yang tersedia untuk elemen "<tbody>" di HTML5, meskipun beberapa atribut, termasuk align, bgcolor, char, charoff, dan valign, tersedia di HTML4.01. Namun, atribut tersebut sudah digunakan, dan Anda sebaiknya hanya menggunakan CSS untuk memodifikasi elemen <tbody>.

4.2 FRAME

Dengan menggunakan framing pada HTML, dimungkinkan untuk membuat dokumen HTML dari tampilan file HTML yang telah dibagi menjadi beberapa bagian, dengan setiap bagian berisi satu atau beberapa bagian yang bertentangan satu sama lain. Ini memungkinkan penggunaan bandwidth internet yang efisien dan kelancaran proses pengunduhan. Berikut adalah contoh sintaks pembuatan frame standar yang sering digunakan.

```

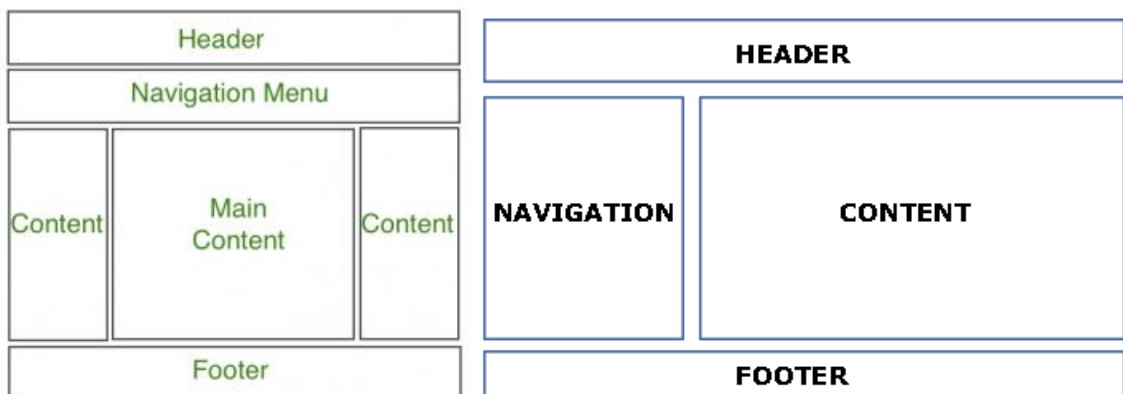
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Frame</title>
5 </head>
6 <frameset rows | cols = #,#,... Border="#">
7   <frame name="#" src="url"></frame>
8 </frameset>
9 </html>

```

Tabel 4.1 Atribut yang digunakan dalam frame

Atribut	Fungsi
Frameset cols	Membuat frame vertikal dengan lebar kolom tertentu
Frameset rows	Membuat frame horisontal dengan tinggi baris tertentu
Frame src	Memasukkan dokumen HTML ke dalam frame
Noframe	Memasukkan body text untuk browser yang tidak menampilkan frame

Dengan memanfaatkan frame ini, kita dapat menentukan tampilan layout website kita lebih mudah dan rapi. Contoh layout web :



Gambar 4.4 Contoh Layout Web

Dan yang akan di praktikan dalam modul ini adalah layout seperti di bawah ini

- **Marquee** adalah alat dalam HTML yang memungkinkan teks bergerak atau berhenti bergerak. Atribut apa saja yang dapat dituliskan pada sebuah marquee dapat dilihat pada tabel 4.2 berikut ini :

Tabel 4.2 Atribut Marquee

NAMA ATRIBUT	KETERANGAN	VALUE
width	mengubah lebar teks	Pixel dalam %
height	mengubah tinggi teks	Pixel dalam %
direction	Mengubah arah dari gerakan teks	Left Right Up down
behaviour	Mengubah gerakan	Slide: Teks bergerak sekali lalu berhenti Scroll: Teks bergerak berputar frasa yang berbeda adalah: teks bergerak dari kiri ke kanan lalu berulang
scrollDelay	mengubah waktu tunda gerakan dalam mili detik	Value
scrollAmount	mengubah r kecepatan gerakan	Value dalam pixel
bgColor	mengubah warna latar belakang teks	color
Loop	mengubah jumlah perulangan	Value -1 infinite
Vspace	mengubah ruang antara atas/bawah dengan teks	Value
Hspace	mengubah ruang antara kiri/kanan dengan teks	Value

Teks berjalan dari kanan ke kiri:

```
<MARQUEE align="center" direction="right" height="200" scrollAmount="2" width="100%">
marquee dari kanan ke kiri
</MARQUEE>
```

Teks dengan atribut behaviour:

```
<MARQUEE align="center" direction="left" height="200" scrollAmount="3" width="70%"
behavior="alternate">
marquee dengan atribut behaviour
</MARQUEE>
```

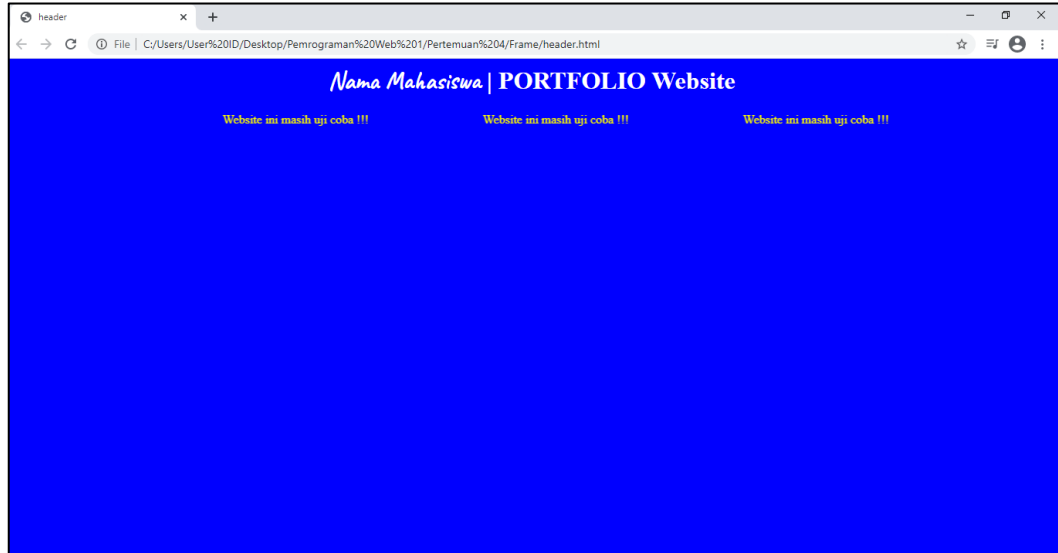
eksperimen lainnya:

```

<MARQUEE behavior="scroll" direction="left" scrollamount="1">Slow scroll speed</MARQUEE>
<MARQUEE behavior="scroll" direction="left" scrollamount="10">Medium scroll speed</MARQUEE>
<MARQUEE behavior="scroll" direction="left" scrollamount="20">Fast scroll speed</MARQUEE>

```

Dan ketika header.html ini di buka pada web browser maka dia menampilkan seperti berikut ini



Gambar 4.6 hasil header.html

2. menu.html

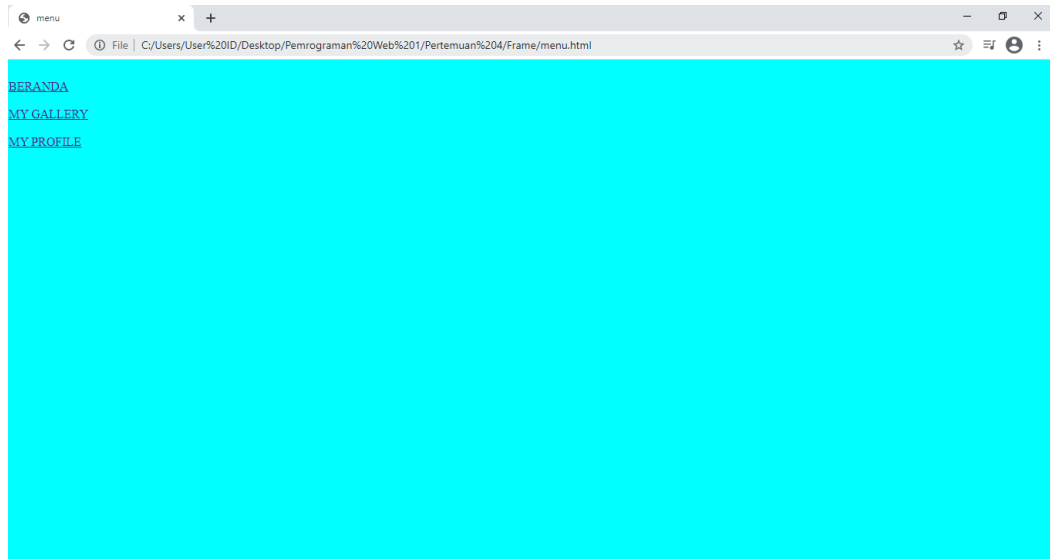
```

1 <html>
2 <head>
3 <title>menu</title>
4 </head>
5 <body bgcolor="#00FFFF">
6 <br>
7 <a href="content_beranda.html" target="content">BERANDA</a>
8 <br><br>
9 <a href="content_galeri.html" target="content">MY GALLERY</a>
10 <br><br>
11 <a href="#">MY PROFILE</a>
12 </body>
13 </html>

```

Keterangan :

- Pada bagian ini hanya membuat hyperlink dengan tampilan teks yang di arahkan atau di targetkan pada "content". Nama content tersebut nantinya kita buat dengan menggunakan frame
- Untuk line 11 merupakan link buntu yang mana dia membuat link namun belum aktif atau tidak akan ada respon apa – apa ketika link tersebut di klik
- Jika halaman tersebut di buka dengan web browser maka hasilnya akan seperti di bawah ini



Gambar 4.7 Hasil menu.html

3. content_beranda.html

```

1 <html>
2 <head>
3   <title>content</title>
4 </head>
5 <body bgcolor="#F0FFF0">
6   <h2 align="center">BERANDA</h2>
7   <!-- Isi conten beranda Anda disini -->
8 </body>
9 </html>
10 |
  
```

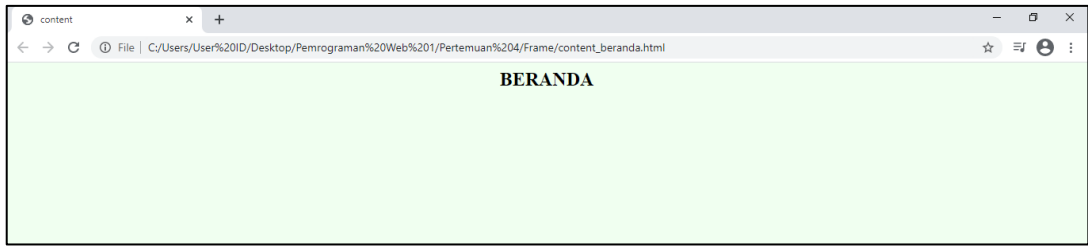
Keterangan :

- Anda dapat mengisi halaman content_beranda.html ini dengan sesuka hati Anda. Hasilnya akan seperti gambar 4.8
- Anda juga dapat membuat dokumen html untuk content – content yang lainnya untuk di tautkan ke dalam hypelink dokumen menu.html
- Misalkan Anda membuat dokumen **content_galeri.html** di bawah ini, dan hasilnya seperti gambar 4.9

```

1 <html>
2 <head>
3   <title>content</title>
4 </head>
5 <body bgcolor="#F0FFF0">
6   <h2 align="center">GALERY FOTO KARYA SAYA</h2>
7   <!-- Isi conten Galery Anda disini -->
8 </body>
9 </html>
  
```

- Hanya di bedakan isi nya saja antar content satu dengan content yang lainnya

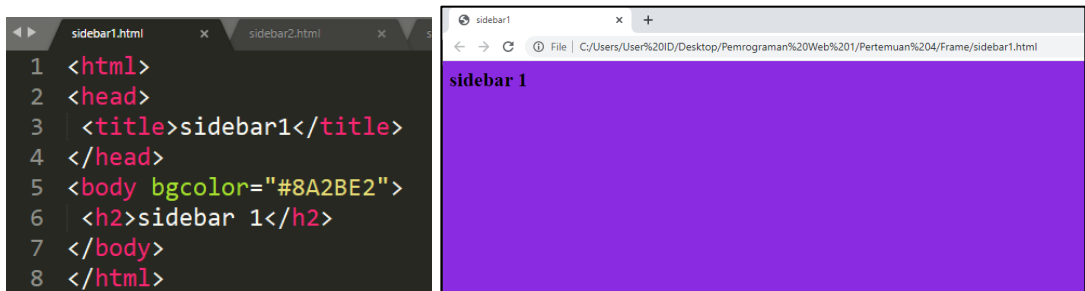


Gambar 4.8 Hasil content_beranda.html



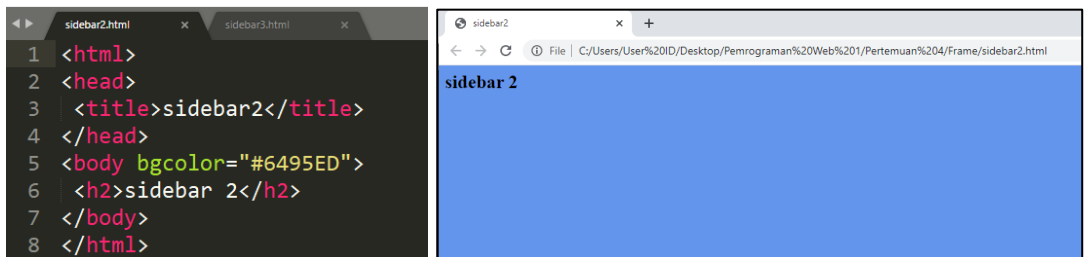
Gambar 4.9 Hasil content_galeri.html

4. sidebar1.html



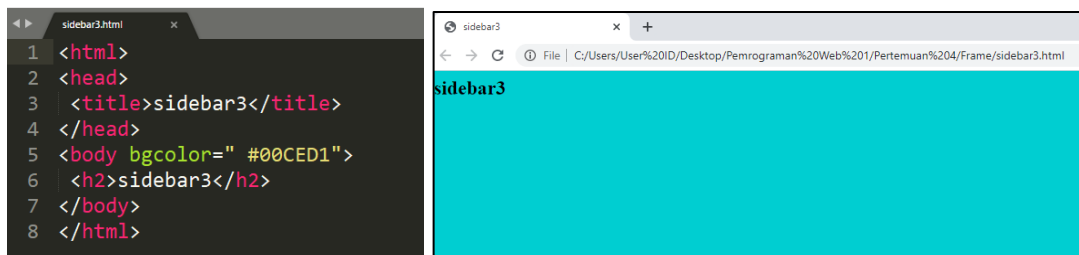
Gambar 4.10 syntac dan hasil sidebar1.html

5. sidebar2.html



Gambar 4.11 syntac dan hasil sidebar2.html

6. sidebar3.html



Gambar 4.12 syntac dan hasil sidebar3.html

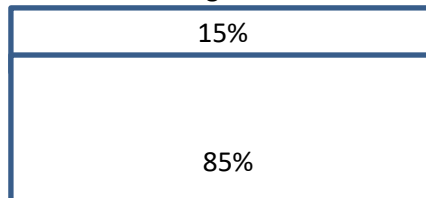
Setelah membuat 6 dokument html di atas, sekarang saatnya membuat dokumen html fame yang berfungsi menggabungkan ke 6 dokumen di atas di dalam satu layout web.

7. index.html

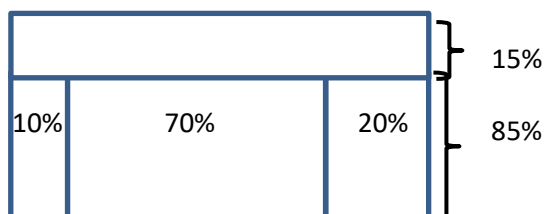
```
index.html x
1 <html>
2 <head>
3 <title>Beranda</title>
4 </head>
5
6 <frameset rows="15%,85%" border="1">
7 <frame name="header" src="header.html"></frame>
8 <frameset cols="10%,70%,20%" border="1">
9 <frame name="menu" src="menu.html"></frame>
10 <frame name="content" src="content_beranda.html"></frame>
11 <frameset rows="30%,30%,30%" border="1">
12 <frame name="sidebar1" src="sidebar1.html"></frame>
13 <frame name="sidebar2" src="sidebar2.html"></frame>
14 <frame name="sidebar3" src="sidebar3.html"></frame>
15 </frameset>
16 </frameset>
17 </frameset>
18 </html>
```

Keterangan :

- **line 6** : membagi baris menjadi 2 bagian dengan ukuran baris 15% dan baris 2 dengan ukuran 85% dengan ukuran border = 1



- **Line 7** : memberi name="header" pada baris pertama (15%) dimana name ini nanti dapat di gunakan untuk target hyperlink. Selain itu pada line 7 ini juga memanggil dokumen header.html yang sudah kita buat sebelumnya
- **Line 8** : membagi kolom pada baris ke dua (85%) menjadi tiga bagian dengan ukuran masing – masing 10%, 75% dan 20% dengan border = 1



- **Line 9** : Memberikan name menu dan memanggil dokumen menu.html untuk di tampilkan pada bagian layout (ukuran 10%)


- **Line 10** : Memberikan name content dan memanggil dokumen content_beranda.html untuk di tampilkan pada bagian layout (ukuran 70% tadi)
- **Line 11** : membagi bagian layout pada baris ke 2 kolom ke 3 (ukuran 20% tadi) menjadi 3 baris dengan masing – masing berukuran 30%



- **Line 12 – 14** : Masing – masing memberikan nama sidebar1, sidebar2, dan sidebar3 serta memanggil dokumen sidebar1.html, sidebar2.html dan sidebar3.html untuk di tampilkan pada bagian yang telah di atur pada line 11 tadi.

TUGAS MANDIRI 2 :

Tambahkan dokument **content_profile.html** kemudian silahkan tampilkan biodata diri anda seperti berikut : *(isi data dengan identitas diri Anda masing – masing)*

BIODATA DIRI		
DATA DIRI	KETERANGAN	FOTO
Nama	Dendy Kurniawan	
NPM	12345678	
Jurusan	Sistem Komputer	
Jenis Kelamin	Laki – laki	
TTL	Kendal, 1 Januari 1991	
Alamat	Jalan Suka Maju No. 57 Kendal	
Hobi	Programing	

Ketentuan :

1. Halaman biodata ini akan tampil di dalam layout saat menu MY PROFILE di klik
2. Untuk bagian header, content beranda, content gallery, sidebar 1- 3 jika Anda modifikasi isinya dan di tampilkan lebih menarik, akan mempengaruhi nilai.

BAB 5

PENGENALAN CASCADING STYLE SHEET (CSS)

TUJUAN :

- a. Mahasiswa Mampu Memahami dan mengetahui Penggunaan CSS pada HTML
- b. Mahasiswa Mengetahui selector dalam CSS
- c. Mahasiswa Mengetahui dan mengimplementasi cara Menulis CSS
- d. Mahasiswa Mampu mengubah background dengan CSS
- e. Mahasiswa Mampu mengubah Hyperlink dengan CSS

Overview :

CSS (Cascading Style Sheets) adalah lembar gaya khusus bahasa yang digunakan untuk mengubah format dokumen yang telah diterbitkan menggunakan markup. Pada tahun 1996, CSS mulai dikenal lebih luas sebagai standar untuk membangun website. Nama CSS berasal dari fakta bahwa setiap deklarasi hubungan homoseksual yang berbeda satu sama lain dapat dilaksanakan secara perlahan sebelum menciptakan hubungan antara orang tua dan anak untuk setiap hubungan gay.

Aplikasi CSS yang paling umum adalah memformat halaman web yang ditulis dalam HTML dan XHTML. Spesifikasi CSS dibuat oleh World Wide Web Consortium (W3C). CSS memungkinkan gambar identik ditampilkan dengan cara yang berbeda untuk metode presentasi yang berbeda, seperti input braille, layar, cetak, dan teks (yang dapat dibaca dengan cara berbeda tergantung pengaturan dasar-suara browser). Dokumen dalam HTML atau XML yang serius juga dapat disajikan secara berbeda, baik dengan menggunakan jenis font yang berbeda atau skema warna yang berbeda dengan CSS.

5.1 OVERVIEW CSS

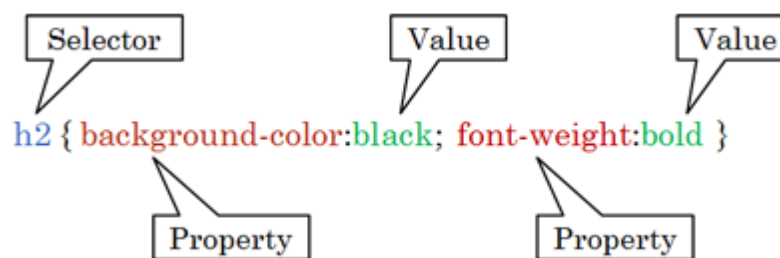
Singkatan CSS singkatan dari "Cascading Style Sheets." Sesuai dengan namanya, CSS memiliki definisi "bahasa style sheet" yang mengacu pada bahasa pemrograman yang digunakan dalam desain web. CSS adalah satu-satunya bahasa pemrograman yang paling banyak digunakan untuk membuat situs web header. Saat mendesain header situs web, CSS menggunakan id dan penanda yang dapat dikenali oleh pengguna secara sekilas.

Dalam bidang pemrograman dan teknologi yang berkembang pesat, CSS tidak lagi hanya digunakan bersamaan dengan HTML dan XHTML, namun masih mampu diterapkan pada pengembangan aplikasi Android. CSS dapat digunakan menggunakan id, class, atau tag tag, yang biasanya disebut sebagai "pemilik".

Font, ukuran font, gaya font, dan format font semuanya dapat diubah dengan CSS. Buat tata letak yang sempit, tinggi, dan ringan, buat situs web yang responsif, ubah bidang formulir, dan lakukan beberapa tugas lain dengan CSS tetapi saya tidak akan membahasnya secara mendalam dalam pelajaran mempelajari prinsip-prinsip CSS ini.

Margin dapat digunakan untuk mengubah ukuran font, warna dapat diubah untuk mengubah warna huruf, dan definisi font dapat diubah untuk mengubah jenis huruf. Menggunakan "latar belakang" dapat digunakan untuk memindahkan teks atau gambar lebih jauh dari penampil. Gunakan "ukuran font" untuk mengubah ukuran font. jenis font yang menggunakan "font-family" dan beberapa gaya font lainnya.

a) Selector



Gambar 5.1 Selector CSS

Karena CSS digunakan untuk mengelola atau bahkan memodifikasi konten HTML, diperlukan metode untuk "gaiting" atau menghubungkan kode CSS dengan tag HTML yang sesuai. Dalam CSS, inilah yang dimaksud dengan Selector.

Sesuai dengan namanya, sebuah pemilih dapat digunakan untuk mencari halaman online yang perlu diubah ukurannya atau ditata. Misalnya, "cari seluh tag <p>", "cari seluh tag HTML yang memiliki atribut class="warning", atau "cari seluh link yang ada di dalam tag <p>".

CSS paling sederhana untuk digunakan adalah tag dari dokumen HTML standar, seperti tag <p>, <h1>, , atau dll. Pemilih CSS dapat menjadi objek yang kompleks tergantung pada kebutuhan pengguna.

1) Universal Selector

Hanya satu pemilih universal tanda "*" yang tersedia di CSS. Selektor Ini dimaksudkan untuk "mencari" semua tag yang tersedia. CSS Pemilah Universal Contoh:

```
* {
  color: blue;
  background-color: white;}
```

Kode CSS yang disediakan di sini efektif untuk membuat setiap tag HTML berwarna coklat dan tampilan latar menjadi hijau. Pada selector universal ini biasanya juga di gunakan untuk mereset settingan default yang ada pada web browser.

2) *Element Type Selector*

Sintaks untuk pemilihan yang pada dasarnya adalah tag untuk HTML itu sendiri adalah Pemilih Jenis Elemen, juga dikenal sebagai Pemilih Tag. Setiap tag HTML dapat digunakan sebagai pemilih, dan pilihan khusus ini akan mencakup semua tag yang ditemuinya. Menggunakan Pemilih Tipe Elemen Contoh CSS:

```
h1 {
  text-decoration: underline;
}
p {
  font-size:14px;
}
```

Kode CSS pada contoh ini akan membuat semua tag dengan tag h1 bergerak ke kiri dan semua tag dengan tag p memiliki lebar 14 piksel. Pengaruh pemilih jenis elemen adalah dari tag pertama hingga tag terakhir. Jika ada tag l di dalam tag "<p>", maka tag tersebut juga akan memiliki lebar 14 piksel hingga diikuti oleh tag penutup "</p>".

3) *Class selector*

Class Selector adalah satu-satunya pemilih yang paling populer dan sering digunakan. Pemilih Kelas akan "mencari" setiap tag yang memiliki atribut kelas dengan nilai yang sesuai. Untuk menggunakan Pemilih Kelas, Anda harus memiliki tag HTML dengan atribut kelas. Contohnya:

```
<p class="paragraf_pertama"> Ini adalah sebuah paragraf pertama</p>
<h1 class="judul">Judul Artikel</h1>
<h2 class="penting">Sub Judul Artikel</h2>
```

Ingatlah bahwa kita menambahkan nama kelas itu sendiri sebagai kelas atribut untuk setiap tag pada daftar di atas. Nama kelas dapat diberikan oleh lebih dari satu tag, dan sebuah tag dapat mencakup lebih dari satu kelas. Misalnya pada paragraf terakhir dari contoh di atas, tag h2 memiliki atribut class="judul penting berwarna". Tag ini terdiri dari tiga kelas: judul, penting, dan kelas berwarna.

Sebaliknya, kode Pemilih Kelas CSS adalah sebagai berikut:

```

.paragraf_pertama {
    color: red;
}

.judul {
    font-size:20px;
}

.penting {
    color:red;
    font-size: 1em;
}

```

Di CSS, pemilih kelas diterapkan dengan memberi tanda centang di depan nama kelas. Sebagai contoh kita, setiap kelas yang memiliki nilai "paragraf pertama" akan menyebabkan teks menjadi merah. Dan setiap judul kelas akan memiliki jenis huruf 20 piksel.

4) ID Selector

Selector yang paling dasar dan sering digunakan adalah ID Selector, yang bekerja berdampingan dengan class Selector (walaupun tidak sesering class selector). Penggunaan pemilih ID hampir identik dengan penggunaan pemilih kelas, kecuali bahwa dengan pemilih kelas, atribut seperti kelas digunakan untuk mengidentifikasi tag HTML, dengan pemilih ID, atribut seperti id digunakan. Contoh penggunaan ID atribut dalam tag HTML:

```

<p id="paragraf_pembuka"> Ini adalah sebuah paragraf pembuka</p>
<h1 id="judul_utama">Judul Artikel</h1>
<h2 id="sub_judul">Sub Judul Artikel</h2>

```

Selain digunakan sebagai pemilih CSS, atribut id juga berfungsi sebagai tag unik untuk masing-masing tag (terutama dipakai untuk kode JavaScript). Karena itu, setiap ID yang digunakan harus unik dan tidak boleh sama dengan yang lain. Dengan kata lain, sebuah ID hanya boleh digunakan satu kali di dalam satu halaman web dan tidak boleh sama. Contoh penggunaan kode pemilih id pemilih kelas CSS adalah sebagai berikut:

```

#paragraf_ pembuka {
    color: red;
}

#judul_utama {
    font-size:20px
}

```

Di CSS, kami menggunakan tanda hash "#" untuk menunjukkan bahwa kami sedang mencari tag dengan id tertentu.

5) Attribute Selector

Dibandingkan dengan selektor-selektor sebelumnya, selektor ini sedikit lebih rumit. Pemilih atribut ini digunakan untuk menemukan setiap tag dengan atribut yang telah diformat dengan benar. Konteks penggunaan pemilih atribut CSS adalah sebagai berikut:

```
[href] {  
    font-size:20px ;  
}  
  
[type="submit"] {  
    width:30px;  
}
```

Seperti yang terlihat dari contoh di atas, setiap atribut yang digunakan harus ditempatkan di antara tanda kurung "[" dan "]". [href] akan berfungsi dengan semua tag yang memiliki atribut bernama "href", terlepas dari nilainya (yang biasanya ditemukan pada tag "<a>"). Misalnya, [type="kirim"] Ini akan berfungsi dengan tag yang memiliki atribut dengan nilai kirim, yang dalam hal ini akan berfokus pada tombol kirim formulir. Meskipun memiliki kemampuan untuk mengidentifikasi tag yang cukup spesifik, atribut ini sering digunakan.

b) Property

Properti CSS adalah elemen atau gaya yang akan dipisahkan dari tag HTML. CSS memiliki beberapa properti yang dapat diterapkan untuk menghasilkan hasil akhir yang kita inginkan. Hampir setiap properti CSS dapat diterapkan ke semua pengguna.

Jika pemilih digunakan, misalnya, untuk "menemukan setiap tag <p>", maka propertinya adalah "apa pun yang ingin Anda ekstrak dari tag <p> itu", seperti ukuran teks, warna, dan jenis font.

c) Value

Nilai CSS adalah ukuran properti. Misalnya, nilai atau ciri properti backgroundcolor, yang digunakan untuk mengubah warna latar belakang suatu pilihan, dapat berupa kombinasi merah, biru, hitam, atau putih.

5.2 PENEMPATAN CSS

CSS menggunakan pemilih (id dan kelas) untuk mengidentifikasi elemen yang akan dimodifikasi atau menerima pesan CSS. Jika HTML dianggap sebagai tiang pada bangunan rumah tertentu, maka CSS berfungsi sebagai dekorasi dan aksesoris pada bangunan rumah tersebut. Ada tiga teknik pengembangan CSS, yaitu:

- Inline CSS Style
- Internal CSS Style
- External CSS Style

Pada pelajaran kali ini, kita akan membahas dan mempelajari tiga teknik penulisan CSS (inline dan internal).

a) In Line Style

Pengkodean sintaks gaya inline adalah teknik pengkodean sintaks CSS yang tidak memerlukan penyeleksi (id dan class). Oleh karena itu, sintaks CSS digunakan baik secara eksplisit maupun implisit untuk elemen HTML. Sintaks CSS diimplementasikan dalam atribut style=" ".

Contoh gaya CSS sebaris:

Untuk membuat contoh penulisan CSS dengan gaya inline dapat menyediakan file HTML. Langkah awal buatlah HTML file dengan diberi nama index.html.

index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Penulisan CSS dengan Inline Style</title>
5 </head>
6 <body>
7
8
9 <h1 style="color:blue">Belajar CSS Dasar Di Universitas Stekom</h1>
10
11
12 </body>
13 </html>
14
```

Perhatikan contoh di bawah ini. Sintaks CSS diterapkan ke elemen h1 melalui atribut style=" ". Properti CSS yang disebut warna memiliki fungsi mengubah font yang digunakan untuk peringatan. Akibatnya, dalam contoh gaya CSS sebaris ini, tajuk h1 dibuat menggunakan jenis huruf yang diubah menjadi peringatan birch (Biru).

Dan ketika peluncuran selesai, hasilnya adalah sebagai berikut:



Gambar 5.2 hasil dari index.html in line style

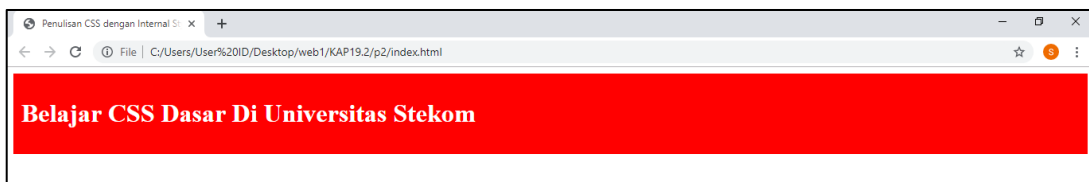
b) Internal / Embedded CSS Style

Penyisipan Sintaks Gaya Internal adalah teknik implementasi CSS yang memungkinkan integrasi file tunggal dengan file HTML atau PHP. Sintaks CSS dijelaskan dalam tag gaya dan diakhiri dengan tag /style>. Faktanya, tag HTML ditempatkan di dalam tag.

CSS Internal Contoh Gaya

```
index.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Penulisan CSS dengan Internal Style</title>
5 <style type="text/css">
6 #tulisan{
7 color: white;
8 }
9 .box{
10 background: red;
11 padding: 10px;
12 }
13 </style>
14 </head>
15 <body>
16 <div class="box">
17 <h1 id="tulisan">Belajar CSS Dasar Di Universitas Stekom</h1>
18 </div>
19 </body>
20 </html>
```

Perhatikan contoh penataan CSS internal di atas. sintaks css digabungkan dengan html dalam file yang sama. Sintaks CSS diimplementasikan dalam tag gaya. Padding CSS untuk sintaks memiliki fungsi menyorot teks di dalam saudara elemen. Dalam contoh ini, kami memberikan batas sekitar 10 piksel atau 10 piksel. CSS mengelola kelas pemilih dengan awalan "." dan mengelola id pemilih dengan awalan "#."



Gambar 5.3 hasil dari index.html internal embedded CSS style

c) External Style

Sintaks CSS dengan Gaya Eksternal adalah teknik penulisan yang dapat bekerja dengan file CSS dan HTML. Cara terbaik untuk menggunakan CSS adalah dengan menggunakan teknik penulisan ini karena ini mencegah kode program kita rusak karena sintaks untuk CSS ada di file yang sama dengan kode kita. File CSS dan HTML ditautkan bersama menggunakan gaya eksternal (contoh CSS):

```
<link rel="stylesheet" type="text/css" href="file css anda">
```

Untuk membuat contoh CSS menggunakan gaya eksternal, gabungkan satu file html dengan satu file CSS. Pada bagian ini, saya telah membuat sebuah file dengan memberi nama yaitu index.html beserta style.css.



Gambar 5.4 Contoh file html dan CSS

index2.html

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Penulisan CSS dengan External Style</title>
5     <link rel="stylesheet" type="text/css" href="style.css">
6   </head>
7
8   <body>
9     <div class="box">
10      <h1 id="tulisan">Belajar CSS Dasar Di Universitas Stekom</h1>
11    </div>
12  </body>
13 </html>

```

style.css

```

1 #tulisan{
2   color: white;
3 }
4 .box{
5   background: blue;
6   padding: 10px;
7 }

```

Jika di buka hasilnya akan seperti di bawah ini



Gambar 5.5 Eksternal CSS

5.3 MEMULAI CSS

a) MENGUBAH BAGROUND DENGAN CSS

Background atau batas bawah header website hanyalah dua elemen yang harus didesain agar menghasilkan header website yang terlihat tajam dan kekinian. Ada berbagai metode untuk mengubah latar belakang situs web atau tajuknya. Memanfaatkan gambar sebagai latar belakang

situs web dimungkinkan. Jika menggunakan gambar bukan yang Anda inginkan, Anda juga dapat mengubah warna tampilan latar menggunakan CSS jika Anda tidak ingin menggunakannya. Gunakan properti latar belakang CSS untuk mengubah warna latar belakang. Selain digunakan untuk mengubah backdrop body website, properti background juga dapat digunakan untuk mengubah gambar background dari elemen HTML tertentu.

Cara mengganti background dengan css

Dalam pelajaran CSS, metode untuk menghapus latar belakang menggunakan CSS akan dibahas. cara menghilangkan background dengan css. Lanjutkan proses memodifikasi CSS untuk header. buat file HTML baru dengan nama unik Anda sendiri. Di lokasi ini saya membuatnya dengan nama index3.html. lalu buat file CSS dengan nama yang sama seperti Anda, tetapi saya sudah memberinya nama style3.css dalam contoh ini.

1. **Mengubah warna background dengan CSS**

Perhatikan informasi di bawah ini tentang cara mengubah warna latar belakang menggunakan CSS.

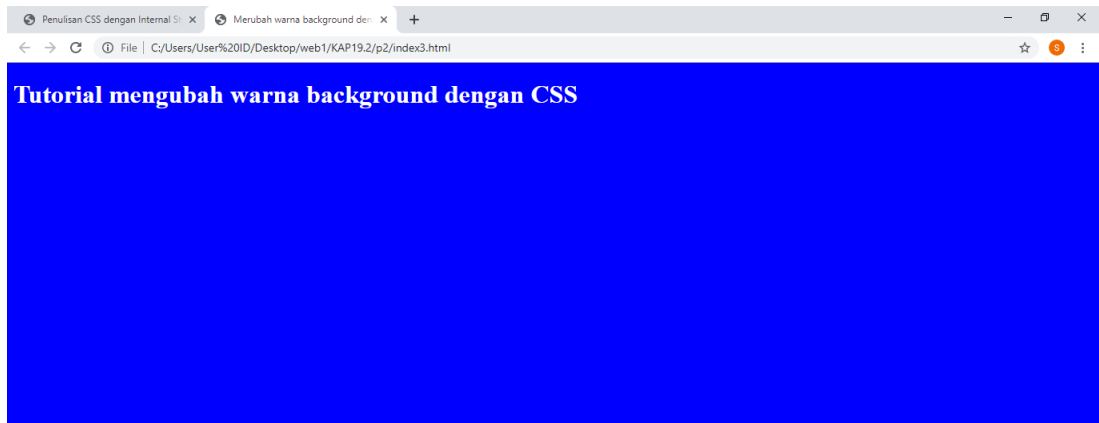
Index3.html

```
index3.html x style3.css x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Merubah warna background dengan CSS</title>
5 <link rel="stylesheet" type="text/css" href="style3.css">
6 </head>
7
8 <body>
9 <h1>Tutorial mengubah warna background dengan CSS</h1>
10 </body>
11 </html>
```

style3.css

```
index3.html x style3.css x
1 body{
2 background: blue;
3 color: white;
4 }
```

Hasilnya



Gambar 5.6 Mengubah Warna Background

Pada kode CSS diatas, kita menentukan body yang akan dimodifikasi. Kami kemudian menentukan warna latar belakang menjadi biru (biru), dan warna font menjadi putih (putih).

2. *Mengubah gambar background dengan CSS*

Akan dijelaskan pada bagian berjudul "Mengubah Tampilan Website dengan CSS" cara menggunakan gambar untuk mengubah tampilan header website. Untuk melakukannya, Anda harus menyediakan file grafik untuk digunakan sebagai gambar latar situs web. Dalam hal ini, saya memberikan gambar menghadap ke belakang yang akan saya gunakan sebagai foto sepia. Untuk mencegah hal ini terjadi, tempatkan file gambar dalam satu folder dengan file html dan CSS. Nama file gambarnya adalah stekom.png.



Gambar 5.7 File html gambar dan css

index4.html

```
index4.html x style4.css
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Membuat background dengan gambar</title>
5 <link rel="stylesheet" type="text/css" href="style4.css">
6 </head>
7
8 <body>
9 <h1>Membuat background gambar dengan CSS</h1>
10 </body>
11 </html>
```

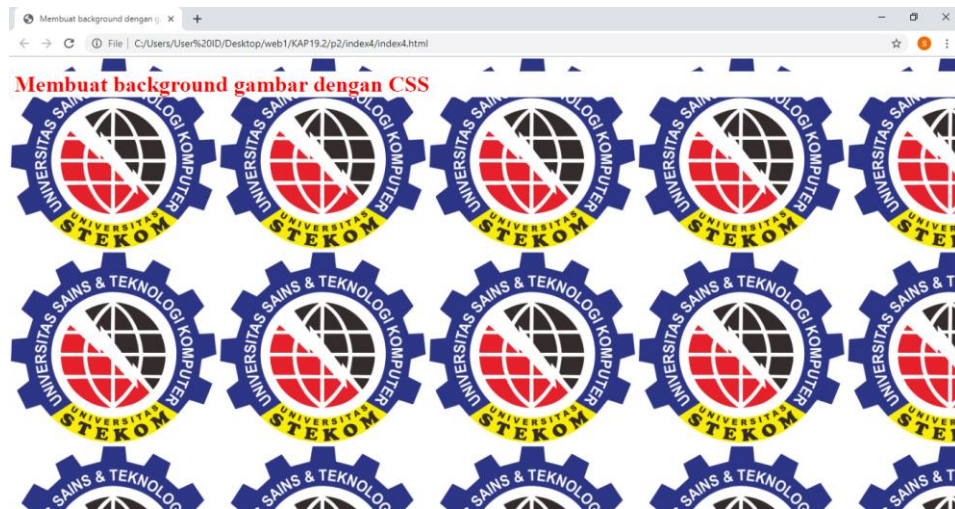
style4.css

```
index4.html x style4.css x
1 body{
2   background: url('stekom.png');
3   background-size: 300px;
4   color: #fff;
5 }
6 h1{
7   color: red;
8   background-color: white;
9 }
```

Perhatikan sintaks CSS di atas. Gunakan sintaks berikut untuk membuat gambar latar:

background: url('stekom.png');

Hasilnya



Gambar 5.8 Mengubah Background dengan gambar

The game has successfully been used as a backdrop halaman. Silahkan ber Eksperimen di rubah nilai pixel atau mengganti tempat penyimpanan style.css dan juga mengganti warna serta gambar baground dari tutorial diatas.

b) HYPERLINK DENGAN CSS

Hyperlink dibuat untuk mengubah halaman target saat diklik. Dalam HTML, hyperlink atau kalimat dibuat dengan tag "a" dan diakhiri dengan tag "/a". Dan menggunakan CSS untuk membuat, memodifikasi, atau membuat hyperlink cukup sederhana. Dalam kursus pembelajaran CSS ini, prosedur pembuatan hyperlink yang disempurnakan dengan CSS akan dibahas. Lanjutkan dengan menambahkan peringatan pada tautan, mengubah bentuk tautan menjadi tombol, membuat teks hyperlink menjadi kuning saat diklik, dan seterusnya.

Ada dua status yang dikontrol oleh HTML dan dapat diubah menggunakan CSS. Yaitu:

- Link : Ini adalah koneksi default yang aktif.
- Visited : ini adalah status link yang sudah dibuka.
- Hover : adalah status tautan saat penunjuk mouse digerakkan di atasnya.
- Active : status tautan atau hyperlink saat diklik.

Sebaliknya, status penggunaan atau pemeliharaan tautan dapat dinyatakan menggunakan sintaks berikut:

- a:link = untuk link biasa
- a:visited = mewakili status tautan yang baru saja diklik.
- a:hover = Saat mouse digerakkan di atas tautan, status tautan itu ditunjukkan oleh atribut.
- a:aktif = Status tautan atau hyperlink saat diklik ditunjukkan dengan atribut.

Contoh

index.html

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Hyperlink CSS</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <a class="link" target="_blank" href="https://stekom.ac.id/">Klik di sini</a>
9 </body>
10 </html>

```

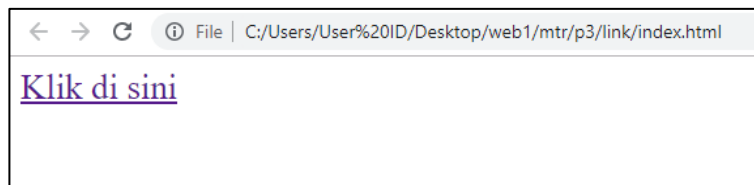
style.css

```

1 .link{
2   font-size: 20pt;
3 }
4 .link:hover{
5   color: red;
6 }
7

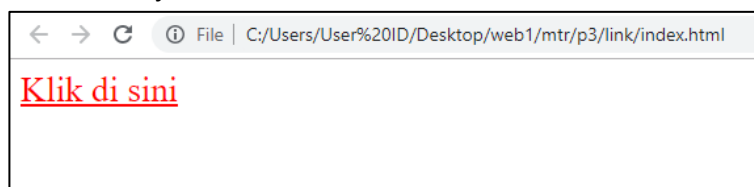
```

Hasil ketika tidak ditunjuk kursor



Gambar 5.9 Hyperlink dengan CSS

Ketika di tunjuk kursor



Gambar 5.10 Hyperlink dengan CSS ketika di hover cursor

Silahkan Anda coba di bawah ini

index2.html

```
index2.html x style2.css x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Hyperlink CSS</title>
5   <link rel="stylesheet" type="text/css" href="style2.css">
6 </head>
7 <body>
8   <a class="contoh-link" target="_blank" href="https://stekom.ac.id/">Klik di
  sini</a>
9 </body>
10 </html>
```

style2.css

```
index2.html x style2.css x
1 .contoh-link,
2 .contoh-link:link,
3 .contoh-link:active,
4 .contoh-link:visited{
5   font-size: 20pt;
6   background: blue;
7   color: white;
8   text-decoration: none;
9   padding: 10px;
10  font-family: sans-serif;
11 }
12 .contoh-link:hover{
13   background: red;
14 }
```

Amatilah hasilnya dan silahkan bereksperimen mengganti value dari masing – masing style nya.

BAB 6

MENGATUR FONT, FORMAT TEKS, MENGUBAH LIST DAN MENGATUR BORDER DENGAN CSS

TUJUAN :

- a. Mahasiswa mampu mengatur font dengan CSS
- b. Mahasiswa mampu mengatur format teks dengan CSS
- c. Mahasiswa mampu mengubah list dengan CSS
- d. Mahasiswa mampu mengubah border dengan CSS

Overview :

Desain web merupakan hal yang sangat penting di pertimbangkan. Semakin bagus desain yang di berikan, maka konsumen ataupun pengunjung web akan merasa tertarik untuk mengetahui isi web dan nyaman dalam membuka web tersebut.

Dengan menggunakan CSS kita dapat mengatur font, format teks, mengubah gaya list dan border seperti yang akan di pelajari dalam BAB 6 disini.

6.1 MENGATUR FONT DENGAN CSS

Tutorial saat ini akan menjelaskan cara menyesuaikan jenis huruf menggunakan CSS. seperti Merubah, hiasan Ukuran, Warn, dan Typefaces (garis miring, huruf tebal, tulisan miring).

Beberapa sintaks CSS yang digunakan untuk mengubah font adalah:

- font-size digunakan untuk mengubah ukuran font.

Tabel 6.1 font size

No	Value	Value
1	x-small	x-large
2	small	large
3	xx-small	xx-large
4	smaller	larger
5	medium	initial
6	..px, ..em, dll	..%

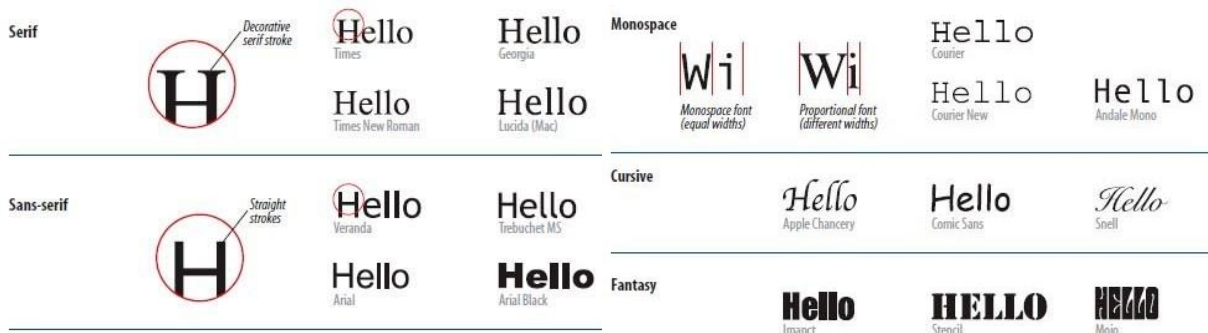
- font-weight : digunakan untuk mengubah font pada ketebalan

Tabel 6.2 font weight

No	Value	Deskripsi
1	bold	membuat font pada teks/karakter menjadi tebal
2	bolder	membuat font pada teks/karakter menjadi lebih tebal
3	lighter	membuat font pada teks/karakter menjadi lebih tipis
4	normal	membuat font pada teks/karakter menjadi normal
5	initial	mengatur value dari properti menjadi default
6	100,200,....,900	mengatur ketebalan font pada teks/karakter berdasarkan ukuran nilai yang dipilih

- font-family : untuk mengubah jenis font (fantasy, inherit, serif, monospace)

Tabel 6.3 font family



Dalam pemilihan font-family Anda juga bisa langsung mengetikkan jenis font seperti pada pilihan font di MS Office dengan di himpit tanda petik, contoh

```
font-family: 'algerian';
```

- font-style : digunakan untuk menulis ulang karakter dalam jenis huruf.
- color : digunakan untuk membuat jenis huruf peringatan atau warna huruf.

Berikut terdapat contoh cara mengubah font dengan css.

index.html

```
index.html x style.css x
2 <html>
3 <head>
4   <title>Font CSS</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <h1 class="tulisan_satu">1. Belajar CSS Dasar di Universitas Stekom</h1>
9   <hr>
10  <h1 class="tulisan_dua">2. Belajar CSS Dasar di Universitas Stekom</h1>
11  <hr>
12  <h1 class="tulisan_tiga">3. Belajar CSS Dasar di Universitas Stekom</h1>
13  <hr>
14 </body>
15 </html>
```

style.css

```
index.html x style.css x
1 .tulisan_satu{
2   color: red;
3   font-family: monospace;
4   font-style: normal;
5 }
6
7 .tulisan_dua{
8   color: green;
9   font-size: 24pt;
10  font-style: italic;
11  font-family: Impact, Charcoal, sans-serif;
12 }
13
14 .tulisan_tiga{
15  color: blue;
16  font-weight: bold;
17  font-style: oblique;
18  font-family: 'algerian';
19 }
```

Hasilnya dapat dilihat dalam gambar 6.1 di bawah ini



Gambar 6.1 Hasil mengatur font dengan css.

Anda dapat melihat contoh cara mengubah font menggunakan CSS di atas. Anda dapat menggunakan karakter "color" untuk mengubah warna font, sedangkan "font-family" dan "font-style" digunakan untuk mengubah jenis font (miring, normal, dan opsi lainnya). Font-size digunakan untuk mengubah ukuran font. Demikian pula, font-weight digunakan untuk menyorot sisi font tebal.

Note : Anda dapat melihat pilihan di <https://www.w3schools.com/css/>

6.2 FORMAT TEXT DENGAN CSS

Dalam buku ini akan diperlihatkan cara menggunakan CSS untuk memformat atau memodifikasi teks sehingga akan terlihat seperti yang Anda inginkan dan berfungsi sesuai kebutuhan. Misalnya, Anda dapat mengubah justifikasi teks menjadi justify atau yang biasa Anda sebut "rata kanan", mengubah indent spacing teks menjadi justified, mengubah kursif teks menjadi huruf besar, huruf kecil, tebal, atau miring, mengubah warna teks menjadi merah, dan ubah desain teks menjadi ornamental. Silakan baca informasi di bawah ini tentang mengubah format teks yang dapat dilakukan dengan menggunakan CSS.

Ada satu konstruksi CSS yang dapat digunakan untuk mengubah format teks, yaitu sintaks beberapa.

- **color**

Apa pun yang dapat ditentukan sebagai peringatan atau kode peringatan digunakan untuk mengubah teks peringatan.

- **text-align**

Untuk mengubah perataan teks atau konten lainnya, tersedia nilai berikut: tengah untuk membuat teks rata tengah, kiri untuk membuat teks rata kiri, kanan untuk membuat teks rata kanan, dan justify untuk membuat teks rata kanan dan rata kiri.

- **text-decoration**

Ketika digunakan untuk mengubah dekorasi teks, nilai yang tersedia adalah none untuk membuat teks tanpa dekorasi, underline untuk membuat teks memiliki dekorasi di pojok kiri atas, line-through untuk membuat dekorasi yang mencoreng di dalam teks, dan garis bawah untuk membuat dekorasi di dalam teks. pojok kanan bawah (garis bawah).

- **text-transform**

When used to change the size of the capital letters in a text, the options available are uppercase for larger capital letters, lowercase for smaller capital letters, and capitalized for larger capital letters at the beginning of each word.

- **text-indent**

Nilai yang dapat digunakan untuk mengubah perataan sebaris teks dapat berupa piksel atau nilai lainnya tergantung situasi.

- **text-spacing**

Digunakan untuk mengatur perataan karakter dalam teks, dengan nilai dalam format isi yang terdiri dari piksel dan nilai lainnya.

- **word-spacing**

Digunakan untuk mengurangi jarak antar kata dalam teks, nilai isi juga memiliki gambar jam.

- **line-height**

Digunakan untuk mengubah spasi bar-to-bar pada nilai teks dengan nilai desimal satu digit.

- **text-shadow**

Saat digunakan untuk mengubah warna efek bayang teks, nilai yang dipilih pertama menetapkan nilai jarak di bagian atas, yang kedua menetapkan jarak di bagian bawah, dan yang ketiga menetapkan peringatan. 2 px kali 5 px biru sebagai contoh.

- **vertical-align**

Kiri, kanan, dan tengah adalah glif yang digunakan untuk membuat rata-kiri, rata-atas, dan rata-bawah, masing-masing, untuk teks vertikal.

Untuk mendemonstrasikan cara mengubah format teks menggunakan CSS. membuat file CSS dan HTML. Di sini, saya telah membuat file style.css dan index.html.

a) Cara mengubah warna text dengan CSS

Perhatikan contoh berikut untuk mengubah peringatan teks menggunakan CSS.

index.html

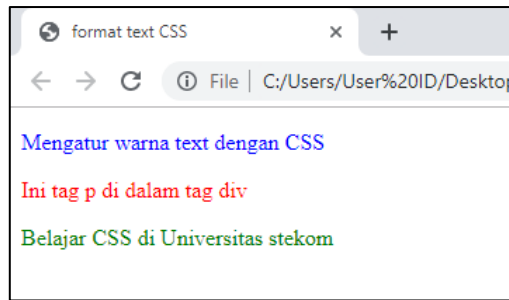
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>format text CSS</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7
8 <body>
9   <p class="tulisan_warna">Mengatur warna text dengan CSS</p>
10
11   <div>
12     <p>Ini tag p di dalam tag div</p>
13   </div>
14
15   <p>Belajar CSS di Universitas stekom</p>
16 </body>
17 </html>
18
```

style.css

```
1 .tulisan_warna{
2   color: blue;
3 }
4
5 div p{
6   color: red;
7 }
8 p {
9   color: green;
10 }
```

Seperti yang bisa dilihat pada contoh di bawah ini, terdapat tag paragraf dengan class tulisan warna. Lalu, matikan peringatan merah dengan CSS. Kemudian terdapat tag p didalam tag div serta terdapat juga tag p di luar tag div. jika ingin mengatur ssebuah tag yang terdapat pada dalam tag lain (tag p di dalam tag div) selector yang kita gunakan adalah tag yang awal di ikuti (spasi) dengan tag yang di dalam.

