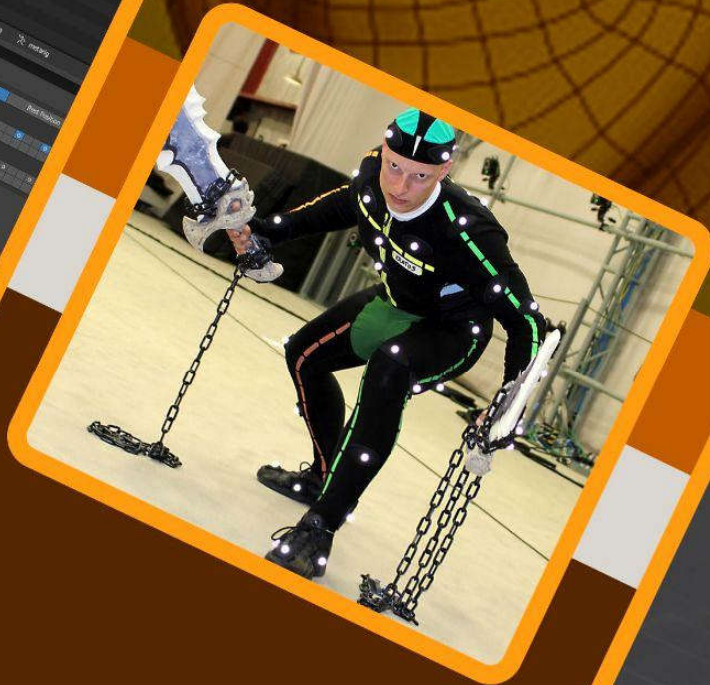
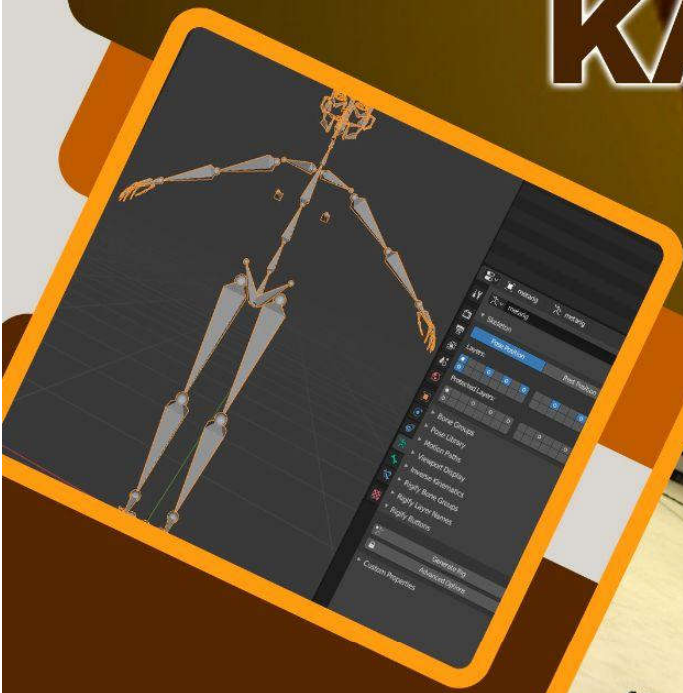




YAYASAN PRIMA AGUS TEKNIK

MOTION CAPTURE DALAM ANIMASI KARAKTER 3D



OLEH:
SETIYO ADI NUGROHO, S.E, M.KOM

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

MOTION CAPTURE DALAM ANIMASI KARAKTER 3D

SETIYO ADI NUGROHO, S.E, 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

ISBN 978-623-8120-00-0 (PDF)



MOTION CAPTURE DALAM ANIMASI KARAKTER 3D

Penulis :

Setiyo Adi Nugroho, S.E, S.Kom, M.Kom.

ISBN : 978-623-8120-00-0 (PDF)

Editor :

Andik prakasa Hadi, S.Kom., M.Kom.

Penyunting :

Daniel Rudjiono, S.Kom., M.Kom.

Desain Sampul dan Tata Letak :

Setiyo Adi Nugroho, S.E, S.Kom, 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

Distributor Tunggal :

Universitas STEKOM

Jl. Majapahit no 605 Semarang

Telp. (024) 6723456

Fax. 024-6710144

Email : info@stekom ac id

Hak cipta dilindungi undang-undang

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

KATA PENGANTAR

Penulis mengucapkan puji syukur kepada Tuhan yang Maha Esa, bahwa buku yang berjudul **“Motion Capture Dalam Animasi Karakter 3d”** telah dapat diselesaikan. Buku ini membahas tentang trend pemanfaatan metode animasi dengan penangkapan gerakan manusia asli untuk dimanfaatkan dalam gerakan animasi karakter manusia, yang disebut motion capture atau lebih sering disingkat mocap. konsep buku ini tidak hanya membahas apa itu motion capture saja saja tetapi juga membahas tentang animasi, motion capture dan penerapan motion capture dalam animasi karakter.

Pada bab 1 mengenal animasi karakter 3d yang berisikan tentang penjelasan animasi karakter 3d, animator 3D ,dan bagaimana menghidupkan animasi karakter 3d dan teknik animasinya. bab 2 menjelaskan tentang pengenalan motion capture berisi jenis-jenis motion capture dan teknik dalam pengambilan motion capture. Pada bab 3 membahas aplikasi motion capture dalam kehidupan dibidang periklanan,hiburan kesehatan dan olahraga. Pada bab 4 berisi motion capture data membahas tentang beberapa format capture yang ada serta membahas secara detail format motion capture yang sangat umum dipakai. pada bab 5 membahas sumber data motion capture seandainya membutuhkan langsung. Bab 6 membahas tentang mengedit motion capture bvh dengan bvhacker dan blender, pada bab 7 membahas 2 aplikasi bantu animasi karakter dengan motion capture yaitu makehuman dan mixamo.

Pada bab 8,9, 10 membahas penerapan motion capture ada 3 software animasi 3d yang sangat terkenal yaitu pada bab 8 penerapan motion capture pada 3ds max,bab 9 penerapan motion capture pada maya, bab 10 menggunakan motion capture dengan makehuman dan blender .

Pada bab 11 dan 12 membahas perkembangan teknologi deteksi gerakan dengan AI yaitu pada bab 11 membahas penerapan ai motion capture, dimana machine learning mengubah permainan motion capture dan penerapannya dengan deepmotion, plask.ai dan openpose bab 12 membahas proses facial motion capture dengan blender dan openpose.

Semoga buku ini dapat bermanfaat bagi para mahasiswa yang fokus pada pembuatan animasi dan game , terutama pada animasi karakter 3 dimensi

Semarang, Desember 2022

Penulis

Setiyo Adi Nugroho, S.E., S.Kom.,M.Kom

DAFTAR ISI

KATA PENGANTAR	iv
DAFTAR ISI	v
BAB 1 MENGENAL ANIMASI KARAKTER 3D	1
1.1 Animasi Karakter 3d	1
1.2 Animator 3D	2
1.3 Dua model karakter animasi 3D	3
1.3.1 Model realistis	3
1.3.2 Model stylist	3
1.4 Menghidupkan Animasi Karakter 3d	3
1.5 Teknik Animasi	6
1.5.1 Animasi Keyframed	6
1.5.1.1 Pose-to-pose	7
1.5.1.2 Lurus ke depan	7
1.5.1.3 Animasi Hibrida	8
1.5.2 Motion-Capture Animation	8
1.5.3 Animasi Prosedural	10
BAB 2 PENGENALAN MOTION CAPTURE	11
2.1 Pendahuluan	11
2.2 Jenis-Jenis Motion Capture	13
2.3 Teknik dalam pengambilan motion capture	13
2.3.1 Mechanical Motion Capture	13
2.3.2 Magnetic Motion Capture	15
2.3.3 Optical Motion Capture	16
2.3.4 Digital Armature	18
2.3.5 Pelacak Bagian Tubuh Kecil / parsial	20
BAB 3. APLIKASI MOTION CAPTURE DALAM KEHIDUPAN	22
3.1 Periklanan	22
3.2. Hiburan	22
3.3 Kesehatan	24
3.4. Olahraga	24
4.1 Pendahuluan	26
4.2. Terminologi	27

4.3. beberapa format capture yang ada	28
4.3.1. BVA dan format file BVH	28
4.3.2. Format file MNM	28
4.3.3. Format file LSHO MBX	29
4.3.4 Format MVNX	29
4.3.5. Format file ASK/SDL	29
4.3.6 Format file AOA	29
4.3.7 Format file ASF/AMC	29
4.3.8 Format file BRD	29
4.3.9 Format file GRC	30
4.3.10. Format file HTR dan GTR	30
4.3.11. Format file TAK	30
4.3.12 format file TRC	30
4.3.13 Format file CSM	30
4.3.14 Format file V/VSK	30
4.3.15 Format file GMS	31
4.3.16 Format file HDF	31
4.3.17 Format file FBX	31
4.3.18 Format file PZ2	31
4.4 membahas format motion capture yang sangat umum dipakai	32
4.4.1 Format file ASF/AMC Format	32
4.4.2 BioVision: BVH (Data Hirarki BioVision)	40
4.4.3 Format C3D	48
4.4.4 FBX	52
4.5 kesimpulan	56
BAB 5 SUMBER DATA MOTION CAPTURE	57
5.1 CMU Graphics Lab Motion Capture Database	57
5.1.1 data motion capture BVH	58
5.2 Bandai Namco	59
5.3 mixamo	60
BAB 6 MENEDIT MOTION CAPTURE BVH	62
6.1 BVHACKER	62
6.2 mengedit bvh dengan blender	70
BAB 7 APLIKASI BANTU ANIMASI KARAKTER DENGAN MOTION CAPTURE	73
7.2 Mixamo	79

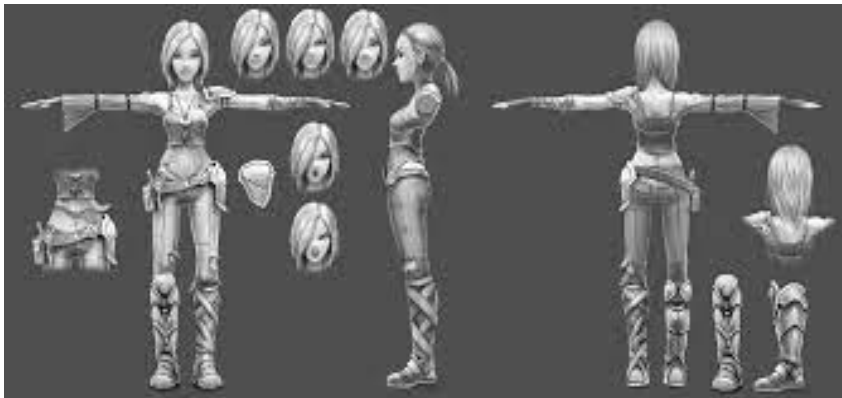
7.2.1 penerapan mixamo pada blender	80
BAB 8 PENERAPAN MOTION CAPTURE PADA 3DS MAX	103
8.1 biped	104
8.2 Membuat Biped	104
8.3 Penargetan motion capture BVH dalam 3ds Max	105
8.4 Bekerja dengan Data Motion-Capture	111
8.5 Menggunakan High-Frequency Data dan Looping	112
8.6 motion capture file yang memiliki prop data	114
BAB 9 PENERAPAN MOTION CAPTURE PADA MAYA	118
9.1 Import Motion Capture ke Maya	118
9.2 Memposing skkeleton ke model karakter	122
9.4 Mengimport Animasi Motion Capture	124
9.5 Membuat Clips Dari Motion Capture Data	126
9.6 Mengexport Clip Motion Capture Dan Digunakan Lagi Bersama Sama	127
9.7 Mengatur Arah Gerakan Pada Clip	131
9.8 Memperpanjang durasi clip	135
9.9 Mengedit Sebagian Gerakan Dari Motion Capture.	137
9.10 Menggabungkan Beberapa Motion Capture	139
BAB 10 MENGGUNAKAN MOTION CAPTURE DENGAN MAKEHUMAN DAN BLENDER	142
10.1. open source software Blender	142
10.2 Menggunakan Motion Capture pada obyek Makehuman Secara Langsung	142
10.3 Memanfaatkan Motion Capture Obyek Makehuman Pada Blender Dengan Addons	146
10.3.1 untuk versi 2.79 kebawah	146
10.3.2 pada blender 2.8 keatas	147
BAB 11 MOTION CAPTURE BERBASIS AI	152
11.1 Machine Learning: mengubah permainan Motion Capture	152
11.2 ai motion capture dengan deep motion	156
11.3 AI motion capture dengan PLASK.AI	178
11.4 Motion capture dengan OpenPose	195
bab 12 proses facial motion capture	199
12.1 facial motion capture	199
12.1 facial motion capture dengan blender saja	199
12.3 facial motion capture dengan openpose	231
Daftar Pustaka	254

BAB 1 MENGENAL ANIMASI KARAKTER 3D

1.1 Animasi Karakter 3d

Sebelum masuk ke dalam pembahasan motion capture, kita perlu memahami dulu apa yang dimaksud dengan animasi karakter 3 Dimensi. Animasi Karakter 3D adalah pembuatan model atau gambar karakter yang, Bergantung pada jenis grafik yang digunakan, animasi bisa dalam format 2D atau 3D. Animasi karakter tiga dimensi berbeda secara signifikan dari animasi 2D atau sering disebut animasi kartun , di mana animasi dicapai dengan bantuan perspektif dan lukisan karakter dan lingkungan yang sesuai.

Saat ini, teknologi memungkinkan Anda untuk segera membuat model tiga dimensi dan menunjukkan bentuknya melalui pencahayaan. Animasi 3D sendiri adalah gerakan otomatis atau transformasi objek dalam ruang dan waktu.



Sederhananya, sebelumnya saat kita membuat animasi perlu memframing pergerakan setiap karakter frame demi frame. Sekarang cukup membuat model karakter tiga dimensi, setelah itu dapat dipindahkan di ruang cloud tanpa usaha dan gambar tambahan. Namun, walaupun kedengarannya sederhana, tetapi pada kenyataannya – animasi model karakter 3D adalah proses yang cukup sulit.

Untuk membuat gambar bergerak, tidak cukup hanya memiliki akses ke komputer dan program pintar. Penting juga untuk dapat membayangkan bagaimana sang karakter dapat bergerak, gaya apa yang mempengaruhinya pada saat yang sama (bukan gaya style, tetapi, misalnya, gravitasi, gesekan, dan hambatan).

Dengan bantuan animasi 3 dimensi , karakter virtual "hidup kembali" dan mendapat kesempatan untu/k bergerak, bergerak, dan melakukan tindakan apa pun sesuai keinginan animator.



Untuk menyebutkan contoh di mana karakter animasi paling sering digunakan cukup mudah. Masing-masing dari kita mengingat setidaknya 2-3 film yang diproduksi oleh Disney Studios.



Di dalamnya, kita bisa melihat berbagai karakter kartun dalam bentuk 3d yang menyampaikan emosi, menciptakan suasana, menceritakan kisah tertentu.

1.2 Animator 3D

Dalam membuat sebuah animasi pasti dibutuhkan animator. Animator adalah orang yang menghidupkan karakter atau objek di layar. animasi terjadi ketika sekelompok gambar diam yang sedikit berbeda satu sama lain ditampilkan sedemikian rupa yaitu, dalam urutan berurutan dan

dengan kecepatan yang memadai, sehingga mata kita percaya bahwa ada sesuatu yang bergerak. Jadi animator harus mampu membuat karakter 3D bergerak frame demi frame dan sedemikian rupa sehingga penonton percaya objek 3D mati pada frame diam itu hidup. sedemikian rupa — yaitu, dalam urutan berurutan dan dengan kecepatan yang memadai — sehingga mata kita percaya ada sesuatu yang bergerak. Jadi animator harus mampu membuat karakter 3D bergerak frame demi frame dan sedemikian rupa sehingga penonton percaya objek 3D mati pada frame diam itu hidup.

Animator biasanya adalah seniman yang tidak terlalu teknis, jadi hanya diperlukan pemahaman dasar tentang perangkat lunak untuk masuk ke industri. Paket animasi 3D saat ini memiliki fitur animasi yang sebenarnya lebih sederhana dibandingkan dengan alat untuk pekerjaan animasi 3D lainnya seperti pencahayaan, pemodelan, wiring, dan VFX.

Tapi apa yang animator bisa saja kurang dalam pengetahuan teknis perangkat lunak, mereka harus menebusnya dengan pemahaman tentang kinerja akting dan keterampilan pengamatan animasi. Mereka juga membutuhkan pengetahuan dasar yang kuat tentang gerak, berat, keseimbangan, dan waktu untuk membuat penonton menahan rasa tidak percaya pada apa yang mereka lihat dan membayangkan bahwa sebuah karakter benar-benar hidup.

Menjadi mahir dalam animasi membutuhkan waktu bertahun-tahun, dan itu adalah pekerjaan yang tidak akan pernah Anda bisa kuasai sepenuhnya. Kurangnya penguasaan inilah yang menurut sebagian besar animator menarik. Anda tidak akan pernah tahu segalanya tentang perilaku manusia dan akan selalu belajar.

1.3 Dua model karakter animasi 3D

1.3.1 Model realistik

Karakter animasi komputer realistik adalah alat yang sangat diperlukan untuk membuat efek khusus, dimanapun mereka ditampilkan. Saat ini, para pengembang proses ini telah menjadi sangat terampil dan menggambarkan, misalnya, dinosaurus dan sejenisnya dengan sangat tepat, sehingga mereka terlihat seperti hidup dan sangat cocok dengan lingkungan dengan aktor manusia.

1.3.2 Model stylist

Jauh lebih kreatif dan menyenangkan untuk membuat prototipe dunia di sekitar kita – di sini seniman memiliki lebih banyak ruang untuk kreativitas. Tentu saja, animasi komputer memungkinkan Anda mereproduksi realitas secara akurat, tetapi contoh terbaiknya biasanya jauh dari kehidupan sehari-hari. Orang-orang menerima dunia seperti yang diciptakan oleh seniman animator.

1.4 Menghidupkan Animasi Karakter 3d

Salah satu persyaratan dalam membuat animasi yang memiliki karakter adalah animasi karakter harus "hidup", atau, seperti yang mereka katakan, memiliki variasi. Penonton harus percaya bahwa sang pahlawan benar-benar berjalan, berpenampilan, memiliki rambut ikal dapat tersenyum, dan berinteraksi dengan lingkungan.

Membuat dan menganimasikan karakter 3D adalah membuat cerita yang diceritakan bukan dengan kata-kata di atas kertas (atau layar), tetapi dengan gerakan. Misalnya, mimik sang karakter dapat menjelaskan banyak hal tentang karakter tersebut. Namun untuk ini, ia harus memiliki wajah 3d yang cukup "berkualitas" untuk membuat otot wajahnya bergerak.

Saat animasi 3D semakin matang, berbagai jenis animasi telah berkembang di industri video game: animasi dalam game dan sinematik game (juga disebut cut-scenes).



cutscene final fantasy 13

Animasi dalam game pada dasarnya adalah animasi siklus yang diprogram agar sesuai dengan pengontrol atau input keyboard pemain—misalnya, berlari ke depan lalu ke samping, atau melompat dan memanjat.



Tindakan dalam game membutuhkan animator untuk memiliki rasa dan keseimbangan yang baik dari objek atau karakter animasi, tetapi kebutuhan yang relatif kecil untuk kinerja halus secara keseluruhan. Animasi sinematik game, bagaimanapun, mirip dengan animasi film dan televisi, karena animasi ini terutama bergantung pada kinerja untuk melanjutkan penceritaan keseluruhan game.

Film animasi memiliki dualitas serupa, antara animasi makhluk (pergerakan makhluk hidup yang realistis atau fiksi) dan animasi pertunjukan Performance animation (yang mengkomunikasikan cerita dan aspek pertunjukan yang lebih halus). Animasi makhluk harus berdasarkan realita atau persepsi kita tentang realita, seperti yang terlihat di Jurassic Park misalnya. Kita belum pernah melihat dinosaurus bergerak tetapi dengan mempelajari makhluk hari ini kita memiliki ide bagus tentang bagaimana mereka akan bergerak.

Performance animation adalah tentang gerakan dan keseimbangan yang hebat dari objek dan karakter animasi, tetapi juga akting karakter seperti yang terlihat di sebagian besar film fitur animasi dari Pixar, DreamWorks, dan ImageWorks. Karakter ini tidak nyata dan biasanya dibuat dalam desain bergaya, tetapi karakter tersebut masih dapat membuat penonton berhubungan dan berempati dengan karakter ini melalui penampilan mereka.

Sebagian besar animator akan memilih satu area animasi dan tetap di dalamnya apakah akan menjadi animator 3d atau 2d, Tetapi ada juga beberapa animator melewati batas. Masing-masing bidang animasi ini membutuhkan keterampilan dan teknik yang berbeda, tetapi alat dan prinsip animasinya selalu sama. Animator memiliki salah satu pekerjaan tersulit dalam industri animasi 3D karena sebenarnya tidak ada jawaban benar atau salah—hanya interpretasi.

Animator harus menjadi karakter yang mereka gambarkan. Animator juga harus benar-benar memahami bidikan yang mereka kerjakan untuk dapat menceritakan kisahnya. Ini berlaku bahkan jika seorang animator sedang mengerjakan komponen adegan selain karakter itu sendiri.

Animator yang mengerjakan proyek visualisasi produk, misalnya, harus dapat menunjukkan kepada audiens apa yang perlu mereka lihat dengan cara yang bersih dan efisien. Jika palu sederhana yang memalu paku ke balok kayu adalah tugas animasi, misalnya, animator harus dapat menganimasikan kamera dan paku dan palu sedemikian rupa sehingga penonton memahami apa

yang perlu mereka ketahui. Animator harus menggunakan semua Dua Belas Prinsip Dasar Animasi untuk menyampaikan informasi ini. Kesalahan terbesar yang sering ditemui dan dibuat oleh animator muda adalah mereka berpikir bahwa bidikan dianimasikan hanya karena objeknya bergerak.

Objek atau karakter harus selalu menceritakan sebuah cerita dan menjadi hidup untuk benar-benar dianimasikan. Memindahkan objek itu mudah dilakukan dengan komputer, tetapi dibutuhkan seorang seniman untuk membuat objek tersebut bergerak dengan tujuan dan berkesan hidup.

1.5 Teknik Animasi

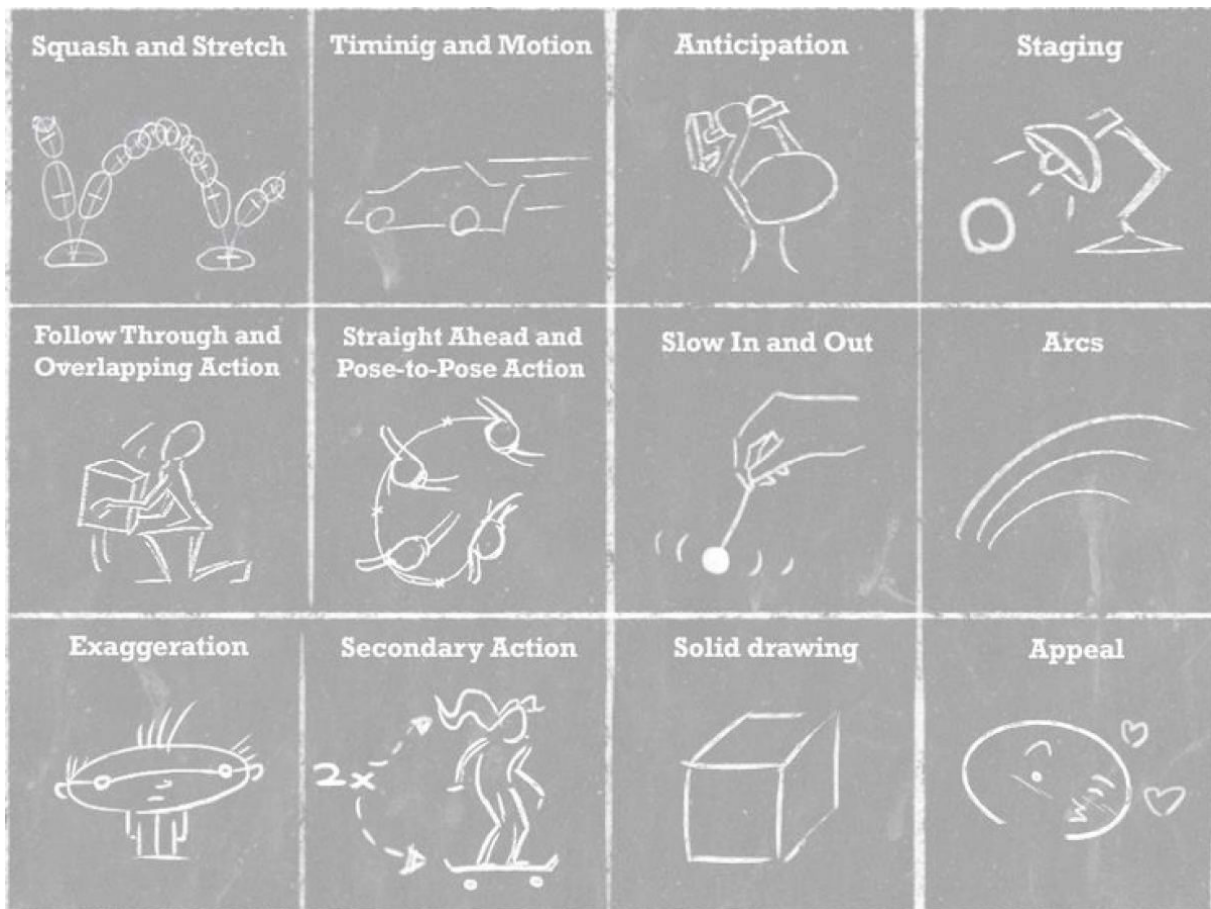
Teknik yang Anda gunakan sebagai animator akan sangat membantu dan bisa juga menghambat jenis animasi yang Anda coba buat. Banyak animator memiliki teknik favorit, sebagai contoh Eye darts adalah gerakan mata yang cepat saat memindai objek yang Anda lihat. Bagaimana cara mendapatkan referensi video makhluk yang tidak ada? Anda mencari video dari sesuatu yang akan bergerak dengan cara yang sama dan menggunakannya untuk membantu animasi Anda. shoot adalah segmen dari urutan yang lebih besar, biasanya diberikan kepada animator dengan jumlah frame tertentu. Blocking adalah penempatan pose kunci karakter dalam bidikan atau posisi kunci objek. Animator berfokus pada pengaturan waktu dasar serta komposisi. Sebuah teknik animasi bisa jadi dalam beberapa situasi teknik tertentu lebih baik daripada yang lain—tidak hanya dalam hal animasi yang dibuat, tetapi dalam hal membuat Anda menjadi animator yang lebih efisien dan lebih cepat.

Ada tiga jenis animasi 3D: yaitu animasi keyframed, motion capture dan prosedural, dan masing-masing memiliki kelebihan dan kekurangan dan pendekatan yang terpisah juga.

1.5.1 Animasi Keyframed

Animasi keyframed 3D mirip dengan animasi 2D tradisional. Animator membuat pose atau posisi untuk karakter atau objek dan menetapkan kunci pada titik waktu tertentu. Animator kemudian membuat pose atau posisi kedua dan menetapkan kunci lain pada titik waktu lain. Animasi 2D tradisional bekerja dengan cara yang sama, tetapi setiap frame akan digambar. Ini adalah tugas yang memakan waktu dan sulit dan itulah sebabnya sebagian besar studio memiliki lebih banyak animator daripada peran staf lain untuk suatu produksi. Jenis animasi ini digunakan di berbagai industri dan masih menjadi bentuk animasi 3D paling populer saat ini.

Animasi keyframe memiliki empat pendekatan yang digunakan saat ini:



12 prinsip animasi

1.5.1.1 Pose-to-pose

Pose-to-pose adalah prinsip animasi dari Dua Belas Prinsip Dasar Animasi . Teknik ini memungkinkan animator untuk meletakkan pose kunci (proses yang disebut blocking) yang menceritakan kisah pengambilan gambar dan memungkinkan animator mengerjakan aliran pengambilan gambar. Setelah pose kunci disetujui dari animator atau sutradara utama, animator akan berada di antara pose kunci hingga karakter atau objek dianimasikan sepenuhnya. Pendekatan ini populer bagi banyak animator saat ini karena memungkinkan mereka untuk menunjukkan pekerjaan mereka kepada penyelia mereka di berbagai tahap untuk disetujui sehingga hanya sedikit waktu yang terbuang untuk menghidupkan kembali bidikan. Kelemahan dari pendekatan ini adalah animator muda seringkali tidak memiliki pengalaman yang cukup dengan animasi untuk dapat membuat pose yang kuat dan memahami sepenuhnya apa yang dibutuhkan pose dalam tumpang tindih sebelum melihat aksi bergerak. Jadi beberapa waktu terbuang sia-sia, harus masuk kembali dan memperbaiki animasi yang tumpang tindih di tahap akhir animasi.

1.5.1.2 Lurus ke depan

Lurus ke depan, juga merupakan teknik animasi dari Dua Belas Prinsip Dasar Animasi asli, adalah pendekatan untuk menganimasikan setiap frame secara berurutan—frame 1, frame 2, frame

3, dan seterusnya—hingga animasi selesai. Ini adalah favorit banyak animator sejak lama karena menyediakan cara untuk menciptakan spontanitas dalam animasi; Anda bisa mendapatkan inspirasi di tengah animasi untuk menciptakan sesuatu yang baru dan menarik. Tetapi karena animasi menjadi entitas komersial, studio tidak senang dengan pendekatan ini. Merencanakan urutan atau film yang lebih besar itu sulit karena tidak ada cara untuk membuat alur total dari proyek yang besar.

Setiap animator akan menganimasikan frame sebanyak yang mereka rasa diperlukan untuk bidikan mereka. Ini berarti bahwa kadang-kadang gambar animasi terlalu panjang atau terlalu pendek untuk proyek secara keseluruhan. Juga sulit untuk mengkritik animasi karena animator akan menyelesaikan seluruh animasi untuk dilihat, dan satu-satunya cara nyata untuk memperbaiki kesalahan atau membuat perubahan adalah memulai dari awal.

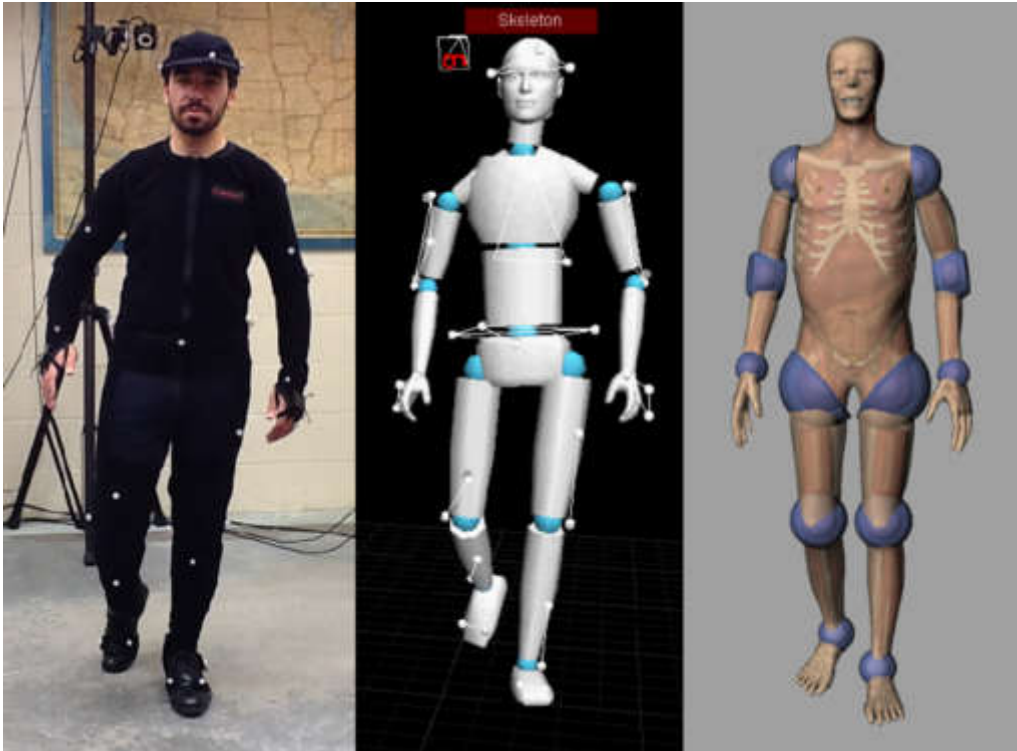
1.5.1.3 Animasi Hibrida

Animasi Hibrida adalah perpaduan antara pose-ke-pose dan lurus ke depan. Alur kerja ini menawarkan semua keuntungan dari dua alur kerja dan menghilangkan kerugiannya. Animator membuat pose kunci untuk bidikan dan biasanya membuat tahap di antaranya. Kemudian animator menggunakan alur kerja lurus ke depan untuk akhir proses untuk menambahkan tindakan tumpang tindih dan penyelesaian animasi. Hal ini memungkinkan animator untuk bekerja pada aliran dasar dan kecepatan bidikan sejak dini dan mendapatkan persetujuan dari supervisor sebelum bertindak terlalu jauh. Ini juga memungkinkan sutradara untuk memeriksa bidikan untuk memastikannya sesuai dengan keseluruhan proyek. Alur kerja ini adalah salah satu pendekatan yang paling banyak digunakan dalam animasi 3D saat ini.

hierarki Animasi adalah alur kerja yang berguna dalam membuat animasi siklus—animasi dari suatu tindakan yang berulang-ulang. Frame pertama dan terakhir dari siklus animasi harus identik untuk menciptakan efek. Berjalan adalah contoh animasi siklus yang baik dari kaki kiri, kaki kanan, kaki kiri, kaki kanan. Banyak video game menggunakan siklus untuk membuat animasi berjalan, berlari, berdiri, melompat, menembak, dan sekarat dalam game, misalnya. Tetapi karena rig animasi didasarkan pada hierarki kontrol, pendekatan ini memungkinkan animator untuk memulai dengan tubuh bagian bawah terlebih dahulu, memblokir langkah-langkah berjalan, misalnya.

Kemudian setelah tubuh bagian bawah selesai, animator dapat menganimasikan batang tubuh, lalu leher, lalu kepala. Setelah inti tubuh dianimasikan, animator menganimasikan lengan, dimulai dengan bahu, lalu berlanjut ke siku, lalu pergelangan tangan. Alur kerja ini juga merupakan cara yang baik bagi animator muda untuk memecah aksi dan tidak terjebak dengan begitu banyak kontrol dan bagian tubuh sekaligus. Animasi Jalur Animasi jalur adalah teknik yang memungkinkan animator untuk mengatur objek pada jalur dan membuat objek tersebut mengikuti jalur pada sejumlah frame. Animasi jalur memungkinkan objek tidak hanya menerjemahkan sepanjang jalur tetapi juga memutar sepanjang jalur dengan perbankan dan memutar.

1.5.2 Motion-Capture Animation



Alur kerja motion-capture cukup konsisten di industri 3 dimensi. Suatu sistem menangkap kinerja atau gerakan dan kemudian mentransfer data ke dalam paket perangkat lunak animasi 3D.

Motion capture menjadi populer di industri video game dan film karena seorang aktor dapat memerankan adegan tersebut dengan seorang sutradara, dan penampilan tersebut kemudian diterapkan pada karakter 3D.

Ada dua jenis teknologi utama yang digunakan dalam pendekatan ini, Tipe dasarnya adalah sistem penanda, di mana seorang aktor mengenakan setelan dengan set penanda yang dilacak dan ditriangulasi oleh serangkaian kamera untuk membuat kumpulan data tiga dimensi. Kumpulan data itu diterapkan pada kerangka rig karakter 3D untuk menggerakkan posisi dan rotasi masing-masing sambungan tersebut untuk menciptakan gerakan. Jenis lainnya adalah sistem tanpa penanda, di mana data dikirimkan langsung ke komputer. Setelah berada di perangkat lunak 3D, data diterapkan ke rig untuk mengevaluasi gerakan. Animator saat ini membersihkan animasi jika ada kesalahan dalam data gerak. Kemudian animator menambahkan sesuatu yang dirasa perlu. Biasanya, pergerakan bagian yang kecil seperti, tangan, jari, dan wajah tidak ditangkap dengan penangkap gerak / motion capture tetapi akan ditambahkan oleh animator dalam proses pembuatan animasinya. Namun, seiring kemajuan teknologi, animasi wajah dan jari ditambahkan ke tangkapan motion capture agar tetap mendapatkan data aslinya. Banyak opsi perangkat lunak tersedia, tetapi alur kerja intinya selalu sama.

Menggunakan Motion Capture Motion capture, atau mocap, adalah teknik merekam gerakan orang sungguhan sehingga dapat diterapkan pada karakter digital. Kebanyakan orang saat ini cenderung mengasosiasikan motion capture dengan industri film dan video game. Kita semua pernah melihat cuplikan di balik layar yang menunjukkan aktor menyelesaikan sesi motion-capture. Namun motion capture digunakan di banyak bidang lain—termasuk militer, kedokteran, dan biomekanik—karena berbagai alasan.

pemanfaatan motion capture ini sangat penting dalam Performa Gerak Real-Time. Performa gerak real-time adalah tempat industri animasi 3D ingin melihat teknologi motion capture ini. Itu berarti menangkap sepenuhnya penampilan seorang aktor hanya dalam satu pengambilan — gerakan tubuh, bentuk wajah, dan semua pelengkap termasuk jari tangan dan kaki. Saat ini, sudah ada teknologi yang dapat menangkap semua bagian dari pemain manusia ini, tetapi sebagian besar harus menangkapnya secara terpisah. Sebagian besar sistem saat ini dapat menangkap sebagian besar kinerja dalam sekali pengambilan, tetapi penangkapan masih belum selengkap yang diinginkan oleh industri animasi 3D.

Film Avatar dan video game LA Noire sama-sama mendorong batas penangkapan kinerja gerak di industrinya masing-masing. Avatar menciptakan banyak teknik penangkapan real-time baru yang memungkinkan sutradara James Cameron untuk melihat model CG tingkat proxy yang bergerak secara real time di monitor saat aktor manusia berada di lokasi pengambilan. Seniman digital di Avatar juga mampu menangkap gerak wajah aktor nyata tepat waktu dengan menangkap gerak pertunjukan tubuh dengan tingkat detail yang tinggi, yang belum pernah menjadi alur kerja langsung sebelumnya. LA Noire, dikembangkan untuk konsol oleh Team Bondi dan diterbitkan oleh Rockstar Games, menciptakan sistem motion capture baru yang disebut MotionScan yang memungkinkan sejumlah besar detail dalam animasi wajah yang hingga saat ini tetap tak tertandingi. Sistem ini memungkinkan Anda membuat 3D mesh deforming secara real time dengan menggunakan banyak kamera HD yang dikalibrasi ke area tertentu.

1.5.3 Animasi Prosedural

Dalam animasi prosedural, seorang animator tidak ada hubungannya dengan gerakan. Perangkat lunak menciptakan gerakan, dan animator hanya menambahkan parameter untuk membuat gerakan merespons dengan cara tertentu. Animasi prosedural sangat baru di tingkat konsumen animasi 3D dan semakin banyak digunakan di industri video game.

Jenis animasi ini memungkinkan karakter latar belakang atau lawan yang dikendalikan komputer untuk bereaksi terhadap lingkungan dan tindakan dengan cara yang realistis dan meyakinkan. Ini adalah salah satu jenis animasi terbaru dan masih dilihat oleh berbagai industri saat ini. Video game sangat tertarik dengan jenis ini karena dapat sangat meningkatkan produktivitas jalur produksi dengan membuat komputer menganimasikan karakter secara real time untuk bereaksi sesuai dengan apa yang terjadi di dalam game. Pada awalnya caranya adalah meminta animator menganimasikan banyak hasil berbeda yang mungkin terjadi selama permainan, sekarang sudah banyak digantikan perannya oleh komputer.

BAB 2 PENGENALAN MOTION CAPTURE

2.1 Pendahuluan

animasi memberikan langkah pertama di awal abad ke-20, ketika pada tahun 1911, kartunis Winsor McCay drew karakter dalam beberapa lembar kertas dengan sedikit perubahan antara ini dan kemudian sampel mereka pada tingkat yang konstan untuk menciptakan ilusi gerak.



Winsor McCay little nemo

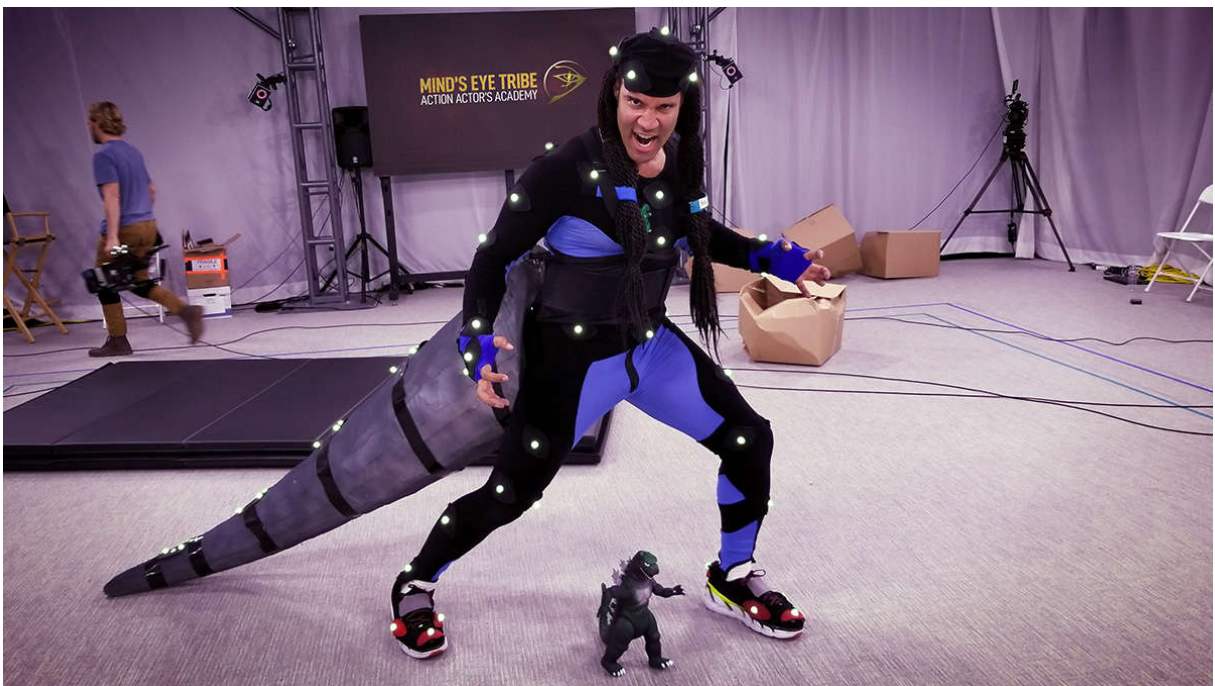
Proses animasi tidak mengalami inovasi yang cukup besar sampai komputer mulai mengambil tempat dalam proses pembuatan. Dengan lahirnya keyframing, yang mengurangi jumlah sampel yang dibutuhkan untuk membuat animasi, animator melihat pekerjaan mereka jauh lebih disederhanakan. Proses ini memakan waktu karena, pada saat itu, setiap seniman dipaksa untuk secara individual menganimasikan setiap pose/frame.

Dengan diperkenalkannya keyframing, artis menentukan frame awal dan akhir dari animasi dan frame perantara dari gerakan yang dihasilkan secara otomatis. Namun beberapa animasi masih tidak mungkin untuk dibuat ulang karena kompleksitas bawaannya, misalnya animasi berjalan manusia, yang sangat rumit karena artikulasi kita.



Untuk mempercepat proses animasi lebih jauh, diciptakanlah motion capture, sebuah sarana dimana kita menangkap pergerakan objek di dunia nyata dan kemudian memasukkan data pergerakan yang ditangkap dalam model dunia tiga dimensi dalam lingkungan virtual.

Proses motion capture pertama berkembang dengan sistem mekanis yang cukup rumit dan membatasi jumlah kebebasan yang dapat dialami aktor, sangat membatasi spektrum animasi yang dapat ditangkap. Ini terjadi terutama karena ini adalah sistem mekanis yang menggunakan setelan yang sangat ketat dan sejumlah besar kabel yang menghalangi pergerakan aktor. Mereka termasuk sistem akustik, mekanik, optik dan magnetik, selanjutnya dibagi dalam sistem penanda dan tanpa penanda.



Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

Saat ini, motion capture banyak digunakan dalam industri game, film, dan animasi sebagai sarana untuk menyediakan animasi tubuh dan/atau wajah yang cepat dan hemat biaya untuk menganimasikan satu atau berbagai karakter.

2.2 Jenis-Jenis Motion Capture

Sistem motion capture manusia diklasifikasikan sebagai sistem outside-in, inside-out, dan inside-in systems.. Nama-nama ini menunjukkan di mana sumber tangkapan dan sensor ditempatkan.

- Sistem outside-in menggunakan sensor eksternal untuk mengumpulkan data dari sumber yang ditempatkan pada tubuh. Contoh sistem tersebut adalah perangkat pelacakan berbasis kamera, di mana kamera adalah sensornya dan penanda reflektif adalah sumbernya.
- Sistem inside-out, memiliki sensor yang ditempatkan pada tubuh yang mengumpulkan eksternal sumber. Sistem elektromagnetik, yang sensornya bergerak dalam medan elektromagnetik yang dihasilkan secara eksternal, adalah contoh sistem inside-out.
- Sistem inside-in memiliki sumber dan sensor yang ditempatkan pada tubuh. Contoh dari perangkat ini adalah setelan elektromekanis, di mana sensornya adalah potensiometer atau goniometer dan sumbernya adalah sendi sebenarnya di dalam tubuh. Teknologi utama yang digunakan saat ini yang mewakili kategori ini adalah optik, elektromagnetik, dan sistem pelacakan manusia elektromekanis.

2.3 Teknik dalam pengambilan motion capture

Ada beberapa teknik berbeda dalam teknologi motion capture beberapa teknik yang sering digunakan adalah Mechanical motion capture, Magnetic motion capture, optical motion capture

2.3.1 Mechanical Motion Capture



Teknik motion capture ini dicapai melalui penggunaan exoskeleton. Setiap sambungan kemudian dihubungkan ke encoder sudut. Nilai pergerakan setiap encoder (rotasi) dicatat oleh komputer sehingga dengan mengetahui posisi relatif encoder (sendi) dapat membangun kembali gerakan tersebut di layar menggunakan perangkat lunak. Offset diterapkan pada setiap encoder, karena sangat sulit untuk secara tepat mencocokkan posisi mereka dengan hubungan sebenarnya.



Keuntungan dari Mechanical Motion Capture

- Rentang tangkapan bisa sangat besar.
- baju elektromekanis lebih murah daripada sistem optik dan magnetik.
- bajunya portabel.
- Pengumpulan data real time dimungkinkan.
- Data tidak mahal untuk ditangkap.
- Sensor tidak terkena gangguan.
- Dimungkinkan untuk menangkap beberapa pemain secara bersamaan dengan banyak pengaturan.

Kekurangan Mechanical Motion Capture

- Sistem memiliki sampling rate yang rendah.
- Mereka terasa menonjol karena jumlah perangkat keras.
- Sistem menerapkan batasan pada sendi manusia.
- Konfigurasi sensor susah diperbaiki.
- Sebagian besar sistem tidak menghitung terjemahan global tanpa sensor magnetik.

2.3.2 Magnetic Motion Capture



Magnetic motion capture dilakukan melalui medan elektro-Magnet yang diperkenalkan dimana sensor berupa kumparan sensor listrik, sensor direpresentasikan pada tanda tempat dalam 3 sumbu x,y,z . Untuk mengetahui posisinya pada bidang penangkapan gangguan yang dibuat oleh sensor melalui antena maka kita dapat mengetahui orientasinya.



Keuntungan dari Pelacak Magnetik

- Output data waktu nyata dapat memberikan umpan balik langsung.

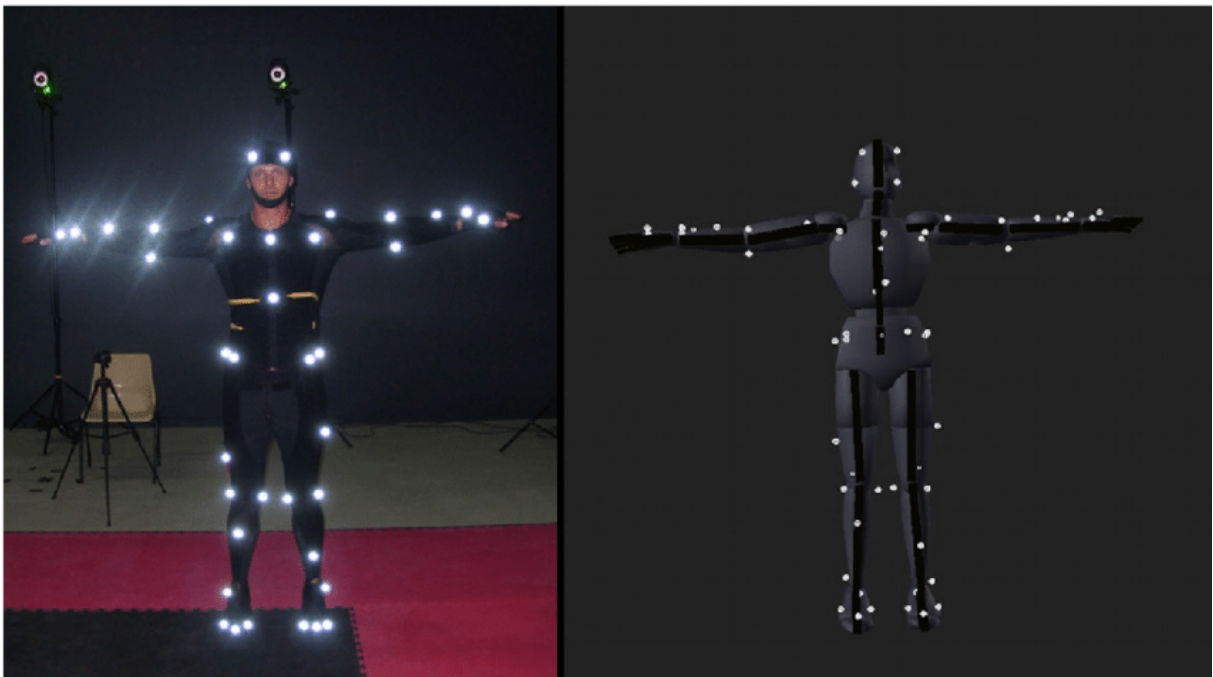
Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

- Data posisi dan orientasi tersedia tanpa postprocessing.
- Pelacak magnetik lebih murah daripada sistem optik.
- Sensor tidak pernah terganggu.
- Dimungkinkan untuk menangkap beberapa pemain yang berinteraksi secara bersamaan dengan beberapa pengaturan.