Hasilnya dapat dilihat dalam gambar 6.2 dibawah ini



Gambar 6.2 Hasil Mengubah Warna Teks dengan CSS

b) Cara menggunakan text align CSS

index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Format text CSS</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <p class="t_kiri">Mengatur format text dengan CSS</p>
9   <p class="t_kanan">Mengatur format text dengan CSS</p>
10  <p class="t_tengah">Mengatur format text dengan CSS</p>
11  <p class="t_justify"> Belajar tutorial CSS dasar di Universitas Stekom.
12  Mengatur format text dengan CSS. belajar CSS dasar di Universitas Stekom
13  Semarang tahun 2020. Lorem ipsum dolor sit amet, consectetur adipiscing elit,
14  sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad
15  minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
16  commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit
17  esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
18  non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
19 </p>
20 </body>
21 </html>
```

style.css

```
1 .t_kiri{
2   text-align: left;
3   color: red;
4 }
5
6
7 .t_kanan{
8   text-align: right;
9   color: green;
10 }
11
12
13 .t_tengah{
14   text-align: center;
15   color: blue;
16 }
17
18
19 .t_justify{
20   text-align: justify;
21 }
```

Hasil dapat dilihat pada gambar 6.3 di bawah ini



Gambar 6.3 Hasil dari menggunakan text align CSS

c) Contoh mengatur format text lainnya

index.html

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Format text CSS</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <p class="t_satu">Mengatur format text dengan CSS</p>
9   <p class="t_dua">Mengatur format text dengan CSS</p>
10  <p class="t_tiga">Mengatur format text dengan CSS</p>
11  <p class="t_empat"> Belajar tutorial CSS dasar di Universitas Stekom.
12  Mengatur format text dengan CSS. belajar CSS dasar di Universitas Stekom Semarang
tahun 2020. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis
aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat
nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui
officia deserunt mollit anim id est laborum.
13  </p>
14 </body>
15 </html>

```

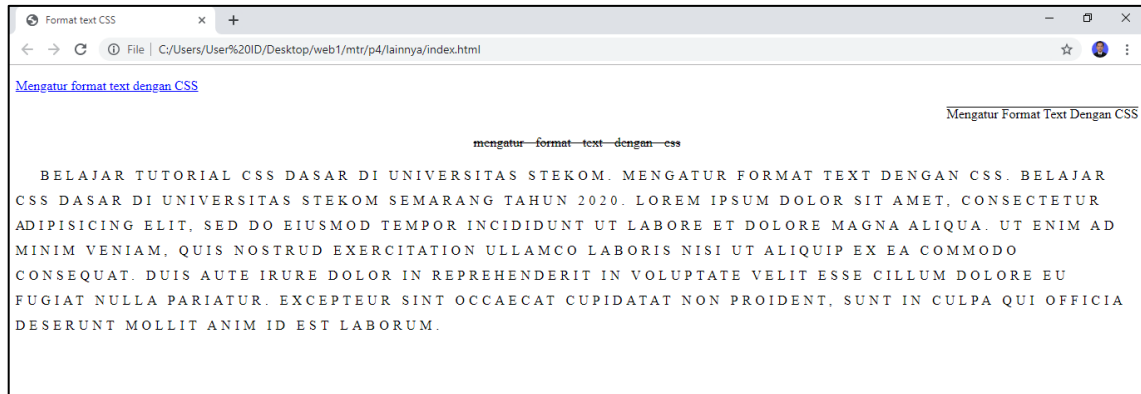
style.css

```

1 .t_satu{
2   color: blue;
3   text-decoration: underline;
4 }
5
6
7 .t_dua{
8   text-align: right;
9   text-transform: capitalize;
10  text-decoration: overline;
11 }
12
13
14 .t_tiga{
15  text-align: center;
16  text-transform: lowercase;
17  text-decoration: line-through;
18  word-spacing: 10px;
19 }
20
21
22 .t_empat{
23  text-transform: uppercase;
24  text-indent: 30px;
25  line-height: 30px;
26  letter-spacing: 5px;
27 }

```

Hasilnya dapat dilihat pada gambar di bawah ini



Gambar 6.4 hasil dari mengubah format teks dengan CSS

6.3 MODIFIKASI LIST DENGAN CSS

CSS has the capability to manage or modify HTML source code. CSS cukup membantu jika ingin membuat list dengan model yang diinginkan, seperti yang terdiri dari elemen angular, titik, romawi, dan lainnya. Dengan menggunakan CSS, Anda dapat memodifikasi format header daftar untuk mengelola daftar HTML. Atribut "list-style-type", yang menunjukkan gaya daftar, dapat digunakan untuk mengatur gaya daftar HTML dengan CSS.

Pertimbangkan contoh berikut untuk memodifikasi daftar HTML dengan CSS.

Index.html

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Mengubah List</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <h1>Tutorial cara mengubah list dengan CSS di <br/> Universitas Stekom</h1>
9 <!-- unordered list -->
10  <ul class="makanan">
11    <li>Bakso</li>
12    <li>Mie Goreng</li>
13    <li>Sate Padang</li>
14    <li>Rujak</li>
15  </ul>
16  <ul class="minuman">
17    <li>Soft drink</li>
18    <li>Teh manis</li>
19    <li>Kopi</li>
20    <li>Jus jeruk</li>
21    <li>Susu</li>
22  </ul>
23
24
25 <!-- ordered list -->
26  <ol class="alamat">
27    <li>Semarang</li>
28    <li>Kendal</li>

```



```

29     <li>Kudus</li>
30     <li>Ungaran</li>
31 </ol>
32 <ol class="mobil">
33     <li>Sedan</li>
34     <li>Mini Bus</li>
35     <li>Truk</li>
36     <li>Pick Up</li>
37 </ol>
38 </body>
39 </html>
40 |

```

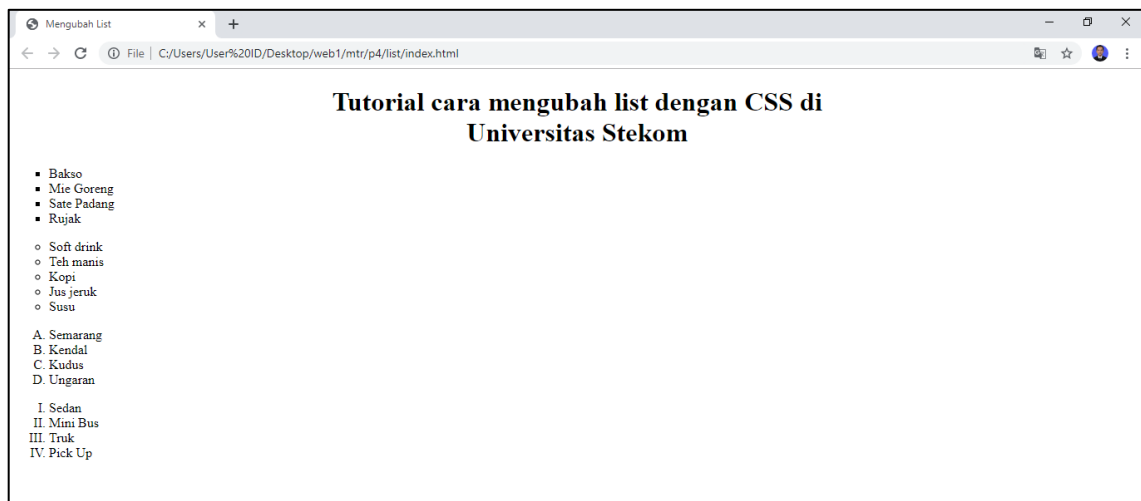
Style.css

```

1 h1{
2     text-align: center;
3 }
4
5 ul.makanan{
6     list-style-type: square; /* list dengan bentuk square */
7 }
8
9 ul.minuman{
10     list-style-type: circle; /* list dengan bentuk lingkaran */
11 }
12
13 ol.alamat{
14     list-style-type: upper-alpha; /* list dengan bentuk alpha */
15 }
16
17 ol.mobil{
18     list-style-type: upper-roman; /* list dengan bentuk romawi */
19 }
20 |

```

Hasilnya



Gambar 6.5 modifikasi list dengan CSS

CSS Belajar List Building Menggunakan CSS. Dapat dilihat dari contoh di bawah ini bahwa menggunakan properti dan nilai CSS akan menghasilkan daftar dengan format persegi:

```
list-style-type: square; /* list dengan bentuk square */
```

Anda dapat menggunakan fungsi properti dan nilai CSS untuk membuat daftar dengan bentuk lingkaran kecil.

```
list-style-type: circle; /* list dengan bentuk lingkaran */
```

Gunakan properti dan nilai CSS untuk membuat daftar dengan format abjad.

```
list-style-type: upper-alpha; /* list dengan bentuk alphabet */
```

Gunakan CSS properti dan nilai untuk membuat daftar dengan format angka romawi:

```
list-style-type: upper-roman; /* list dengan bentuk romawi */
```

sedikit tambahan untuk menghilangkan tanda list anda bisa menggunakan property dan value sebagai berikut:

```
list-style-type: none; /* menghilangkan tanda list */
```

6.4 MEMODIFIKASI BORDER DENGAN CSS

Dalam buku ini terdapat pembahasan mengenai manipulasi perbatasan dengan menggunakan perintah CSS. CSS memberi Anda opsi untuk menyesuaikan warna, ukuran dan gaya perbatasan secara manual. Misalnya border yang terbuat dari titik-titik, strip, dan pakaian bias model.

a) MENGUBAH BORDER STYLE

Ada beberapa gaya atau model yang tersedia di CSS untuk membuat grafik. Ada banyak gaya dan gaya lain yang dapat diterapkan pada halaman menggunakan CSS, termasuk yang terbuat dari titik-titik, putus-putus, dan garis ganda. Untuk membuat situs web yang juga menyediakan gaya nilai yang diinginkan, gunakan properti gaya batas CSS. Berikut contoh nya, silahkan di perhatikan!

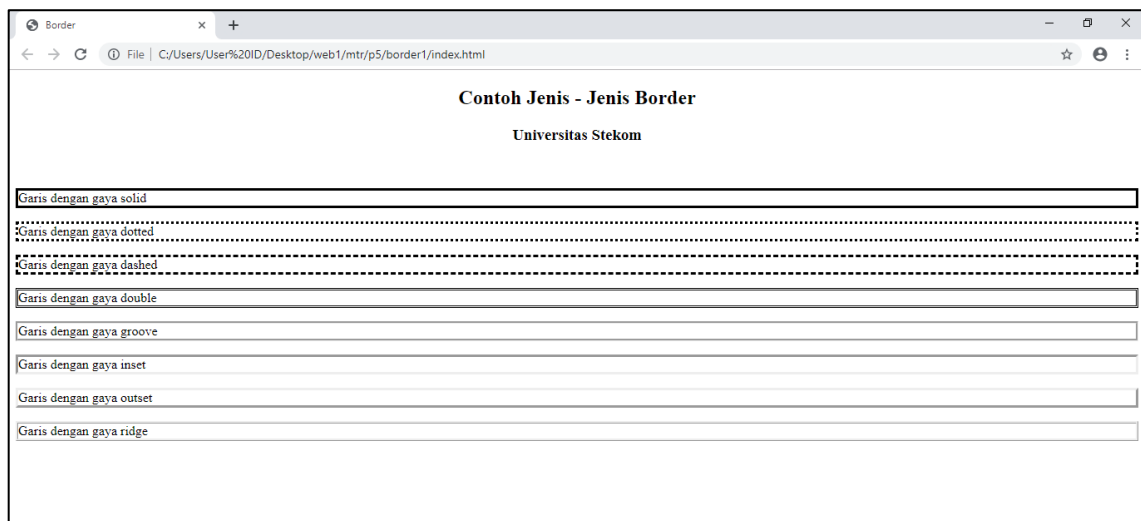
index.html

```
index.html x style.css x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Border</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <h2>Contoh Jenis - Jenis Border</h2>
9   <h3>Universitas Stekom</h3>
10  <br>
11  <p id="garis_1">Garis dengan gaya solid</p>
12  <p id="garis_2">Garis dengan gaya dotted</p>
13  <p id="garis_3">Garis dengan gaya dashed</p>
14  <p id="garis_4">Garis dengan gaya double</p>
15  <p id="garis_5">Garis dengan gaya groove</p>
16  <p id="garis_6">Garis dengan gaya inset</p>
17  <p id="garis_7">Garis dengan gaya outset</p>
18  <p id="garis_8">Garis dengan gaya ridge</p>
19 </body>
20
```

style.css

```
1 h2, h3 {
2     text-align: center;
3 }
4 #garis_1{
5     border-style:solid;
6 }
7 #garis_2{
8     border-style: dotted;
9 }
10 #garis_3{
11     border-style: dashed;
12 }
13 #garis_4{
14     border-style: double;
15 }
16 #garis_5{
17     border-style: groove;
18 }
19 #garis_6{
20     border-style: inset;
21 }
22 #garis_7{
23     border-style: outset;
24 }
25 #garis_8{
26     border-style: ridge;
27 }
28
```

Hasilnya



Gambar 6.6 hasil mengubah border style dengan CSS

Sedikit tambahan informasi, jika Anda cermati dalam selector style.css yang menargetkan pada **h2** dan **h3**, pen - style an dapat di laksanakan secara bersamaan. Dengan memanggil tag 1 dan tag 2 yang di pisahkan dengan tanda koma = **h2, h3 { isi style 1; }**

b) MENGATUR UKURAN BORDER

Anda dapat menggunakan properti CSS borderwidth untuk mengubah lebar border. Berikut adalah contoh cara menentukan lebar batas properti untuk membuat bentuk garis.

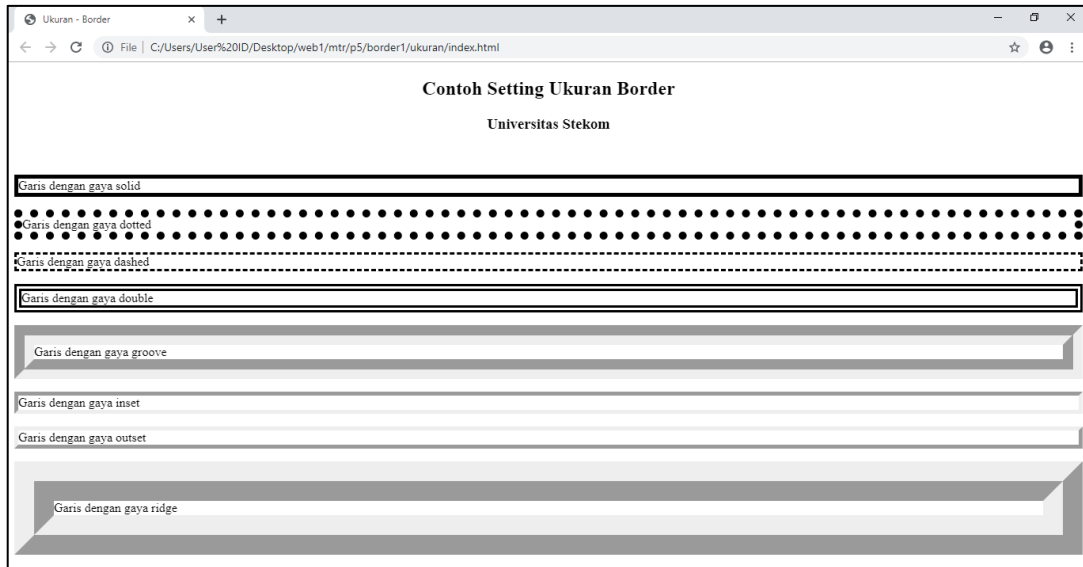
index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Ukuran - Border</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <h2>Contoh Setting Ukuran Border</h2>
9   <h3>Universitas Stekom</h3>
10  <br>
11  <p id="garis_1">Garis dengan gaya solid</p>
12  <p id="garis_2">Garis dengan gaya dotted</p>
13  <p id="garis_3">Garis dengan gaya dashed</p>
14  <p id="garis_4">Garis dengan gaya double</p>
15  <p id="garis_5">Garis dengan gaya groove</p>
16  <p id="garis_6">Garis dengan gaya inset</p>
17  <p id="garis_7">Garis dengan gaya outset</p>
18  <p id="garis_8">Garis dengan gaya ridge</p>
19 </body>
20 </html>
```

style.css

```
1 #garis_1{
2   border-style:solid; border-width: 5px;
3 }
4 #garis_2{
5   border-style: dotted; border-width: 10px;
6 }
7 #garis_3{
8   border-style: dashed; border-width: 3px;
9 }
10 #garis_4{
11   border-style: double; border-width: 9px;
12 }
13 #garis_5{
14   border-style: groove; border-width: 25px;
15 }
16 #garis_6{
17   border-style: inset; border-width: 5px;
18 }
19 #garis_7{
20   border-style: outset; border-width: 5px;
21 }
22 #garis_8{
23   border-style: ridge; border-width: 50px;
24 }
25 h2, h3 {
26   text-align: center;
27 }
28
```

Hasilnya dapat dilihat pada halaman berikutnya



Gambar 6.7 hasil dari mengatur ukuran border

c) MEMBERI WARNA PADA BORDER

Gunakan properti CSS border-color untuk menambahkan peringatan ke border. Ini adalah contoh pembuatan peringatan pada grafik dengan menggunakan CSS.

index.html

```

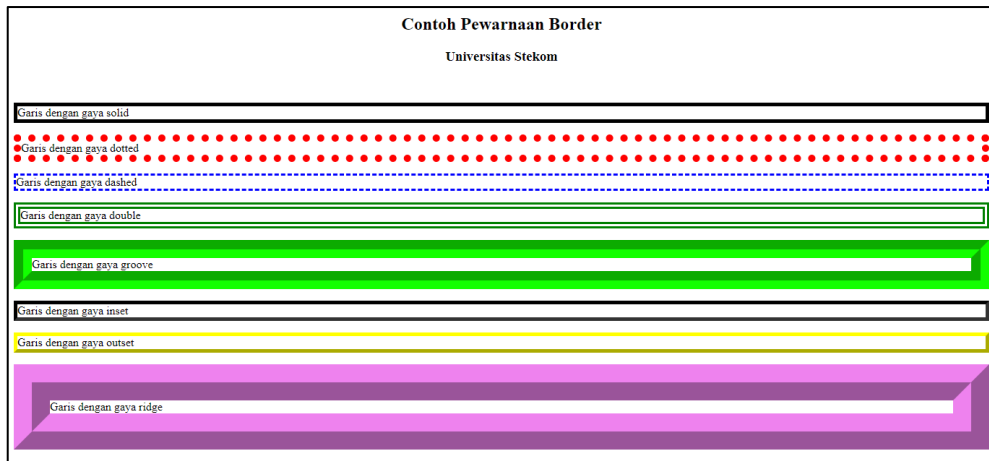
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Warna - Border</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <h2>Contoh Pewarnaan Border</h2>
9   <h3>Universitas Stekom</h3>
10  <br>
11  <p id="garis_1">Garis dengan gaya solid</p>
12  <p id="garis_2">Garis dengan gaya dotted</p>
13  <p id="garis_3">Garis dengan gaya dashed</p>
14  <p id="garis_4">Garis dengan gaya double</p>
15  <p id="garis_5">Garis dengan gaya groove</p>
16  <p id="garis_6">Garis dengan gaya inset</p>
17  <p id="garis_7">Garis dengan gaya outset</p>
18  <p id="garis_8">Garis dengan gaya ridge</p>
19 </body>
20 </html>

```

style.css

```
1 h2, h3 {
2     text-align: center;
3 }
4 #garis_1{
5     border-style:solid;
6     border-width: 5px;
7 }
8 #garis_2{
9     border-style: dotted;
10    border-width: 10px;
11    border-color:red;
12 }
13 #garis_3{
14    border-style: dashed;
15    border-width: 3px;
16    border-color:blue;
17 }
18 #garis_4{
19    border-style: double;
20    border-width: 9px;
21    border-color:green;
22 }
23 #garis_5{
24    border-style: groove;
25    border-width: 25px;
26    border-color:#12ff00;
27 }
28 #garis_6{
29    border-style: inset;
30    border-width: 5px;
31    border-color:#333333;
32 }
33 #garis_7{
34    border-style: outset;
35    border-width: 5px;
36    border-color:yellow;
37 }
38 #garis_8{
39    border-style: ridge;
40    border-width: 50px;
41    border-color: violet;
42 }
43
```

Hasilnya



Gambar 6.8 hasil dari membuat warna border dengan CSS

d) CARA CEPAT MEMANIPULASI BORDER DENGAN CSS

Bagaimana mengelola border maupun membuat garis border dengan Teknik yang cepat melalui pemanfaatan CSS, properti CSS border secara terus menerus dapat menjadi jawaban dari masalah ini. demikian pula secara terus menerus gisi nilai ukuran, garis dan jenis peringatan. Pertimbangkan contoh berikut untuk mempelajari cara melintasi batas dengan cepat menggunakan CSS.

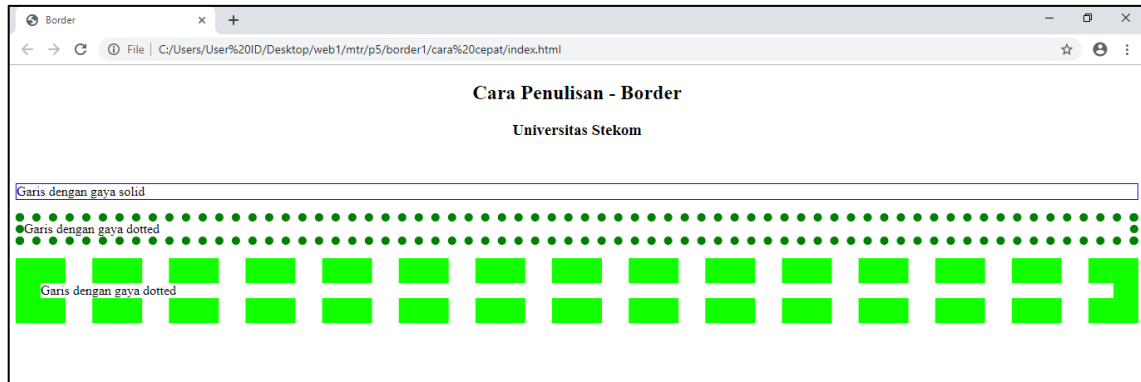
index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Border</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <h2>Cara Penulisan - Border</h2>
9   <h3>Universitas Stekom</h3>
10  <br>
11  <p id="garis_1">Garis dengan gaya solid</p>
12  <p id="garis_2">Garis dengan gaya dotted</p>
13  <p id="garis_3">Garis dengan gaya dotted</p>
14 </body>
15 </html>
```

style.css

```
1 #garis_1{
2   border:1px solid blue;
3 }
4 #garis_2{
5   border:10px dotted green;
6 }
7 #garis_3{
8   border:30px dashed #12ff00;
9 }
10 h2, h3 {
11   text-align: center;
12 }
```

Hasilnya pada halaman berikut ini



Gambar 6.9 hasil dari memanipulasi border dengan CSS

Perhatikan contoh sintaks CSS di bawah ini. Dimungkinkan untuk terus mencatat dimensi, jenis batas, dan peringatan yang diinginkan dalam batas properti untuk membangun batas dengan cepat.

border:2px solid blue;

Contoh di atas memiliki peringatan hitam putih pekat dan diposisikan pada batas lebar 1 piksel. Berikut ini adalah beberapa properti lain yang digunakan untuk mengubah batas menggunakan CSS:

- border-bottom, Mengatur garis yang terletak di bawah.
- border-bottom-color, Mengatur warna garis yang terletak di bawah.
- border-bottom-style, Mengatur style garis yang terletak di bawah.
- border-bottom-width, Mengatur ukuran garis yang terletak di bawah.
- border-color, Mengatur warna garis.
- border-left, Mengatur garis yang terletak di sebelah kiri.
- border-left-color, Mengatur warna garis yang terletak di sebelah kiri.
- border-left-style, Mengatur style garis yang terletak di sebelah kiri.
- border-left-width, Mengatur ukuran garis yang terletak di sebelah kiri.
- border-right, Mengatur garis yang terletak di sebelah kanan.
- border-right-color, Mengatur warna garis yang terletak di sebelah kanan.
- border-right-style, Mengatur style garis yang terletak di sebelah kanan.
- border-right-width, Mengatur ukuran garis yang terletak di sebelah kanan.
- border-style, Mengatur style garis.
- border-top, Mengatur garis yang terletak di sebelah atas.
- border-top-color, Mengatur warna garis yang terletak di sebelah atas.
- border-top-style, Mengatur style garis yang terletak di sebelah atas.
- border-top-width, Mengatur ukuran garis yang terletak di sebelah atas.
- border-width, Mengatur ukuran garis.

BAB 7

MENGATUR MARGIN, PADDING, POSITION FLOAT DAN TABEL DENGAN CSS

TUJUAN :

- a. Mahasiswa mampu mengatur margin dengan CSS
- b. Mahasiswa mampu mengatur padding dengan CSS
- c. Mahasiswa mampu mengatur position dengan CSS
- d. Mahasiswa mampu menampilkan web dengan teknik float dengan CSS
- e. Mahasiswa mampu mengatur gaya table dengan CSS

Overview :

Margin dan padding adalah dua elemen yang paling sering digunakan saat menggunakan CSS untuk mendesain situs web. Karena margin dan padding digunakan untuk mengubah sisi elemen yang bersebelahan dengan elemen lainnya.

Untuk mengubah posisi elemen HTML, pemosisian CSS digunakan. Kadang-kadang, kami ingin membuat atau mengubah posisi satu elemen dalam kaitannya dengan elemen lainnya. Oleh karena itu, dengan menggunakan properti position CSS, Anda dapat menentukan posisi elemen HTML saat ini sesuai dengan preferensi Anda.

Salah satu teknik yang paling sering digunakan dalam desain web adalah floating text. Pelampung digunakan untuk memindahkan elemen yang berorientasi horizontal.

Anda dapat memberikan gaya CSS untuk kebutuhan tabel karena juga akan diterjemahkan ke bahasa Inggris oleh BAB untuk membuat hasil yang menarik untuk tabel.

7.1 MARGIN DAN PADDING DENGAN CSS

a) Mengenal Margin pada CSS

Margin adalah bagian terluar dari elemen apa pun. Misalnya, jika Anda ingin menurunkan kecepatan elemen tertentu, Anda dapat melakukannya dengan menggunakan sintaks margin. Ada beberapa istilah yang berhubungan dengan margin, seperti margin atas, yang ditulis sebagai "margin-top" di CSS, margin bawah, atau batas di sekitar bagian bawah header, yang ditulis sebagai "margin-bottom", "margin-left," yang berfungsi sebagai border di sekitar elemen kiri, dan "margin-right," yang mengacu pada margin untuk bagian bawah header. Namun, jika Anda hanya menggunakan sintaks "margin" untuk hari ini saja, maka secara otomatis Anda akan mempengaruhi elemen at, bawah, kiri, dan kanan. Ini adalah contoh cara menggunakan margin di CSS.

index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Margin CSS</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <div class="box">
9     <h1>Ini adalah box</h1>
10  </div>
11  <div class="box-dua">
12    <h1>Ini adalah box dua</h1>
13  </div>
14 </body>
15 </html>
```

style.css

```
1 .box{
2   background: yellow;
3   height: 200px;
4   width: 300px;
5   margin: 20px;
6 }
7
8 .box-dua{
9   background: blue;
10  height: 100px;
11  width: 200px;
12  margin-left: 70px;
13 }
```

Hasilnya



Gambar 7.1 margin dengan CSS

Perhatikan contoh di bawah ini. Dalam kasus tata letak dua kolom, pengaturan margin kiri atau margin atas biasanya 70 piksel atau piksel, dan dalam kasus tata letak tiga kolom, margin atau batas di sekitar setiap baris biasanya 20 piksel.

b) Mengenal Padding pada CSS

Padding adalah komponen dari elemen tertentu. Anda dapat menggunakan pengisi sintaks untuk mengubah spasi baris pada elemen tertentu yang Anda kenal. mirip dengan margin dengan sisi-sisinya. seperti di atas, kiri, kanan, dan bawah. Padding di bagian atas, bawah, kiri, dan kanan elemen adalah contoh dari jenis padding ini. Padding di bagian atas, juga dikenal sebagai sisi dalam sebelah di bagian atas elemen, ditentukan dalam CSS sebagai "padding-top", sedangkan padding di bagian bawah, juga dikenal sebagai jarak di bagian bawah elemen, ditentukan sebagai "padding-bottom," "padding-left," dan "padding-right," yang berada di sisi luar. Jika memakai sintaks "padding" maka secara otomatis akan mengatur perataan elemen di header, body, dan footer. Ini merupakan contoh dari cara penggunaan padding di dalam CSS.

index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Padding CSS</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <h1>UNIVERSITAS STEKOM</h1>
9   <hr>
10  <div class="box">
11    <h1>Ini adalah box</h1>
12  </div>
13  <div class="box-dua">
14    <h1>Ini adalah box dua</h1>
15  </div>
16 </body>
17 </html>
```

style.css

```
1 h1{
2   text-align: center;
3 }
4
5 .box{
6   background: blue;
7   height: 100px;
8
9   width: 300px;
10  padding: 10px;
11 }
12
13 .box-dua{
14   background: red;
15   height: 100px;
16   width: 600px;
17   padding-left: 70px;
18 }
```

Hasilnya dapat dilihat pada halaman berikut ini



Gambar 7.2 padding dengan CSS

Dapat dilihat dari contoh diatas perbedaan ketika menggunakan padding dan padding-left. Sekarang Anda coba dengan menggunakan padding-right, padding-top, padding-bottom dan serta di coba tanpa menggunakan height dan width.

7.2 MENGATUR POSITION DENGAN CSS

Posisi dalam CSS digunakan untuk mengubah posisi elemen HTML. terkadang kami ingin membuat atau menggeser posisi satu elemen relatif terhadap elemen lainnya. Jadi, dengan menggunakan properti position CSS, Anda dapat menentukan posisi elemen HTML tertentu sesuai dengan preferensi Anda.

Ada beberapa properti CSS yang dapat digunakan untuk menunjukkan posisi elemen HTML tertentu:

- Static
- relative
- Fixed
- Absolute

Ini adalah praktik standar untuk menggunakan properti CSS lainnya, seperti atas, kiri, bawah, dan kanan, untuk membuat posisi atau mengubah posisi elemen individual. namun, properti yang disebutkan di atas tidak akan berfungsi jika lokasinya tidak segera diubah. karena posisi properti lain bergantung pada status mereka saat ini.

a) STATIC

Untuk menjadikan elemen sebagai elemen statis secara default, gunakan posisi statis. Elemen akan secara otomatis kembali ke posisi semula. Contoh penggunaan posisi statis

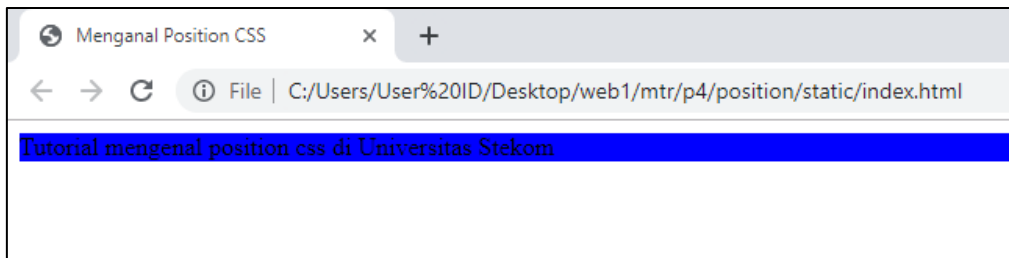
```
index.html x style.css x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Mengenal Position CSS</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <div class="kotak">Tutorial mengenal position css di Universitas Stekom</div>
9 </body>
10 </html>
```

Style.css

```
index.html x style.css x
1 .kotak{
2 background: blue; position: static;
3 }
4
```

Note : dalam penulisan perintah dapat di sambung ke kanan, namun setiap perintah ahrus di tutup dengan tanda ;

Hasilnya



Gambar 7.3 Positon static dengan CSS

b) RELATIVE

Setiap elemen HTML yang menggunakan position relative akan tetap pada posisi defaultnya.

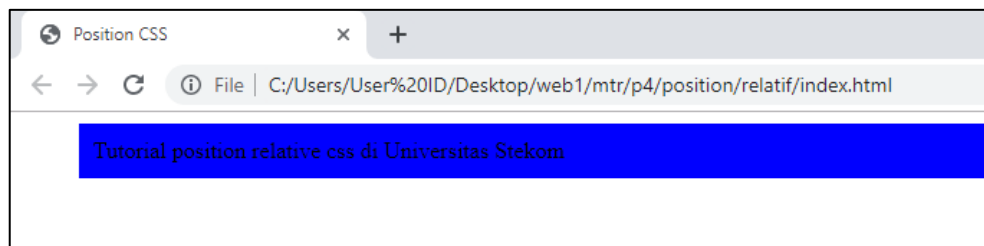
index.html

```
index.html x style.css x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Position CSS</title>
5 <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8 <div class="kotak">Tutorial position relative css di Universitas Stekom</div>
9 </body>
10 </html>
11
```

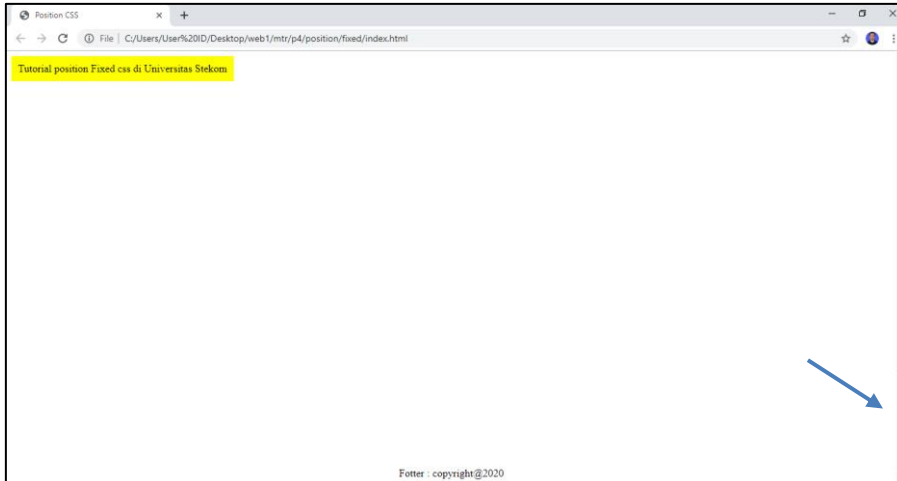
Style.css

```
index.html x style.css x
1 .kotak{
2 background: blue; position: relative; left: 40px; padding: 10px;
3 }
```

Hasilnya



Gambar 7.4 Positon relative dengan CSS



Gambar 7.6 Hasil Positon fixed dengan CSS pada posisi setelah di scroll ke bawah

d) ABSOLUTE

Elemen HTML yang menggunakan pemosisian absolut akan tumpang tindih dengan elemen HTML lainnya. Posisi adalah satu-satunya properti posisi CSS yang paling berguna. Metode terbaik adalah membuat menu drop-down menggunakan HTML dan CSS.

Index.html

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Position CSS</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <div class="kotak">Tutorial position css di Universitas Stekom</div>
9   <div class="kotak2">text pada absolute</div>
10 </body>
11 </html>

```

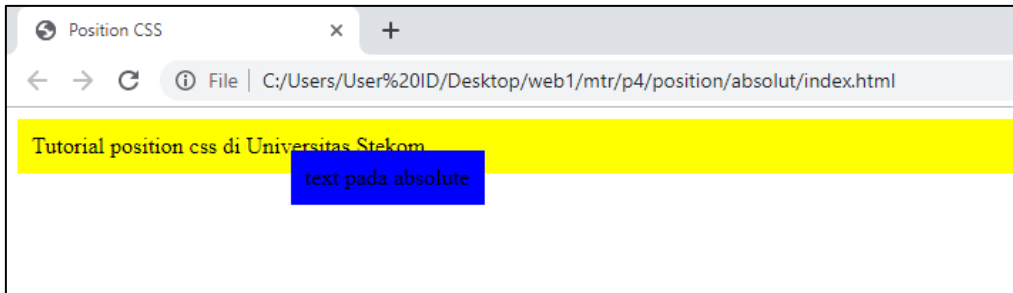
Style.css

```

1 .kotak{
2   background: yellow;
3   position: static;
4   left: 40px;
5   padding: 10px;
6 }
7
8 .kotak2{
9   background: blue;
10  position: absolute;
11  left: 200px;
12  padding: 10px;
13  top: 30px;
14 }

```

Hasilnya seperti berikut



Gambar 7.7 Positon absolute dengan CSS

7.3 TEKNIK FLOAT DENGAN CSS

Kebutuhan akan teknik floating dalam desain web adalah salah satu Teknik yang sering di gunakan dalam CSS. Pelampung digunakan untuk memutar elemen berorientasi horizontal. Ada empat kemungkinan nilai yang dapat digunakan untuk pelampung properti: **left**, **right**, **inherit** dan **none**.

Contoh yang paling sering menggunakan pelampung adalah saat mencoba membuat spanduk untuk situs web yang memiliki konten yang dikunci di sidebar. Berikut adalah contoh cara menggunakan properti float CSS.

index.html

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Float</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <h2>Tutorial Float dengan CSS </h2>
9   <h3>di Universitas Stekom</h3>
10  <div class="kotak">
11    
12    <p>
13      Belajar tutorial pemrograman web, di Universitas Stekom. tutorial belajar css
tentang penggunaan float di css. cara menggunakan float di css. cara membuat element
berbaris dengan float css. belajar css menggunakan float css. Belajar tutorial
pemrograman web, di Universitas Stekom. tutorial belajar css tentang penggunaan
float di css. cara menggunakan float di css. cara membuat element berbaris dengan
float css. belajar css menggunakan float css. Belajar tutorial pemrograman web, di
Universitas Stekom. tutorial belajar css tentang penggunaan float di css. cara
menggunakan float di css. belajar css menggunakan float css. Belajar tutorial
pemrograman web, di Universitas Stekom. tutorial belajar css tentang penggunaan
float di css. cara menggunakan float di css. cara membuat element berbaris dengan
float css. belajar css menggunakan float css. Belajar tutorial pemrograman web, di
Universitas Stekom. tutorial belajar css tentang penggunaan float di css. cara
menggunakan float di css. cara membuat element berbaris dengan float css. belajar css
menggunakan float css. Belajar tutorial pemrograman web, di Universitas Stekom.
tutorial belajar css tentang penggunaan float di css. cara menggunakan float di css.
cara membuat element berbaris dengan float css. belajar css menggunakan float
css. Belajar tutorial pemrograman web, di Universitas Stekom. tutorial belajar css
tentang penggunaan float di css. cara menggunakan float di css. cara membuat element
berbaris dengan float css. belajar css menggunakan float css.
14    </p>
15  </div>
16 </body>
17 </html>

```

Letakan file html, css dan gambar di dalam 1 folder



Gambar 7.8 letak html css dan gambar

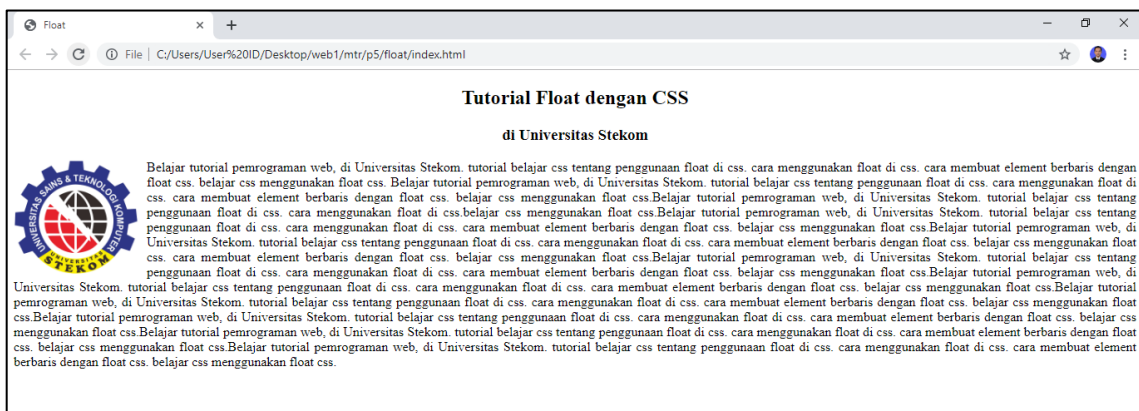
style.css

```

1  h2, h3{
2      text-align: center;
3  }
4  .gambar{
5      width: 150px; float: left;
6      margin-right: 10px;
7  }
8  p {
9      text-align: justify;
10 }
11

```

Hasilnya



Gambar 7.9 Teknik float dengan CSS

Perhatikan contoh di bawah ini. Untuk membuat gambar diturunkan ke kiri dan teks membungkus gambar, kita menerapkan float:left ke elemen gambar. Untuk informasi lebih lanjut, Anda dapat menggunakan properti CSS float:right dan inherit untuk melihat bagaimana berbagai nilai properti float berbeda satu sama lain.

7.4 DESAIN TABEL DENGAN CSS

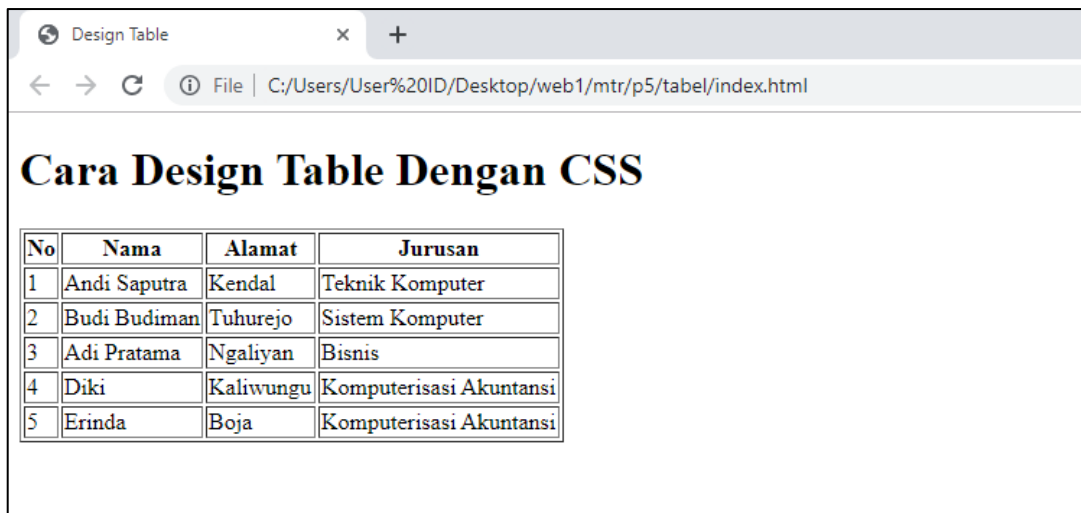
Pada saat pelajaran sebelumnya tentang pembuatan tabel HTML, banyak dari kita percaya bahwa tampilan dari tabel HTML standar sangat berbeda warna. karena tata letak tabel HTML standar memiliki tata letak yang bias saja. Oleh karena itu, penjelasan tentang cara membuat tabel HTML menggunakan CSS akan diberikan dalam pelajaran belajar CSS.

Dalam pelajaran ini, saya akan membahas petunjuk langkah demi langkah untuk mendesain tabel. Hal pertama yang kami buat adalah kerangka tabel.

index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Design Table</title>
5 </head>
6 <body>
7   <h1>Cara Design Table Dengan CSS</h1>
8   <table border="1">
9     <tr>
10      <th>No</th>
11      <th>Nama</th>
12      <th>Alamat</th>
13      <th>Jurusan</th>
14    </tr>
15    <tr>
16      <td>1</td>
17      <td>Andi Saputra</td>
18      <td>Kendal</td>
19      <td>Teknik Komputer</td>
20    </tr>
21    <tr>
22      <td>2</td>
23      <td>Budi Budiman</td>
24      <td>Tuhurejo</td>
25      <td>Sistem Komputer</td>
26    </tr>
26    </tr>
27    <tr>
28      <td>3</td>
29      <td>Adi Pratama</td>
30      <td>Ngaliyan</td>
31      <td>Bisnis</td>
32    </tr>
33    <tr>
34      <td>4</td>
35      <td>Diki</td>
36      <td>Kaliwungu</td>
37      <td>Komputerisasi Akuntansi</td>
38    </tr>
39    <tr>
40      <td>5</td>
41      <td>Erinda</td>
42      <td>Boja</td>
43      <td>Komputerisasi Akuntansi</td>
44    </tr>
45  </table>
46 </body>
47 </html>
```

Dan jika menggunakan browser maka hasilnya akan seperti pada kalimat ini.



Gambar 7.10 design tabel tanpa CSS

Seperti yang ditunjukkan di atas. Nampak hasilnya masih standar. Hal ini dapat dibuat agar lebih menarik tentang desain tabel. Buat file CSS dan sambungkan secara permanen ke file HTML yang sesuai. Tutorial sebelumnya tentang mempelajari HTML sambil mengintegrasikan CSS dan HTML dapat ditemukan di sini. Bisa dilihat pada line ke 5 pada file index.html

Kemudian setelah itu silahkan berikan nama class table1 pada tag table pada line 11 di bawah ini. Untuk tag h3 pada line 9 itu optional saja, bisa ditambahkan dan juga bisa diabaikan. Untuk line di bawahnya tidak ada perubahan sama sekali. Hasilnya akan seperti gambar di bawah ini.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Design Table</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <h1>Cara Design Table Dengan CSS</h1>
9   <h3>Desain tabel dengan css</h3>
10
11  <table class="table1">
12    <tr>
13      <th>No</th>
14      <th>Nama</th>
15      <th>Alamat</th>
16      <th>Jurusan</th>
17    </tr>
```

Kemudian buat style.css nya di bawah ini

```
1 /*design table 1*/
2 .table1 {
3   font-family: sans-serif; color: #232323;
4   border-collapse: collapse;
5 }
6
7 .table1, th, td {
8   border: 1px solid #999; padding: 8px 20px;
9 }
10
```

Cermati juga selector nya. .table1, th, td artinya semua selector yang ditunjukkan semuanya akan diubah seperti pada line 8. Berbeda dengan cara pemanggilan tag di dalam tag atau tag di dalam class

dimana tidak menggunakan symbol tanda koma. Dan jika index.html di buka pada aplikasi web browser maka akan seperti ini



Gambar 7.11 proses design tabel dengan CSS

Harap perhatikan gambar di atas. Meja di sini sekarang jauh lebih ramai. Di tabel elemen, kami menyediakan font yang dapat digunakan oleh penulis. karena kita akan mengubah font supaya terlihat lebih cerah dan hidup. Pada tutorial sebelumnya, saya telah menjelaskan cara mengubah jenis huruf. Silakan baca panduan CSS tentang cara mengubah font. Kemudian ganti nomor kode font dengan kode #232323. Dan konsep yang paling penting adalah border-collapse:collapse. Ini memiliki fungsi membuat grafik grafik tabel lebih mudah dibaca.

Kemudian, untuk elemen table head dan table body, kami menggunakan kode warna #999 untuk peringatan tabel. Selain itu, tambahkan padding di bagian bawah dengan lebar sekitar 8 px dan tinggi 20 px.

Sekarang coba modifikasi style.css menjadi seperti di bawah ini, bisa juga di tambahkan dengan sesuka anda.

```

1  .table1 {
2     font-family: sans-serif;
3     color: #444;
4     border-collapse: collapse;
5     width: 50%;
6     border: 1px solid #f2f5f7;
7  }
8  .table1 tr th{
9     background: #35A9DB;
10    color: #fff;
11    font-weight: normal;
12  }
13  .table1, th, td {
14    padding: 8px 20px;
15    text-align: center;
16  }
17  .table1 tr:hover {
18    background-color: #f5f5f5;
19  }
20  .table1 tr:nth-child(even) {
21    background-color: #f2f2f2;
22  }
23

```

Dan hasilnya seperti di bawah ini



Gambar 7.12 hasil akhir design tabel dengan CSS

Silahkan amati selector nya, kemudian anda coba ganti value nya dan coba amati perubahannya. Pada bagian hasil ketika di tampilkan ke baris no 2 dan 4 ketika di tunjuk kursor akan berubah warna, namun pada baris 1,3 dan 5 warna baground tidak akan berubah karena bisa di lihat pada style.css selector pada line 20 – 22

nth-child digunakan untuk memilih elemen sesuai dengan aturan perilaku anak (anak). Ada tag pembuka dan penutup untuk setiap elemen. Contoh tag pembuka adalah "<div>", dan contoh tag penutup adalah "</div>", yang dilambangkan dengan garis miring "/". Akibatnya, setiap elemen yang ada berada di dalam batas elemen lain (dalam tag pembuka dan penutup) bisa disebut anak atau bayi

dari unsur yang ada di sekitarnya. Namun tidak semua unsur memiliki tag penutup. anak ke-n ditulis dengan kode seperti::anak ke-n(X): X = bisa ditulis dengan angka, kata kunci, atau rumus.

BAB 8

ANIMASI GAMBAR & MEMBUAT TEMPLATE WEB

TUJUAN :

- a. Mahasiswa Mampu Memberikan Animasi atau Efek pada Gambar
- b. Mahasiswa Mampu Membuat Template Web
- c. Mahasiswa Mampu Membuat Tampilan Web Responsive

Overview :

Tampilan sangat mempengaruhi dari kenyamanan para pengunjung web, semakin bagus tampilan dan efek yang di berikan, memberikan kesan berbeda dalam menyajikan sebuah web. Hal tersebut dapat membuat para pengunjung merasa di sajikan hal yang berbeda dalam membuka web tersebut. Efek gambar yang di pelajari disini seperti membuat efek blur, transisi dan zoom in maupun zoom out.

Sebagai seorang web programing, wajib untuk dapat membuat suatu tampilan atau template dari sebuah website. Pada BAB ini juga akan di bahas cara membuat template web yang cukup menarik dari HTML di hias dengan CSS dan sedikit mengambil dari javascript.

8.1 MEMBERIKAN EFEK PADA GAMBAR DENGAN CSS

Beberapa arahan CSS tersedia untuk mendapatkan efek zoom yang baik pada gambar saat diarahkan, meskipun digabungkan dengan berbagai rotasi, penskalaan, pengaburan, dan elemen lainnya. Ada beberapa animasi di banyak situs Web saat ini. Misalnya, animasi sederhana yang memperbesar gambar saat melayang di dalam bingkai

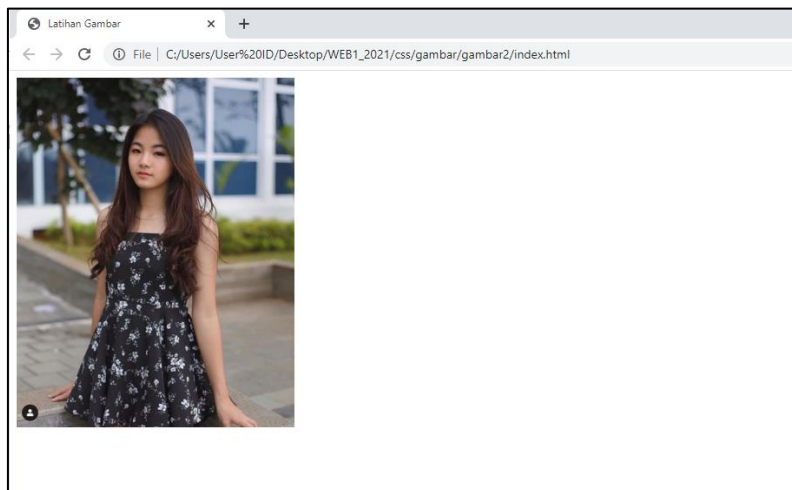
CSS sangat membantu untuk menyempurnakan tampilan konten website, seperti dengan menambahkan warna, box shadow, padding, margin, align, dan elemen lainnya. Modul ini akan menjelaskan cara membuat animasi gambar melayang tanpa menggunakan JavaScript. Tanpa menggunakan JavaScript dan hanya sedikit menggunakan CSS. Silahkan baca informasi di bawah ini secara perlahan.

File HTML

index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Latihan Gambar</title>
5   <link rel="stylesheet" type="text/css" href="basic.css">
6 </head>
7 <body>
8   <div class="img-hover-zoom">
9     
10  </div>
11 </body>
12 </html>
```

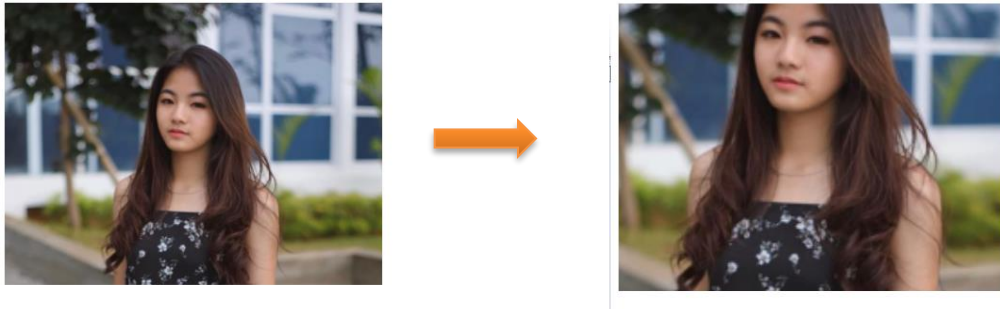
Dokumen HTML diatas hanya menampilkan gambar dengan tag img. Pada line 5 disana di gunakan untuk memanggil dokumen css yang akan kita gunakan untuk memberikan efek transisi zoom, transisi dan sejenisnya. Dan jika dokumen CSS belum diisikan maka tampilan index.html akan seperti di bawah ini



Gambar 8.1 menginputkan gambar dengan HTML

Silahkan Anda coba satu per satu mengisikan dokumen css seperti di bawah ini

a) Basic Zoom (basic.css)

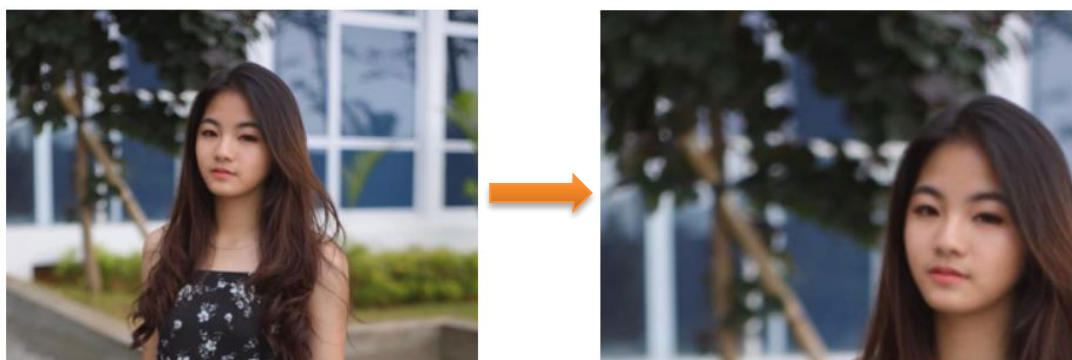


Gambar 8.2 Zoom image dengan CSS

Basic.css

```
index.html x quick.css x basic.css x
1 .img-hover-zoom {
2   width: 100%;
3   max-width: 300px;
4   height: 300px;
5   overflow: hidden;
6 }
7 .img-hover-zoom img {
8   transition: transform .5s ease;
9   width: 100%;
10  max-width: 500px;
11  height: 500px;
12 }
13 .img-hover-zoom:hover img {
14   transform: scale(1.5);
15 }
```

b) Quick zoom-in (quick.css)

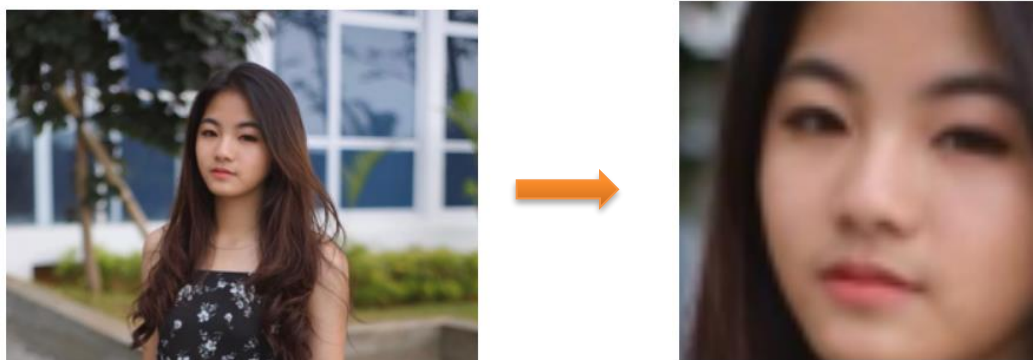


Gambar 8.3 Quick zoom-in image dengan CSS

quick.css

```
index.html x quick.css x basic.css x
1 .img-hover-zoom {
2   width: 100%;
3   max-width: 300px;
4   height: 300px;
5   overflow: hidden;
6 }
7 .img-hover-zoom img {
8   transform-origin: 0 0;
9   transition: transform .25s, visibility .25s ease-in;
10 }
11 .img-hover-zoom:hover img {
12   transform: scale(2);
13 }
```

c) Point-zoom (point.css)

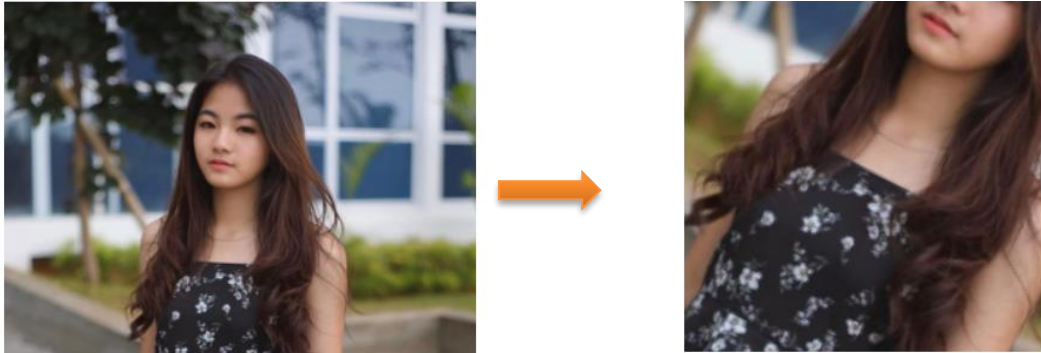


Gambar 8.4 point-zoom image dengan CSS

Point.css

```
index.html x point.css x
1 .img-hover-zoom {
2   width: 100%;
3   max-width: 300px;
4   height: 300px;
5   overflow: hidden;
6 }
7 .img-hover-zoom img {
8   transform-origin: 45% 20%;
9   transition: transform 1s, filter .5s ease-out;
10 }
11 .img-hover-zoom:hover img {
12   transform: scale(5);
13 }
14
```

d) Zoom dan Rotasi (rotasi.css)

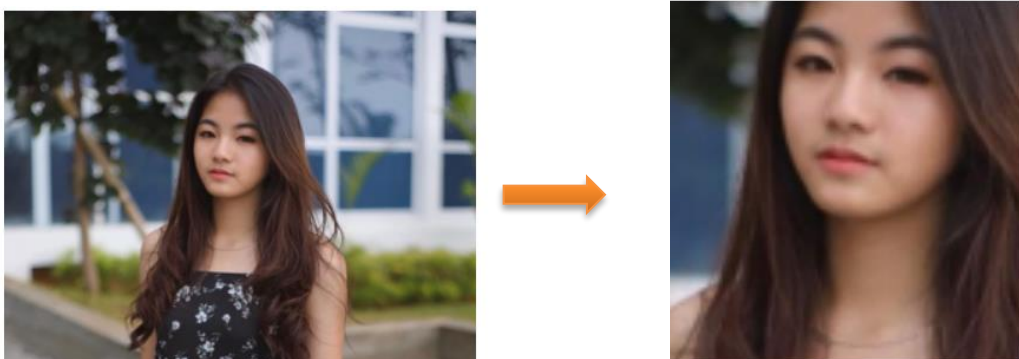


Gambar 8.5 Zoom dan Rotasi image dengan CSS

rotasi.css

```
index.html x rotasi.css x
1 .img-hover-zoom {
2   width: 100%;
3   max-width: 300px;
4   height: 300px;
5   overflow: hidden;
6 }
7 .img-hover-zoom img {
8   transition: transform .5s ease-in-out;
9 }
10 .img-hover-zoom:hover img {
11   transform: scale(2) rotate(25deg);
12 }
13
```

e) Slow Motion (slow.css)



Gambar 8.6 slow motion image dengan CSS

Slow.css

```
index.html x slow.css x
1 .img-hover-zoom {
2   width: 100%;
3   max-width: 300px;
4   height: 300px;
5   overflow: hidden;
6 }
7 .img-hover-zoom img {
8   transform-origin: 45% 25%;
9   transition: transform 5s, filter 3s ease-in-out;
10  filter: brightness(150%);
11 }
12 .img-hover-zoom:hover img {
13   filter: brightness(100%);
14   transform: scale(3);
15 }
16
```

f) Brighten and Zoom-in (brighten.css)



Gambar 8.7 brighten and zoom-in image dengan CSS

Brighten.css

```
index.html x brighten.css x
1 .img-hover-zoom {
2   width: 100%;
3   max-width: 300px;
4   height: 300px;
5   overflow: hidden;
6 }
7 .img-hover-zoom img {
8   transition: transform 2s, filter 1.5s ease-in-out;
9   transform-origin: center center;
10  filter: brightness(50%);
11 }
12 .img-hover-zoom:hover img {
13   filter: brightness(100%);
14   transform: scale(1.3);
15 }
16
```

g) Blur-out with Zooming-in (blur.css)



Gambar 8.8 Blur-out with Zooming-in image dengan CSS

Blur.css

```
index.html x blur.css x
1  .img-hover-zoom {
2    width: 100%;
3    max-width: 300px;
4    height: 300px;
5    overflow: hidden;
6  }
7  .img-hover-zoom img {
8    transition: transform 1s, filter 2s ease-in-out;
9    filter: blur(2px);
10   transform: scale(1.2);
11  }
12 .img-hover-zoom:hover img {
13   filter: blur(0);
14   transform: scale(1);
15 }
```

h) Colorize with zooming-in (color.css)



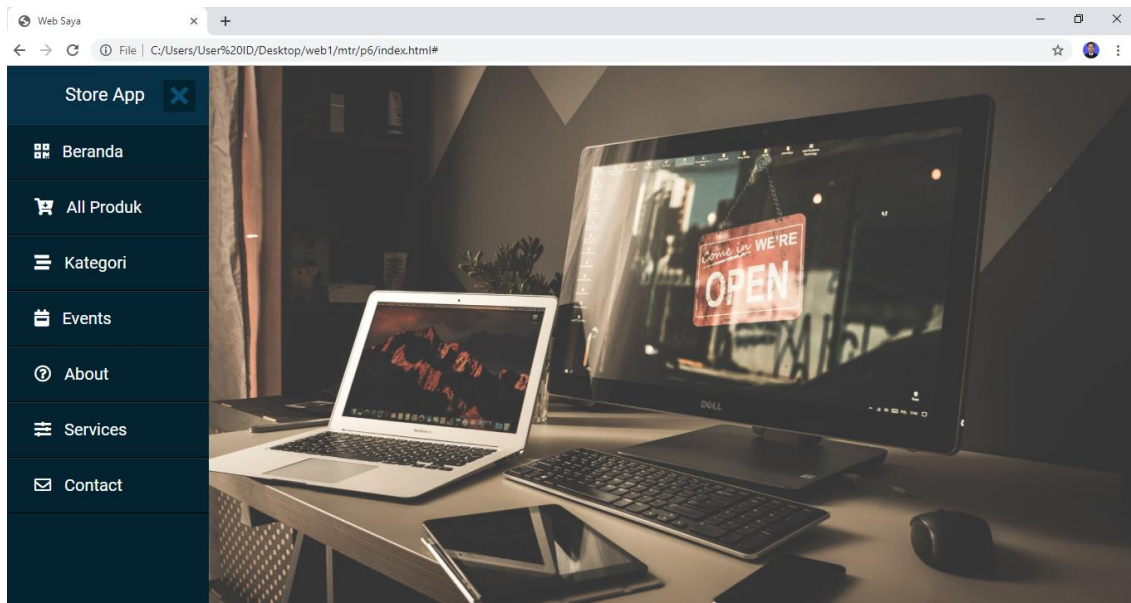
Gambar 8.9 Colorize with zooming-in image dengan CSS

Color.css

```
1 .img-hover-zoom {
2   width: 100%;
3   max-width: 300px;
4   height: 300px;
5   overflow: hidden;
6 }
7 .img-hover-zoom img {
8   transition: transform .5s, filter 1.5s ease-in-out;
9   filter: grayscale(100%);
10 }
11 .img-hover-zoom:hover img {
12   filter: grayscale(0);
13   transform: scale(1.1);
14 }
15
```

8.2 MEMBUAT TEMPLATE WEB

Pada tutorial sub poin ini akan di bahas cara pembuatan template dengan menggunakan HTML, CCS dan di tambah java script sedikit untuk icon – icon nya. Hasil akhir akhir yang nantinya terselesaikan adalah seperti ini



Gambar 8.10 Contoh template web

Silahkan ikuti tahapan di bawah ini

Index.html

```
index.html x style.css x
1 <!DOCTYPE html>
2 <html lang="en" dir="ltr">
3   <head>
4     <meta charset="utf-8">
5     <title>Web Saya</title>
6     <link rel="stylesheet" type="text/css" href="style.css">
7     <!-- fontawesome online -->
8     <script src="https://kit.fontawesome.com/a076d05399.js"></script>
9     <!-- Offline, simpan 1 folder dengan file html -->
10    <!-- <script src="fontawesome.js"></script> -->
11  </head>
12  <body>
13    <input type="checkbox" id="check">
14    <label for="check">
15      <i class="fas fa-bars" id="btn"></i>
16      <i class="fas fa-times" id="cancel"></i>
17    </label>
18    <div class="sidebar">
19      <header>Store App</header>
20      <ul>
21        <li><a href="#"><i class="fas fa-qrcode"></i>Beranda</a></li>
22        <li><a href="#"><i class="fas fa-cart-arrow-down"></i>All Produk</a></li>
23        <li><a href="#"><i class="fas fa-stream"></i>Kategori</a></li>
24        <li><a href="#"><i class="fas fa-calendar-week"></i>Events</a></li>
25        <li><a href="#"><i class="far fa-question-circle"></i>About</a></li>
26        <li><a href="#"><i class="fas fa-sliders-h"></i>Services</a></li>
27        <li><a href="#"><i class="far fa-envelope"></i>Contact</a></li>
28      </ul>
29    </div>
30    <section>
31  </section>
32 </body>
33 </html>
```

Satu-satunya atribut global HTML yang dapat diterapkan ke elemen HTML apa pun adalah atribut lang, dan atribut ini sangat membantu karena aksesibilitasnya (aksesibilitas). Elemen HTML yang memungkinkan deklarasi diterapkan ke semua konten halaman tertentu adalah yang paling sering digunakan. Misalnya, deklarasi `html lang="en">` memberi tahu browser bahwa semua konten di bagian yang diterjemahkan adalah dalam bahasa Inggris.

Nilai Atribut ltr adalah kata kunci kiri-ke-kanan yang akan membuat teks membentang dari atas ke bawah web browser. RTL, di sisi lain, adalah singkatan dari kanan ke kiri yang akan mencetak teks dari atas ke bawah.

Data tentang data adalah meta tag, dan itu tidak ditujukan untuk pengguna melainkan untuk browser web atau "program robot" seperti mesin pencari (search engine). Menurut definisi resmi, charset adalah kumpulan instruksi yang digunakan untuk mengubah bit-bit dalam HTML menjadi karakter. Oleh karena itu, charset UTF-8 dalam dokumen ini memberikan instruksi kepada browser web tentang cara merender karakter dalam dokumen HTML sebagai UTF-8. Saat ini, satu-satunya set karakter yang digunakan adalah ASCII (charset="us-ascii"), meskipun set karakter ini dibatasi paling banyak 128 karakter. Kemudian ubah ASCII menjadi karakter ANSI atau cetak dengan charset ISO-8859-1. Karakter ANSI dapat mencakup hingga 256 karakter dan sering digunakan dalam HTML versi 2.0 hingga 4.01. Permasalahan yang diberikan pada karakter ASCII atau ANSI tidak menyertakan karakter non-latin, seperti huruf Arab, China, Jepang, dll, sehingga untuk keperluan ini dikembangkanlah set

untuk masing-masing bahasa, seperti big5 untuk Akibatnya, hal ini menimbulkan perbedaan pada set karakter tergantung bahasa yang akan digunakan untuk menulis kode HTML. Ini adalah sesuatu yang dapat diprogram menggunakan sintaks HTML, tetapi akan lebih praktis jika semua karakter disertakan dalam satu set. Sejak awal pengembangan XHTML, istilah "Unicode Transformation Format-8" (UTF-8) telah digunakan untuk merujuk pada kumpulan karakter yang mewakili sekitar 10.000 karakter lebih banyak daripada yang sekarang digunakan di dunia dan telah menjadi karakter standar. ditetapkan untuk seluruh dunia.

"Font Awesome" adalah kumpulan ikon yang dibuat dengan harga yang sama dan memanfaatkan sistem "Font Icon", yang memungkinkan pengguna mengubah warna dan ukuran font tanpa harus mempelajari CSS, dan membuat ikon lebih kecil dan lebih bulat daripada Ikon Gambar.

Style.css

```
index.html x style.css x
1 @import url('https://fonts.googleapis.com/css?family=Roboto:300,400,400i,500');
2 *{
3   padding: 0;
4   margin: 0;
5   list-style: none;
6   text-decoration: none;
7 }
8 body {
9   font-family: 'Roboto', sans-serif;
10 }
11 .sidebar {
12   position: fixed;
13   left: -250px;
14   width: 250px;
15   height: 100%;
16   background: #042331;
17   transition: all .5s ease;
18 }
19 .sidebar header {
20   font-size: 22px;
21   color: white;
22   line-height: 70px;
23   text-align: center;
24   background: #063146;
25   user-select: none;
26 }
27 .sidebar ul a{
28   display: block;
29   height: 100%;
30   width: 100%;
31   line-height: 65px;
32   font-size: 20px;
33   color: white;
34   padding-left: 40px;
35   box-sizing: border-box;
36   border-bottom: 1px solid black;
37   border-top: 1px solid rgba(255,255,255,.1);
38   transition: .4s;
39 }
40 ul li: hover a{
41   padding-left: 50px;
42 }
43 .sidebar ul a i{
```



```

44     margin-right: 16px;
45 }
46 #check{
47     display: none;
48 }
49 label #btn,label #cancel{
50     position: absolute;
51     background: #042331;
52     border-radius: 3px;
53     cursor: pointer;
54 }
55 label #btn{
56     position: fixed;
57     left: 40px;
58     top: 25px;
59     font-size: 35px;
60     color: white;
61     padding: 6px 12px;
62     transition: all .5s;
63 }
64 label #cancel{
65     position: fixed;
66     z-index: 1111;
67     left: -195px;
68     top: 17px;
69     font-size: 30px;
70     color: #0a5275;
71     padding: 4px 9px;
72     transition: all .5s ease;
73 }
74 #check:checked ~ .sidebar{
75     left: 0;
76 }
77 #check:checked ~ label #btn{
78     left: 250px;
79     opacity: 0;
80     pointer-events: none;
81 }
82 #check:checked ~ label #cancel{
83     left: 195px;
84 }
85 #check:checked ~ section{
86     margin-left: 250px;
87 }
88 section{
89     background: url(bg.jpeg) no-repeat;
90     background-position: center;
91     background-size: cover;
92     height: 100vh;
93     transition: all .5s;
94 }
95

```

Istilah "pemilih universal" mengacu pada pemilih yang digunakan untuk memilih setiap elemen di dalam lingkup saat ini.

```
* {  
  border: 1px solid grey;  
}
```

Oleh karena itu, setiap elemen akan memiliki batas padat, lebar 1px, berwarna abu-abu di sekelilingnya.

Pemilih universal dapat digunakan untuk mengatur ulang CSS. Mengapa kita perlu mengatur ulang? Ada beberapa elemen ramah browser CSS dalam kode sumber HTML, seperti padding dan margin pada elemen saat ini. Tujuan reset adalah untuk menghapus padding dan margin saat ini. Reset Contoh CSS:

```
* {  
  padding: 0;  
  margin: 0;  
}
```

Karenanya, tidak ada elemen yang memiliki padding atau margin.

BAB 9

MENGISI KOTEN & MEMBUAT HALAMAN RESPONSIVE

TUJUAN :

- a. **Mahasiswa Dapat Mengisi Konten dari Template yang Sudah di Buat sebelumnya**
- b. **Mahasiswa Mampu Membuat Tautan Link berdasarkan Name Section**
- c. **Mahasiswa Mampu Membuat Halaman Web Menjadi Responsive**

Overview :

Setelah anda membuat template web pada BAB sebelumnya, pada BAB kali ini berfokus dalam pengisian konten web beserta membuat link berdasarkan name ID. Untuk hyperlink name ID sudah pernah diajarkan juga pada buku ini, silahkan anda pelajari lagi pada BAB ke 3 sub poin hyperlink.

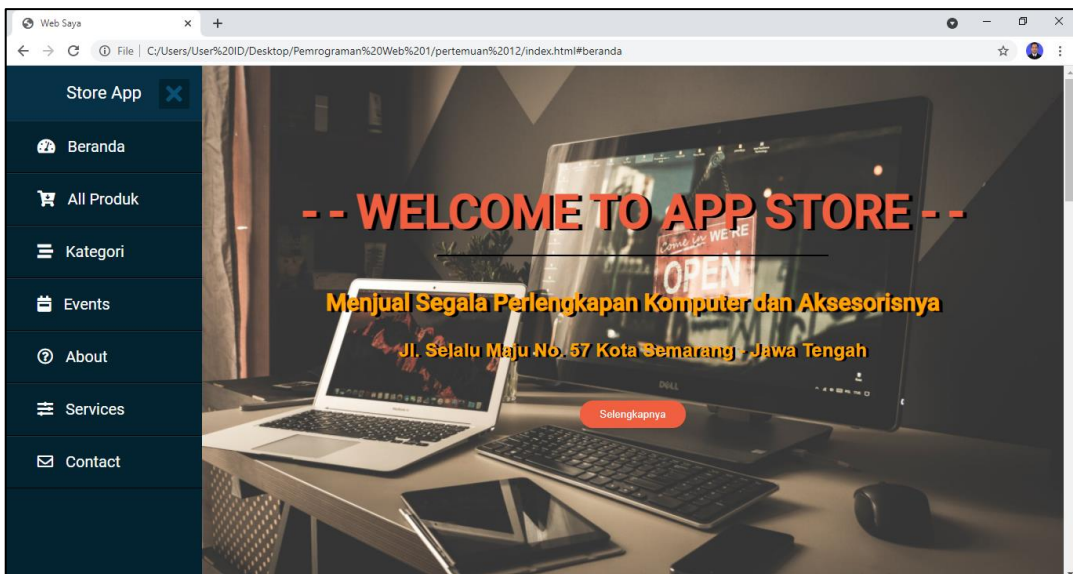
Saat ini, sebagian besar pengguna internet mengakses situs web melalui perangkat seluler. Akibatnya, setiap aplikasi situs web harus menjunjung tinggi prinsip akuntabilitas (tata letak responsif). Istilah "tata letak responsif" mengacu pada teknik untuk membuat halaman web yang menggunakan teks fleksibel, gambar fleksibel, dan file media fleksibel dalam lembar gaya. Aplikasi web yang menggunakan tata letak responsif mendapat manfaat dari fakta bahwa informasi yang ditampilkan di situs web dapat dibaca dan dipahami oleh pengunjung tanpa kesulitan dan dapat diakses tanpa penundaan dari perangkat seluler apa pun.

9.1 MENGISI KONTEN

Setelah template HTML dan CSS tahapan sebelumnya telah selesai, silahkan Anda isikan konten pada masing – masing menubar pada bagian section. Kemudian silahkan Anda atur sendiri mengenai tampilan misalkan font-size, padding, margin, dan lain – lain sedemikian rupa sehingga tampilan halaman sangat bagus dan nyaman untuk di lihat.

Dalam pembuatan web tidak hanya isi konten saja yang di perhatikan, pewarnaan, tata letak, gaya atau style atau bentuk huruf juga juga harus di perhatikan. Karena style yang bagus akan membuat user atau pengujung menjadi nyaman untuk membuka web tersebut lebih lama. Bahkan style yang bagus juga membuat user tertarik untuk melihat isi kontennya.

Dalam pengisian konten silahkan Anda masukkan kedalam section dan berikan ID pada setiap section tersebut untuk di gunakan proses pemanggilan pada side-bar nya. Berikut ini contoh dalam membuat section Beranda yang hasilnya seperti di bawah ini.



Gambar 9.1 Halaman Beranda

Selanjutnya atur file index.html seperti di bawah ini

```
19 <header>Store App</header>
20 <ul>
21 <li><a href="#beranda"><i class="fas fa-tachometer-alt"></i>Beranda</a></li>
22 <li><a href="#produk"><i class="fas fa-cart-arrow-down"></i>All Produk</a></li>
23 <li><a href="#kategori"><i class="fas fa-stream"></i>Kategori</a></li>
24 <li><a href="#event"><i class="fas fa-calendar-week"></i>Events</a></li>
25 <li><a href="#about"><i class="far fa-question-circle"></i>About</a></li>
26 <li><a href="#servis"><i class="fas fa-sliders-h"></i>Services</a></li>
27 <li><a href="#contact"><i class="far fa-envelope"></i>Contact</a></li>
28 </ul>
29 </div>
30
31 <section id="beranda">
32 <h1>- - WELCOME TO APP STORE - -</h1>
33 <br>
34 <hr>
35 <h3>Menjual Segala Perlengkapan Komputer dan Aksesorisnya</h3>
36 <br>
37 <h2>Jl. Selalu Maju No. 57 Kota Semarang - Jawa Tengah</h2>
38 <button><a href="#about">Selengkapnya</a></button>
39 </section>
40
```

Section untuk Tampilan Beranda

```
41 <section id="about">
42 <h1>ABOUT - APP STORE</h1>
43 <br>
44 <h2>App Store adalah toko yang menjual segala perlengkapan komputer maupun laptop
```

Dalam panggilan `` menggunakan ID dari section. Anda juga bisa menghubungkan file HTML lain dengan tag dan atribut `<a href >`. Atur gaya tampilan section id beranda tersebut se suka hati Anda menggunakan css, contoh :

```
86 margin-left: 250px;
87 }
88 section{
89 margin-left: 250px;
90 text-align: center;
91 background: url(bg.jpeg) no-repeat;
92 background-position: center;
93 background-size: cover;
94 height: 100vh;
95 transition: all .5s;
96 }
97
98 h1{
99 padding-top: 150px;
100 color: #F05F40;
101 text-shadow: 5px 2px black;
102 font-size: 60px;
103 hr {
104 border-color: black;
105 margin-left: 300px;
106 margin-right: 300px;
107 border-width: 1px;
108 }
109 h3 {
110 padding-top: 40px;
111 color: orange;
112 text-shadow: 5px 2px black;
113 font-size: 30px;
114 }
115 h2{
116 color: orange;
117 text-shadow: 4px 1px black;
118 font-family: sans-serif;
119 text-align: center;
120 margin: 10px
121 }
122 button {
```

Anda bisa menggunakan selector Tag / ID maupun Class

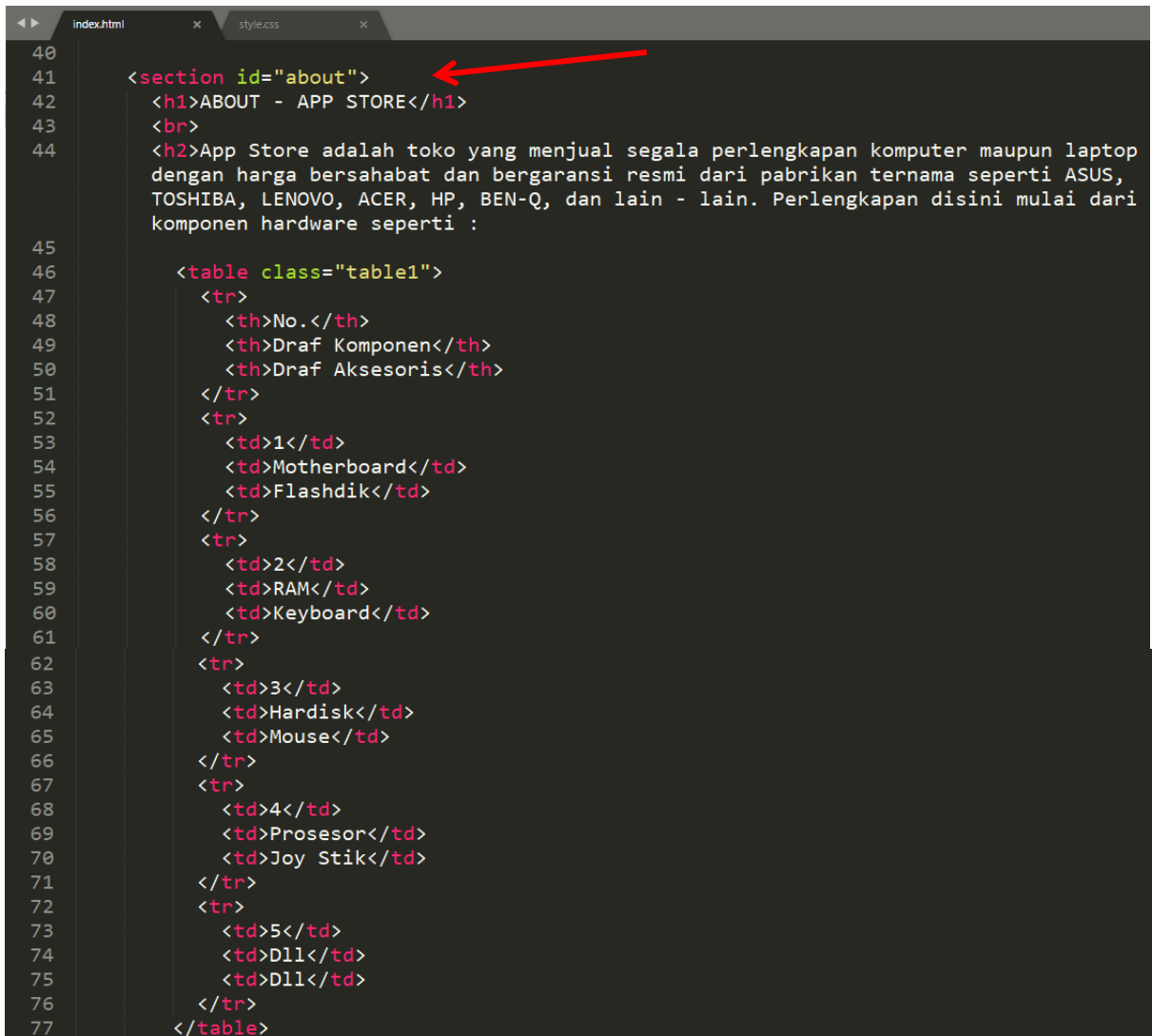
```

123 padding: 10px;
124 border:none;
125 background-color: #F05F40;
126 border-radius: 25px;
127 padding-right: 25px;
128 padding-left: 25px;
129 margin-top: 40px;
130 }
131 button a{
132 color: white;
133 }

```

Setelah membuat tahapan di atas, silahkan anda buat juga section untuk halaman lain. Saya contohkan disini membuat section untuk about yang nanti bisa di panggil lewat sidebar About dan juga Tampilan selengkapnya pada halaman Beranda. Silahkan buat section lagi (untuk halaman about) di bawah section halaman beranda.

index.html



```

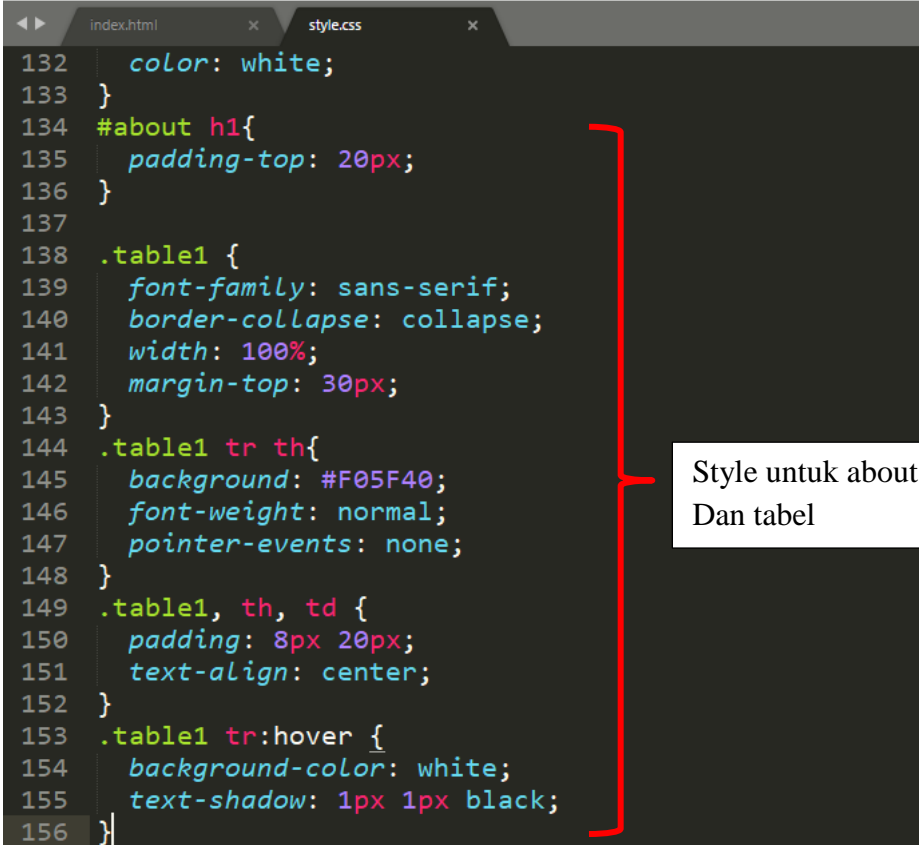
40
41 <section id="about">
42 <h1>ABOUT - APP STORE</h1>
43 <br>
44 <h2>App Store adalah toko yang menjual segala perlengkapan komputer maupun laptop
    dengan harga bersahabat dan bergaransi resmi dari pabrikan ternama seperti ASUS,
    TOSHIBA, LENOVO, ACER, HP, BEN-Q, dan lain - lain. Perlengkapan disini mulai dari
    komponen hardware seperti :
45
46 <table class="table1">
47 <tr>
48 <th>No.</th>
49 <th>Draf Komponen</th>
50 <th>Draf Aksesoris</th>
51 </tr>
52 <tr>
53 <td>1</td>
54 <td>Motherboard</td>
55 <td>Flashdik</td>
56 </tr>
57 <tr>
58 <td>2</td>
59 <td>RAM</td>
60 <td>Keyboard</td>
61 </tr>
62 <tr>
63 <td>3</td>
64 <td>Hardisk</td>
65 <td>Mouse</td>
66 </tr>
67 <tr>
68 <td>4</td>
69 <td>Prosesor</td>
70 <td>Joy Stik</td>
71 </tr>
72 <tr>
73 <td>5</td>
74 <td>Dll</td>
75 <td>Dll</td>
76 </tr>
77 </table>

```

```
78     <br>
79     Alamat toko ini ada di Jl. Selalu Maju No. 57 Kota Semarang - Jawa Tengah, Telp
      085757757757
80   </h2>
81
82   </section>
```

Kemudian setelah ini silahkan atur gaya css sesuai dengan kehendak anda pada bagian style.css nya, contoh gaya css seperti dibawah ini

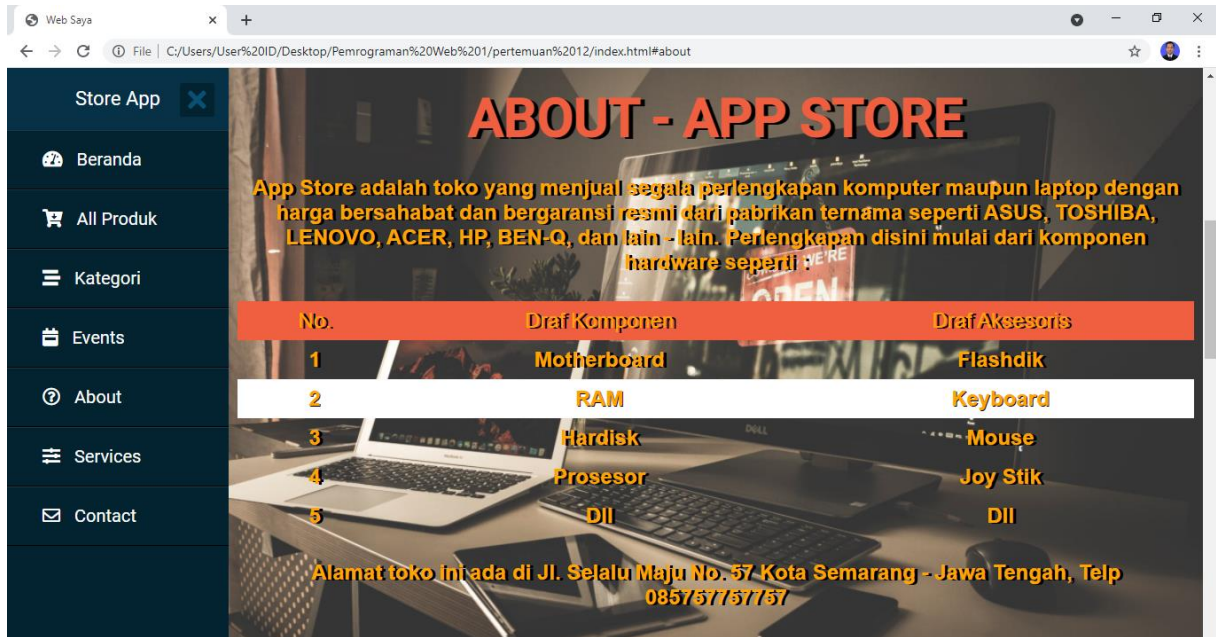
Style.css



```
132   color: white;
133 }
134 #about h1{
135   padding-top: 20px;
136 }
137
138 .table1 {
139   font-family: sans-serif;
140   border-collapse: collapse;
141   width: 100%;
142   margin-top: 30px;
143 }
144 .table1 tr th{
145   background: #F05F40;
146   font-weight: normal;
147   pointer-events: none;
148 }
149 .table1, th, td {
150   padding: 8px 20px;
151   text-align: center;
152 }
153 .table1 tr:hover {
154   background-color: white;
155   text-shadow: 1px 1px black;
156 }
```

Style untuk about
Dan tabel

Hasilnya dapat dilihat pada halaman berikutnya



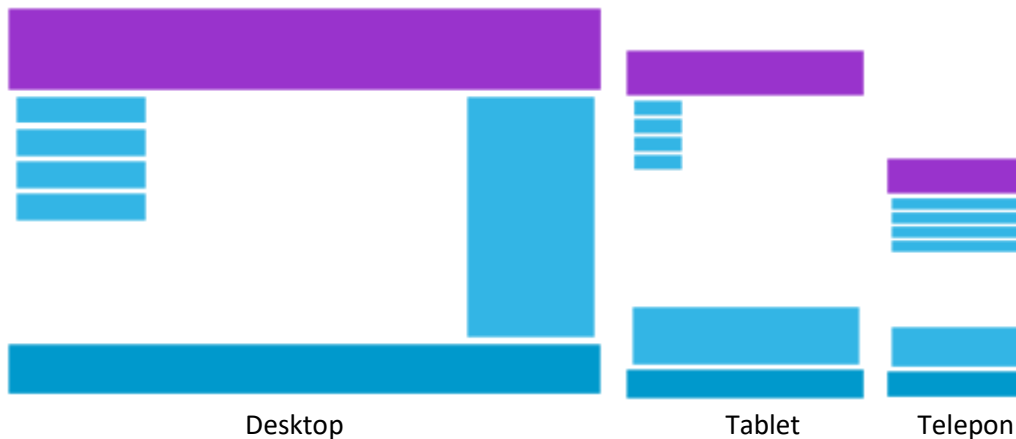
Gambar 9.2 Halaman About

Silahkan Anda lanjutkan sendiri untuk pembuatan section untuk menampilkan halaman yang lainnya beserta atur gaya CSS nya sesuai kehendak Anda.

9.2 MEMBUAT HALAMAN MENJADI RESPONSIVE

Desain responsif web membuat beranda situs web Anda tampak bagus dari segala sudut. Desain responsif web hanya menggunakan HTML dan CSS. Situs web dapat dilihat di berbagai perangkat, termasuk desktop, tablet, dan ponsel. Apa pun perangkatnya, situs web Anda akan terlihat bagus dan mudah digunakan.

Situs web tidak dapat memperbarui informasi secara otomatis agar sesuai dengan perangkat yang lebih kecil; sebagai gantinya, konten harus disesuaikan agar cocok dengan perangkat mana pun yang digunakan.



Gambar 9.3 tampilan Halaman web di berbagai perangkat

Ini disebut sebagai desain web responsif saat menggunakan CSS dan HTML untuk menyesuaikan ukuran, bentuk, warna, font, atau atribut konten lainnya sehingga tampak sesuai di mana pun ditempatkan.

1. Ubahlah sidebar menjadi terlihat saat halaman pertama kali di buka dengan cara mengubah value `left : -250px;` menjadi `left : 0px;` ini akan membuat sidebar akan selalu terlihat.

```
11 .sidebar {
12     position: fixed;
13     left: -250px;
14     width: 250px;
11 .sidebar {
12     position: fixed;
13     left: 0px;
14     width: 250px;
```

2. Atur posisi saat web di buka dengan menggunakan layar lebar seperti monitor PC atau laptop yang lebih dari 1300px.

```
163 }
164
165 /* responsive media query start */
166 @media (max-width: 1300px) {
167     .sidebar{
168         left: 0px;
169     }
170 }
171
```

@media di gunakan untuk merubah tampilan browser agar tampilan website bisa menjadi responsive. @media dapat berisi min-width dan max-width.

3. Atur tampilan saat web di buka dengan menggunakan handphoe atau layar di bawah 500px;

```
172
173 @media (max-width: 500px) {
174     section{
175         margin-left: 0px;
176     }
177     .sidebar {
178         position: fixed;
179         left: -250px;
180         width: 150px;
181         height: 100%;
182         background: #042331;
183         transition: all .5s ease;
184     }
185     .sidebar header {
186         font-size: 17px;
187         color: white;
188         line-height: 50px;
189         text-align: center;
190         background: #063146;
191         user-select: none;
```

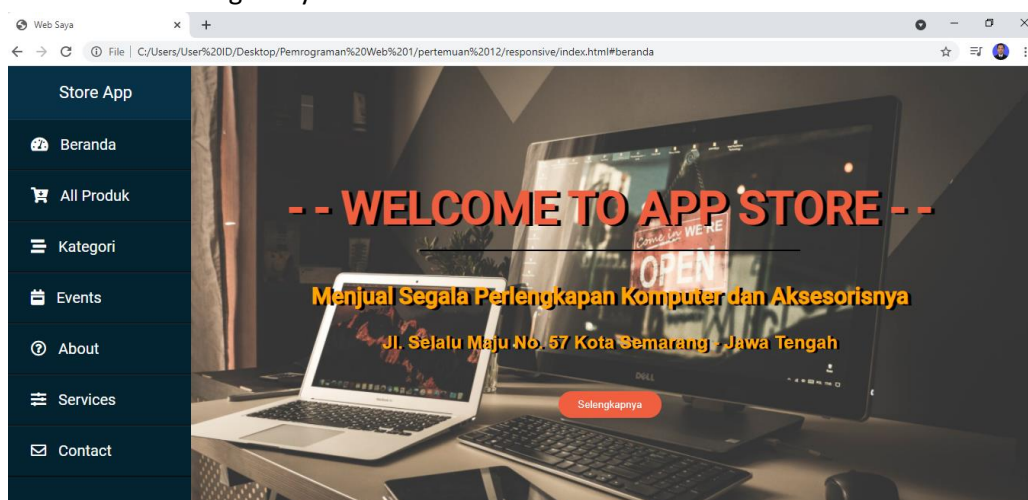
```

192 }
193 .sidebar ul a{
194   display: block;
195   height: 100%;
196   width: 100%;
197   line-height: 50px;
198   font-size: 15px;
199   color: white;
200   padding-left: 20px;
201   box-sizing: border-box;
202   border-bottom: 1px solid black;
203   border-top: 1px solid rgba(255,255,255,.1);
204   transition: .4s;
205 }
206 ul li:hover a{
207   padding-left: 30px;
208 }
209 .sidebar ul a i{
210   margin-right: 6px;
211 }
212 #check:checked ~ section{
213   margin-left: 150px;
214 }
215 #check:checked ~ label #cancel{
216   left: 155px;
217   color: white;
218   font-size: 10;
219 }
220

```

Untuk syntac diatas ini sebenarnya sama dengan syntac yang ada pada atasnya saat kita belajar mengisi konten. Bedanya hanya terletak pada value nya saja karena kita akan mengatur tampilan menjadi lebih kecil.