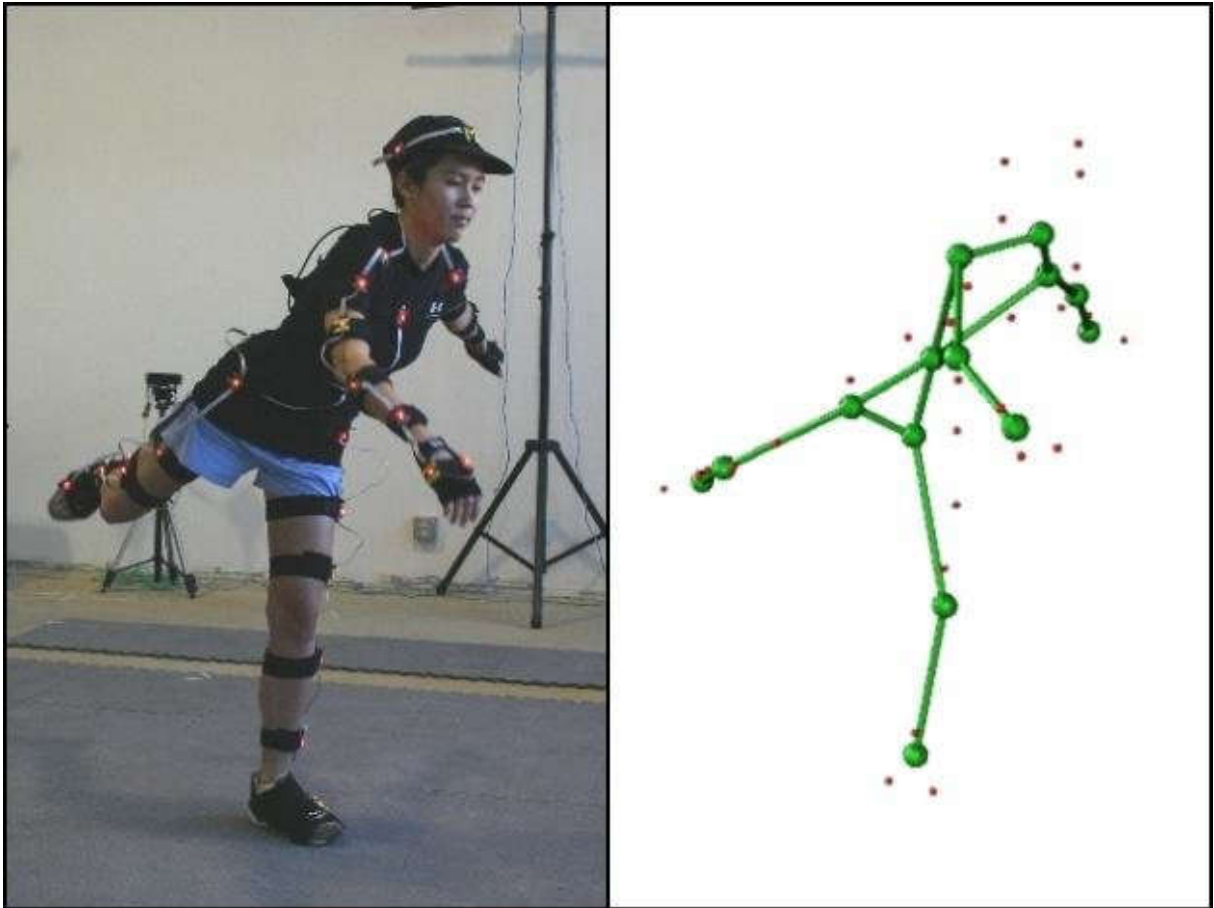
Kekurangan Pelacak Magnetik

- Sensitivitas pelacak terhadap logam dapat menghasilkan keluaran yang tidak teratur.
- Aktor dibatasi oleh kabel dalam banyak kasus.
- Pelacak magnetik memiliki laju pengambilan sampel yang lebih rendah daripada beberapa sistem optik.
- Area pengambilan lebih kecil daripada yang dimungkinkan dengan sistem optik.
- Sulit untuk mengubah konfigurasi marker.

2.3.3 Optical Motion Capture



Penangkapan didasarkan pada pemotretan optik beberapa kamera yang disinkronkan, sintesis koordinat (x, y) dari objek yang sama dari sudut yang berbeda memungkinkan untuk menyimpulkan koordinat (x, y, z) . Metode ini melibatkan pertimbangan masalah kompleks seperti paralaks optik, distorsi lensa yang digunakan, dll. Dengan demikian, sinyal mengalami banyak interpolasi. Namun kalibrasi yang benar dari parameter ini akan membantu akurasi tinggi dari data yang dikumpulkan.



Prinsip pengoperasiannya mirip dengan radar: kamera memancarkan radiasi, biasanya inframerah, dipantulkan oleh penanda dan kemudian dikembalikan ke kamera yang sama. Memeriksa informasi masing-masing kamera (minimal dua kamera) untuk menentukan posisi marker di ruang virtual.

Keuntungan Sistem Optik

- Data optik sangat akurat dalam banyak kasus.
- Penanda dalam jumlah yang lebih besar dapat digunakan.
- Mudah untuk mengubah konfigurasi penanda.
- Pelaku tidak dibatasi oleh kabel.
- Sistem optik memungkinkan area kinerja yang lebih besar daripada kebanyakan sistem lainnya.
- Sistem optik memiliki frekuensi penangkapan yang lebih tinggi, menghasilkan lebih banyak sampel per detik.

Kelemahan Sistem Optik

- Data optik memerlukan pemrosesan akhir yang ekstensif.
- Perangkat kerasnya mahal, antara \$100.000 dan \$250.000.
- Sistem optik tidak dapat menangkap gerakan saat marker tertutup dalam waktu lama/ periode waktu tertentu.

- Penangkapan harus dilakukan di lingkungan yang terkendali, jauh dari warna gangguan cahaya dan reflektif.

2.3.4 Digital Armature

Armature digital dapat diklasifikasikan menjadi dua jenis:

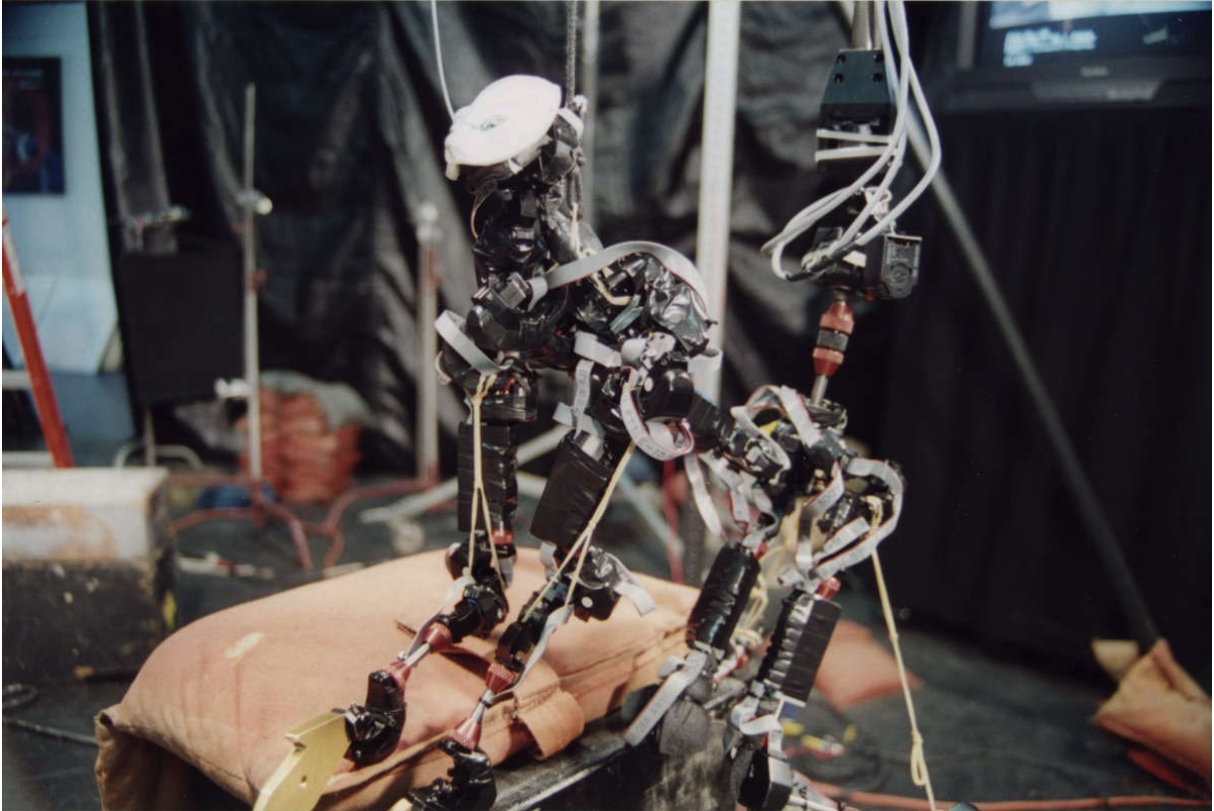
- armature keyframing atau stop-motion
- armature real-time atau puppeteering.

Seperti exoskeleton mekanis, kedua jenis motion capture terdiri dari serangkaian modul kaku yang dihubungkan oleh sambungan yang rotasinya diukur dengan potensiometer atau sensor sudut. Sensor biasanya adalah perangkat analog, tetapi disebut "digital" karena hasil pembacaan diubah menjadi sinyal digital untuk diproses oleh sistem komputer. Angker ini biasanya modular untuk mengakomodasi desain karakter yang berbeda.

Armature keyframing awalnya digunakan untuk membantu animator stop-motion menganimasikan karakter digital; alat ini kadang tidak benar-benar dianggap sebagai sistem motion capture karena tidak digerakkan oleh pemain secara live. disini dibahas sebagai motion capture karena sebagian besar armature yang tersedia secara komersial juga dapat digunakan sebagai armature real-time.

Beberapa alat ini dapat memiliki tujuan ganda, seperti perangkat yang awalnya disebut Dinosaur Input Device (DID), yang dirancang oleh Craig Hayes di Tippett Studio di Berkeley, California. Nama ini diambil karena unit ini digunakan untuk menganimasikan beberapa dinosaurus digital di Jurassic Park. Belakangan, perangkat itu digunakan untuk menganimasikan bug di Starship Troopers. Sekarang disebut Perangkat Input Digital.

Konsep dasar di balik armature keyframing adalah bahwa animator memposisikan perangkat secara manual untuk menghasilkan setiap keyframe dalam animasi. Karakter dalam perangkat lunak animasi diatur dengan struktur mekanis yang setara dengan angker. Dengan menekan sebuah tombol, animator menggunakan komputer untuk merekam pose angker ke dalam karakter digital untuk frame waktu tertentu. Ini dilakukan melalui program driver yang menghubungkan perangkat, biasanya dicolokkan ke port serial, ke perangkat lunak animasi. Setelah semua pose kunci direkam, perangkat lunak memperlakukannya sebagai frame utama biasa sementara animasi antar frame utamain dibuat dengan perangkat lunak itu sendiri.



Amatur puppeteering sangat mirip dengan armatur keyframing, kecuali gerakannya ditangkap secara real time seperti yang dilakukan oleh satu atau lebih dalang. Contoh dari penyiapan semacam itu adalah angker berpemilik yang dikembangkan oleh Boss Film Studios untuk menangkap gerakan Sil, karakter alien yang dirancang HR Giger untuk film *Species*. Tidak banyak armatur yang tersedia secara komersial, sehingga sebagian besar perusahaan produksi harus merancangya sendiri.

Keunggulan Digital Armatures

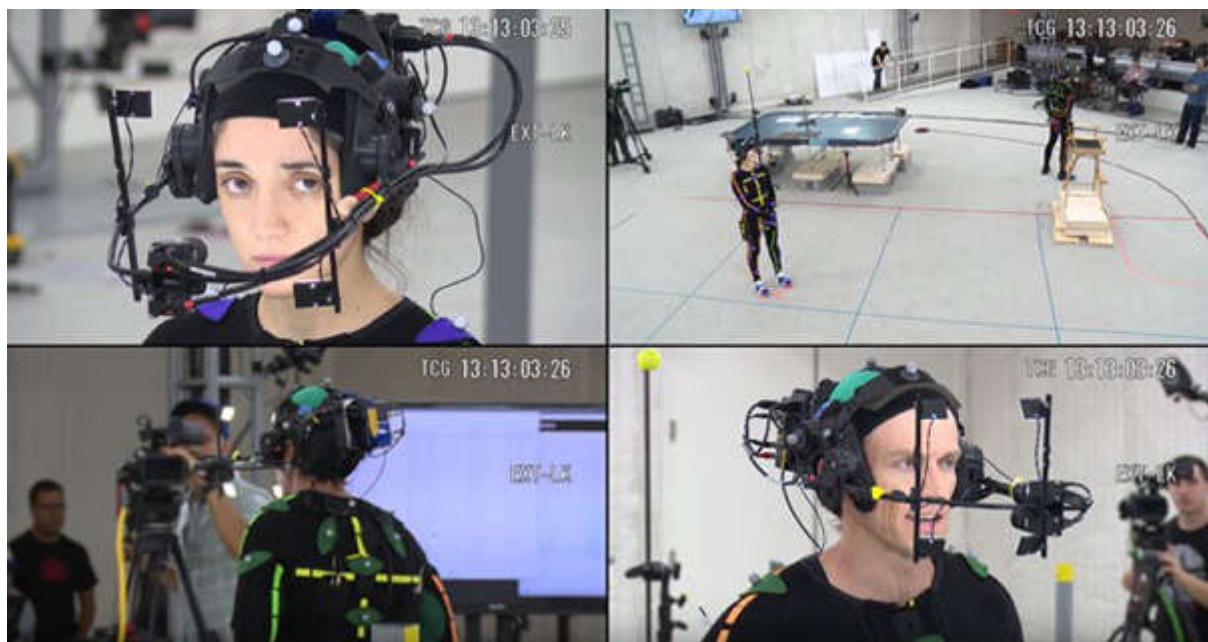
- Digital armature mudah digunakan untuk beberapa karakter yang dianimasikan menggunakan stop motion.
- Pengaturan untuk jenis karakter yang berbeda mudah dilakukan.
- Biayanya rendah.
- Angker adalah perangkat input digital yang baik untuk animator stop-motion.
- Data segera tersedia.

Kelemahan Digital Armatures

- Terjemahan global biasanya tidak ditangkap.
- Pada kebanyakan armatur, semua sambungan adalah sambungan engsel.
- Armature real-time memiliki laju pengambilan sampel yang rendah dibandingkan dengan

2.3.5 Pelacak Bagian Tubuh Kecil / parsial

Mengingat dalam tubuh kita banyak bagian yang kecil dan memiliki gerakan yang sangat detail, Untuk menangkap gerakan bagian tubuh kecil/ parsial , diperlukan pelacak khusus, karena perangkat pelacakan seluruh tubuh tidak memiliki resolusi yang diperlukan. Namun, ada beberapa pengecualian. Misalnya, praktis tidak mungkin melacak gerakan wajah dan gerakan jari dengan pelacak elektromagnetik karena sensornya lebih besar daripada lokasi penempatannya. Dimungkinkan untuk menangani situasi seperti itu dengan pelacak optik dalam beberapa kasus. Ada beberapa perangkat mekanis dan optik untuk menangkap gerakan wajah.



Yang paling populer adalah pelacak wajah optik real-time, yang terdiri dari kamera yang ditempatkan di struktur yang dipasang di kepala pemain, sehingga bergerak bersama pemain. Perangkat menangkap gerakan penanda kecil yang ditempatkan di berbagai area wajah. kitangnya, ini adalah perangkat 2D yang tidak dapat menangkap gerakan tertentu seperti kerutan bibir, sehingga semua data berada dalam satu bidang dan tidak terlalu realistis. Data gerakan wajah tiga dimensi dapat ditangkap dengan sistem optik menggunakan dua atau tiga kamera, menghasilkan hasil yang jauh lebih baik, tetapi tidak secara real time.



Untuk gerakan tangan, ada beberapa jenis sarung tangan yang memiliki teknologi faktor bentuk kecil. Model 18 sensor mengukur sebagian besar rotasi jari, penculikan (sudut antara jari yang berdekatan), persilangan ibu jari, lengkungan telapak tangan, dan rotasi pergelangan tangan.

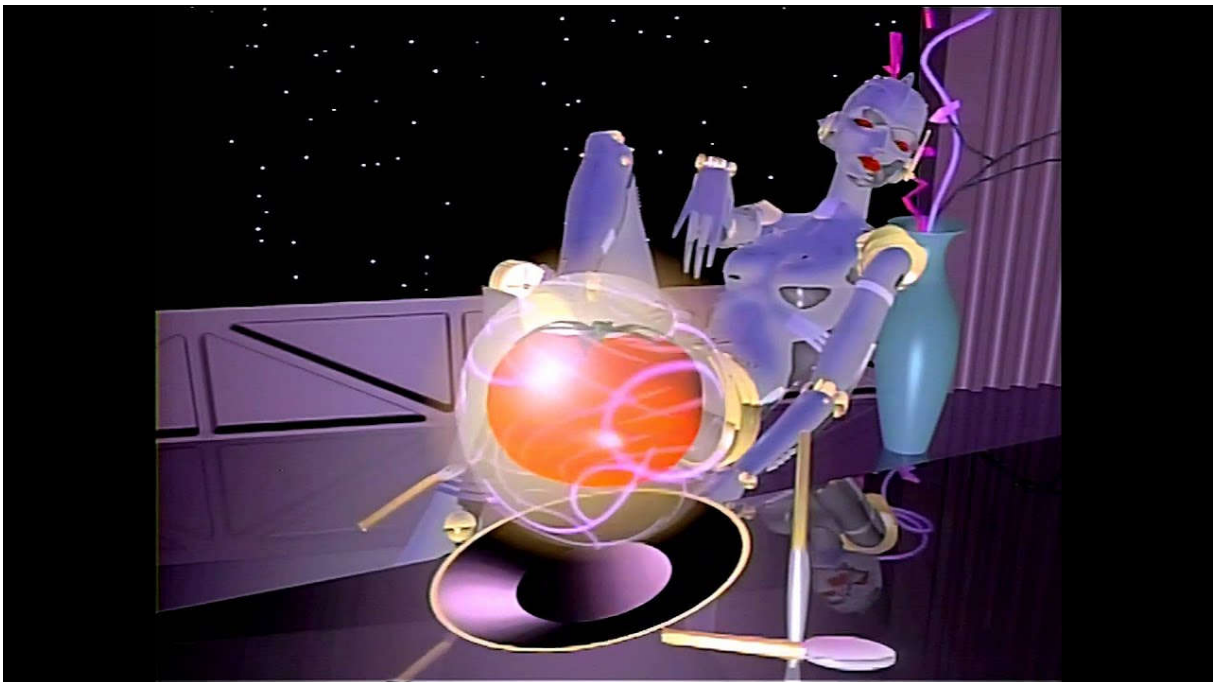
Model 22 sensor menambahkan sensor untuk sambungan distal jari telunjuk, tengah, manis, dan kelingking. Sensor dapat menangkap rotasi minimum $0,5^\circ$ dan dapat bekerja hingga sekitar 110 sampel per detik. Seperti banyak perangkat serupa lainnya, ini membutuhkan perangkat ketiga untuk mengukur posisi dan orientasi global.

BAB 3. APLIKASI MOTION CAPTURE DALAM KEHIDUPAN

Banyak yang menganggap bahwa motion capture ini hanya digunakan pada animasi dan game , tapi sebenarnya penerapannya tidak hanya dibidang itu saja. banyak bidang yang dapat memanfaatkan teknologi ini dan meskipun dunia hiburan adalah yang paling banyak mendapatkan manfaat, penerapan motion capture ini tidak hanya pada industri hiburan saja. berikut ini penjelasan tentang penerapan motion capture di berbagai bidang.

3.1 Periklanan

Ada banyak area yang dapat diuntungkan dari penggunaan motion capture. Pada tahun 1984, animasi pertama karakter yang menggunakan teknik ini dibuat dalam aksi publisitas yang ditugaskan oleh asosiasi produsen makanan kaleng terbesar di Amerika Serikat. Animasi berubah menjadi iklan yang disebut Brilliance, atau Robot Seksi, yang disiarkan di Kejuaraan Super Bowl, pada tahun 1985.



Brilliance, atau Robot Seksi

Penggunaan teknologi ini, meskipun mengesankan, bahkan membutuhkan waktu untuk digunakan secara luas di bidang komunikasi, mendapatkan kekuatan di akhir 90-an, ketika teknologi lebih efisien dan mudah diakses.

3.2 Hiburan

Antara pertengahan hingga akhir 80-an dan awal 90-an, teknologi ini ditingkatkan dan diterapkan lebih sering pada proyek-proyek hiburan, yang banyak digunakan dalam animasi real-time

untuk televisi — pada tahun 1991, seorang produser Prancis bernama Media lab, dikembangkan untuk acara TV karakter "Mat, hantu", yang disiarkan setiap hari selama lebih dari tiga tahun dalam penampilan satu menit. Pada tahun yang sama film Terminator II diluncurkan, menggunakan teknik untuk menganimasikan karakter T-1000, sebuah robot yang terbuat dari logam cair.



Meskipun ini bukan penggunaan pertama dari teknik dalam film, ini adalah penggunaan pertama dari motion capture pada karakter yang relevan. Saat ini, bioskop masih menjadi salah satu industri terbesar yang menggunakan teknik ini. Video game juga memiliki peran representatif dalam menggunakan motion capture. Pada tahun 90-an media ini bertanggung jawab atas sebagian besar penggunaan motion capture untuk animasi, mempertahankan penggunaannya saat ini untuk hiburan. Bahkan saat ini, banyak game teknis dan artistik yang sangat berkualitas menggunakan motion capture, yang juga digunakan di konsol game digital seperti Nintendo Wii, Xbox dan Playstation EyeToy, membawa gerakan pemain ke dalam game.

3.3 Kesehatan

Bidang ini sebagian besar diuntungkan oleh proses motion capture. Salah satu bidang utama yang dilayani oleh teknologi ini adalah analisis biomekanik, yang dapat digunakan untuk memverifikasi kondisi seseorang yang membutuhkan prostesis.



Motion capture dapat menganalisis bagaimana seseorang melakukan gerakannya dan di mana ia mengerahkan lebih banyak kekuatan untuk mengidentifikasi titik-titik di tubuh di mana beban kerja terjadi, sehingga memberikan data untuk konstruksi prostesis khusus untuk setiap orang dan analisis kemajuan yang belum pernah terjadi sebelumnya dari pekerjaan rehabilitasi.

3.4. Olahraga

Motion capture dapat digunakan untuk menganalisis kinerja atlet, memberikan informasi tentang cara orang bergerak; memungkinkan mereka untuk meningkatkan kinerja mereka dengan mengenali kesenjangannya. Penangkapan bekerja dalam situasi tertentu, menganalisis atlet tertentu, dan dalam aspek umum, menganalisis data kinerja semua atlet yang terlibat dalam pertandingan.



Sistem Mocap, telah berevolusi dari sistem yang sederhana, sangat terbatas, tidak ramah pengguna (dari segi perangkat lunak), hingga yang sangat mobile dan khusus. Jenis sistem yang dibahas dengan jelas semuanya memiliki skenario kasus optimal untuk penerapannya.

Namun, mocap optik telah berkembang lebih dari sistem saudaranya. Hal ini, sebagian besar, karena kemampuannya untuk beradaptasi dengan sangat baik dengan persyaratan utama industri film dan video game, yang telah berinvestasi dan dengan demikian membantu pengembangan teknologi ini. Terlepas dari kelebihanannya, masih ada perbaikan yang dapat dilakukan di bidang ini (beberapa di antaranya diusulkan di bagian Penelitian Masa Depan).

Selain perbaikan ini, beberapa sistem mocap masih sangat terbatas dalam hal area di mana mereka dapat menangkap gerakan, terbatas pada gudang atau studio yang disesuaikan. Sistem ini juga memiliki anggaran yang sangat tinggi, yang, dalam beberapa kasus, mengesampingkannya. Mereka bisa mendapatkan keuntungan dari versi anggaran yang lebih rendah, lebih mudah diakses oleh publik dan perusahaan kecil.

4.1 Pendahuluan

Animasi karakter tradisional adalah proses yang melibatkan banyak waktu, menggunakan seniman terampil untuk secara manual berpose karakter menggunakan alat seperti 3D Studio Max, Maya atau Blender. Namun dengan tuntutan yang semakin meningkat ditempatkan pada grafik komputer untuk tampilan 3D yang lebih baik dan dalam waktu yang lebih singkat, diperlukan teknik animasi karakter alternatif. Sementara penggunaan forward dan backward kinematics dapat digunakan untuk mempercepat proses penentuan posisi karakter tradisional, fase pembuatan keseluruhan masih relatif lambat.

Oleh karena itu, untuk memenuhi tuntutan animasi karakter modern banyak rumah produksi telah beralih ke teknologi motion capture, dalam upaya untuk menemukan solusi yang lebih baik. Dengan bantuan teknik motion capture banyak konfigurasi postur yang melelahkan dihilangkan sebagai animasi karakter. direkam langsung dari aktor yang melakukan gerakan yang diinginkan, sehingga mengurangi kebutuhan seniman untuk memposisikan secara manual karakter.

Perangkat penangkap gerak / motion capture/ mocap memungkinkan perekaman gerakan langsung dengan melacak sejumlah titik kunci dalam ruang Seiring waktu, yang diterjemahkan ke dalam representasi digital 3 dimensi. Subjek yang ditangkap dapat berupa apa saja yang ada di dunia nyata, dengan titik-titik kunci yang diposisikan pada objek sedemikian rupa sehingga mewakili orientasi bagian objek yang bergerak, misalnya sendi atau titik pivot. Untuk melakukan triangulasi posisi penanda secara akurat, setidaknya digunakan 4 kamera, namun umumnya tidak lebih dari 32 kamera yang digunakan.

Keberhasilan motion capture telah melahirkan sejumlah rumah produksi yang dapat merekam dan menyediakan data gerak”, namun banyak perusahaan telah mengembangkan format file mereka sendiri. Ini berarti bahwa format file data motion capture jauh dari standar, namun sifat ASCII dari banyak format membuatnya cukup mudah untuk didekode dan dipahami dengan pemeriksaan data yang sederhana.

Bergantung pada sistem yang Anda gunakan, Anda harus berurusan dengan beberapa format file motion capture yang berbeda. Sistem real-time biasanya menghasilkan aliran data yang dapat digunakan langsung dalam perangkat lunak animasi dengan bantuan plug-in khusus. Data sistem optik tidak dihasilkan secara real time, sehingga tidak dapat dialirkan ke perangkat lunak animasi; sebaliknya, itu diimpor dari file data. Paket seperti Softimage, Maya, dan 3D Studio Max menyertakan konverter yang memungkinkan Anda mengimpor file data dalam beberapa format file,

tetapi saat melakukan penyiapan karakter, penting untuk mengetahui item data apa yang terdapat dalam file ini.

File data sistem optik melewati beberapa tahap sebelum menjadi final. Saat data dikumpulkan, file aliran video dibuat untuk masing-masing kamera sistem, berisi gambar mentah yang diambil. Ini disebut data mentah. File-file ini, dikombinasikan dengan file yang berisi kalibrasi panggung, digunakan untuk melacak data tiga dimensi.

Tahap kedua adalah file yang dihasilkan setelah pelacakan, yang berisi koordinat Cartesian dari masing-masing penanda. Data dalam file ini disebut data terjemahan global; itu mewakili posisi setiap penanda dalam referensi ke world, tanpa menyertakan definisi hierarki atau kerangka apa pun.

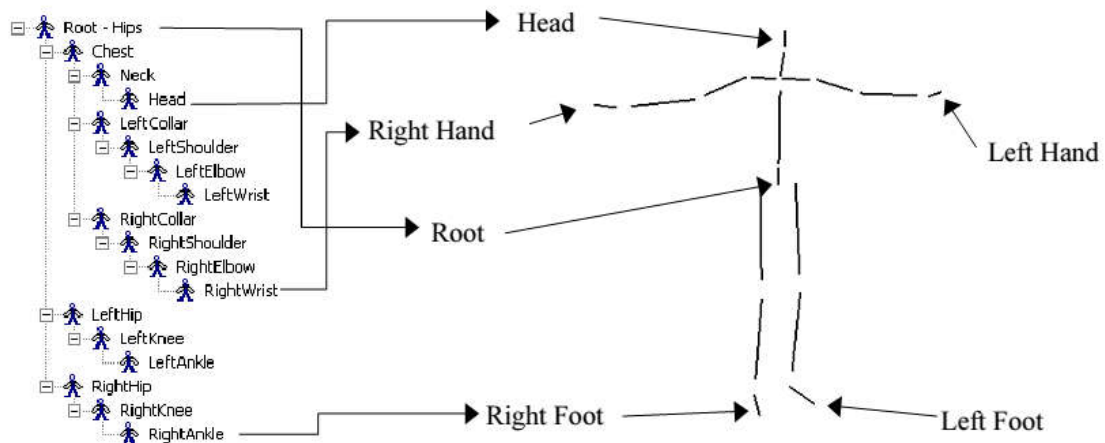
4.2. Terminologi

Bagian ini memberikan tinjauan singkat tentang terminologi dan gaya notasi yang akan digunakan untuk menjelaskan proses yang terlibat dalam membaca dan memproses data motion capture.

Daftar berikut menguraikan beberapa kata kunci yang lebih penting yang akan digunakan untuk mengidentifikasi dan menjelaskan aspek-aspek yang berbeda dari suatu gerakan:

- Skeleton — Seluruh karakter yang diwakili oleh gerakan tersebut.
- bone — Entitas dasar dalam mewakili kerangka. Setiap tulang mewakili segmen terkecil dalam gerakan yang tunduk pada terjemahan individu dan perubahan orientasi selama animasi. Kerangka terdiri dari sejumlah tulang (biasanya dalam struktur hierarkis, seperti yang diilustrasikan pada gambar 2.1), di mana setiap tulang dapat dikaitkan dengan jaring simpul untuk mewakili bagian tertentu dari karakter, misalnya tulang paha atau humerus.
- channel atau Derajat Kebebasan (DOF) — Setiap tulang dalam kerangka dapat mengalami perubahan posisi, Orientasi, dan skala selama animasi, di mana setiap parameter dirujuk ke channel asa (atau DOF). Perubahan data channel dari waktu ke waktu menimbulkan animasi.
- Frame — Setiap animasi terdiri dari sejumlah frame dimana untuk setiap frame data channel untuk setiap tulang didefinisikan. Data motion capture dapat ditangkap setinggi 240 frame per detik, namun dalam banyak aplikasi kecepatan 30 atau 60 frame per detik cenderung menjadi norma.

Frame rate tinggi digunakan untuk menangkap gerakan yang mengandung konten frekuensi tinggi seperti kombinasi aksi karate. Meskipun dalam banyak kasus detail tambahan tidak dapat ditampilkan selama pemutaran waktu nyata karena kecepatan refresh maksimum perangkat keras tampilan, ini dapat memberikan informasi yang berguna untuk menambahkan gerakan kabur ke animasi atau hanya untuk analisis gerakan.



Gambar Struktur Hirarki untuk Manusia

Kecepatan refresh monitor harian yang terjangkau saat ini maksimal sekitar 100 hz , dan kecepatan 60fps berkelanjutan dalam game komputer modern dianggap sebagai pencapaian yang sangat baik.

4.3. beberapa format capture yang ada

Ada banyak format yang digunakan dalam motion capture antara lain

4.3.1. BVA dan format file BVH

BVH adalah singkatan dari Biovision Hierarchical Data, yang dikembangkan oleh perusahaan penangkap gerak / motion capture / motion capture bernama Biovision. Format BVA (juga dikembangkan oleh Biovision) adalah format lama yang merupakan pendahulu BVH. Format BVH terutama digunakan sebagai representasi standar gerakan dalam animasi struktur humanoid. Saat ini salah satu format data gerak paling populer dan telah diadopsi secara luas oleh komunitas animasi (mungkin karena spesifikasinya yang sederhana).

4.3.2. Format file MNM

Format file ini memungkinkan penggantian nama segmen file BVH agar sesuai dengan konvensi yang digunakan di Autodesk 3D Studio Max. Nama yang ditentukan oleh pengguna dikaitkan dengan label yang telah ditentukan sebelumnya untuk segmen berkaki dua. misalnya. Humerus = L UpperArm
Format file ini juga memungkinkan penggantian nama penanda file CSM agar sesuai dengan konvensi yang digunakan di Autodesk 3D Studio Max. Nama yang ditentukan oleh pengguna dikaitkan dengan label standar yang diharapkan oleh Character Studio. misalnya. LeftShoulder =

4.3.3. Format file LSHO MBX

Format MBX adalah format kerangka hierarki biner yang eksklusif untuk sistem motion capture Noitom Perception Neuron. File dapat dimuat ke perangkat lunak Noitom Axis Studio dan kemudian diekspor ke BVH dan format lain dari perangkat lunak mereka.

4.3.4 Format MVNX

Format MVNX adalah format berbasis XML terbuka yang dapat dibaca manusia untuk menyimpan Xsens data motion capture. Formatnya berisi posisi dan orientasi 3D dari semua segmen yang ditangkap dengan Xsens MVN. Selain itu, formatnya menyertakan beberapa variabel lain untuk diekspor seperti sudut sambungan, kecepatan segmen dan akselerasi bebas, pusat lintasan massa, dan data sensor yang dikalibrasi dari pelacak gerak individu. Data MVNX juga dapat diimpor ke program perangkat lunak terkemuka termasuk MATLAB dan Excel.

4.3.5. Format file ASK/SDL

Format ini merupakan varian dari format file BVH yang dikembangkan oleh Biovision. File ASK (Alias Skeleton) hanya berisi informasi tentang kerangka dan, akibatnya, tidak berisi informasi apa pun tentang saluran atau pergerakan. Koordinat offset bersifat absolut, tidak seperti BVH yang relatif. File SDL yang terkait dengan file ASK berisi data gerakan tetapi dapat berisi banyak informasi lain mengenai adegan daripada sampel gerakan itu sendiri.

4.3.6 Format file AOA

Adaptative Optics adalah perusahaan yang didedikasikan untuk pembuatan dukungan perangkat keras untuk motion capture. Format ASCII hanya menjelaskan penciluk dan posisinya pada setiap periode pengambilan sampel.

4.3.7 Format file ASF/AMC

Format ini dikembangkan oleh Acclaim, sebuah perusahaan video game. Setelah masuk dalam domain publik telah digunakan oleh Oxford Metrics (Vicon Motion Capture Systems). Format Acclaim terdiri dari dua file berbeda, satu untuk kerangka dan yang lainnya untuk pergerakan. Pemisahan antara kedua jenis ini dilakukan karena kerangka yang sama sering digunakan untuk berbagai gerakan yang berbeda. File yang berisi deskripsi kerangka dalam file ASF (Acclaim Skeleton File) dan file yang berisi data pergerakan adalah file AMC (Acclaim Motion Capture data).

4.3.8 Format file BRD

Format ini secara unik digunakan oleh sistem motion capture Ascension Technology "Flock of Birds" yang dikembangkan oleh LambSoft. Itu memungkinkan menyimpan data yang berasal dari sistem motion capture magnetik.

4.3.9 Format file GRC

Format file GRC adalah format file untuk menyimpan data motion capture dari mocap Synertial. GRC mencakup data RAW dari sensor inersia (seperti rotasi, akselerasi, dan kekuatan medan magnet), detail kerangka, posisi absolut akar kerangka, dan berbagai metadata (catatan, TimeCode, ..).

Berkat fakta bahwa data RAW dari file ini dibaca oleh Synertial SDK dan struktur kerangka dihitung ulang setiap kali dibutuhkan, format file hemat memori. Data format file GRC kompatibel dan dapat diekspor ke format file BVH dan FBX menggunakan alat perangkat lunak Synertial.

4.3.10. Format file HTR dan GTR

Format HTR (Hierarchical Translation Rotation) telah dikembangkan sebagai format asli untuk kerangka perangkat lunak Analisis Gerak. Ini telah dibuat sebagai alternatif dari format BVH untuk mengatasi kelemahan utamanya. Ada varian HTR yang disebut format GTR (Global Translation Rotation) dan merupakan format yang sama tanpa informasi struktural.

4.3.11. Format file TAK

Tak, dilafalkan "take", digunakan oleh Motive dikembangkan oleh OptiTrack. File tersebut dapat berisi informasi tentang:

- Posisi penanda dan kesalahan residu
- Informasi kerangka dan 6DoF tubuh kaku dan rotasi
- Force Plate Data
- Audio
- Informasi umum tentang pengambilan, seperti frekuensi gambar, properti penghitungan penanda, filter kamera, metode sinkronisasi, data/waktu sistem perekaman
- SMPTE Timecode
- Recorded Camera Data

4.3.12 format file TRC

Format file TRC adalah format file lain dari Motion Analysis. Ini tidak hanya berisi data mentah dari sistem motion capture seluruh tubuh yang mereka kembangkan, tetapi juga data keluaran yang berasal dari pelacak wajah mereka.

Format file TRC, sebaliknya dari kebanyakan lainnya, tidak berbasis kerangka.

4.3.13 Format file CSM

Format CSM adalah format pelacakan optik yang digunakan oleh Character Studio (animasi dan plugin skinning untuk 3ds Max) untuk mengimpor data marker.

4.3.14 Format file V/VSK

Format file V adalah format data gerakan biner yang dikembangkan oleh Vicon Motion Systems. File ini biasanya digunakan bersamaan dengan file VSK yang juga dikembangkan oleh Vicon Motion System. File VSK berisi hierarki kerangka. File V dapat berisi data berikut:

- Data penanda
- Translasi segmen global dan data rotasi
- Data rotasi lokal (dengan data translasi root)

Format file C3D

Format file C3D adalah domain publik, format file biner dikembangkan di pertengahan 1980-an di The National Institutes of Health di Bethesda, Maryland. Ini menyimpan informasi koordinat 3D, data analog dan informasi terkait yang digunakan dalam pengambilan data gerak 3D dan operasi analisis selanjutnya. Format file C3D, sebaliknya kebanyakan lainnya, tidak berbasis kerangka, dan biner.

4.3.15 Format file GMS

Format GMS (Gesture and Motion Signal) adalah format tingkat rendah, biner, minimal, tetapi umum, untuk menyimpan Gesture dan Motion Signals dengan cara yang fleksibel, teratur, dan optimal. Format GMS memperhitungkan fitur minimal yang dibutuhkan oleh format yang membawa informasi gerakan/gerakan: dimensi fleksibel untuk sinyal, strukturasi serbaguna, jenis fleksibel dari variabel yang disandikan, dan properti spasial dan temporal dari sinyal gerakan dan gerakan. GMS mendapat dukungan dari FP6 Network of Excellence IST-2002-002114 – "Enactive Interfaces".

Format file GMS, sebaliknya dari kebanyakan lainnya, BUKAN berbasis kerangka, dan biner.

4.3.16 Format file HDF

Format file biner tertutup yang dikembangkan oleh House of Moves untuk digunakan dalam perangkat lunak milik mereka yang disebut (pada saat itu) Diva. Format file ini pada dasarnya adalah dump dari adegan Diva. Ini mencakup semua data penanda translasi serta semua data tulang rotasi di tempat kejadian dan banyak lagi.

4.3.17 Format file FBX

Format FBX (.fbx) dimiliki oleh Autodesk sejak 2006. Blender Foundation telah menerbitkan spesifikasi tidak resmi untuk FBX biner.

4.3.18 Format file PZ2

Format PZ2 digunakan oleh perangkat lunak gambar 3D populer Poser dan DAZ Studio. Formatnya dapat menyandikan animasi tubuh dan wajah, dan dapat diterapkan dengan menjatuhkan file ke karakter di viewport. Ini berbasis teks, mirip dengan struktur XML, dan dapat dengan mudah diedit. Ada juga varian wajah saja (tanpa leher) yang disebut FC2. Perangkat lunak motion capture yang dapat menghasilkan PZ2 termasuk Zign Track, F-Clone, Kinect Capture, Brekel Face, dan Faceshift melalui skrip gratis.

4.4 membahas format motion capture yang sangat umum dipakai

4.4.1 Format file ASF/AMC Format

ini dikembangkan oleh [Acclaim](#), sebuah perusahaan video game. Setelah masuk ke domain publik, itu telah digunakan oleh Oxford Metrics (Vicon Motion Capture Systems). Format Acclaim terdiri dari dua file yang berbeda, satu untuk kerangka dan satu lagi untuk gerakan. Pemisahan antara kedua jenis ini dilakukan karena kerangka yang sama sering digunakan untuk berbagai gerakan yang berbeda. File yang berisi deskripsi skeleton dalam file ASF (Acclaim Skeleton File) dan file yang berisi data pergerakan adalah file AMC (Acclaim Motion Capture data).

CONTOH CODE walk.asf

```
#
# Comment line
##
#
:version 1.10
:name BioSkeleton
:units
  mass 1.0
  length 1.0
  angle deg
:documentation
  Example of an Acclaim skeleton
  To be used with "Walk.amc"
:root
  axis XYZ
  order TX TY TZ RZ RY RX
  position 0.0 0.0 0.0
  orientation 0.0 0.0 0.0
:bonedata
begin
  id 1
  name hips
  direction 0.000000 1.000000 0.000000
  length 0.000000
  axis 0.00000 0.00000 0.00000 XYZ
  dof rx ry rz
  limits (-180.0 180.0)
    (-180.0 180.0)
    (-180.0 180.0)
end
begin
```



```

id 2
name hips1
direction 0.000000 1.000000 0.000000
length 4.310000
axis 0.000 0.000 0.000 XYZ
end
begin
id 3
name chest
direction 0.000000 1.000000 0.000000
length 0.000000
axis 0.00000 0.00000 0.00000 XYZ
dof rx ry rz
limits (-180.0 180.0)
(-180.0 180.0)
(-180.0 180.0)
end
begin
id 4
name chest1
direction 0.000000 1.000000 0.000000
length 16.870001
axis 0.000 0.000 0.000 XYZ
end
begin
id 5
name neck
direction 0.000000 1.000000 0.000000
length 4.020000
axis 0.00000 0.00000 0.00000 XYZ
dof rx ry rz
limits (-180.0 180.0)
(-180.0 180.0)
(-180.0 180.0)
end
begin
id 6
name head
direction 1.000000 0.000000 0.000000
length 0.000000
axis 0.00000 0.00000 0.00000 XYZ
dof rx ry rz
limits (-180.0 180.0)

```

```

    (-180.0 180.0)
    (-180.0 180.0)
end
begin
  id 7
  name chest2
  direction 0.068191 0.991126 0.114101
  length 14.811435
  axis 0.000 0.000 0.000 XYZ
end
begin
  id 8
  name leftcollar
  direction 1.000000 0.000000 0.000000
  length 4.830000
  axis 0.00000 0.00000 0.00000 XYZ
  dof rx ry rz
  limits (-180.0 180.0)
    (-180.0 180.0)
    (-180.0 180.0)
end
begin
  id 9
  name leftuparm
  direction 0.000000 -1.000000 0.000000
  length 12.160000
  axis 0.00000 0.00000 0.00000 XYZ
  dof rx ry rz
  limits (-180.0 180.0)
    (-180.0 180.0)
    (-180.0 180.0)
end
begin
  id 10
  name leftlowarm
  direction 0.000000 -1.000000 0.000000
  length 9.700000
  axis 0.00000 0.00000 0.00000 XYZ
  dof rx ry rz
  limits (-180.0 180.0)
    (-180.0 180.0)
    (-180.0 180.0)
end

```

```
begin
  id 11
  name lefthand
  direction 1.000000 0.000000 0.000000
  length 0.000000
  axis 0.00000 0.00000 0.00000 XYZ
  dof rx ry rz
  limits (-180.0 180.0)
    (-180.0 180.0)
    (-180.0 180.0)
end
begin
  id 12
  name chest3
  direction -0.068191 0.991126 0.114101
  length 14.811435
  axis 0.000 0.000 0.000 XYZ
end
begin
  id 13
  name rightcollar
  direction -1.000000 0.000000 0.000000
  length 5.390000
  axis 0.00000 0.00000 0.00000 XYZ
  dof rx ry rz
  limits (-180.0 180.0)
    (-180.0 180.0)
    (-180.0 180.0)
end
begin
  id 14
  name rightuparm
  direction 0.000000 -1.000000 0.000000
  length 12.480000
  axis 0.00000 0.00000 0.00000 XYZ
  dof rx ry rz
  limits (-180.0 180.0)
    (-180.0 180.0)
    (-180.0 180.0)
end
begin
  id 15
  name rightlowarm
```

```
direction 0.000000 -1.000000 0.000000
length 10.060000
axis 0.00000 0.00000 0.00000 XYZ
dof rx ry rz
limits (-180.0 180.0)
(-180.0 180.0)
(-180.0 180.0)
end
begin
id 16
name righthand
direction 1.000000 0.000000 0.000000
length 0.000000
axis 0.00000 0.00000 0.00000 XYZ
dof rx ry rz
limits (-180.0 180.0)
(-180.0 180.0)
(-180.0 180.0)
end
begin
id 17
name hips2
direction 1.000000 0.000000 0.000000
length 3.240000
axis 0.000 0.000 0.000 XYZ
end
begin
id 18
name leftupleg
direction 0.000000 -1.000000 0.000000
length 17.290001
axis 0.00000 0.00000 0.00000 XYZ
dof rx ry rz
limits (-180.0 180.0)
(-180.0 180.0)
(-180.0 180.0)
end
begin
id 19
name leftlowleg
direction 0.000000 -1.000000 0.000000
length 16.400000
axis 0.00000 0.00000 0.00000 XYZ
```

```
dof rx ry rz
limits (-180.0 180.0)
  (-180.0 180.0)
  (-180.0 180.0)
end
begin
  id 20
  name leftfoot
  direction 1.000000 0.000000 0.000000
  length 0.000000
  axis 0.00000 0.00000 0.00000 XYZ
  dof rx ry rz
  limits (-180.0 180.0)
    (-180.0 180.0)
    (-180.0 180.0)
end
begin
  id 21
  name hips3
  direction -1.000000 0.000000 0.000000
  length 3.240000
  axis 0.000 0.000 0.000 XYZ
end
begin
  id 22
  name rightupleg
  direction 0.000000 -1.000000 0.000000
  length 17.680000
  axis 0.00000 0.00000 0.00000 XYZ
  dof rx ry rz
  limits (-180.0 180.0)
    (-180.0 180.0)
    (-180.0 180.0)
end
begin
  id 23
  name rightlowleg
  direction 0.000000 -1.000000 0.000000
  length 15.510000
  axis 0.00000 0.00000 0.00000 XYZ
  dof rx ry rz
  limits (-180.0 180.0)
    (-180.0 180.0)
```

```

    (-180.0 180.0)
end
begin
  id 24
  name rightfoot
  direction 1.000000 0.000000 0.000000
  length 0.000000
  axis 0.000000 0.000000 0.000000 XYZ
  dof rx ry rz
  limits (-180.0 180.0)
    (-180.0 180.0)
    (-180.0 180.0)
end
:hierarchy
begin
  root hips
  hips hips1 hips2 hips3
  hips1 chest
  chest chest1 chest2 chest3
  chest1 neck
  neck head
  chest2 leftcollar
  leftcollar leftuparm
  leftuparm leftlowarm
  leftlowarm lefthand
  chest3 rightcollar
  rightcollar rightuparm
  rightuparm rightlowarm
  rightlowarm righthand
  hips2 leftupleg
  leftupleg leftlowleg
  leftlowleg leftfoot
  hips3 rightupleg
  rightupleg rightlowleg
  rightlowleg rightfoot
end

```

Periksa contoh file "Walk.asf". Di sana Anda akan melihat bahwa kata kunci dalam file semuanya dimulai dengan titik dua ":". Kata kunci akan menetapkan nilai global atau menunjukkan awal bagian data.

Kata kunci ":version" menunjukkan versi definisi kerangka. Dokumen ini untuk versi 1.10. Kata kunci ":name" memungkinkan kerangka diberi nama selain nama file.

Kata kunci ":units" menunjukkan bagian yang menentukan satuan yang akan digunakan untuk berbagai jenis data. Ini juga menentukan nilai default untuk parameter lain. Sejumlah spesifikasi dapat ditemukan di sini, penggunaan nilai-nilai ini seringkali khusus untuk program. Idealnya Anda harus menyimpan nilai-nilai ini dan kemudian menuliskannya lagi jika Anda melakukan modifikasi pada file. Jika Anda berniat untuk hanya membaca data gerak maka Anda dapat mengabaikan nilai-nilai yang tidak menarik bagi Anda. Di bagian ini Anda akan menemukan satuan yang digunakan untuk sudut dan terkadang nilai default untuk massa dan panjang segmen. Bagian ":documentation" memungkinkan penyimpanan informasi dokumentasi yang akan bertahan dari satu pembuatan file berikutnya. Informasi komentar sederhana dalam file tidak dijamin akan dipertahankan jika file tersebut dibaca ke dalam memori daripada disimpan ke file lain, mungkin dengan modifikasi.

Bagian ":root" mendefinisikan segmen khusus dari adegan (yah, ini khusus untuk cara format file ditentukan, Anda benar-benar dapat memperlakukan ini seperti segmen lainnya dengan cara lain). Ini adalah segmen akar dari hierarki kerangka. Ini sangat mirip dengan semua segmen lainnya tetapi tidak berisi informasi arah dan panjang. Kata kunci "axis" di bagian root menentukan urutan rotasi objek root. Kata kunci "order" menentukan saluran gerakan yang diterapkan ke root dan dalam urutan apa saluran tersebut akan muncul di file AMC. Kata kunci "position" dan "orientation" masing-masing diikuti oleh triplet angka yang menunjukkan posisi awal dan orientasi root. Ini biasanya, tetapi tidak selalu, nol.

Bagian ":bonedata" berisi deskripsi setiap segmen dalam hierarki. Deskripsi ini hanya untuk segmen. Bagian hirarki, yang akan datang berikutnya, akan menjelaskan organisasi induk dari segmen-segmen tersebut. Definisi segmen dikurung oleh pasangan kata kunci "begin" dan "end" (perhatikan kurangnya tanda titik dua di setiap kata kunci). Dalam definisi segmen Anda akan menemukan:

-
- "id" Ini adalah nomor yang memberikan id unik untuk segmen tersebut. Ini sebenarnya tidak diperlukan karena setiap segmen juga diberi nama dan nama tersebut digunakan untuk bagian hierarki dan di file AMC.
- "name" Ini memberikan nama segmen. Setiap segmen harus memiliki nama yang unik. Seringkali Anda akan melihat segmen dengan nama yang mirip seperti "hips" dan "hips11". Ruas yang memiliki angka pada akhirnya biasanya merupakan turunan dari ruas dengan nama yang sama tetapi tanpa nomor. Segmen dengan angka biasanya ada hanya untuk mengisi celah antara segmen induk dan segmen anak. Seringkali segmen dengan angka di bagian akhir tidak memiliki data motion capture. Anda tidak memperlakukan ini dengan cara khusus, ini hanya dianggap sebagai fitur yang menarik.
- "direction" Ini adalah arah segmen. Ini menentukan bagaimana segmen harus digambar dan juga menentukan arah dari segmen induk ke segmen anak. Arah dan panjang segmen menentukan offset anak dari induk, jika ada anak segmen.
- "length" Panjang segmen. Dengan nilai arah memberikan informasi yang dibutuhkan untuk menggambar segmen.
- "axis" Ini memberikan sumbu rotasi untuk segmen tersebut. Dengan menetapkan ini sebagai nilai terpisah, data gerak dapat terpisah dari informasi gambar dan hierarki. Ini sangat

berguna untuk aplikasi yang mungkin menyediakan alat pengeditan gerakan yang peka terhadap kunci gimbal.