Ketika di buka dengan layar lebar :



Gambar 9.4 responsive dengan layar komputer

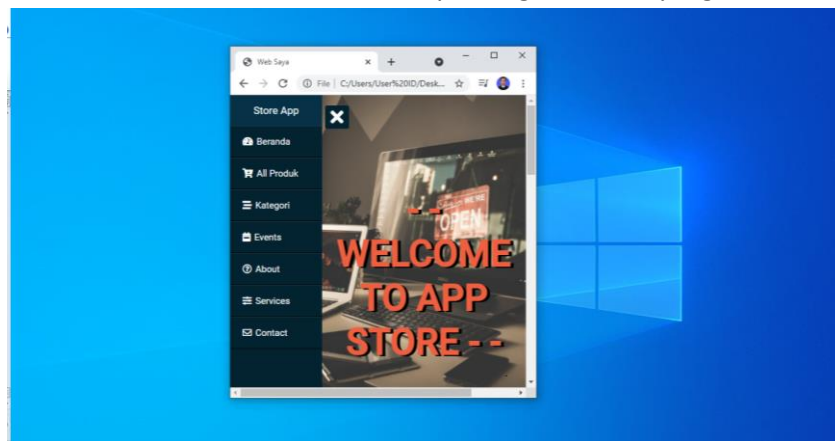
Ketika di buka dengan layar kecil :

- Sidebar akan terhidden dan di tampilkan tombol menu secara otomatis



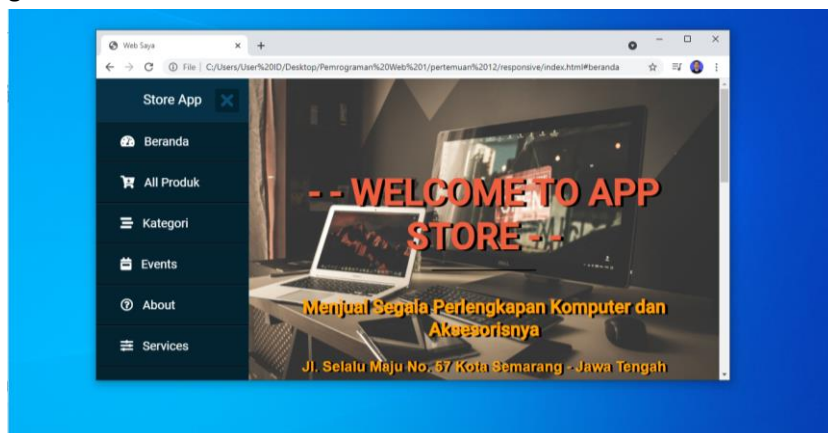
Gambar 9.5 responsive dengan layar smartphone sidebar tersembunyi

- Saat menu di klik , sidebar akan tampil dengan ukuran yang lebih kecil



Gambar 9.6 responsive dengan layar smartphone sidebar terbuka

- Saat layar di buat sedikit besar. Bandingkan ukuran menubar gambar di atas dengan gambar di bawah ini



Gambar 9.7 responsive dengan layar tablet

BAB 10

JAVA SCRIPT DAN ATURAN DASAR

TUJUAN :

- a. **Mahasiswa Memahami Pengertian JavaScript**
- b. **Mahasiswa Memahami Sejarah JavaScript**
- c. **Mahasiswa Memahami Istilah yang Ada Pada JavaScript**
- d. **Mahasiswa Mampu Membuat Dokument JavaScript**
- e. **Mahasiswa Mampu Membuat Tampilan Output Pada JavaScript**

Overview :

JavaScript adalah bahasa pemrograman yang paling banyak digunakan oleh programmer situs web atau pengembang situs web. Ada kemungkinan bahwa meskipun Anda sudah familiar dengan gaya ini, Anda tidak menyadari maknanya. Berbeda dengan orang yang cuek, Anda yang sudah memprogramkan dunia pemrograman dan kode kemungkinan besar akan memahaminya dengan mudah.

Jika Anda adalah satu-satunya orang di dunia yang tertarik mempelajari JavaScript, Anda dapat membaca lebih lanjut tentangnya di bagian bawah halaman ini. Kali ini, kita akan membahas setiap aspek JavaScript menggunakan bahasa yang bahkan orang awam pun dapat dengan mudah memahaminya, meminimalkan penggunaan istilah penuh jargon. Sekarang saya telah mengatakan itu, silakan baca penjelasan yang lebih panjang di bawah ini.

10.1 INTRODUCTION

a) PENGERTIAN DARI JAVASCRIPT

Apa sebenarnya javascript seperti yang telah disebutkan di paragraf sebelumnya? Javascript adalah salah satu dari sedikit bahasa pemrograman internet yang dapat membuat situs web Anda lebih menarik dan layak huni. Kesalahan eksekusi Javascript terjadi di jendela browser pengguna atau tempat lain selain server, yang berbeda dari PHP.

Tampilan Javascript berbeda dari HTML dan CSS. CSS dan HTML masing-masing digunakan untuk mengubah format konten tabular. Oleh karena itu desain keseluruhannya sangat berbeda. Ada beberapa contoh orang yang menggunakan HTML, CSS, dan javascript secara bersamaan, termasuk konstruksi rumah, dekorasi interior, dan lampu serta pintu-pintu.

Akibatnya, HTML digambarkan sebagai sebuah blok bangunan yang dapat digunakan untuk merenungkan segala sesuatu, bahkan tempat-tempat dengan alamat tetap seperti tembok, pintu, atap, dan lain-lain. Sebaliknya, CSS adalah alat dekorasi rumah yang dapat mengubah tampilan wallpaper, karpet, dan hal-hal lain di rumah Anda. Javascript berguna untuk hal-hal yang perlu responsif dan menghidupkan, seperti menyalakan lampu atau membuat pintu.

Javascript membuat situs web lebih responsif karena, meski tidak ada masalah, isinya masih akan berubah secara signifikan. Anda dapat menemukan nomor rumah Anda. Tidak adanya lampu atau pintu mungkin tidak menjadi masalah, tetapi rumah besar mungkin menjadi redup dan mulai mengeluarkan sedikit kelembapan. Juga, kehadiran javascript membuat situs web agak tidak menarik.

Bahasa ini termasuk yang paling mudah dipelajari para segelintir. Beberapa website populer yang sekarang tersedia menggunakan bahasa Netscape. Menggunakan Javascript memungkinkan Anda mengirim elemen ke halaman web yang diinginkan.

Selain kompatibel dengan berbagai platform dan browser web, JavaScript cukup serbaguna. Penggunaan memorinya sangat kecil, ringan, dan cepat dibandingkan dengan bahasa pengkodean lainnya. Namun, javascript tidak seaman bahasa pemrograman lainnya. Anda dapat melanjutkan membaca paragraf berikutnya untuk kejelasan dan detail lebih lanjut.

b) SEJARAH

Seperti yang telah disebutkan sebelumnya, Javascript adalah bahasa pemrograman yang dikembangkan oleh Netscape. Proses desain membutuhkan waktu sepanjang hari. Nama Javascript adalah Mocha, yang dibuat oleh Mona, Live Script, dan kemudian Javascript.

Hanya komunitas Netscape yang menggunakan Javascript versi asli, September 1995, dan fungsinya telah diperbarui. Namun seiring berjalannya waktu, Javascript terus berkembang hingga bisa berfungsi seperti sekarang.

Pada tahun 1996, JavaScript sebagian besar disebut sebagai ECMAScript. Situasi ini dimulai pada tahun 1998. ECMAScript 2 dan ECMAScript 3 dibuat masing-masing pada tahun 1998 dan 1999. ECMAScript terus berkembang hingga akhirnya berevolusi menjadi JavaScript yang digunakan saat ini. Hampir 92 persen situs web pada tahun 2016 menggunakan Javascript. Mungkin jumlahnya akan lebih tinggi dari saat ini. Bolehkah saya mau?

Rupanya, selama dua tahun ke depan, javascript berkembang sebagai satu-satunya bahasa pemrograman yang paling populer dan penting. Jika Anda sering menggunakan internet, Anda mungkin tahu apa itu javascript, familiar dengan namanya, atau lebih kabur dan belum pernah melihat sintaksnya.

c) CARA KERJA DAN FUNGSI JAVASCRIPT

Untuk lebih memahami siapa kami sebagai sebuah tim, kami akan meninjau fungsi dan metode kerja Javascript. HTML, CSS, PHP, dan bahasa pengkodean lainnya akan sering diperiksa oleh server sebelum dikirim ke pengguna. Lebih tepatnya, ketika Anda membuat situs web menggunakan HTML, CSS, PHP, dan bahasa lainnya, bahasa situs web tersebut akan diterjemahkan secara otomatis dan diperiksa oleh server web setiap kali pengunjung meminta akses ke sana.

Server web akan segera mulai menghapus cookie yang ada, setelah itu mereka akan ditinjau dengan cermat oleh pengunjung situs web sebelum dikirim ke mereka, memungkinkan mereka untuk mendapatkan data dan informasi lainnya dari situs yang baru dikunjungi. Metode kerja ini berbeda dengan metode kerja yang digunakan oleh Javascript.

Cara kerja dari javascript adalah bahwa pengguna situs web akan dikirim ke file dengan ekstensi `evil.js` atau ke halaman yang telah disematkan secara permanen. Oleh karena itu, dimungkinkan untuk menyebutnya sebagai bahasa klien. Lain kali klien mengunjungi situs web, skrip yang disebutkan di atas akan aktif di titik akhir klien. Aksara yang sama juga digunakan pada dimakat yang sedang digunakan. Oleh karena itu, masalahnya bukan terletak pada server web melainkan pada masing-masing klien atau pengunjung situs web.

Namun, tersedia browser yang menawarkan opsi bagi pengguna untuk menentukan apakah mereka ingin mengaktifkan atau tidak. Beberapa fitur tidak akan berfungsi atau, lebih buruk lagi, tidak akan tersedia jika hal ini tidak terjadi. Dan Anda sebagai awam mungkin bisa memahami hal ini. Jika Anda sering menggunakan internet, mungkin ada situs web yang meminta Anda untuk mengaktifkan fungsi javascriptnya. Itulah satu-satunya hal yang disorot dalam paragraf ini. Namun, ada browser web yang menawarkan opsi aktif yang tidak ada.

d) KELEBIHAN JAVASCRIPT

JavaScript adalah satu-satunya bahasa pemrograman dengan banyak kekurangan. Jika dibandingkan dengan bahasa pemrograman bahasa lain, keterampilan lain lebih mudah dipelajari. Penanganan dan koreksi kesalahan juga lebih mudah. Selain itu, Anda tidak memerlukan kompiler karena browser web dapat mengartikannya menggunakan HTML.

Javascript juga dapat diaktifkan pada halaman web atau peristiwa yang ditargetkan, seperti selama klik atau gerakan mouse. File Javascript juga dapat digunakan di berbagai platform dan browser web, menjadikannya cukup populer dan penggunaannya hampir universal. Selain itu, Javascript dapat digunakan untuk memvalidasi informasi saat ini, membuatnya kurang diinginkan untuk memasukkan data secara manual.

Dibandingkan dengan bahasa pemrograman lain, Javascript lebih ringkas, lebih cepat, dan lebih efisien saat membandingkan penggunaan memori. Alasan yang disebutkan di atas hanya menjadikan Javascript sebagai bahasa pemrograman yang kurang disukai. Setiap tindakan memiliki beberapa risiko, bahkan menggunakan javascript.

e) KEKURANGAN JAVASCRIPT

Jenis ketidakpastian lainnya adalah adanya risiko ledakan yang tinggi. Seperti yang sudah kita ketahui, Javascript sangat populer. Kemungkinan hal ini akan mendorong peretas, penipu, atau aktor jahat lainnya untuk menyerang dan mencari potensi kelemahan dalam sistem.

Situasi saat ini baru-baru ini menarik perhatian organisasi Javascript, tetapi juga membawa risiko yang signifikan, sehingga yang menggunakannya harus waspada. Selain itu, Javascript dapat digunakan untuk mengaktifkan kode komputer khusus pengguna. Setiap kali mereka menggunakannya, pengguna harus berhati-hati.

Masalah lain yang tidak terkait dengan popularitas masalah ini adalah keamanan browser. Meskipun sebagian besar browser web mendukung javascript, masih ada beberapa yang tidak dan tidak dapat digunakan dalam konteks saat ini. Ini biasanya melibatkan seseorang mematikan browser web mereka. Jika dinyatakan, apakah ada tanda-tanda masalah? Karena baik pemilik situs web maupun pengunjung tidak dapat memaksimalkan konten mereka, inilah satu-satunya alasan.

f) PERBEDAAN JAVASCRIPT DENGAN BAHASA PEMROGRAMAN LAINNYA

Bahasa yang akan dibahas bersama adalah PHP, HTML, CSS, dan Javascript. Elemen pertama adalah Javascript, atau JS mungkin disertakan. Fungsi dasar JavaScript adalah membuat situs web apa pun lebih menarik dan menarik.

JS adalah bahasa pemrograman yang diaktifkan oleh browser pengguna, berbeda dengan bahasa pemrograman lain yang diaktifkan oleh server. Pengembang umumnya tidak perlu melanjutkan prosedur asli untuk menyediakan fungsionalitas yang diperlukan karena JS biasanya beralih ke posisi bawahan organisasi lain.

Anda mungkin sudah mengetahui bahwa HTML, juga dikenal sebagai Hyper Text Markup Language, sering digunakan di situs web dan berfungsi sebagai titik awal untuk berbagai bagian situs web. Setiap orang yang memiliki situs web mungkin merasa canggung menggunakan tag HTML "img" untuk gambar, "p" untuk paragraf, dan beberapa simbol lainnya.

Sebaliknya, CSS, juga dikenal sebagai Cascading Style Sheets, sering digunakan untuk mengubah gaya dan format konten. Dalam HTML, setiap elemen harus dimasukkan secara manual, namun dalam CSS, hal ini tidak diperlukan untuk menampilkan konten apa pun. Anda tidak diharuskan untuk memahami berungkali tampilan elemen.

PHP sering disebut sebagai "sisi server" karena sebagian besar pemrosesan datanya terjadi di server, tidak seperti JavaScript, yang diluncurkan dari komputer klien atau biasanya disebut sebagai "sisi klien". Bahasa pemrograman PHP sering disertakan dalam WordPress dan sistem manajemen lainnya. PHP sering digunakan dalam pengembangan back end karena menawarkan toples terbaik untuk memfasilitasi proses transfer informasi.

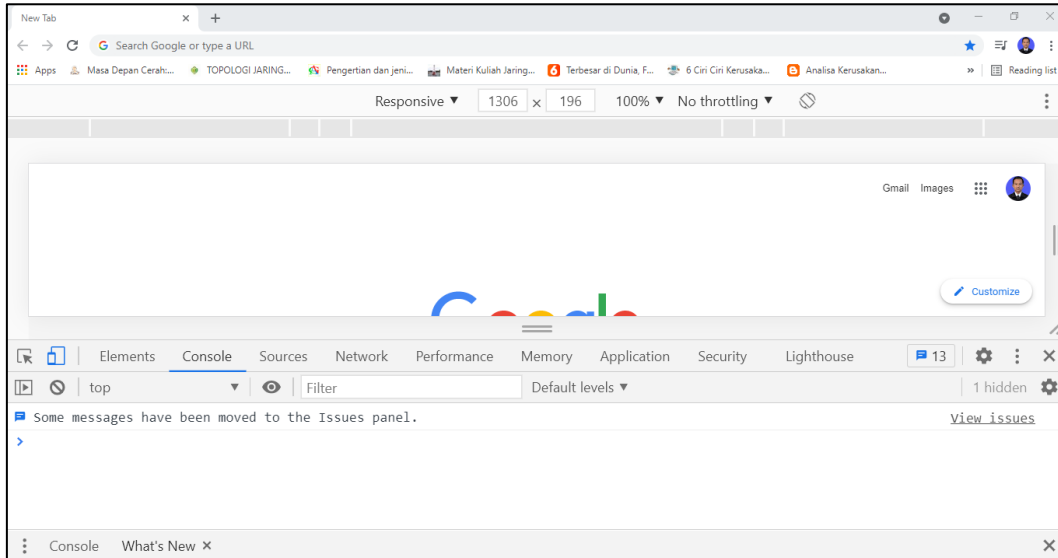
10.2 CARA MEMBUAT JAVASCRIPT DI WEBSITE

Javascript adalah bahasa pemrograman yang sederhana untuk dipelajari dan pengembangannya sederhana. Tidak diragukan lagi, menggunakan tag "skrip" diperlukan untuk menyematkan string skrip Java di halaman. Javascript juga digunakan dengan benar di header situs web Anda. Tetapi Anda juga dapat memilih untuk tidak menggunakannya di lain waktu atau dengan elemen lain. Kode dalam JS juga dapat dieksekusi sebagai file terpisah.

Saat ini, Javascript tidak hanya digunakan untuk aplikasi sisi klien (browser). Selain itu, Javascript digunakan di server, konsol, program desktop, aplikasi seluler, jaringan IoT, game, dan platform lainnya. Ini membuat Javascript lebih populer dan bahasa yang paling sering digunakan pengguna Github.

a) MENGENAL CONSOLE

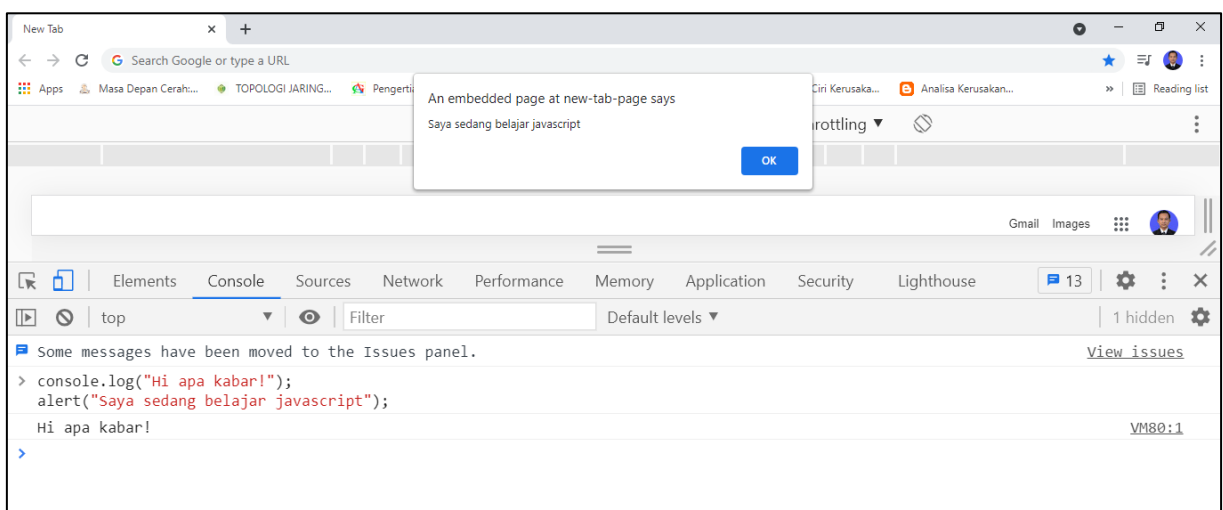
Konsole merupakan bagian untuk menampilkan pesan error jika kita melakukan kesalahan dalam menuliskan syntac javascript. Untuk memampilkkan halaman ini Anda dapat membuka browser kemudian klik kanan – Inspect elemet – Console



Gambar 10.1 Console pada browser

Didalam console ini kita jug adapt menulis fungsi atau code javascript dan hasilnya langsung bisa di tampilkan pada browser. Coba kita tulis kode seperti di bawah ini pada console :

```
console.log("Hi apa kabar!");  
alert("Saya sedang belajar javascript");
```



Gambar 10.2 hasil dari perintah console

Setelah penginstalan konsol Javascript, berikut ini yang mungkin Anda harapkan:

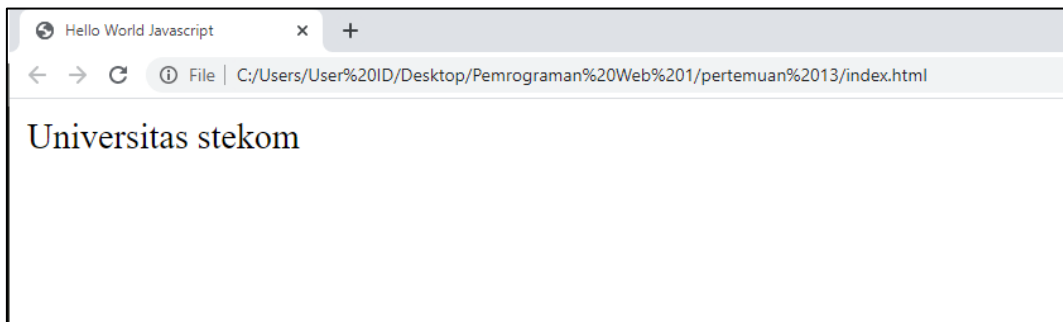
- Konsol dapat digunakan untuk memodifikasi kode atau fungsi Javascript;
- Konsol dapat digunakan untuk melihat kesalahan saat men-debug program.

a) MEMBUAT PROGRAM JAVASCRIPT

Mulai editor teks, lalu buat file dengan nama index.html dengan kode berikut:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Hello World Javascript</title>
5   </head>
6   <body>
7     <script>
8       console.log("Saya belajar Javascript");
9       document.write("Universitas Stekom");
10    </script>
11  </body>
12 </html>
13
```

maka akan terlihat seperti ini:

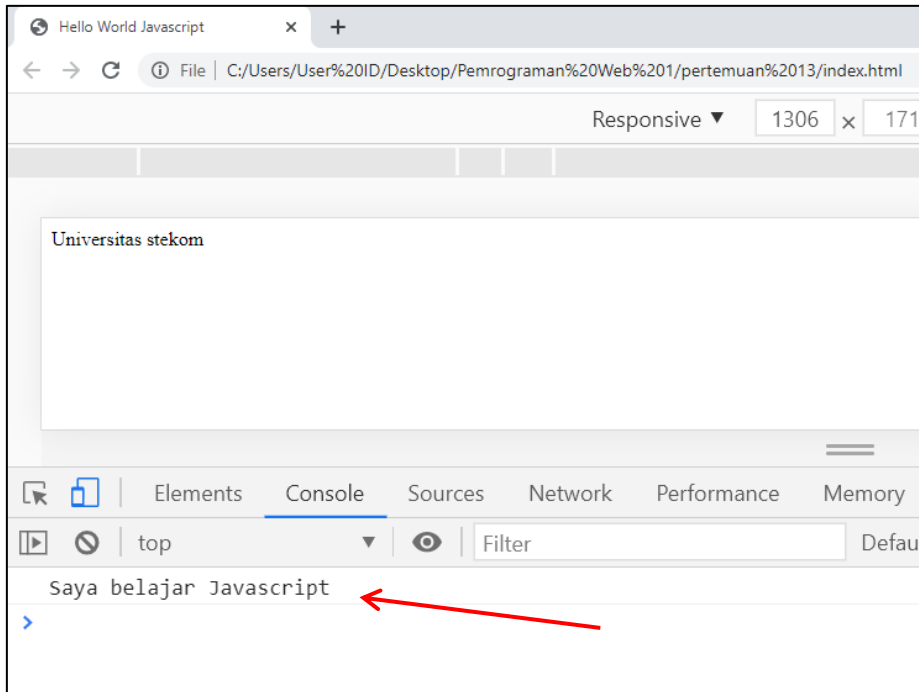


Gambar 10.3 hasil dari program javascript

Kenapa `console.log("Saya belajar Javascript");` tidak di tampilkan?

Karena faktanya `console.log()` atau fungsi terkait akan mengirimkan pesan ke konsol Javascript. Sebagai alternatif, karena fungsi `document.write()` adalah untuk menulis ke dokumen HTML, informasinya akan ditampilkan pada saat itu.

Mulai sekarang, mulailah mengetik javascript konsol. Jadi, kita akan melihat sebuah pesan. "Saya belajar Javascript":



Gambar 10.4 perintah console

Selain `document.write()` dan `console.write()` ada kode lain untuk menampilkan output pada javascript. Namun sebelum membahas lebih dalam, kita pelajari dulu mengenai Menulis Java Script di HTML

b) CARA MENULIS KODE JAVASCRIPT DI HTML

Pada contoh di bawah ini, kami telah memasukkan kode JavaScript menggunakan teknologi embedding HTML (ditempel). Meskipun semua telah dikatakan, masih ada beberapa hal yang memerlukan perhatian kita :

- a. **Embed** (kode HTML untuk Javascript diuji terus menerus. Konteks: saat ini)
- b. **Inline** (Kode JavaScript tertanam dalam atribut HTML.)
- c. **Eksternal** (File HTML dan kode Javascript kompatibel.)

1) Penulisan Kode Javascript Embed

Dalam hal ini, kode Javascript tertanam dalam HTML dengan menggunakan elemen "`<script>`." Tag ini dapat ditulis dalam tag "`<head>`" dan "`<body>`".

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Universitas Stekom</title>
5     <script>
6       // ini adalah penulisan kode javascript
7       // di dalam tag <head>
8       console.log("Hello Stekom, dari Head");
9     </script>
10  </head>
11  <body>
12    <p>Tutorial Javascript untuk Pemula</p>
13    <script>
14      // ini adalah penulisan kode javascript
15      // di dalam tag <body>
16      console.log("Hello Stekom, dari body");
17    </script>
18  </body>
19 </html>
20
```

Meskipun dapat di tulis pada bagian <head> ataupun <body>, Namun demikian, banyak orang menyarankan untuk menempatkannya di <body> karena akan membuat website memuat lebih cepat.

2) Penulisan Kode Javascript Inline

Dengan metode ini, kami akan menyematkan kode JavaScript dalam atribut HTML. Metode ini sering digunakan untuk mengaktifkan fungsi utama acara saat ini.

Saat ditautkan, diklik, misalnya.

```
<a href="#" onclick="alert('Yey!')">Klik aku!</a>
```

Atau bisa juga seperti :

```
6 <body>
7   <a href="javascript:alert('Anda Hebat..')">Klik aku!</a>
8 </body>
9 </html>
```

Hasilnya :



Gambar 10.5 inline javascript

Keterangan :

- Pada contoh pertama di atas, kode javascript dituliskan pada event atribut. Satu-satunya atribut event yang ada dalam HTML adalah onclick. Oleh karena itu, kode JavaScript dapat ditulis dalam event atribut.
- Pada HTML, atribut event ditulis terlebih dahulu dengan huruf on, seperti Atribut HTML yang dikenal sebagai onclick digunakan untuk event clicks. Silakan baca contoh di bagian bawah penjelasan ini.

- Pada contoh di bawah ini, aktifkan fungsi alert(). Fungsi ini adalah salah satu untuk menyajikan dialog. Kemudian, dengan menggunakan Javascript, kami menambahkan fungsi alert() ke atribut href. Atribut href biasanya digunakan untuk menentukan tautan atau alamat URL. Karena kami ingin menyematkan kode JavaScript di sini, kami akan mengonversi alamat tautan menjadi javascript dan kemudian menerapkan fungsionalitas yang dimaksud.

Contoh atribut event HTML **onkeyup**

```

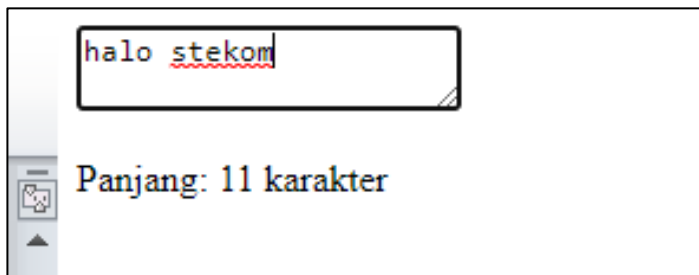
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Universitas Stekom</title>
5 </head>
6 <body>
7   <textarea onkeyup="document.getElementById('counter').innerText = this.value.
   length"></textarea>
8   <p>Panjang: <span id="counter">0</span> karakter</p>
9 </body>
10 </html>

```

Dalam contoh ini, kami menggunakan event keyup yang mendukung onkeyup. Kejadian ini terjadi saat kami menekan tombol pada keyboard. Sederhananya, ketika kita mengklik elemen "textarea", kode JavaScript yang ada di sana akan dieksekusi.

Kode Javascript ini digunakan untuk menampilkan berbagai karakter dinamis.

Hasil:



Gambar 10.6 jumlah karakter dengan javascript

Tabel 10.1 Event system dalam javascript

Event	Keterangan
onError	Event ini dibangkitkan ketika terjadi kesalahan. Event ini dimiliki oleh objek Window dan Image
onLoad	Event ini dibangkitkan ketika suatu objek selesai ditampilkan pada halaman web. Objek yang dimaksud pada keterangan ini adalah objek Window, Frame dan Image
onUnload	Event ini dibangkitkan ketika suatu objek telah dikeluarkan dari ruang memori, atau telah selesai digunakan. Objek ini adalah objek Window, Frame dan Image

Tabel 10.2 Event pengguna beserta keterangan

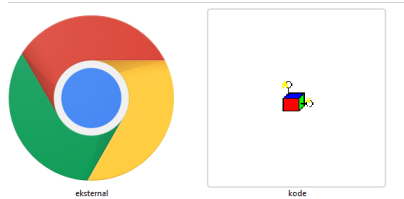
Event	Keterangan
onAbort	Event yang dimiliki oleh objek image ini akan dibangkitkan ketika sebuah gambar dihentikan proses pemunculannya yang dimungkinkan akibat penekanan tombol stop pada browser
onBlur	Event ini dibangkitkan ketika sebuah element (dalam hal ini adalah element window, frame, select, text, dan textarea) kehilangan focusnya
onChange	Event ini dibangkitkan ketika sebuah element (dalam hal ini adalah element select, text dan textarea) telah diubah nilainya sebelum element tersebut kehilangan focusnya
onClick	Event ini dibangkitkan ketika terjadi aksi klik (dilakukan oleh pengguna) terhadap element, dimana element yang dimaksud adalah semua element dari form yang dapat diklik seperti element button (tombol). Aksi klik yang dimaksud adalah adanya penekanan tombol klik kir mouse terhadap suatu element
onContextMenu	Event ini terbangkitkan ketika terjadi aksi klik kanan dari tombol mouse terhadap suatu element
onDbClick	Hampir sama dengan event onClick, namun akan dibangkitkan ketika terjadi aksi klik ganda terhadap suatu element
onFocus	Event ini adalah kebalikan dari event onBlur, dimana akan dibangkitkan ketika suatu element dikenai focus. Element-element yang dimaksud adalah element window, frame, select, text dan textarea
onHelp	Event ini dibangkitkan ketika terjadi penekanan terhadap tombol F1, yang umumnya digunakan berbagai aplikasi sebagai tombol shortcut untuk menampilkan fasilitas bantuan (help) . Anda dapat menggunakan event ini untuk mengalihkan fitur help milik browser menjadi fitur help yang Anda buat sendiri untuk halaman web Anda.
onKeyDown	Event ini dibangkitkan ketika terjadi penekanan ke bawah tombol keyboard
onKeyPress	Event ini dibangkitkan ketika terjadi aksi penekanan tombol keyboard. Aksi penekanan disini adalah sebuah aksi lengkap penekanan tombol keyboard, mulai dari memencet tombol keyboard tersebut hingga melepaskannya
onKeyUp	Event ini dibangkitkan ketika tombol keyboard yang awalnya ditekan lalu dilepas, ketika tombol dilepas event ini bangkit .
onMouseDown	Event ini dibangkitkan, ketika tombol mouse ditekan ke bawah
onMouseMove	Event ini dibangkitkan ketika mouse digerakkan (panah mouse berubah posisinya)
onMouseout	Event ini dibangkitkan ketika panah mouse keluar dari daerah lingkupan suatu element, yaitu element hyperlink dan element area.
onMouseover	Event ini dibangkitkan ketika panah mouse berada diatas element , yaitu element hyperlink dan element area .
onReset	Event yang dimiliki secara khusus oleh objek form ini akan dibangkitkan ketika terjadi penekanan terhadap tombol reset yang dimiliki form yang bersangkutan
onResize	Event ini dibangkitkan ketika jendela browser diubah ukurannya, baik diperbesar maupun diperkecil
onSelect	Event ini dibangkitkan ketika terjadi pemilihan teks pada lement text dan textarea dengan cara memberinya highlight (diblok)

onStop	Event ini dibangkitkan ketika pengguna melakukan penekanan terhadap tombol stop dari browsernya
onSubmit	Event ini dibangkitkan setelah terjadi penekanan tombol submit yang dimiliki sebuah form

3) Penulisan Kode Javascript Eksternal

Dalam metode ini, kami akan menyertakan kode JavaScript dalam file HTML yang diformat dengan ketat. Karena lebih mudah mengelola kode proyek dengan metode ini, cara ini biasanya digunakan untuk proyek-proyek besar.

Buat file html dan file js dalam 1 folder atau menyesuaikan folder



Gambar 10.7 penulisan javascript Eksternal

Isi file kode.js

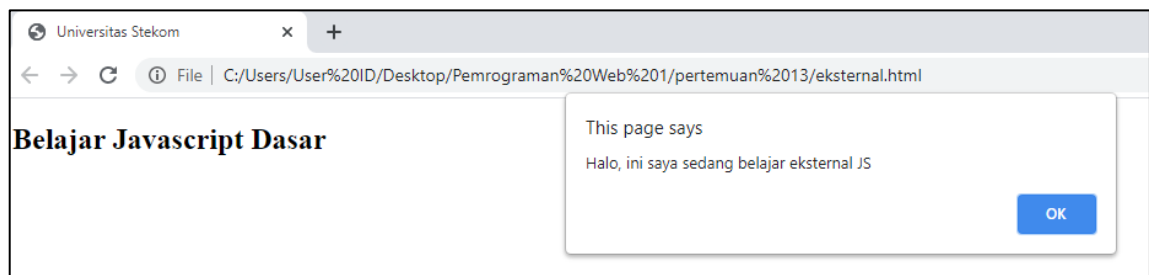
```
1 alert("Halo, ini saya sedang belajar eksternal JS");
```

Isi file eksternal.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Universitas Stekom</title>
5 </head>
6 <body>
7   <h2>Belajar Javascript Dasar</h2>
8
9   <!-- sisipkan file js disini -->
10  <script src="kode.js"></script>
11 </body>
12 </html>
13
```

Untuk menyisipkan file javascript, gunakan tag `<script> ... </script>`

Hasilnya :



Gambar 10.8 alert javascript

Selain itu kita juga dapat menyisipkan javascript yang ada di internet dengan memberikan alamat URL lengkapnya seperti :

```
<script src="https://kit.fontawesome.com/a076d05399.js"></script>
```

c) MENAMPILKAN OUTPUT PADA JAVASCRIPT

Keluaran istilah "keluaran" mengacu pada perangkat lunak yang biasanya digunakan untuk memberikan hasil akhir. Output sering disampaikan dalam bentuk teks dengan kemampuan cetak (). Ada dua cara untuk menampilkan keluaran di Javascript :

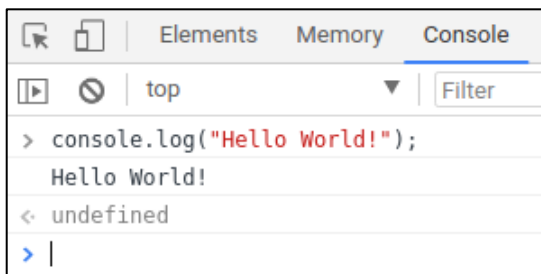
4.1 Menggunakan fungsi Console.log()

Fungsi console.log() adalah berfungsi untuk mengirim teks ke konsol Javascript.

Konteks penggunaan:

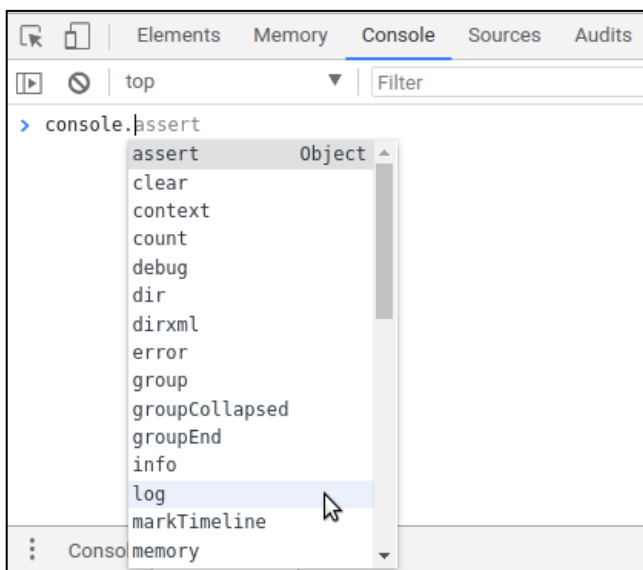
```
console.log("Hello World!");
```

Hasilnya dapat dilihat pada halaman berikutnya :



Gambar 10.9 console.log

Console.log() biasanya digunakan untuk debugging. Karena setiap pernyataan Javascript selalu muncul di Console. Ada beberapa fungsi debug yang tersedia selain console.log(), termasuk console.debug(), console.info(), console.error(), console.dir(), dan lainnya.



Gambar 10.10 atribut console

4.2 Menggunakan Fungsi Alert()

Fungsi alert() adalah untuk menampilkan kotak dialog. Fungsi terletak di objek jendela secara default. Teks komprehensif seperti itu dapat ditulis:

```
window.alert("Hello World!");
```

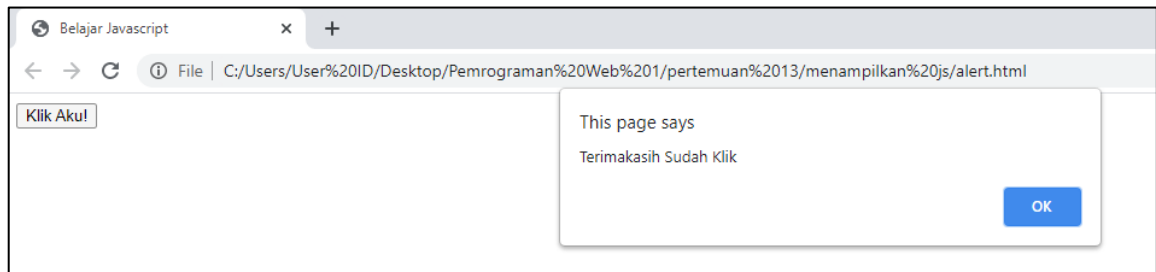
Bisa juga di tulis seperti :

```
alert("Hello World!");
```

Contoh :

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Belajar Javascript</title>
5   <script>
6     alert("Selamat datang di Universitas Stekom");
7
8     function sayHello(){
9       alert("Terimakasih Sudah Klik");
10    }
11  </script>
12 </head>
13 <body>
14   <button onclick="sayHello()">Klik Aku!</button>
15 </body>
16 </html>
17
```

Hasilnya



Gambar 10.11 alert dengan javascript

Pesan alert diatas di masukkan ke dalam function sayHello(). Function akan kita pelajari pada BAB berikutnya.

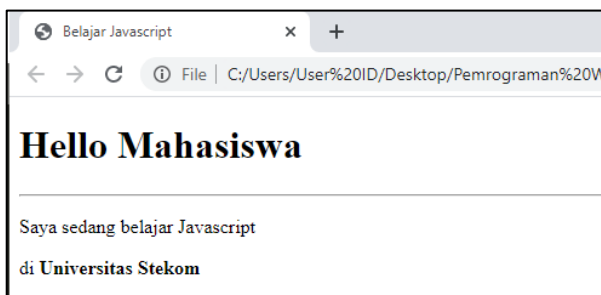
4.3 Menggunakan Fungsi document.write()

Istilah "objek dokumen" mengacu pada objek Javascript yang menampilkan konten HTML. Ada fungsi write() pada objek dokumen yang dapat digunakan untuk mengirimkan konten ke dokumen HTML.

Contoh:


```
documentwrite.html x
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Belajar Javascript</title>
5   <script>
6     document.write("<h1>Hello Mahasiswa</h1>");
7     document.write("<hr>");
8     document.write("<p>Saya sedang belajar Javascript</p>");
9     document.write("<di <b>Universitas Stekom</b>");
10  </script>
11 </head>
12 <body>
13
14 </body>
15 </html>
```

Hasilnya



Gambar 10.12 document.write JS

Seiring dengan fungsi write(), dokumen objek juga menyediakan banyak alat lain untuk memanipulasi dokumen HTML.

4.4 Menggunakan innerHTML

innerHTML adalah properti elemen HTML (objek) yang berisi string HTML. Dengan menggunakan innerHTML, kami dapat mengirim data ke elemen yang lebih spesifik.

Contoh:

```
inner.html x
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Universitas Stekom</title>
5 </head>
6 <body>
7
8   <h1>Tutorial Javascript untuk Pemula</h1>
9   <div id="hasil-output"></div>
10
11
12   <script>
13     // membuat objek elemen
14     var hasil = document.getElementById("hasil-output");
15
16     // menampilkan output ke elemen hasil
17     hasil.innerHTML = "<p>Aku suka Javascript</p>";
18   </script>
19
20 </body>
21 </html>
```

Hasilnya



Gambar 10.13 innerHTML JS

4.5 Menampilkan Dengan Dialog

Memanfaatkan fungsi peringatan adalah metode `alert()`. Fungsi penting dari `alert()` adalah salah satu yang akan meningkatkan komunikasi. Dialog adalah jenis jendela yang paling umum digunakan untuk berkomunikasi dengan pengguna. Ada tiga kotak dialog berbeda dalam Javascript:

- `alert();`
- `confirm();`
- `prompt();`

Ketiga dialog ini memiliki perilaku dan kegunaan yang berbeda-beda.

- **Jendela dialog `alert();`**

Dialog `alert()` biasanya digunakan untuk menampilkan teks atau informasi peringatan yang mendesak. Fungsi `Alert()` aktif di objek `window`. Seseorang dapat menggunakannya dengan cara yang dijelaskan dalam artikel ini:

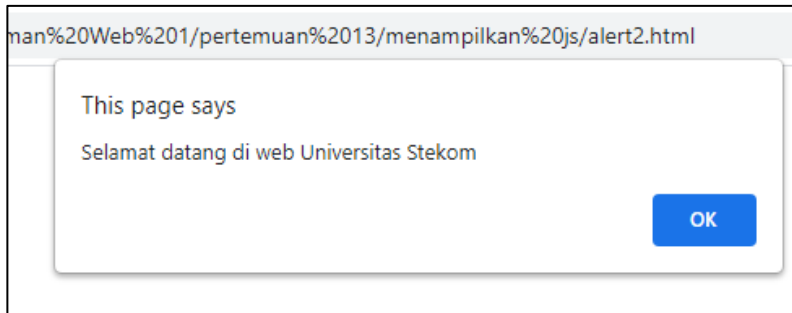
```
window.alert("Hello World!");
```

Atau seperti ini:

```
alert("Hello kawan");
```

Karena jendela memiliki fokus global, kita bebas mengabaikannya. Dialog `alert()` tidak akan menampilkan nilai saat ini saat perintah dijalankan.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Alert</title>
5 </head>
6 <body>
7   <script>
8     alert("Selamat datang di web Universitas Stekom");
9   </script>
10 </body>
11 </html>
12
```



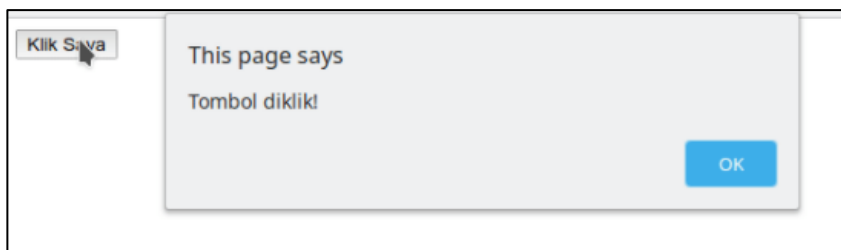
Dialog alert() hanya membutuhkan satu parameter, yaitu teks yang akan ditampilkan pada dialog. Kami menyediakan teks dalam contoh di bawah ini. Salam dan selamat datang di pelajaran Javascript.

Pertanyaan: Bagaimana saya bisa menyertakan kotak dialog alert() ke dalam acara yang sesuai, katakan saja dengan lantang saat tombol diklik?

Kita dapat melakukannya dengan menambahkan fungsionalitas dialog ke event listener. Dengan HTML, Anda dapat menambahkan fungsi alert() ke atribut onClick sehingga akan ditampilkan saat elemen diklik.

Contoh:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Dialog Alert</title>
  </head>
  <body>
    <button onClick="alert('Tombol diklik!')">Klik Saya</button>
  </body>
</html>
```



➤ **Jendela dialog confirm();**

Kotak dialog confirm() digunakan untuk mendapatkan konfirmasi sebelum melakukan tindakan yang relevan.

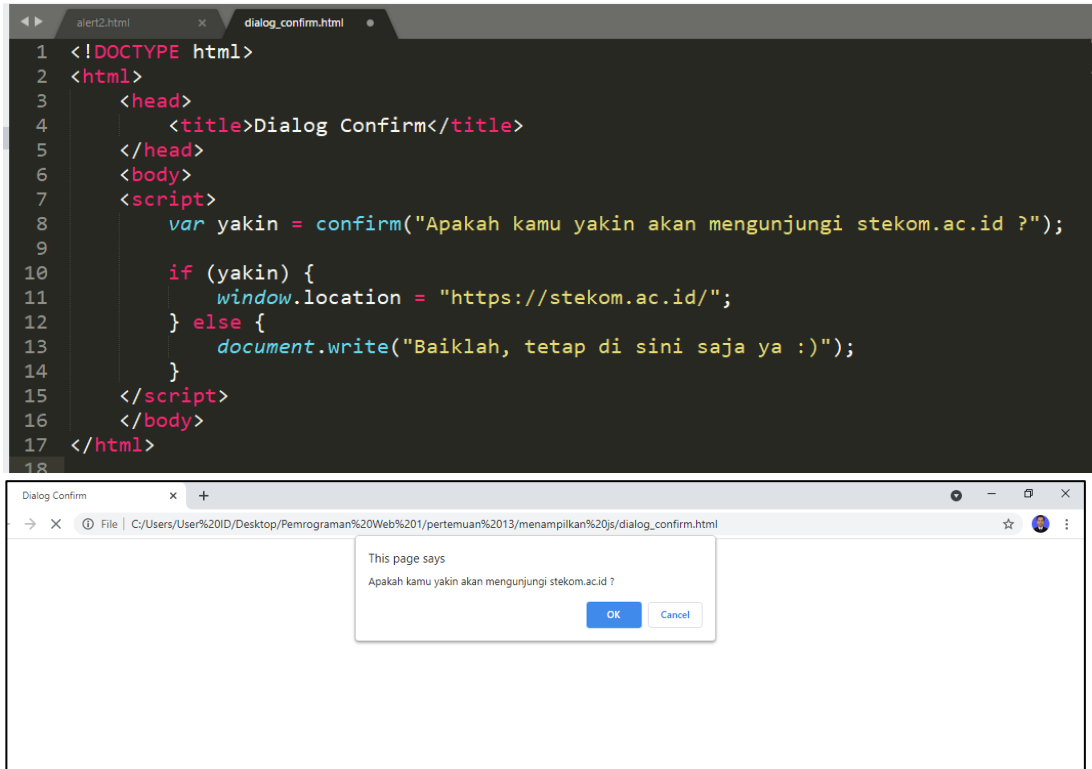
Misalnya: Saat kita sedang mendiskusikan sesuatu, penting untuk melakukan percakapan untuk menegaskan (). Jadi tindakan yang dimaksud cukup berwawasan luas. Kotak dialog konfirmasi memiliki fungsi konfirmasi ().

Contoh:

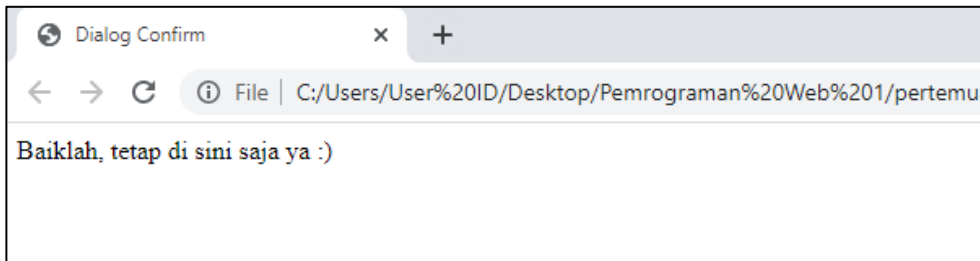
```
confirm("Apakah anda yakin akan menghapus?");
```

Jika Anda memilih tombol OK di kotak dialog confirm(), hasilnya akan benar; jika Anda memilih tombol Batal, hasilnya akan salah. Kita dapat menyematkan kalimat ini ke dalam variabel yang akan dievaluasi.

Contoh:



Jika di klik cancel akan menampilkan



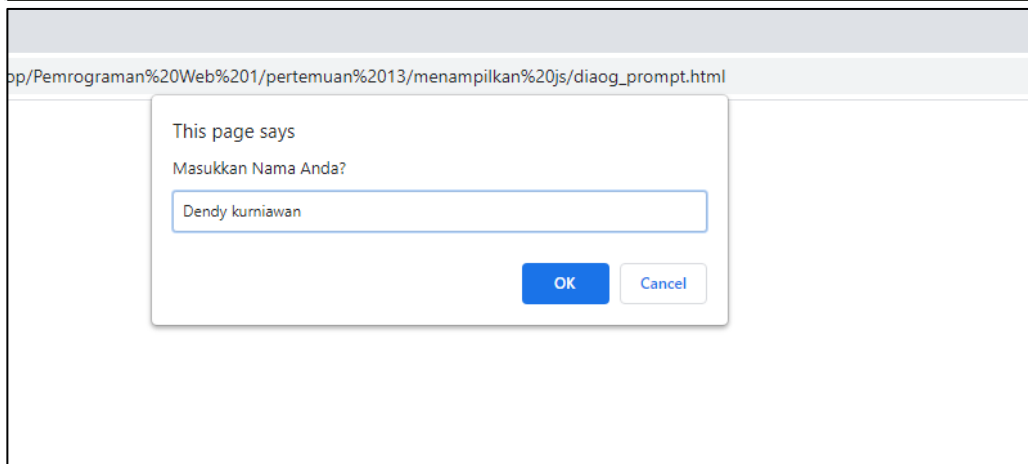
Jika di klik OK akan menampilkan website <https://stekom.ac.id/>

➤ **Jendela dialog prompt();**

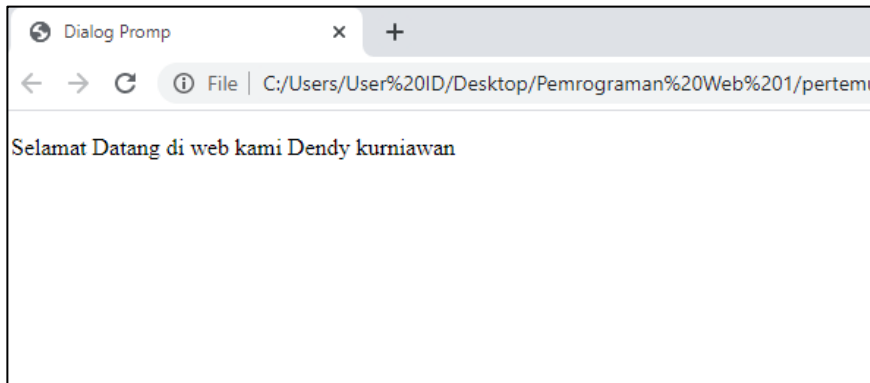
Dialog prompt() memiliki fungsi menerima input pengguna. Masukan dari pengguna akan digunakan untuk menghasilkan string di kotak dialog prompt().

Contoh:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Dialog Promp</title>
5   </head>
6   <body>
7     <script>
8       var nama = prompt("Masukkan Nama Anda?", "");
9       document.write("<p>Selamat Datang di web kami "+ nama +"</p>");
10    </script>
11  </body>
12 </html>
13
```



Setelah di klik OK



Dialog prompt() memiliki beberapa parameter wajib yang harus ditentukan:

- Teks yang akan dimasukkan ke dalam formulir;
- kolom input default.

Pada contoh di atas, kami memberikan nilai default sebagai string dengan karakter garis bawah " ".

```
var nama = prompt("Masukkan Nama Anda?", "");
document.write("<p>Selamat Datang di web kami
```

BAB 11

VARIABEL, TIPE DATA & OPERATOR

TUJUAN :

- a. Mahasiswa Mampu Memahami dan Membuat serta Menampilkan Data dengan Variabel
- b. Mahasiswa Memahami Tipe Data di JavaScript
- c. Mahasiswa Memahami dan Meng Implementasikan Operator di java script

Overview :

Penulisan suatu value yang berkelanjutan dan panjang, dapat di permudah dengan sebuah variable.jadi pada dasarnya variable ini adalah suatu tempat atau wadah kosong yang di beri nama dan dapat menampung value bertipe data apa saja. Ada persyaratan dalam membuat sebuah variable yang akan di bahas pada bab ini.

Javascript adalah bahasa dengan penekanan kuat pada pengetikan dinamis, oleh karena itu tidak seperti bahasa dengan penekanan kuat pada pengetikan statis seperti C, C++, Java, dan sebagainya, di mana tipe data harus ditentukan pada saat pembuatan variabel, tidak memerlukan ini. (harus mendeklarasikan tipe data). Namun demikian kita harus mengetahui type data apa saja yang ada di javascript.

Operator di gunakan untuk melakukan pemrosesan data dari pemberian nilai variable, perbandingan, aritmatika dan logika.

11.1 VARIABEL

Variabel adalah nama yang berisi nilai numerik. Variabel dapat dibuat dengan berbagai jenis nilai numerik, termasuk teks (teks), angka (angka), objek, array, dan lain-lain. Kita dapat menyimpulkan bahwa variabel ini berfungsi sebagai cara untuk menunjukkan sesuatu.



Gambar 11.1 Ilustrasi Variabel

1) Cara Membuat Variabel

Cara khas untuk membuat variabel dalam javascript adalah dengan menggunakan kata kunci var diikuti dengan nama dan definisi variabel.

Contoh:

```
var title = "Belajar Pemrograman Javascript";
```

Pada contoh di bawah ini, kita membuat sebuah variabel bernama judul dengan teks "Pelajari Javascript" sebagai nilainya.

Contoh lagi:

```
var siteName = "Universitas Stekom";  
var url = "https://www.stekom.ac.id";  
var visitorCount = 5921;
```

Perhatikan!

Dalam contoh di atas, kami menggunakan kapital atau desimal signifikan untuk mewakili nama variabel yang terdiri dari dua subskrip. Mengapa Anda tidak menggunakan garis bawah? Dalam Javascript, camelCase digunakan di seluruh kode sebagai namespace. Penggunaan kotak ular atau jenis garis bawah ini diperbolehkan saat ini :

```
var nama_variabel = "universitas Stekom";  
var visitor_count = 1234;
```

Ini tidak akan menimbulkan masalah karena perangkat lunak masih berjalan. Meskipun demikian, sebagian besar pemrogram Javascript menggunakan camelCase.

Perlu diingat, selain menggunakan kata kunci var, Anda juga dapat membuat variabel menggunakan kata kunci let atau tanpa awalan lainnya..

```
// membuat variabel dengan kata kunci let  
let price = 15000;  
  
// membuat variabel tanpa awalan apapun  
stock = 12
```

Apa tujuan penggunaan var, let, dan tanpa awalan?

Kesamaan antara pengguna dan jangkauannya jelas. Di masa mendatang, variabel yang dihapus akan diterjemahkan.

Selanjutnya, "Apa yang akan terjadi jika nilai variabel tidak disetel?"

Akibatnya, nilai variabel tidak ditentukan (belum ditentukan).

Contoh:

```
var x;
```

Maka variable x akan bernilai undefined.

2) Aturan Membuat Variabel

- Penamaan variabel tidak boleh menggunakan angka di depannya.

```
//salah
var 1nama = "Dendy Kurniawan";

//benar
var nama1 = "Dendy Kurniawan";
```

- Penamaan variabel boleh menggunakan awal underscore.

```
var _nama = "Dendy Kurniawan";
```

- Penamaan variabel dianjurkan menggunakan camelCase apabila terdiri dari dua suku kata.

```
var namaLengkap = "Dendy Kurniawan";
```

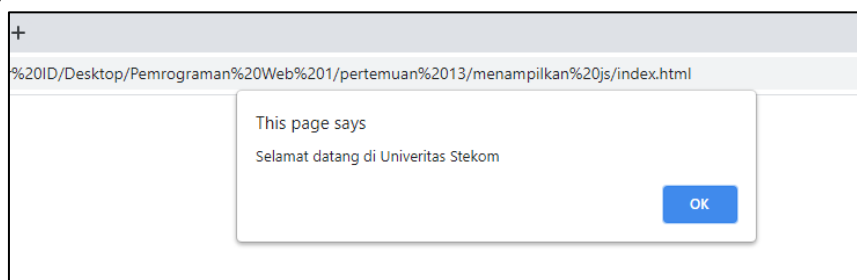
3) Menampilkan Isi Variabel

Untuk menampilkan variabel isi, Anda dapat menggunakan banyak fungsi untuk menghasilkan keluaran seperti:

- **Fungsi console.log()** digunakan untuk mencetak javascript dalam console ;
- **Fungsi document.write()** digunakan untuk mencetak ke dokumen HTML;
- dan **Fungsi alert()** digunakan untuk mencetak ke jendela dialog.


```
index.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Belajar Variabel dalam Javascript</title>
5   <script>
6     // membuat variabel
7     var name = "Univeritas Stekom";
8     var jumlahMhs = 5322;
9     var akreditasi = true;
10    var url = "https://www.stekom.ac.id";
11
12    // menampilkan variabel ke jendela dialog (alert)
13    alert("Selamat datang di " + name);
14
15    // menampilkan variabel ke dalam HTML
16    document.write("Nama Institusi: " + name + "<br>");
17    document.write("Jumlah Mahasiswa: " + jumlahMhs + "<br>");
18    document.write("Terakreditasi: " + akreditasi + "<br>");
19    document.write("Alamat URL: " + url + "<br>");
20  </script>
21 </head>
22 <body>
23
24
25 </body>
26 </html>
```

Hasilnya :



Gambar 11.2 Menampilkan Alert

Setelah di klik OK



Gambar 11.3 Menampilkan Variabel

Arti tambah (+) pada contoh di atas berarti menyambung, bukan mengalikan.

4) Mengisi Ulang Variabel

Variabel dianggap dapat diubah karena memungkinkan manipulasi data di sekitarnya (berubah).

Contoh:

```
// mula-mula kita buat variabel dengan isi seperti ini
var age = 18;

// lalu kita isi ulang
age = 21;
```

Mengapa kita tidak menggunakan kata kunci var saat mengatur variabel ulang nilai?

Karena kata "var" diperlukan saat membuat variabel saja, Sebaliknya, kita cenderung menulis seperti yang terlihat di atas saat ingin makan telur. Setiap kali kita menggunakan kata "var", itu berarti kita membuat variabel baru daripada mengubah yang sudah ada.

11.2 TIPE DATA

Data menurut jenisnya adalah variasi data yang dapat dicocokkan dalam suatu variabel. Ada banyak jenis data dalam program Javascript:

- String (teks)
- Integer atau Number (bilangan bulat)
- Float (bilangan Pecahan)
- Boolean
- Object

Javascript adalah bahasa dengan penekanan kuat pada pengetikan dinamis, oleh karena itu tidak seperti bahasa dengan penekanan kuat pada pengetikan statis seperti C, C++, Java, dan sebagainya, di mana tipe data harus ditentukan pada saat pembuatan variabel, tidak memerlukan ini. (harus mendeklarasikan tipe data).

Contoh:

```
var name = "Dian";
var age = 22;
var single = true;
```

Javascript akan secara otomatis mengidentifikasi jenis data yang kami berikan untuk variabel tersebut.

Penjelasan Variabel contoh diatas sebagai berikut:

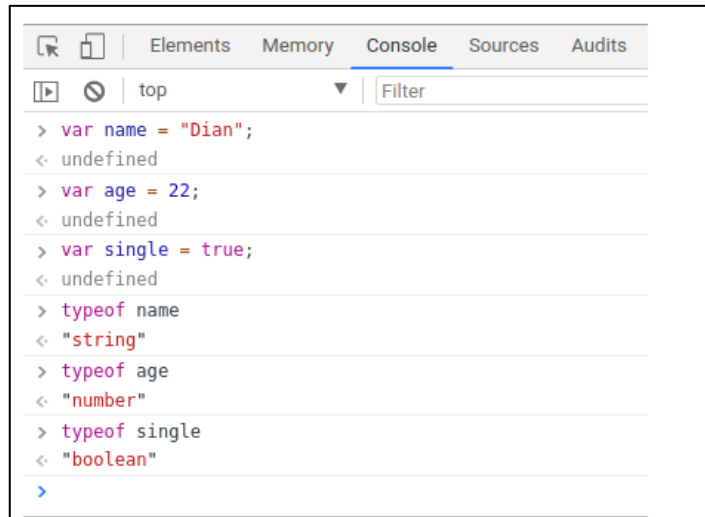
- name memiliki tipe data String;
- age memiliki data integer;
- single memiliki data boolean.

Hal ini juga dapat dibuktikan dengan menggunakan jenis kata kunci typeof.

Contoh:

```
typeof name;
typeof age;
typeof single;
```

Hasilnya pada console:



Gambar 11.4 Menampilkan tipe data dengan console

Bagaimana lagi kita bisa belajar tentang tipe data selain dengan menggunakan typeof? Dengan menggunakan metode eksekusi, kita dapat memahaminya.

Contoh:

```

// untuk tipe data teks atau string penulisannya selalu diapit dengan
tanda petik
var author = "Universitas Stekom";

// sedangkan untuk tipe data number dan boolean tidak diapit dengan tanda
petik
var available = true; // <- tipe data boolean hanya bernilai true dan false
var total = 14;

```

11.3 OPERATOR

Operator merupakan konsep kunci yang perlu dipahami dalam suatu program tertentu. Karena fakta bahwa kami akan sering menggunakannya untuk melakukan berbagai operasi di dalam zona berbahaya lunak. Apa operatornya?

Misalkan ada variabel seperti ini

```

var a = 3;
var b = 4;

```

Bagaimana cara menjumlahkan a dan b ? kita dapat menggunakan tanda plus (+).

```

var c = a + b;

```

Hasil penjumlahan akan disimpan pada variable c. Itulah salah satu contoh dari operator. Operator adalah simbol yang digunakan untuk melakukan operasi pada satu nilai dan variabel.

Operator program pemrograman enam jenis:

- Operator aritmatika;
- Operator Assignment (Penugasan);
- Operator Perbandingan (relasi);
- Operator Logika;
- Operator Bitwise;

➤ Operator Ternary;

Sebenarnya disetiap Bahasa pemrograman apapun, operator wajib ada.

a) **OPERATOR ARITMATIKA**

Operator matematika adalah orang yang melakukan operasi matematika misalnya pengurangan, penjumlahan, pembagian, perkalian, dll. Operator dalam matematika terdiri dari:

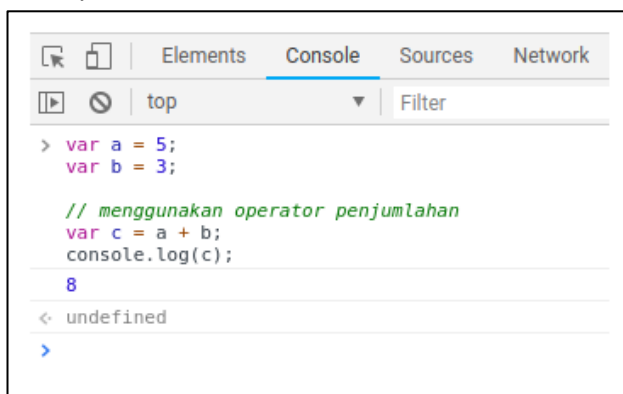
Nama Operator	Simbol
Penjumlahan	+
Pengurangan	-
Perkalian	*
Pemangkatan	**
Pembagian	/
Sisa Bagi	%

Contoh :

```
var a = 5;
var b = 3;

// menggunakan operator penjumlahan
var c = a + b;
console.log(c);
```

Hasilnya :



Gambar 11.5 Operator Aritmatika

Contoh lain yang di cetak lengkap dengan document.write

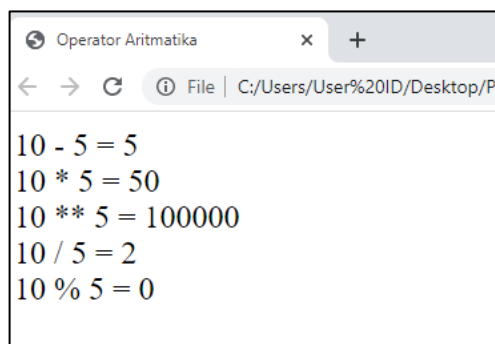
```
operator.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Operator Aritmatika</title>
5 </head>
6 <body>
```

```

7      <script>
8          var a = 10;
9          var b = 5;
10         var c =0;
11
12         // pengurangan
13         c = a - b;
14         document.write(`${a} - ${b} = ${c}<br/>`);
15
16         // Perkalian
17         c = a * b;
18         document.write(`${a} * ${b} = ${c}<br/>`);
19
20         // pemangkatan
21         c = a ** b;
22         document.write(`${a} ** ${b} = ${c}<br/>`);
23
24         // Pembagian
25         c = a / b;
26         document.write(`${a} / ${b} = ${c}<br/>`);
27
28         // Modulo
29         c = a % b;
30         document.write(`${a} % ${b} = ${c}<br/>`);
31     </script>
32 </body>
33 </html>

```

Hasilnya :



Agambar 11.6 mencetak operator dengan document write

Penggunaan tanda petik `...` berfungsi untuk mencetak seluruh syntac javascript. Sedangkan penggunaan tanda `\${..}` di gunakan untuk mengambil value dari variable yang ada di dalam kurung {}. Atau sebenarnya bisa di tulis dengan cara

```
document.write(a + " + " + b + " = " + c + "<br>");
```

Penggabungan Teks

Dalam Javascript, jika Anda melakukan operasi pada suatu jenis data string atau tex dengan menggunakan tanda tambah (+), yang terjadi adalah penggabungan, bukan tanda tambah.

Contoh:

```
var a = "10" + "2";
```

Maka hasilnya :

Kenapa bukan 12? Karena kedua angka yang dimaksud adalah tipe data string. Perlu diingat bahwa dia duduk dengan pensil. Silahkan coba syntac dibawah ini melalui console dan amati hasilnya:

```

> 10 + "3"
< "103"
> 10 * "3"
< 30
> 10 * "tiga"
< NaN
> |
  
```

Gambar 11.7 Penggabungan Teks

b) OPERATOR PENUGASAN

Operator penugasan adalah operator yang digunakan untuk memberi perintah pada variabel. Bias selalu digunakan untuk membuat variabel.

Contoh:

```
var a = 19;
```

Variabel a memberikan instruksi untuk menambahkan angka 19.

Operator penugasan terdiri dari:

Nama Operator	Sombol
Pengisian Nilai	=
Pengisian dan Penambahan	+=
Pengisian dan Pengurangan	-=
Pengisian dan Perkalian	*=
Pengisian dan Pemangkatan	**=
Pengisian dan Pembagian	/=
Pengisian dan Sisa bagi	%=

Operator penugasan mirip dengan operator dalam matematika. Saya juga menggunakannya untuk operasi matematika.

Contoh:

```
var jumlahView = 12;

// menggunakan operator penugasan penjumlahan
// untuk menambah nilai
jumlahView += 1;
```

Hasil:

Variabel jumlahView akan bertambah satu.

Maksudnya dari jumlahView += 1 adalah sebagai berikut :

```
jumlahView = jumlahView + 1;
```

Singkatnya:

Isi variabel "jumlahView" dengan peningkatan lebar dari nilai sebelumnya menggunakan sudut 1++ dan — dapat digunakan untuk mengidentifikasi operator dengan armada besar dan fluktuatif yang ditetapkan untuk beroperasi sebagai satu kesatuan.

Contoh:

```
var a = 2;
a++;
```

Sehingga nilai variabel a akan menjadi 3. Oleh karena itu, pertanyaannya adalah: Bagaimana hubungan operator penugasan dan operator aritmatika? Operator aritmatika hanya melakukan operasi yang melibatkan matematika, sedangkan operator penugasan melakukan operasi matematika dan pengisian.

Berikut adalah contoh operatornya:

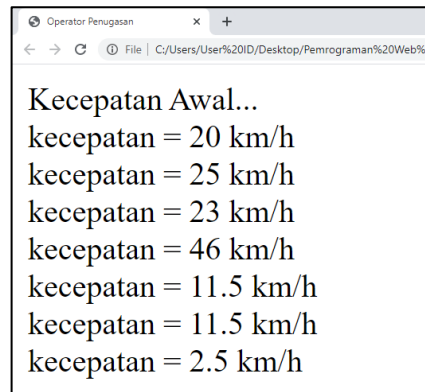
```
operator_penugasan.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Operator Penugasan</title>
5 </head>
6 <body>
7 <script>
8     document.write("Kecepatan Awal...<br>");
9     // pengisian nilai
10    var kecepatan = 20;
11    document.write("kecepatan = "+ kecepatan + " km/h <br/>");
12
13    // pengisian dan menjumlahkan dengan 5
14    kecepatan += 5;
15    document.write("kecepatan = "+ kecepatan + " km/h <br/>");
16
17    // pengisian dan pengurangan dengan 2
18    kecepatan -= 2;
19    document.write("kecepatan = "+ kecepatan + " km/h <br/>");
20
21    // pengisian dan perkalian dengan 2
22    kecepatan *= 2;
23    document.write("kecepatan = "+ kecepatan + " km/h <br/>");
24
25    // pengisian dan pembagian dengan 4
26    kecepatan /= 4;
```

```

27     document.write("kecepatan = "+ kecepatan + " km/h <br/>");
28
29     // pengisian dan pemangkatan dengan 2
30     kecepatan *= 1;
31     document.write("kecepatan = "+ kecepatan + " km/h <br/>");
32
33     // pengisian dan modulo dengan 3;
34     kecepatan %= 3;
35     document.write("kecepatan = "+ kecepatan + " km/h <br/>");
36 </script>
37 </body>
38 </html>

```

Hasilnya :



Gambar 11.8 Operator Penugasan

c) OPERATOR RELASI / PERBANDINGAN

Operatur relasi ini digunakan untuk membandingkan dua bilangan dikenal sebagai operator perbandingan atau relasi. Operator untuk perbandingan akan mengembalikan nilai boolean benar atau salah.

Operator untuk koneksi terdiri dari:

Nama Operator	Simbol
Lebih Besar	>
Lebih Kecil	<
Sama Dengan	== atau ===
Tidak Sama dengan	!= atau !==
Lebih Besar Sama dengan	>=
Lebih Kecil Sama dengan	<=

Contoh :

```

operator_perbandingan.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Operator Perbandingan</title>
5 </head>
6 <body>
7   <script>
8     var a = 25;
9     var b = 15;

```

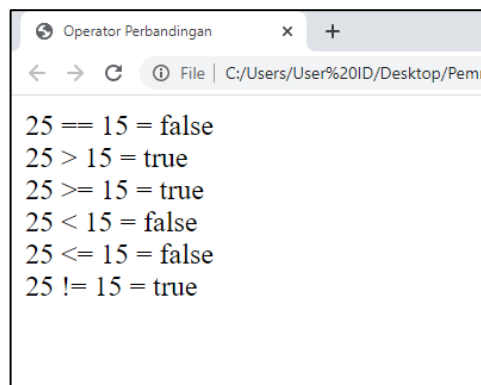


```

10
11 // sama dengan
12 var hasil = a == b;
13 document.write(`${a} == ${b} = ${hasil}<br/>`);
14
15 // lebih besar
16 var hasil = a > b;
17 document.write(`${a} > ${b} = ${hasil}<br/>`);
18
19 // lebih besar samadengan
20 var hasil = a >= b;
21 document.write(`${a} >= ${b} = ${hasil}<br/>`);
22
23 // lebih kecil
24 var hasil = a < b;
25 document.write(`${a} < ${b} = ${hasil}<br/>`);
26
27 // lebih kecil samadengan
28 var hasil = a <= b;
29 document.write(`${a} <= ${b} = ${hasil}<br/>`);
30
31 // tidak samadengan
32 var hasil = a != b;
33 document.write(`${a} != ${b} = ${hasil}<br/>`);
34 </script>
35 </body>
36 </html>

```

Hasilnya :



Gambar 11.9 Operator Relasi

Satu-satunya hal yang akan berubah saat menggunakan simbol == adalah hanya akan membandingkan nilainya saja. Sebaliknya, penggunaan === selain nilai, dia juga akan dibandingkan dengan jenis tipe data. Contoh :

```

// ini akan bernilai true
var a = "4" == 4; //-> true

// sedangkan ini akan bernilai false
var b = "4" === 4; //-> false

```

Mengapa Nilai B tidak benar? "4" (string) dan 4 (integer) Tipe dataanya berbeda.

d) OPERATOR LOGIKA

Untuk operasi yang melibatkan dua nilai boolean, logika operator digunakan. Operator ini terbuat dari :

Nama Operator	Simbol
Logika AND	&&
Logika OR	
Negasi/kebalikan	!

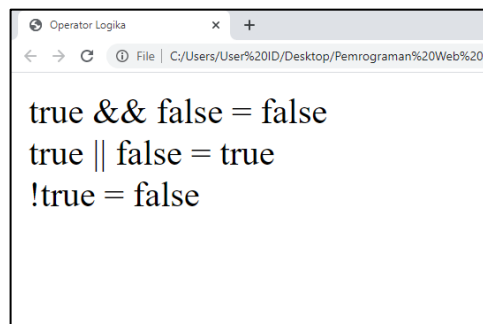
Contoh :

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Operator Logika</title>
5 </head>
6 <body>
7   <script>
8     var a = 25;
9     var b = 15;
10
11     var benar = a > b;
12     var salah = a < b;
13
14     // operator && (and)
15     var hasil = benar && salah;
16     document.write(`${benar} && ${salah} = ${hasil}<br/>`);
17
18     // operator || (or)
19     var hasil = benar || salah;
20     document.write(`${benar} || ${salah} = ${hasil}<br/>`);
21
22     // operator ! (not)
23     var hasil = !benar;
24     document.write(`${!benar} = ${hasil}<br/>`);
25
26   </script>
27 </body>
28 </html>

```

Hasilnya :



Gambar 11.10 Operator Logika

e) **OPERATOR BITWISE**

Nama operator yang digunakan untuk operasi berbasis bit adalah operator bitwise (biner). Operator ini terdiri dari:

Nama	Simbol di Java
AND	&
OR	
XOR	^
Negasi/kebalikan	~
Left Shift	«
Right Shift	»
Left Shift (unsigned)	«<
Right Shift (unsigned)	>»

Operator ini bekerja dengan tipe data berikut: int, long, short, char, dan byte. Operator ini akan berpindah dari bit ke bit. Sebagai contoh, kita memiliki variabel a = 60 dan b = 13. Jika dibuat dalam bentuk biner, hasilnya adalah sebagai berikut:

```
a = 00111100
b = 00001101
```

(Perhatikan biner, angka 0 dan 1) Setelah itu, operasi bitwise dilakukan.

Operasi AND

```
a = 00111100
b = 00001101
a & b = 00001100
```

Operasi OR

```
a = 00111100
b = 00001101
a | b = 00111101
```

Operasi XOR

```
a = 00111100
b = 00001101
a ^ b = 00110001
```

Operasi NOT (Negasi/kebalikan)

```
a = 00111100
~a = 11000011
```

Konsensus pada dasarnya sama dengan Operator Logika. Dengan kata lain, Bitwise digunakan untuk mata uang.

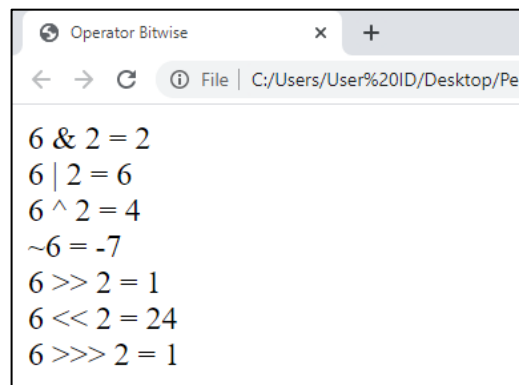
Agar lebih jelas, baca berikut ini:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Operator Bitwise</title>
5 </head>
6 <body>
7   <script>
8     var x = 6;
9     var y = 2;
10
11     // operator bitwise and
12     var hasil = x & y;
13     document.write(`${x} & ${y} = ${hasil}<br/>`);
14
15     // operator bitwise or
16     var hasil = x | y;
17     document.write(`${x} | ${y} = ${hasil}<br/>`);
18
19     // operator bitwise xor
20     var hasil = x ^ y;
21     document.write(`${x} ^ ${y} = ${hasil}<br/>`);
22
23     // operator negasi
24     var hasil = ~x;
25     document.write(`${x} = ${hasil}<br/>`);
26
27     // operator bitwise right shift >>
28     var hasil = x >> y;
29     document.write(`${x} >> ${y} = ${hasil}<br/>`);
30
31     // operator bitwise right shift <<
32     var hasil = x << y;
33     document.write(`${x} << ${y} = ${hasil}<br/>`);
34
35     // operator bitwise right shift (unsigned) >>>
36     var hasil = x >>> y;
37     document.write(`${x} >>> ${y} = ${hasil}<br/>`);
38
39   </script>
40 </body>
41 </html>

```

Hasilnya :



Gambar 11.11 Operator Bitwise

f) OPERATOR TERNARY

Ternary adalah operatornya sekarang. Operator ternary adalah contoh operator dengan tiga asal bagian yang berbeda. Operator -- operator sebelumnya hanya memiliki dua kantong, yaitu kantong kanan dan kiri. Operator biner digunakan di sini. Ada bagian tengah, kiri dan kanan untuk operator trinary secara terpisah.

bagian kiri <operator> bagian tengah <operator> bagian kanan

Operator ternary dalam Javascript biasanya digunakan untuk membuat pernyataan if/else. Tanda tanya dan titik dua merupakan simbol operator untuk ternaries (?). Bentuk dengan cara ini :

<kondisi> ? "benar" : "salah"

Ketahuiilah bahwa Anda dapat menyatakan suatu <kondisi> dengan ekspresi yang menunjukkan benar atau salah. Jika kondisi benar maka dipilih "benar", begitu pula jika kondisi salah maka dipilih "salah". Operator ini unik, karena memunculkan pertanyaan.

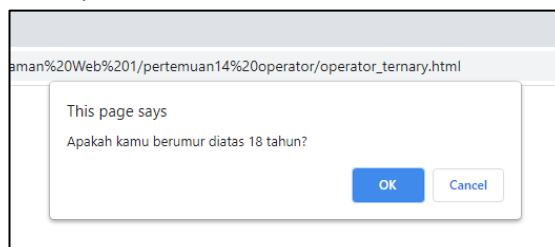


Gambar 11.12 Operator Ternary

Pada contoh di atas, "Kamu suka aku" mengacu pada pertanyaan atau kondisi yang akan diperiksa. Jika rahangnya kencang, katakan itu. Mirip dengan itu, tidak akan. Lebih jelasnya, saya akan mengatakan bahwa Anda harus melanjutkan.

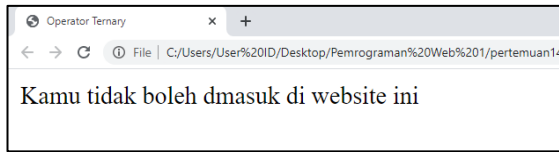
```
operator_ternary.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Operator Ternary</title>
5 </head>
6 <body>
7
8   <script>
9     var pertanyaan = confirm("Apakah kamu berumur diatas 18 tahun?")
10
11     var hasil = pertanyaan ? "Selamat datang di website ini" : "Kamu tidak boleh
12     dmasuk di website ini";
13     document.write(hasil);
14   </script>
15 </body>
16 </html>
```

Hasilnya



Gambar 11.3 Pernyataan Operator Ternary

Jika di klik cancel



Gambar 11.4 Pernyataan Operator Ternary dipilih cancel
Dan jika di klik ok



Gambar 11.4 Pernyataan Operator Ternary dipilih OK

BAB 12

PERCABANGAN & PERULANGAN

TUJUAN :

- a. **Mahasiswa Mampu Memahami dan Mengimplementasikan Percabangan dengan JavaScript**
- b. **Mahasiswa Mampu Memahami dan Mengimplementasikan Perulangan dengan JavaScript**

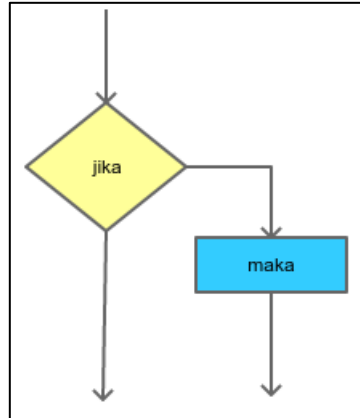
Overview :

Pada sub Bab ini akan di bahas percabangan dan perulangan. Percabangan adalah metode yang digunakan untuk menyampaikan pesan ketika kondisi target program ditemui. Jumlah kondisi mungkin satu, dua, atau lebih. Dengan menggunakan percabangan ini, para programmer dapat mengendalikan maupun mengontrol arah coding dengan mudah sesuai dengan logika.

Sedangkan perulangan merupakan Perulangan adalah jenis kontrol program yang berguna untuk memastikan skrip ditulis dengan benar, terutama untuk program yang menggunakan proses berulang-ulang. Frasa ini sering digambarkan sebagai pengulangan atau perulangan. Jadi dengan menggunakan perulangan, dapat menghemat space memori maupun memperingkas penulisan coding yang berulang – ulang.

12.1 PERCABANGAN

Apa itu percabangan dan mengapa disebut percabangan? Buat yang belum pernah belajar tentang algoritme dan flowchart, mungkin ini istilah yang baru pertama kali kamu dengar. Istilah ini sebenarnya untuk menggambarkan program alur yang bercabang. Pada flowchart, logika "jika...maka" ditampilkan dalam bentuk cabang.

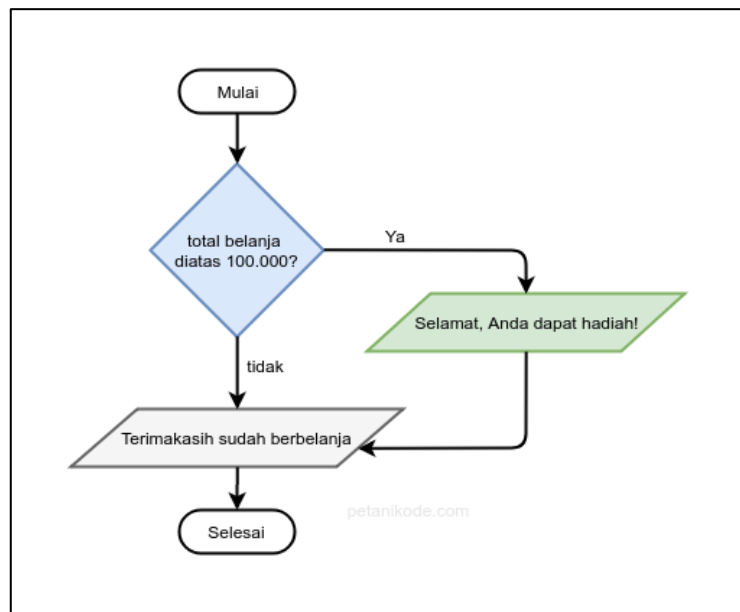


Gambar 12.1 Percabangan

Ini disebut sebagai cabang. Struktur ini selain di sebut sebagai percabangan juga disebut dengan Kontrol decision, flow, struktur kondisi, structure if, dan sebagainya. Percabangan akan dapat membuat perangkat lunak responsif dan memberikan data sesuai dengan kondisi yang kami tentukan. Ada enam jenis cabang dalam Javascript yang perlu dipahami.

a) Percabangan if

Merupakan jenis percabangan yang hanya memiliki satu blok pilihan bila keadaan sudah stabil. Pertimbangkan diagram alur berikut:



Gambar 12.1 Percabangan IF

Flowchart di bawah ini dapat dibaca seperti ini: "Jika seluruh pembelian Anda melebihi Rp 100.000, maka ucapkan "Selamat", dan Anda berhak mendapatkan hadiah." Bagaimana jika saya memiliki 100.000 rupiah?

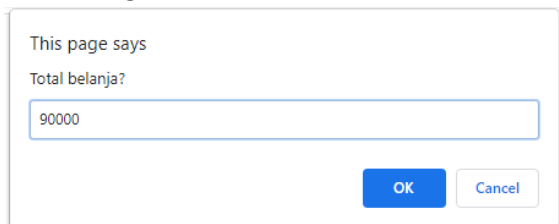
Sehingga pesan tidak tersampaikan.

Contoh dalam program Javascript:

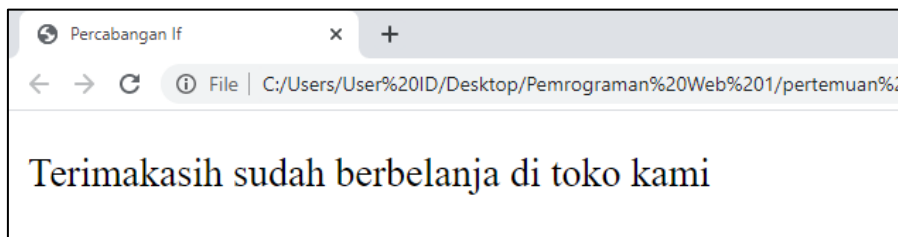
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Percabangan If</title>
5 </head>
6 <body>
7   <script>
8     var totalBelanja = prompt("Total belanja?", 0);
9
10    if(totalBelanja > 100000){
11      document.write("<h2>Selamat Anda dapat hadiah</h2>");
12    }
13
14    document.write("<p>Terimakasih sudah berbelanja di toko kami</p>");
15  </script>
16
17 </body>
18 </html>
```

Hasilnya :

Jika kurang dari 100000

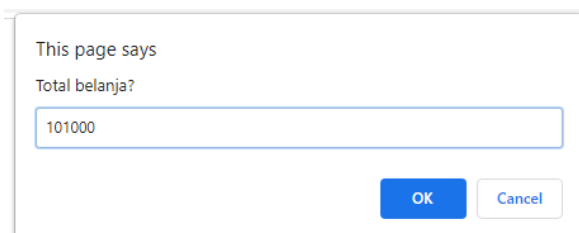


Gambar 12.3 Contoh 1 Input Percabangan IF



Gambar 12.4 Contoh 1 Output Percabangan IF

Jika lebih dari 100.000



Gambar 12.5 Contoh 2 Input Percabangan IF



Gambar 12.6 Contoh 2 Output Percabangan IF

Perhatikan :

```
if(totalBelanja > 100000){  
    document.write("<h2>Selamat Anda dapat hadiah</h2>");  
}
```

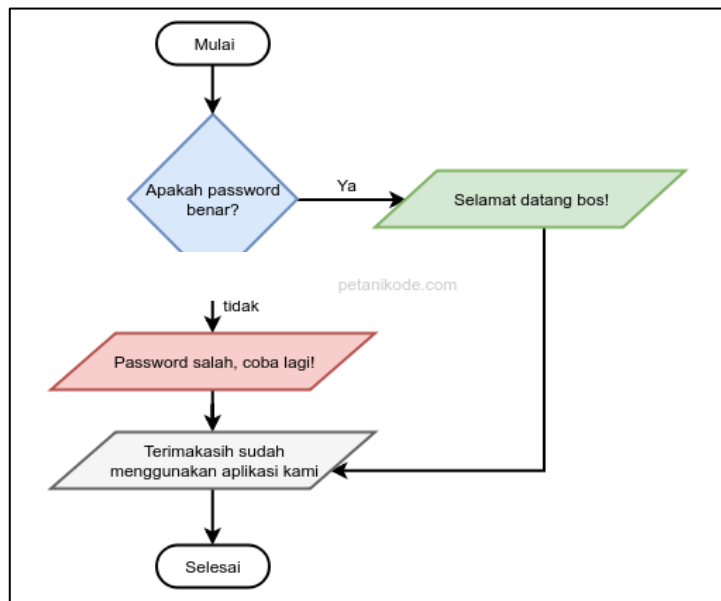
Ini disebut sebagai blok. Blok program ditulis dalam Javascript, yang diawali dengan kalimat “Buka Kurung Kurawal” dan diakhiri dengan kalimat “Tutup Kurung Kurawal”. Jika hanya ada satu ekspresi atau frase dalam satu blok, mungkin tidak perlu mencetak lengan kemeja.

```
if(totalBelanja > 100000)  
    document.write("<h2>Selamat Anda dapat hadiah</h2>");
```

Jika hanya ada satu baris ekspresi atau kalimat dalam satu blok, tanda kurung boleh tidak di tulis.

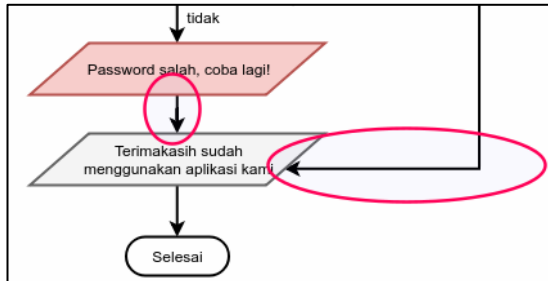
b) PERCABANGAN IF – ELSE

If/else percabangan adalah contoh percabangan dengan dua blok masukan. Pilihan pertama untuk kondisi yang parah, dan Pilihan kedua untuk kondisi yang ringan (selain itu). Perhatikan flowchart berikut:



Gambar 12.7 Percabangan IF – Else

Ini adalah diagram alur untuk pembuatan kata sandi. Jika kata sandi benar, pesan "Selamat Datang Bos!" akan ditampilkan pada blok pesan tersembunyi." Tetapi jika Anda login, Anda akan mendengar pesan berikut: "Password, salah, coba lagi!" Kemudian, karena itu bukan bagian dari percabangan blok if/else , pesan yang saat ini berada di blok abu-abu akan dilaporkan kembali. Perhatikan busur panah, setiap blok harus bergerak ke tengah.



Gambar 12. 8 Penegasan Percabangan If – else

Silahkan simak program dibawah ini !

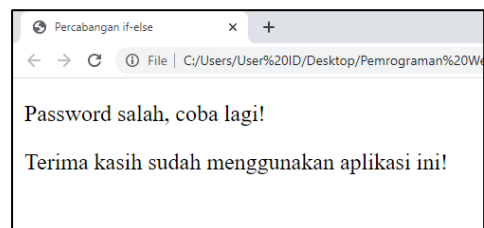
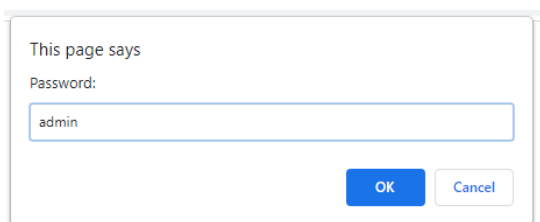
```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Percabangan if-else</title>
5 </head>
6 <body>
7   <script>
8     var password = prompt("Password:");
9
10    if(password == "stekom"){
11      document.write("<h2>Selamat datang bos!</h2>");
12    } else {
13      document.write("<p>Password salah, coba lagi!</p>");
14    }
15
16    document.write("<p>Terima kasih sudah menggunakan aplikasi ini!</p>");
17
18  </script>
19 </body>
20 </html>

```

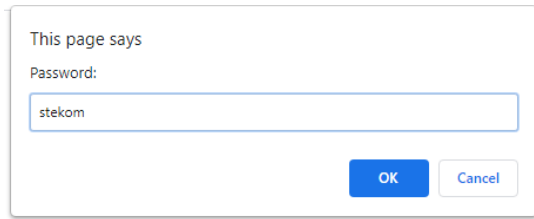
Hasilnya

Jika diisi password salah



Gambar 12.9 Contoh if-else dengan value salah

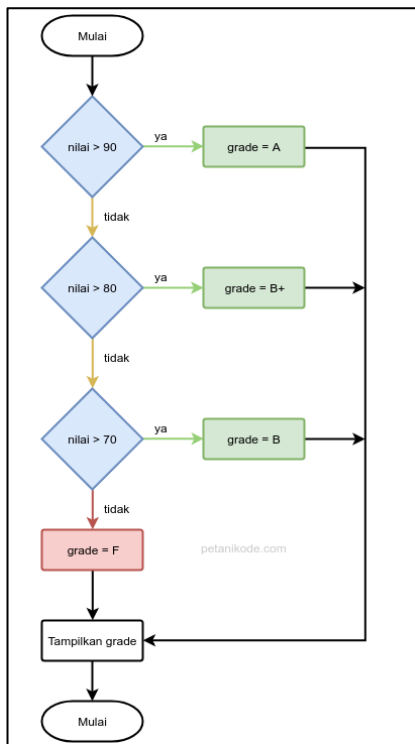
Jika diisi dengan password yang benar



Gambar 12.10 Contoh if-else dengan value benar

c) PERCABANGAN IF – ELSE – IF

If/else/if adalah jenis klausa yang memiliki lebih dari dua blok klausa. Lihat diagram alur di bawah ini:



Gambar 12.11 Percabangan If – Else - If

Perhatikan blok yang memiliki sinyal peringatan. Ini adalah blok untuk klausa if-else-if. Sejumlah blok yang Anda inginkan dapat ditambahkan.

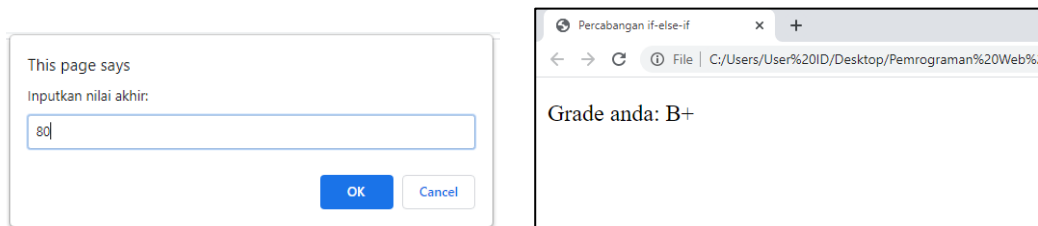
Contoh :

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Percabangan if-else-if</title>
5 </head>
6 <body>
7   <script>
8     var nilai = prompt("Inputkan nilai akhir:");
9     var grade = "";
10
11     if(nilai >= 90) grade = "A"
12     else if(nilai >= 80) grade = "B+"
13     else if(nilai >= 70) grade = "B"
14     else if(nilai >= 60) grade = "C+"
15     else if(nilai >= 50) grade = "C"
16     else if(nilai >= 40) grade = "D"
17     else if(nilai >= 30) grade = "E"
18     else grade = "F";
19
20     document.write(`<p>Grade anda: ${grade}</p>`);
21   </script>
22 </body>
23 </html>

```

Hasilnya :



Gambar 12.12 Contoh Percabangan If – Else – If

Karena hanya ada satu baris perintah dalam program di atas, kami tidak menggunakan kurung kurawal untuk membuat blok kode if/else/if. Yaitu, grade =

Jika kita menggunakan kurung kurawal, maka program diatas akan terlihat seperti ini :

```

<script>
  var nilai = prompt("Inputkan nilai akhir:");
  var grade = "";

  if (nilai >= 90){
    grade = "A"
  } else if(nilai >= 80) {
    grade = "B+"
  } else if(nilai >= 70) {
    grade = "B"
  } else if(nilai >= 60) {
    grade = "C+"
  } else if(nilai >= 50) {
    grade = "C"
  } else if(nilai >= 40) {
    grade = "D"
  } else if(nilai >= 30) {
    grade = "E"
  } else {
    grade = "F";
  }

```

```

}
document.write(`<p>Grade anda: ${grade}</p>`);
</script>

```

d) PERCABANGAN SWITCH – CASE

Klausula switch/case adalah jenis alternatif dari klausula if/else/if.. Strukturnya seperti ini:

```

switch(variabel){
  case <value>:
    // blok kode
    break;
  case <value>:
    // blok kode
    break;
  default:
    // blok kode
}

```

Kami dapat membuat kasus (case) yang diinginkan di kotak saklar. Pada <value>, Anda dapat membandingkan nilai saat ini dengan nilai masa depan variabel. Setiap kasus harus dimulai dengan jeda. Selama pihak yang gagal masih memimpin, tidak perlu membuka segel. Istirahat diperlukan agar program tetap fokus pada kasus berikutnya setelah kasus selesai.

Contoh:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Percabangan switch-case</title>
5 </head>
6 <body>
7   <script>
8     var jawab = prompt("Kamu beruntung! Silahkan pilih hadiahmu dengan
9     memasukan angka 1 sampai 5");
10    var hadiah = "";
11
12    switch(jawab){
13      case "1":
14        hadiah = "Tisu";
15        break;
16      case "2":
17        hadiah = "1 Kotak Kopi";
18        break;
19      case "3":
20        hadiah = "Sticker";
21        break;
22      case "4":
23        hadiah = "Minyak Goreng";
24        break;
25      case "5":
26        hadiah = "Uang Rp 50.000";
27
28        break;
29      default:
30        document.write("<p>Opps! anda salah pilih</p>");
31    }
32
33    if(hadiah === ""){
34      document.write("<p>Kamu gagal mendapat hadiah</p>");

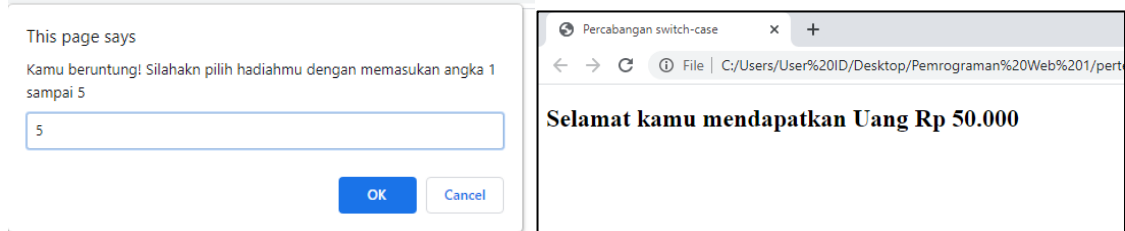
```

```

33     } else {
34         document.write("<h2>Selamat kamu mendapatkan " + hadiah + "</h2>");
35     }
36 </script>
37 </body>
38 </html>

```

Hasilnya :



Gambar 12.13 Contoh switch Case

Selain itu, sakelar atau kasing dapat dibuat dengan cara yang sama:

```

var nilai = prompt("input nilai");
var grade = "";

switch(true){
    case nilai < 90:
        grade = "A";
        break;
    case nilai < 80:
        grade = "B+";
        break;
    case nilai < 70:
        grade = "B";
        break;
    case nilai < 60:
        grade = "C+";
        break;
    case nilai < 50:
        grade = "C";
        break;
    case nilai < 40:
        grade = "D";
        break;
    case nilai < 30:
        grade = "E";
        break;
    default:
        grade = "F";
}

```

1. Berikan nilai sebenarnya ke sakelar sehingga program dapat memasuki blok sakelar.
 2. Kemudian, di dalam blok sakelar, buat kondisi dengan menggunakan casing.
- Hasilnya akan sama seperti dalam kasus pernyataan if/else/if.

e) PERCABANGAN DENGAN OPERATOR TERNARY

Percabangan if/else adalah bentuk lain dari percabangan yang menggunakan operator ternary. Bentuk singkatnya dari if/else bisa dibilang.

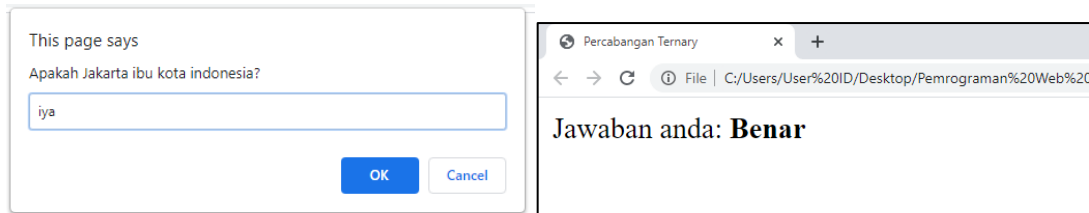
Contoh:

```

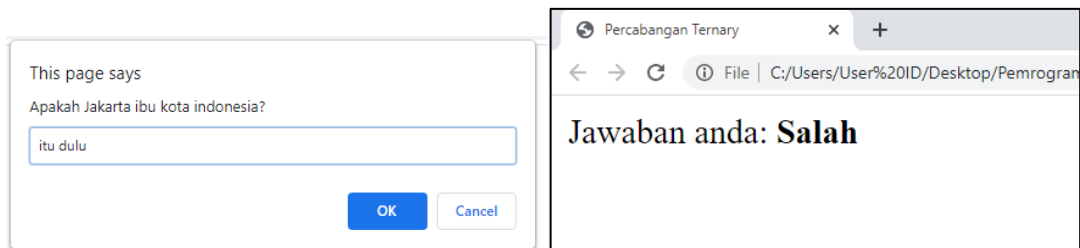
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Percabangan Ternary</title>
5 </head>
6 <body>
7   <script>
8     var jwb = prompt("Apakah Jakarta ibu kota indonesia?");
9
10    var jawaban = (jwb.toUpperCase() == "IYA") ? "Benar" : "Salah";
11
12    document.write(`Jawaban anda: <b>${jawaban}</b>`);
13  </script>
14 </body>
15 </html>

```

Fungsi toUpperCase() berfungsi untuk mengonversi teks yang dimasukkan menjadi huruf kapital semua.



Gambar 12.14 Contoh Operator Ternary Benar



Gambar 12.15 Contoh Operator Ternary Salah

Operator ternary berperan sebagai percabangan if/else:

```

var jawaban = (jwb.toUpperCase() == "IYA") ? "Benar" : "Salah";

```

Jika kondisi yang berada di kurung seperti—(jwb.toUpperCase() == "IYA")— memiliki nilai true, maka jawaban variable akan sama dengan 'Benar'.

Namun jika bernilai false, maka jawaban variable akan bernilai 'Salah'.

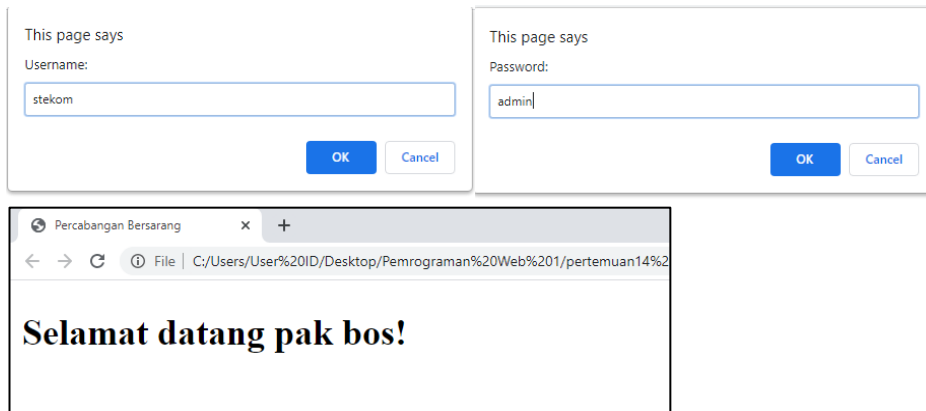
f) PERCABANGAN BERSARANG (NESTED)

Selain itu, di dalam setiap cabang cabang, Anda dapat membuat blok cabang cabang. Ini disebut sebagai nested if atau bersarang.