- "dof" Ini adalah singkatan dari "Degrees of Freedom" dan menentukan jumlah saluran gerak dan urutan kemunculannya di file AMC. Jika kata kunci dof tidak muncul maka segmen tidak mendapatkan data gerak. Tidak ada saluran terjemahan yang akan muncul di sini hanya saluran rotasi dan terkadang saluran panjang.
- "limits" Spesifikasi ini memberikan batasan pada masing-masing saluran dalam spesifikasi dof. Untuk setiap saluran yang muncul akan ada sepasang angka di dalam tanda kurung yang menunjukkan nilai minimum dan maksimum yang diperbolehkan untuk saluran tersebut. Informasi ini tidak digunakan untuk menginterpretasikan data gerakan, ini berguna hanya untuk aplikasi yang mungkin menerapkan fungsi pengeditan gerakan yang membatasi rotasi. Ini juga tidak berarti bahwa data dalam file AMC mungkin dibatasi oleh angka yang diberikan.

4.4.2 BioVision: BVH (Data Hirarki BioVision)

BVA dan format file BVH

BVH singkatan dari **Biovision Hierarchical Data**, yang dikembangkan oleh perusahaan motion capture bernama Biovision. Format BVA (juga dikembangkan oleh Biovision) adalah format lama yang merupakan pendahulu BVH. Format BVH terutama digunakan sebagai representasi standar gerakan dalam animasi struktur humanoid. Saat ini salah satu format data gerak paling populer dan telah diadopsi secara luas oleh komunitas animasi (mungkin karena spesifikasinya yang sederhana).

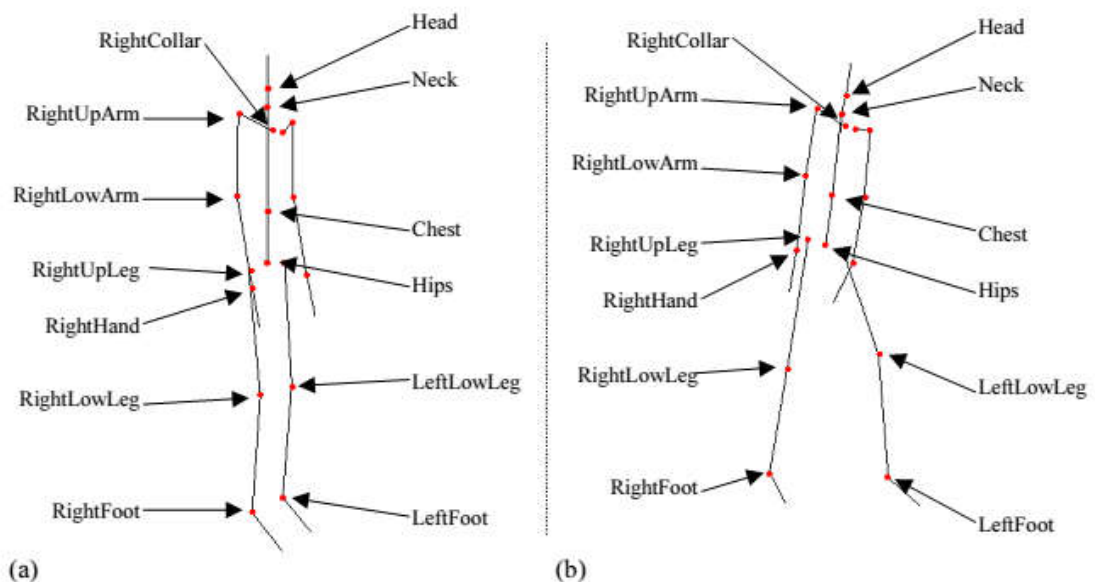
Pada 2019, BVH banyak digunakan dan sebagian besar aplikasi 3D mendukung impor dan ekspor file dalam format ini.

Contoh perangkat lunak yang menggunakan file dalam format BVH:

- Lightwave 3D
- 3ds Max (versi 9 atau lebih baru)
- Blender
- Clara.io
- Cobalt
- Daz Studio
- Esenthel Engine
- Lifeforms
- MakeHuman
- Maya
- Modo
- Poser
- Seamless3d open source gratis
- Second Life / OpenSim
- Avimator dan port Qt-nya Qavimator

- Maxon Cinema 4d
- formZ
- Milkshape 3D
- Papervision 3D
- OpenQwaq
- Webots
- Xsens

Format BVH menggantikan format data BVA BioVision dengan penambahan struktur data hierarkis yang nyata yang mewakili tulang-tulang kerangka. File BVH terdiri dari dua bagian di mana bagian pertama merinci hierarki dan pose awal kerangka dan bagian kedua menjelaskan data channel untuk setiap frame, demikian juga bagian gerak. Ilustrasi posisi dasar dan frame pertama dari sebuah animasi diberikan pada gambar 3.1, di mana datanya tercantum pada gambar 3.2. Contoh file BVH pada gambar 3.2 akan digunakan untuk membahas lebih lanjut format file BVH di sisa bagian ini.



contoh file bvh

```

HIERARCHY
ROOT Hips
{
  OFFSET 0.00  0.00  0.00
  CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
  JOINT Chest
  {
    OFFSET 0.00  5.21  0.00
  }

```

```

CHANNELS 3 Zrotation Xrotation Yrotation
JOINT Neck
{
    OFFSET 0.00  18.65  0.00
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT Head
    {
        OFFSET 0.00  5.45  0.00
        CHANNELS 3 Zrotation Xrotation Yrotation
        End Site
        {
            OFFSET 0.00  3.87  0.00
        }
    }
}
JOINT LeftCollar
{
    OFFSET 1.12  16.23  1.87
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT LeftUpArm
    {
        OFFSET 5.54  0.00  0.00
        CHANNELS 3 Zrotation Xrotation Yrotation
        JOINT LeftLowArm
        {
            OFFSET 0.00  -11.96  0.00
            CHANNELS 3 Zrotation Xrotation Yrotation
            JOINT LeftHand
            {
                OFFSET 0.00  -9.93  0.00
                CHANNELS 3 Zrotation Xrotation Yrotation
                End Site
                {
                    OFFSET 0.00  -7.00  0.00
                }
            }
        }
    }
}
JOINT RightCollar
{
    OFFSET -1.12  16.23  1.87
    CHANNELS 3 Zrotation Xrotation Yrotation

```

```

JOINT RightUpArm
{
    OFFSET -6.07  0.00  0.00
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT RightLowArm
    {
        OFFSET 0.00  -11.82  0.00
        CHANNELS 3 Zrotation Xrotation Yrotation
        JOINT RightHand
        {
            OFFSET 0.00  -10.65  0.00
            CHANNELS 3 Zrotation Xrotation Yrotation
            End Site
            {
                OFFSET 0.00  -7.00  0.00
            }
        }
    }
}

JOINT LeftUpLeg
{
    OFFSET 3.91  0.00  0.00
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT LeftLowLeg
    {
        OFFSET 0.00  -18.34  0.00
        CHANNELS 3 Zrotation Xrotation Yrotation
        JOINT LeftFoot
        {
            OFFSET 0.00  -17.37  0.00
            CHANNELS 3 Zrotation Xrotation Yrotation
            End Site
            {
                OFFSET 0.00  -3.46  0.00
            }
        }
    }
}

JOINT RightUpLeg
{
    OFFSET -3.91  0.00  0.00

```

```

CHANNELS 3 Zrotation Xrotation Yrotation
JOINT RightLowLeg
{
    OFFSET 0.00 -17.63 0.00
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT RightFoot
    {
        OFFSET 0.00 -17.14 0.00
        CHANNELS 3 Zrotation Xrotation Yrotation
        End Site
        {
            OFFSET 0.00 -3.75 0.00
        }
    }
}
}
}
}
}
}
MOTION
Frames: 2
Frame Time: 0.033333
8.03 35.01 88.36 -3.41 14.78 -164.35 13.09 40.30 -24.60 7.88 43.80 0.00 -
3.61 -41.45 5.82 10.08 0.00 10.21 97.95 -23.53 -2.14 -101.86 -80.77 -98.91
0.69 0.03 0.00 -14.04 0.00 -10.50 -85.52 -13.72 -102.93 61.91 -61.18 65.18 -
1.57 0.69 0.02 15.00 22.78 -5.92 14.93 49.99 6.60 0.00 -1.14 0.00 -
16.58 -10.51 -3.11 15.38 52.66 -21.80 0.00 -23.95 0.00
7.81 35.10 86.47 -3.78 12.94 -166.97 12.64 42.57 -22.34 7.67 43.61 0.00 -
4.23 -41.41 4.89 19.10 0.00 4.16 93.12 -9.69 -9.43 132.67 -81.86 136.80
0.70 0.37 0.00 -8.62 0.00 -21.82 -87.31 -27.57 -100.09 56.17 -61.56 58.72 -
1.63 0.95 0.03 13.16 15.44 -3.56 7.97 59.29 4.97 0.00 1.64 0.00 -
17.18 -10.02 -3.08 13.56 53.38 -18.07 0.00 -25.93 0.00

```

Bagian hierarki file dimulai dengan kata kunci HIERARCHY, yang diikuti pada baris berikutnya dengan kata kunci ROOT dan nama tulang yang merupakan root hierarki kerangka. Kata kunci ROOT menunjukkan awal dari struktur hierarki kerangka baru dan meskipun file BVH mampu berisi banyak kerangka, biasanya hanya satu kerangka yang ditentukan per file. Struktur kerangka yang tersisa didefinisikan dalam sifat rekursif di mana definisi setiap tulang, termasuk child-child, dikemas dalam kurung kurawal, yang dibatasi pada baris sebelumnya dengan kata kunci JOINT (atau ROOT dalam kasus tulang root) diikuti dengan nama tulang. Dengan diperkenalkannya kurung kurawal kiri, adalah praktik yang baik untuk membuat indentasi isi tulang (dengan tab) dan menyelaraskan. kurung kurawal penutup dengan kurung buka yang sesuai. Nama tulang yang diidentifikasi dengan awalan JOINT atau ROOT tidak direferensikan lagi dalam file dan karenanya berlebihan, namun beberapa parsing (misalnya Character Studio R2.2 (Max |) memerlukan nama tulang untuk mengurai file

dengan benar. Selanjutnya, meskipun lekukan hierarkis tidak mutlak diperlukan, ini membantu membuat file lebih mudah dibaca oleh manusia. Dalam definisi setiap tulang, baris pertama, dibatasi oleh kata kunci OFFSET, merinci terjemahan asal tulang sehubungan dengan asal induknya (atau secara global dalam kasus tulang root) di sepanjang sumbu x, y and z. Offset melayani tujuan lebih lanjut untuk secara implisit mendefinisikan panjang dan arah tulang induk, namun masalahnya adalah dalam mendefinisikan panjang dan arah tulang yang memiliki banyak child. Biasanya pilihan yang baik untuk menentukan panjang tulang dalam situasi ini adalah dengan menggunakan definisi offset child pertama untuk menyimpulkan informasi tulang induk dan memperlakukan data offset untuk node child lainnya hanya sebagai nilai offset.

Baris kedua dari definisi tulang diawali dengan kata kunci CHANNEL yang mendefinisikan DOF untuk tulang saat ini. Pentingnya urutan channel yang disajikan ada dua.

Pertama, urutan setiap channel yang terlihat di bagian hirarki file sama persis dengan urutan 'data di bagian gerakan file. Misalnya, bagian gerak file berisi informasi untuk channel tulang root dalam urutan yang ditentukan dalam hierarki, diikuti oleh data channel untuk child pertamanya, diikuti oleh data channel untuk child itu, dan seterusnya melalui hierarki. Hal kedua yang perlu diperhatikan sehubungan dengan urutan channel adalah bahwa urutan penggabungan sudut Euler saat membuat matriks rotasi tulang harus mengikuti urutan yang digambarkan di bagian channel. Penting untuk dicatat ini karena urutan Euler ditentukan untuk setiap tulang, oleh karena itu dimungkinkan untuk memiliki urutan yang berbeda untuk tulang yang berbeda, yang perlu diperhitungkan untuk mendapatkan tampilan animasi yang benar. Gambar 3.3 mengilustrasikan segmen file BVH dimana channel rotasi ditentukan secara berbeda untuk sambungan yang berbeda.

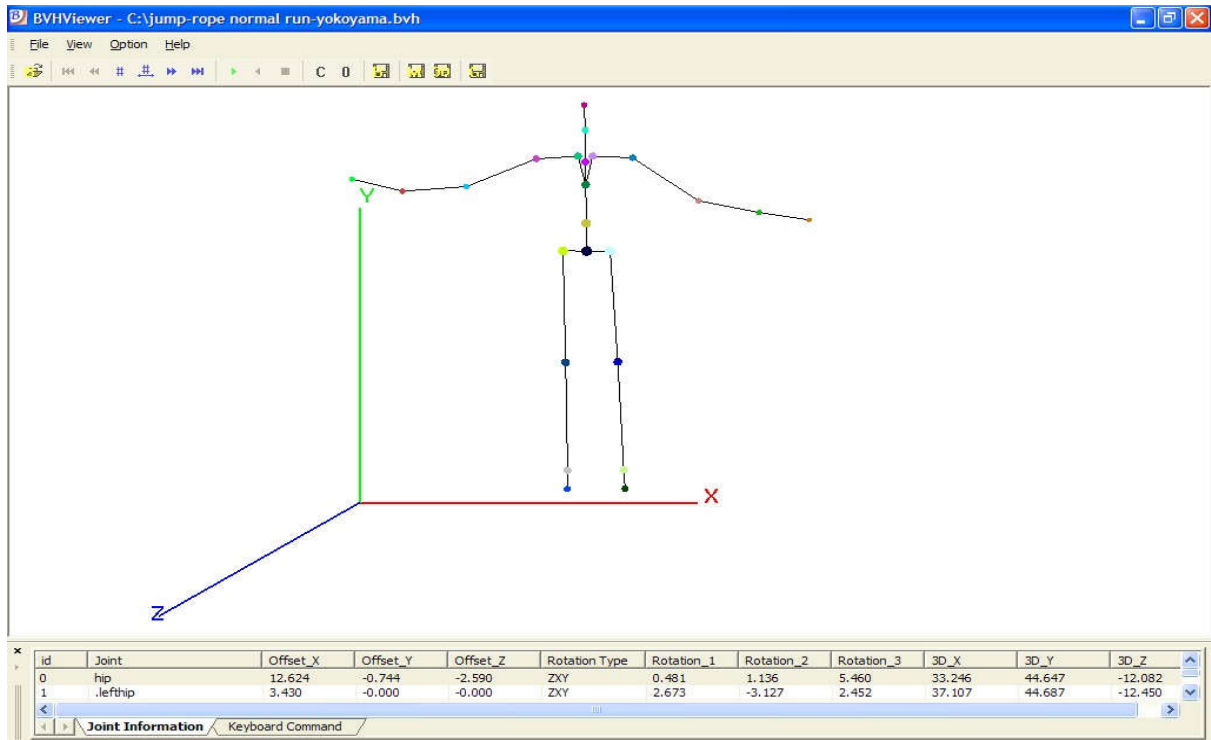
```

...
...
OFFSET -0.145148 7.670311 -0.132667
CHANNELS 3 Xrotation Zrotation Yrotation
JOINT Neck
{
    OFFSET      0.206702 11.265274 -1.343565
    CHANNELS 3 Xrotation Zrotation Yrotation
    JOINT Head
    {
        OFFSET      -0.103351 0.386947 0.663649
        CHANNELS 3 Xrotation Zrotation Yrotation
        JOINT Left Eye
        {
            OFFSET      1.446916 5.400717 4.503910
            CHANNELS 3 Zrotation Yrotation Xrotation
            End Site
            {
                OFFSET 0.000000 0.000000 0.930160
            }
        }
        JOINT Right Eye
        {
            OFFSET      -1.446916 5.400717 4.503910
            CHANNELS 3 Zrotation Yrotation Xrotation
            End Site
            {
                OFFSET 0.000000 0.000000 0.930160
            }
        }
    }
}
...
...

```

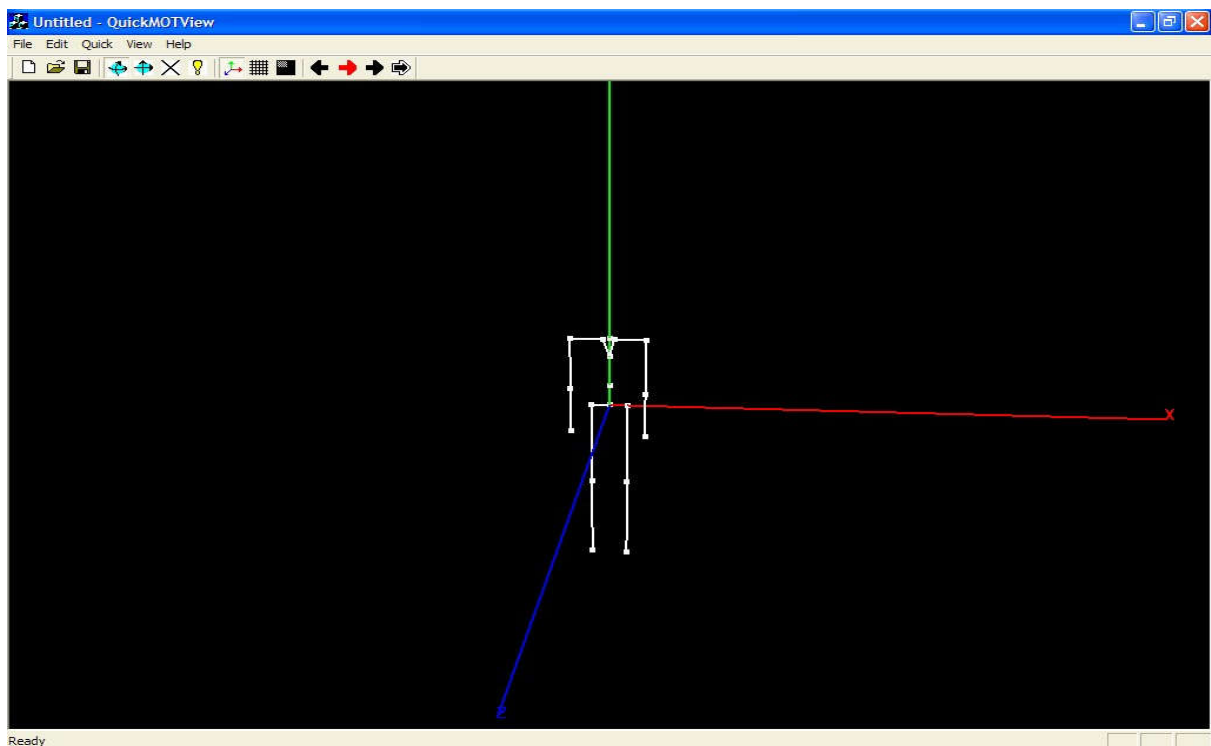
Setelah baris OFFSET dan CHANNEL, baris non-bersarang berikutnya dalam definisi tulang digunakan untuk mendefinisikan item child, dimulai dengan kata kunci JOINT, namun dalam kasus end-effectors, tag khusus digunakan, "Situs Akhir", yang merangkum triple OFFSET yang digunakan untuk menyimpulkan panjang dan orientasi tulang.

normal jump-rope file run-yokoyama.bvh seperti yang dilihat dengan menggunakan BvhViewer .



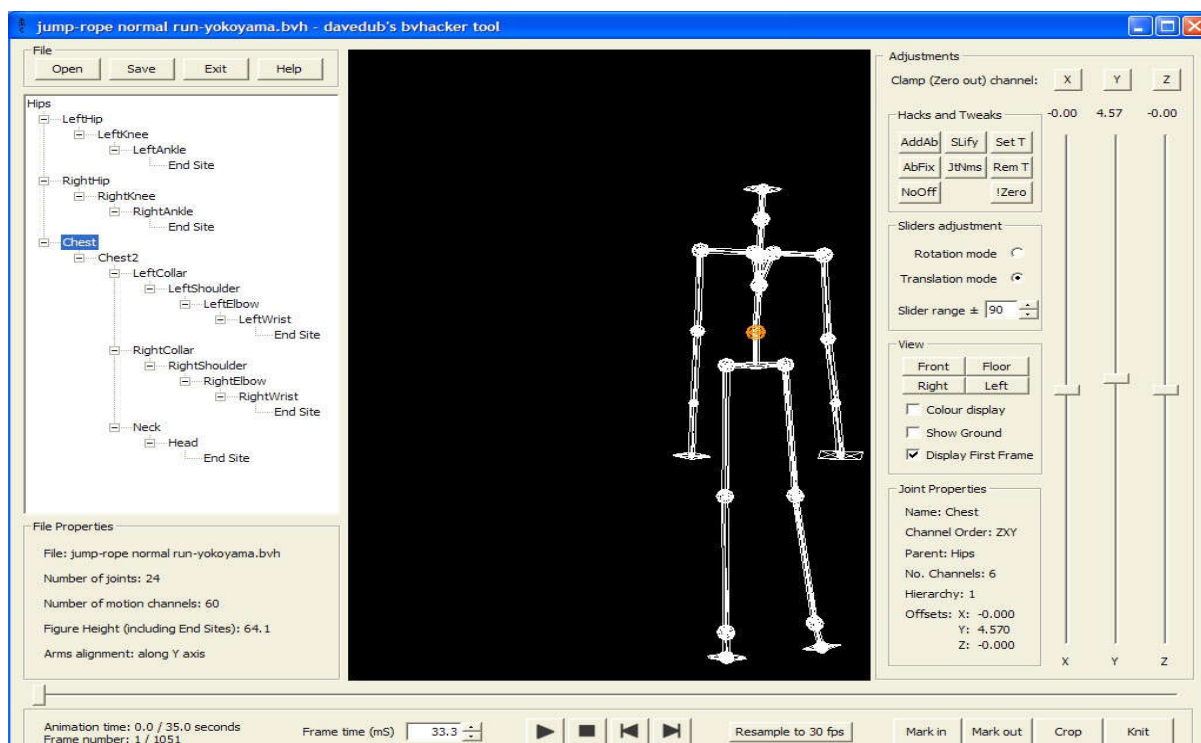
Tampilan BvhViewer dari frame 0 lompat tali normal run-yokoyama.bvh

di bawah menunjukkan frame 0 dari lompat tali normal run-yokoyama.bvh yang sama yang diajukan seperti yang dilihat dengan menggunakan QuickMotView 1.0 .



Tampilan QuickMotView 1.0 dari frame 0 dari jump-rope normal run-yokoyama.bvh

di bawah ini menunjukkan frame 1 dari file jump-rope normal run-yokoyama.bvh yang sama dengan sambungan Dada disorot dan seperti yang dilihat dengan menggunakan bvhacker 1.2 perangkat lunak.



bvhacker 1.2 tampilan frame 0 lompat tali normal run-yokoyama.bvh

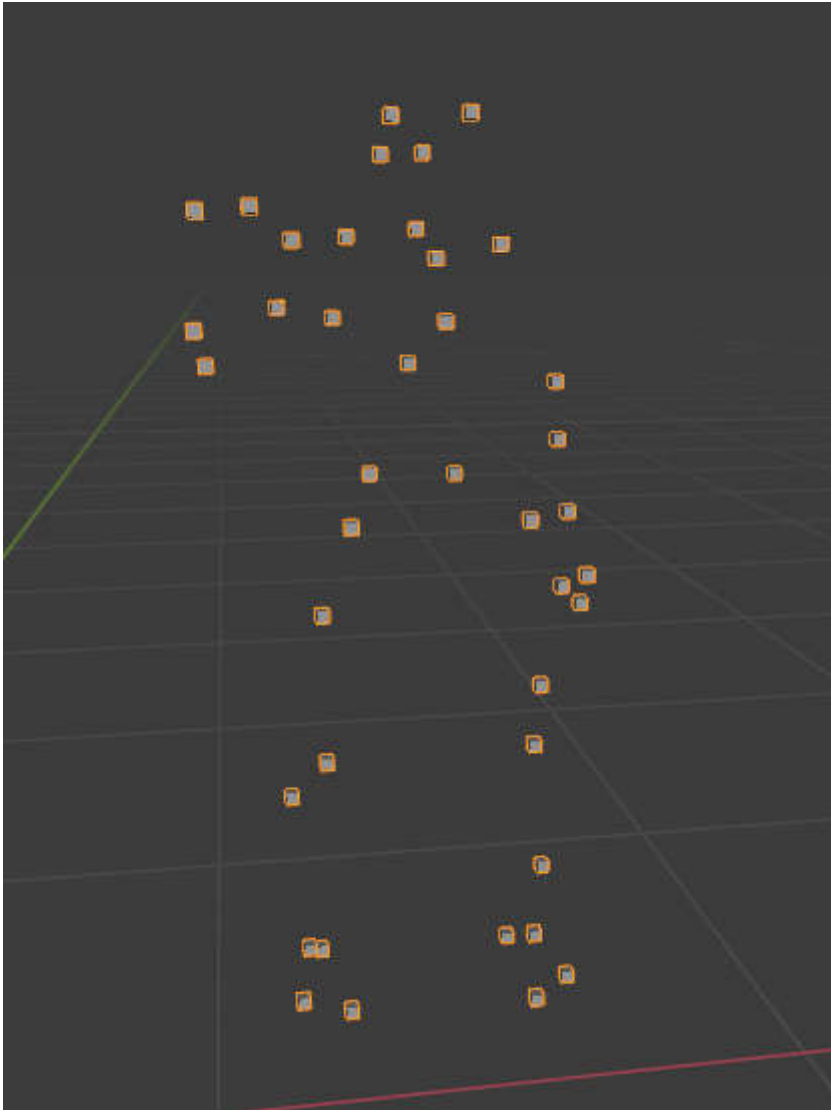
4.4.3 Format C3D

C3D Format file C3D adalah domain publik, format file biner yang dikembangkan pada pertengahan 1980-an di Institut Kesehatan Nasional di Bethesda, Maryland.

C3D adalah format file domain publik yang telah digunakan di laboratorium Biomekanik, Animasi, dan Analisis Gaya Berjalan untuk merekam data analog dan 3D yang disinkronkan sejak pertengahan 1980-an. Ini didukung oleh semua produsen Sistem penangkap gerak / motion capture 3D utama, dan perusahaan di Industri Biomekanik, Motion capture, dan Animasi. File C3D adalah standar yang berisi semua informasi yang diperlukan untuk membaca, menampilkan, dan menganalisis data gerak 3D dengan data analog tambahan dari pelat gaya, elektromiografi, akselerometer, dan sensor lainnya.

Tidak seperti berbagai format 3D lainnya (file Biovision BVH, file OpenSim .TRC dan .MOT dll.) format .C3D adalah standar yang tidak berubah setiap kali produsen merilis produk baru sehingga data yang disimpan dalam format C3D akan tetap dapat dibaca untuk waktu yang lama. File sampel tersedia di situs web ini yang dibuat pada pertengahan 1990-an dan dapat dibuka serta diproses oleh aplikasi C3D apa pun.

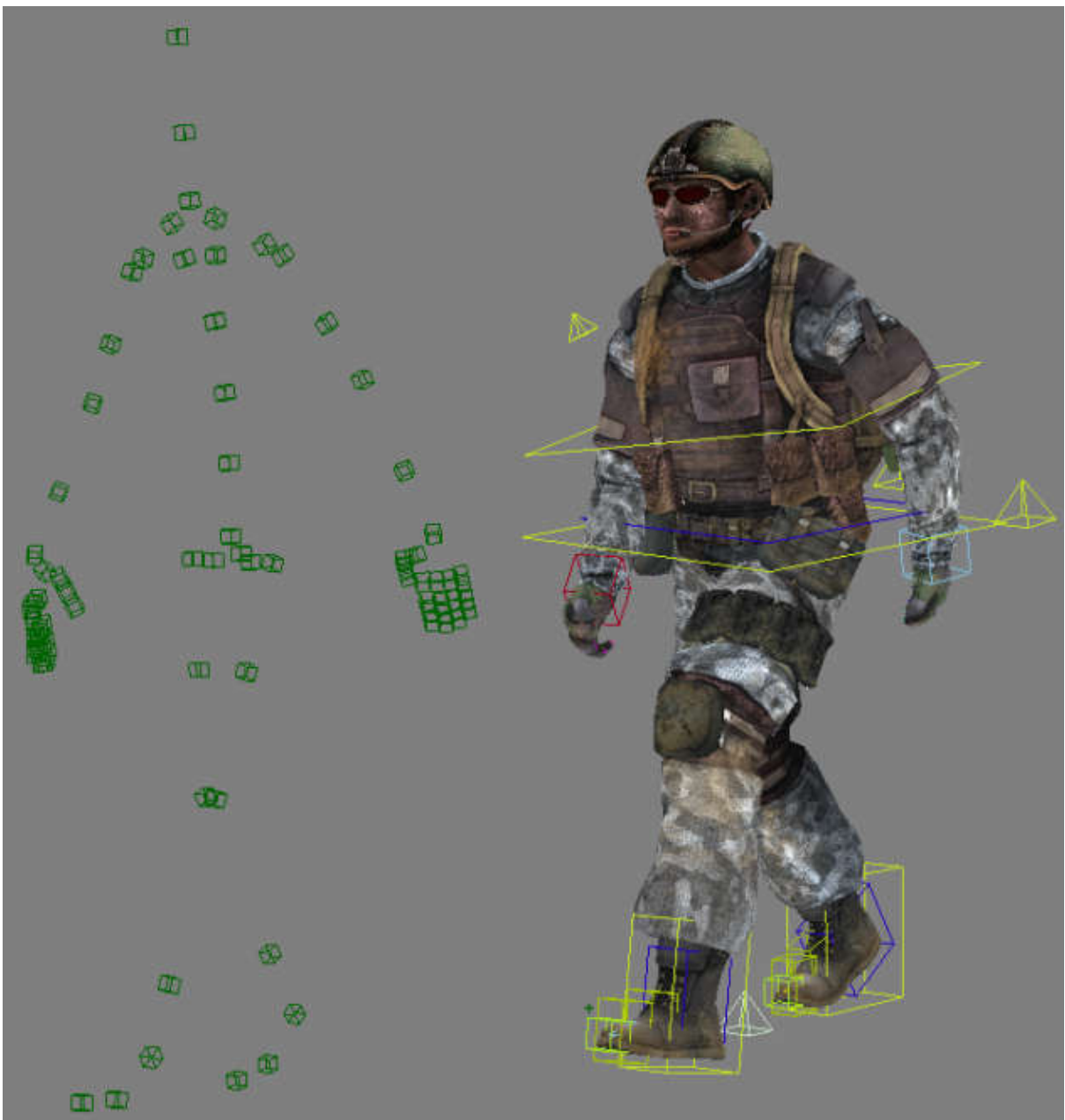
Format file C3D telah digunakan terus menerus sejak 1987. Format file data C3D asli dikembangkan oleh Dr. Andrew Dainis dan dijelaskan dalam spesifikasi file C3D yang diterbitkan yang tersedia dari situs web ini. Format C3D pertama kali digunakan di National Institutes of Health Biomechanics Laboratory di Bethesda, Maryland USA, dan sekarang menjadi format penyimpanan data 3D standar di seluruh dunia.



Desain format file C3D awalnya didorong oleh kebutuhan akan format yang akurat dan efisien untuk menyimpan data yang dikumpulkan secara andal dalam lingkungan motion capture. Format ini menyimpan data koordinat dan numerik 3D dari percobaan pengukuran apa pun, dengan semua berbagai parameter yang menjelaskan data, dalam satu file. Ini menghilangkan kebutuhan data terkait gerakan untuk bepergian dengan catatan tambahan dan informasi pengujian (yang biasanya dipisahkan dari data di beberapa titik dalam perjalanannya) agar dapat diakses.

Kemampuan untuk menyimpan banyak informasi tentang data adalah fitur yang membedakan format C3D dari setiap format biomekanik lainnya. Pada awal desain format C3D, disadari bahwa tidak mungkin satu spesifikasi yang ketat akan sesuai dengan setiap kebutuhan biomekanik. Akibatnya file C3D biasanya menyimpan sejumlah kecil parameter umum yang menggambarkan data 3D dan kemudian memungkinkan pengguna untuk menentukan, menghasilkan, dan menyimpan di dalam file sejumlah item data yang ditentukan pengguna atau lab sehingga siapa pun yang membuka file C3D dapat mengaksesnya.

Akibatnya, format C3D tidak terikat pada pabrik tertentu dan dapat disesuaikan secara bebas untuk menyimpan informasi yang dibutuhkan pengguna untuk penelitian dan analisis serta pemrosesan umum tanpa membuat komitmen terhadap lingkungan perangkat keras tertentu.



Motion capture dalam animasi karakter 3D (setyo Adi Nugroho)

format Ini menyimpan informasi koordinat 3D, data analog dan informasi terkait yang digunakan dalam pengambilan data gerak 3D dan operasi analisis selanjutnya. Pada saat pengembangannya, semua sistem motion capture 3D menyimpan data mereka dalam beberapa file, masing-masing dengan format kepemilikan yang berbeda, membuat pertukaran data antara berbagai laboratorium biomekanik dan analisis gaya berjalan menjadi sangat sulit. Dengan pengenalan dan adopsi format file C3D oleh semua perusahaan motion capture 3D utama, semua informasi 3D yang diperlukan, data analog, dan parameter yang menjelaskan data dapat ditransfer dengan mulus antara peneliti dan laboratorium, terlepas dari perangkat keras atau lingkungan yang digunakan untuk mengumpulkan data. Fitur utama format file C3D tercantum di bawah ini:

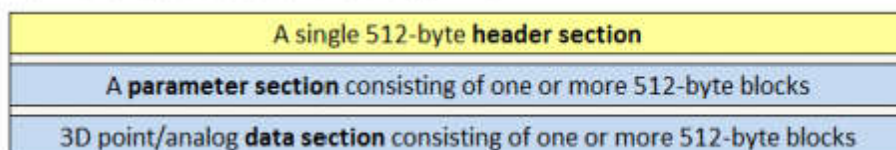
- Kemampuan untuk menyimpan data posisi dan analog 3D dalam bentuk yang diproses dan tidak diproses.
- Simpan informasi yang menjelaskan desain fisik laboratorium seperti saluran EMG yang digunakan, posisi pelat gaya, dan set penanda, dll.
- Simpan informasi uji coba yang berkaitan dengan keadaan sesi pengujian seperti laju sampel, nama file, tanggal, rekaman otot EMG, dll.,
- misalnya ID, usia saat percobaan, dengan parameter fisik seperti berat badan, panjang kaki, dll.
- Simpan hasil analisis yang dihitung seperti waktu berjalan, informasi siklus, dan informasi terkait.
- Ekstensibilitas - format C3D menyediakan kemampuan untuk menyimpan informasi baru tanpa membuat data lama menjadi usang.
- Spesifikasi publik dan deskripsi format C3D memungkinkan siapa saja untuk mengakses data tanpa bergantung pada produsen untuk mendapatkan informasi.

Sebelum pengenalan format file C3D, hampir semua perangkat lunak analisis biomekanik dan gaya berjalan ditulis untuk setiap format file produsen sistem 3D tertentu. Akibatnya, peneliti dan dokter dibatasi untuk menulis perangkat lunak analisis mereka sendiri atau hanya menggunakan perangkat lunak yang disediakan oleh sistem pengumpulan data 3D mereka. Pengenalan format C3D menghasilkan ketersediaan perangkat lunak pihak ketiga yang substansial dan membebaskan komunitas riset dari ketergantungan pada produsen sistem data 3D individu.

Format file C3D, sebaliknya dari kebanyakan sistem motion capture yang lain, tidak berbasis kerangka, dan biner.

Semua file C3D biasanya berisi tiga bagian informasi (header, parameter, dan data). Bagian header file C3D selalu merupakan blok 512 byte pertama dalam file.

File C3D harus dilihat terdiri dari blok yang panjangnya 512 byte (256 16-bit kata-kata) dan, sementara semua bagian dalam file C3D harus dimulai pada blok 512 byte batas, informasi yang disimpan dalam parameter dan bagian data dapat menyeberang batas blok individu.



```

Offset (h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000  02 50 1A 00 40 00 01 00 C2 01 0A 00 AA 3E AB AA
00000010  0B 00 04 00 48 43 00 00 00 00 00 00 00 00 00 00
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000000F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000120  00 00 00 00 00 00 00 00 00 00 39 30 03 00 00 00
00000130  2E 41 7B 14 AC 41 CD CC EA 41 71 3D 00 00 00 00
00000140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000170  00 00 00 00 00 00 00 00 01 01 01 00 00 00 00 00
00000180  00 00 00 00 00 00 00 00 00 00 00 00 52 49 43 20
00000190  52 48 53 20 52 54 4F 20 00 00 00 00 00 00 00 00
000001A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

4.4.4 FBX

FBX, FilmBox, adalah format file 3D populer yang awalnya dikembangkan oleh Kaydara untuk MotionBuilder. Itu diakuisisi oleh Autodesk Inc pada tahun 2006 dan sekarang menjadi salah satu format pertukaran 3D utama seperti yang digunakan oleh banyak aplikasi 3D. FBX tersedia dalam format file biner dan ASCII. Format ini dibuat untuk menyediakan interoperabilitas antara aplikasi pembuatan konten digital. Ada banyak alat yang tersedia untuk konversi dari/ke format file FBX.

Meskipun format FBX adalah hak milik autodesk, banyak aplikasi pemodelan dan animasi non-Autodesk dapat membuka file FBX. Ini memungkinkan pembuat untuk berbagi model 3D dalam format FBX, yang efisien karena menyimpan model sebagai data biner.

anda dapat mengimpor file FBX ke berbagai aplikasi pemodelan dan animasi 3D, termasuk Autodesk Maya (multiplatform) dan Blender (multiplatform). Namun, jika Anda hanya ingin melihat model yang berisi file FBX Anda dengan cepat, cara tercepat untuk melakukannya adalah dengan membukanya dengan Autodesk FBX Review (Windows, Mac, iOS) atau Autodesk Viewer (Web).

File FBX (.fbx) biasanya disimpan dalam format biner (atau asli), tetapi juga dapat disimpan dalam format ASCII. File FBX biner dan file FBX ASCII keduanya menggunakan ekstensi nama file .fbx.

File FBX menyimpan data tentang cameras, lights, meshes, NURBS, dan elemen lain dari pemandangan 3D. Aplikasi seperti Autodesk 3ds Max, Autodesk Maya, dan Autodesk MotionBuilder dapat mengimpor semua atau sebagian data pemandangan dari file FBX. Mereka juga dapat mengekspor data adegan mereka ke file FBX. Semua operasi ini menggunakan FBX SDK.

Ini adalah versi singkat dari file FBX kecil dalam format ASCII. Baris yang dihilangkan ditandai dengan ..., dan kita telah menambahkan beberapa komentar secara manual. Komentar dimulai dengan titik koma (";") di mana saja dalam satu baris.

```
; FBX 7.1.0 project file
; Copyright (C) 1997-2010 Autodesk Inc. and/or its licensors.
; All rights reserved.
; -----
FBXHeaderExtension: {
    ; header information: global file information.
    FBXHeaderVersion: 1003
    FBXVersion: 7100
    CreationTimeStamp: {
        Version: 1000
        Year: 2010
        Month: 1
        Day: 19
        Hour: 16
        Minute: 30
        Second: 28
        Millisecond: 884
    }
    Creator: "FBX SDK/FBX Plugins version 2011.2"
    SceneInfo: "SceneInfo::GlobalInfo", "UserData" {
    ...
    }
GlobalSettings: {
    Version: 1000
    Properties70: {
        P: "UpAxis", "int", "Integer", "", 1
        P: "UpAxisSign", "int", "Integer", "", 1
        P: "FrontAxis", "int", "Integer", "", 2
        P: "FrontAxisSign", "int", "Integer", "", 1
        P: "CoordAxis", "int", "Integer", "", 0
        P: "CoordAxisSign", "int", "Integer", "", 1
        P: "OriginalUpAxis", "int", "Integer", "", -1
        P: "OriginalUpAxisSign", "int", "Integer", "", 1
        P: "UnitScaleFactor", "double", "Number", "", 1
    }
}
```

```

P: "OriginalUnitScaleFactor", "double", "Number", "", 1
P: "AmbientColor", "ColorRGB", "Color", "", 0,0,0
P: "DefaultCamera", "KString", "", "", "Producer Perspective"
P: "TimeMode", "enum", "", "", 6
P: "TimeSpanStart", "KTime", "Time", "", 0
P: "TimeSpanStop", "KTime", "Time", "", 46186158000
}
}
...
; Object definitions
;-----
Definitions: {
  Version: 100
  Count: 2251
  ObjectType: "GlobalSettings" {
    Count: 1
  }
  ObjectType: "Model" {
    Count: 86
    PropertyTemplate: "FbxNode" {
      Properties70: {
        P: "QuaternionInterpolate", "bool", "", "", 0
        P: "RotationOffset", "Vector3D", "Vector", "", 0,0,0
        P: "RotationPivot", "Vector3D", "Vector", "", 0,0,0
        P: "ScalingOffset", "Vector3D", "Vector", "", 0,0,0
        P: "ScalingPivot", "Vector3D", "Vector", "", 0,0,0
      ...}
    }
  }
  ObjectType: "Material" {
    Count: 1
    PropertyTemplate: "FbxSurfacePhong" {
      Properties70: {
        P: "ShadingModel", "KString", "", "", "Phong"
        P: "MultiLayer", "bool", "", "", 0
        P: "EmissiveColor", "ColorRGB", "Color", "", 0,0,0
        P: "EmissiveFactor", "double", "Number", "", 1
        P: "AmbientColor", "ColorRGB", "Color", "", 0.2,0.2,0.2
      ...}
    }
  }
  Model: 21883936, "Model::Humanoid:Hips", "LimbNode" {
    Version: 232
    Properties70: {
      P: "ScalingMin", "Vector3D", "Vector", "", 1,1,1
      P: "NegativePercentShapeSupport", "bool", "", "", 0
      P: "DefaultAttributeIndex", "int", "Integer", "", 0

```

```
P: "Lcl Translation", "Lcl Translation", "", "A+",-271.281097412109,-762.185852050781,528.336242675781
```

```
P: "Lcl Rotation", "Lcl Rotation", "", "A+",-1.35128843784332,2.6148145198822,0.42334708571434
```

```
P: "Lcl Scaling", "Lcl Scaling", "", "A+",1,0.99999988079071,1
```

...

Kit Pengembangan Perangkat Lunak FBX (FBX SDK) memungkinkan pengembang perangkat lunak untuk membuat aplikasi yang menggunakan teknologi FBX, atau untuk mengintegrasikan teknologi FBX ke dalam aplikasi mereka yang sudah ada.

Perhatikan bahwa SDK FBX tidak dicakup oleh GNU General Public License (GPL), dan kode sumbernya tidak tersedia untuk umum. Beberapa kode sumber memang ada untuk menyesuaikan plugin Maya dan 3ds Max FBX, meskipun dikemas dalam SDK Ekstensi FBX, dan bukan dalam SDK FBX itu sendiri. Rilis terbaru dan versi FBX SDK sebelumnya dapat diperoleh di situs web Autodesk FBX di <http://www.autodesk.com/fbx>. FBX SDK tidak dapat didistribusikan ulang atau dikemas ulang tanpa izin tertulis dari Autodesk. Jika Anda ingin mendistribusikan proyek sumber terbuka yang menggunakan SDK FBX, Anda harus menyertakan tautan ke situs web FBX Autodesk sehingga pengguna dapat menginstal versi SDK FBX yang diperlukan.

Pembuat model dapat mengonversi model ke format FBX menggunakan Blender (multiplatform) atau Konverter FBX Autodesk (Windows dan Mac, tetapi tidak didukung sejak 2013). Misalnya, Autodesk FBX Converter dapat mengonversi file .OBJ, .DXF, .3DS, dan .DAE (COLLADA) ke format FBX. CATATAN: Kaydara awalnya mengembangkan format FBX untuk digunakan di Kaydara MotionBuilder. Autodesk mengakuisisi Kaydara pada tahun 2006.

Anda dapat menggunakan Autodesk FBX Converter (Windows, Mac) atau Blender (multiplatform) untuk mengonversi file FBX ke model 3D dan format scene lainnya. Misalnya, Blender dapat mengonversi file FBX ke format berikut:

- [.DAE](#) - Digital Asset Exchange
- ABC - Alembic Scene
- [.USD](#) - Universal Scene Description Format
- [.PLY](#) - Polygon Model
- [.STL](#) - Stereolithography Model
- [.GLB](#), [.GLTF](#) - GL Transmission Format
- [.OBJ](#) - Wavefront 3D Object
- [.X3D](#) - Xara3D Project

4.5 kesimpulan

Sejumlah format file motion capture yang digunakan saat ini telah dibahas dan beberapa sistem yang lebih populer telah dibahas secara rinci. Namun, karena setiap format berisi data yang cukup untuk menampilkan animasi, apakah ada format yang lebih baik dari yang lain? Jawaban atas pertanyaan ini tergantung pada aplikasi yang dimaksudkan dari data gerak, yang dapat bervariasi dari analisis, pengeditan, atau pemutaran sederhana.

Ketika hanya memperhatikan tampilan animasi, aspek yang diinginkan dari data adalah kekompakan dan posisi absolut untuk anggota badan, yang menghilangkan kebutuhan akan algoritma rendering hierarkis. Ini berarti baik untuk membuat dan menampilkan frame ke layar. Properti yang paling berguna untuk pengeditan karakter adalah struktur hierarkis sehingga tulang dapat dipertimbangkan secara lokal dan terlepas dari perubahan yang dilakukan pada anggota badan induk, tulang child selalu mempertahankan hubungan transformasional tertentu. Semua format motion capture modern didasarkan pada struktur hierarkis, yang mencakup format ASF/AMC, BVH, dan HTR. Dari 3 format file tersebut, hierarki file BVH bisa dibilang lebih kompak dan ringkas, namun ini berarti datanya tidak dipecah menjadi sedetail format ASF/AMC atau HTR. Seperti yang telah dibahas sebelumnya, masalah dengan hierarki file BVH adalah bahwa ia mengasumsikan panjang tulang sepanjang vektor sebagai lawan skalar, namun format ASF/AMC dan HTR keduanya menentukan nilai rotasi dan skalar untuk mendapatkan vektor tulang ekuivalen BVH.

Setelah properti data hierarkis, properti lain yang terbukti berguna selama pengeditan gerakan mencakup detail tentang lingkungan tempat data diambil. Contoh data lingkungan termasuk sumbu gravitasi atau bahkan sifat yang lebih mendasar seperti satuan metrik yang digunakan. Format BVH gagal menangkap salah satu detail ini, namun format HTR memberikan tingkat keberhasilan yang lebih baik, sementara ASF/AMC menyediakan lebih banyak data daripada file HTR dengan menyertakan detail tentang batas channel.

Dengan gerakan apapun dalam format hierarkis, algoritma rendering perlu mencerminkan hal ini oleh karena itu prosedur rekursif perlu diterapkan untuk merender animasi ke layar. Gaya rendering ini lebih mahal daripada hanya mendorong koordinat titik ke dalam pipa grafik, jadi dalam banyak kasus setelah pengeditan gerakan apa pun telah dicapai dan tidak ada pengeditan lebih lanjut yang akan dilakukan, file dapat dikonversi ke versi optimal yang digunakan hanya untuk menampilkan .

Sebagai properti tambahan yang perlu dipertimbangkan untuk format pengeditan gerakan, masih menguntungkan untuk memiliki data dalam format yang ringkas. Namun saat ini format cenderung dalam format berbasis ASCII daripada biner yang mengambil rata-rata 3 kali ruang data biner, tapi setidaknya membuat mereka dapat dibaca manusia. Format biner selanjutnya akan membantu dalam penguraian file karena sejumlah besar float akan direpresentasikan sebagai representasi biner mereka sebagai lawan dari berbasis teks yang perlu diubah menjadi representasi float sebelum digunakan. Namun perlu dicatat pada titik ini

BAB 5 SUMBER DATA MOTION CAPTURE

Dalam proses untuk mendapatkan data motion capture dibutuhkan peralatan yang mahal dan proses yang rumit. tapi bagaimana jika kita ingin sekedar memanfaatkan motion capture dalam animasi 3 dimensi kita. solusinya adalah mencari data yang sudah ada dan siap dipakai.

Data motion capture banyak terdapat di internet siap pakai dan tinggal diimplementasikan dalam aplikasi animasi karakter 3 dimensi . File motion capture dapat dibeli dari banyak sumber komersial, tetapi sejumlah besar file mocap juga tersedia untuk diunduh gratis. Berikut adalah beberapa situs yang menyediakan data motion capture gratis.

5.1 CMU Graphics Lab Motion Capture Database

Dihosting di Carnegie-Mellon University, ini adalah perpustakaan besar file mocap yang dapat diunduh secara gratis. Alamat webnya adalah <http://mocap.cs.cmu.edu>.

CMU Graphics Lab Motion Capture Database
[Home](#) | [Search](#) | [Tools](#) | [Info](#) | [FAQs](#) | [Rendered Movies](#) | [Resources](#)

View All: Subjects Motions	Browse: Motion Categories	Search Help <input type="text"/> <input type="text"/> <input type="text"/> <small>subject number motion or keyword</small> <small>(e.g. 41) (e.g. run)</small>
--	---	--

Welcome to the Carnegie Mellon University Motion Capture Database!
This dataset of motions is free for all uses. Search above by subject # or motion category. Check out the "Info" tab for information on the mocap process, the "FAQs" for miscellaneous questions about our dataset, or the "Tools" page for code to work with mocap data. Enjoy!

The collection of the data in this database was supported by NSF Grant #0196217.

NEW! More easy-to-use motion capture formats! Bruce Hahn has released both 3D Studio Max and MotionBuilder bvh's, hosted at [cgspeed](http://cgspeed.com). The site includes links to our data in other formats, too. See the "Resources" tab above.

NEW! The Quality of Life Grand Challenge: Kitchen Capture dataset is now up. This capture of kitchen activities includes mocap, microphones, accelerometers, and more: <http://kitchen.cs.cmu.edu>.

NEW! More data formats are here! See the "Resources" tab for a list of external sites that have converted our data into numerous other formats. Check out the "Info" tab above for an extended explanation of motion capture file formats, and our motion capture setup. We've also added new "Tools" for working with both .asf/.amc and Vicon's .vsk/.v files.

data disini terdapat **2605** gerakan dalam **6** categories dan **23** subcategories.

CMU Graphics Lab Motion Capture Database

Home | Search | Tools | Info | FAQs | Rendered Movies | Resources

View All: [Subjects](#) | [Motions](#) | [Motion Categories](#)

Browse:

Search Help: subject number (e.g. 41) | motion or keyword (e.g. run)

Trial #	Subject 18 - Motion Description
1	walk, shake hands (2 subjects - subject A)
2	walk, shake hands (2 subjects - subject A)
3	A pulls B; B resists (2 subjects - subject A)
4	A pulls B; B resists (2 subjects - subject A)
5	A pulls B by the elbow; B resists (2 subjects - subject A)
6	A pulls B by the elbow; B resists (2 subjects - subject A)
7	navigate busy sidewalk; A leads the way, takes B by the arm (2 subjects - subject A)
8	conversation - explain with hand gestures (2 subjects - subject A)
9	conversation - explain with hand gestures, walk (2 subjects - subject A)
10	quarrel - angry hand gestures (2 subjects - subject A)
11	quarrel - angry hand gestures (2 subjects - subject A)
12	friends meet, hang out; A sits, B joins A (2 subjects - subject A)
13	run, scramble for last seat (2 subjects - subject A)
14	run, scramble for last seat (2 subjects - subject A)
15	clucken dance (2 subjects - subject A)

Trial # [Subject 19](#) - Motion Description

website ini berisi gerakangerakan umum yang ada dalam kehidupan yang bisa kita implementasikan dalam animasi kita

CMU menghosting file mocap dalam tiga format: tvd, c3d, dan amc.

5.1.1 data motion capture BVH

Namun, beberapa aplikasi yang dapat memanfaatkan mocap hanya dapat membaca file BVH, jadi tidak satu pun dari file ini yang dapat digunakan secara langsung.

Untungnya, B. Hahne di <http://www.cgsped.com> telah mengonversi file CMU ke BVH. Berkas yang dikonversi ada di <http://sites.google.com/a/cgsped.com/cgsped/motion-capture>.

cgsped

Motion Capture

This page contains links and information about Motion Capture software and datasets.

BVH conversions of the 2500-motion Carnegie-Mellon motion capture dataset:

- Daz-friendly version** (released July 2010, by B. Hahne)
- 3dsMax-friendly version** (released May 2009, by B. Hahne)
- MotionBuilder-friendly version** (released July 2008, by B. Hahne)

Tutorials on how to use the BVH conversions:

- Using the 3dsMax-friendly version in 3dsMax by B. Hahne
- Using the Daz-friendly BVH release within Daz Studio - videos released July 2010 by B. Hahne
 - [Daz-friendly BVH tutorial video part 1](#)
 - [Daz-friendly BVH tutorial video part 2](#)

Other motion capture datasets:

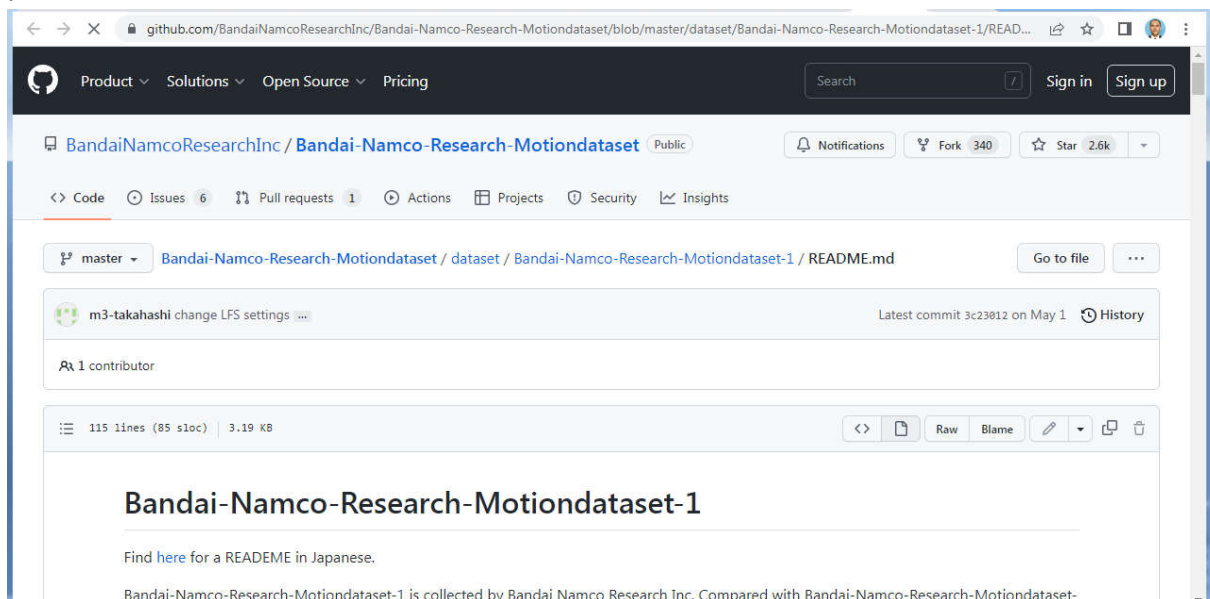
BVH and related files from the Ohio State University ACCAD [Open Motion Project](#) [link broken as of Aug. 2021]

www.mocapdata.com

Di dalam website ini terdapat hasil konversi yang cocok untuk mengaplikasikan mocap kedalam software yang hanya bisa menerima BVH. tersedia untuk DAZ , 3ds max dan Motionbuilder untuk pengguna blender bisa memanfaatkan hasil konversi cocap untuk 3ds max tanpa ada kendala apa apa.

5.2 Bandai Namco

Bandai Namco Research, divisi R&D dari penerbit dan pengembang game Bandai Namco, telah merilis perpustakaan besar dari data Motion capture gratis untuk penelitian dan penggunaan pribadi.



Data, direkam di studio mocap milik Bandai Namco, terdiri lebih dari 3.000 gerakan individu. File disediakan dalam format BVH dan mencakup berbagai kegiatan, termasuk berjalan, berlari, berkelahi dan menari, dan berbagai gaya gerak, termasuk aktif, lelah, dan bahagia. library ini juga disertai dengan skrip gratis untuk memvisualisasikan data di Blender. Lebih dari 3.000 gerakan mocap gratis dalam format BVH, direkam untuk penelitian animasi

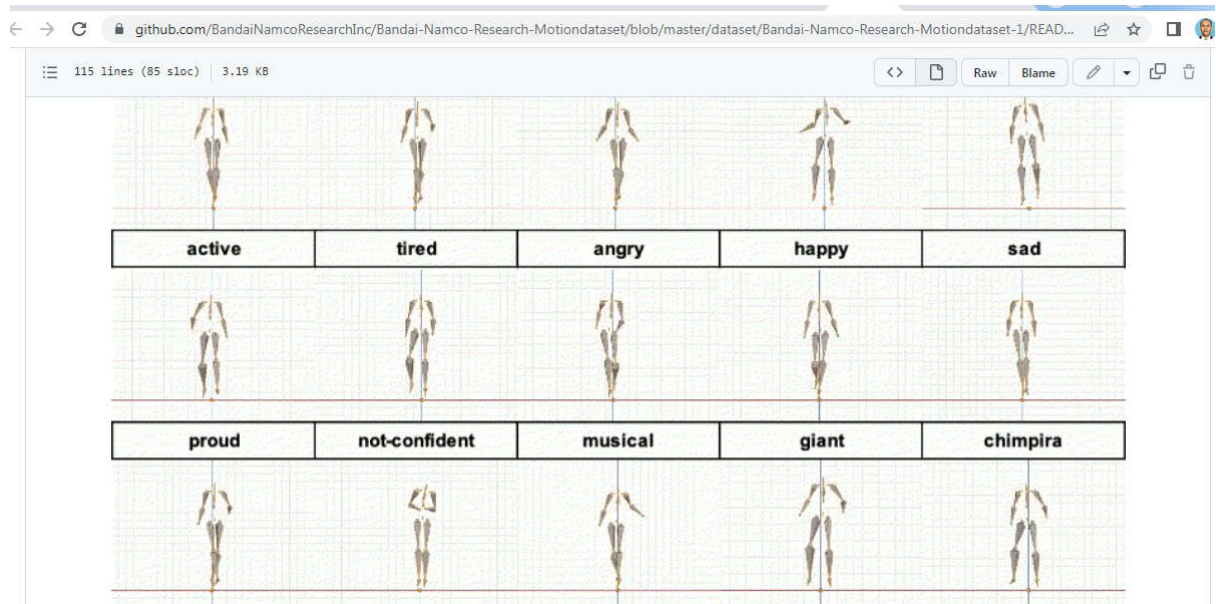
Pustaka terdiri dari dua kumpulan data terpisah, yang pertama terdiri dari 175 gerakan individu yang mencakup 17 jenis gerakan, termasuk berkelahi dan menari, dan 15 gaya, meniru gerakan manusia, raksasa, dan kera.

Yang kedua jauh lebih besar, berisi 2.902 gerakan individu, tetapi berfokus pada berjalan, berlari, dan melambai, dalam 7 gaya inti: normal, aktif, dan kelelahan; muda dan tua; maskulin dan feminin. Bersama-sama, mereka terdiri dari 3.077 gerakan individu, dan 421.604 frame gerakan.

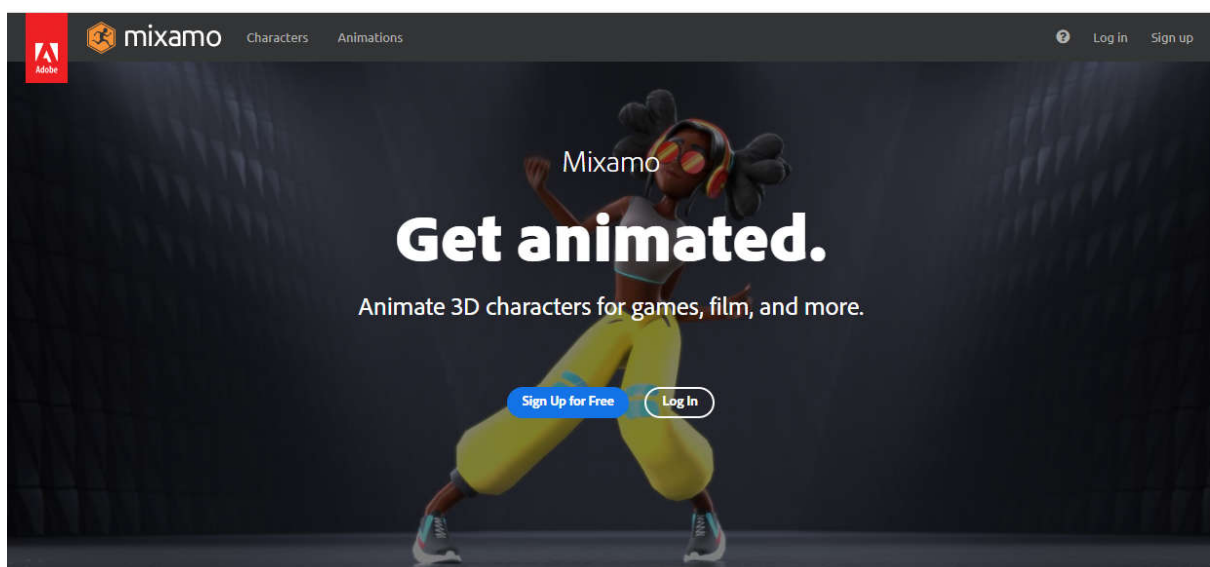
Gerakan ditangkap dari tiga aktor profesional, dengan pasca-pemrosesan data mentah Bandai Namco untuk menghilangkan noise dan menyelaraskan proporsi ke kerangka karakter standar, dan dipasok pada 30fps.

Kompatibel dengan sebagian besar aplikasi DCC dan mesin game File disediakan dalam format BVH, didukung secara native oleh banyak aplikasi DCC, termasuk 3ds Max, Blender, Cinema 4D, Houdini dan MotionBuilder, dan dalam bahasa Maya melalui skrip impor gratis.

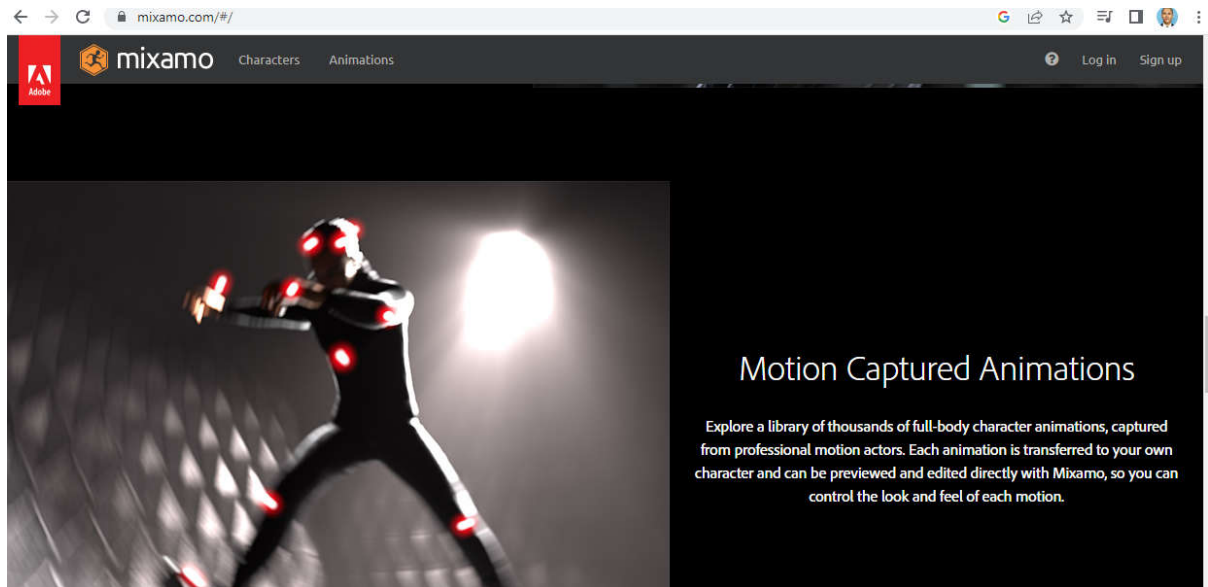
Animasi kemudian dapat diekspor dalam format FBX untuk digunakan di mesin game seperti Unity dan Unreal Engine. Selain itu, Bandai Namco telah menyediakan skrip Python gratisnya untuk memvisualisasikan data BVH di Blender. File tersedia gratis untuk penelitian dan penggunaan pribadi di bawah lisensi Creative Commons Attribution Non-Commercial No Derivatives .



5.3 mixamo



Mixamo adalah perusahaan teknologi grafis komputer 3D. Berbasis di San Francisco, perusahaan mengembangkan dan menjual layanan berbasis web untuk animasi karakter 3D. Teknologi Mixamo menggunakan metode pembelajaran mesin untuk mengotomatiskan langkah-langkah proses animasi karakter, termasuk pemodelan 3D hingga rigging dan animasi 3D.



mixamo memiliki perpustakaan ribuan animasi karakter seluruh tubuh, diambil dari aktor gerak profesional. Setiap animasi ditransfer ke karakter Anda sendiri dan dapat dipratinjau dan diedit langsung dengan Mixamo, sehingga Anda dapat mengontrol tampilan dan nuansa setiap gerakan. untuk pemanfaatan mixamo ini akan dibahas terpisah dalam bab lain di buku ini

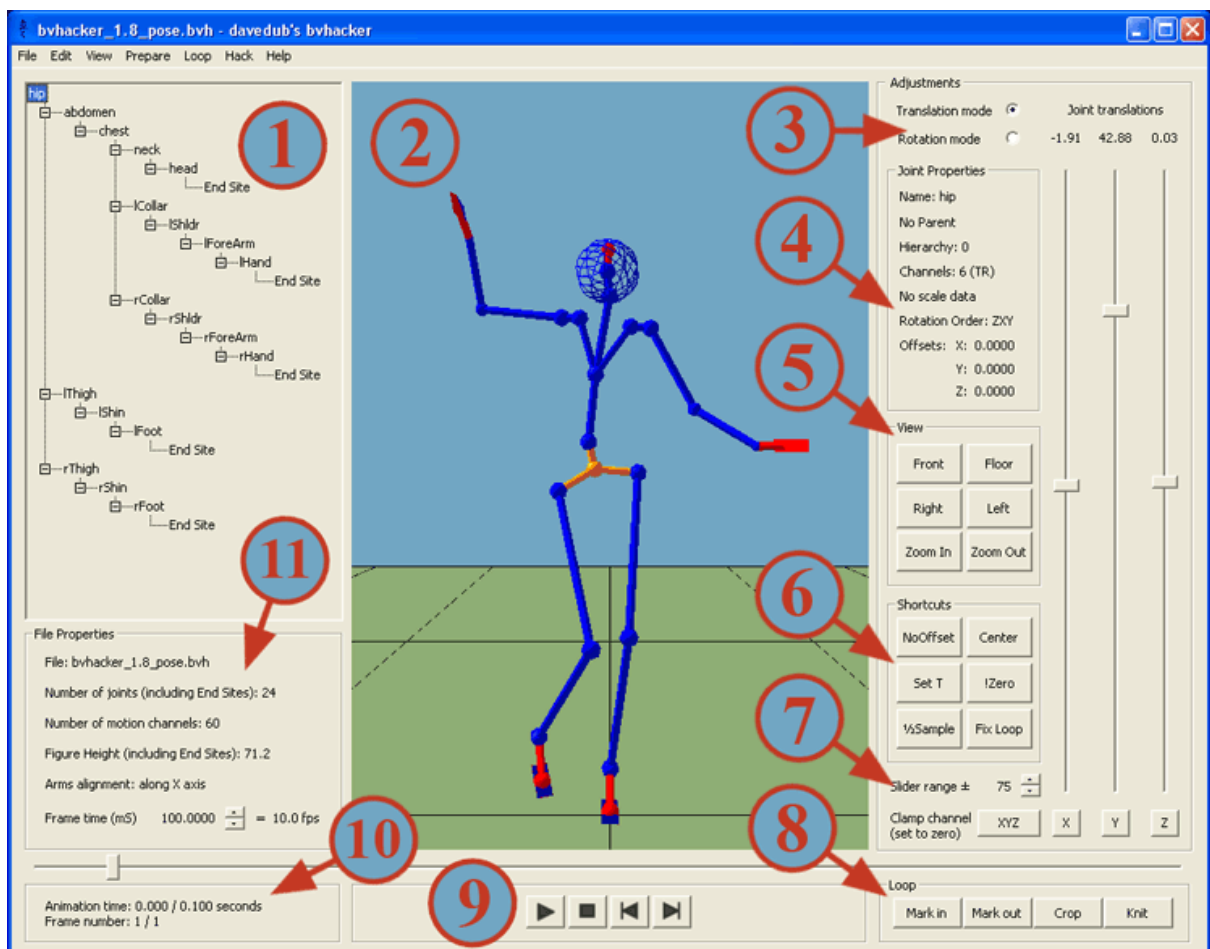
BAB 6 MENGEDIT MOTION CAPTURE BVH

BVH adalah format yang sangat umum dalam pemanfaatan motion capture dalam animasi tapi dalam data yang diperoleh biasanya tidak selalu sesuai dengan keinginan dan kebutuhan animasi kita, maka kita perlu mengeditnya lebih dahulu. dengan mendedit motioncapture sebelum diterapkan dalam animasi akan lebih tepat dan sesuai kebutuhan.

6.1 BVHACKER

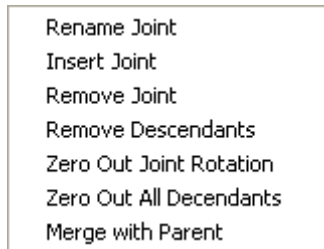
program yang sangat membantu dalam mengedit motion capture khususnya dalam format BVH adalah BVHACKER

berikut ini tampilan dari BVHACKER



Bagian 1: Scene browser

Tampilan ini menunjukkan hierarki joint yang terdapat dalam file bvh. Sendi dipilih dari sini. joint dapat ditambahkan, dihapus, diganti namanya, dinolkan atau digabungkan dengan Parentnya dengan mengklik kanan dan memilih opsi yang sesuai dari menu konteks:



Memasukkan, Menghapus atau Mengganti nama joint juga dapat dilakukan dengan menggunakan tombol pintas yang sesuai (Ins, Del, masing-masing F2). joint berikutnya / sebelumnya di browser adegan dapat dinavigasi menggunakan tombol cursor atas dan bawah.