Contoh:

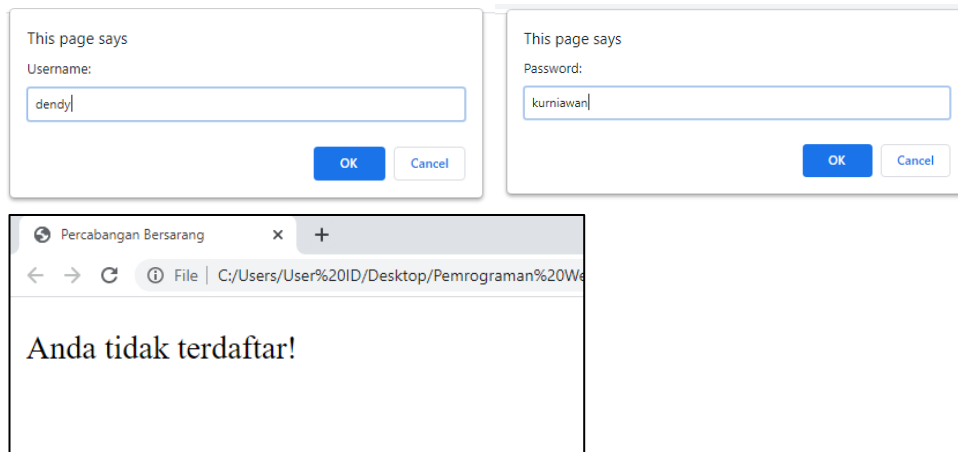

```
nested.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Percabangan Bersarang</title>
5 </head>
6 <body>
7   <script>
8     var username = prompt("Username:");
9     var password = prompt("Password:");
10
11     if(username == "stekom"){
12       if(password == "admin"){
13         document.write("<h2>Selamat datang pak bos!</h2>");
14       } else {
15         document.write("<p>Password salah, coba lagi!</p>");
16       }
17     } else {
18       document.write("<p>Anda tidak terdaftar!</p>");
19     }
20   </script>
21 </body>
22 </html>
```

Hasilnya :



Gambar 12.16 Contoh Operator Bersarang Benar

Jika salah



Gambar 12.17 Contoh Operator Bersarang Salah

Namun demikian dalam menyusun syntac seperti diatas sebenarnya ada cara lain (tidak menggunakan perulangan bersarang), namun menggunakan operator logika “&&” seperti :

```
if(username == "stekom" && password == "admin"){
    document.write("<h2>Selamat datang pak bos!</h2>");
} else {
    document.write("<p>Password salah, coba lagi!</p>");
}
```

12.2 PERULANGAN

Perulangan, atau kadang-kadang disebut sebagai "looping", adalah proses yang dilakukan secara diam-diam di latar belakang batas yang sudah dipahami. Menggunakan proses perulangan dapat membuat penulisan kode komputer menjadi lebih efektif. Jika di minta untuk mencetak tulisan secara berulang “Universitas Stekom” di website sebanyak 10 kali, mungkin ada yang akan membuat dengan perintah :

```
document.write("Universitas Stekom");
document.write("Universitas Stekom");
document.write("Universitas Stekom");
document.write("Universitas Stekom");
document.write("Universitas Stekom");
document.write("Universitas Stekom");
document.write("Universitas Stekom");
document.write("Universitas Stekom");
document.write("Universitas Stekom");
document.write("Universitas Stekom");
```

Memang tidak salah dengan menggunakan teknik seperti diatas, namun demikian hal tersebut akan membuat syntac menjadi panjang, apalagi jika disuruh mencetak sebanyak 1000 kali. Hal tersebut dapat diatasi dengan teknik Perulangan.

Tidak peduli seberapa banyak Anda tahu, perulangan akan membantu Anda menggunakan kode berulang-ulang. Ada tiga jenis ulang dalam Javascript. Biasanya, perulangan ini terbagi menjadi dua. Loop yang tidak dapat dihitung dan dapat dihitung.

Perbedaanya:

Counted Loop adalah modul perl yang jelas dan sudah memiliki banyak perulangannya.

Uncounted Loop adalah permutasi yang membuat tidak jelas seberapa sering pengguna harus mengulangnya.

Perulangan yang merupakan Counted Loop:

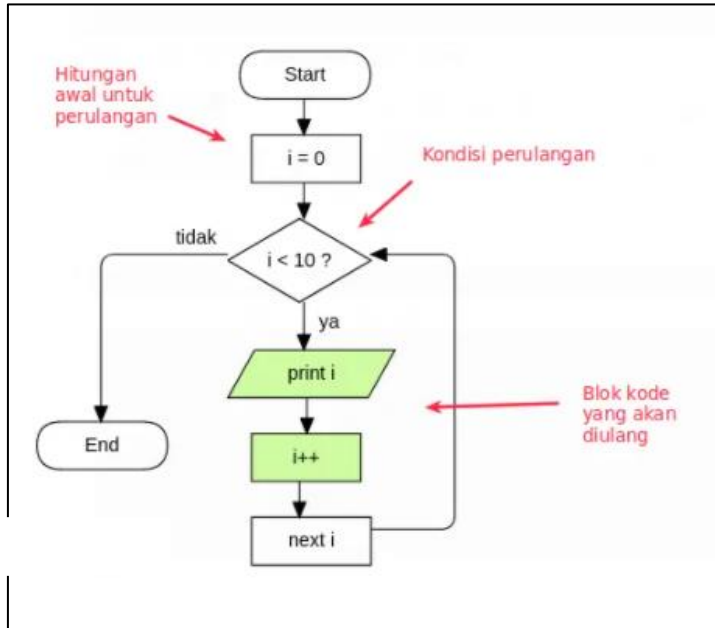
- Perulangan For
- Perulangan Foreach
- Perulangan Repeat

Perulangan yang merupakan Uncounted Loop:

- Perulangan While
- Perulangan Do/WHile

1) PERULANGAN FOR

Perulangan untuk adalah perulangan yang sering dijumpai dalam lingkaran tertutup karena sudah jelas seberapa sering akan terjadi.



Gambar 12.18 Perulangan

Contoh codingnya :

```
for(let i = 0; i < 10; i++){
    document.write("<p>Perulangan ke-" + i + "</p>")
}
```

Penting untuk mengetahui kondisi yang ada pada pakiaian setelah kata for.

Kondisi ini akan berarti:

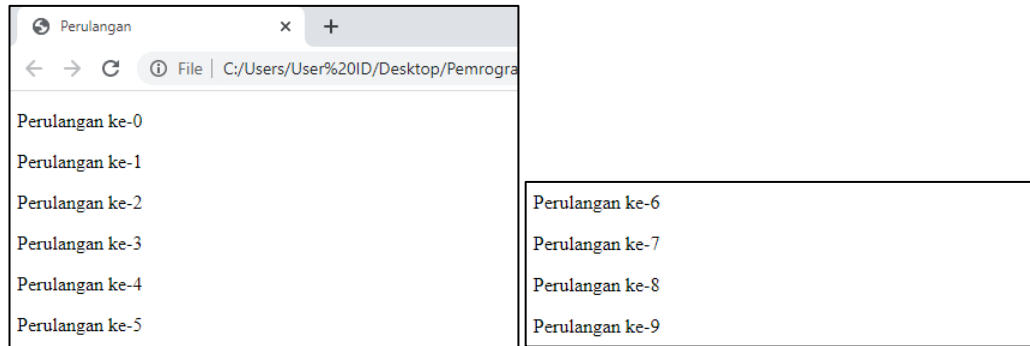
- Hitungan berawal dari angka 0 (i=0);
- Sampai angka berapa hitungannya? Sampai $i < 10$;
- Kemudian setiap perulangannya i akan ditambah dengan +1 (i++);

```
for(var nilaiAwal = 1; nilaiAwal <= 10; nilaiAwal++)
```

Fungsi variabel i dalam perulangan for adalah menghitung nilai hitungan.

Dan masing-masing memiliki perulangan. saya selalu diam. Karena kita sudah membahasnya di bagian i++.

hasil keluaran:



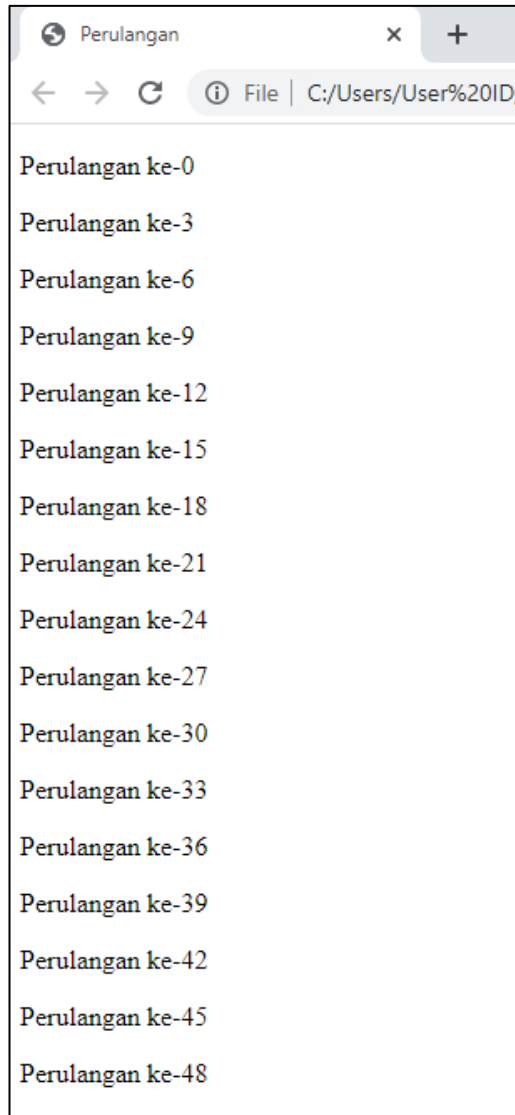
Gambar 12.18 Contoh Perulangan

Perlu diketahui juga nama variable tidak harus "i", anda bisa mengganti dengan nama lain, contoh :

```
for(counter = 0; counter < 50; counter+=3){  
    document.write("<p>Perulangan ke-"+counter+"</p>");  
}
```

Nama variable menggunakan "counter" dan syntac tersebut memulai perulangan dari nol lalu di setiap perulangan nilai variable counter akan di tambah 3 (counter+3).

Hasilnya :



Gambar 12. 19 Contoh Perulangan +3

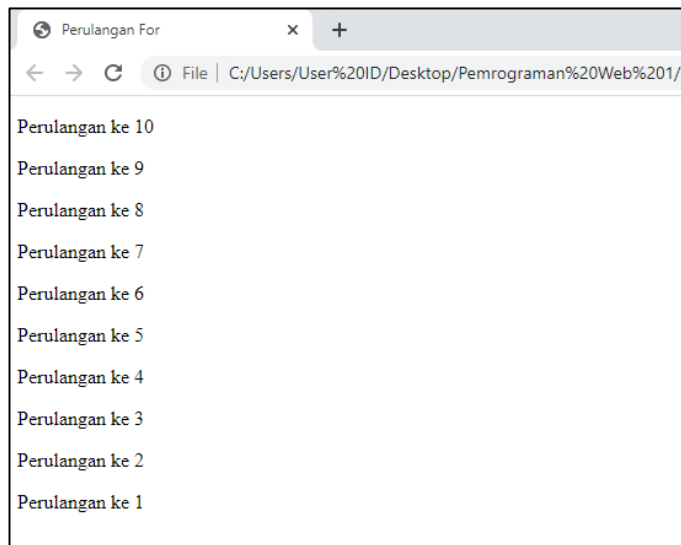
Apa jadinya jika counter perulangannya dari akanga terbesar ke terkecil? Hal ini biasanya dilakukan saat mencoba menghitung mundur. Caranya isi harga pembukaan loket dengan harga pasaran. Misalnya akan dimulai dari 10 dan turun menjadi 0. Pada counter, kita mulai menghitung dengan 10. Ketika kondisi perbandingan berubah, kita menghasilkan counter > 0 . Artinya perulangan akan dilakukan selama nilai counter lebih besar dari 0. Selanjutnya, di setiap perulangan, idi dengan cara mengubah pencacah menjadi nilai (-1) atau (counter--). Contoh :

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Perulangan For</title>
5 </head>
6 <body>
7   <script>
8     for(counter = 10; counter > 0; counter--){
9       document.write("<p>Perulangan ke "+counter+"</p>");
10    }
11  </script>
12 </body>
13 </html>

```

Hasilnya :



Gambar 12.20 Contoh Perulangan Decrement

Mengapa tidak berakhir pada nol (nol)? Menurut kondisi yang kita gunakan, pencacah adalah > 0. Jika pencacah disetel ke nol, asumsi akan salah. Namun, jika kita menggunakan operator yang lebih dekat hubungannya dengan (>=), maka syaratnya akan terpenuhi jika pencacahnya nol.

2) PERULANGAN WHILE

Perulangan while merupakan perulangan yang termasuk dalam perulangan uncounted loop. Perulangan sekaligus dapat menjadi perulangan yang terhitung loop dengan memberikan counter di dalamnya.



Gambar 12.21 Perulangan While

Misalnya

```
while(true) {  
    console.log('Hello World!');  
}
```

Syntac diatas akan mencetak hello word pada console sebanyak banyaknya tanpa henti (ini disebut sebagai invinite loop / perulangan tanpa batas). Bagaimana kita memberikan batas ? pertama dengan cara confirm user

```
var ulang = true;  
while(ulang) {  
    console.log('hello world');  
    ulang = confirm('lagi?');  
}
```

Gambar 12.22 Perulangan dengan Confirm

Cara kedua dengan menggunakan program

```
nilai awal  
while(kondisi terminasi) {  
  
    aksi  
  
    increment / decrement  
}
```

Gambar 12.23 Perulangan denganPrograman

Increment = Penambahan

Decrement = Pengurangan

```
var nilaiAwal = 1;  
while(nilaiAwal <= 5) {  
    console.log('hello world');  
    nilaiAwal++;  
}
```

Gambar 12.24 Perulangan dengan In/Decrement

Contoh program :

```
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4     <title>Perulangan While</title>  
5 </head>  
6 <body>  
7     <script>  
8         var ulang = confirm("Apakah anda mau mengulang?");  
9         var counter = 0;  
10  
11         while(ulang){
```

```
12     var jawab = confirm("Apakah anda mau mengulang?")
13     counter++;
14     if(jawab == false){
15         ulangi = false;
16     }
17 }
18
19     document.write("Perulangan sudah dilakuakn sebanyak "+ counter + "
20         kali");
21 </script>
22 </body>
23 </html>
```

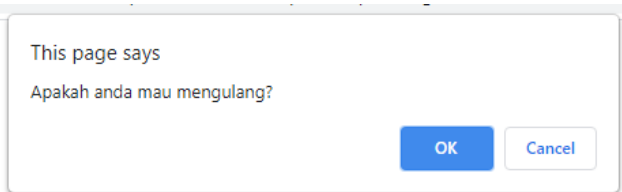
Atau bisa juga di tulis dengan

```
var ulangi = confirm("Apakah anda mau mengulang?");
var counter = 0;

while(ulangi){
    counter++;
    ulangi = confirm("Apakah anda mau mengulang?");
}

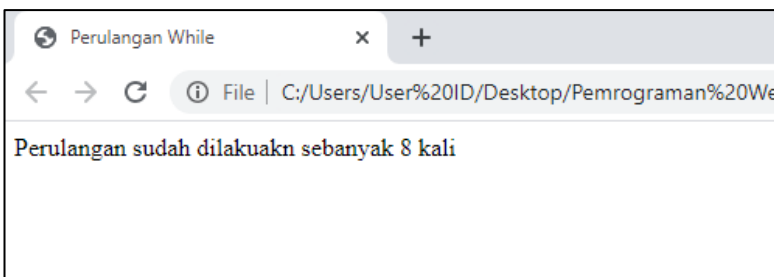
document.write("Perulangan sudah dilakuakn sebanyak "+ counter + "
    kali");
```

Hasilnya



Gambar 12.25 Contoh Program perulangan confirm

Jika Anda mengklik OK, pesan akan menampilkan confirm itu lagi sambil menghitung jumlah counter nya (berapa kali di klik OK), namun jika di klik cancel maka akan menampilkan hitungan perulangan



Gambar 12.26 Contoh Program perulangan confirm OK

Silahkan perhatikan :

```
while(ulangi){
    counter++;
    ulangi = confirm("Apakah anda mau mengulang?");
}
```

Selama variabelnya benar, akan ada bahaya. Kemudian kami menggunakan fungsi confirm() untuk menampilkan dialog konfirmasi. Selama kita memilih "Ok" di popup konfirmasi, nilai variabel

ulangi akan tetap benar. Namun, jika Anda memilih Cancel, nilai variabelnya akan salah. Ketika status ulang variabel salah, perulangan akan diamati.

3) PERULANGAN DO – WHILE

Mirip dengan perulangan sementara, perulangan dilakukan. Hal yang sama berlaku: Do/while akan melakukan perulangan sekali lagi sebelum memeriksa kondisi di dalam suit while.

Bentuk dengan cara ini:

```
do {  
    // blok kode yang akan diulang  
} while (<kondisi>;
```

Jadi perbedaannya :

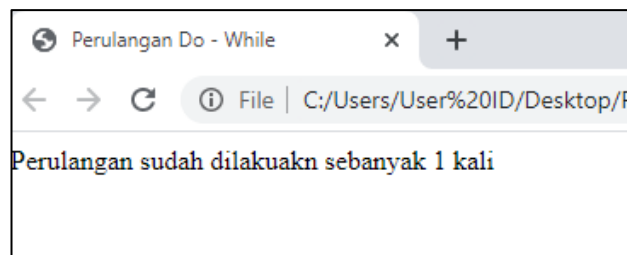
“Perulangan do/while akan mengecek kondisi di belakang (sesudah mengulang), sedangkan while akan mengecek kondisi di depan atau awal (sebelum mengulang)”.

Contoh :

```
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4   <title>Perulangan Do - While</title>  
5 </head>  
6 <body>  
7 <script>  
8   var ulangi = confirm("Apakah anda mau mengulang?");  
9   var counter = 0;  
10  
11   do {  
12     counter++;  
13     ulangi = confirm("Apakah anda mau mengulang?");  
14   } while(ulangi)  
15  
16   document.write("Perulangan sudah dilakuakn sebanyak "+ counter + " kali");  
17 </script>  
18 </body>  
19 </html>
```

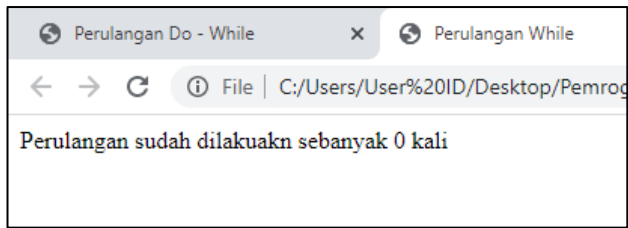
Hasil jika di dibandingkan dengan perulangan while jika sama – sama di klik cancel

Do while



Gambar 12.27 Perulangan do while

While



Gambar 12.28 Perulangan while

4) PERULANGAN FOREACH

Biasanya, perulangan foreach digunakan untuk memindai item dalam array. Loop ini adalah loop terhitung karena jumlah loop akan dikurangi dengan panjang dari array. Ada dua cara menggunakan foreach perulangan di Javascript:

- Gunakan "for" dengan operator "in"
- Gunakan "method forEach()"

Contoh:

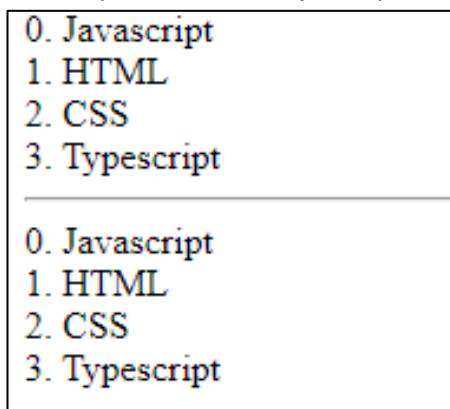
Berikut beberapa contoh ekspresi "foreach" tanpa operator "in":

```
var languages = ["Javascript", "HTML", "CSS", "Typescript"];
for(i = 0; i < languages.length; i++){
  document.write(i+" . "+ languages[i] + "<br/>");
}
```

Dengan menggunakan operator seperti ini, perulangan yang sekarang bisa dibuat lebih kuat:

```
var languages = ["Javascript", "HTML", "CSS", "Typescript"];
for(i in languages){
  document.write(i+" . "+ languages[i] + "<br/>");
}
```

Kedua syntac diatas menjadi seperti ini :



Gambar 12.29 Perulangan foreach

Pendekatan kedua untuk membuat perulangan masing-masing adalah dengan menggunakan fungsi forEach() dari sebuah array.

Contoh :

```
// kita punya array seperti berikut
var days = ["Senin", "Selasa", "Rabu", "Kamis", "Jum'at", "Sabtu", "Minggu"];

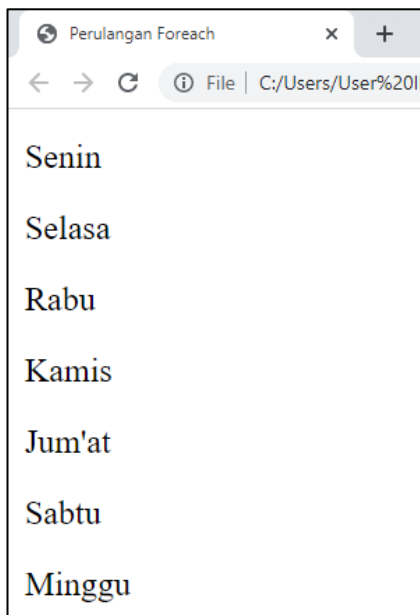
// Kemudian kita tampilkan semua hari
// dengan menggunakan method foreach
days.forEach(function(day){
    document.write("<p>" + day + "</p>");
});
```

Parameter fungsi callback dari metode forEach(). Terbukti, Anda dapat menggunakan fungsi panah seperti ini:

```
// kita punya array seperti berikut
var days = ["Senin", "Selasa", "Rabu", "Kamis", "Jum'at", "Sabtu", "Minggu"];

// Kemudian kita tampilkan semua hari
// dengan menggunakan method foreach
days.forEach((day) => {
    document.write("<p>" + day + "</p>");
});
```

Hasilnya



Gambar 12.30 contoh perulangan foreach

5) PERULANGAN DENGAN METODE REPEAT()

Perulangan dengan metode atau fungsi repeat() termasuk dalam loop penghitungan perulangan. Fungsi ini khusus digunakan untuk mengedit satu tulisan (string). Bisa dikatakan: Ini adalah single dari perulangan for.

Contoh:

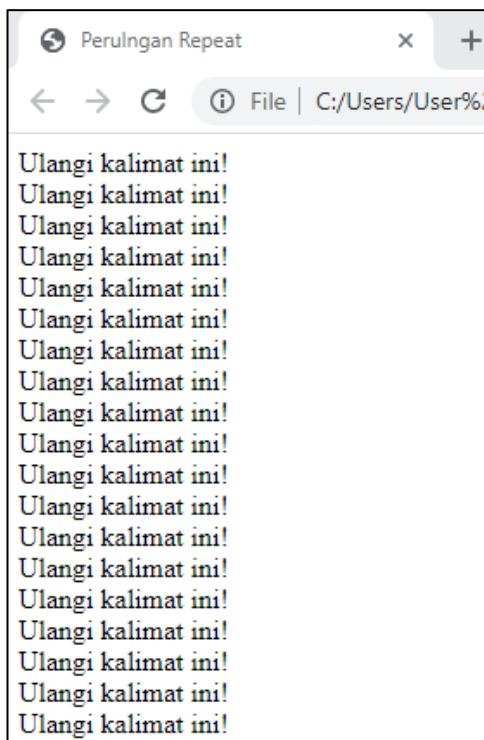
Dalam hal kami mempekerjakan perulangan untuk:

```
for( let i = 0; i < 100; i++){  
    document.write("Ulangi kalimat ini!");  
}
```

Jika gunakan metode repeat maka akan seperti

```
document.write("Ulangi kalimat ini! <br>".repeat(100));
```

Hasilnya



Gambar 12.31 Perulangan repeat

PAHAM !

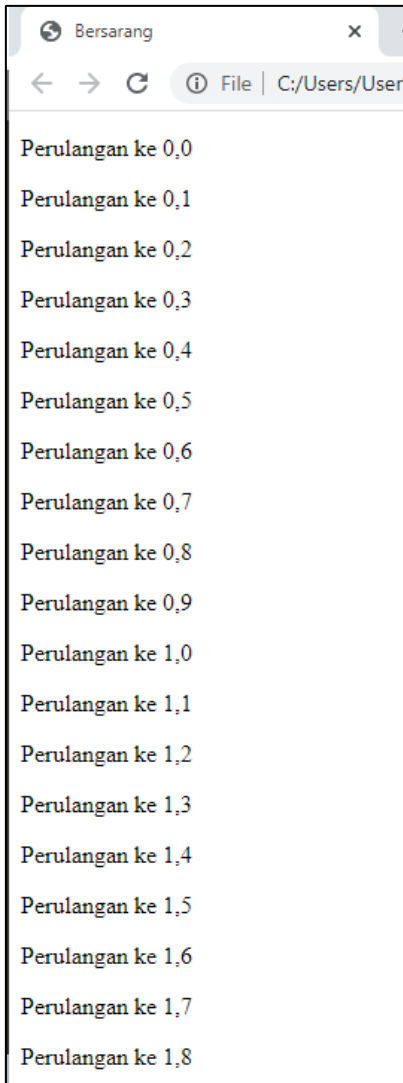
Contoh program perulangan lain (Perulangan Bersarang)

Kita bisa membuat perulangan di blok perulangan. Ini disebut sebagai loop bersarang, loop sarangi, atau loop di dalam loop.

Mari, silakan baca yang berikut ini:

```
for(let i = 0; i < 10; i++){  
    for(let j = 0; j < 10; j++){  
        document.write("<p>Perulangan ke " + i + ", " + j + "</p>");  
    }  
}
```

Hasilnya



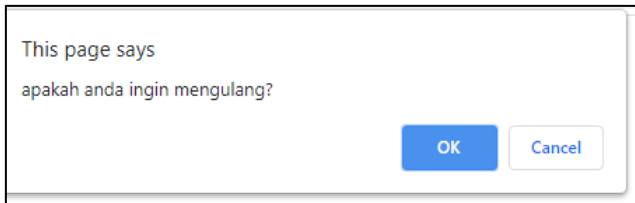
Gambar 12.32 contoh 2 perulangan bersarang

Dalam perulangan tersebut di atas, kami menggunakan dua perulangan untuk. Perulangan pertama menggunakan Variabel I sebagai Counter, sedangkan Perulangan kedua menggunakan Variabel J sebagai Counter.

Dengan kata lain:

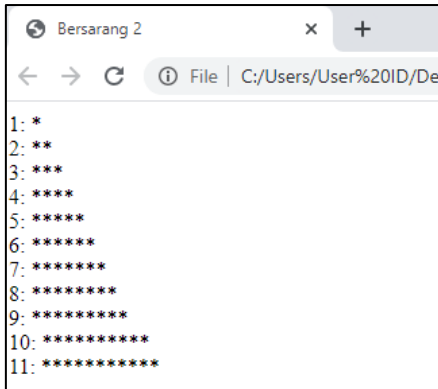
```
7 <script>
8     var ulang = confirm("apakah anda ingin mengulang?");
9     var counter = 0;
10
11     while (ulang) {
12         counter++;
13         var bintang = "*".repeat(counter) + "<br>";
14         document.write(counter + ": " + bintang);
15         ulang = confirm("apakah anda ingin mengulang?");
16     }
17 </script>
```

Hasilnya :



Gambar 12.33 contoh 3 pernyataan perulangan bersarang

Seberapa kali anda melakukan perulangan dengan klik OK, setelah anda klik cancel maka akan di tampilkan jumlah perulangan namun berupa tanda bintang



Gambar 12.34 contoh 3 hasil perulangan bersarang

Catatn : “Bila perulangan lebih banyak, maka komputer akan lebih lama memprosesnya. Tentu ini akan membuat website atau aplikasi kita berjalan lambat. Untuk itu, gunakan perulangan dengan bijak”.

BAB 13

STRUKTUR DATA ARRAY & DOM

TUJUAN :

- a. Mahasiswa Mampu Memahami dan Mengimplementasikan Struktur Data Array di JavaScript
- b. Mahasiswa Mampu Memahami dan Mengimplementasikan Program JavaScript dengan Menggunakan DOM

Overview :

Menyimpan beberapa value dalam satu variable bisa di lakukan dengan teknik data Array. Pada sub bab ini nanti akan di bahas cara menyimpan value lebih dari satu ke dalam satu variable. Serta juga akan di bahas cara menampilkan data urut berdasarkan index dan mengupdate atau menghapus value pada index ke sekian.

Pada paragraf ini juga akan dibahas DOM (Document Object Model). Tugas pertama dan terpenting Javascript di web adalah membuat header situs web yang dinamis. Ini dapat dilakukan dengan Javascript menggunakan DOM. Singkatan DOM adalah singkatan dari Document Object Model. Dengan demikian, dokumen HTML yang dimodelkan setelah objek.

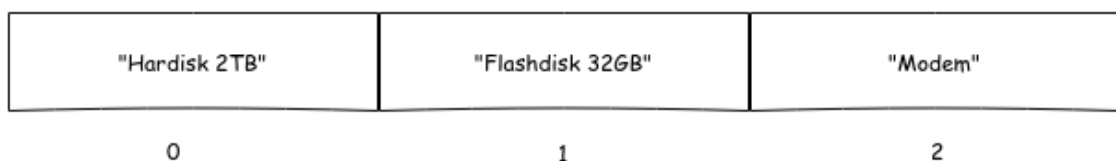
13.1 STRUKTUR DATA ARRAY

Metode atau proses yang digunakan untuk menyimpan data dalam memori komputer disebut struktur data. Satu-satunya struktur data yang sering digunakan dalam program adalah array. Saat kita membuat aplikasi online yang akan menampilkan banyak data seperti contoh di bawah ini

```
var produk1 = "Modem";  
var produk2 = "Hardisk";  
var produk3 = "Flashdisk";  
  
document.write(`${produk1}<br>`);  
document.write(`${produk2}<br>`);  
document.write(`${produk3}<br>`);
```

Cara menampilkan data dengan teknis diatas boleh saja dilakukan, namun cara tersebut kurang efektif. Misalkan nanti ada 100 data atau produk yang akan di tampilkan maka kita harus membuat variable sebanyak 100x dan juga mencetak document.write sebanyak 100x. ada cara lain agar penulisan coding lebih sederhana, yaitu dengan menggunakan array.

Struktur data yang dikenal sebagai array digunakan untuk mengelompokkan semua data di satu lokasi. Setiap elemen data dalam Array memiliki indeks, sehingga memudahkan kita untuk memahaminya.



Gambar 13.1 Ilustrasi Data Array

Indeks array terus berkembang dari angka nol (0). Batas atas array akan didorong oleh banyaknya data yang disertakan di dalamnya.

a) Cara Membuat Array pada Javascript

Dalam pemrograman javascript, array ini dapat dibuat dengan symbol kurung siku [...]

Misal:

```
var products = [];
```

Oleh karena itu berbagai item akan memiliki bermacam-macam pilihan. Setelah memasukkan data ke dalam array, setiap titik data kemudian direpresentasikan dengan tanda koma (,).

Contoh:

```
var products = ["Flashdisk", "SDD", "Monitor"];
```

Javascript memiliki sifat dynamic typing yang memungkinkan kita dapat menampung atau mencampur type data apapun di dalam array.

Contoh :

```
var myData = [12, 2.1, true, 'C', "Stekom"];
```


b) Cara Mengambil Data dari Array

Array akan menambahkan semua data dan memberinya nomor indeks sehingga mudah diakses. Indeks array secara konsisten menyusut dari node 0. Pertimbangkan bahwa kita memiliki array yang mirip dengan ini:

```
var makanan = ["Nasi Goreng", "Mie Ayam", "Mie Gelas"];
```

Bagaimana cara kita mengambil nilai "Mie Ayam"?

Jawabannya seperti ini:

```
makanan[1]
```

Perlu diingat index dimulai dari angka 0.

Contoh :

```
array1.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Array</title>
5 </head>
6 <body>
7 <script>
8   // membuat array
9   var products = ["Senter", "Radio", "Antena", "Obeng"];
10
11  // mengambil radio
12  document.write(products[1]);
13 </script>
14 </body>
15 </html>
```

Hasilnya



Gambar 13.2 Contoh Menampilkan Index Array

Mencetak isi Array dengan perulangan

Disaat kita akan menampilkan seluruh isi data yang ada di array, bisa menggunakan teknik seperti

```
document.write(products[0]);
document.write(products[1]);
document.write(products[2]);
document.write(products[3]);
document.write(products[4]);
```

Namun demikian, jika data yang akan di tampilkan jumlahnya 100 data maka syntak akan menjadi panjang sekali. Maka dari itu dapat menggunakan teknik seperti contoh

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Array dan perulangan</title>
5 </head>
6 <body>
7   <script>
8     // membuat array
9     var products = ["Senter", "Radio", "Antena", "TV"];
10
11     document.write("<h3>Daftar Produk:</h3>");
12     document.write("<ol>");
13     // menggunakan perulangan untuk mencetak semua isi array
14     for(let i = 0; i < products.length; i++){
15       document.write("<li>${ products[i] }</li>");
16     }
17     document.write("</ol>");
18   </script>
19 </body>
20 </html>
```

Hasilnya



Gambar 13.3 Contoh menampilkan data array dengan perulangan for

Perhatian!

Gunakan properti length untuk membuat array panjang pada contoh di atas. Ada 4 data dalam produk array, sehingga panjang properti menjadi 4. Kemudian gunakan properti ini untuk mengurangi jumlah pengulangan di lapangan.

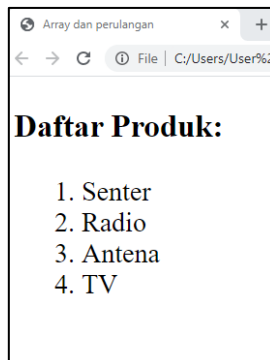
```
for(let i = 0; i < products.length; i++){
  document.write("<li>${ products[i] }</li>");
}
```

Kemudian, di dalam blok untuk, kami mengevaluasi produk menggunakan indeks yang sesuai dengan variabel i.

Contoh cara lain menampilkan data array dengan forEach

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Array dan perulangan</title>
5 </head>
6 <body>
7   <script>
8     // membuat array
9     var products = ["Senter", "Radio", "Antena", "TV"];
10
11     document.write("<h3>Daftar Produk:</h3>");
12     document.write("<ol>");
13     // menggunakan perulangan untuk mencetak semua isi array
14     products.forEach((data) => {
15       document.write("<li>${data}</li>");
16     });
17     document.write("</ol>");
18   </script>
19 </body>
20 </html>
```

Hasilnya :



Gambar 13.4 Contoh menampilkan data array dengan perulangan foreach

c) Cara Menambahkan Data ke Dalam Array

Ada dua metode yang dapat digunakan untuk menambahkan data ke array:

- Mengisi menggunakan indeks,
- Mengisi menggunakan metode push ().

Contoh Mengisi dengan database

Sebagai contoh, ini adalah array yang kita miliki:

```
var buah = ["Apel", "Jeruk", "Manggis"];
```

Ada tiga data dalam array dengan indeks:

0 = Apel,

1 = Jeruk,

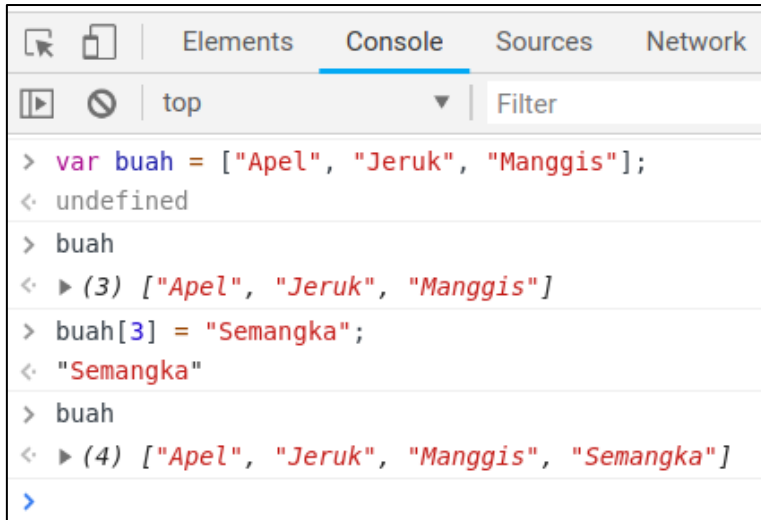
2 = Manggis.

Jika Anda ingin menambahkan lebih banyak data ke indeks ke-3, Anda dapat melakukannya dengan menggunakan sintaks berikut:

```
buah[3] = "Semangka";
```

Sekarang array buah akan berisi 4 data.

Contoh dalam program :



```
> var buah = ["Apel", "Jeruk", "Manggis"];
< undefined
> buah
< ▶ (3) ["Apel", "Jeruk", "Manggis"]
> buah[3] = "Semangka";
< "Semangka"
> buah
< ▶ (4) ["Apel", "Jeruk", "Manggis", "Semangka"]
>
```

Gambar 13.5 menambahkan data array

"Semangka" berhasil dimasukkan ke dalam array buah. Satu-satunya kelemahan metode ini adalah Anda perlu mengetahui jumlah titik data atau ukuran array sebelum Anda dapat menambahkan lebih banyak.

Jika kita memasukkan angka yang benar untuk indeks sembarangan, maka data yang ada di indeks tersebut akan dihapus.



```
> buah
< ▶ (4) ["apel", "jeruk", "manggis", "semangka"]
> buah[2] = "lemon";
< "lemon"
> buah
< ▶ (4) ["apel", "jeruk", "Lemon", "semangka"]
>
```

Gambar 13.6 mengupdate data array

Terdapat teknik lain agar programmer tidak perlu mengetahui berapa jumlah data didalam array nya. Disini dapat menggunakan method push(). Method push() ini akan menambahkan data kedalam ekor array atau pada bagian belakang sendiri.

Contoh :

```

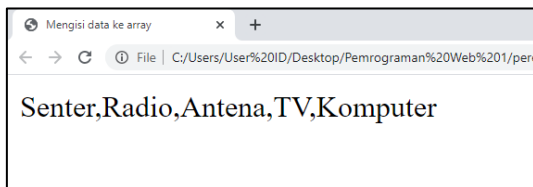
<script>
  // membuat array
  var products = ["Senter", "Radio", "Antena", "TV"];

  // menambahkan tv ke dalam array products
  products.push("Komputer");

  // menampilkan isi array
  document.write(products);

```

Hasilnya



Gambar 13.7 menambahkan data array paling belakang

Kita juga bisa menambahkan beberapa data sekaligus dengan cara seperti ini:

```

products.push("Alarm", "Frezer", "Majikom");

```

d) Cara Menghapus Array

Mirip dengan menambahkan data ke array, penghapusan data dapat dilakukan dengan salah satu dari dua cara.

- Gunakan delete;
- Gunakan method pop ().

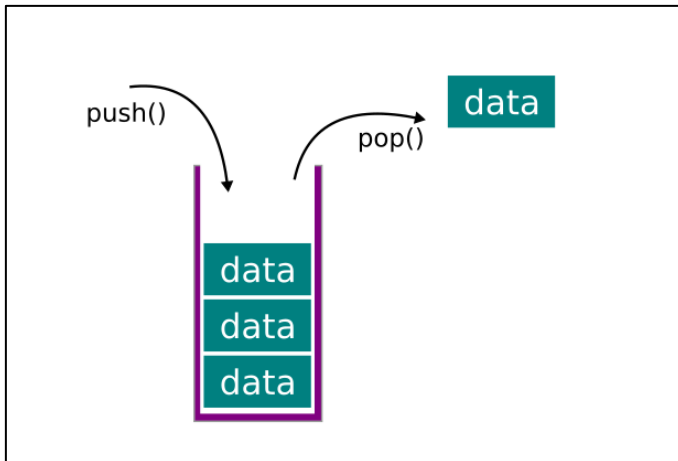
Contoh menggunakan delete



Gambar 13.8 menghapus data array

Dengan tombol hapus, Anda dapat menghapus nomor indeks tertentu dari data. Sebaliknya, `pop()` akan mengembalikan data dari belakang. Jika terjadi penghapusan, dia akan memasukkan ruang kosong ke dalam array. Akibatnya, sangat tidak mungkin Akerna Array akan terus memiliki Panjang 4.

Metode `pop()` digunakan dalam pemanggilan metode kedua untuk membalik metode `push()`. Array yang ada di bagian bawah metode `pop()` akan dihancurkan. Dalam Javascript, array dapat digunakan sebagai tumpukan (tumpukan) yang memiliki simbol LIFO (Last in Last out).



Gambar 13.9 ilustrasi menambahkan dan mengurangi data array
Contoh

```

> delete buah[2];
↳ true
> buah
↳ ▶ (4) ["apel", "jeruk", empty, "semangka"]
> buah.pop();
↳ "semangka"
> buah.pop();
↳ undefined
> buah.pop();
↳ "jeruk"
> buah
↳ ▶ ["apel"]
> |

```

Gambar 13.10 contoh menambahkan dan menghapus data array

Jika kita menggunakan fungsi pop() lebih dari tiga kali, array hanya akan menyertakan data [apel]. Data awal dalam array adalah empat. Metode pop() akan menghapus nilai yang tidak valid atau hilang dari array.

e) Menghapus Data dari Depan

Dengan menggunakan metode shift, Anda juga bisa mendapatkan data dari masa lalu ().

Contoh:

```

var bunga = ["Mawar", "Melati", "Anggrek", "Sakura"];

// hapus data dari depan
bunga.shift();

```

Maka data yang terhapus adalah "Mawar".

f) Menghapus Data pada Indeks Tertentu

Jika Anda ingin memasukkan data ke dalam tinta saat ini, Anda harus menggunakan fungsi atau metode penyambungan (). Ada dua parameter yang diperlukan untuk fungsi ini:

```
array.splice(<indeks>, <total>);
```

Keterangan:

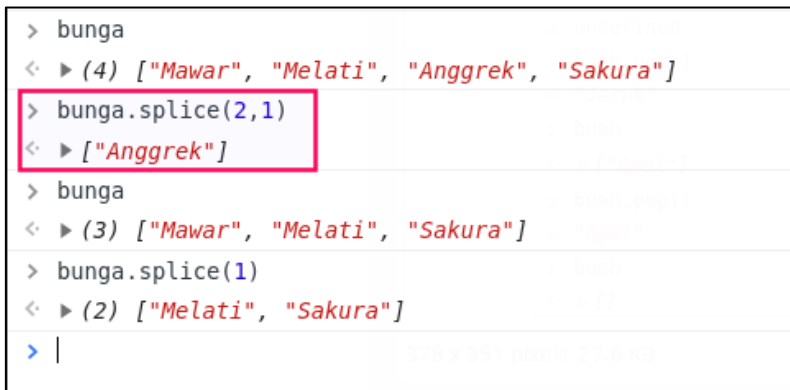
- <indeks> adalah indeks dari data dalam array yang akan dimusnahkan;
- <total> adalah jumlah data yang akan dikumpulkan dari indeks tersebut di atas.

Untuk hanya memproses satu bagian data pada satu waktu, merupakan praktik standar untuk mengganti seluruh angka dengan 1.

Contoh:

```
var bunga = ["Mawar", "Melati", "Anggrek", "Sakura"];  
  
// hapus Anggrek  
bunga.splice(2, 1);
```

Pada console :



```
> bunga  
< ▶ (4) ["Mawar", "Melati", "Anggrek", "Sakura"]  
> bunga.splice(2,1)  
< ▶ ["Anggrek"]  
> bunga  
< ▶ (3) ["Mawar", "Melati", "Sakura"]  
> bunga.splice(1)  
< ▶ (2) ["Melati", "Sakura"]  
> |
```

Gambar 13.11 menghapus data array paling depan

Pada contoh di atas, jika kita tidak menentukan data "<total>" yang akan hilang, maka semua data dari indeks "pilih" juga akan hilang.

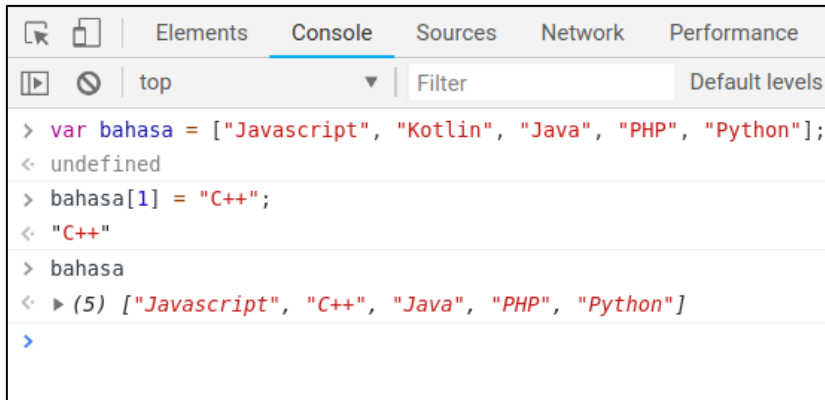
g) Mengubah isi Array

Mengubah isi array dapat dilakukan dengan cara berikut

```
var bahasa = ["Javascript", "Kotlin", "Java", "PHP", "Python"];  
bahasa[1] = "C++";
```

Maka "Kotlin" akan diganti dengan "C++".

Percobaan pada console:



```
> var bahasa = ["Javascript", "Kotlin", "Java", "PHP", "Python"];
< undefined
> bahasa[1] = "C++";
< "C++"
> bahasa
< ▶ (5) ["Javascript", "C++", "Java", "PHP", "Python"]
>
```

Gambar 13.12 contoh mengganti data array

h) Method-method Array

Selain metode atau fungsi yang sudah kita bahas, ada beberapa metode dalam array yang perlu kita pahami.

1) Method filter()

Fungsi metode filter() adalah untuk mengekstrak data dari array. Parameter wajib untuk metode filter() adalah fungsi panggilan balik, sama seperti untuk metode forEach().

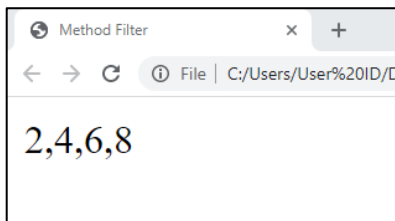
Contoh:

```
<script>
  const angka = [1, 2, 3, 4, 5, 6, 7, 8, 9];

  // Kita ambil data yang hanya habis dibagi dua saja
  const hasil = angka.filter((item) => {return item % 2 === 0});

  document.write(hasil);
</script>
```

Hasilnya :



Gambar 13.13 method filter array

Pada contoh di atas, kami menyediakan fungsi panah sebagai fungsi callback yang akan melakukan pensinyalan terkait array. Karena itu, Anda dapat membuat sesuatu yang jauh lebih rumit dari ini:

```
const hasil = angka.filter(item => item % 2 === 0);
```

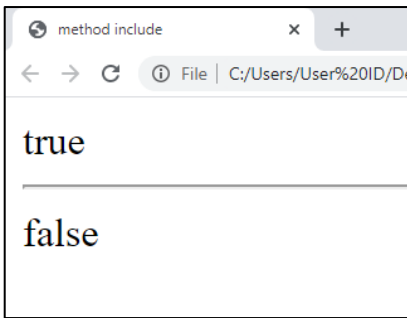
2) Method includes()

Tujuan dari metode ini adalah untuk menentukan apakah sepotong data ada dalam array atau tidak. Biasanya digunakan untuk melakukan pencarian untuk memastikan data sudah ada di dalam array.

Contoh:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>method include</title>
5 </head>
6 <body>
7 <script>
8   var tanaman = ["Padi", "Kacang", "Jagung", "Kedelai"];
9
10  // apakah kacang sudah ada di dalam array tanaman?
11  var adaKacang = tanaman.includes("Kacang");
12
13  document.write(adaKacang);
14
15  document.write("<hr>");
16
17  // apakah bayam ada?
18  var adaBayam = tanaman.includes("Bayam");
19
20  document.write(adaBayam);
21 </script>
22 </body>
23 </html>
```

Hasilnya :



Gambar 13.14 method includes array

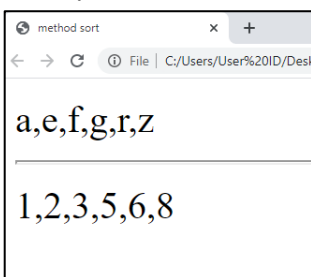
3) Method sort()

Method sort() Berfungsi untuk memasukkan data ke dalam array.