Bagian 2: Jendela tampilan utama

Jendela OpenGL tempat animasi ditampilkan. Bidang lantai dan warna latar belakang dapat disesuaikan dari panel tampilan dan menu tampilan. Navigasi dasar diimplementasikan melalui klik dan seret tombol mouse.

Kontrol:

- Tombol kiri mouse: Kamera Dolly (putar)
- Tombol mouse kanan: Kamera zoom
- Tombol tengah mouse: Kamera miring

Bagian 3: Dynamic slider controls

- Dynamic slider controls untuk menampilkan dan menyesuaikan nilai translasi atau rotasi X,Y, dan Z dari joint yang dipilih saat ini
- Nilai X,Y dan Z dapat diedit dengan mengklik dua kali salah satu nilai XY atau Z dan mengetikkan nilai baru
- Tombol Radio untuk pemilihan mode penyesuaian rotasi atau terjemahan. Juga mempengaruhi tombol penjepit. (Pintasan keyboard: 'R' dan 'T')

Bagian 4: Joint properties

Menampilkan rincian untuk joint yang dipilih saat ini

- Joint Name : Nama joint yang saat ini dipilih, sebagaimana ditentukan dalam file bvh
- Parent: Nama joint yang saat ini dipilih Parent
- Joint hierarchy value, (mulai dari nol untuk pinggul/ROOT)
- Channel: Tiga saluran data untuk setiap saluran TRS (T=translasi, R=rotasi, S=Skala)
- Scale: Jika file bvh memiliki data Skala, maka ditampilkan di sini
- Rotation Order: Urutan saluran gerakan untuk joint (biasanya ZXY)
- Offsets: offset joint ini dari Parentnya (dapat diedit)

Bagian 5: Pengaturan

- tampilan Default: menunjukkan tampilan tepat di depan gambar, sejajar dengan pinggul gambar

- Lantai: memberikan tampilan di lantai untuk memudahkan pengaturan tinggi gambar dalam kaitannya dengan tanah
- Right: menampilkan gambar dari sisi kanan
- Left: menampilkan gambar dari sisi kiri
- Zoom In: Memperbesar tampilan
- Zoom Out: Memperkecil tampilan

Bagian 6 : shortcut

- NoOffset: Nol v alues untuk offset pinggul. (Pintasan keyboard 'N')
- center: Memusatkan pinggul kerangka di panel X dan Z pada frame pertama (Pintasan keyboard 'C')
- Set T : Menambahkan frame referensi di awal animasi dengan semua nilai rotasi sendi disetel ke nol (Pintasan keyboard 'S')
- !Zero: Mencegah SL mengabaikan joint dengan rotasi nol di frame pertama dengan menyetel rotasi kecil. Harus digunakan sebelum menambahkan kerangka referensi t-stance. (Pintasan keyboard 'Z') Contoh: Membuat sampel ulang animasi menjadi setengah dari frame asli per detik (fps)
- Fix Loop: Perbaiki otomatis untuk Second Life ['Frame pertama animasi yang diunggah diduplikasi'](#) bug (Pintasan keyboard 'F')

Bagian 7: Slider range and zero out channel buttons

- Clamp buttons (X,Y,Z atau semua): Mengatur semua nilai ke nol untuk joint yang dipilih saat ini. Menjepit terjemahan jika mode terjemahan dipilih, menjepit rotasi jika mode rotasi dipilih
- Slider range: Mengatur rentang \pm untuk penggeser dinamis

Bagian 8: Animation cropping and looping commands

- Mark in & Mark outi: Mengatur titik awal dan akhir untuk fungsi pemotongan
- crop: Hapus semua frame di luar rentang yang ditentukan oleh titik Tandai masuk dan Tandai
- Knit: Padukan awal dan akhir animasi untuk memperlancar transisi saat memutar animasi dalam satu lingkaran

Bagian 9: Transport controls

- slider control bertindak sebagai kursor untuk mengesek melalui animasi
- Standard play, pause, stop, step forward dan step back buttons
- Pintasan keyboard untuk kontrol transport adalah:

Putar / jeda: spasi

Langkah maju: tombol kursor kanan

Langkah mundur: tombol kursor kiri

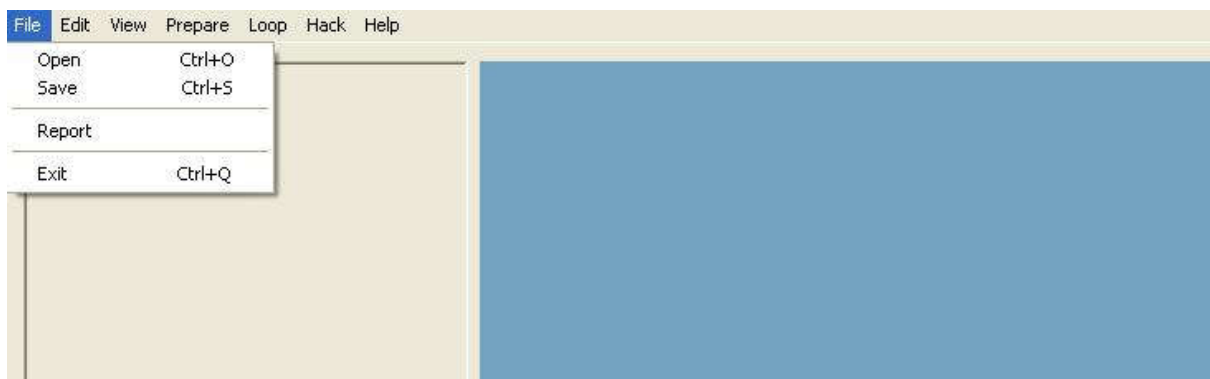
Bagian 10: Waktu animasi dan detail frame

- Animasi time: Menampilkan detik yang berlalu dan jumlah total detik
- Nomor frame: Menampilkan frame saat ini dan jumlah total frame

Bagian 11: Properti

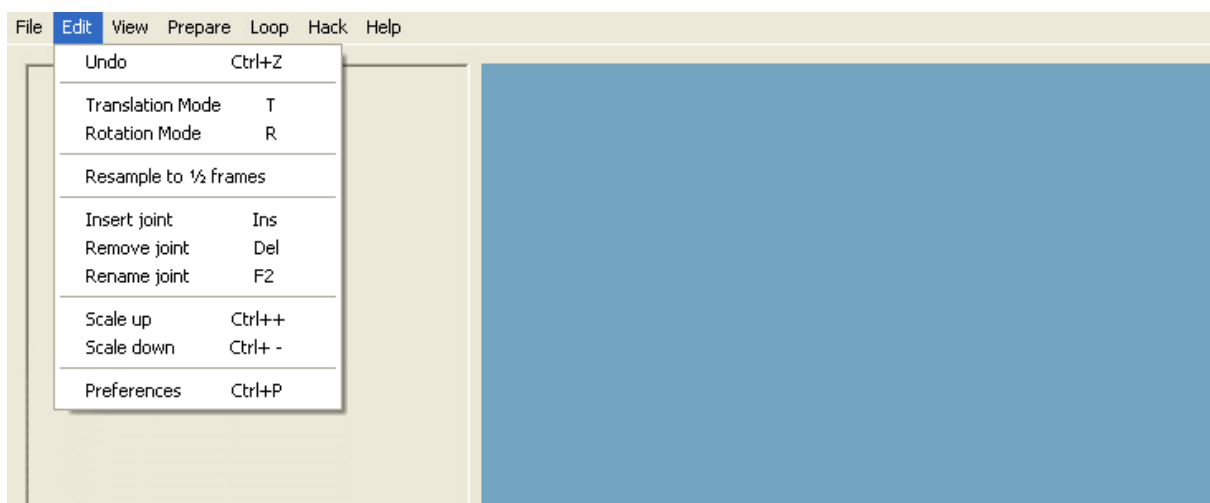
- file nama file bvh
- Number of joints. Termasuk Situs Akhir
- number motion channels. Biasanya tiga untuk setiap rotasi sendi, ditambah tiga untuk nilai translasi pinggul
- Figure height(jumlah nilai y dari situs ujung pinggul ke kepala ditambah jumlah semua nilai y dari situs ujung pinggul ke kaki)
- Arms alignment: Sumbu di mana lengan sejajar sendiri ketika semua rotasi joint disetel ke nol
- time frame: Jumlah milidetik per frame (dapat disesuaikan). Juga menampilkan frame animasi per detik (fps)

File



- Open: Membuka file bvh untuk diedit. File juga dapat diseret dan dijatuhkan ke bvhacker untuk pembukaan cepat (Pintasan keyboard 'Ctrl O')
- Simpan: Menyimpan file bvh saat ini ke disk (Pintasan keyboard 'Ctrl S')
- Laporan: Menghasilkan laporan yang memberikan peringatan dan kemungkinan masalah untuk file disiapkan untuk SL
- Keluar: Menutup bvhacker (Pintasan keyboard 'Ctrl Q')

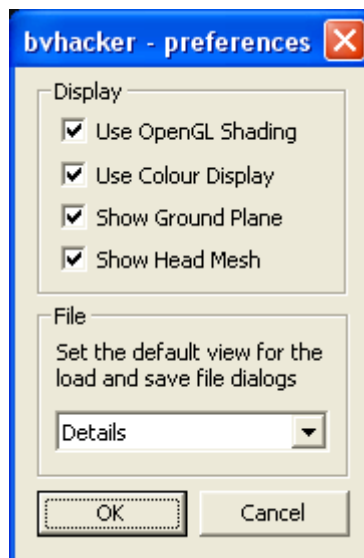
Edit



- Undo: Membatalkan operasi terakhir (Pintasan keyboard Ctrl-Z)
- Mode terjemahan: Mengatur penggeser X,Y,Z dan tombol penjepit untuk menyesuaikan terjemahan untuk saat ini joint yang dipilih (Pintasan keyboard 'T')

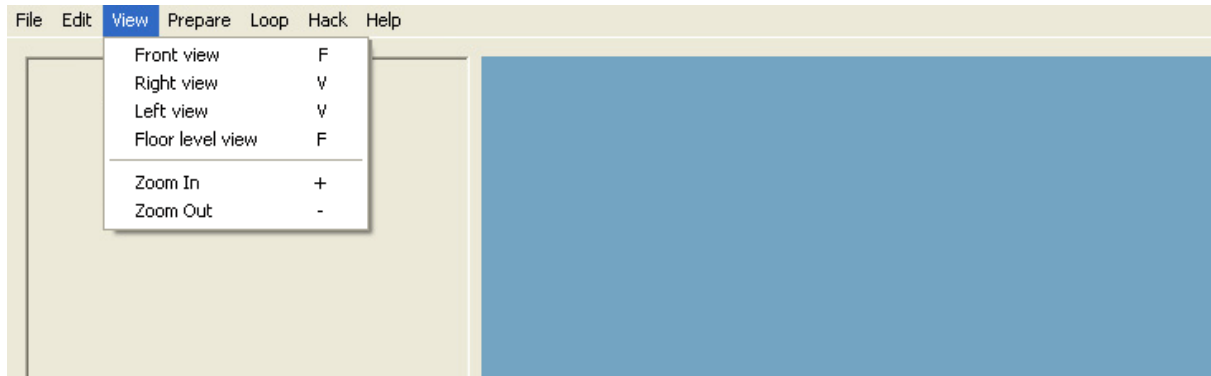
- Mode rotasi: Mengatur penggeser X,Y,Z dan tombol penjepit untuk menyesuaikan rotasi untuk joint yang dipilih saat ini (Pintasan keyboard 'R')
- Contoh ulang ke frame: Membuat sampel ulang animasi menjadi setengah frame asli per detik (fps)
- Insert joint: Menyisipkan joint di atas joint yang dipilih saat ini (Pintasan keyboard 'Sisipkan')
- delete joint: Menghapus joint yang dipilih saat ini (Pintasan keyboard 'Hapus')
- rename joint: Mengganti nama joint yang dipilih saat ini (Keyboard pintasan 'F2')
- Skala up: Menskalakan semua nilai terjemahan gabungan dengan faktor dua (Pintasan keyboard 'Ctrl +')
- Menurunkan: Menskalakan semua nilai terjemahan gabungan hingga setengahnya (Pintasan keyboard 'Ctrl -')
- Preferences: Menampilkan dialog preferensi (Pintasan keyboard 'Ctrl P')

Dialog Preferensi



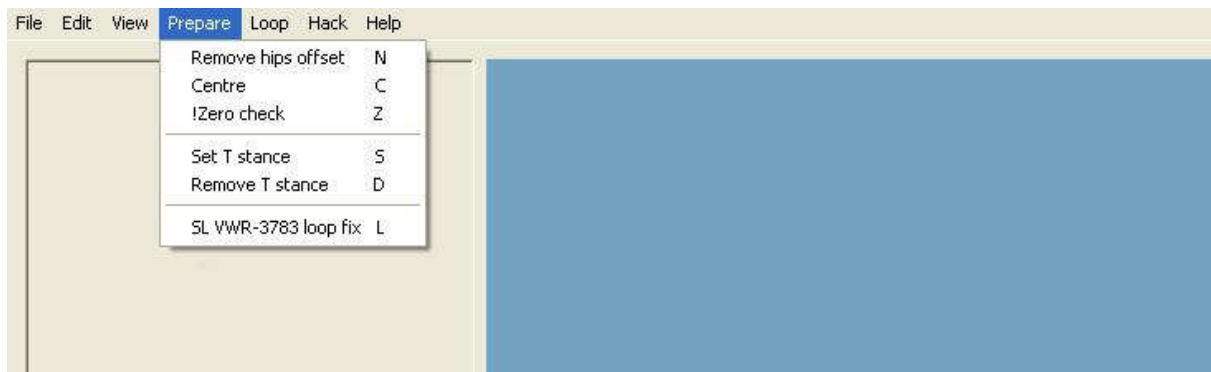
- Gunakan OpenGL Shading: Gunakan shading dan lighting di tampilan utama
- Gunakan Color Display: Beralih antara Warna dan Hitam & Putih warna tampilan
- Show Ground Plane: Beralih tampilan ground plane
- Show Head Mesh: Beralih tampilan head mesh
- Default Tampilan File: Atur Default tampilan untuk membuka dan menyimpan dialog file

View



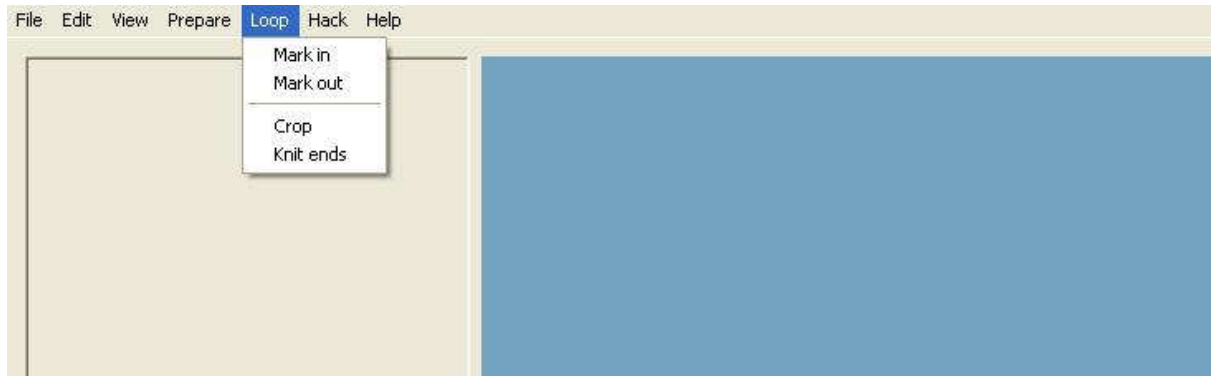
- Front view: Menampilkan tampilan tepat di depan gambar, sejajar dengan pinggul gambar
- Tampilan kanan: Menampilkan gambar dari sisi kanan
- Tampilan kiri: Menampilkan gambar dari sisi kiri
- Tampilan lantai : Memberikan tampilan pada tingkat lantai untuk memudahkan pengaturan tinggi gambar dalam kaitannya dengan tanah
- Shading: Gunakan shading OpenGL sederhana pada kerangka
- Zoom In: Perbesar tampilan
- Zoom Out: Perkecil tampilan

Persiapan



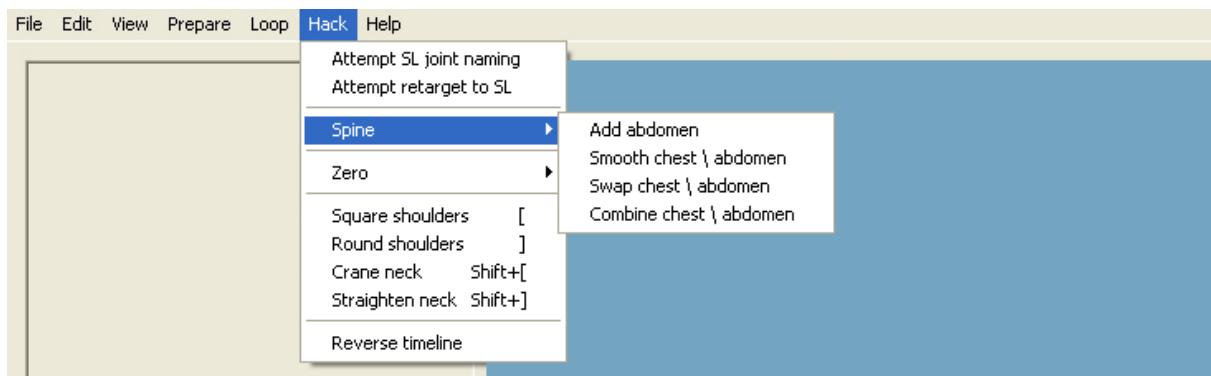
- Hapus offset pinggul: Menghilangkan nilai offset dari nilai saluran pinggul, jika disetel (Pintasan keyboard 'N')
- Tengah: Pusatkan kerangka di panel X dan Z pada frame pertama (Pintasan keyboard 'C')
- !Zero check: Mencegah SL mengabaikan joint dengan rotasi nol di frame pertama dengan mengatur putaran kecil. Harus digunakan sebelum frame t-stance (referensi) ditambahkan. (Pintasan keyboard 'Z')
- Setel posisi T: Menambahkan frame referensi di awal animasi dengan semua nilai rotasi gabungan disetel ke nol (Pintasan keyboard 'T')
- Hapus posisi T: Menghapus frame pertama animasi (Pintasan keyboard 'D')
- SL VWR-3783 loop fix: Perbaiki otomatis untuk Second Life ['Frame pertama dari animasi yang diunggah diduplikasi'](#) bug (Pintasan keyboard 'L')

Loop



- Mark in: Mengatur frame awal untuk fungsi pemotongan
- Tandai: Atur end frame untuk fungsi crop
- Crop: Hapus semua frame di luar rentang yang ditentukan oleh titik Mark in dan Mark out
- Knit end: Padukan awal dan akhir animasi untuk memuluskan transisi saat memutar animasi dalam satu lingkaran.

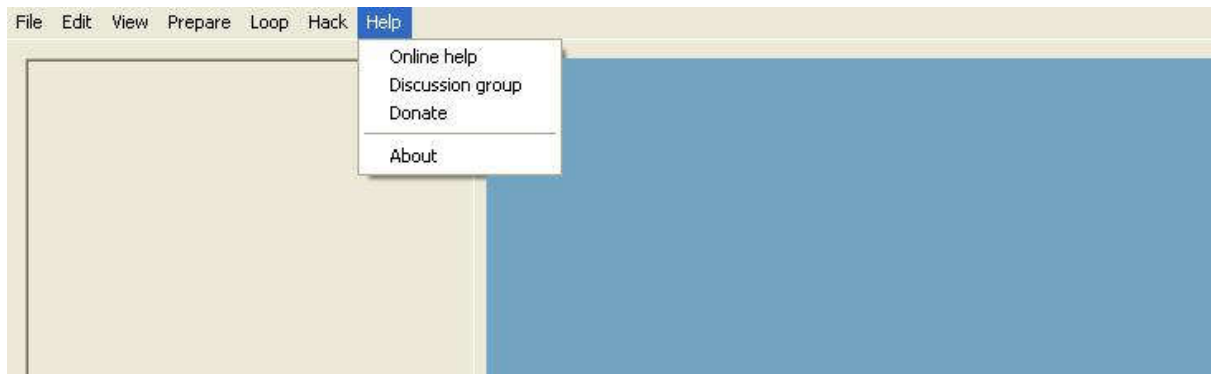
Hack



- Attempt SL joint renaming : Upaya untuk mengubah nama joint agar sesuai dengan konvensi penamaan joint SL
- Attempt retarget to SL: Upaya untuk mengkonfigurasi ulang joint ke kerangka SL dengan menyesuaikan ukuran, kesejajaran lengan, dan urutan rotasi.
- Spine submenu->Add abdomen: Menambahkan sendi perut antara pinggul dan sendi tulang belakang pertama
- Spine submenu->Swap chest/abdomen: Memindahkan data rotasi dada ke sendi perut
- Spine submenu->Smooth chest/abdomen: Mendistribusikan semua kelengkungan tulang belakang atas dada dan perut secara alami
- Spine submenu->Combine chest / abdomen Menggabungkan rotasi dari dada dan perut menjadi satu sendi dan menghapus yang lain
- Zero: Nol kombinasi yang berbeda dari sendi
- Square shoulders: Angkat bahu. Pintasan keyboard '['
- Bahu bundar: Turunkan bahu. Pintasan keyboard ']'
- Round shoulders: Gerakkan kepala ke depan. Pintasan keyboard 'Ctrl+['
- Straighten Neck: Gerakkan kepala ke belakang. Pintasan keyboard 'Ctrl+]'

- Reverse Timeline: Atur animasi untuk diputar mundur

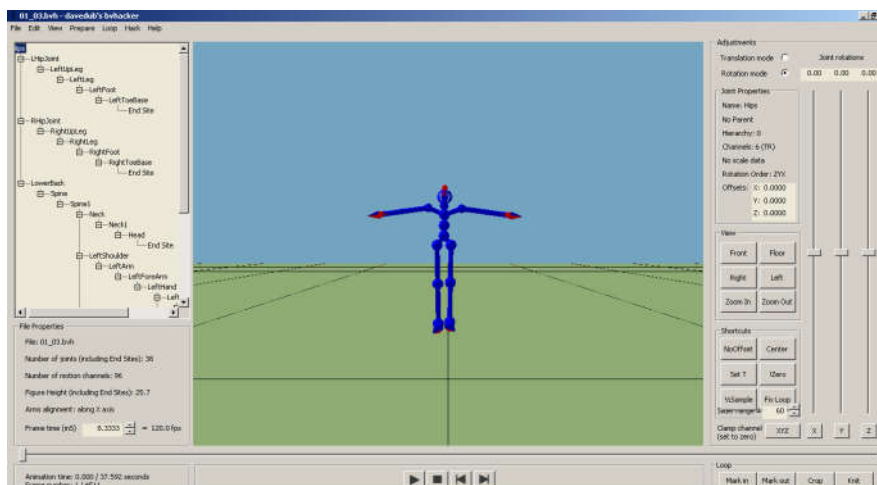
help



- online help: Membuka halaman bantuan online bvhacker di jendela browser
- discussion Grup: Laporkan bug, kirim permintaan fitur, beri komentar, ajukan pertanyaan...
- Donate: Jika Anda menggunakan bvhacker secara teratur, mohon donasi hanya \$10
- About: Menampilkan dialog 'tentang' yang memberikan informasi versi dan hak cipta tentang bvhacker

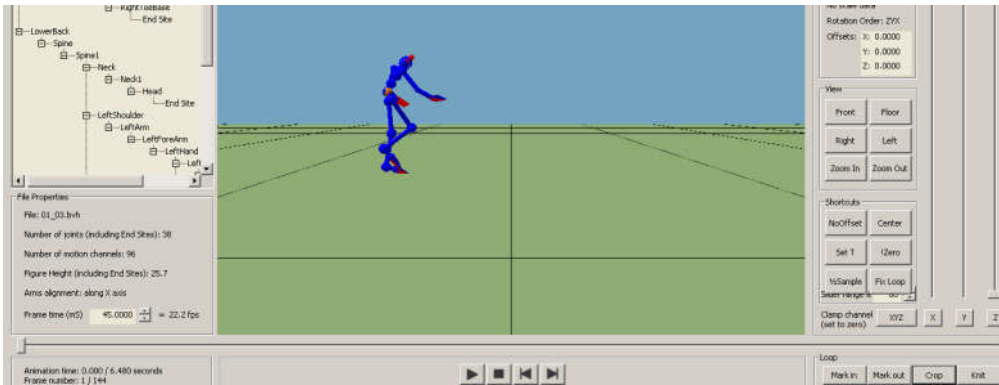
Untuk mengedit gerakan dari bvh yang mungkin terlalu panjang, umpamanya gerakan memanjat tangga, yang anda perlukan hanya memegang tangga saja maka anda perlu bisa melakukan pengeditan pemotongan dan kalau perlu memanipulasi gerakan dari motion capture tersebut.

contoh penerapan bvhacker untuk melakukan pemotongan bvh



Perhatikan pada bagian bawah terdapat tombol mark in mark out dan crop. Geser Slide ke arah awal gerakan yang kita butuhkan, lalu tekan mark in. Geser slide ke arah akhir gerakan yang kita butuhkan lalu tekan mark out lalu tekan crop

Maka animasi akan terpotong sesuai dengan yang kita inginkan, sesuaikan juga fps dari animasi ke 25 fps atau sesuai dengan keinginan kita dan kehalusan hasil, jika ingin hasil yang halus pengaturan fps ini bisa juga kita lewat

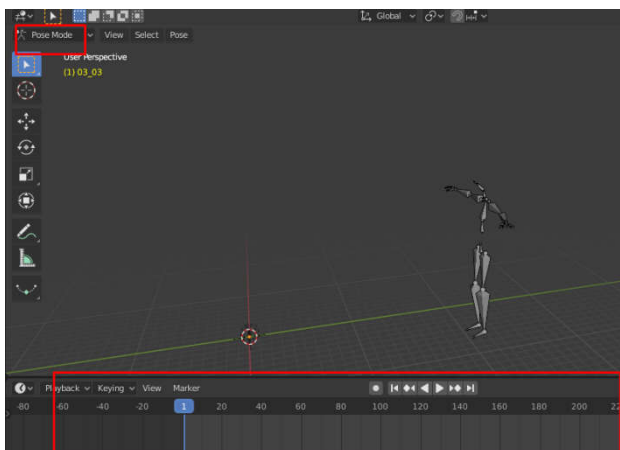


Setelah itu tekan save untuk mendapatkan file yang kita butuhkan. File tersebut akan berekstensi bvh dan akan siap dimanfaatkan dalam blender.

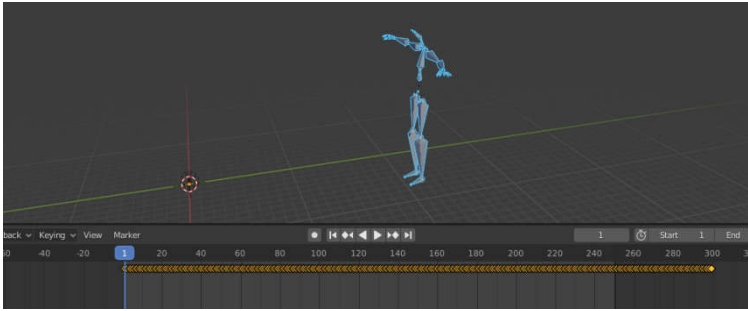
6.2 mengedit bvh dengan blender

kita juga bisa mengedit bvh dengan blender terutama dengan blender 2.8 keatas karena pengaturan keyframe pada animasi lebih mudah dibanding versi terdahulu.

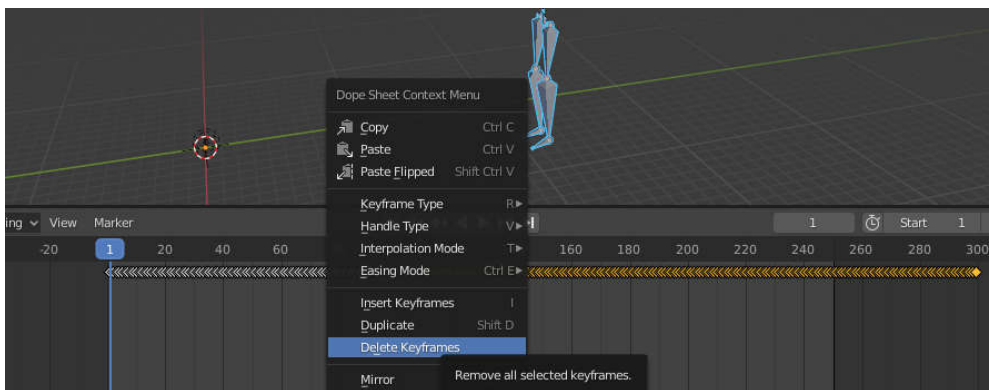
caranya kita import bvh ke blender, pindah ke pose mode



lalu kita select semua tulang kita bisa lihat keyframe dari keseluruhan tulang

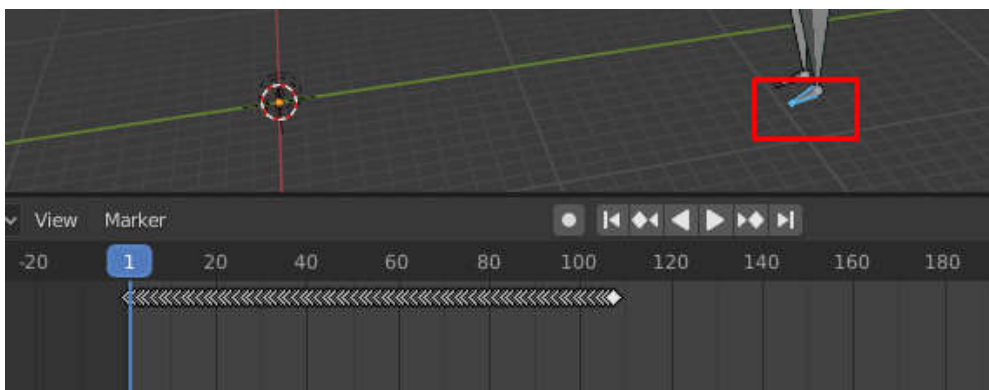


jadi keyframe keyframe ini bisa kita hapus seandainya hanya menginginkan animasi pendek atau sebagian saja cukup klik kanan delete keyframe

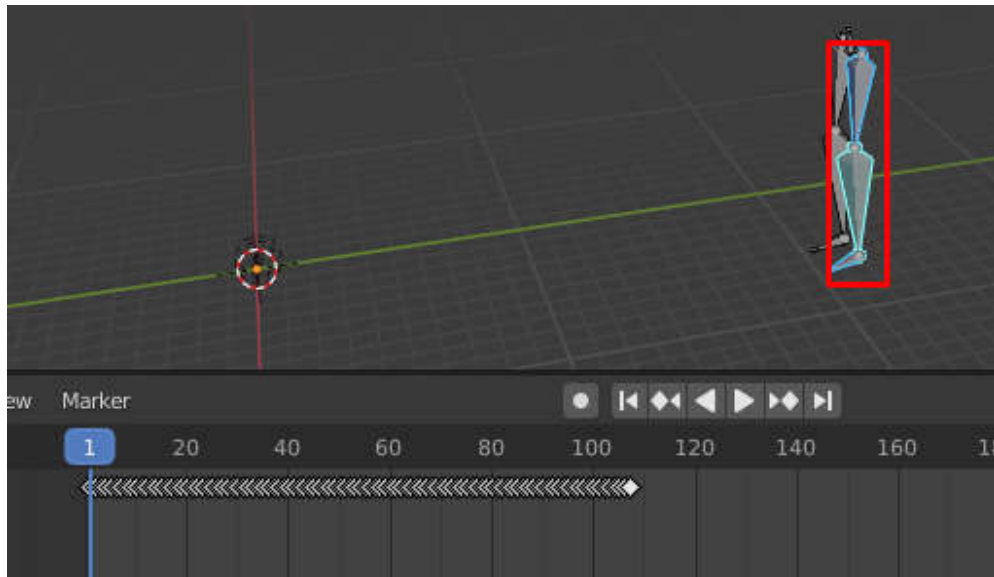


ingin mengurangi jumlah keyframe juga bisa tinggal dimanipulasi keyframanya hapus yang dirasa tidak perlu.

pada blender kita juga bisa memanipulasi keyframe dari bvh ini per tulang



atau dalam sekelompok tulang



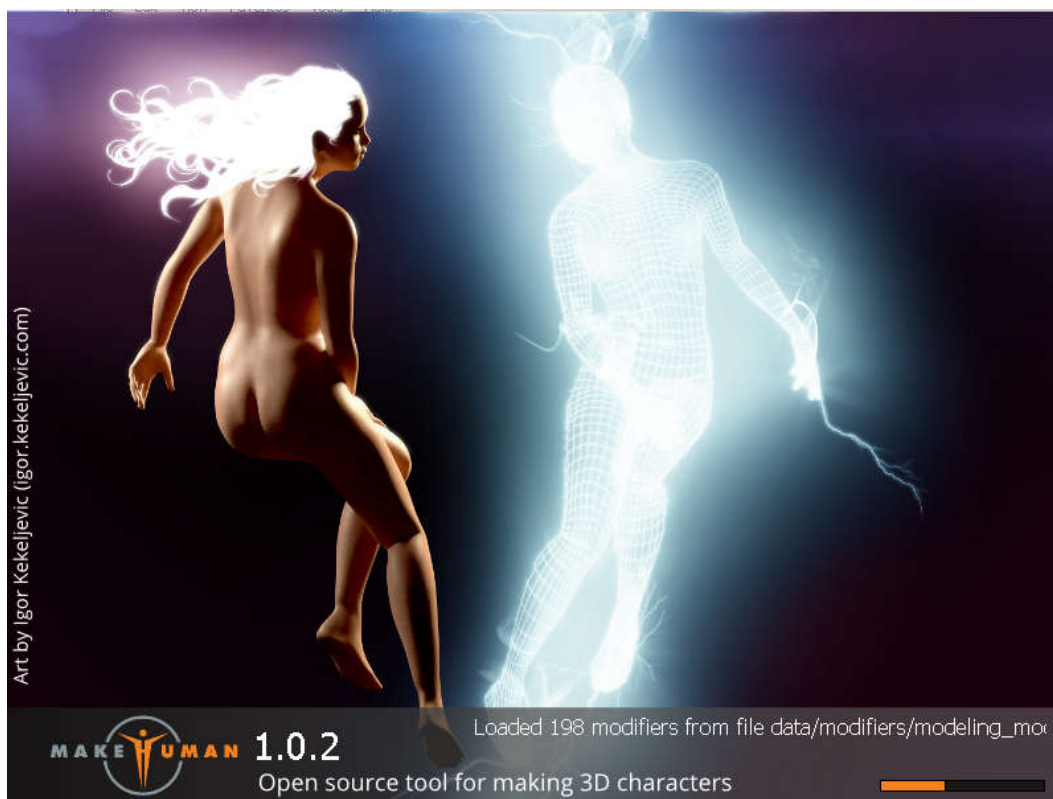
intinya setiap obyek yang kita pilih maka keyframe yang ada dibawah adalah keyframe dari obyek yang kita pilih , tidak pada obyek yang lain dan siap untuk dimanipulasi.

BAB 7 APLIKASI BANTU ANIMASI KARAKTER DENGAN MOTION CAPTURE

7.1 makehuman

Membuat karakter 3 dimensi dengan maya sama seperti proses modeling , hanya model nya berbentuk manusia, atau binatang. Bisa menggunakan metode low polygon modeling yang sudah dipelajari pada maya modeling.

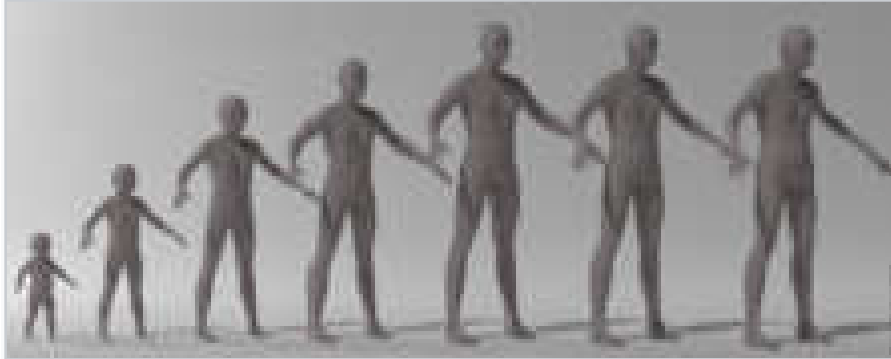
Salah satu kendala dalam modeling manusia adalah dibutuhkan bentuk yang berkarakter, yang dalam hal ini membutuhkan waktu yang lama untuk membuat obyek manusia yang layak. Untuk mempercepat pembuatan manusia, kita bisa menggunakan program bantu, salah satunya adalah program makehuman yang merupakan program free yang ditujukan untuk membuat obyek manusia dengan sangat cepat dan mudah,



makehuman 32 bit

MakeHuman adalah a Grafik komputer 3D middleware yang gratis dan open source dirancang untuk prototipe pembuatan fotorealistik humanoid. Ini dikembangkan oleh komunitas pemrogram, seniman, dan akademisi yang tertarik dengan pemodelan karakter 3D.

MakeHuman dikembangkan menggunakan teknologi morphing 3D. Mulai dari androgini , dapat diubah menjadi berbagai macam karakter (pria dan wanita), mencampurnya dengan interpolasi linier. Misalnya, dengan empat *target* (bayi, remaja, muda, tua), dimungkinkan untuk mendapatkan semua bentuk perantara.



Interpolasi karakter MakeHuman: yang ke-1, ke-3, ke-5, dan ke-7 adalah *target*, sedangkan yang lain adalah bentuk perantara.

Menggunakan teknologi ini, dengan basis data target morphing yang besar, hampir memungkinkan untuk mereproduksi karakter apa pun yang diinginkan. GUI sangat sederhana untuk mengakses dan dengan mudah menangani ratusan morphing. Pendekatan MakeHuman adalah menggunakan penggeser dengan parameter umum seperti tinggi, berat, jenis kelamin, etnis, dan otot. Untuk membuatnya tersedia di semua sistem operasi, mulai dari 1.0 alpha 8 dikembangkan dengan Python menggunakan OpenGL dan Qt, dengan arsitektur yang sepenuhnya direalisasikan dengan plugin.

Alat ini dirancang khusus untuk pemodelan model manusia 3D virtual, dengan sistem pose sederhana dan lengkap yang mencakup simulasi gerakan otot. Antarmukanya mudah digunakan, dengan akses cepat dan intuitif ke berbagai parameter yang diperlukan dalam memodelkan bentuk manusia.

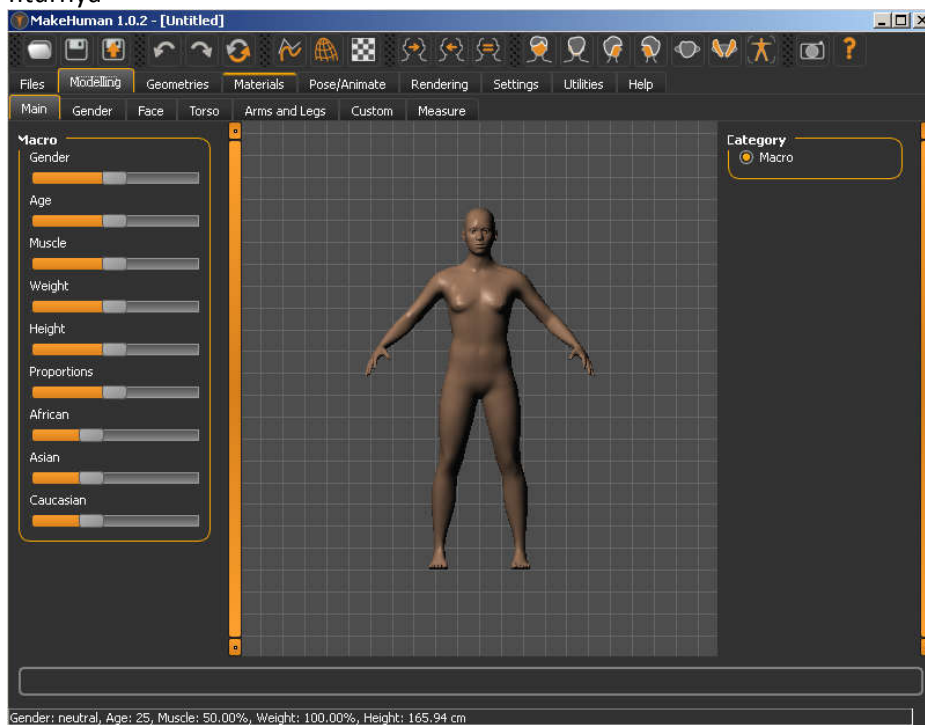
Perkembangan MakeHuman berasal dari studi teknis dan artistik yang mendetail tentang morfologis tubuh manusia. Pekerjaan berkaitan dengan *morphing*, menggunakan interpolasi linier baik terjemahan dan rotasi. Dengan kedua metode ini, bersama dengan perhitungan sederhana faktor bentuk dan algoritme relaksasi mesh, hasil seperti simulasi gerakan otot yang menyertai rotasi anggota badan dapat dicapai.

MakeHuman gratis dan open source, dengan kode sumber dan basis data dirilis di bawah GNU Affero GPL. Model yang diekspor dari versi resmi dirilis dengan pengecualian ini, CC0, agar dapat digunakan secara luas dalam gratis dan tidak bebas. Proyek-proyek ini mungkin atau mungkin tidak dikomersialkan.



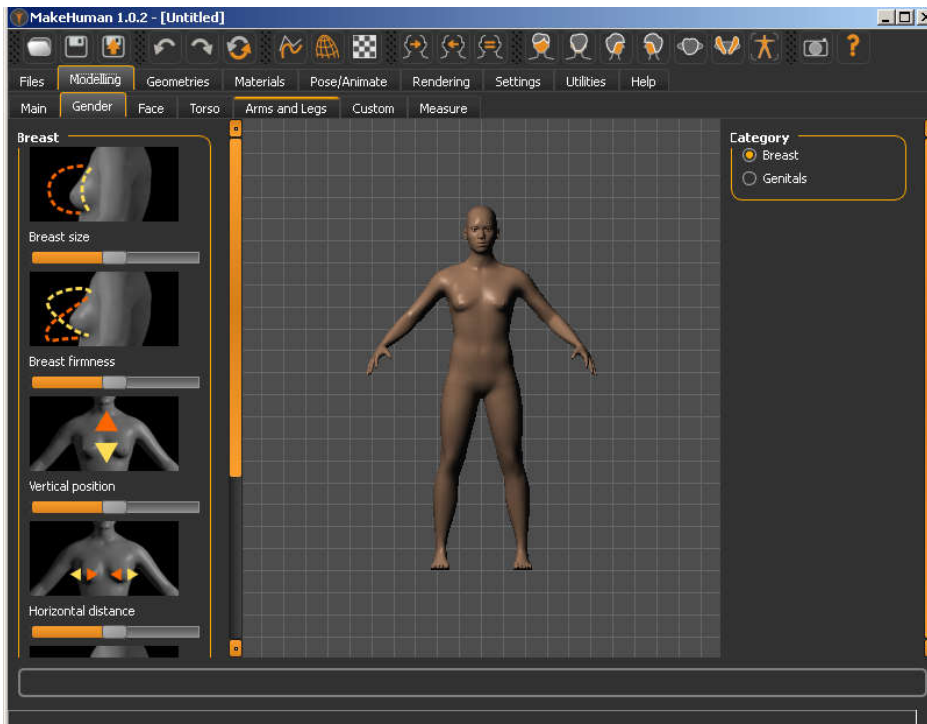
makehuman untuk versi 64bit

Program ini memiliki pengaturan bentuk utama seperti gender, usia berat dan tinggi proporsi tubuh, kemiripan ke jenis orang afrika asia atau eropa. Semakin baru versinya semakin banyak fitur nya

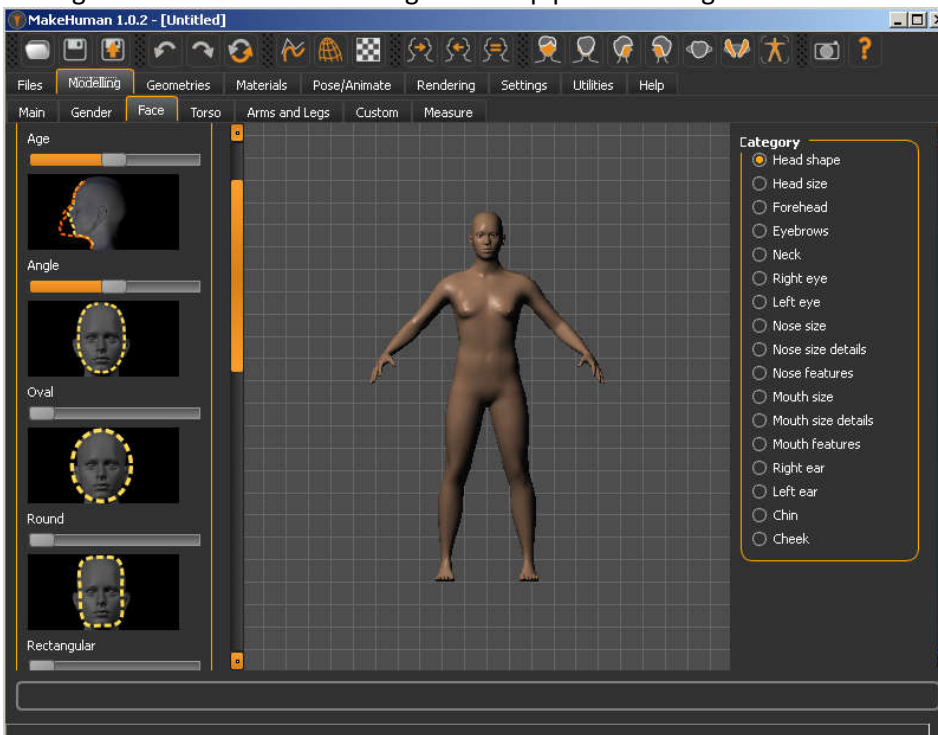


memiliki pengaturan gender seperti bentuk dada yang berbeda antara laki laki dan perempuan

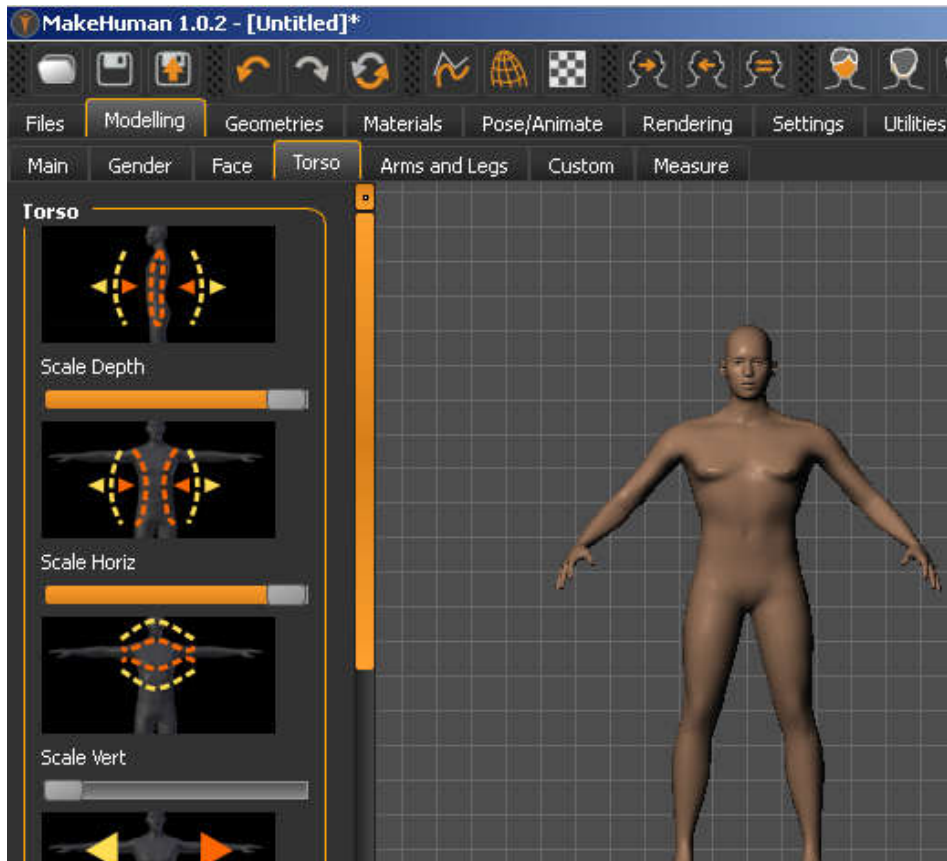
Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)



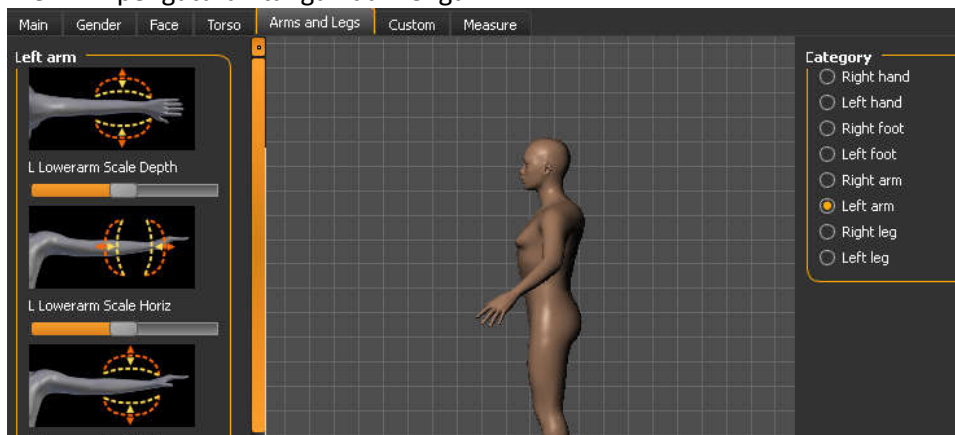
memiliki pengaturan wajah seperti bentuk kepala , besar kepala, bentuk dahi alis panjang leher mata hidung bentuk mulut bentuk telinga bentuk pipi dan rahang



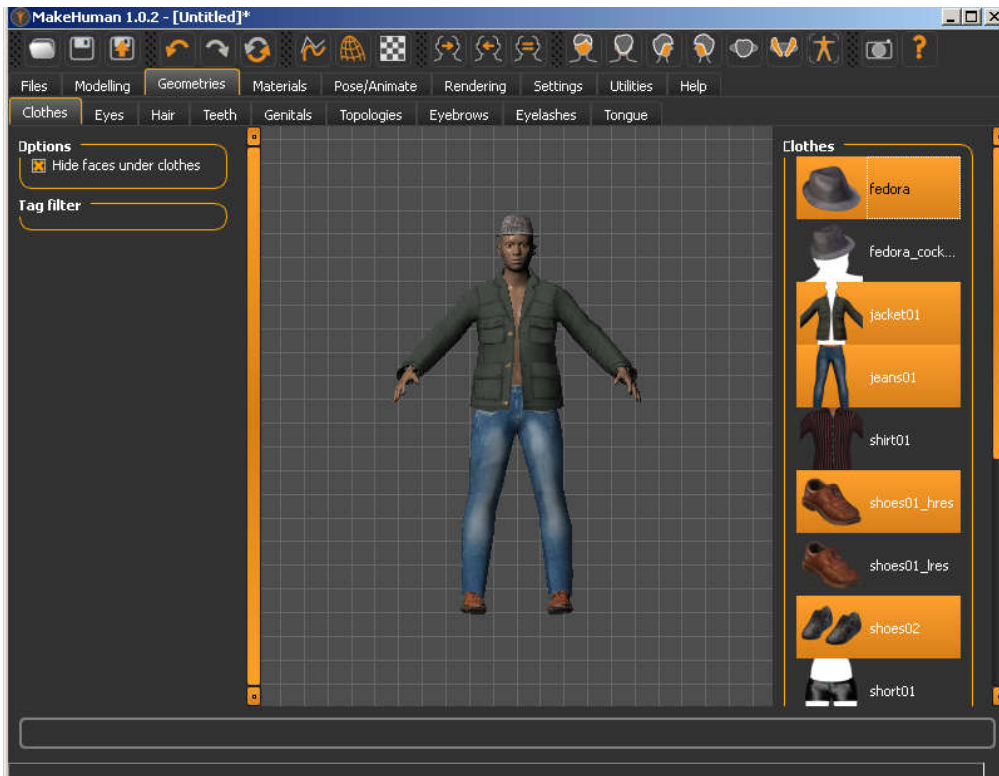
memiliki pengaturan badan



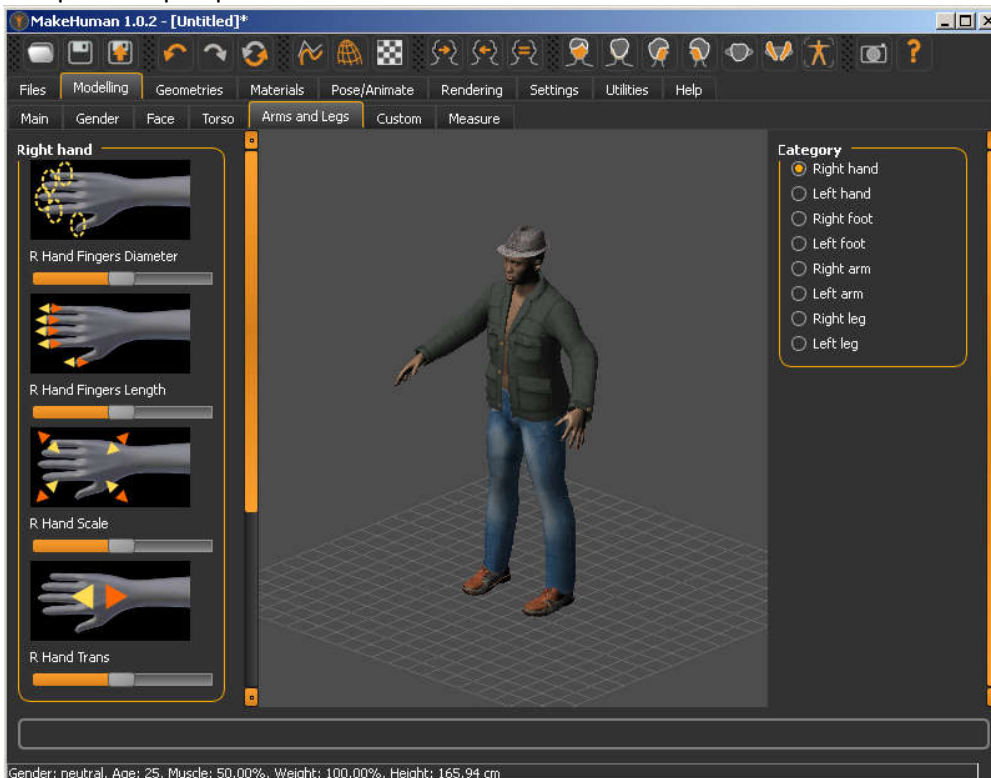
memiliki pengaturan tangan dan lengan



juga dapat ditambahkan obyek geometri untuk model seperti baju sepatu jaket



juga dapat ditambahkan material memiliki pengaturan tampilan baik kiri kanan depan belakang maupun dari perspektif



Secara total memiliki 1170 morphing untuk modeling karakter manusia secara efektif .

Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

Program ini sangat mudah digunakan sangat mudah dipahami, dan tidak membutuhkan waktu belajar yang lama karena memang hanya ditujukan untuk membuat model manusia saja. Dalam waktu kurang dari satu jam anda bisa membuat obyek manusia 3 dimensi yang anda inginkan.

Blender exchange format adalah yang membedakan antara versi baru dengan yang terdahulu. Pada versi 1.02 memiliki format mhx sementara pada versi terbaru saat buku ini ditulis, versi 1.20 memiliki format mhx2 dan tidak menyediakan format mhx. Ini yang membuat kita harus memilih , jika masih menggunakan blender 2.7 kebawah akan menggunakan versi 1.02 akan lebih mudah , sementara jika kita menggunakan blender 2.8 keatas tidak suport mhx maka harus menggunakan versi 1.20 yang berformat mhx2. Jika anda masih menggunakan windows 32bit , blender 2.8 dan makehuman 1.20 tidak akan jalan. Oleh karenanya silahkan disesuaikan dengan kondisi komputer anda.

7.2 Mixamo

Mixamo didirikan pada tahun 2008 oleh Stefano Corazza dan Nazim Kareemi sebagai spin-off Lab Biomotion Universitas Stanford,

Mixamo merilis layanan rigging otomatisnya pada tahun 2011. Itu diikuti dengan peluncuran produk animasi wajah real-time, Face Plus, di 2013 dan peluncuran resmi perangkat lunak pembuat karakter Fuse 3D pada Maret 2014. Pada Agustus 2014, Mixamo meluncurkan struktur harga baru. Mixamo diakuisisi oleh Adobe Systems pada 1 Juni 2015.

Fuse



Models dibuat dengan Fuse.

Pada Maret 2014, Mixamo secara resmi meluncurkan Fuse Character Creator di Game Developers Conference. Fuse bermula dari penelitian yang dilakukan oleh Siddhartha Chaudhuri yang berasal dari kelompok penelitian Vladlen Koltun di Stanford dan teknologi Substansi dari Allegorithmic. Versi awal dari layanan ini telah dirilis pada November 2013

Ini memungkinkan pengguna untuk membuat karakter 3D dengan menggabungkan bagian tubuh, pakaian, dan tekstur yang dapat disesuaikan bersama-sama Karakter tersebut kemudian dapat diekspor ke dalam model 3D atau mesin game lainnya.

Pada bulan Maret 2014, Mixamo meluncurkan Fuse 1.0 di Steam dengan kemampuan mengimpor dan mengintegrasikan konten buatan pengguna ke dalam sistem pembuatan karakter. Fuse telah diperbarui untuk memungkinkan impor dan pengeditan model karakter yang dihasilkan oleh Microsoft Kinect 2.0 pada bulan Agustus tahun itu.

Pengembangan Fuse dihentikan pada September 2019. Pada 13 September 2020, program tersebut dihapus dari Adobe Creative Cloud dan tidak lagi tersedia untuk diunduh dari Adobe. Versi 1.3 masih tersedia untuk diunduh di Steam Marketplace.

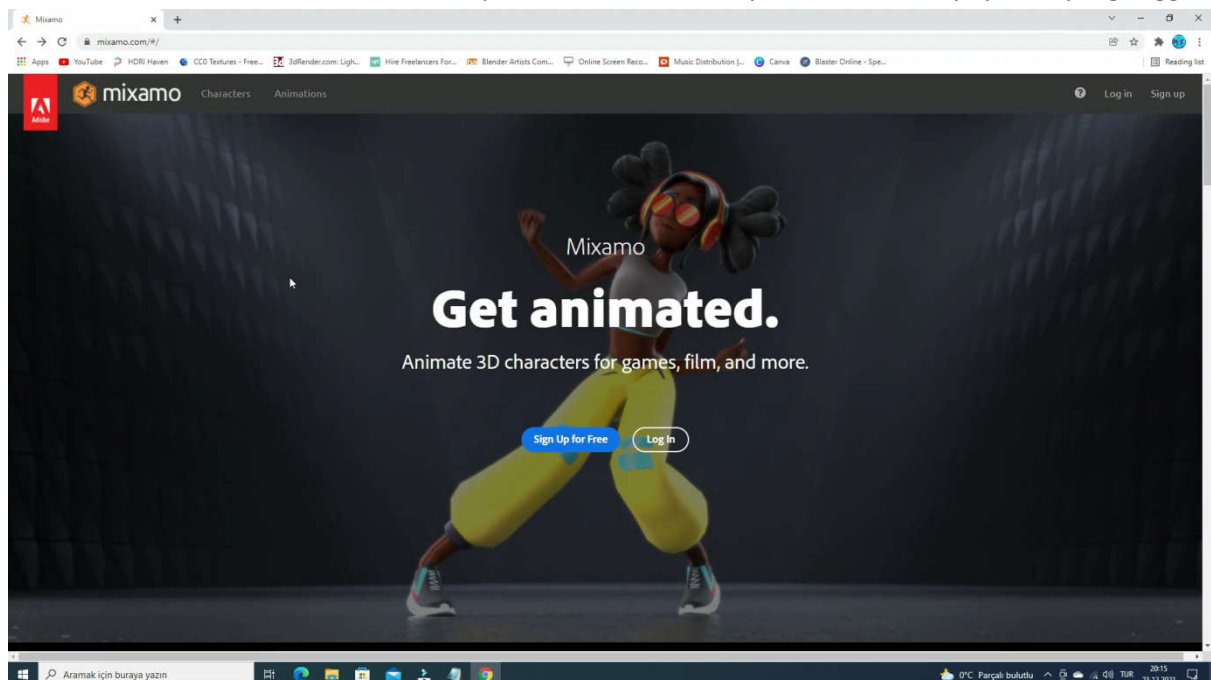
Layanan ini telah dihentikan untuk menyimpan model dan model 3D khusus yang diunggah melalui CC sekering di server Adobe. Anda masih dapat mengunggah model 3D khusus ke layanan rigging otomatis online dan memasangnya, tetapi model tersebut tidak akan disimpan. Mixamo juga menyediakan layanan rigging model otomatis online yang dikenal sebagai Auto-Rigger, yang merupakan layanan rigging online pertama di industri. AutoRigger menerapkan pembelajaran mesin untuk memahami di mana anggota badan model 3D berada dan memasukkan "kerangka", atau rig, ke dalam model 3D serta menghitung bobot skinning. Layanan dapat memakan waktu hingga 2 menit.

Didalam Mixamo ada toko animasi yang menampilkan model 3D yang dapat diunduh dan rangkaian animasi. Animasi dibuat di Mixamo menggunakan Motion capture dan dibersihkan oleh animator keyframe . Semua animasinya bekerja dengan karakter yang dibuat di Fuse dan/atau dengan AutoRigger Mixamo.

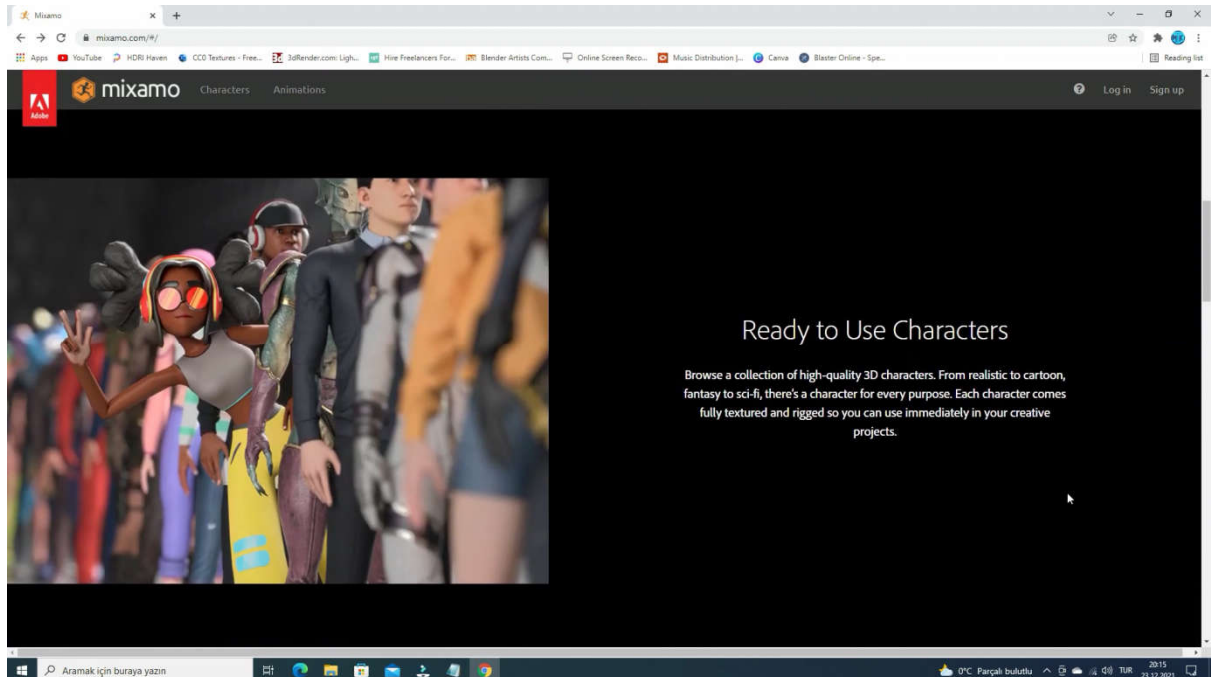
7.2.1 penerapan mixamo pada blender

mengimpor karakter yang rigged dan animasi gratis ke dalam Blender.

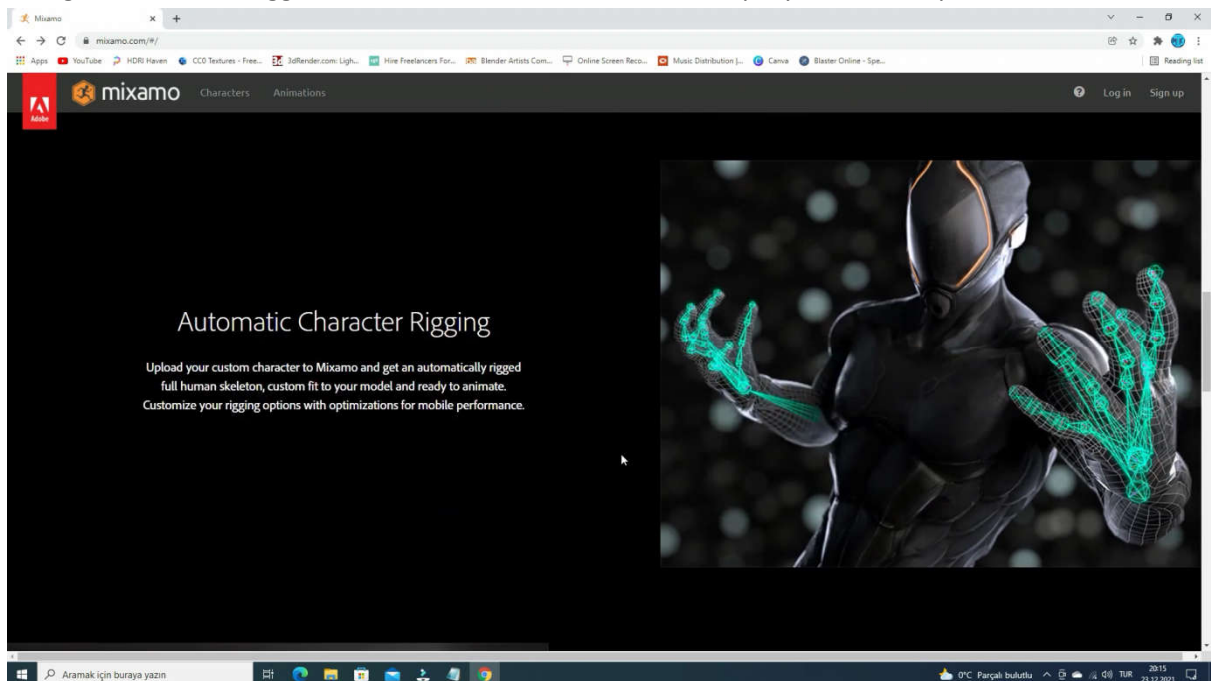
pada bagian ini Kita akan menggunakan addon mixamo. Mixamo adalah layanan berbasis web gratis untuk animasi karakter 3D. Anda dapat menemukan banyak karakter siap pakai yang rigged.



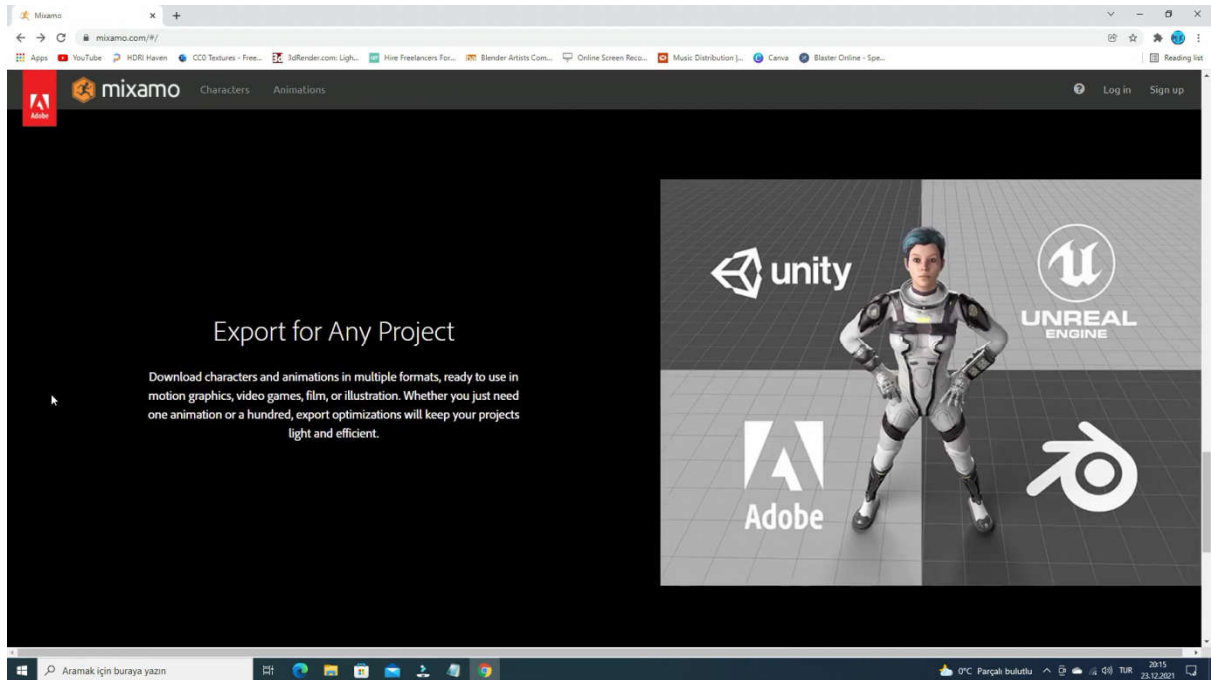
Atau, Anda dapat mengunggah karakter kustom Anda sendiri ke mixamo. Jadi, Anda bisa mendapatkan kerangka manusia utuh yang rigged secara otomatis.



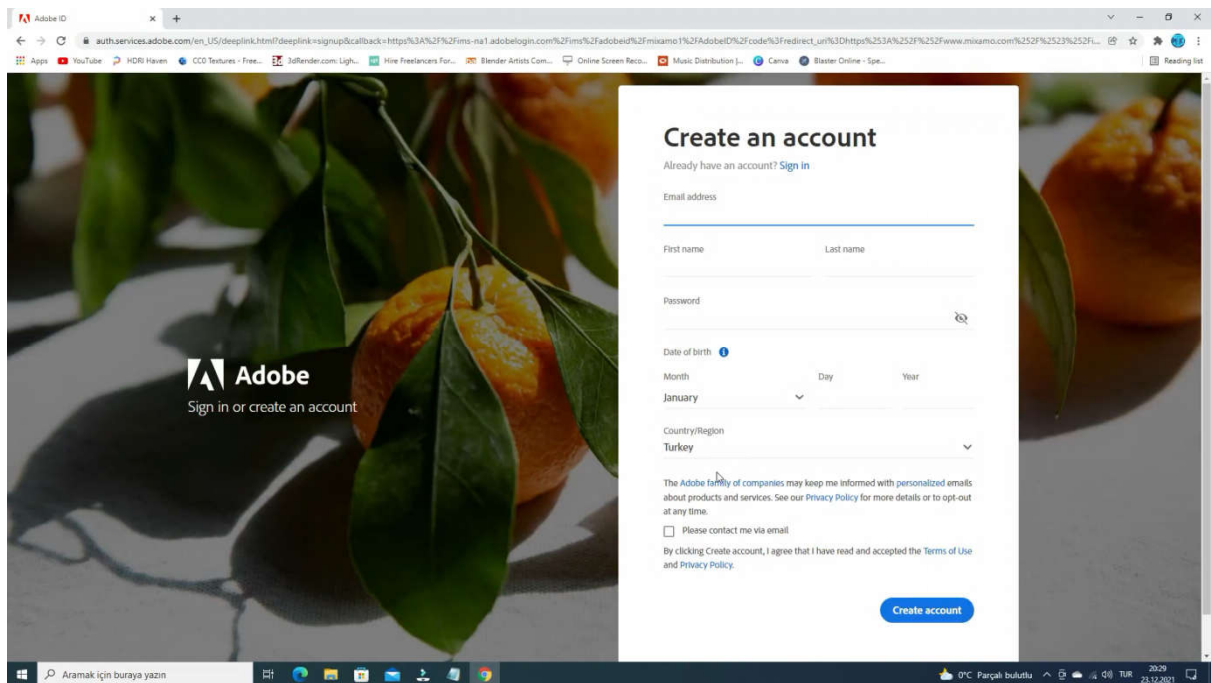
Anda juga dapat menambahkan animasi siap pakai ke karakter Anda secara instan. Anda dapat mengunduh dan menggunakan karakter animasi Anda untuk proyek film dan permainan video Anda.



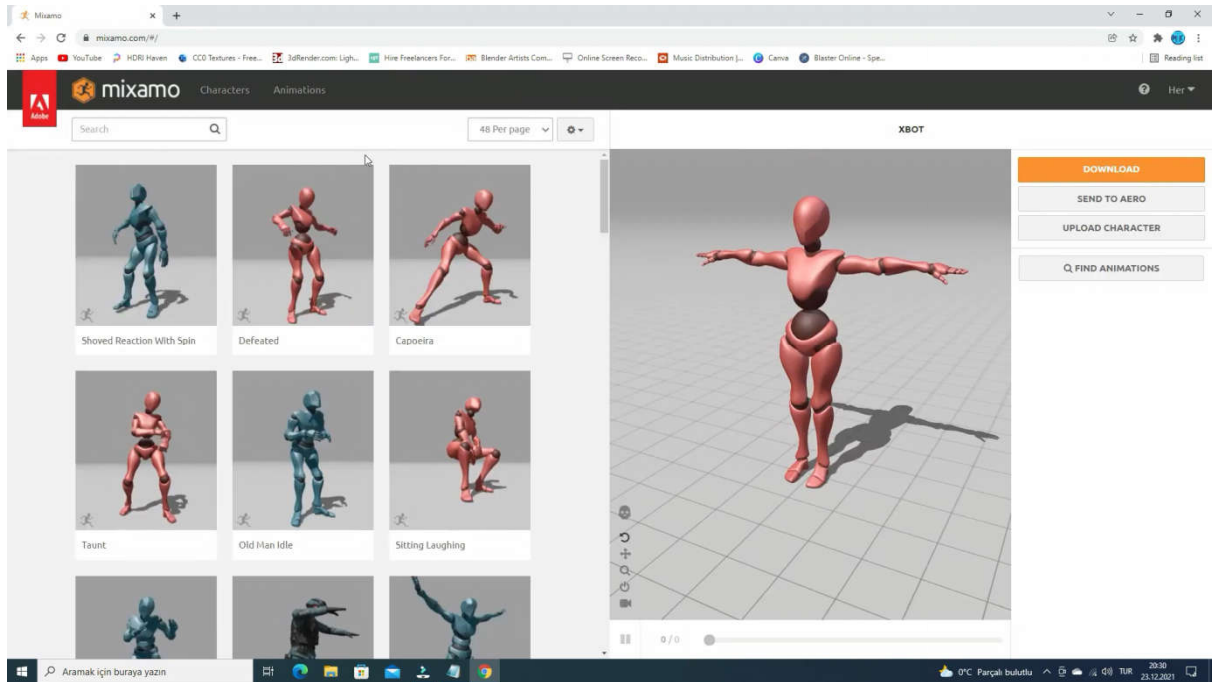
Selain itu, Anda dapat menggunakan karakter dan animasi bebas royalti untuk proyek pribadi, dan nirlaba. Mari kita lihat cara menggunakan mixamo.



Pertama, kita memerlukan akun adobe.

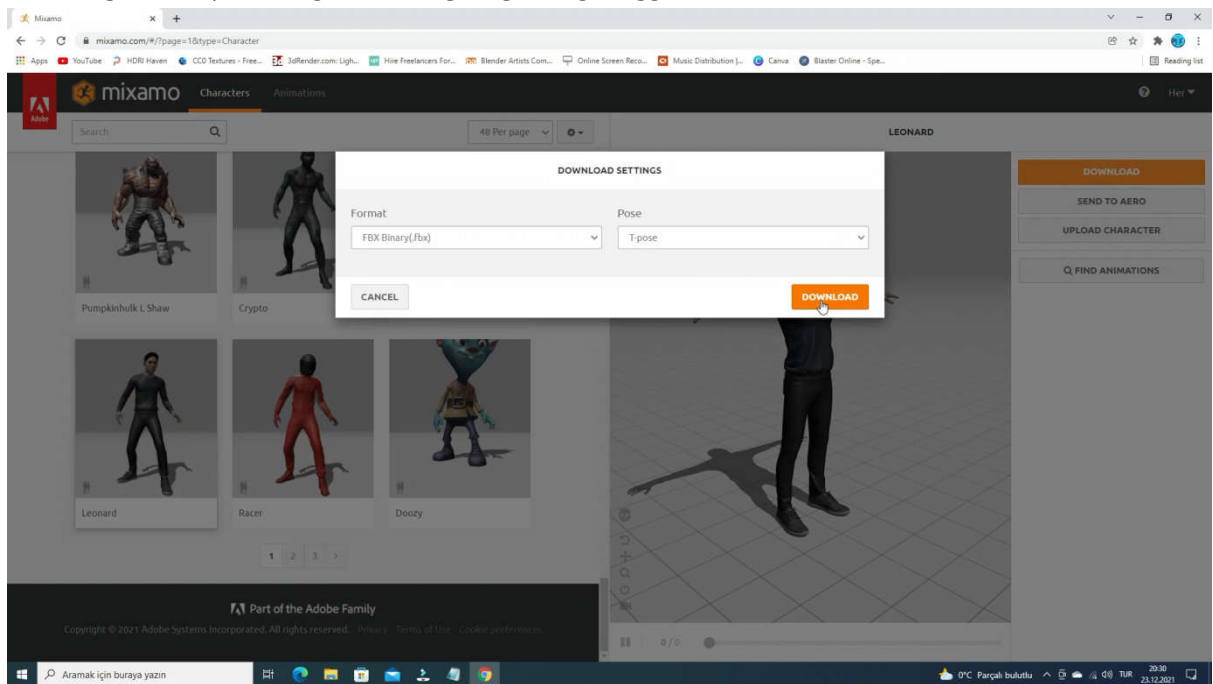


Kita dapat membuat atau masuk dengan akun google Anda. Setelah masuk, klik tab karakter.



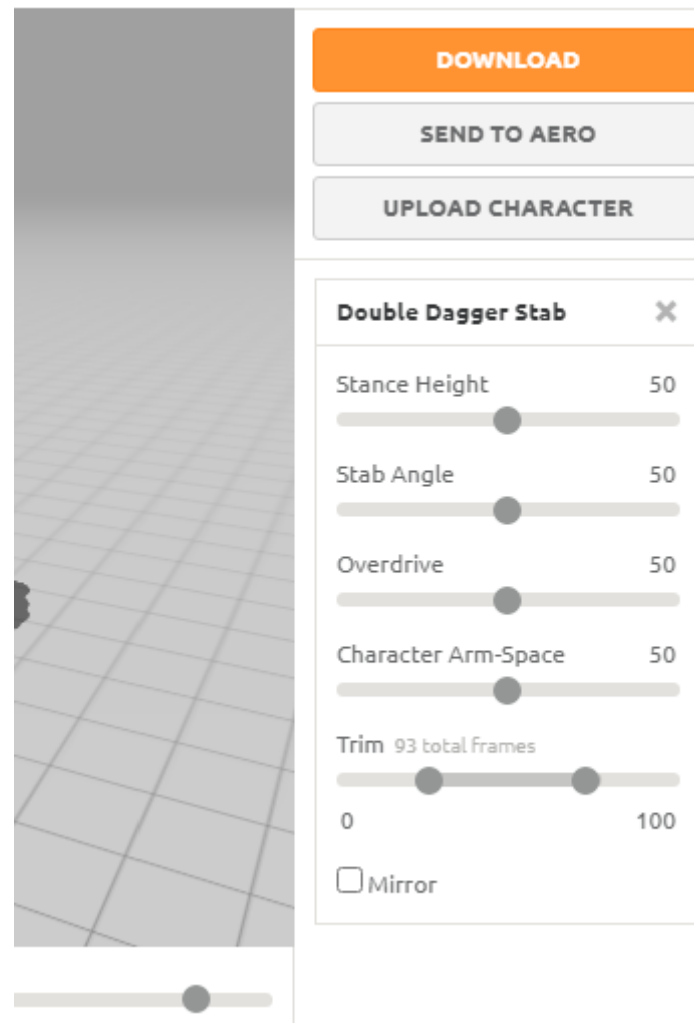
pilih karakter.

Sekarang, kita dapat mengunduh langsung sebagai rigged



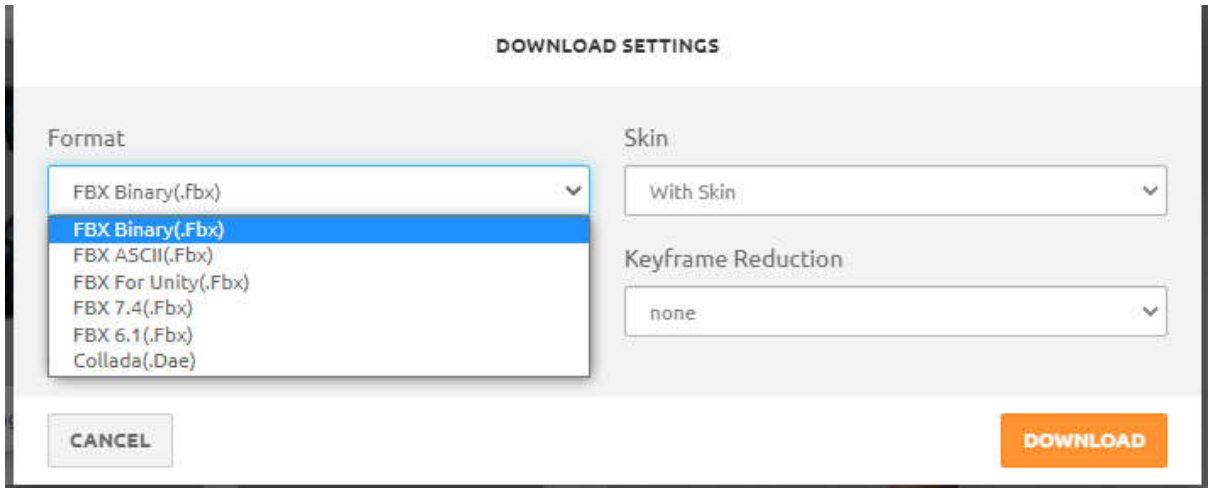
Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

atau dapat menambahkan animasi. Mari klik dan pilih animasi. Kita juga dapat mempercepat

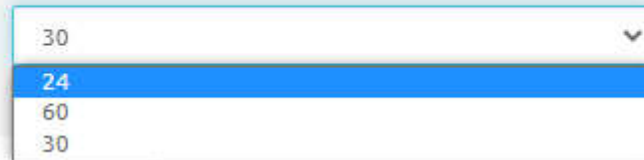


animasi atau trim.

Setelah pengaturan animasi, mari klik tombol unduh. Kita dapat memiliki format unduhan yang berbeda.

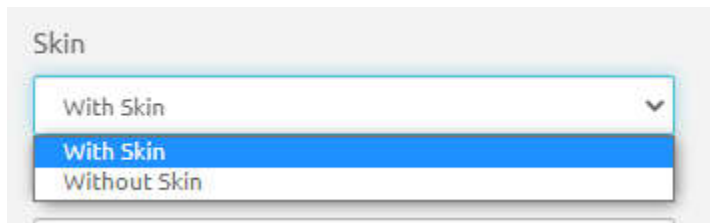


Frames per Second

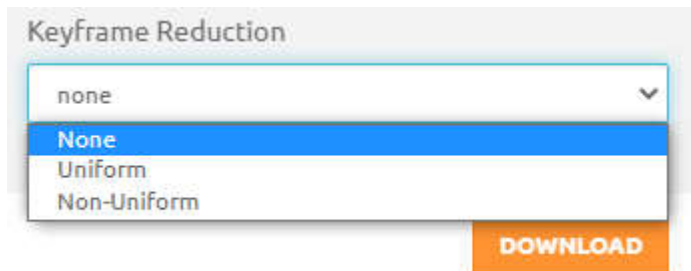


Kita dapat mengatur FPS animasi.

Jika kita memilih opsi tanpa kulit, itu hanya akan mengunduh animasi tanpa model.

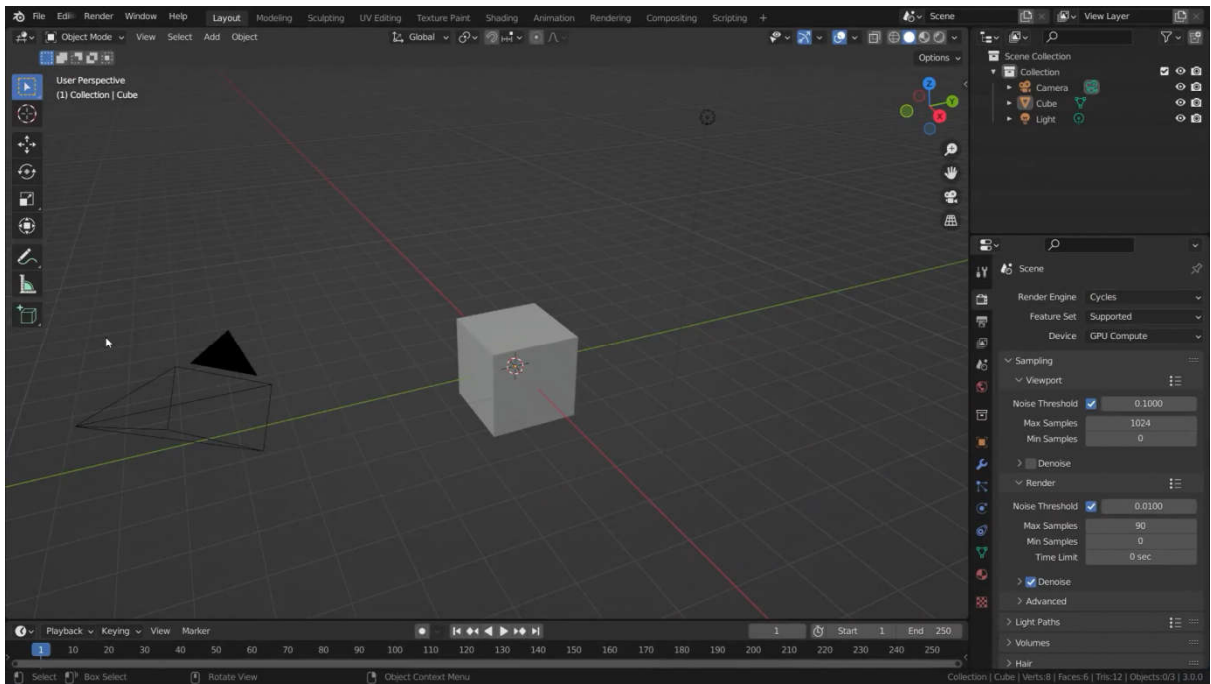


kalau perlu bisa dilakukan pengurangan keyframe

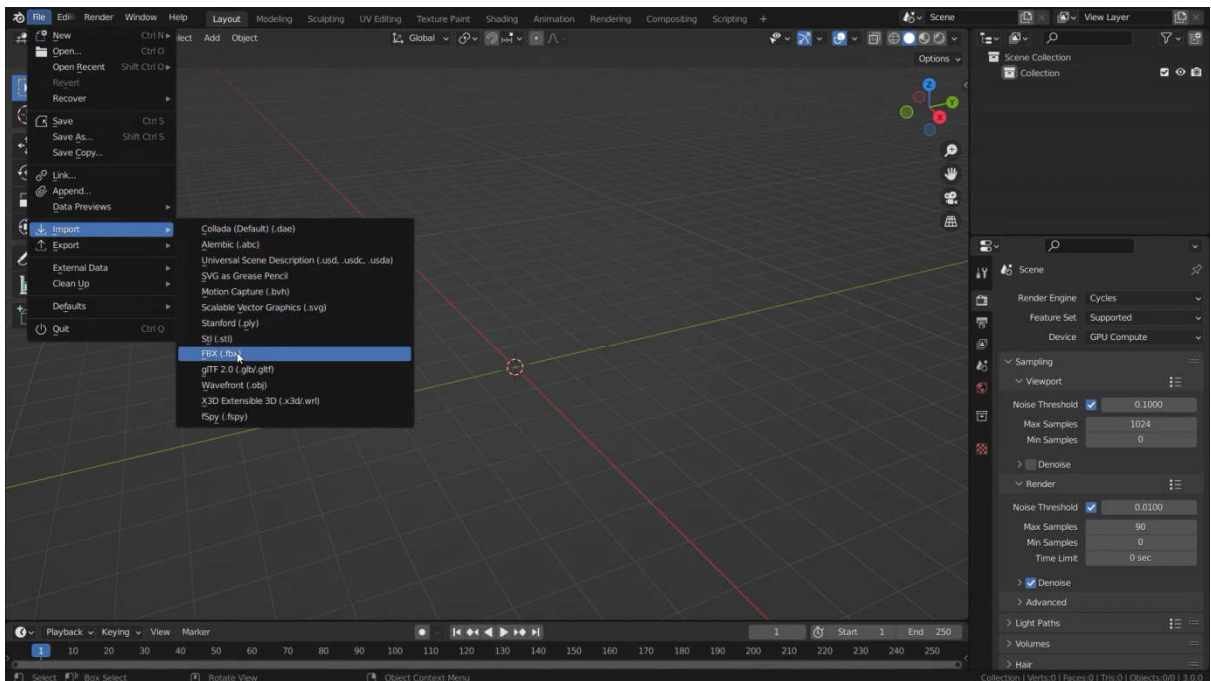


Setelah pengaturan pengunduhan, klik tombol unduh.

Sekarang, mari impor karakter kita ke Blender.



Pertama, tekan tombol A untuk memilih semua objek dalam adegan, dan menghapusnya. Ayo ke menu file, impor, dan pilih format fbx.



Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

pilih file fbx kita dan klik tombol impor FBX.



melihat tekstur. Mari tekan tombol play dan lihat bagaimana animasi bekerja.



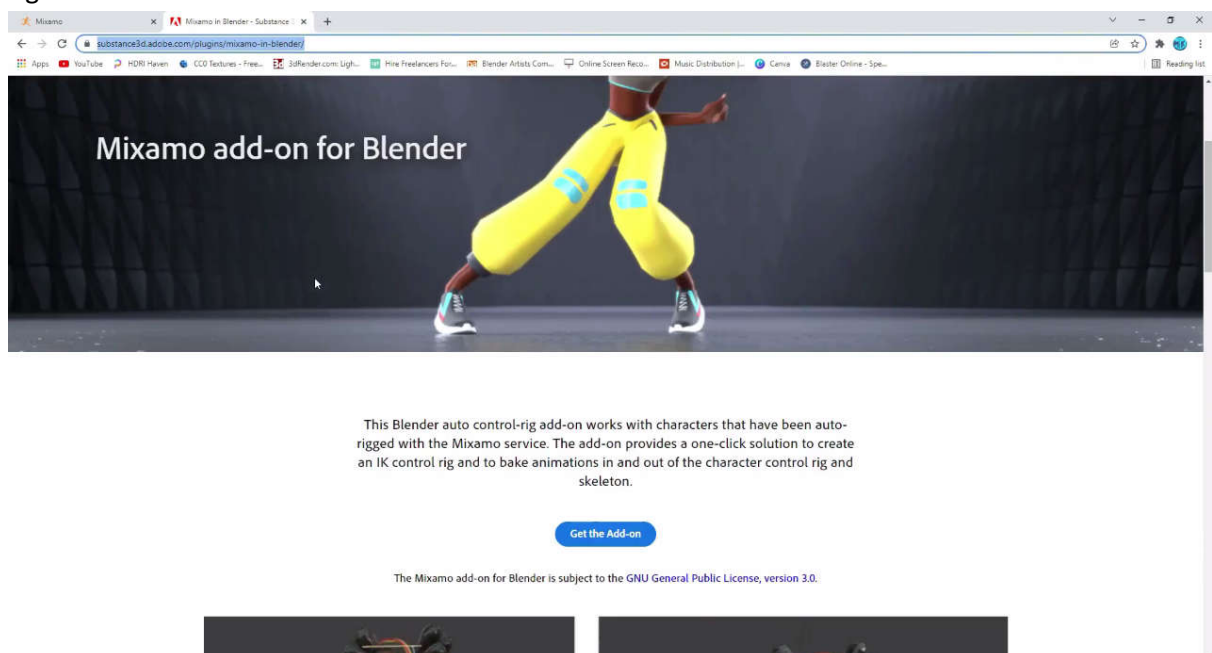
Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

Seperti yang bisa Anda lihat, karakter ini hadir dengan rig dan animasi.



Sekarang, kita ingin membuat rig kontrol untuk karakter ini. Untuk melakukan itu, kita akan mengunduh add-on mixamo untuk blender.

pergi ke halaman unduhan.



klik tombol dapatkan add-on.

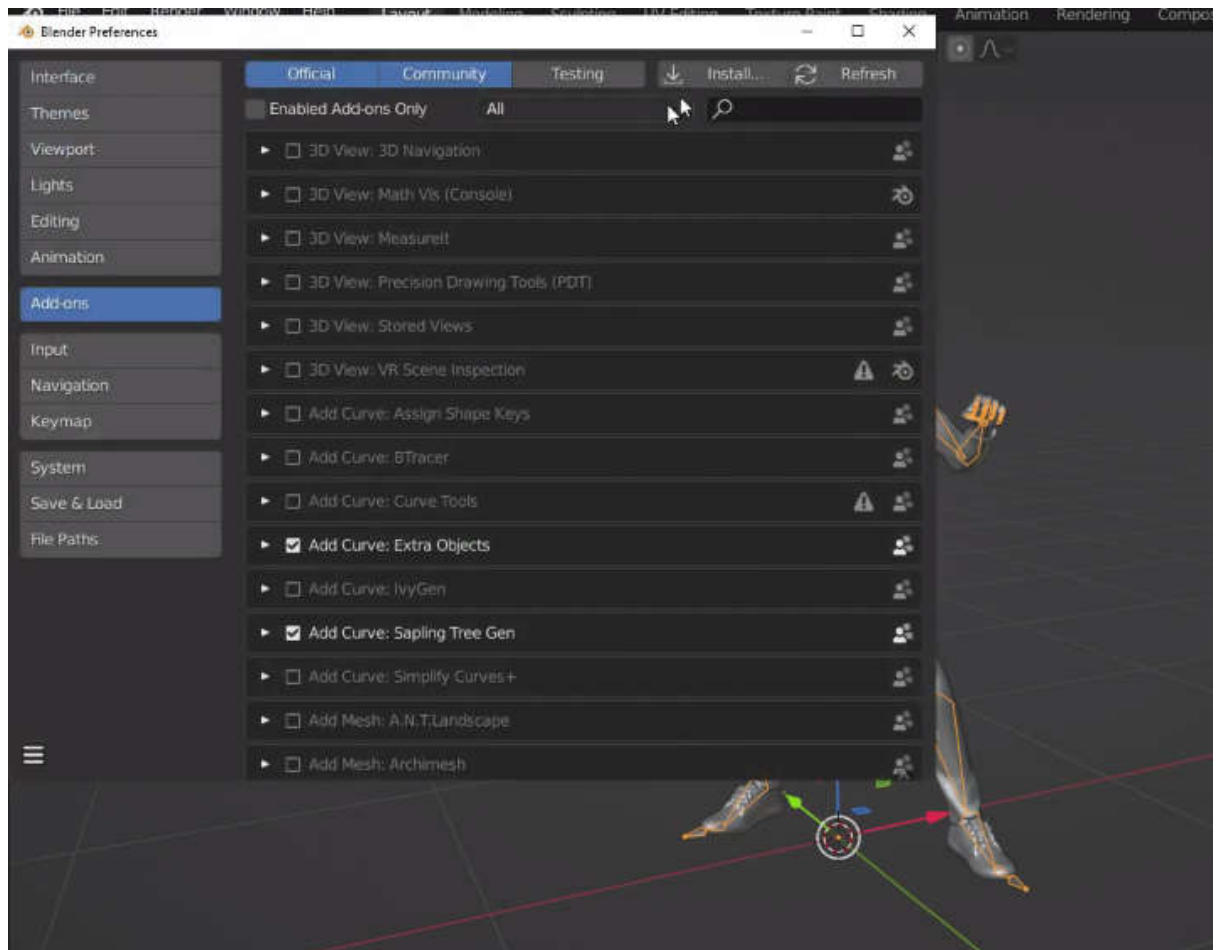
Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

This Blender auto control-rig add-on works with characters that have been auto-rigged with the Mixamo service. The add-on provides a one-click solution to create an IK control rig and to bake animations in and out of the character control rig and skeleton.

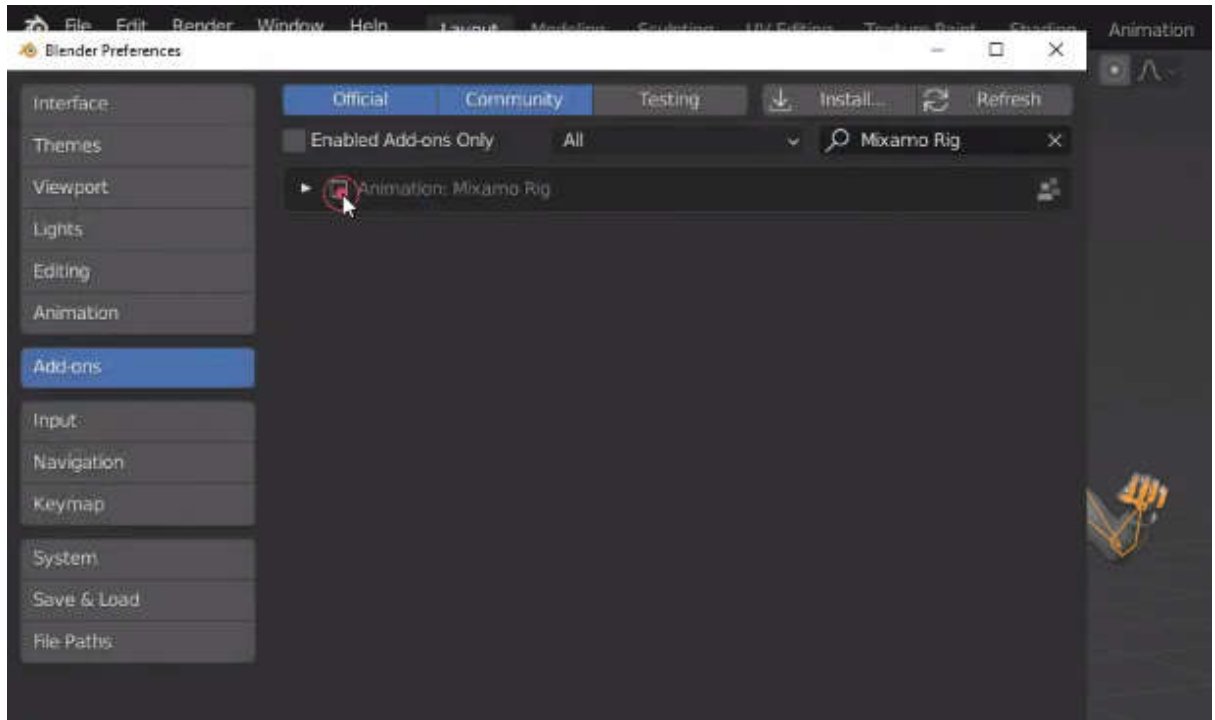
[Get the Add-on](#)

The Mixamo add-on for Blender is subject to the [GNU General Public License, version 3.0](#).

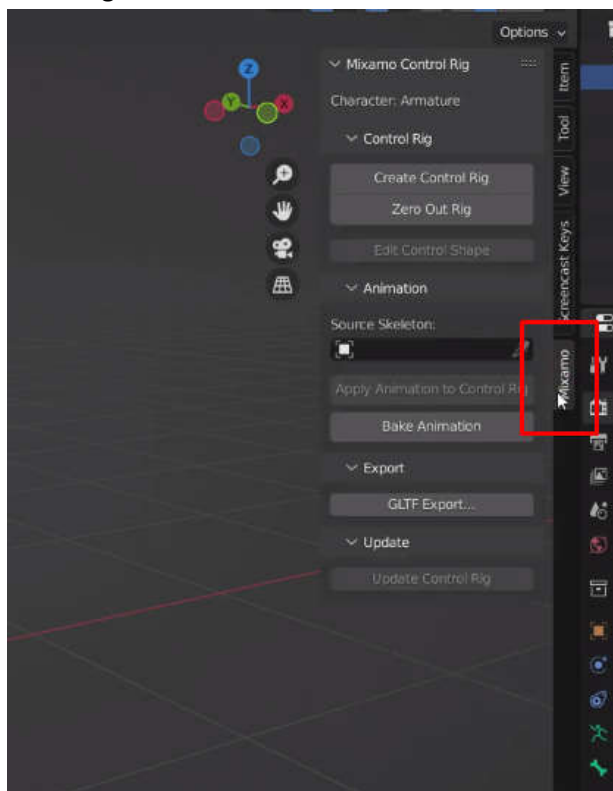
Setelah mengunduh, mari beralih ke Blender lagi. Buka menu edit, preferensi, dan klik pada addons. Ayo klik tombol install dan temukan file yang baru saja kita unduh. Klik tombol instal.



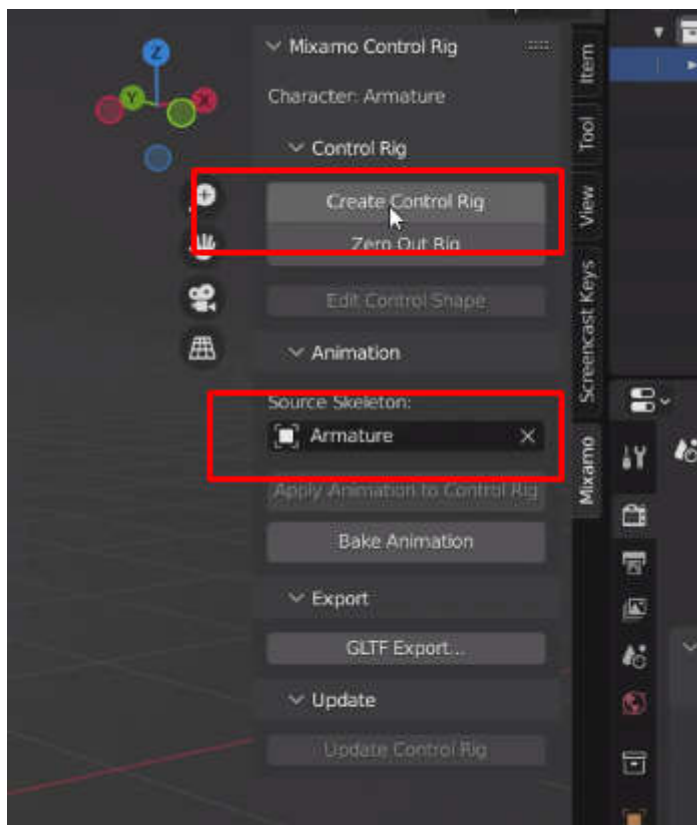
Jangan lupa aktifkan addon mixamo.



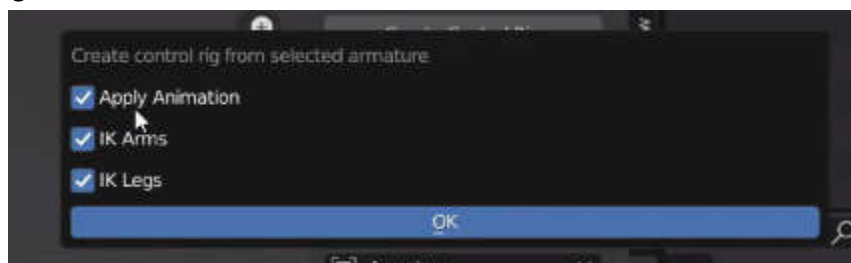
Sekarang, tekan tombol N dan buka menu toolbar samping. Anda dapat melihat add-on mixamo sekarang.



Pertama, mari pilih armature, lalu klik tombol buat rig kontrol.



Mari hapus centang pada opsi apply animation, jadi animasi yang sudah ada tidak akan diterapkan ke rig.

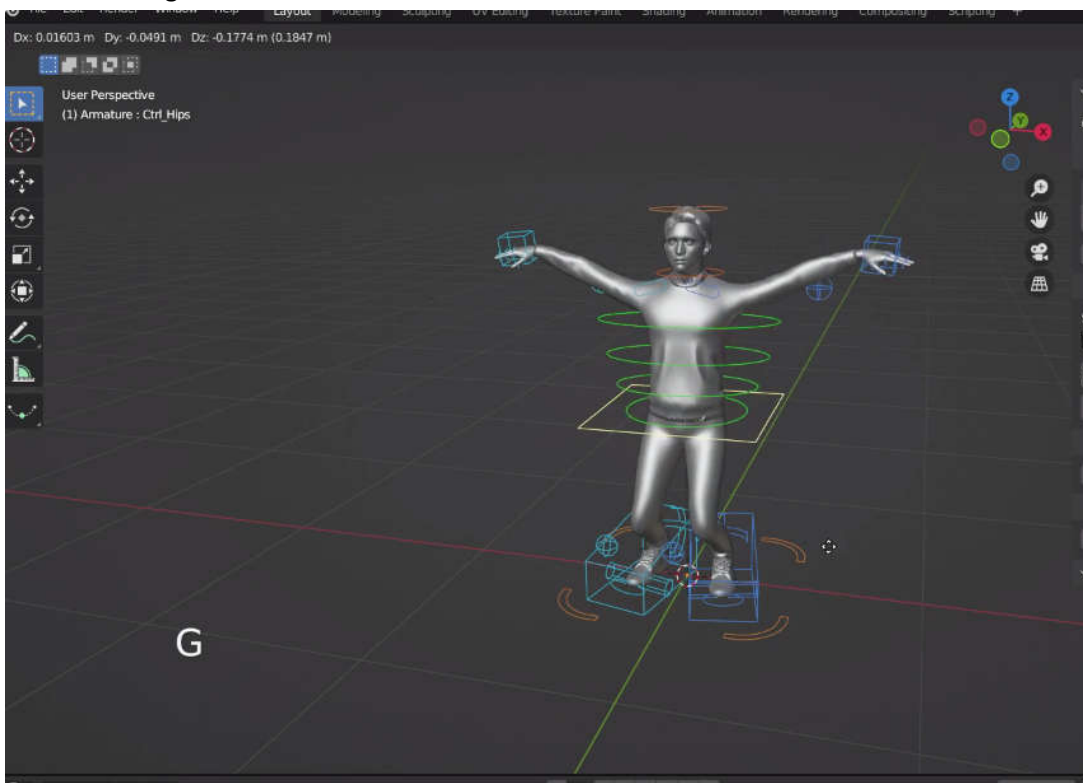


Centang kotak Inverse Kinematics untuk lengan dan kaki, dan Tekan ok.

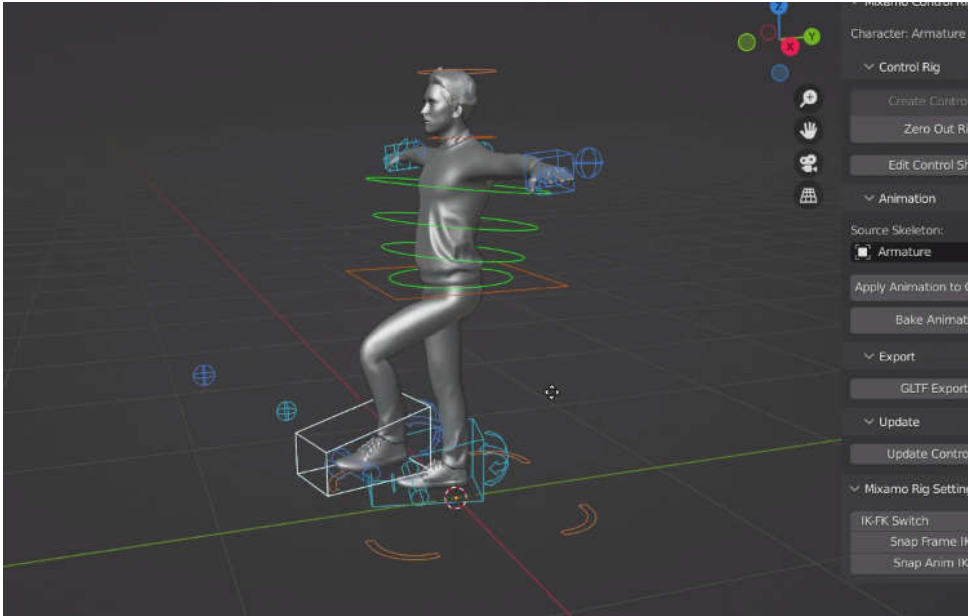


Mari beralih ke mode pose. Sekarang, mari pilih rig kontrol pinggul dan tekan tombol G.

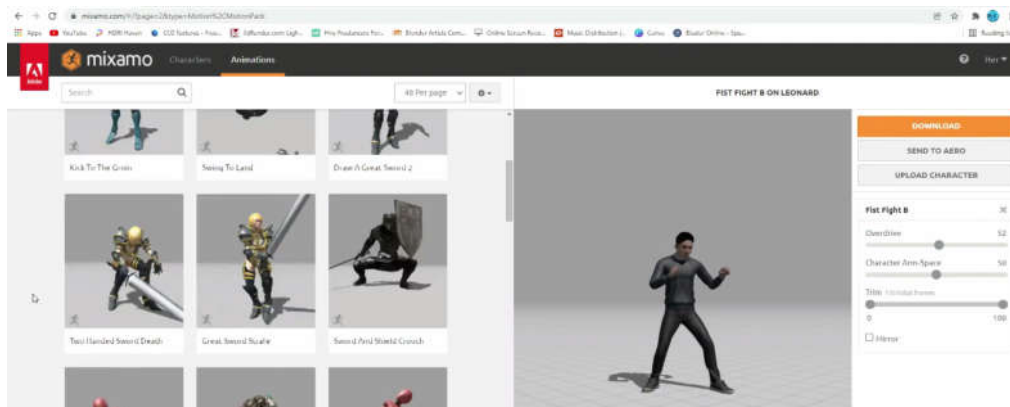
Seperti yang Anda lihat, saat kita menggerakkan rig kontrol pinggul, kaki juga menekuk secara otomatis dengan ini.



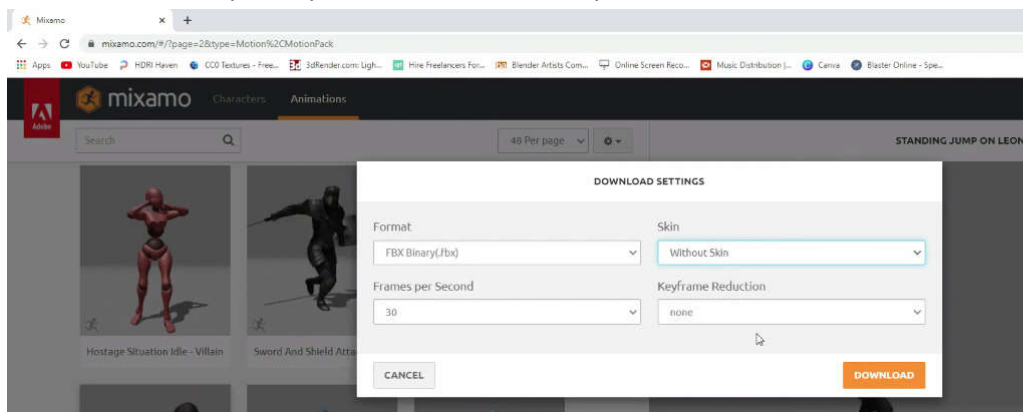
Juga ketika kita menggerakkan rig IK kaki kiri, kita dapat melihat bahwa inverse kinematic bekerja dengan baik.



Kita dapat membuat lebih banyak pose dengan mudah menggunakan perangkat kontrol ini. Kita juga dapat mengimpor file animasi dan menerapkannya ke rig kontrol. Mari kembali ke mixamo lagi. Pilih animasi lain.



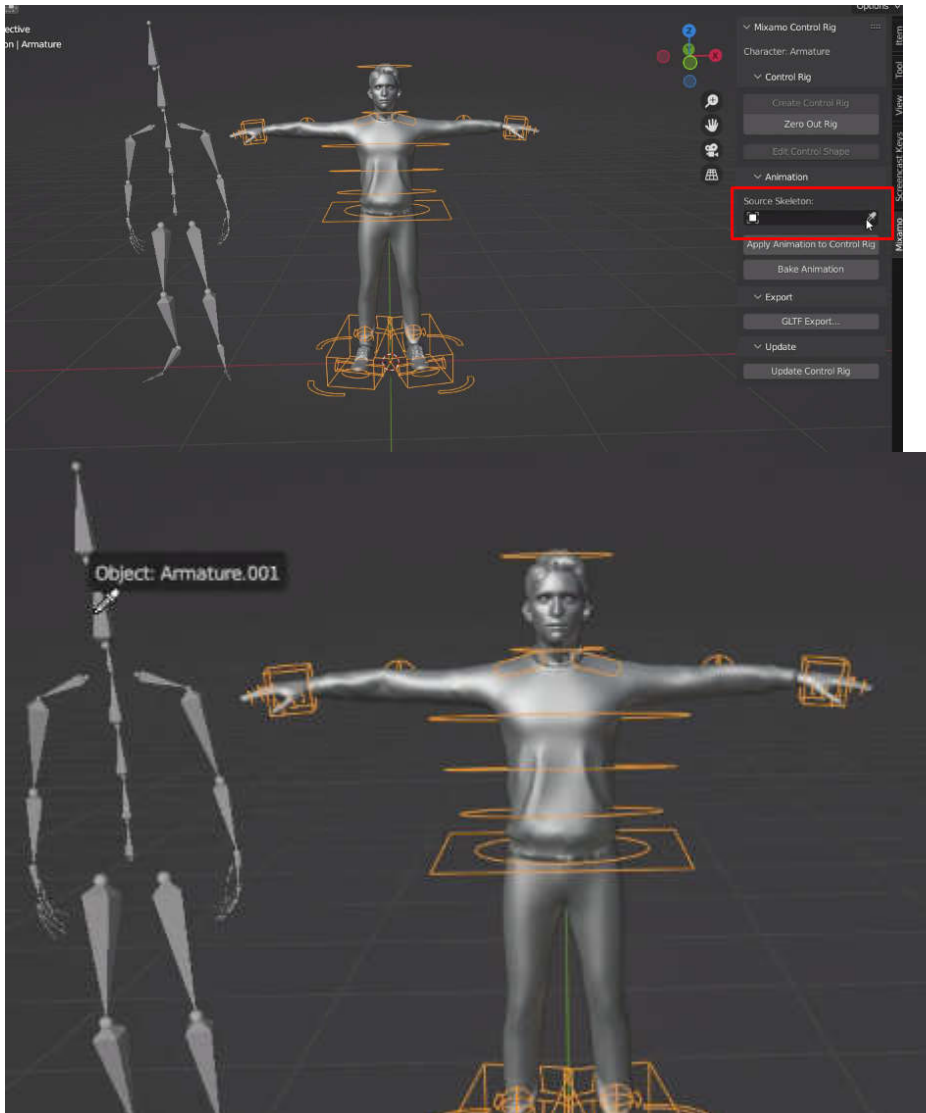
Kita harus memilih opsi tanpa skin. Karena kita hanya butuh animasi.



Kemudian kembali ke blender. Impor file fbx yang baru saja kita unduh.

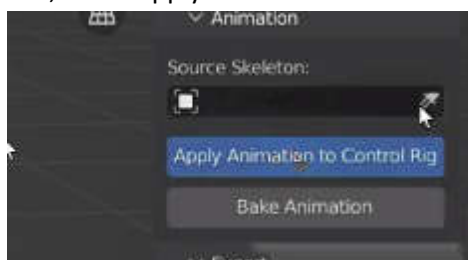


Mari kita hapus kerangka sumber sebelumnya, dan pilih angker baru dengan pipet.

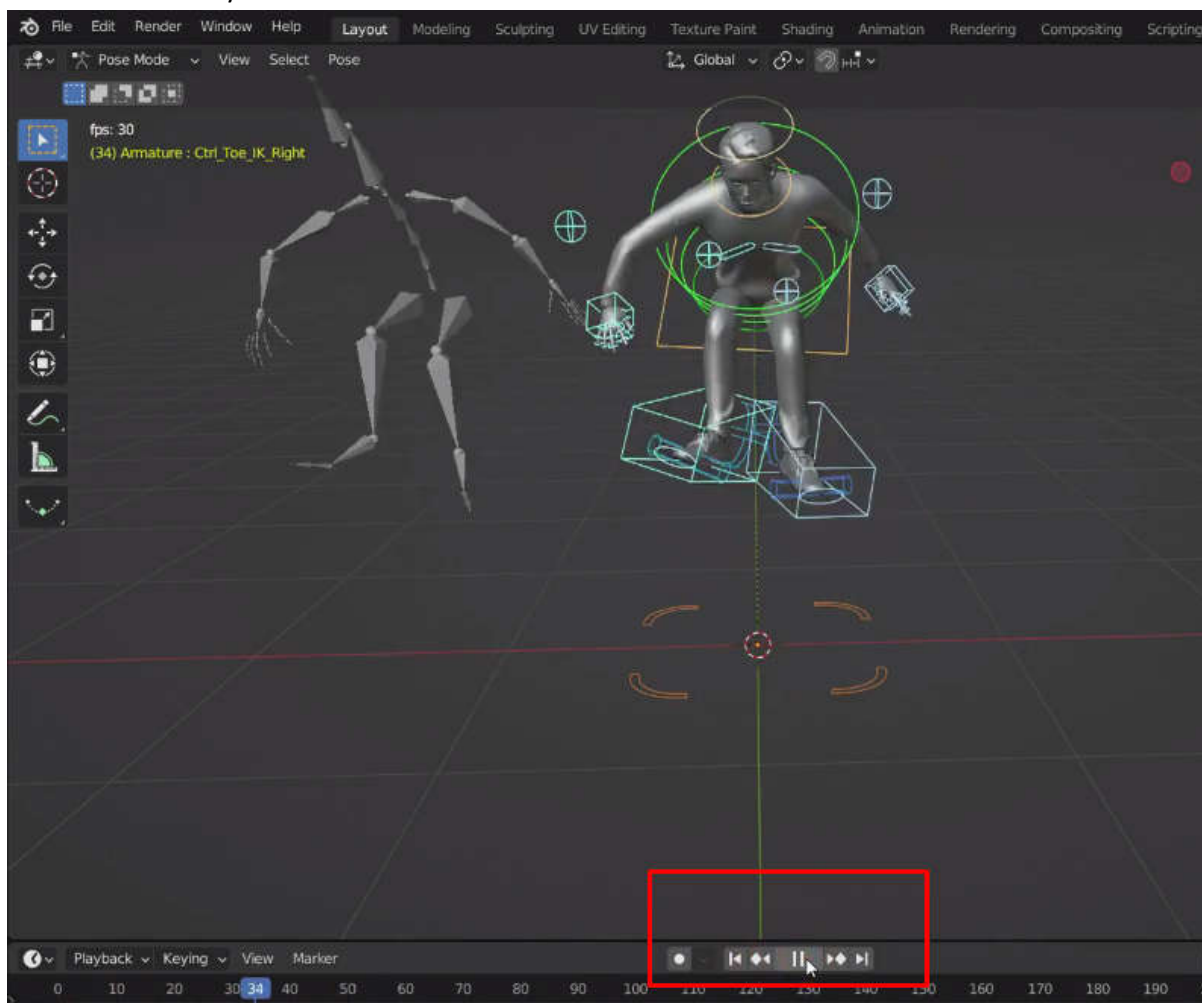


Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

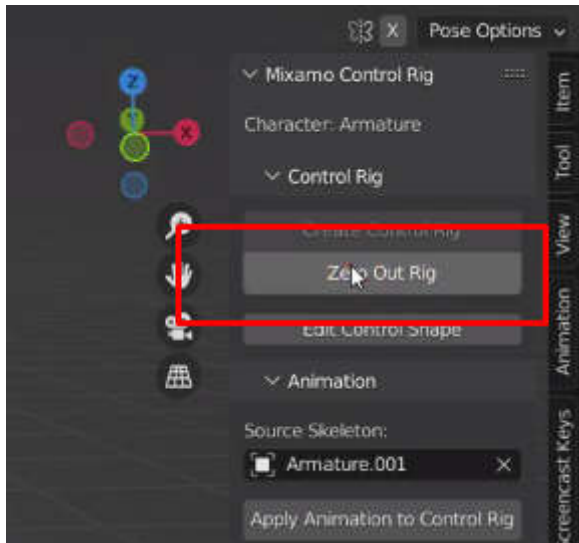
Lalu, tekan apply animation to control rig.



mainkan animasinya.



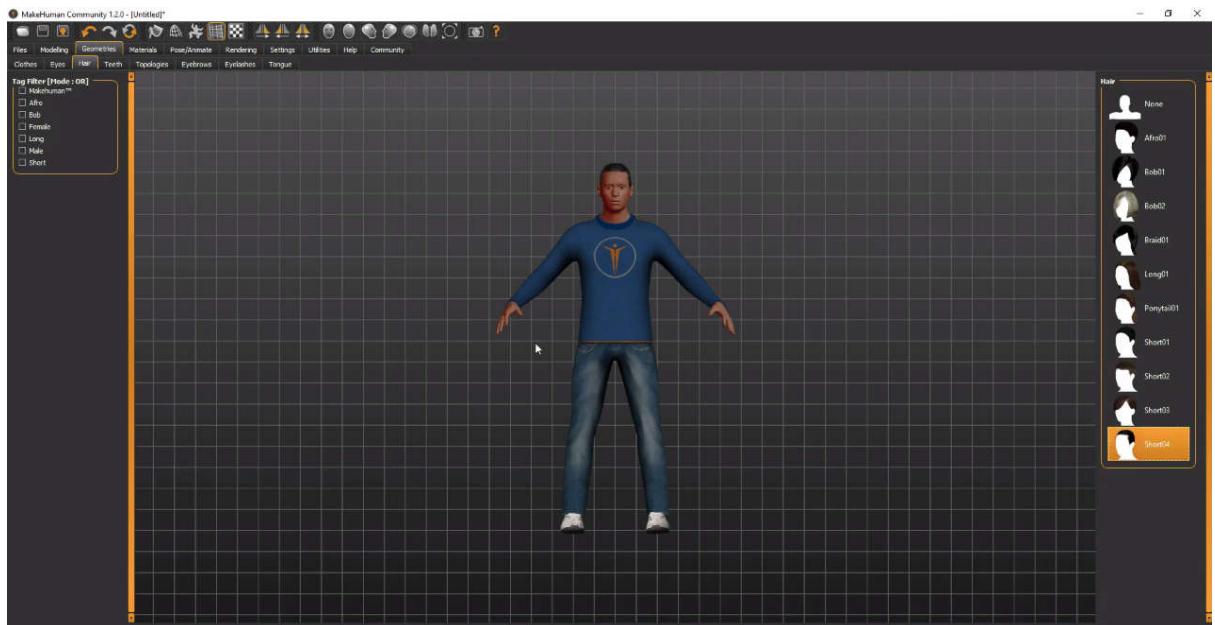
Jika Anda ingin mengubah animasi ini lagi, tekan tombol zero out rig.



Lalu, impor animasi baru, dan terapkan ke rig kontrol.

Kita juga bisa mengunggah karakter khusus kita ke mixamo. Pertama-tama kita membutuhkan karakter. Kita bisa membuat karakter dengan blender atau software grafik komputer lainnya. Tapi, itu akan memakan waktu lama.

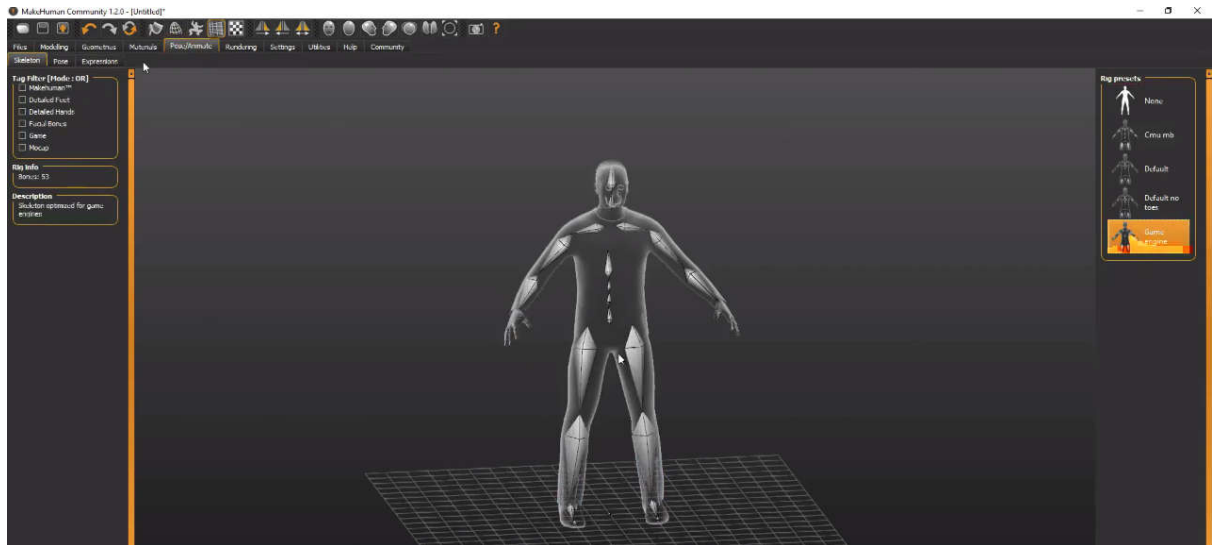
Jangan khawatir. kita bisa memanfaatkan program gratis bernama make human.



Make Human adalah software pembuat karakter 3D gratis dan open source.

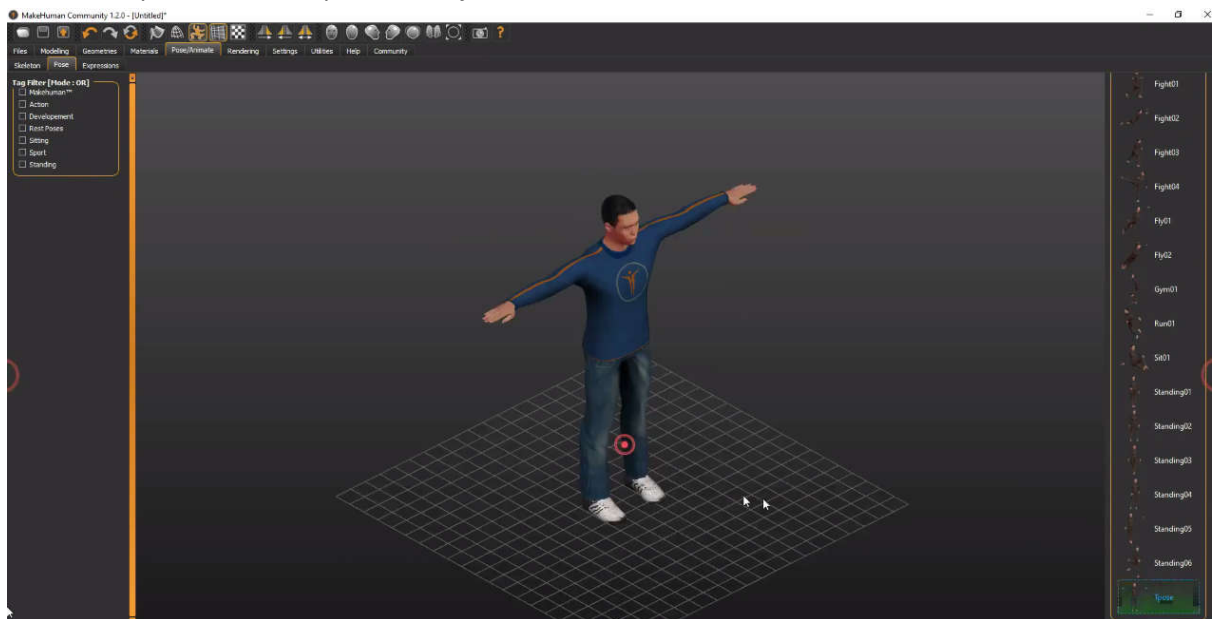
Mari kita membuat situs web manusia dan mengunduh programnya. Kemudian, instal program tersebut ke komputer kita. Kita telah menjalankan programnya. Mari kita pilih jenis kelamin karakter kita terlebih dahulu. Jika kita menggeser ini semua ke kanan, maka akan menjadi laki-laki. Jika kita menggeser ini ke kiri, maka akan menjadi perempuan. Kita juga dapat menyesuaikan usia, tinggi, berat, dan struktur tubuh karakter.

Kita bisa menambahkan pakaian, rambut, dan banyak lagi. Kita juga bisa menambahkan rig untuk animasi.

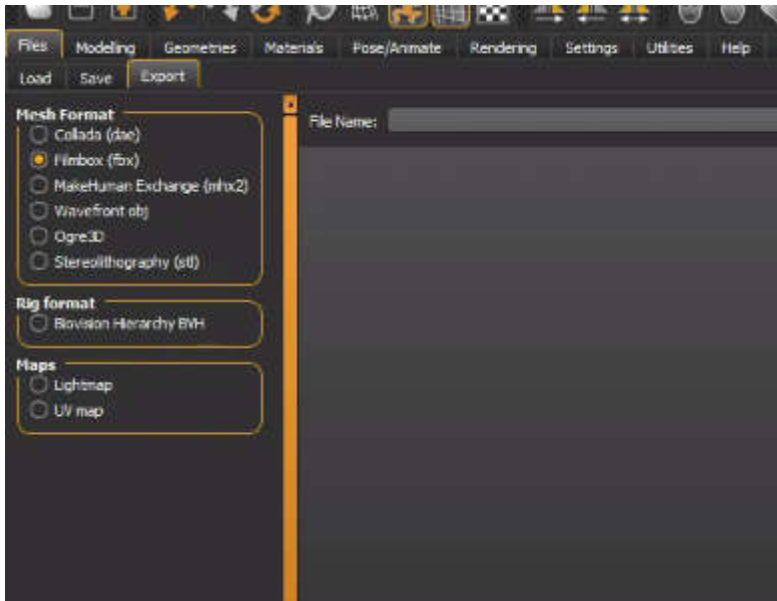


Tapi, kita tidak ingin menambahkan rig. Karena kita akan menambahkan rig di mixamo.

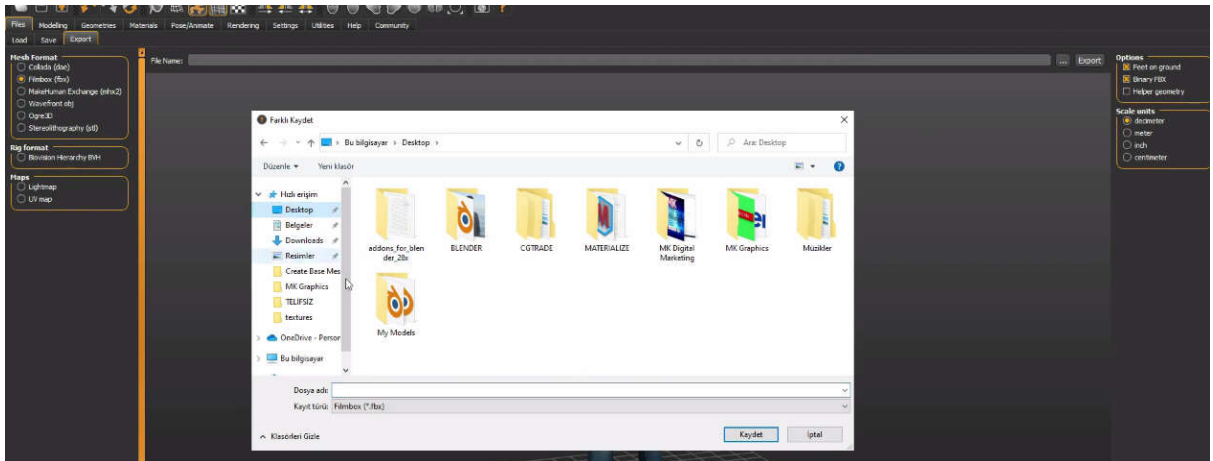
Mari klik tab pose, dan buat pose T menjadi karakter.



Mari ekspor karakter kita. Klik tombol ekspor dan pilih format file sebagai fbx.

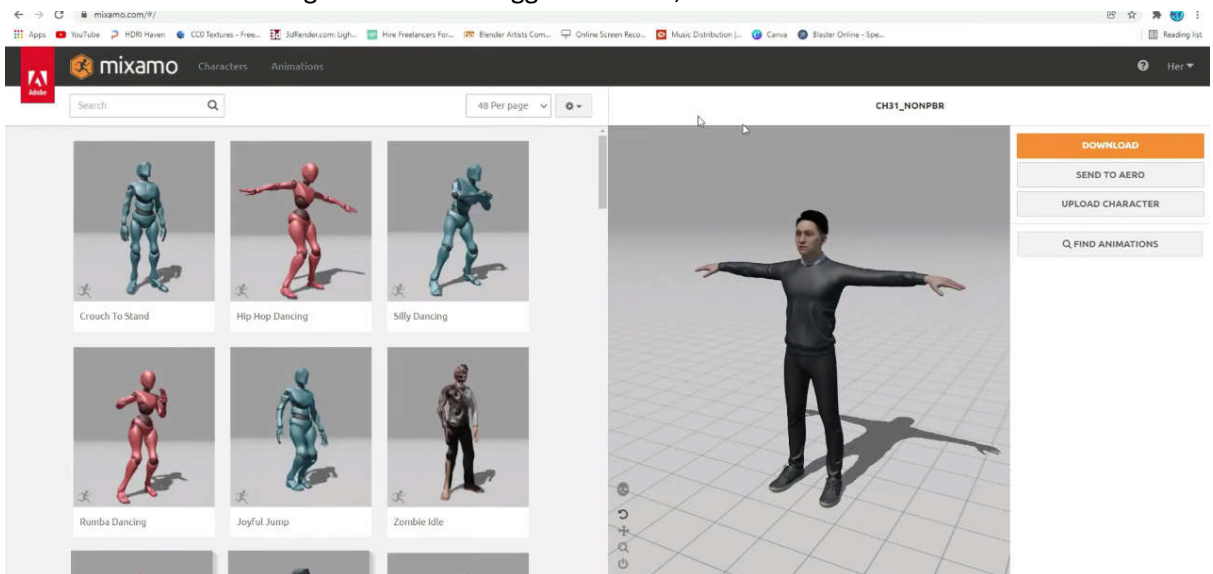


Lalu, mari pilih ke mana kita akan mengekspor file.



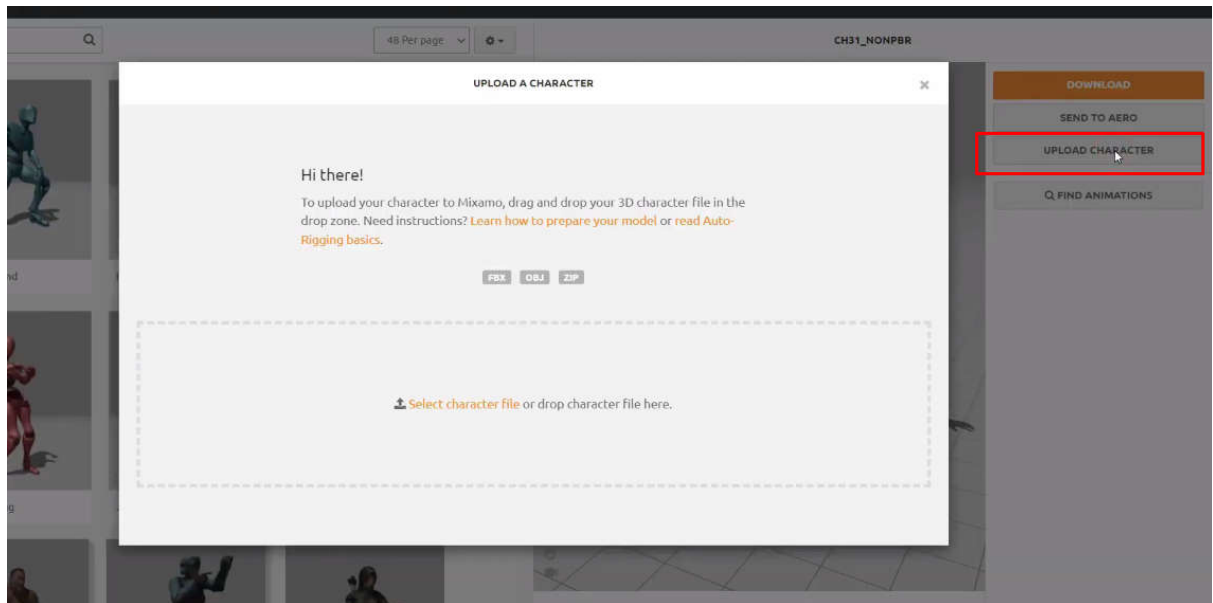
Mari pilih desktop, beri nama karakter kita dan simpan filenya.

Mari kembali ke mixamo lagi. Klik tombol unggah karakter,

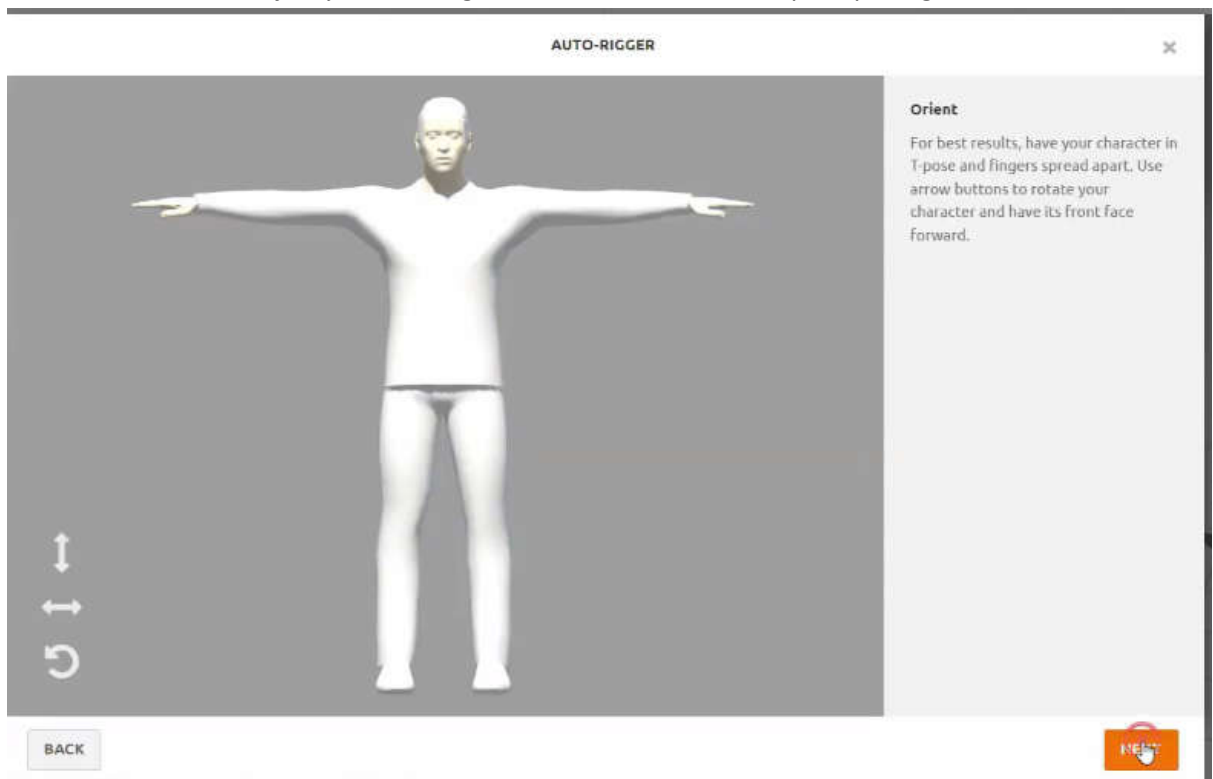


Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

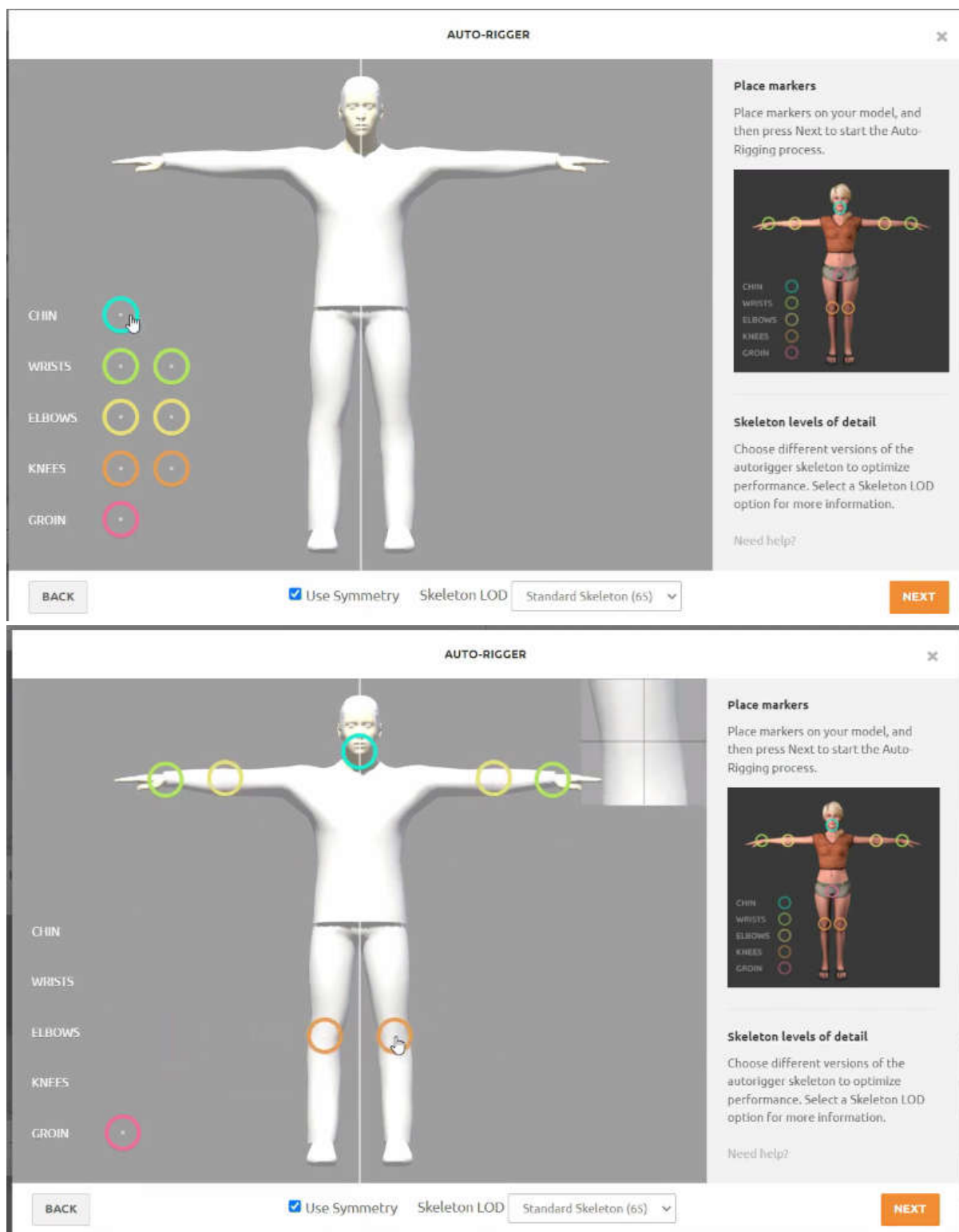
dan pilih file karakter yang baru saja kita buat.



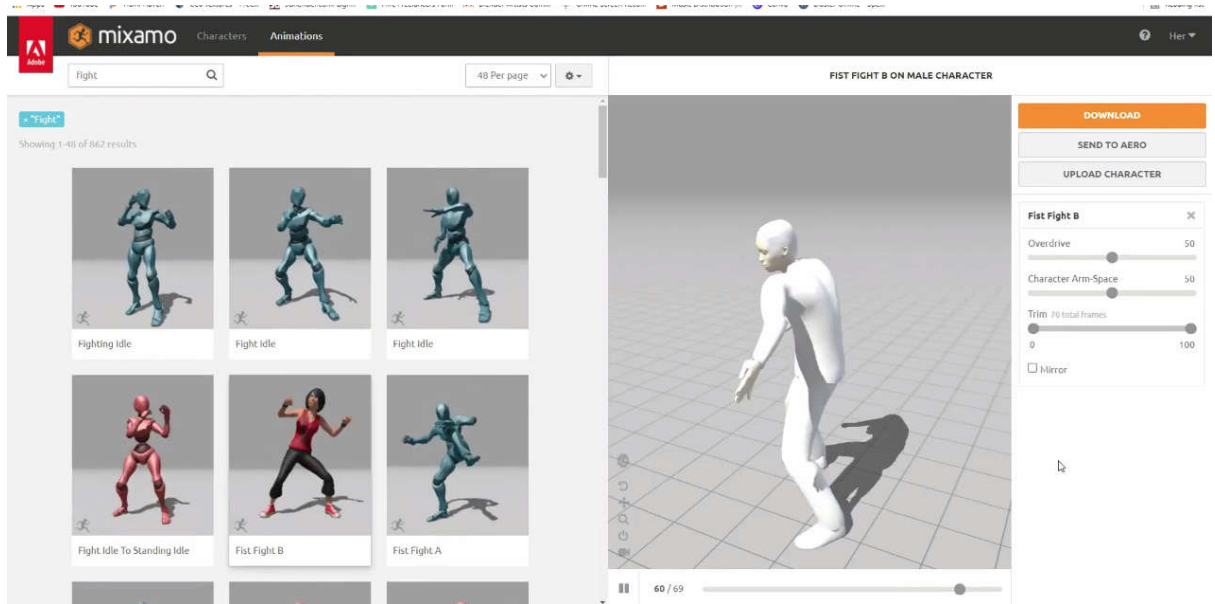
Mari tekan tombol selanjutnya. Sekarang, mari tandai model kita seperti pada gambar referensi.



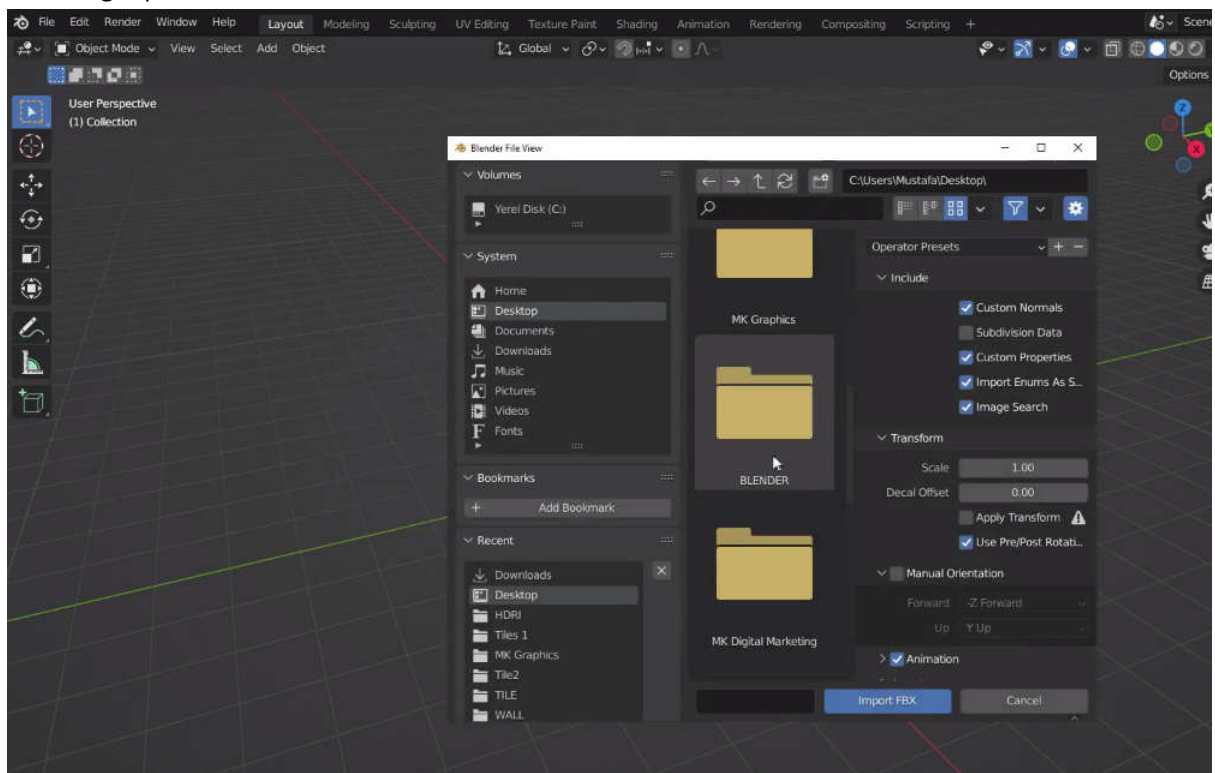
Mari tekan tombol selanjutnya untuk auto rigging.



Setelah proses rigging, mari tambahkan animasi dan unduh karakternya.



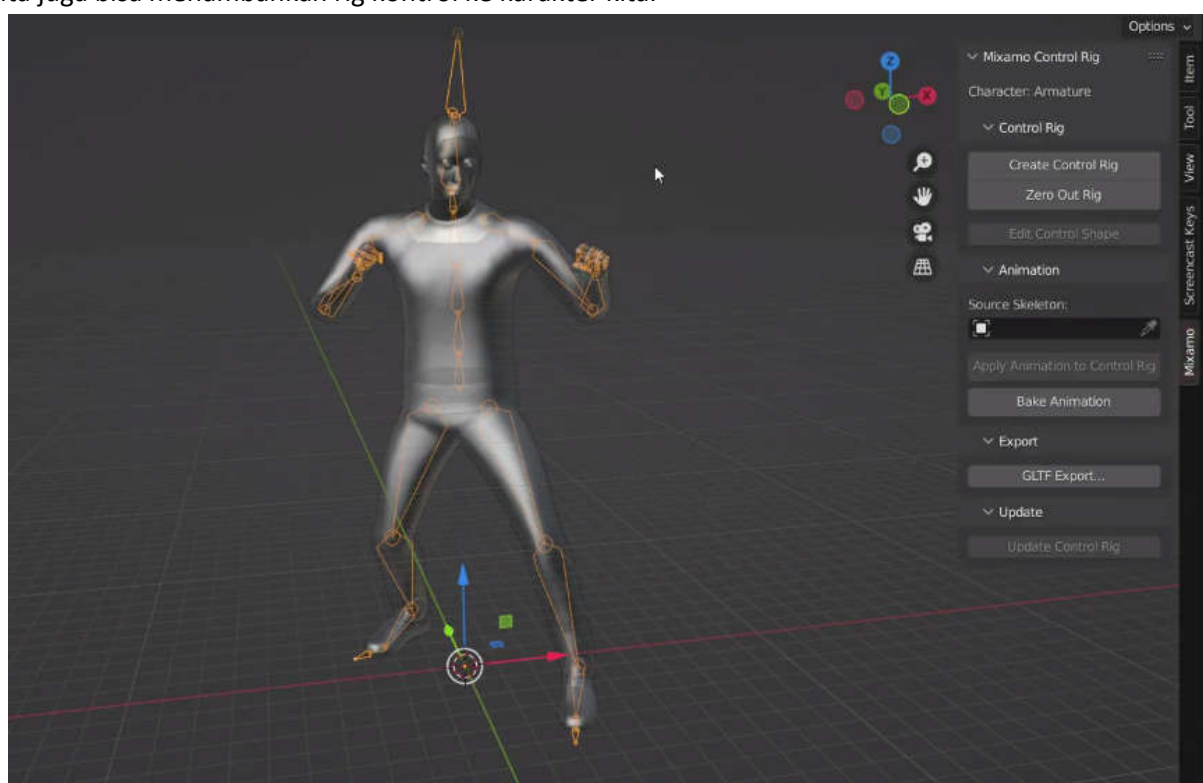
Mari mengimpor karakter ke dalam Blender.



Ayo mainkan animasinya.



Kita juga bisa menambahkan rig kontrol ke karakter kita.

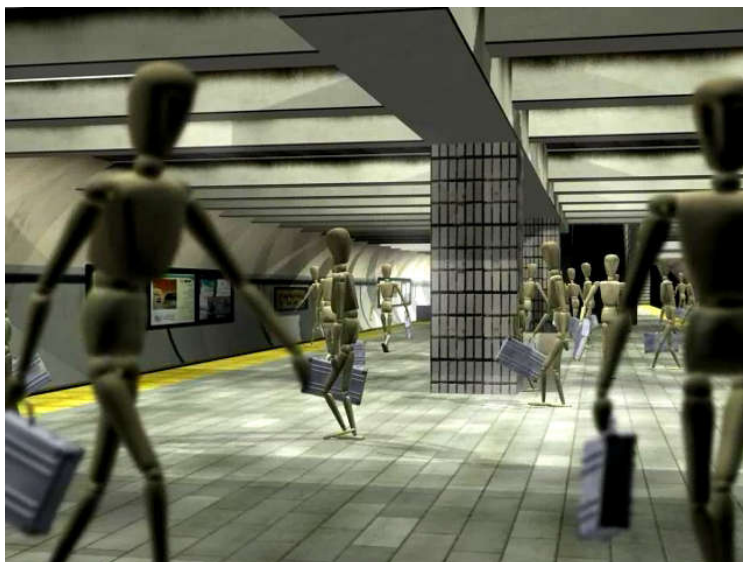


Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

BAB 8 PENERAPAN MOTION CAPTURE PADA 3DS MAX

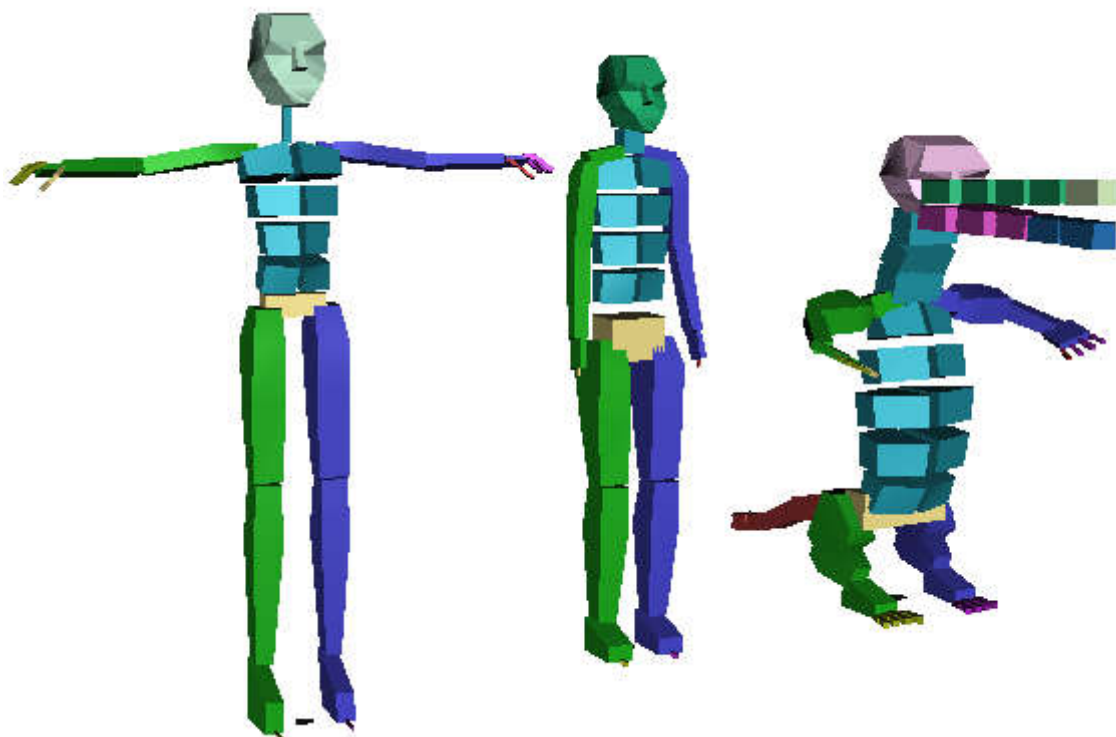
Autodesk 3ds Max, sebelumnya 3D Studio dan 3D Studio Max adalah program grafis komputer 3D profesional untuk membuat animasi 3D, model, permainan dan gambar. software ini dikembangkan dan diproduksi oleh Autodesk Media and Entertainment. yang fleksibel plugin dan harus digunakan pada Microsoft Windows . software ini sering digunakan oleh pengembang video game, banyak komersial TV , dan visualisasi arsitektur . juga digunakan untuk efek film dan pra-visualisasi. 3ds Max menampilkan shader (seperti oklusi ambien dan hamburan bawah permukaan), simulasi dinamis, sistem partikel, radiosity, peta normal pembuatan dan rendering iluminasi global yang dapat disesuaikan antarmuka pengguna, dan bahasa skripnya.

Fitur character studio dalam 3ds Max menyediakan alat profesional untuk menghidupkan karakter 3D. Ini adalah sebuah lingkungan di mana Anda dapat dengan cepat dan mudah membangun kerangka (juga dikenal sebagai character rigs) dan kemudian menghidupkan mereka, sehingga menciptakan gerakan animasi. Anda dapat menggunakan kerangka animasi untuk mendorong pergerakan geometri, sehingga menciptakan karakter virtual.



8.1 biped

biped adalah sosok berkaki dua: manusia atau hewan atau imajiner. Setiap biped adalah angker yang dirancang untuk animasi, dibuat sebagai hierarki yang ditautkan. Kerangka berkaki dua memiliki sifat khusus yang membuatnya langsung siap untuk dianimasikan. Seperti manusia, biped dirancang khusus untuk berjalan tegak, meskipun Anda dapat menggunakan biped untuk membuat makhluk berkaki banyak. Sendi kerangka berkaki dua dibatasi agar sesuai dengan tubuh manusia. Kerangka berkaki dua juga dirancang khusus untuk dianimasikan dengan jejak kaki studio karakter, yang membantu memecahkan masalah animasi umum yaitu mengunci kaki ke tanah.



Objek induk dari hierarki berkaki dua adalah pusat massa objek berkaki dua, yang dinamai *Bip001* secara default.

8.2 Membuat Biped