Contoh:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>method sort</title>
5 </head>
6 <body>
7 <script>
8   var alfabet = ['a','f','z','e','r','g'];
9   var angka = [3,1,2,6,8,5];
10
11   document.write(alfabet.sort());
12   document.write("<hr>");
13   document.write(angka.sort());
14 </script>
15 </body>
16 </html>
```

Hasilnya



Gambar 13.15 method sort array

13.2 DOM (Document Object Model)

Belajar JavaScript pada intinya terdiri dari dua bidang utama, yaitu JavaScript Inti (juga dikenal sebagai Core JavaScript) dan API yang disediakan oleh web browser (yang dikenal dengan istilah DOM, singkatan dari Document Object Model).

"Bahasa Pemograman JavaScript" adalah nama resmi untuk Core JavaScript, juga dikenal sebagai JavaScript inti. Pada bagian Core JavaScript inilah kita akan belajar tentang aturan

pemrograman yang umumnya dipelajari seperti cara mendefinisikan variabel, perbedaan tipe data, cara membuat array, cara menulis struktur IF, dan cara membuat objek.

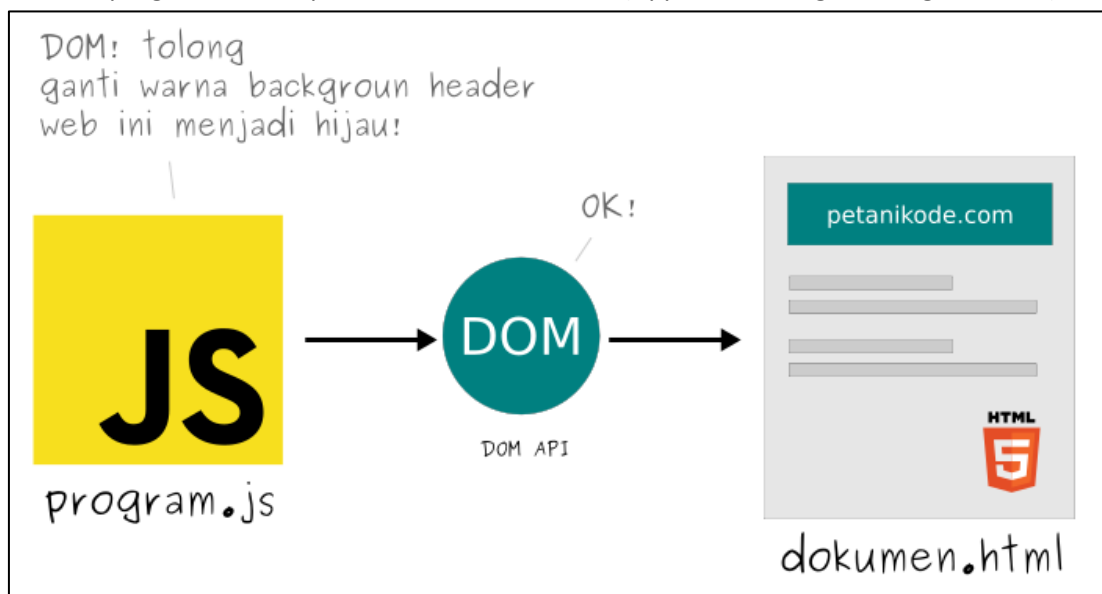
Bab Inti JavaScript membahas tata bahasa atau "bahasa" JavaScript. Tidak akan terlalu sulit untuk memahami aturan penyisipan JavaScript jika Anda pernah menggunakan bahasa pemrograman seperti C++ atau PHP. Perbedaan JavaScript dengan bahasa pemrograman lainnya adalah JavaScript lebih fokus pada objek. Namun, JavaScript "tidak mengharuskan" menggunakan objek. Ucapkan "tidak harus" karena Anda dapat membuat beberapa program menggunakan kode yang berfungsi sempurna tanpa menggunakan objek. Berikut ini adalah fungsi peringatan.

Core JavaScript adalah implementasi lain dari JavaScript yang tidak kompatibel dengan "lingkungannya"—browser web. Sampai saat ini, JavaScript hanya dikenal sebagai bahasa skrip sisi klien yang digunakan di browser web. Namun demikian, perkembangan JavaScript saat ini menghambat penggunaannya di server (salah satunya dengan menggunakan node.js).

a) PENGERTIAN DOM

Tugas pertama dan terpenting Javascript di web adalah membuat header situs web yang dinamis. Ini dapat dilakukan dengan Javascript menggunakan DOM. Singkatan DOM adalah singkatan dari Document Object Model. Dengan demikian, dokumen HTML yang dimodelkan setelah objek.

Objek dari dokumen ini menawarkan berbagai fungsi dan atribut/data yang dapat kita gunakan untuk membuat program Javascript. Disebutkan bahwa API (Application Programming Interface).



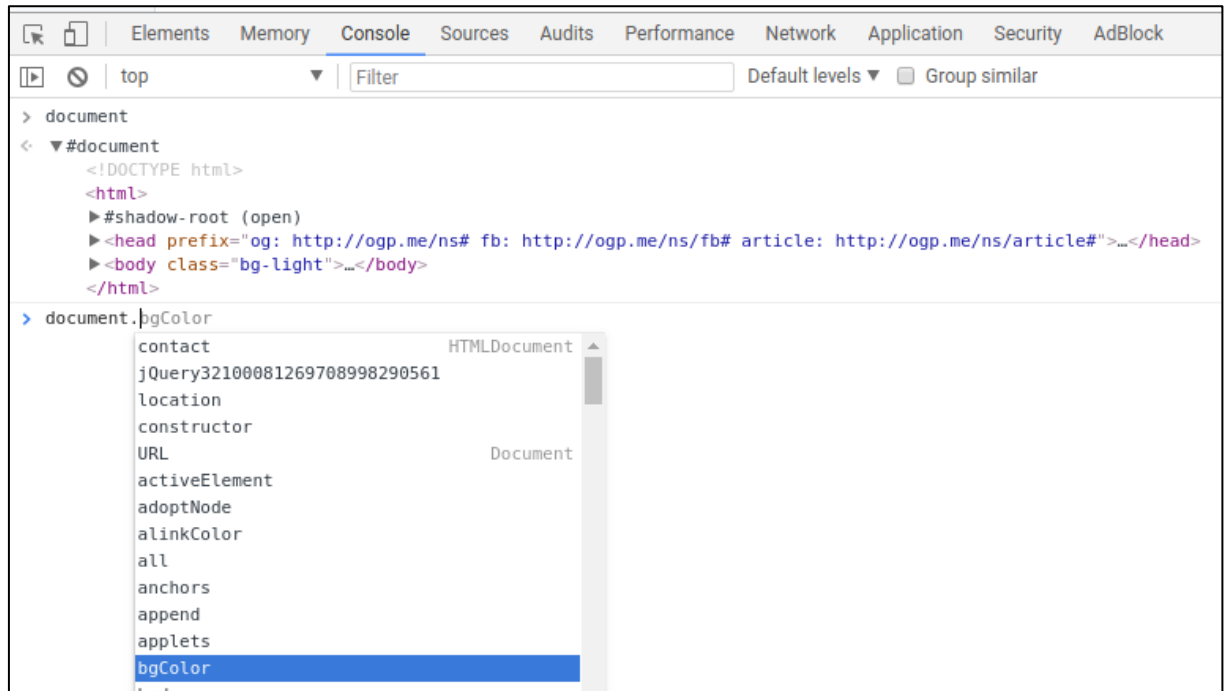
Gambar 13.16 Ilustrasi DOM (sumber : petanikode.com)

- DOM bukan hanya untuk dokumen HTML. Selain itu, DOM dapat digunakan untuk membuat dokumen dalam format XML dan SVG.
- DOM tidak hanya didukung oleh Javascript; itu juga didukung oleh bahasa pemrograman lain.

1) Bagaimana Menggunakan DOM ?

Seperti yang sudah kita ketahui, tujuan dari DOM adalah untuk memodelkan dokumen HTML. Dokumen adalah nama objek DOM Javascript. Objek ini berisi semua informasi yang

diperlukan untuk membuat HTML. Jika kita memilih untuk membuka dokumen dalam Javascript di konsol, maka kode HTML akan ditampilkan.



Pada dokumen objek terdapat atribut dan fungsi yang dapat digunakan dalam mengelola dokumen HTML. Seperti misalnya fungsi dokumen, tulis ().

Fungsi ini dapat digunakan dalam menambahkan konten ke dokumen HTML. Misalnya, Anda dapat mengetikkan kode berikut ke editor Javascript atau editor teks.

```
document.write("Hello World");  
document.write("<h2>Saya Sedang Belajar Javascript</h2>");
```

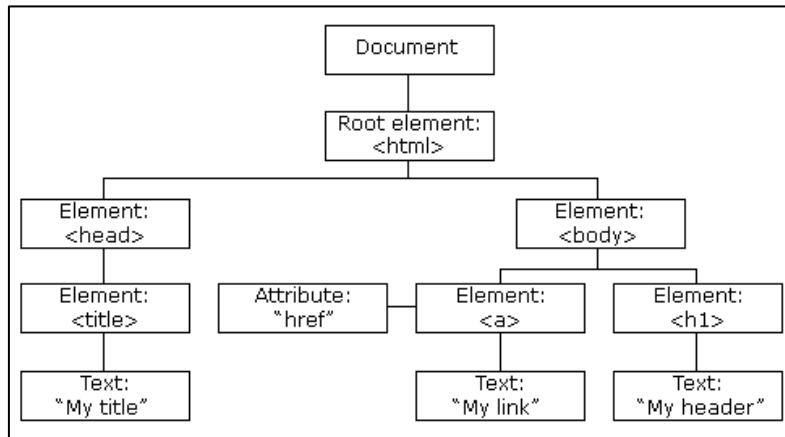
hasilnya :



Gambar 13.17 menggunakan DOM

b) DOM SELECTOR / MENGAkses ELEMENT HTML

Objek ini adalah contoh dari dokumen HTML. Objek ini memiliki semua fungsi dan atribut elemen HTML dan dapat ditampilkan dalam format pohon berikut:



Gambar 13.18 Objek DOM

Fitur objek model JavaScript menyediakan semua alat yang diperlukan untuk membuat HTML dinamis:

- JavaScript memiliki kemampuan untuk memodifikasi setiap elemen HTML di halaman
- JavaScript dapat memodifikasi setiap atribut dalam dokumen HTML.
- JavaScript memiliki kemampuan untuk Mengubah semua aturan CSS dalam dokumen
- Menghapus elemen dan atribut HTML yang ada
- Menambah elemen dan atribut HTML baru
- Bertentangan dengan semua kebijakan HTML yang ada dalam dokumen
- Membuat acara HTML baru dalam dokumen

Ada beberapa fungsi yang dapat digunakan jika Anda ingin mendapatkan elemen saat ini:

- `getElementById()` berfungsi untuk memilih elemen berdasarkan id atribut.
- Fungsi `getElementByName()` digunakan untuk mencari elemen berdasarkan nama atribut.
- `getElementByClassName()` memiliki fungsi menemukan elemen berdasarkan atribut kelas.
- `getElementByTagName()` digunakan untuk mengidentifikasi orang berdasarkan nama tag mereka.
- `getElementByTagNameNS()` memiliki fungsi menemukan elemen berdasarkan nama tag.
- `querySelector()` memiliki fungsi memilih elemen berdasarkan input.
- fungsi `querySelectorAll()` untuk memilih elemen berdasarkan input.
- Dan Lain Lain.

Penggunaan fungsi yang disebutkan di atas untuk mengakses elemen yang ditargetkan sangat umum.

Contoh:

Katakanlah kita memiliki kode HTML seperti ini:

```
<div id="tutorial"></div>
```

Jadi, untuk menemukan elemen yang dimaksud di Javascript, gunakan fungsi `getElementByl()` sebagai berikut:

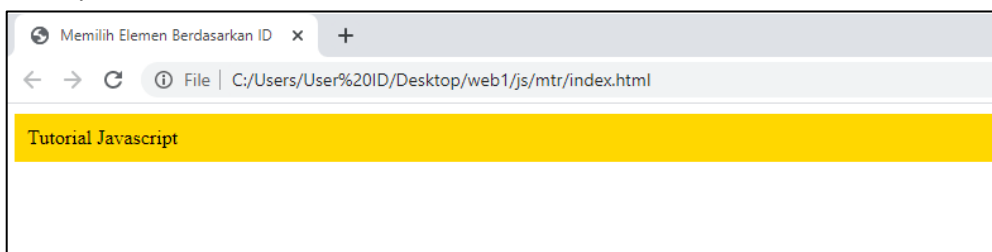
```
// memilih elemen dengan id 'tutorial'  
var tutorial = document.getElementById('tutorial');
```

Variabel instruksional akan menjadi sebuah objek bagi DOM dari elemen yang dipilih. contoh lebih lengkap:

Index.html

```
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4   <title>Memilih Elemen Berdasarkan ID</title>  
5 </head>  
6 <body>  
7  
8   <!-- Elemen div yang akan kita pilih dari JS -->  
9   <div id="tutorial"></div>  
10  
11  
12   <script type="text/javascript">  
13     // mengakses elemen tutorial  
14     var tutorial = document.getElementById("tutorial");  
15  
16     // mengisi teks ke dalam elemen  
17     tutorial.innerText = "Tutorial Javascript";  
18  
19     // memberikan CSS ke elemen  
20     tutorial.style.backgroundColor = "gold";  
21     tutorial.style.padding = "10px";  
22  
23   </script>  
24  
25 </body>  
26 </html>
```

Hasilnya



Gambar 13.19 Contoh Program DOM

Seperti yang ditunjukkan pada contoh di atas, kami menggunakan seseorang dengan fungsionalitas dokumen. `getElementByld()`. Kemudian buat objek untuk elemen yang disebutkan

di atas. Setelah itu, Anda dapat melakukan apa pun yang Anda inginkan, seperti mengedit teks dan gaya CSS.

Pertanyaan selanjutnya: Apa yang akan terjadi jika ada lebih dari satu elemen yang dipilih? Idealnya, Anda harus memilih elemen berdasarkan nama tag atau atribut khusus lainnya.

Jawabannya: Elemen yang masuk akan menjadi array. Karena kami memiliki berbagai elemen.

Baris ini akan menyertakan objek DOM yang terdiri dari anggota populasi yang relevan.

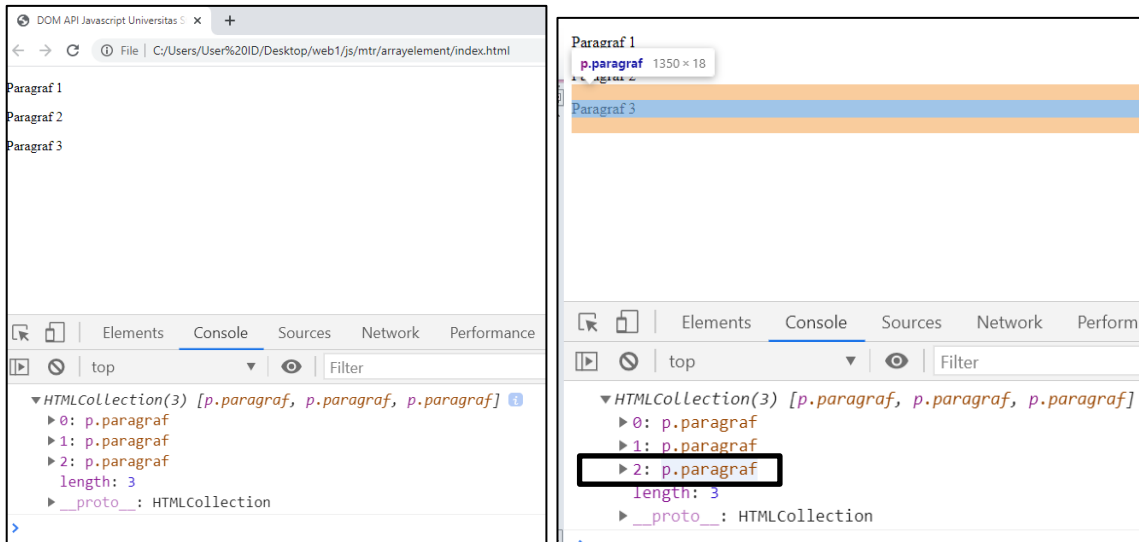
Contohnya:

```
index.html x
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <title>DOM API Javascript Universitas Stekom</title>
6 </head>
7
8 <body>
9
10   <p class="paragraf">Paragraf 1</p>
11
12   <p class="paragraf">Paragraf 2</p>
13
14   <p class="paragraf">Paragraf 3</p>
15
16   <script>
17     var paragraf = document.getElementsByClassName("paragraf");
18     console.log(paragraf);
19   </script>
20
21 </body>
22
23 </html>
```

Pada contoh di atas, kita memiliki tiga paragraf lengkap dengan paragraf yang menamai siswa. Kemudian kita akan mencoba mengakses paragraf selanjutnya menggunakan Javascript menggunakan fungsi atau method `getElementsByClassName()`.

Hasilnya kemudian dilaporkan ke konsol Javascript.

Sebagai hasilnya,



Gambar 13.20 Contoh 2 Program DOM

Akan ada tiga objek DOM dari paragraf yang terkandung dalam array yang akan berisi paragraf variabel.

Kami akan bereksperimen dengan mengubah peringatan tekton dari paragraf pertama.

Paragraf pertama akan berakhir pada indeks posisi ke-0 di dalam array.

Lihat tip konsol Javascript berikut:

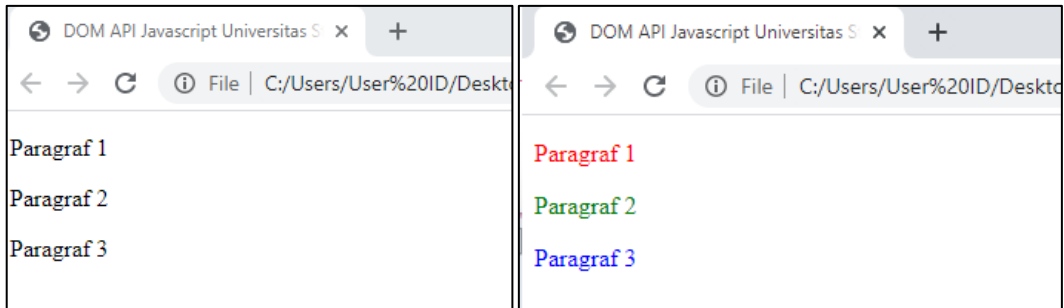
```

12 <p class="paragraf">Paragraf 2</p>
13
14 <p class="paragraf">Paragraf 3</p>
15
16 <script>
17
18 var paragraf = document.getElementsByClassName("paragraf");
19 setInterval(function () {
20     paragraf[0].style.color = "red";
21     paragraf[1].style.color = "green";
22     paragraf[2].style.color = "blue";
23
24     setTimeout(function () {
25         paragraf[0].style.color = "black";
26         paragraf[1].style.color = "black";
27         paragraf[2].style.color = "black";
28     }, 500)
29 }, 1000);
30
31 </script>
32
33 </body>
34
35 </html>

```

Kami menggunakan fungsi setInterval() dan setTimeout() untuk menentukan durasi animasi. Dalam kode tersebut, rentang waktu (interval) adalah 1000 menit atau satu hari. Namun, dibutuhkan 500 milidetik atau 0,5 hari untuk mengubah peringatan menjadi ancaman nyata.

Hasil:



Gambar 13.21 Contoh 3 Program DOM

c) MEMBUAT ELEMEN DENGAN DOM API

Selain itu, DOM menyediakan fungsi untuk membuat elemen HTML. Fungsi pertama disebut `createElement()`.

Contoh:

```
document.createElement('p');
```

Maka, akan tercipta elemen `<p>` baru. Tapi, itu tidak akan dipublikasikan di situs web.

Kenapa tidak dilaporkan? Karena kita belum menambahkannya ke badan dokumen. Untuk melampirkannya ke badan dokumen, Anda dapat menggunakan fungsi `add()`.

Contoh:

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <title>Membuat DOM API Javascript</title>
6   <!-- Universitas Stekom -->
7 </head>
8
9 <body>
10
11   <script>
12     // membuat element h1
13     var judul = document.createElement("h1");
14
15     // mengisi konten elemen
16     judul.textContent = "Belajar Javascript di Universitas Stekom";
17
18     // menambahkan elemen ke dalam tag body
19     document.body.append(judul);
20   </script>
21
22 </body>
23 </html>

```

Hasilnya

:



Gambar 13.22 Contoh Membuat Elemen dengan DOM API

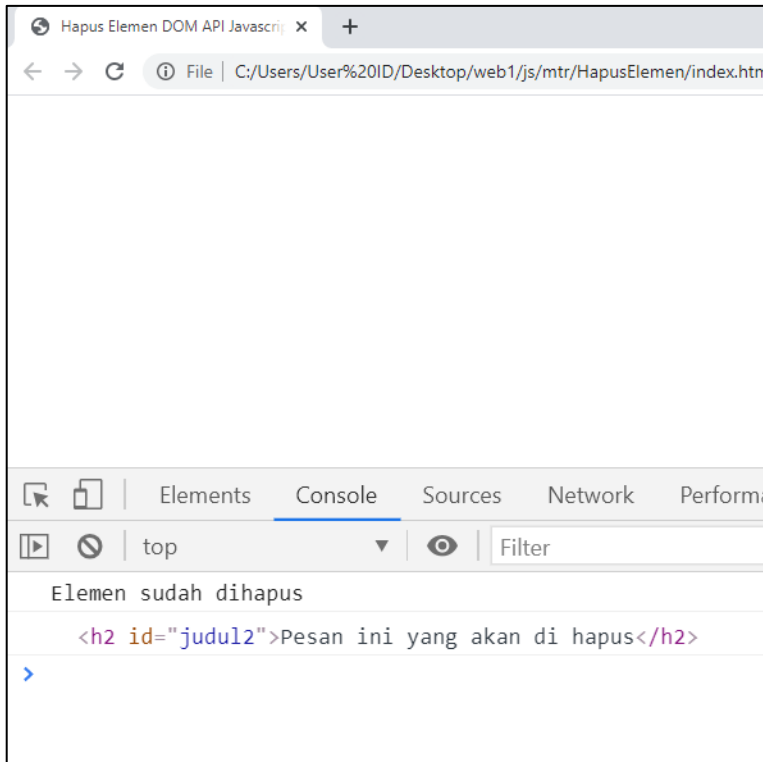
d) MENGHAPUS ELEMEN DENGAN DOM API

Jika sekarang Anda menggunakan fungsi `append()` untuk menambahkan elemen, maka Anda harus menggunakan fungsi `hapus` untuk menghapus elemen dari daftar `()`.

Berikut ini contohnya:

```
index.html x
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <title>Hapus Elemen DOM API Javascript</title>
6 </head>
7
8 <body>
9
10   <h2 id="judul2">Pesan ini yang akan di hapus</h2>
11
12   <script>
13     // memilih elemen berdasarkan ID
14     var Hapus = document.getElementById('judul2');
15
16     // menghapus elemen yang sudah dipilih
17     Hapus.remove();
18
19     console.log("Elemen sudah dihapus");
20     console.log(Hapus);
21   </script>
22
23 </body>
24 </html>
```

hasilnya :



Gambar 13.23 Contoh Menghapus Elemen dengan DOM API

Elemen berhasil diunduh dari situs web, tetapi masih ada di memori. Catatan : Anda bisa mencoba menghapus perintah pada JS line 17 : `Hapus.remove();` maka pesan h2 akan di tampilkan.

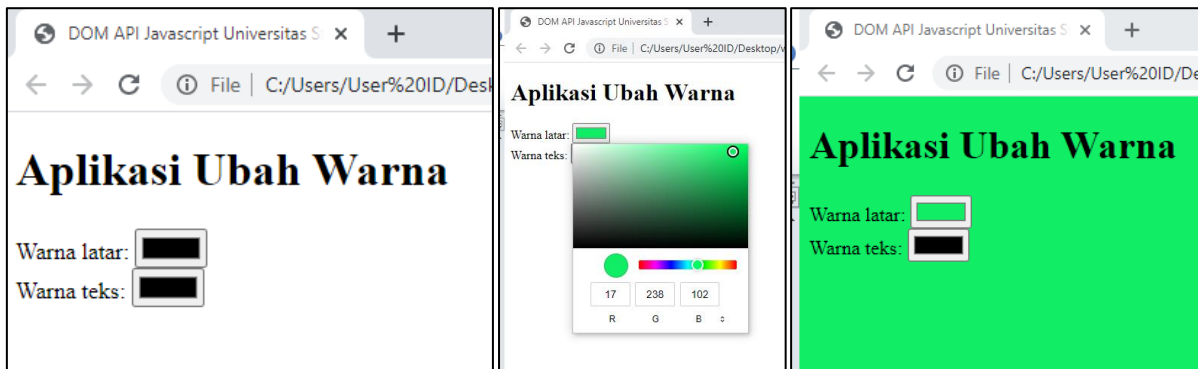
e) CONTOH PROGRAM YANG MEMANFAATKAN DOM

Kami tahu cara mengakses elemen menggunakan DOM, dan kami juga memahami cara memperkuat dan menciutkan elemen. Jika pembicara tampak goyah, jalankan program contoh berikut:

Index.html

```
index.html x
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <title>DOM API Javascript Universitas Stekom</title>
6 </head>
7
8 <body>
9
10  <h1>Aplikasi Ubah Warna</h1>
11  <label>Warna latar: </label>
12  <input type="color" id="bg-color" />
13  <br>
14  <label>Warna teks: </label>
15  <input type="color" id="text-color" />
16
17  <script>
18    var bgColor = document.getElementById('bg-color');
19    var txtColor = document.getElementById('text-color');
20
21    bgColor.addEventListener("change", (event) => {
22      document.body.style.backgroundColor = bgColor.value;
23    });
24
25    txtColor.addEventListener("change", (event) => {
26      document.body.style.color = txtColor.value;
27    });
28  </script>
29
30 </body>
31 </html>
```

Hasilnya :



Gambar 13.24 Contoh Program Yang memanfaatkan DOM

f) DOKUMEN DOM

Menemukan Elemen HTML

Method	Description
<code>document.getElementById(id)</code>	Find an element by element id
<code>document.getElementsByTagName(name)</code>	Find elements by tag name
<code>document.getElementsByClassName(name)</code>	Find elements by class name

Mengubah Elemen HTML

Property	Description
<code>element.innerHTML = new html content</code>	Change the inner HTML of an element
<code>element.attribute = new value</code>	Change the attribute value of an HTML element
<code>element.style.property = new style</code>	Change the style of an HTML element
Method	Description
<code>element.setAttribute(attribute, value)</code>	Change the attribute value of an HTML element

Anda bisa melihat dokumen DOM yang lebih lengkap di situs w3school pada :

https://www.w3schools.com/js/js_htmlDOM_document.asp

BAB 14

MEMBUAT ANIMASI SEDERHANA DENGAN CSS

TUJUAN :

- a. Mahasiswa Mampu Memahami Animasi dengan CSS dan JavaScript
- b. Mahasiswa Dapat Membuat Animasi dengan CSS dan JavaScript

Overview :

Tampilan sangat mempengaruhi kenyamanan dari seorang pengunjung website. Menambahkan animasi yang tidak berlebihan juga mampu memberikan kesan hidup di tampilan website kita. Pada BAB ini akan di ajarkan membuat animasi frame, keyframe serta animasi membuat tulisan dengan menggunakan CSS dan Javascript.

14.1 MEMBUAT ANIMASI TRANSISI

Pada sub poin ini akan di ajarkan cara membuat efek transisi dengan menggunakan CSS dan javascript dimana dalam selector javascriptnya menggunakan teknik DOM. Ikuti langkah – langkah di bawah ini

a) TRANSISI DENGAN CSS

- 1) Buat dokumen HTML seperti di bawah ini dan koneksikan ke CSS

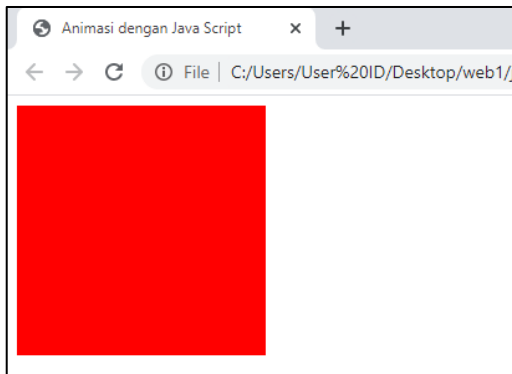
index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Animasi dengan Java Script</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <div id="kotak"></div>
9
10 </body>
11 </html>
```

Style.css

```
1 #kotak{
2   width: 200px;
3   height: 200px;
4   background-color: red;
5 }
```

Hasilnya

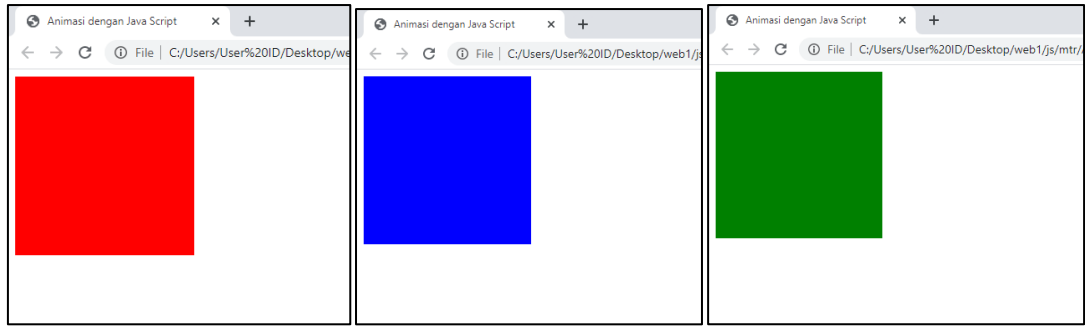


Gambar 14.1 Animasi Transisi

- 2) Berikan atribut hover dan active pada id kotak

```
1 #kotak{
2   width: 200px;
3   height: 200px;
4   background-color: red;
5 }
6
7 #kotak:hover{
8   background-color: blue;
9 }
10
11 #kotak:active{
12   background-color: green;
13 }
```

Lihatkan hasilnya



Gambar 14.2 Hasil Animasi Transisi Warna

- 3) Perubahan warna sangat begitu cepat. Kita dapat menambahkan efek transisi seperti berikut ini pada style #kotak

```
index.html x style.css x
1 #kotak{
2   width: 200px;
3   height: 200px;
4   background-color: red;
5   transition-property: background-color;
6   transition-duration: 0.5s;
7   transition-timing-function: linear;
8   transition-delay: 0s;
9 }
10
11 #kotak:hover{
12   background-color: blue;
13 }
14
15 #kotak:active{
16   background-color: green;
17 }
```

- 4) Silahkan lihat perubahannya. Ada beberapa property transisi selain yang di contohkan di atas. Contoh property lain anda dapat lihat di bawah ini dan silahkan bereksperimen mencobanya.

CSS Properti yang dapat dianimasikan :

background-color	letter-spacing	shape-margin
background-position	line-height	shape-outside
border-color	margin	text-decoration
border-radius	max-height	text-indent
border-width	max-width	text-shadow
border-spacing	min-height	top
bottom	min-width	transform
box-shadow	opacity	transform-origin
clip	outline	vertical-align
color	padding	visibility
font-size	perspective	width
font-weight	perspective-origin	word-spacing
height	right	z-index
left	shape-image-treshold	

Contoh ketika mencoba property lain anda tidak perlu menulis ulang property dalam css. Cukup Anda tambahkan value property yang akan anda coba di belakang value background-color dengan di pisahkan dengan tanda koma

```

\mnt\ANIMASIJS\style.css (ANIMASIJS) - Sublime Text (UNREGISTERED)
Goto Tools Project Preferences Help
index.html x style.css x
1 #kotak{
2   width: 200px;
3   height: 200px;
4   background-color: red;
5   transition-property: background-color, width;
6   transition-duration: 0.5s, 1s;
7   transition-timing-function: linear, ease;
8   transition-delay: 0s;
9 }
10
11 #kotak:hover{
12   background-color: blue;
13   width: 400px;
14 }
15
16 #kotak:active{
17   background-color: green;
18   width: 40px;
19 }
  
```

- Pada value transition-delay (panah kuning) hanya di tulis 1 value. Hal tersebut dapat di lakukan jika kita menghendaki property background-color dengan width memiliki nilai atau value yang sama.
- Pada bagian hover dan active jangan lupa di berikan property width dengan valuenya (panah putih).
- Anda juga dapat menulis syntac seperti di bawah ini agar lebih ringkas

```

mnt\ANIMASIJS\style.css (ANIMASIJS) - Sublime Text (UNREGISTERED)
Goto Tools Project Preferences Help
index.html x style.css x
1 #kotak{
2   width: 200px;
3   height: 200px;
4   background-color: red;
5   transition: background-color 1s linear 0s,
6               width .5s ease 0s;
7 }
8
9 #kotak:hover{
10   background-color: blue;
11   width: 400px;
12 }
13
14 #kotak:active{
15   background-color: green;
16   width: 40px;
17 }
  
```

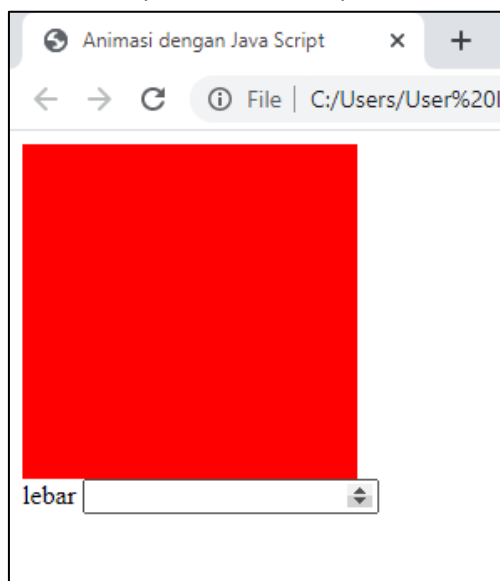
- Value property di pisahkan dengan spasi, namun antar property pisahkan dengan koma. Silahkan lihat perubahannya, aka nada efek transisi perubahan warna dan bentuk.
- 5) Perubahan width dengan value input text. Silahkan hapus property transisi width yang ada pada id kotak sehingga menjadi seperti

```
index.html x style.css x
1 #kotak{
2   width: 200px;
3   height: 200px;
4   background-color: red;
5   transition: width .5s ease 0s;
6 }
7
8 #kotak:hover{
9   background-color: blue;
10  width: 400px;
11 }
12
13 #kotak:active{
14   background-color: green;
15   width: 40px;
16 }
```

6) Pada index.html berikan input text dengan type number

```
index.html x style.css x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Animasi dengan Java Script</title>
5 </head>
6 <body>
7   <link rel="stylesheet" type="text/css" href="style.css">
8   <div id="kotak"></div>
9   lebar <input type="number" step="10"
10  onchange="kotak.style.width=this.value+'px'">
11 </body>
12 </html>
```

Lihat hasilnya, silahkan isi input text tersebut



Gambar 14.3 Transisi dengan Value angka

b) TRANSISI DENGAN JAVASCRIPT

1) Atur index.html seperti di bawah ini

```
index.html x style.css x mainjs x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Animasi dengan Java Script</title>
5 </head>
6 <body>
7   <link rel="stylesheet" type="text/css" href="style.css">
8   <div id="kotak"></div>
9
10  <script src="main.js"></script>
11
12  lebar <input type="number" step="10"
13    onchange="kotak.style.width=this.value+'px'">
14 </body>
15 </html>
```

2) CSS disini hanya untuk mengatur height width dan warna background saja

```
index.html x style.css
1 #kotak{
2   width: 200px;
3   height: 200px;
4   background-color: red;
5
6 }
7
```

3) File main.js

- Dan berikan perintah pada js nya. Caranya sama seperti css tadi= property durasi timing delay

```
FOLDERS
└─ ANIMASIJS
  <─ index.html
  /* main.js
  /* style.css

index.html x style.css x mainjs x
1 var kotak = document.getElementById("kotak");
2 kotak.style.transition="width 1s ease 0s";
```

Timing function selain ease dan linier ada juga yang lain : steps(3) menggunakan berapa langkah

```
FOLDERS
└─ ANIMASIJS
  <─ index.html
  /* main.js

index.html x style.css x mainjs x
1 var kotak = document.getElementById("kotak");
2 kotak.style.transition="width 1s steps(3) 0s";
```

- Kita juga bisa menambahkan perintah disaat transisi berakhir menggunakan keyword transitionend

```

1 var kotak = document.getElementById("kotak");
2 kotak.style.transition="width 1s steps(3) 0s";
3
4 kotak.addEventListener("transitionend",function(){
5     alert("Transisi Telah Selesai...");
6 });

```

Jadi ketika transisi selesai, akan menampilkan dialog alert "Transisi Telah Selesai..."

- Selain menggunakan alert kita juga bisa gunakan untuk hal lain. Contohnya ketika transisi berakhir dia akan kembali ke ukuran semula

```

FOLDERS
  ANIMASJUS
    <> index.html
    /* main.js
    /* style.css

```

```

1 var kotak = document.getElementById("kotak");
2 kotak.style.transition="width 1s linear 0s";
3
4 kotak.addEventListener("transitionend",function(){
5     kotak.style.width='200px';
6 });
7

```

- Atau anda bisa mengarahkan ke halaman web

```

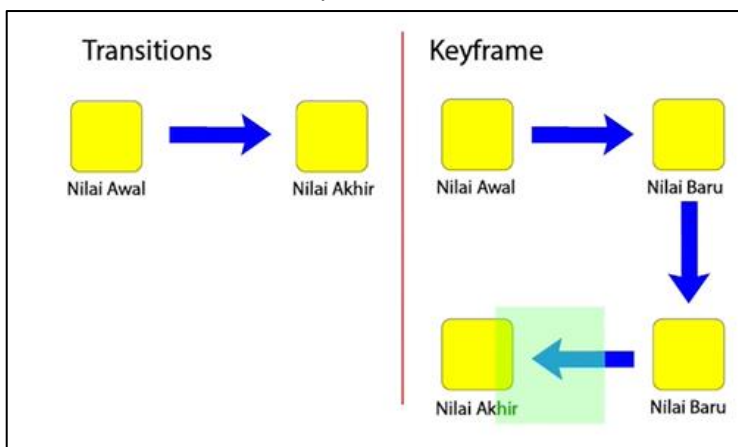
1 var kotak = document.getElementById("kotak");
2 kotak.style.transition="width 1s linear 0s";
3
4 kotak.addEventListener("transitionend",function(){
5     window.location="https://stekom.ac.id";
6 });
7

```

c) ANIMASI KEYFRAME

Keyframe dalam animasi atau film menurut Wikipedia adalah sebuah gambar yang mendefinisikan titik permulaan dan akhir dari transisi apapun. Gambar tersebut disebut "frame" karena posisi mereka pada waktu diukur di frame pada sebuah strip film.

Perbedaan Transisi dan Keyframe



Gambar 14.4 Animasi Keyframe

Sekarang kita coba untuk membuat animasi keyframe tersebut.

1) Buatlah file html seperti berikut

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Keyframe</title>
5 </head>
6 <body>
7   <link rel="stylesheet" type="text/css" href="style.css">
8   <div id="objek"></div>
9
10 </body>
11 </html>

```

2) Buat animasi keyframe dengan css

```

mt\ANIMASI.JS\KEYFRAME\style.css (ANIMASI.JS) - Sublime Text (UNREGISTERED)
Goto Tools Project Preferences Help

1 #objek{
2   width: 200px;
3   height: 200px;
4   border-radius: 100%;
5   background: blue;
6   animation-name: animasi1;
7   animation-duration: 2s;
8 }
9
10 @keyframes animasi1{
11   0% {transform: translateX(0px)}
12   100% {transform: translateX(500px)}
13 }

```

Selain contoh diatas masih terdapat property animasi lain seperti

NO	PROPERTY	KETERANGAN
1	animation-timing-function: ease;	Perubahan atau jalannya animasi. Value ease dan linier
2	animation-iteration-count: 1;	akan mengulang berapa x. 1 atau 2 atau 3 dst, atau infinite (mengulang terus)
3	animation-direction: normal;	ini arah animasi. Reverse terbalik. Alternate bolak- balik. Alternate-reverse bolak balik dimulai dari kanan
4	animation-play-state: running;	Memulai atau menghentikan animasi. running berjalan, paused berhenti
5	animation-delay: 0s;	waktu tunggu sebelum animasi berjalan
6	animation-fill-mode: none;	Akhir dari animasi. None animasi kembali ke posisi awal, forwards animasi berakhir di posisi akhir(count set 1)

Dalam penulisan property dapat juga disingkat seperti dibawah ini

```

1 #objek{
2   width: 200px;
3   height:200px;
4   border-radius: 100%;
5   background: blue;
6   animation: animasi1 2s linear infinite normal
7   running 0s forwards;
8

```

Namun beberapa value tidak perlu di tulis semua. tulis yang di butuhkan saja (buang yg settingan default) sehingga

```

1 #objek{
2   width: 200px;
3   height:200px;
4   border-radius: 100%;
5   background: blue;
6   animation: animasi1 2s linear infinite;
7

```

3) Tambahkan titik koordinat X dan Y

```

4   border-radius: 100%;
5   background: blue;
6   animation: animasi1 3s ease infinite;
7
8
9 }
10
11 @keyframes animasi1{
12   0% {transform: translateX(0px)}
13   25% {transform: translateX(500px)}
14   50% {transform: translateX(500px) translateY(400px)}
15   75% {transform: translateX(0px) translateY(400px)}
16   100% {transform: translateX(0px) translateY(0px)}
17 }

```

4) Anda juga dapat membuat animasi lebih dari satu.

```

4   border-radius: 100%;
5   background: blue;
6   animation: animasi1 3s ease infinite,
7             animasi2 .5s ease infinite;
8
9
10 }
11
12 @keyframes animasi1{
13   0% {transform: translateX(0px)}
14   25% {transform: translateX(500px)}
15   50% {transform: translateX(500px) translateY(400px)}
16   75% {transform: translateX(0px) translateY(400px)}
17   100% {transform: translateX(0px) translateY(0px)}
18 }
19
20 @keyframes animasi2{
21   0% {border-radius: 100%}
22   100% {border-radius: 0%}
23 }

```

5) Dapat juga di tambahkan perubahan warna

```
index.html x style.css x
4 border-radius: 100%;
5 background: blue;
6 animation: animasi1 3s ease infinite,
7 animasi2 .5s ease infinite;
8
9
10 }
11
12 @keyframes animasi1{
13 0% {transform: translateX(0px)}
14 25% {transform: translateX(500px)}
15 50% {transform: translateX(500px) translateY(400px)}
16 75% {transform: translateX(0px) translateY(400px)}
17 100% {transform: translateX(0px) translateY(0px)}
18 }
19
20 @keyframes animasi2{
21 0% {border-radius: 100%; background: red}
22 100% {border-radius: 0%; background: blue}
23 }
```

d) Contoh lain

1) Show hidden password

Index.html

```
index.html x main.js x style.css x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Login</title>
5 <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8 <form action="#" class="jw-form">
9 <input type="text" value="" id="myuser" placeholder="Username"><br>
10 <input type="password" value="" id="mypass" placeholder="Password"><br>
11 <input type="button" id="show" value="Show Password" onclick="ShowPassword()">
12 <input type="button" style="display:none" id="hide" value="Hide Password" onclick=
"HidePassword()">
13 </form>
14 <script src="main.js"></script>
15 </body>
16 </html>
```

Style.css

```
index.html x main.js x style.css x
1 header{
2 margin: 30px auto;
3 text-align: center;
4 color: #333;
5 }
6 header h1{
7 font-size: 20px;
8 }
9 header h4{
10 font-size: 16px;
```

```

11 }
12 form{
13     max-width: 300px;
14     margin: 50px auto;
15     padding: 20px;
16     background: #f4f7f8;
17     border-radius: 8px;
18     border: 1px solid #d4f7f8;
19 }
20 input[type="text"],
21 input[type="password"]{
22     border: 1px solid #d4f7f8;
23     padding: 15px;
24     font-size: 17px;
25 }
26 input[type="button"]{
27     margin-top: 10px;
28     border: none;
29     padding: 5px;
30     font-size: 13px;
31 }

```

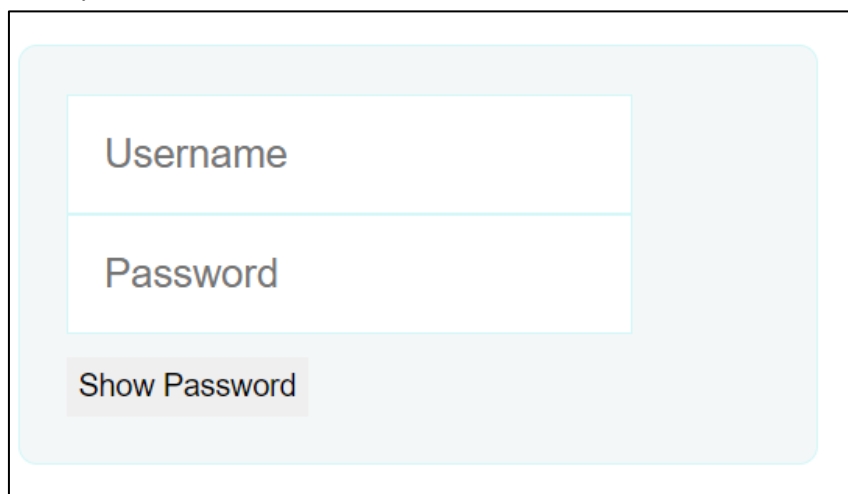
Main.js

```

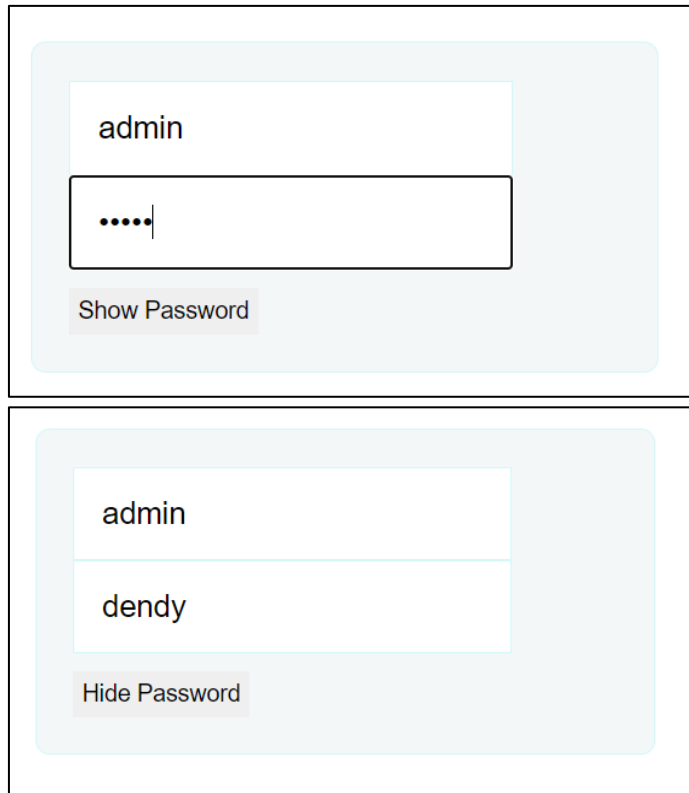
1 function ShowPassword()
2 {
3     if(document.getElementById("mypass").value!="")
4     {
5         document.getElementById("mypass").type="text";
6         document.getElementById("show").style.display="none";
7         document.getElementById("hide").style.display="block";
8     }
9 }
10
11 function HidePassword()
12 {
13     if(document.getElementById("mypass").type == "text")
14     {
15         document.getElementById("mypass").type="password"
16         document.getElementById("show").style.display="block";
17         document.getElementById("hide").style.display="none";
18     }
19 }

```

Hasilnya



Gambar 14.5 Animasi show hidden password



Gambar 14.5 Animasi show hidden password

2) Animasi Text

Index.html

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Portofolio</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6   <script src="https://cdn.jsdelivr.net/npm/typed.js@2.0.9"></script>
7 </head>
8 <body>
9   <div class="text-1">Halo, Namaku adalah</div>
10  <div class="text-2">Dendy Kurniawan</div>
11  <div class="text-3">Saya seorang <span class="typing"></span></div>
12  <a href="#">Selengkapnya</a>
13  <script src="main.js"></script>
14 </body>
15 </html>

```

Style.css

```

1 body{
2   background: black;
3 }
4 .text-1{
5   font-size: 27px;
6   color: white;
7 }
8 .text-2{
9   font-size: 75px;
10  font-weight: 600;
11  margin-left: -3px;
12  color: white;

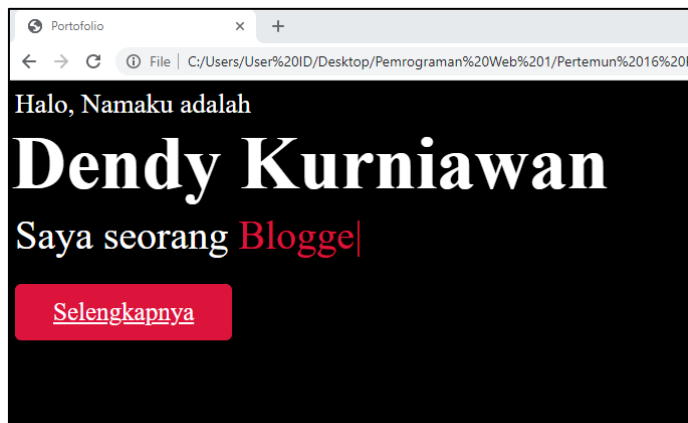
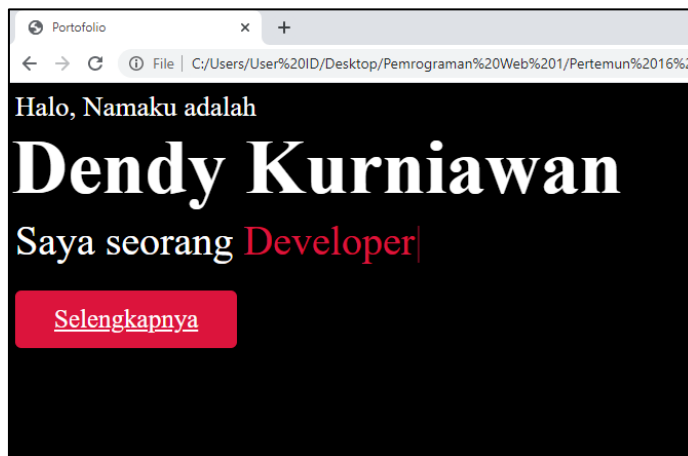
```

```
13 }
14 .text-3{
15     font-size: 40px;
16     margin: 5px 0;
17     color: white;
18 }
19 .text-3 span{
20     color: crimson;
21     font-weight: 500;
22 }
23 a{
24     display: inline-block;
25     background: crimson;
26     color: #fff;
27     font-size: 25px;
28     padding: 12px 36px;
29     margin-top: 20px;
30     font-weight: 400;
31     border-radius: 6px;
32     border: 2px solid crimson;
33 }
```

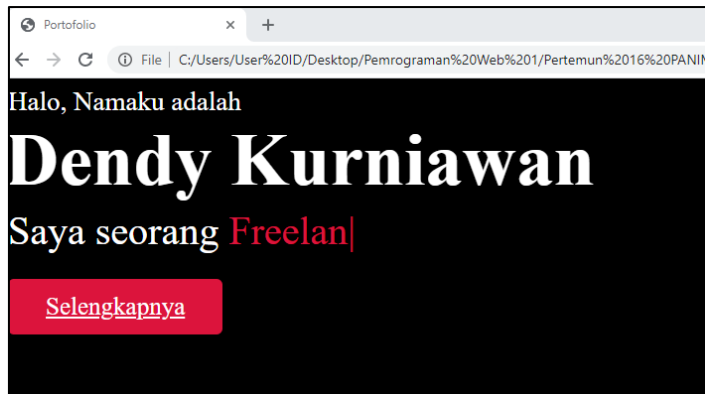
Main.js