Tombol untuk membuat Biped muncul di panel Buat di bawah Sistem. Anda membuat biped dengan mengklik tombol ini dan kemudian menyeret di viewport aktif. Anda menentukan ketinggian biped secara interaktif saat Anda menggerakkan kursor.

Selama pembuatan biped, Anda dapat mengubah salah satu pengaturan default yang digunakan untuk menentukan strukturnya. Pengaturan default, untuk Lengan, Tautan Leher, Tautan Tulang Belakang, dan sebagainya, adalah untuk sosok manusia.



Jika Anda mengaktifkan File .fig Terbaru, biped yang Anda buat akan menggunakan parameter yang disimpan di file FIG (gambar) terakhir yang Anda muat.

Mengubah Parameter Biped

Seperti objek 3ds Max lainnya, Anda dapat mengubah parameter biped pada panel Create pada waktu pembuatan. Namun, untuk memodifikasi atau menganimasikan biped, Anda menggunakan parameter pada panel Motion. Untuk informasi selengkapnya tentang mengubah parameter biped, lihat Peluncuran Struktur.

Prosedur

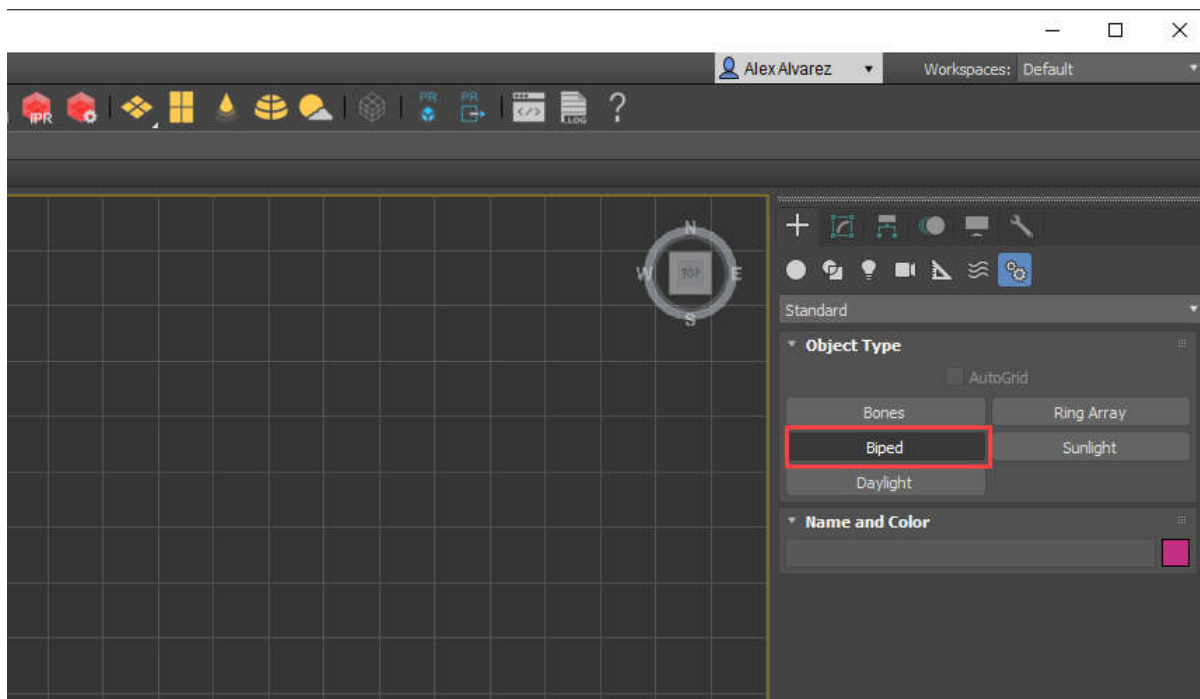
Untuk membuat biped pada sebuah permukaan:

1. Pada  panel Create, klik  (Systems), kemudian pada rollout Object Type, klik Biped.
2. Nyalakan AutoGrid.
3. Pindahkan kursor Anda ke geometri apa pun di area pandang. Alat Transform bergerak dengan kursor Anda untuk menunjukkan lokasi AutoGrid.
4. drag biped.
biped akan bersentuhan dengan geometri.

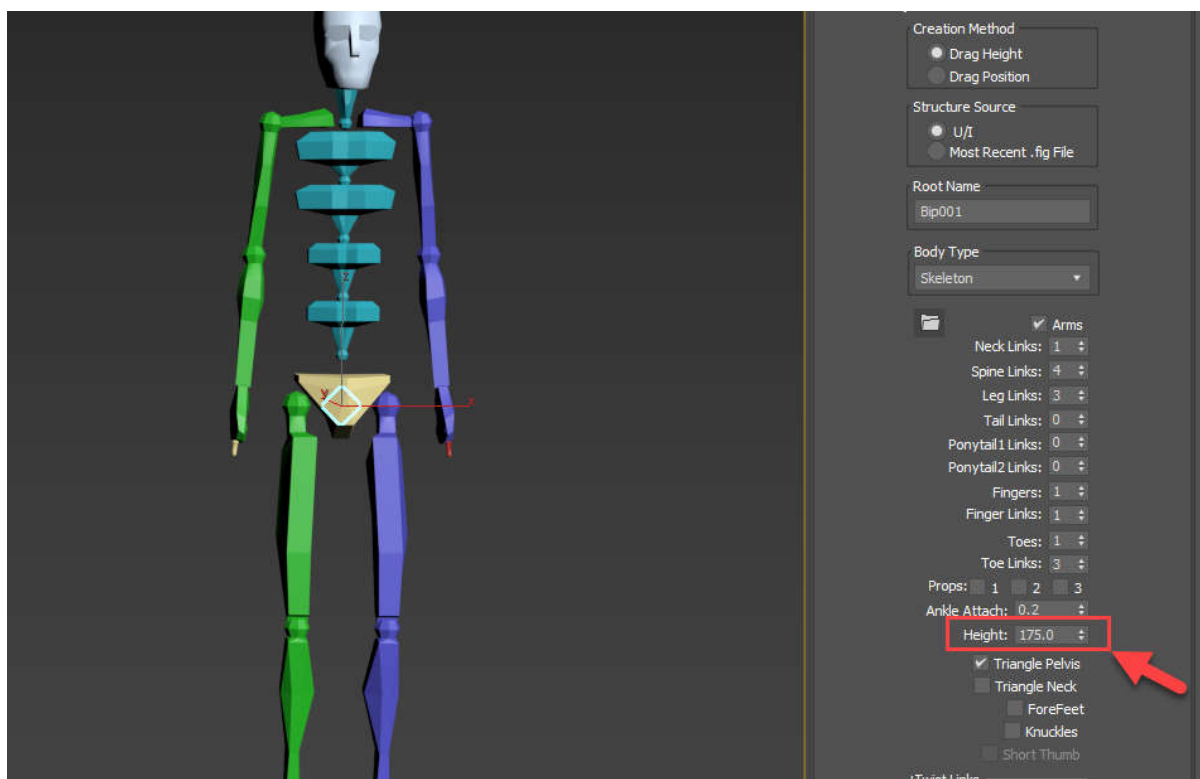
8.3 Penargetan motion capture BVH dalam 3ds Max

siapkan karakter Biped untuk baking data animasi kita

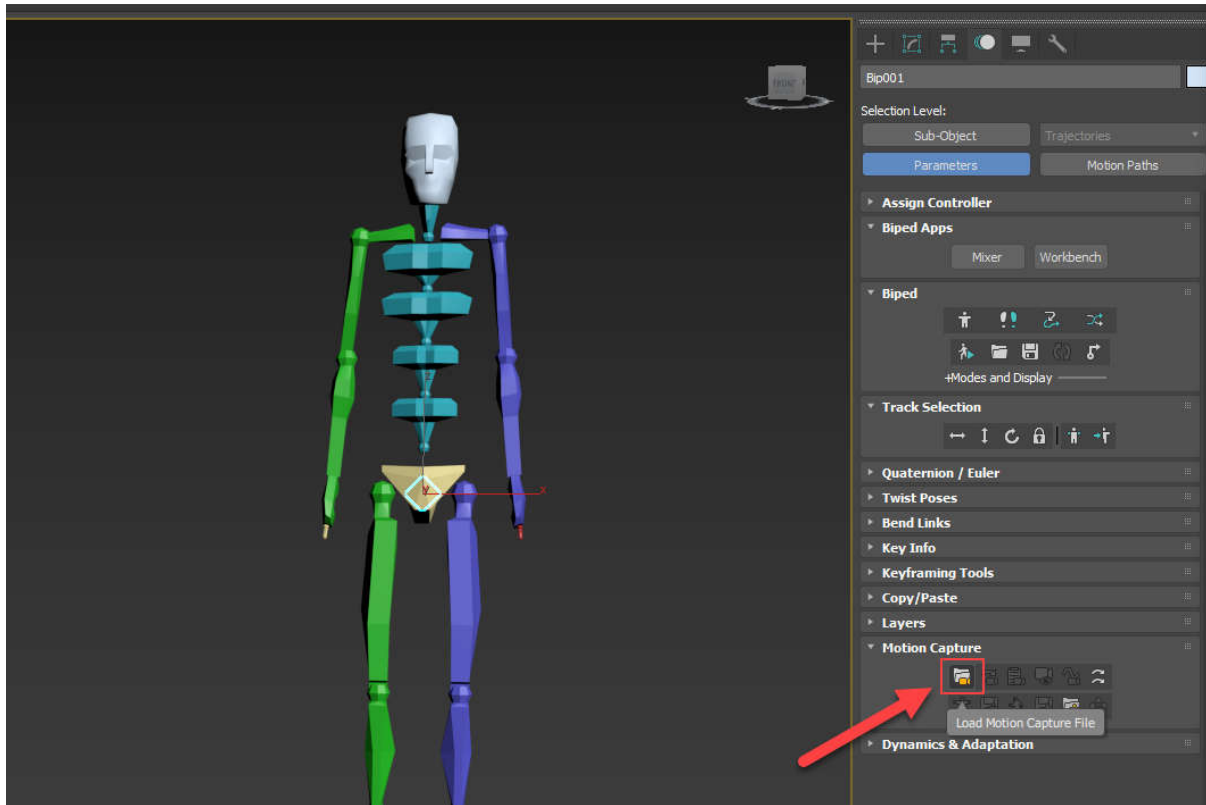
- Buka 3ds Max
- Klik BIPED



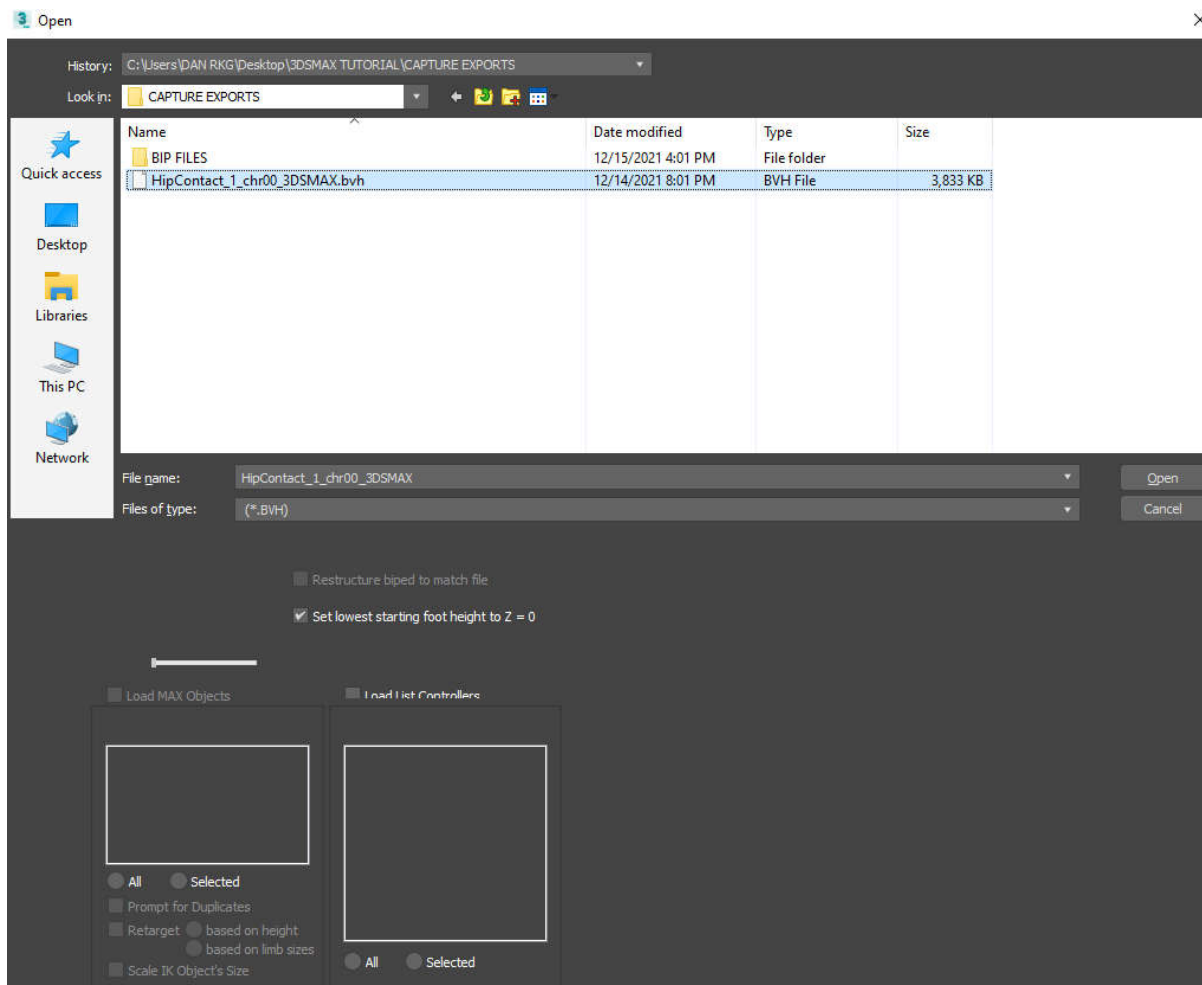
- Perbarui Tinggi karakter menjadi 175



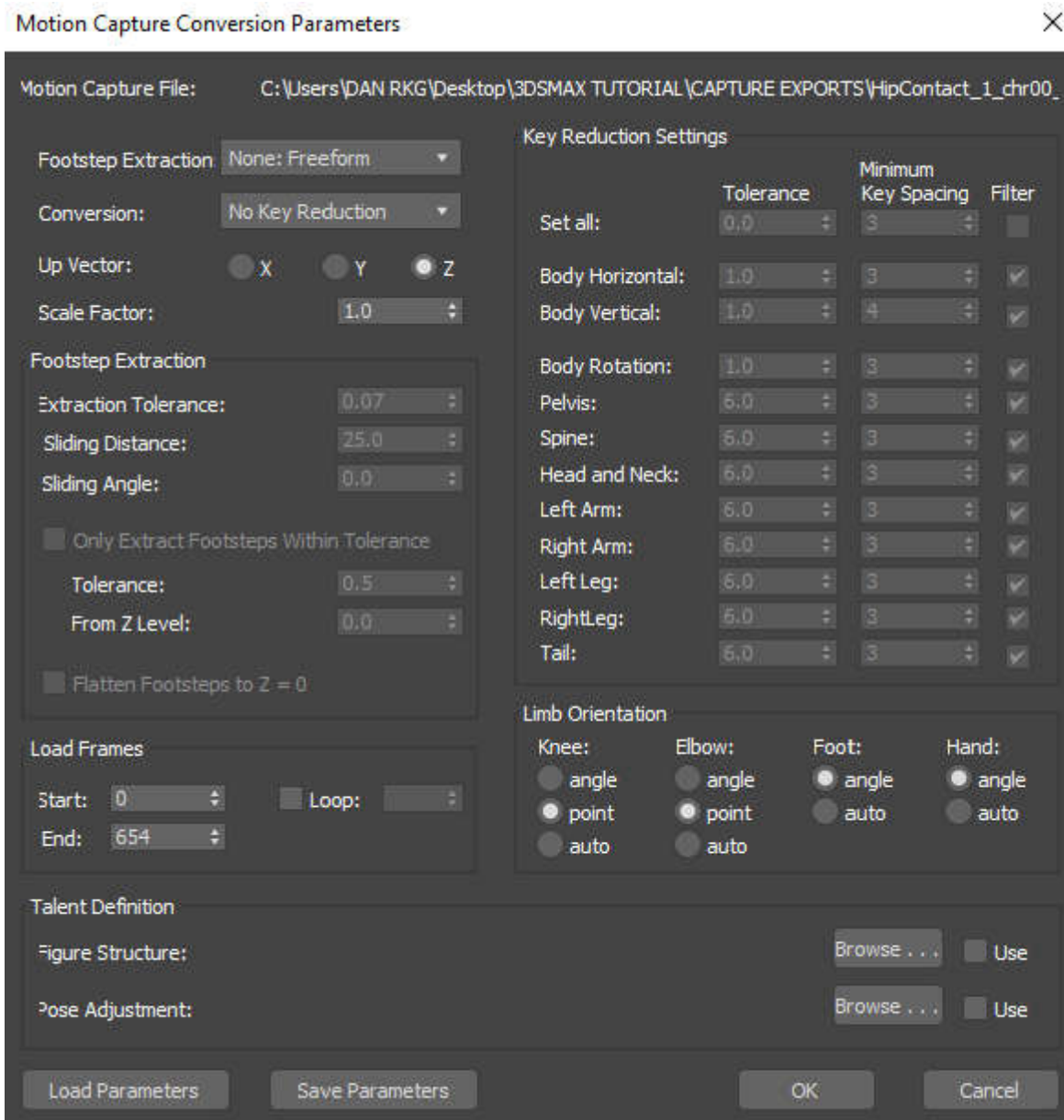
- Impor file BVH yang diekspor dari Axis Studio dengan Memuat File Motion capture



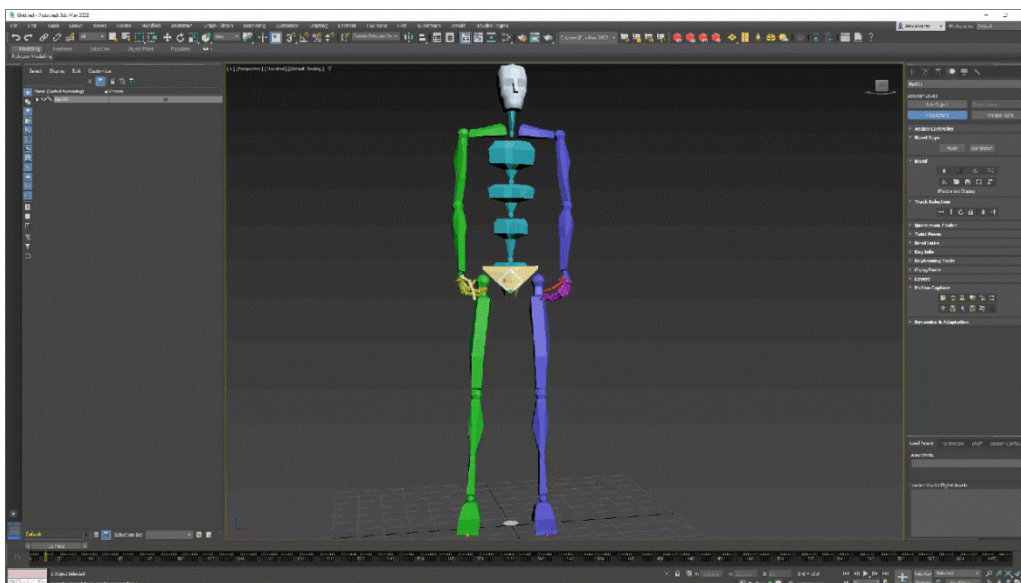
- Di jendela impor, perbarui Jenis File untuk mencari untuk file .BVH
- Pilih hasil tangkapan Anda



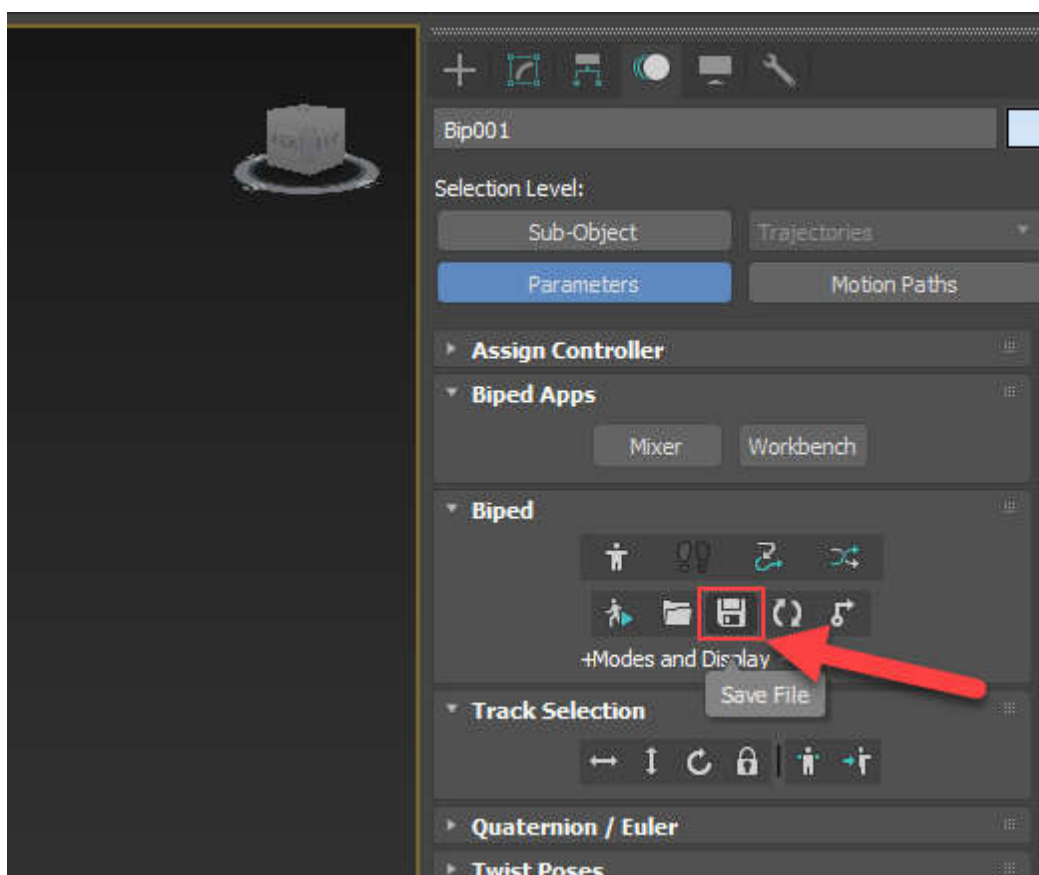
- Di jendela impor, biarkan opsi default saja



- Animasi Anda sekarang dibuat menjadi karakter BIPED.



- Sekarang simpan hasilnya ke folder lokal Anda. Ini akan membuat file asli .bip 3Ds Max

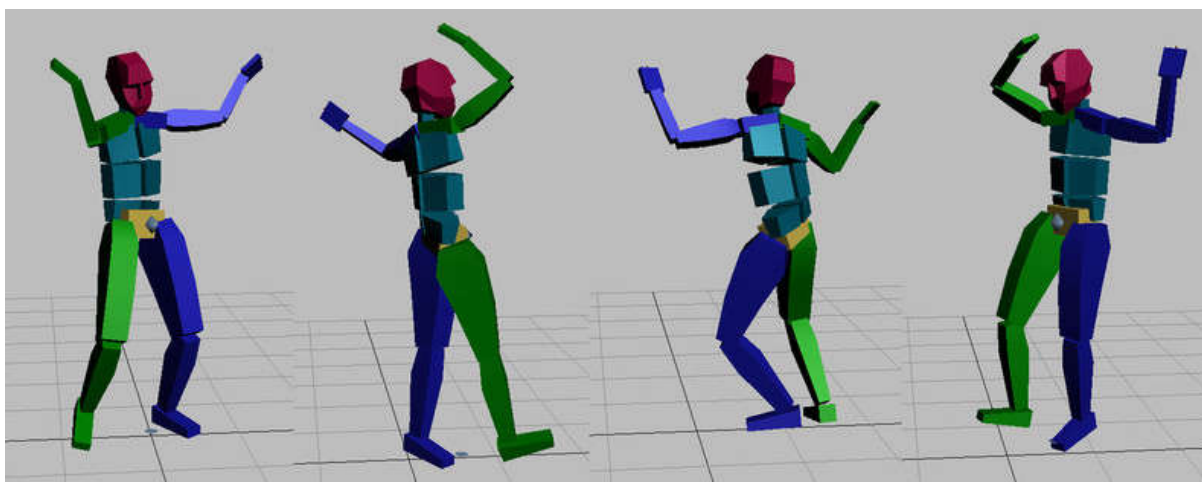


Kita dapat men save footstep motion untuk penggunaan di masa datang dari Biped rollout, pilih Save File.

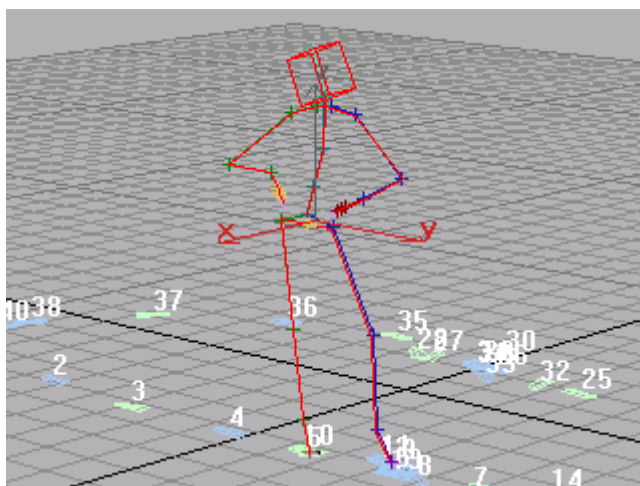
Save File dialog displays. Tentukan folder dimana akan disimpan. Beri nama **motion tersebut** dan klik Save. footstep motion disimpan dalam BIP file.

8.4 Bekerja dengan Data Motion-Capture


Motion capture adalah praktik mendapatkan data gerak dari aktor langsung yang melakukan berbagai tindakan. Data gerak ditangkap (diambil) melalui sensor yang ditempatkan di sendi dan ekstremitas aktor.





3ds Max tidak melakukan motion capture, tetapi menerima data motion capture dalam format yang paling umum digunakan.

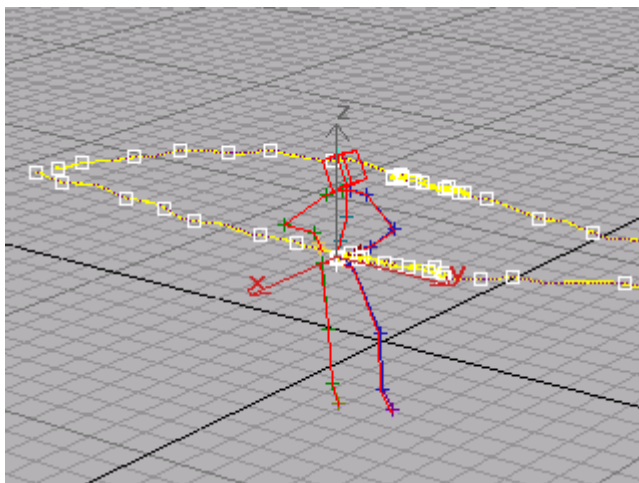


Tampilkan dan bandingkan lintasan:

1. Pilih bagian mana pun dari biped.
2.  Pada peluncuran Motion Capture, klik Show Buffer Trajectory.

Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)


3.  Dalam peluncuran Biped, buka bilah ekspansi. Di grup Tampilan, klik Lintasan.
4.  Di grup Tampilan, klik Preferensi.
Dialog Preferensi Tampilan muncul.
5. Di grup Lintasan, aktifkan Basis Tulang. Klik OK untuk menutup dialog.





6. Pilih berbagai bagian berkaki dua untuk membandingkan lintasan yang disaring dan mentah untuk setiap bagian tubuh berkaki dua.
7. Pada grup Tampilan, klik Preferensi lagi. Pada dialog Preferensi Tampilan, matikan Tampilkan Seluruh Lintasan. Klik Oke.
Sekarang lintasan hanya menampilkan 50 frame sebelum dan sesudah frame saat ini. Perhatikan bahwa lintasan antara data buffer dan filter sangat dekat.

8.5 Menggunakan High-Frequency Data dan Looping

Dengan gerakan frekuensi tinggi, seperti lari cepat, Anda dapat kehilangan nuansa gerakan jika menggunakan pengaturan filter default. tapi sebenarnya ini bisa dimodifikasi, pada bagian ini Anda akan meminimalkan pengurangan kunci pada kaki seorang sprinter sehingga kaki mengikuti data gerakan mentah lebih dekat. Anda juga akan belajar mengulang file gerakan dalam dialog Parameter Motion capture.


1. Di Front viewport, buat biped.
2.  Pada panel Motion, dalam peluncuran Motion Capture, klik Load Motion Capture File. Memilih *.csm* sebagai jenis file.
3. Buka *file.csm*.
Dialog Parameter Konversi Tangkapan Gerakan ditampilkan.
4. Pilih Aktif untuk Ekstraksi Langkah Kaki, dan Gunakan Pengurangan Kunci untuk Konversi. Klik Oke.
Klip memuat tetapi tidak memiliki jejak kaki.

5.  Dalam peluncuran Biped, buka bilah ekspansi. Di grup Tampilan, alihkan dari Objek ke Tulang di flyout.
6.  Dalam peluncuran Motion Capture, aktifkan Show Buffer. Sosok tongkat merah mewakili data motion capture mentah. Sosok tongkat lainnya mewakili data yang difilter. Perhatikan bahwa Anda tidak dapat menggunakan mode In Place dengan Show Buffer.

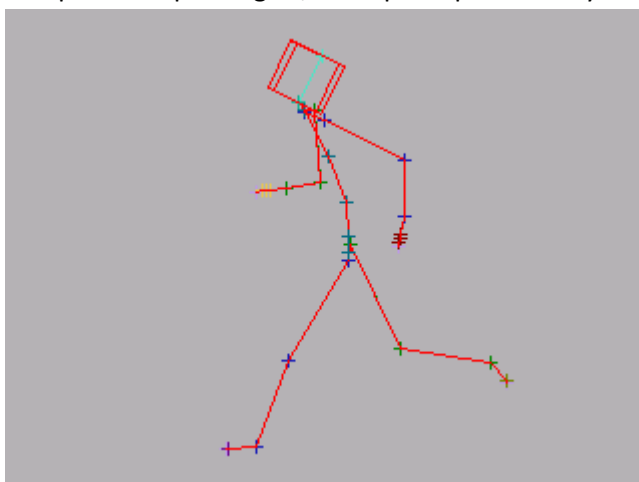
Bandingkan datanya:

1. Gunakan **Alt+W** untuk memaksimalkan Left viewport.
2. Klik Mainkan Animasi.
Data yang difilter tertinggal dari data mentah di kaki, membuat sprint lebih terlihat seperti pengocokan cepat. Namun, ini sulit dilihat karena pemutaran cepat dan bracketing kunci.




3. Klik kanan tombol Mainkan Animasi. Dalam dialog Time Configuration, di grup Playback, aktifkan Real Time dan 1/4x Speed untuk memperlambat gerakan. Matikan key bracketing dengan memilih Customize > Preferences > Animation > Key Bracket Display group > None.
4.  Dalam peluncuran Motion Capture, klik Convert From Buffer. Buffer motion capture berisi data motion capture mentah. Anda akan memasukkan parameter untuk memperbaiki kelambatan dan mengekstrak langkah kaki. Dalam dialog Parameter Konversi Tangkapan Gerakan, pilih Aktif untuk Ekstraksi Langkah Kaki dan Gunakan Pengurangan Kunci untuk Konversi.
5. Di grup Ekstraksi Langkah Kaki, atur Toleransi Ekstraksi ke **1.23**.
6. Di grup Key Reduction Settings, ubah Minimum Key Spacing untuk kedua kaki menjadi **1**. Klik Oke.
Kaki yang disaring cocok dengan kaki mentah selama sprint. Kaki memiliki kunci di

hampir setiap bingkai, tetapi biped lainnya memiliki pengurangan kunci.



Mengulangi file tangkapan gerak:

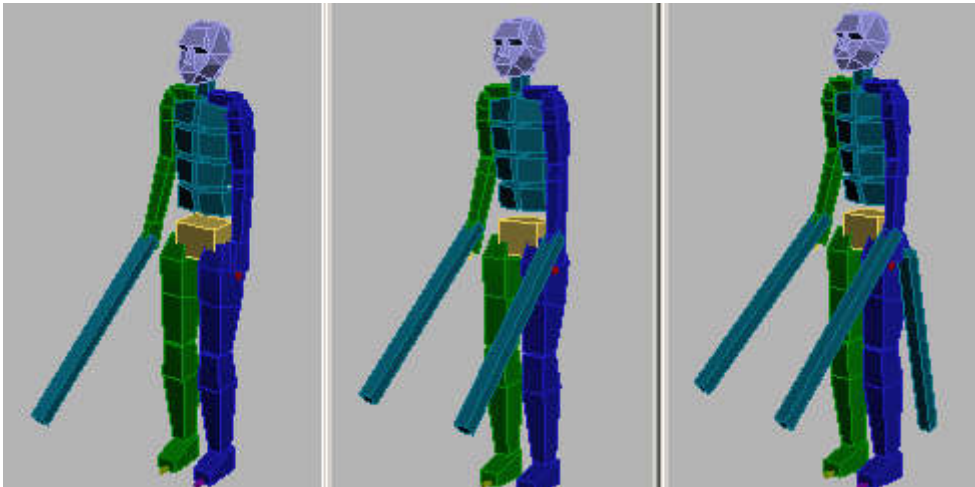
1.  Dalam peluncuran Motion capture, klik Muat File Motion capture. Memilih *.csm* sebagai jenis file.
2. Buka *file* *berformat* *.csm*. Dialog Parameter Konversi Tangkapan Gerakan ditampilkan. Semua pengaturan pada dialog ini harus sama dengan yang Anda masukkan sebelumnya.
3. Di grup Load Frames, aktifkan Loop dan masukkan **1** di kolom Loop. Klik Oke. Gerakan sprint dilingkarkan, menciptakan periode sprint yang lebih lama. Anda juga dapat mengulang gerakan dalam Grafik Aliran Gerak dengan hasil serupa.

8.6 motion capture file yang memiliki prop data

Jika file motion capture Anda menyertakan data penyangga, 3ds Max dapat membaca data tersebut dan menerapkannya ke objek penyangga.


Saat Anda memuat file CSM (file character studio marker), data gerakan diterapkan ke prop. Jika penyangga berpindah dari satu tangan ke tangan lainnya, program secara otomatis menentukan peralihan ruang koordinat penyangga saat penyangga dipindahkan dari satu tangan ke tangan lainnya.

Saat memuat file BIP untuk penargetan ulang ke karakter lain, alat peraga bergerak dengan masing-masing tangan. Perangkat lunak memastikan bahwa tangan bersatu pada posisi yang tepat pada penyangga selama transfer.



Biped dengan banyak penyangga sebagai pedang


Muat file CSM:

1. Buat biped dengan satu penyangga.
Biped muncul dengan penyangga tunggal yang ditampilkan sebagai kotak yang diposisikan di dekat tangan kanan biped.
2.  Buka panel Gerakan. Dalam peluncuran Motion Capture, klik Load Motion Capture File.
3. Pilih jenis file *.csm*. klik Buka.
Dialog Parameter Konversi Motion capture ditampilkan.
4. Pilih opsi filter default. Jika Footstep Extraction diaktifkan, langkah kaki akan muncul.
Klik Oke.
Biped menyesuaikan diri dengan data gerak.

juga kita bisa bekerja dengan layer

kita bisa menggunakan lapisan untuk menambahkan perubahan. File gerak adalah gerakan berlari. Anda akan menambahkan lapisan dan memutar pusat massa sehingga karakter berjalan di dinding.

Muat gerakan:

1. Buat biped.
2.  Buka panel Gerakan. Dalam peluncuran Motion Capture, klik Load Motion Capture File.
3. Buka *jez.sprint.csm*.
4. Pada dialog Parameter Konversi Tangkapan Gerak, defaultnya mencakup tidak ada ekstraksi langkah kaki dan tidak ada pengurangan tombol. Klik OK untuk menerima

default.




File gerak dimuat dengan kunci di setiap bingkai.

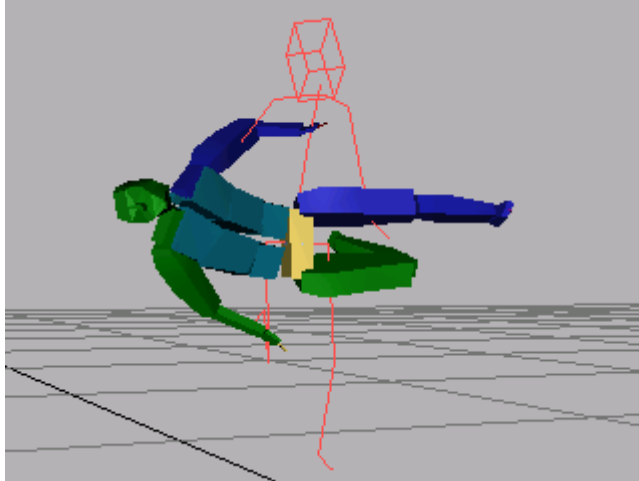
5. Klik Mainkan untuk melihat gerakan berlari.



Menambahkan lapisan:

Langkah-langkah berikut mengilustrasikan penggunaan tombol Snap Set untuk mengubah posisi asli pelari.

1.  Pada peluncuran Lapisan, klik Buat Lapisan.
Lapisan dibuat.
2. Pindahkan penggeser waktu ke bingkai 8, di mana kaki kiri karakter baru saja menyentuh ke bawah.
3. Aktifkan Kunci Otomatis untuk memulai animasi.
4. Dalam peluncuran Pemilihan Track, klik Rotasi Tubuh.
Ini memilih pusat jalur rotasi massa.
5. Di area pandang, klik dan seret untuk memutar pusat massa **-90** derajat terhadap sumbu X.
6. Klik Mainkan Animasi.
Karakter berjalan di dinding tak terlihat.
7. Pindahkan penggeser waktu ke bingkai 0.
8.  Pada peluncuran Layers, klik Snap Set Key.
Karakter terkunci ke posisi semula.
9. Pindahkan penggeser waktu ke bingkai 16.
Kaki kanan baru saja menyentuh ke bawah.
10.  Pada peluncuran Layers, klik Snap Set Key.
Karakter terkunci ke posisi semula.
11. Klik Mainkan Animasi untuk melihat hasilnya.
Karakter berjalan di tanah, lalu di dinding, dan kembali ke tanah lagi.



Untuk meningkatkan efeknya, Anda dapat memutar objek tulang belakang dan meregangkan kaki agar karakter tersebut tampak melompat ke dinding.

BAB 9 PENERAPAN MOTION CAPTURE PADA MAYA

Apa Itu Perangkat Lunak Maya? Perangkat lunak Autodesk Maya adalah aplikasi memungkinkan untuk membuat animasi 3D, pemodelan, simulasi, rendering, dan lainnya. aplikasi ini serbaguna, dan banyak yang menganggapnya sebagai standar industri untuk animasi. Banyak studio film terkenal menggunakan Maya, termasuk Blue Sky Studios, Framestore, dan Moving Picture Company. Faktanya, perangkat lunak tersebut telah digunakan untuk menganimasikan film pemenang penghargaan seperti "Frozen" dan "Wreck It Ralph".

Karena Motion capture data adalah data yang dihasilkan dengan menangkap gerakan manusia nyata atau hewan dan digitalisasi melalui alat khusus penangkap gerak / motion capture 3D sehingga dapat diterapkan untuk karakter animasi 3D termasuk pada aplikasi maya . Motion capture data berisi keyframe animation yang dibutuhkan dalam animasi maya. Motion capture data sangat berguna karena dapat menangkap nuansa halus gerak dan gesture yang membuat tindakan karakter seperti hidup. Motion capture data yang secara substansial dapat mengurangi pekerjaan yang berkaitan dengan pose dan keyframing karakter Anda.

Banyak sumber data motion capture yang tersedia di internet, salah satunya dari Carnegie Mellon University. Meskipun data motion capture tidak dikhususkan pada maya tetapi dapat diimport dengan mudah berikut langkah langkahnya

9.1 Import Motion Capture ke Maya

Buka alamat webnya di <http://mocap.cs.cmu.edu/>

Opera | MAYA 6 ... | MAYA RI... | free 3d ... | Fbx 3D | Creative ... | Carnegie... | Downlo... | http://m...

Web mocap.cs.cmu.edu/search.php Elements: 1/1 Search with Google

CMU Graphics Lab Motion Capture Database

[Home](#) | [Search](#) | [Tools](#) | [Info](#) | [FAQs](#) | [Rendered Movies](#) | [Resources](#)

View All: [Subjects](#) | [Motions](#) | **Browse:** [Motion Categories](#)

Search Help

subject number | motion or keyword
(e.g. 41) (e.g. run)

Subject #1 (climb, swing, hang on playground equipment) [file index](#) - - asf framerate Feedback

Image #	Trial #	Motion Description	tvd	c3d	amc	mpg	Animated	framerate	Feedback
1	1	playground - forward jumps, turn around	tvd	c3d	amc	mpg	Animated	120	Feedback
2	2	playground - clmb	tvd	c3d	amc	mpg	Animated	120	Feedback
3	3	playground - clmb, hang, swing	tvd	c3d	amc	mpg	Animated	120	Feedback
4	4	playground - clmb	tvd	c3d	amc	mpg	Animated	120	Feedback
5	5	playground - clmb, go under	tvd	c3d	amc	mpg	Animated	120	Feedback
6	6	playground - clmb, sit, dangle legs, descend	tvd	c3d	amc	mpg	Animated	120	Feedback
7	7	playground - clmb, sit, dangle legs, jump down	tvd	c3d	amc	mpg	Animated	120	Feedback
8	8	playground - clmb, sit, dangle legs, rock back, lower self to ground	tvd	c3d	amc	mpg	Animated	120	Feedback
9	9	playground - clmb, hang, hold self up with arms straight, swing, drop, sit, dangle legs, go under	tvd	c3d	amc	mpg	Animated	120	Feedback
10	10	playground - clmb, swing, lean back, drop	tvd	c3d	amc	mpg	Animated	120	Feedback
11	11	playground - clmb, hang, lean over, jump down	tvd	c3d	amc	mpg	Animated	120	Feedback
12	12	playground - clmb, pull up, dangle, sit, lower self to ground	tvd	c3d	amc	mpg	Animated	120	Feedback
13	13	playground - clmb, go under, jump down	tvd	c3d	amc	mpg	Animated	120	Feedback
14	14	playground - clmb, jump down, dangle, legs push off against	tvd	c3d	amc	mpg	Animated	120	Feedback

Subject #2 (various expressions and human behaviors) [file index](#) - - asf framerate Feedback

Image # Trial # Motion Description

Download [amc2mov.exe](#) dan [asf2mel.exe](#) di [web](#)

<http://mocap.cs.cmu.edu/asfamcmel/asfamcmel.php>

Address http://mocap.cs.cmu.edu/search.php?subjectnumber=22&trinum=1

CMU Graphics Lab Motion Capture Database

[Home](#) | [Search](#) | [Tools](#) | [Resources](#) | [Rendered Movies](#) | [FAQs](#) | [Suggest a Motion](#)

View All: [Subjects](#) | [Motions](#) | **Browse:** [Motion Categories](#)

Search Help

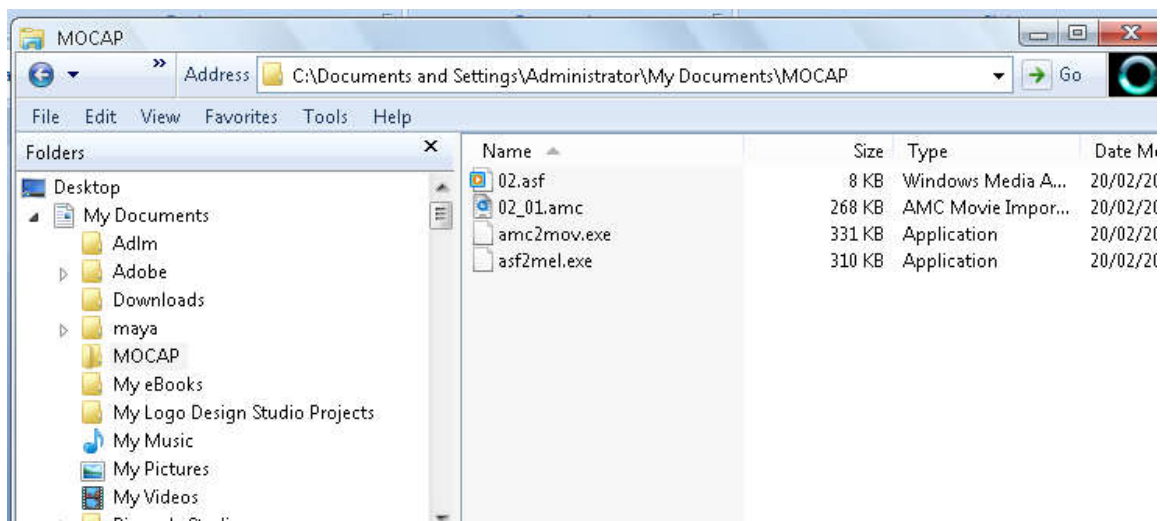
subject number | motion or keyword
(e.g. 1) (e.g. run)

motion calibration file #22
movement file #22-1 **watch the animation**

Subject #22 (human interaction (2 subjects - subject A)) [file index](#) - - asf framerate Feedback

Trial #	Motion Description	tvd	c3d	amc	mpg	Animated
1	B sits; A pulls up B (2 subjects - subject A)	tvd	c3d	amc	mpg	Animated

kelompokkan amc2mov.exe dan asf2mel.exe serta file asf dan amc yang sudah didownload kedalam satu folder.



BUAT FILE BAT DENGAN MENGGUNAKAN NOTEPAD

Buka notepad copy paste kan teks dibawah