```
1 var typed = new Typed(".typing", {
2     strings: ["YouTuber", "Developer", "Blogger", "Designer", "Freelancer"],
3     typeSpeed: 100,
4     backSpeed: 60,
5     loop: true
6 });
7
```

Hasilnya



Gambar 14.6 Animasi Menulis Teks



Gambar 14.6 Animasi Menulis Teks

BAB 15

MEMBUAT ANIMASI SEDERHANA DENGAN CSS

TUJUAN :

- a. Mahasiswa Mampu Memahami Cara Mengonlinekan Website
- b. Mahasiswa Dapat Mengonlinekan Website yang sudah di Buatnya

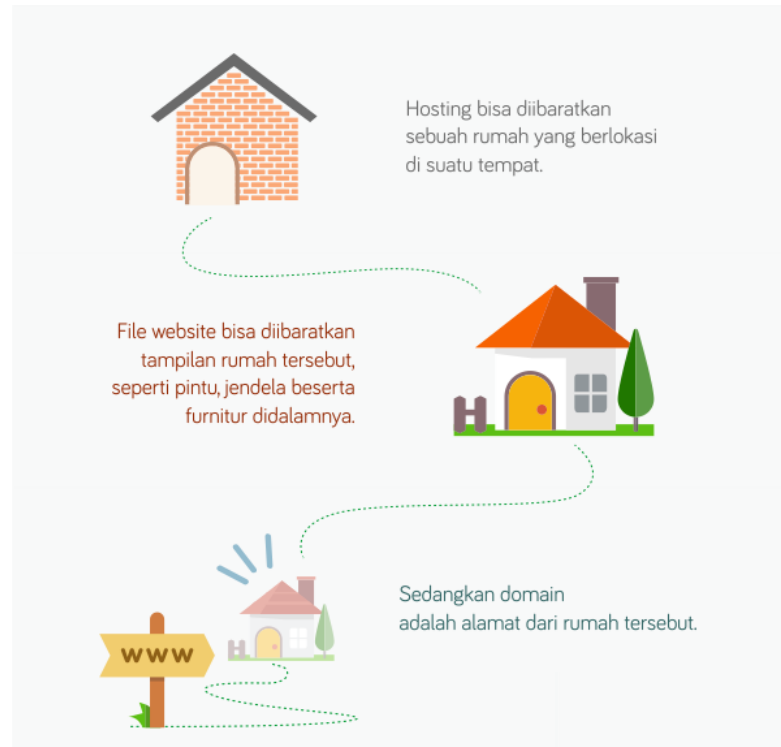
Overview :

Pada BAB ini akan di ajarkan cara mengonlinekan website secara gratis. Semua mahasiswa diwajibkan mengonlinekan web masing – masing Karena nanti dalam penilaian UAS, mahasiswa harus mengirimkan link web online nya ke pada Dosen.

15.1 PENGANTAR WEB ONLINE

Jika Anda membuat sebuah blog dengan menggunakan Wordpress.com atau Blogger.com, maka web tersebut akan langsung Online. Namun jika Anda membuat web dengan pemrograman HTML, atau PHP dan sejenisnya yang menggunakan bahasa pemrograman, maka Anda harus meng Onlinekan web tersebut secara manual. Pada bab ini akan di bahas cara meng online kan web HTML secara gratis. Namun sebelumnya Anda harus mengenal terlebih dahulu istilah Hosting dan Domain.

Hosting berfungsi sebagai sarana untuk menyimpan semua file website agar dapat diakses secara online. Setiap file, termasuk tema, plugin, teks, media, kode HTML dan CSS, dan bahkan draf konten, dihosting. Sebaliknya, domain Anda berfungsi sebagai alamat situs web Anda sehingga pengguna dapat dengan mudah mengaksesnya menggunakan browser. Misalnya, ketika seseorang ingin mengakses website Niagahoster, mereka hanya perlu mengetikkan www.niagahoster.co.id di browser mereka sebelum website Niagahoster dimuat. Domain tidak harus selalu berakhir .COM. Anda memiliki opsi untuk memilih salah satu dari banyak ekstensi domain yang tersedia. Diawali dengan .CO.ID,.WEB.ID,.NET,.INFO,.ONLINE,.SITE,.WEBSITE, dan diakhiri dengan .XYZ

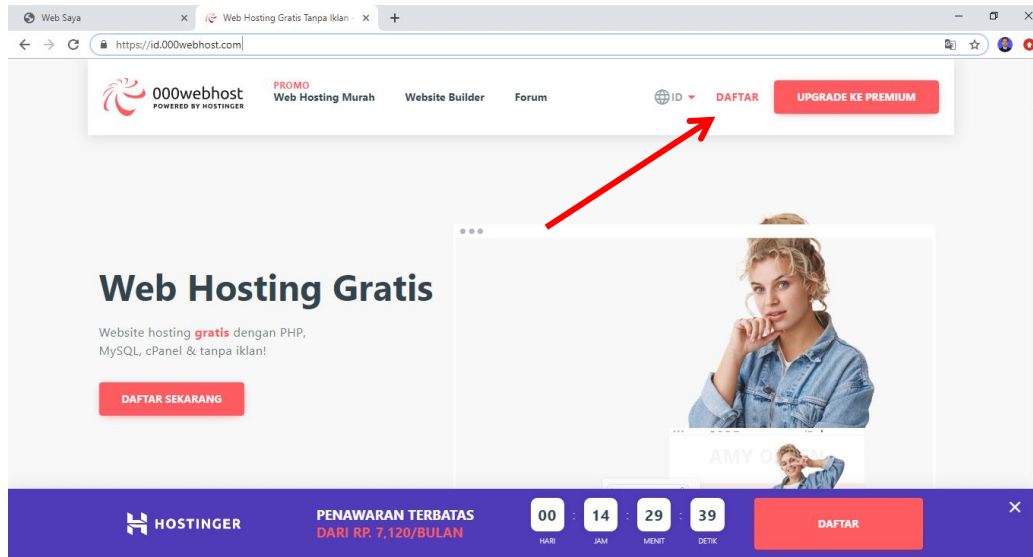


Gambar 15.1 Analogi Hosting dan Domain (sumber Niagahoster)

15.2 MENGONLINEKAN WEB

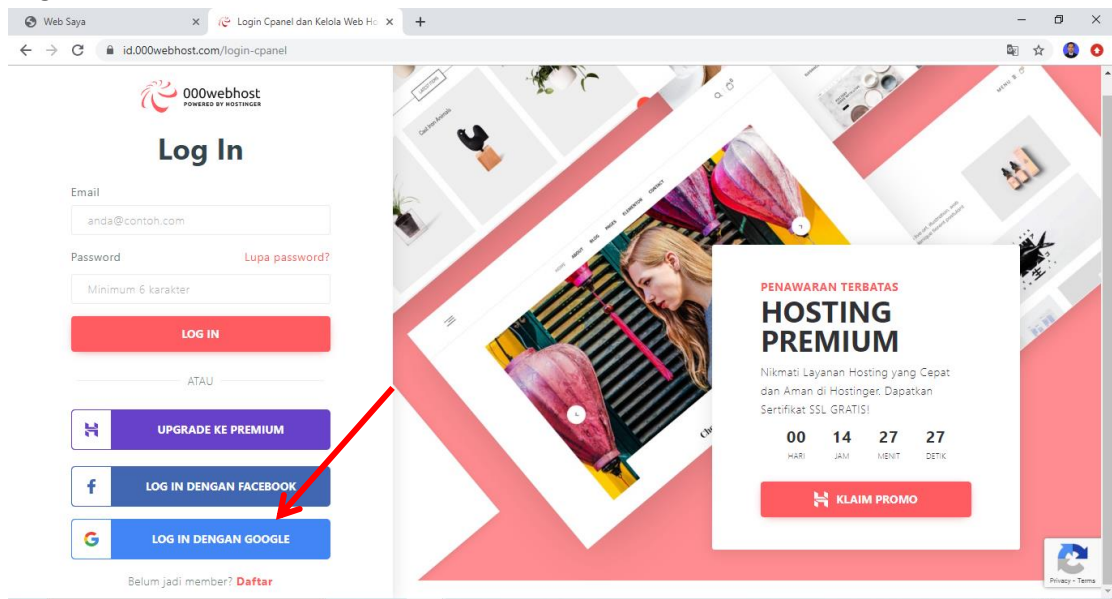
000webhost adalah layanan hosting gratis dan mudah digunakan. Anda tidak perlu lagi menghabiskan uang untuk web hosting, apalagi Anda baru memulai website. Selain itu, menyiapkan situs web dengan 000webhost tidaklah sulit; pemula bahkan dapat melakukannya secara efektif. Penggunaan sistem antarmuka yang mudah digunakan menyebabkan seluruh proses instalasi berjalan dengan cepat. Mari, ayo pergi bersama.

1. Silahkan Anda mendaftar 000webhost terlebih dahulu, kunjungi link di <https://id.000webhost.com/> klik daftar



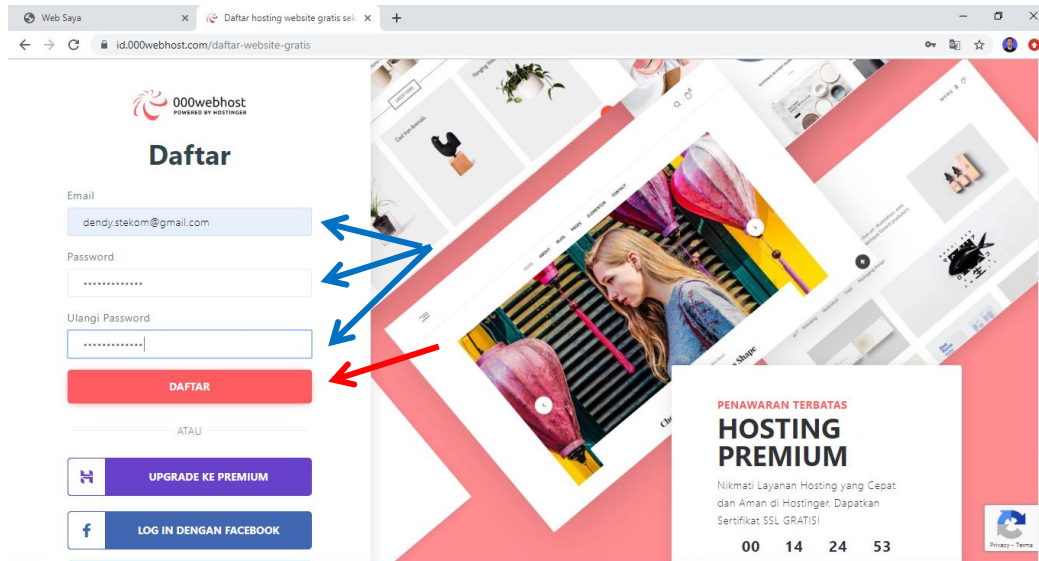
Gambar 15.2 halaman utama 000webhost

2. Maka Anda akan di arahkan ke halaman seperti ini. Anda sebenarnya bisa login menggunakan FB atau Google langsung. Namun disini yang saya ajarkan adalah mendaftar manual saja meskipun nantinya juga akan menggunakan email. Silahkan klik pada bagian Daftar pada bagian bawah kiri



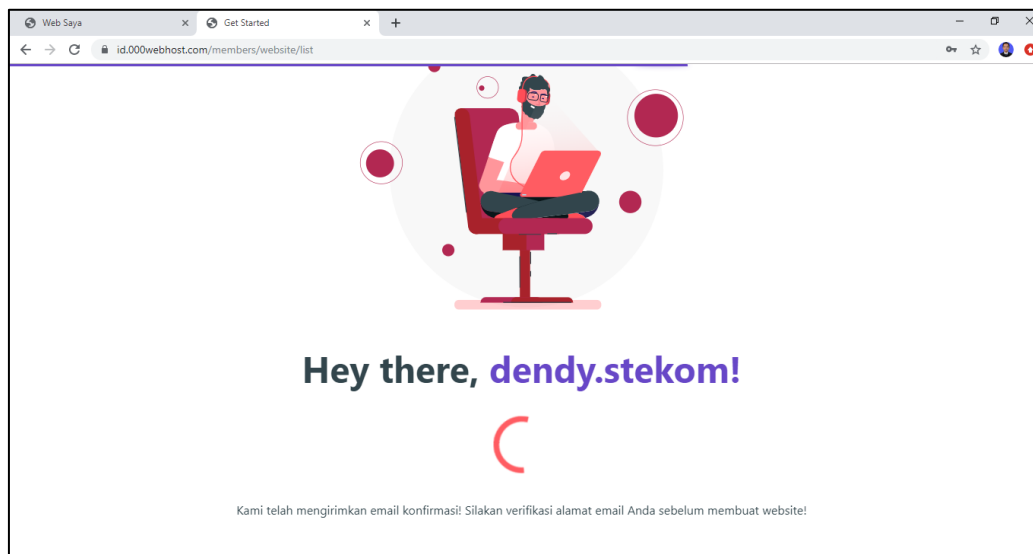
Gambar 15.3 Halaman login 000webhost

3. Anda akan dikirim ke gedung yang mirip dengan yang ada di bawah., silahkan isikan email Anda yang masih aktif dan buatlah password untuk akun 000webhostnya kemudian klik Daftar. *Note : Kata sandi harus tepat satu angka, satu huruf besar dan kecil, dan tepat delapan karakter, atau lebih.*



Gambar 15.4 Halaman Register 000webhost

4. Setelah klik daftar maka webhost akan mengirim pesan verifikasi ke email yang Anda daftarkan tadi.



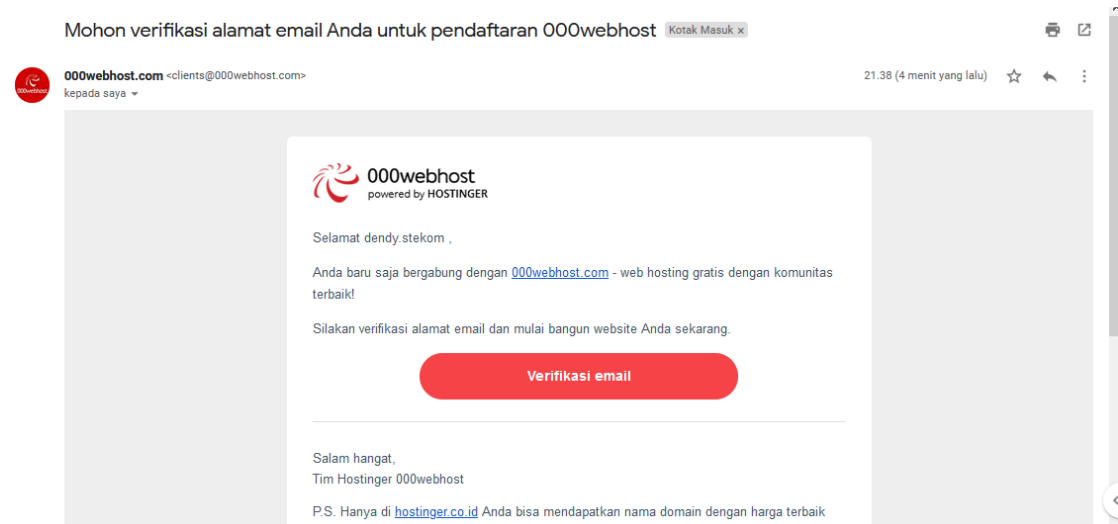
Gambar 15.5 Pendaftaran selesai

5. Silahkan masuk ke Email Anda untuk melakukan verifikasi nya. Klik pesan dari 000webhost.

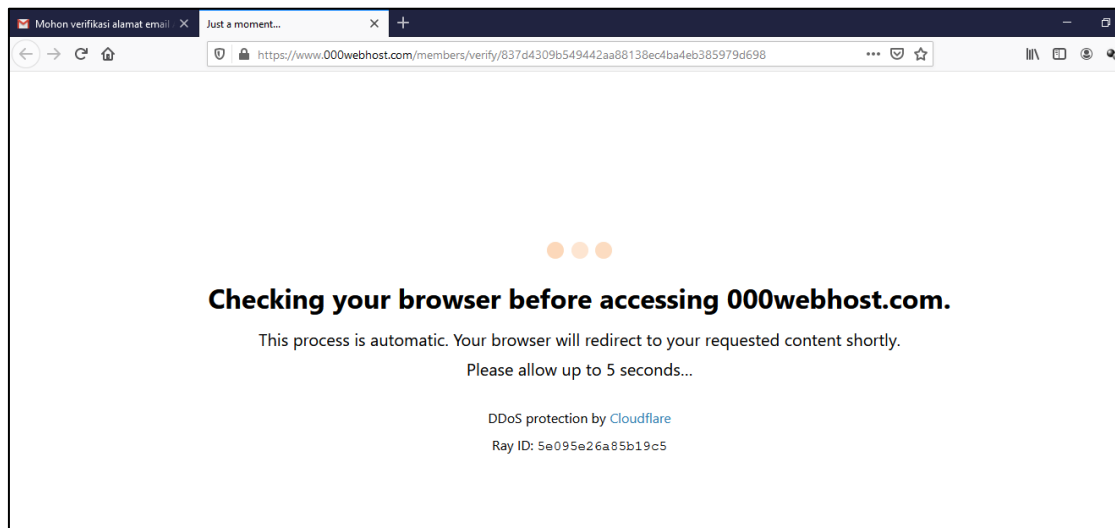


Note : Jika tidak terdapat pesan ini di kotak masuk, Silahkan anda coba cari di folder spam

6. Setelah membuka pesan nya, silahkan klik Verifikasi Email dan tunggu sampai selesai

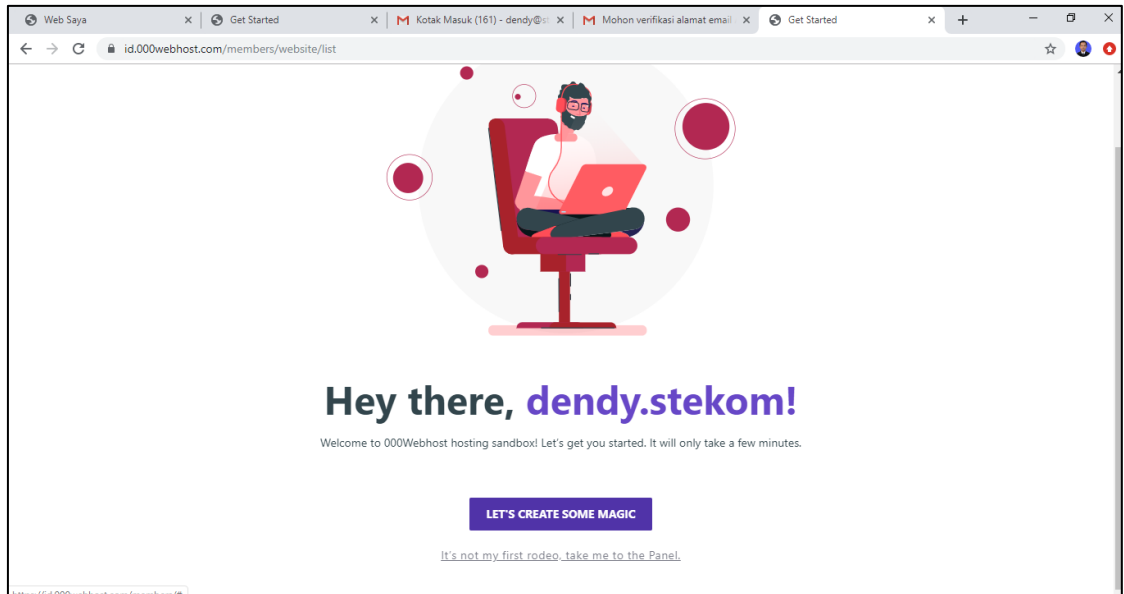


Gambar 15.6 Verifikasi Email



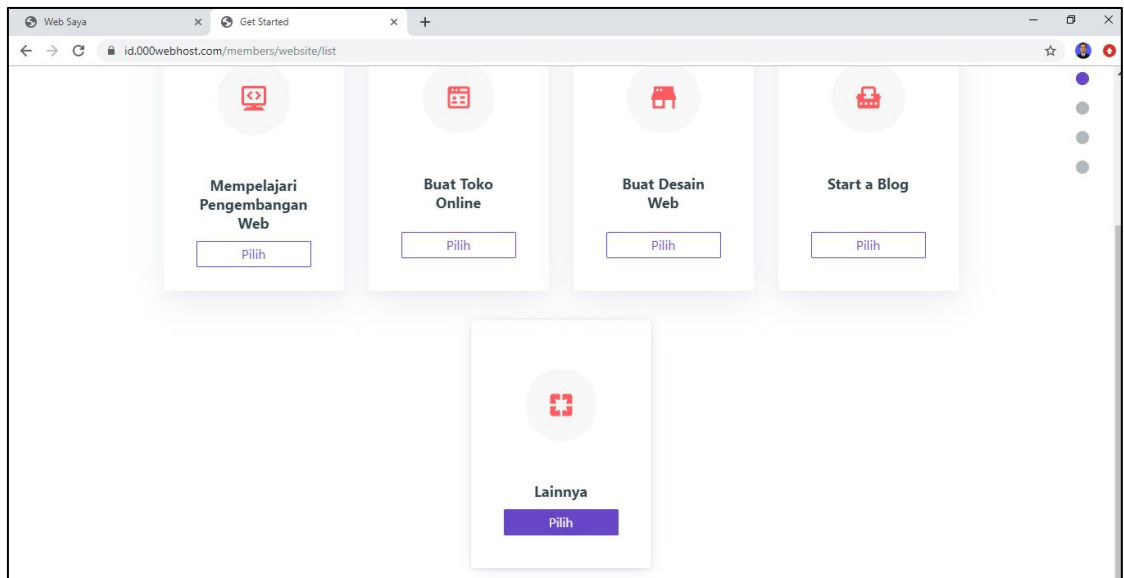
Gambar 15. Verifikasi Berhasil

7. Setelah verifikasi selesai, Anda akan dibawa ke gedung yang ditunjukkan pada paragraf di bawah ini. Anda juga dapat mencari di goolge dengan kata kunci login 000webhost atau anda dapat mengunjungi langsung link hasil pencarian tersebut di <https://id.000webhost.com/members/website/list> silahkan klik LET'S CREAT SOME MAGIC



Gambar 15.8 Halaman webhost setelah di verifikasi email

8. Kemudian pilih lainnya yang paling bawah



Gambar 15.9 Menu lainnya

9. Isikan nama website yang Anda kehendaki, disini saya contohkan nama website saya appstorendeny. Dan silahkan ganti password nya agar lebih mudah di ingat. Ini berbeda dengan password akun 000webhost yang ada di tahap sebelumnya. kemudian klik kirim.

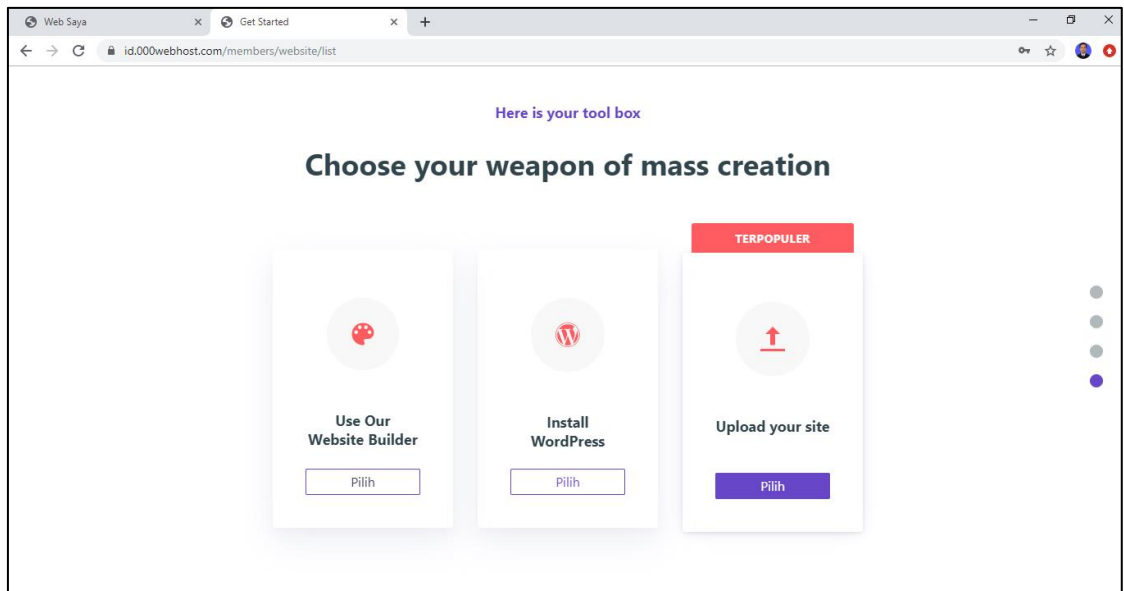
A great start is half the work

Name Your Project

Tampilkan password [BUAT PASSWORD](#)

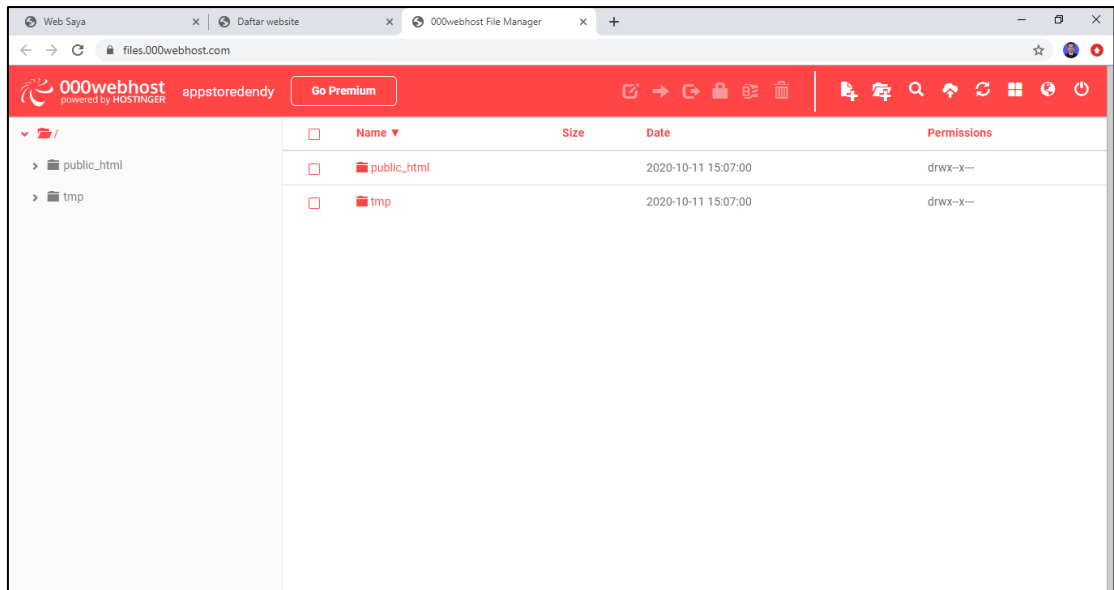
Gambar 15.10 penamaan projek

10. Jika nama website tersedia maka pendaftaran akan berhasil. Namun jika gagal ada kemungkinan nama tidak tersedia karena nama sudah di gunakan orang lain / menggunakan kata hubung / menggunakan karakterter yang tidak di dukung. Silahkan Anda ganti dengan nama lain atau di tambah kata lain tanpa tanda hubung.
11. Jika sudah berhasil akan di tampilkan halaman seperti berikut ini dan silahkan klik Upload Your Site



Gambar 15.11 Pemilihan Jenis Projek

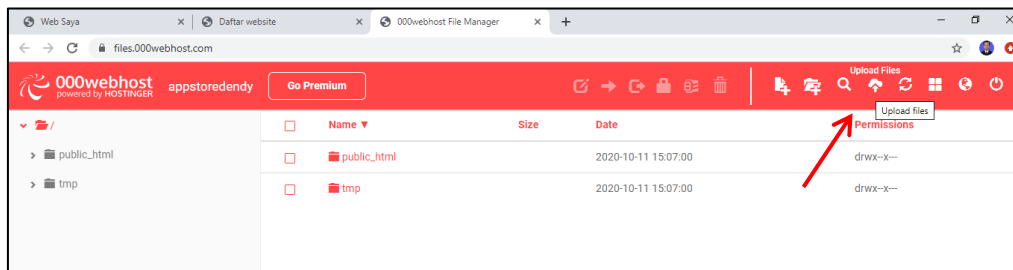
12. Akan di tampilkan halaman seperti berikut ini



Gambar 15.12 Halaman file 000webhost

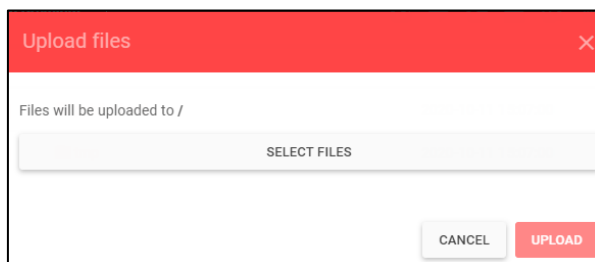
13. Disitulah nanti file – file HTML, CSS, image JS dan lain – lain mengenai website Anda akan disimpan (didalam folder public_html).

14. Silahkan klik upload file pada bagian kanan atas



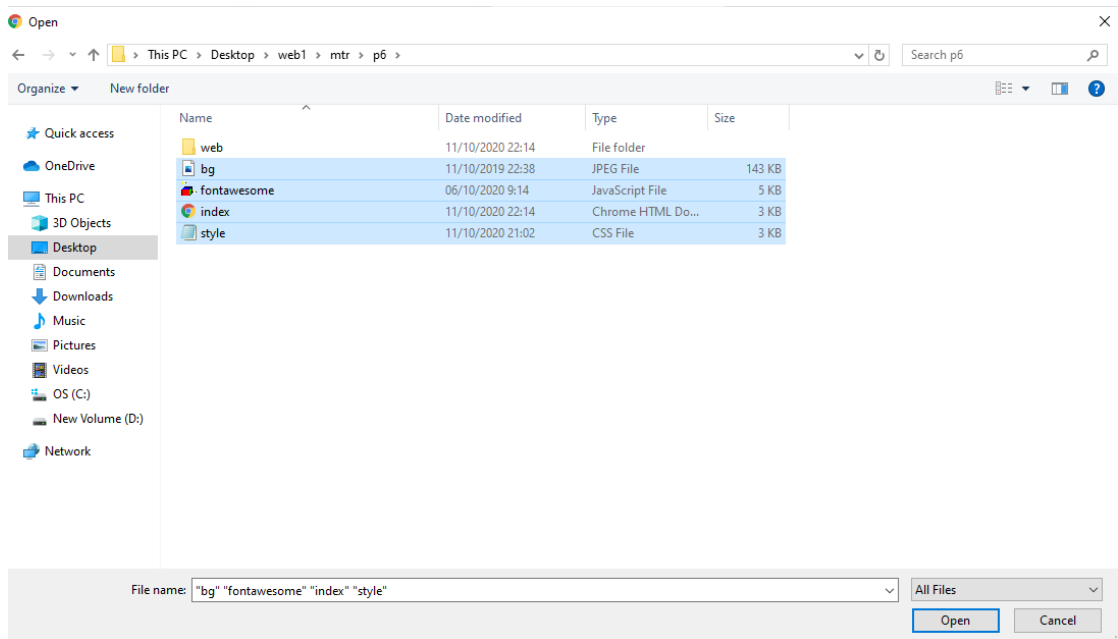
Gambar 15.13 Upload File di 000webhost

15. Klik select file



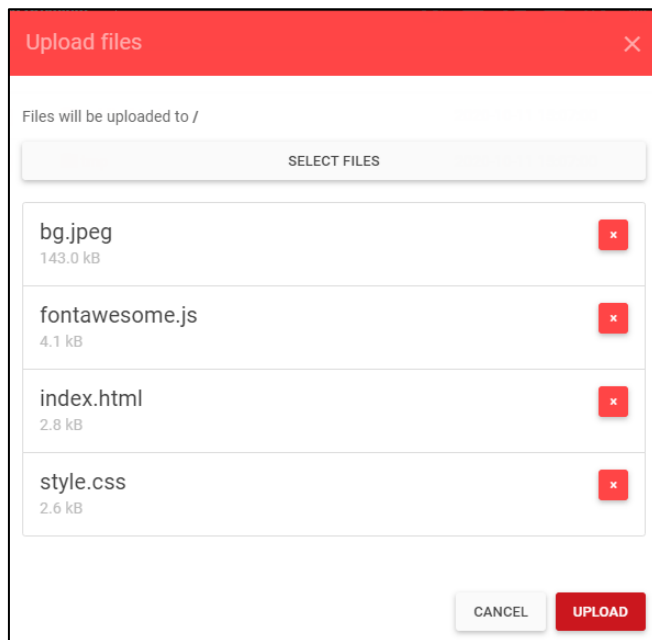
Gambar 15.14 Browse File

16. Pilih semua file yang berhubungan dengan web Anda kemudian klik Open



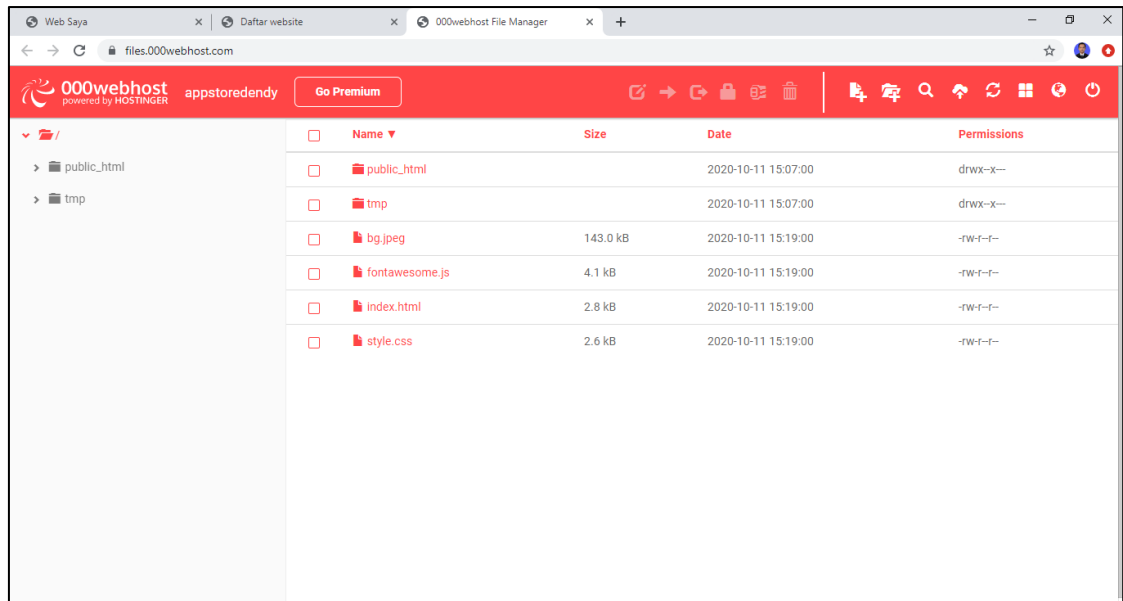
Gambar 15.15 Pilih File yang akan di upload

17. Klik Upload



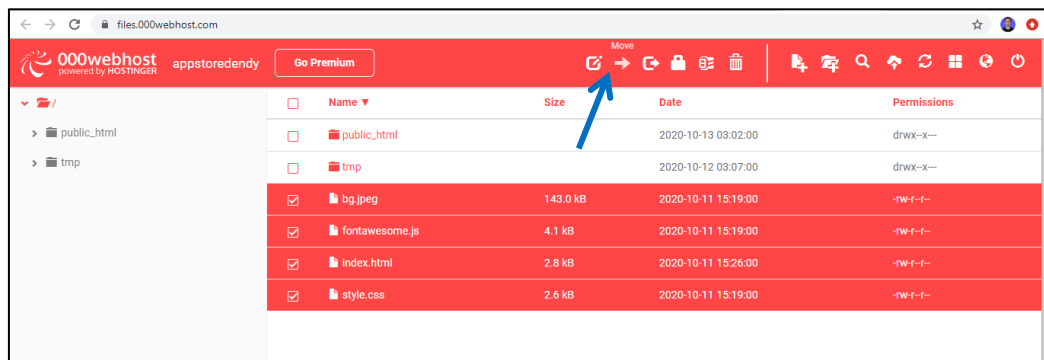
Gambar 15.16 Daftar file yang akan di upload

18. Seluruh file telah terupload

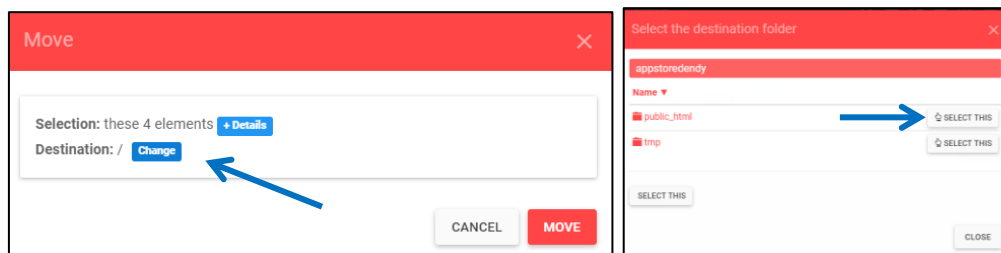


Gambar 15.17 File setelah berhasil di upload

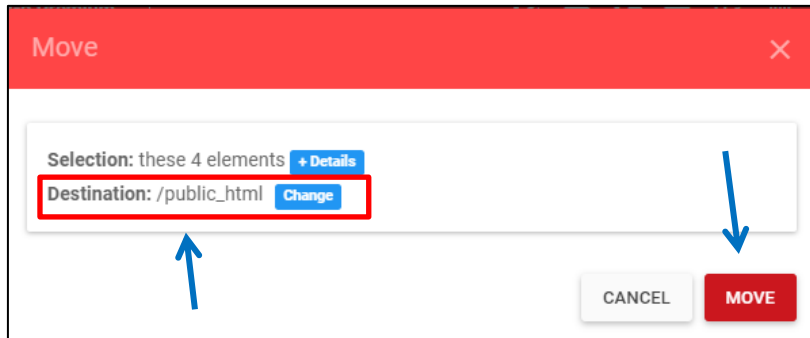
19. Setelah itu anda masukkan seluruh file yang terupload tadi ke dalam folder Public_html. Centang semua file dan pilih move dan change folder target, pilih public_html dan ok



Gambar 15.18 Pindah File

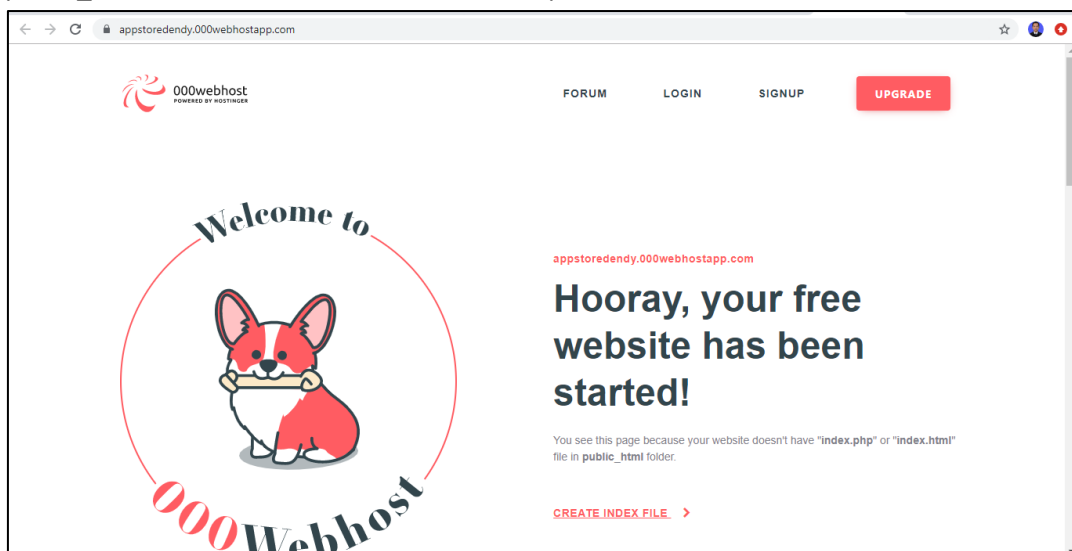


Gambar 15.19 Pilih Tujuan Folder



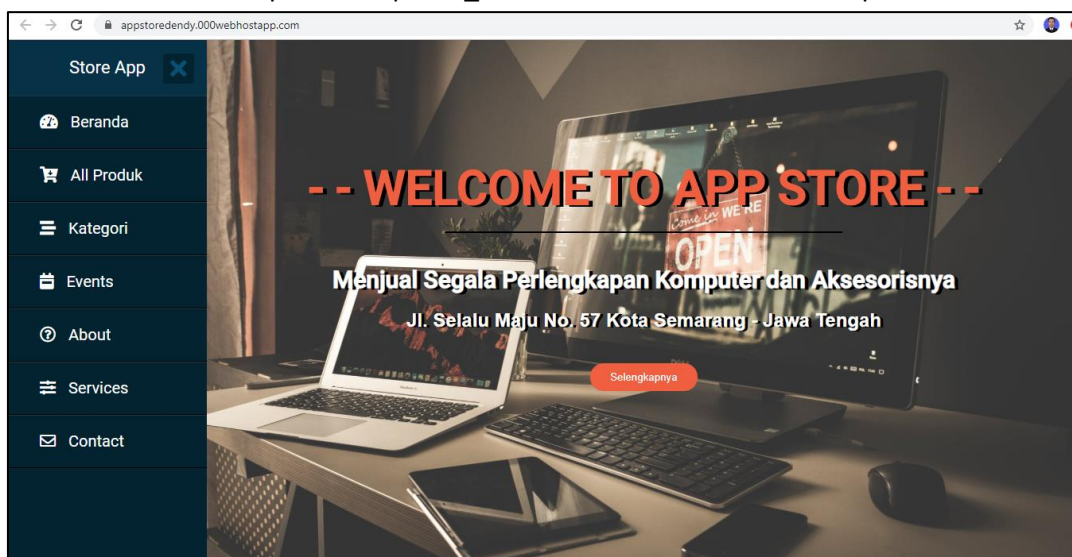
Gambar 15.20 Tombol Simpan Memindahkan File

20. Jika anda belum mengupload file HTML atau belum memasukkan file HTML ke folder public_html maka halaman web anda akan seperti di bawah ini



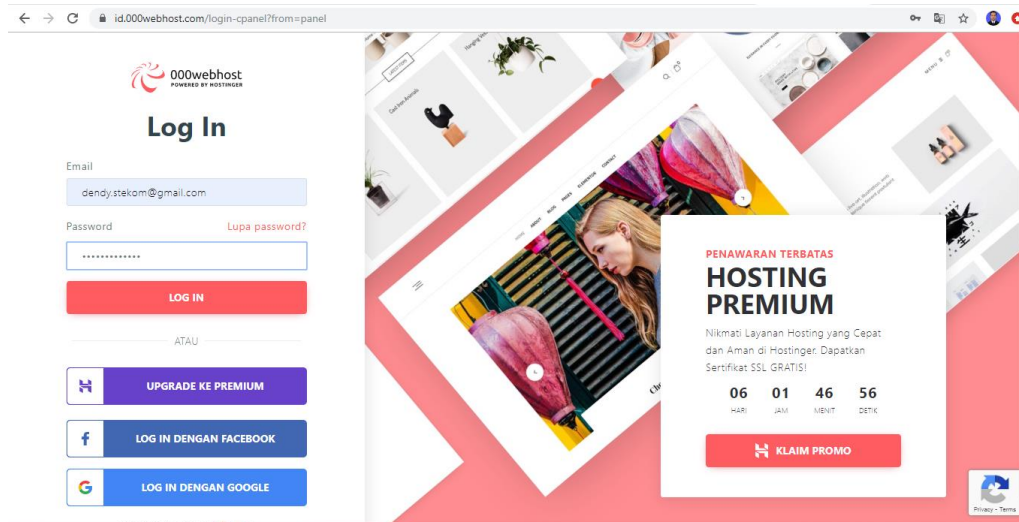
Gambar 15.21 Halaman Depan Web Sebelum

21. Setelah seluruh file di pindah ke public_html maka web anda sudah tampil online



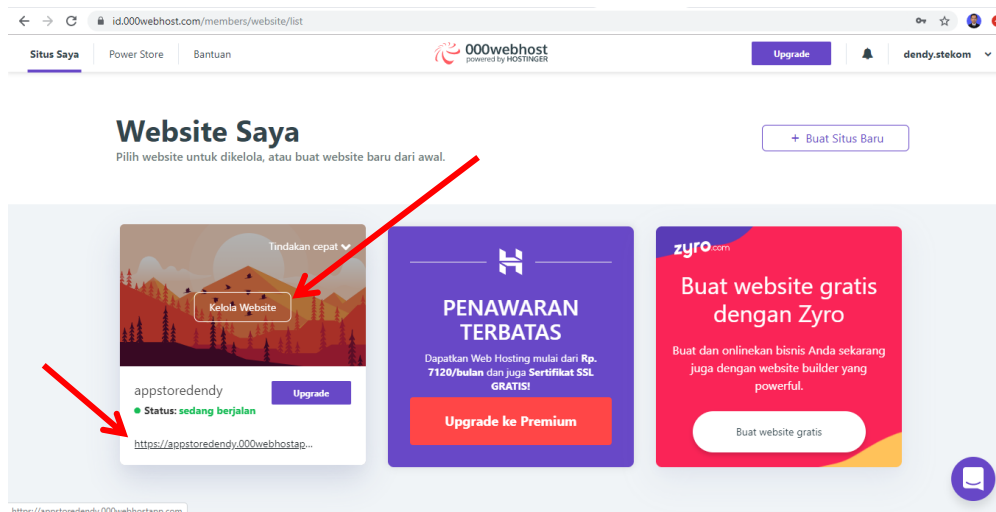
Gambar 15.22 Halaman Depan Web Sesudah

22. Dilain hari anda dapat setting atau membuka kembali web anda dengan mengunjungi <https://id.000webhost.com/login-cpanel?from=panel> atau cari di web browser dengan kata kunci login 000webhost isikan email dan paswrod ketika mendaftarkan akun webhost tersebut (pada tahap ke 3 tutorial ini)



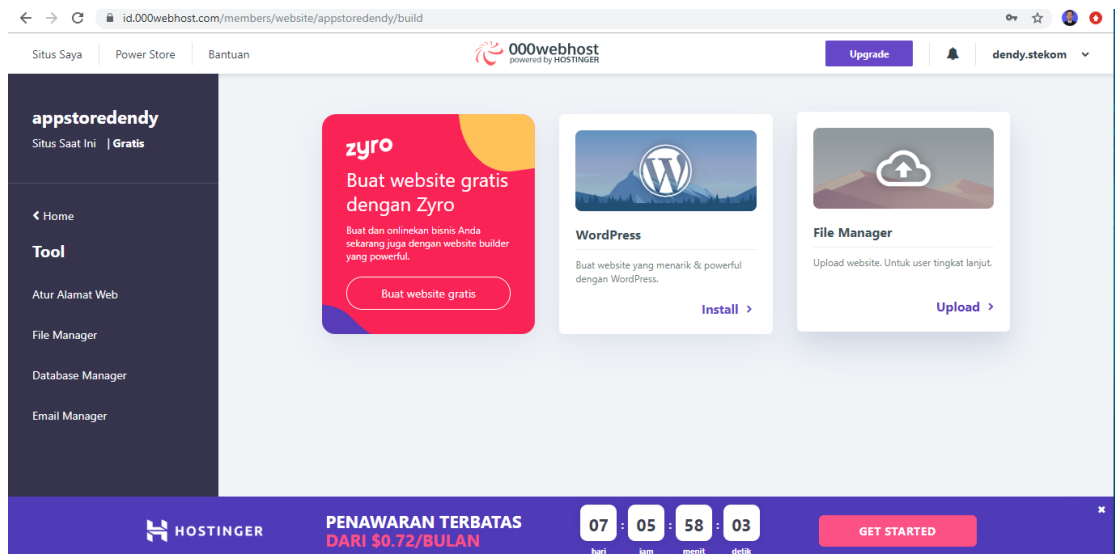
Gambar 15.23 Halaman Login webhost

23. Anda akan diarahkan ke halaman berikut ini. Disini anda bisa mengelola web anda dan juga melihat web anda secara online



Gambar 15.24 Halaman Setelah Login

24. Halaman kelola



Gambar 15.25 Halaman Kelola Web

Penutup

Saya sangat bersyukur dapat menyelesaikan buku ini, kemudian saya berharap buku ini akan memungkinkan Anda untuk menguasai dasar-dasar HTML, CSS dan Javascript untuk membuat situs web Anda sendiri atau untuk mengedit tampilan template atau tema di situs web.

Sekali lagi jika menurut anda ebook ini bermanfaat, mohon untuk diberikan kepada teman atau saudara agar ilmu yang terkandung dalam ebook ini dapat berkembang bagi yang membutuhkan.

DAFTAR PUSTAKA

- [1] Bernadhed, Mulia Sulistiyono (2017); WEB PROGRAMING
- [2] George Reed (2018). Web Design with HTML, CSS, JavaScript and jQuery Set. John Wiley & Sons Inc
- [3] Didik Setiawan (2017). Buku Sakti Pemrograman Web: HTML, CSS, PHP, MySQL & Javascript. Start Up
- [4] Moh. Muthohir (2021). MUDAH MEMBUAT WEB BAGI PEMULA (MENGENAL HTML, HTML5, CSS DAN JAVASCRIPT)
- [5] <https://www.w3schools.com/>
- [6] <https://www.php.net/>
- [7] <https://www.youtube.com/>
- [8] <https://id.000webhost.com/>
- [9] <https://www.petanikode.com>
- [10] <https://www.researchgate.net>
- [11] <https://www.digilib.stiestekom.ac.id>
- [12] <https://www.pakarkoding.com>
- [13] <https://www.slideshare.net>
- [14] <https://www.bootup.ai>
- [15] <https://www.scribd.com>
- [16] <https://notepad-plus-plus.org/downloads/>.
- [17] https://www.adobe.com/id_id/products/dreamweaver/free-trial-download.html
- [18] <https://www.sublimetext.com/download>
- [19] <https://code.visualstudio.com/download>
- [20] <https://www.mozilla.org/id/firefox/new/>
- [21] https://www.google.com/intl/id_id/chrome/
- [22] <https://www.microsoft.com/id-id/edge/home?form=MA13FJ>

BELAJAR PEMROGRAMAN WEB DASAR

HTML, CSS & JAVA SCRIPT
Untuk Pemula

Dendy Kurniawan, S.kom, M.kom

BIODATA PENULIS



Dendy Kurniawan, S. Kom., M. Kom adalah seorang Dosen asal Kendal Jawa Tengah kelahiran 1994. Ia saat ini menjadi Dosen Program Studi Sistem Komputer pada Fakultas Studi Akademik Universitas Stekom. Ia menyelesaikan studi strata satu (S1) di salah satu perguruan tinggi yang dahulu bernama Sekolah Tinggi Elektronika dan Komputer yang sekarang sudah beralih nama menjadi Universitas Sains dan Teknologi Komputer (Universitas Stekom) jurusan Sistem Komputer tahun 2017.

Pendidikan pasca sarjana atau Strata dua (S2) pada program Magister Sistem Informasi Fakultas Teknologi Informasi UKSW Salatiga 2019. Hasil karyanya berupa penelitian yang dipublikasikan di beberapa Jurnal Ilmiah Nasional, Jurnal Ilmiah Nasional Terakreditasi dan Jurnal Internasional seperti Journal of Technology Informatics and Engineering.



YAYASAN PRIMA AGUS TEKNIK

PENERBIT :

YAYASAN PRIMA AGUS TEKNIK

JL. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-623-8120-28-4 (PDF)



9 786238 120284