```
asf2mel -f asffile.ASF import_skeleton-asffile.mel > asf2mel-out.txt
```

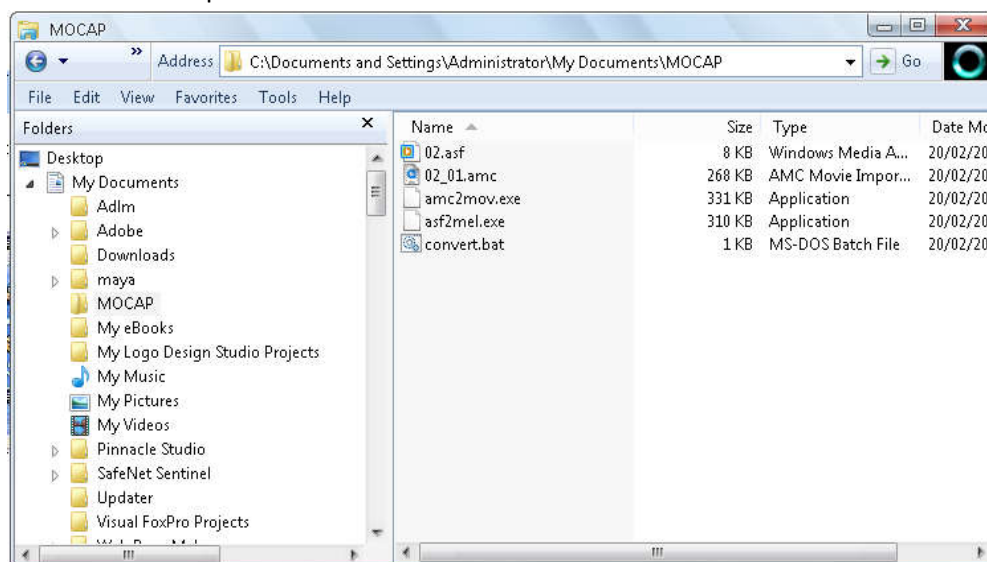
```
amc2mov -f asffile.ASF amcfile.AMC 60 import_mov-asffile.mel amcfile.AMC.MOV > asf2mov-out.txt
```

lalu edit disesuaikan dengan nama file asf file dan amc yang sudah didownload , contoh editannya karena file asf yg didownload namanya 02.asf dan amc yg didownload namanya 02_01.amc maka

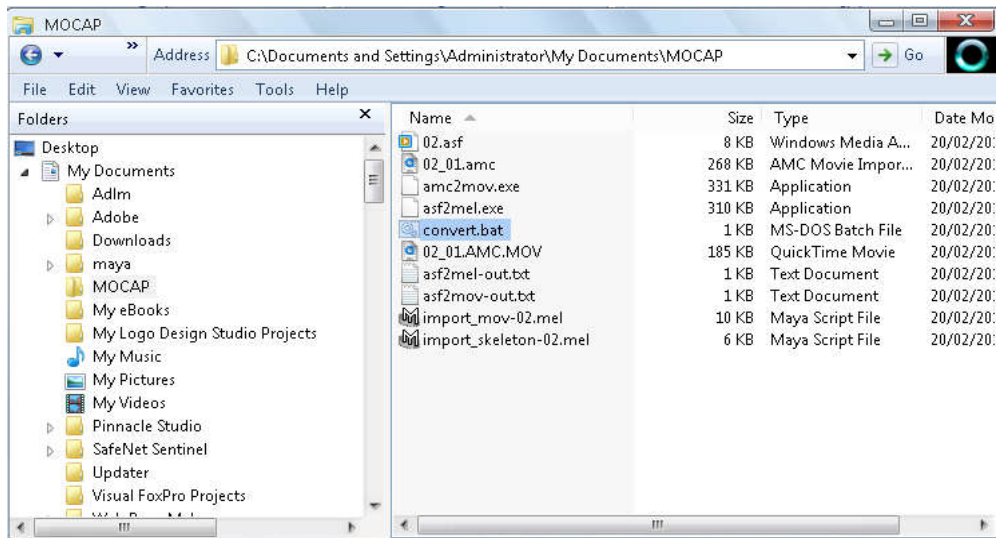
```
asf2mel -f 02.ASF import_skeleton-02.mel > asf2mel-out.txt amc2mov -f 02.ASF 02_01.amc 60 import_mov-02.mel 02_01.AMC.MOV > asf2mov-out.txt
```

simpan dengan nama convert.bat

lalu double click pada convert.bat

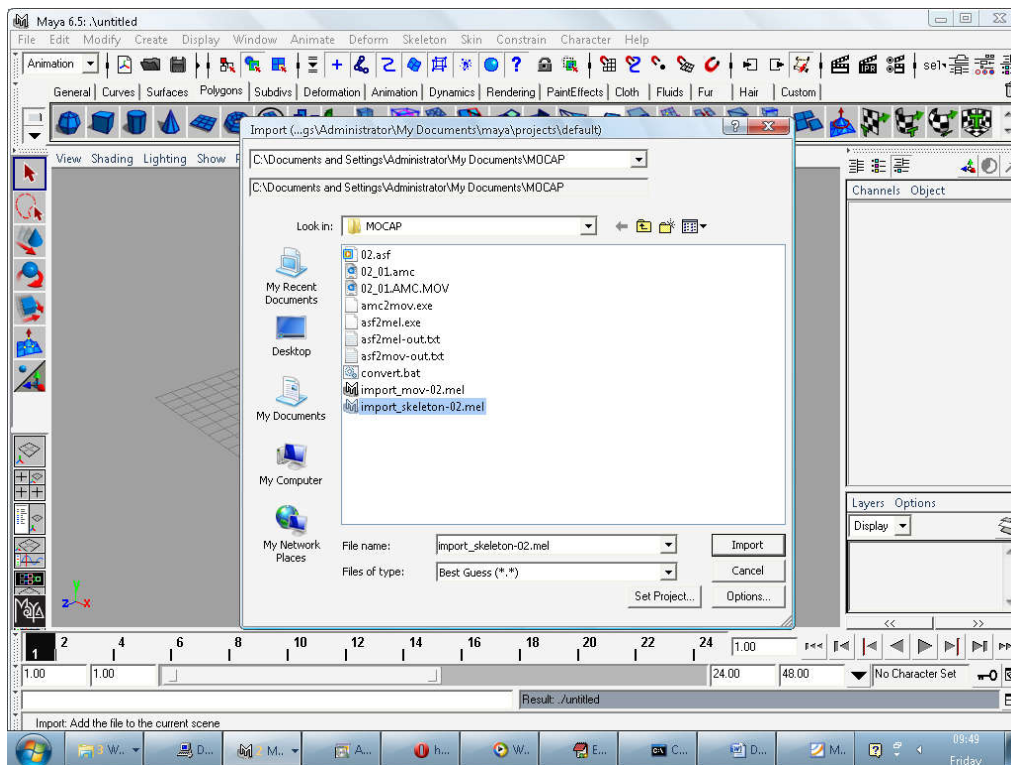


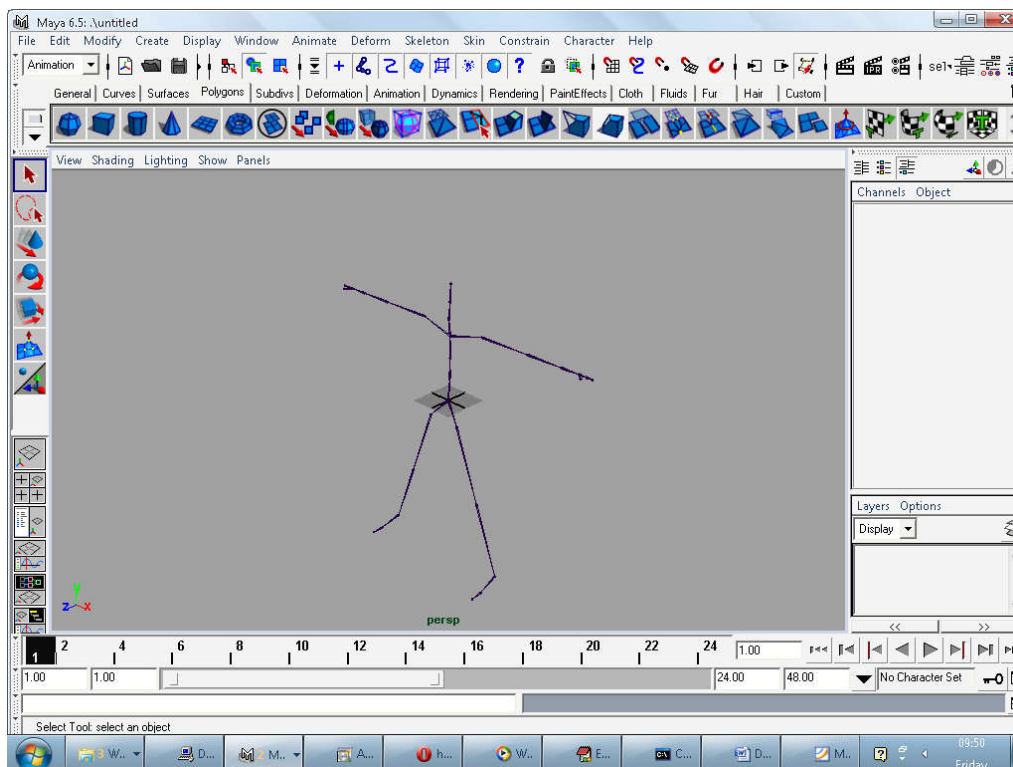
Setelah didouble klik maka akan ada tambahan file sbb



File ini akan berguna dalam membuat animasi karakter di maya

Buka maya klik file import dan import file” import_skeleton-02.mel”

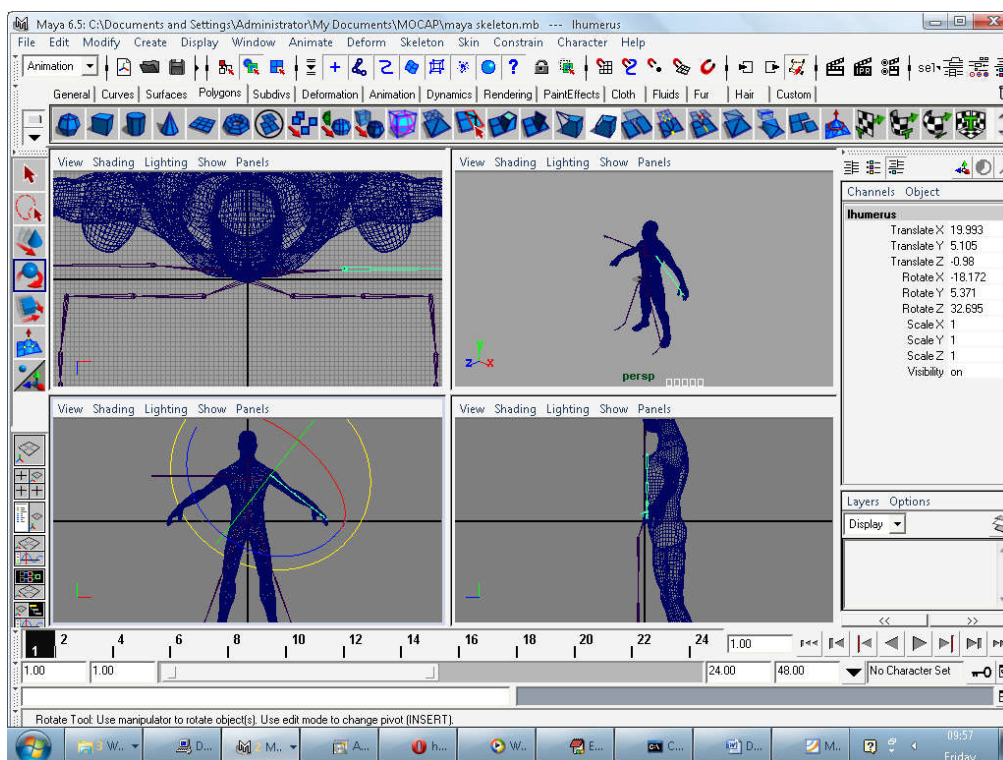




Anda akan langsung mendapatkan skeleton yang siap pakai untuk dianimasikan

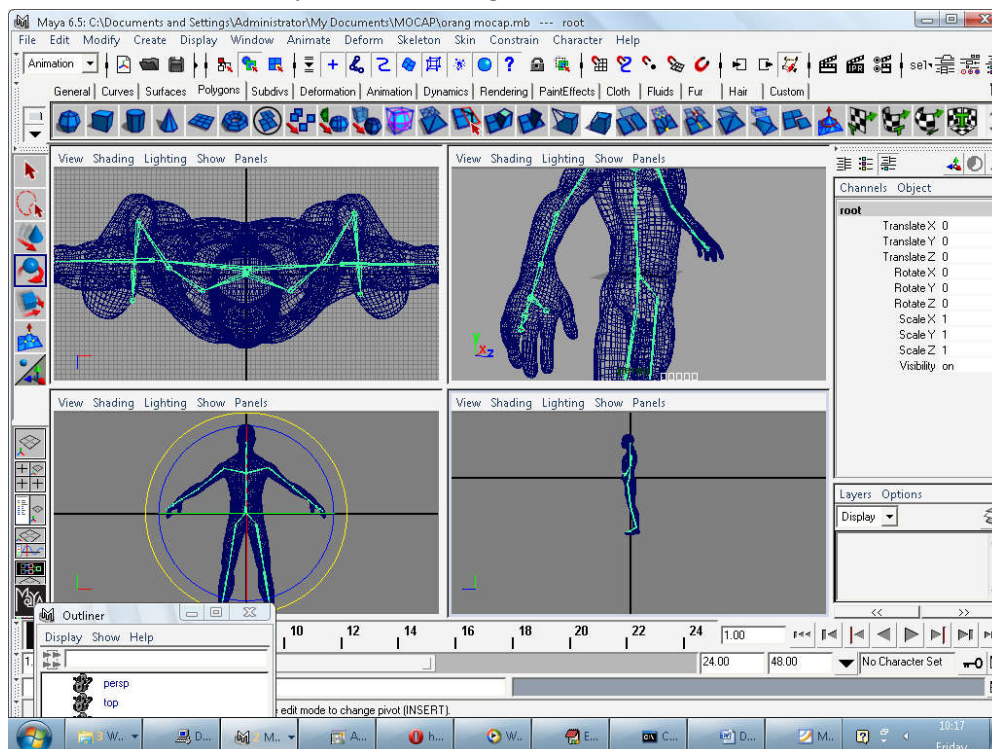
9.2 Memposing skeliton ke model karakter

Pertama tama atur posisi skeliton sesuai dengan karakter



Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

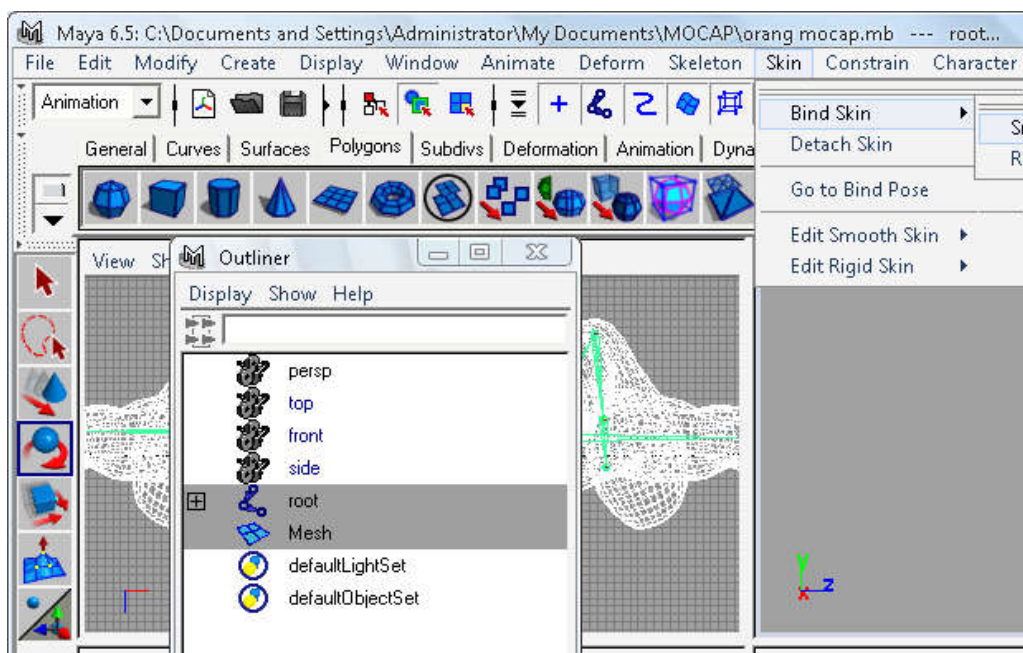
Sesuaikan ukuran model dengan tinggi skeleton. **skeleton tidak boleh diresize**. Lakukan rotasi pada skeleton untuk menyesuaikan dengan model. Skeleton harus berada didalam model



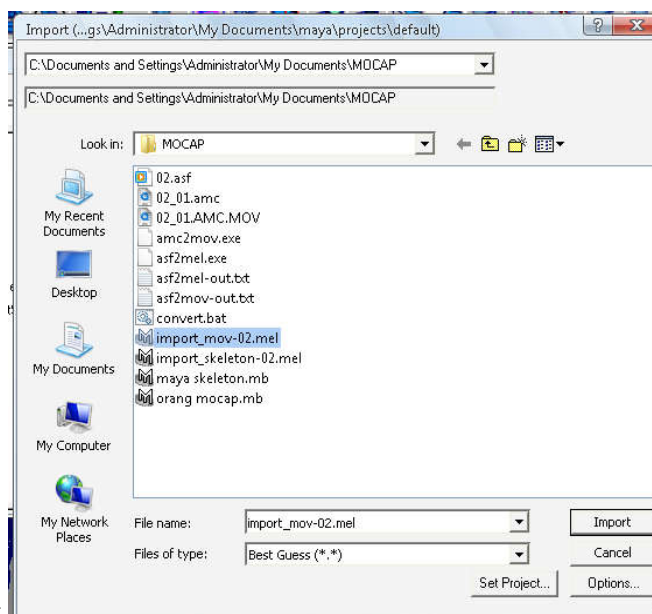
pilih model dan skeleton lalu pilih skin > bind skin > smooth bind

pastikan model karakter sudah menempel di tulang dengan cara melakukan rotasi pada tulang , jika model bergerak mengikuti gerakan tulang berarti proses skinning berhasil

9.4 Mengimport Animasi Motion Capture

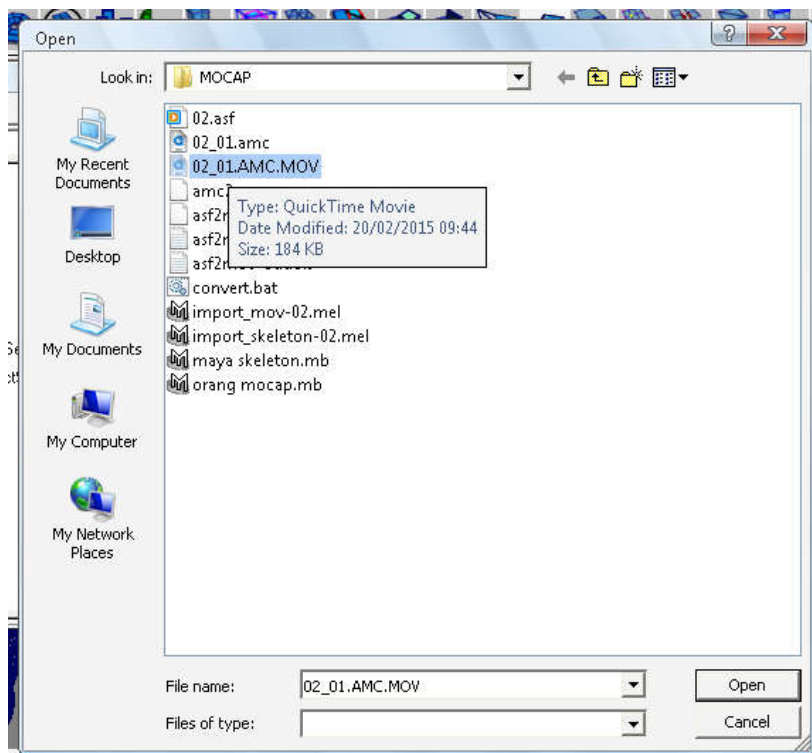


setelah model terkoneksi dengan skeletonnya import "import_mov-02.mel"

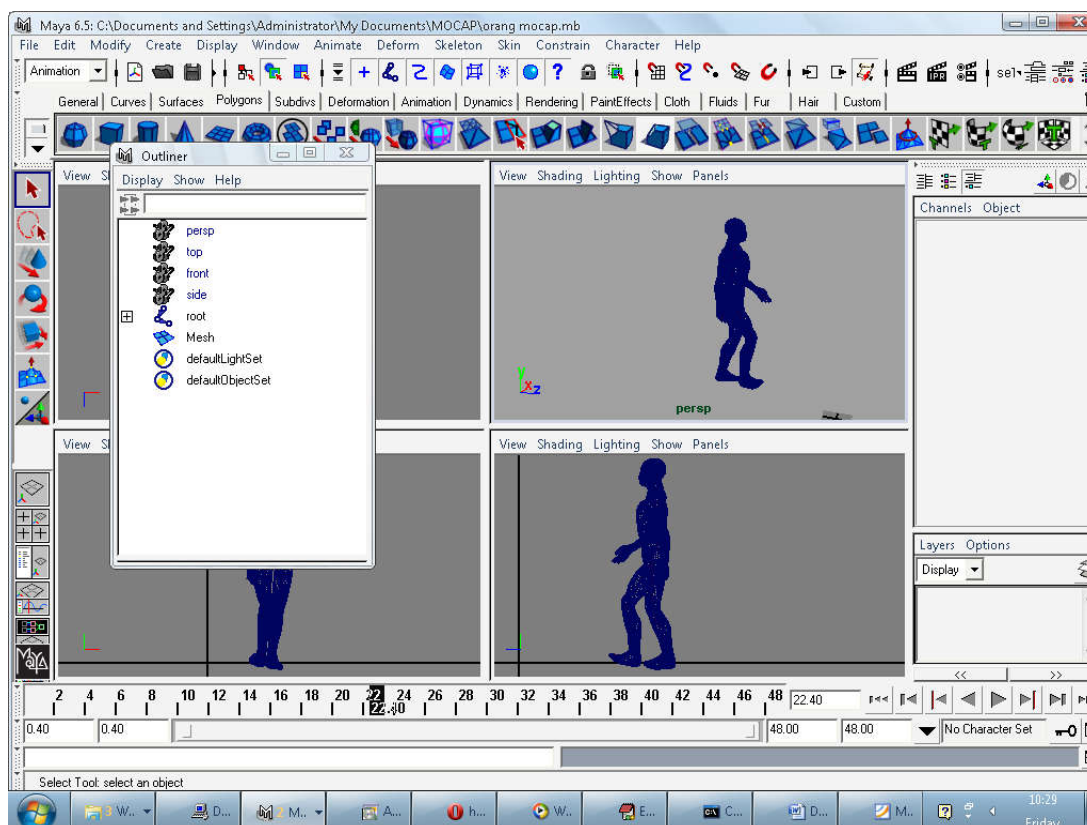


setelah import akan terbuka dialog

Akan terbuka dialog open dengan sendirinya , buka file "02_01.AMC.MOV"



Model anda akan bergerak sesuai dengan motion capture yang di pilih



Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

Untuk **preview** gerakan animasi

1. Klik play pada Time Slider playback controls untuk memutar animasi.
2. Tekan stop pada the Time Slider's playback control.

9.5 Membuat Clips Dari Motion Capture Data

Ketika membuat klip dari motion capture, kita dapat memulai dengan membuat klip sumber dari seluruh urutan gerakan. Anda dapat membuat klip dan menyimpannya ke Visor atau Outliner langsung tanpa memuat itu ke track. Anda kemudian dapat kembali ke klip sumber dan memanfaatkannya lagi jika diperlukan.


1. pada Time Slider, set Playback Start Time ke 1.



2. tekan Go to Start untuk menuju ke frame 1 pada Time Slider.



skeleton pada frame 1.

3. Pada persp view, pilih skeleton
4. Pada Trax Editor menu, pilih Create > Clip > .
5. Pada Create Clip Options window, click Edit > Reset Settings untuk memilih ke default settings, dan atur sebagai berikut:
 - o Ketik WalkSit in the Name field.
 - o Keys - Leave Keys in Timeline: On
 - o Clip in Visor Only: On
 - o Click Create Clip.

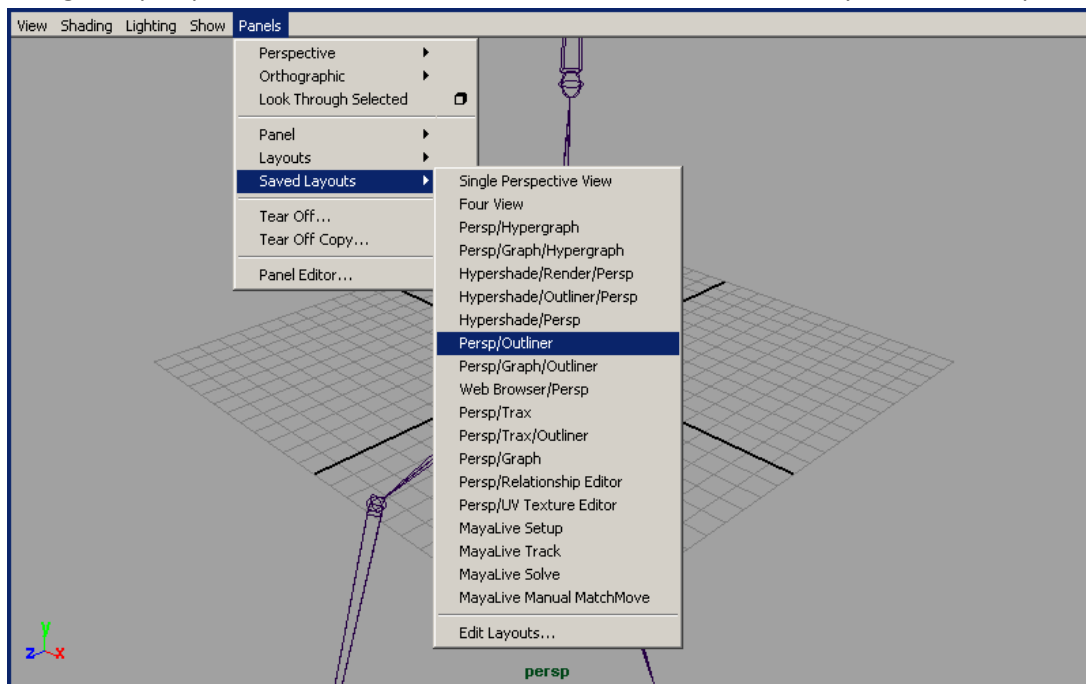


Source klip pada outliner akan bernama WalkSitSource.

9.6 Mengexport Clip Motion Capture Dan Digunakan Lagi Bersama Sama

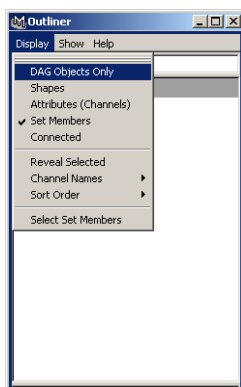
Import bone dan animasi dari motion capture sehingga terdapat animasi.

Aktifkan Trax editor. Untuk mengatur Setting layar yang pas untuk trax editor adalah dengan mengklik perspective view's menu select Panels > Saved Layouts > Persp/Trax/Outliner.

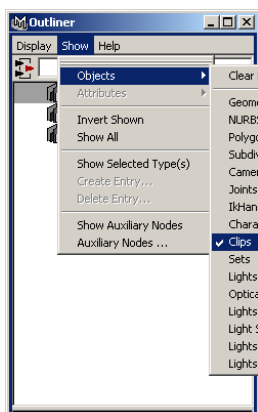


Menampilkan clip pada outliner

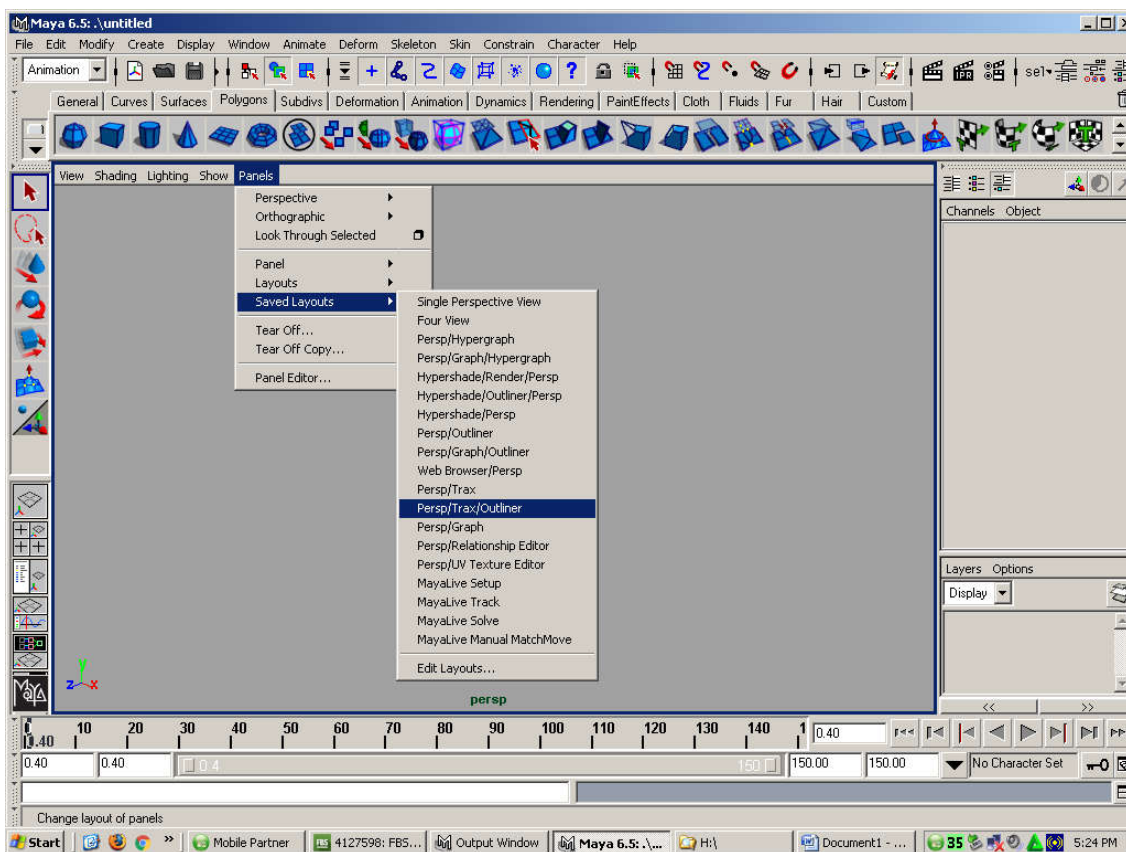
1. pada Outliner menu, uncheck Display > DAG Objects Only.

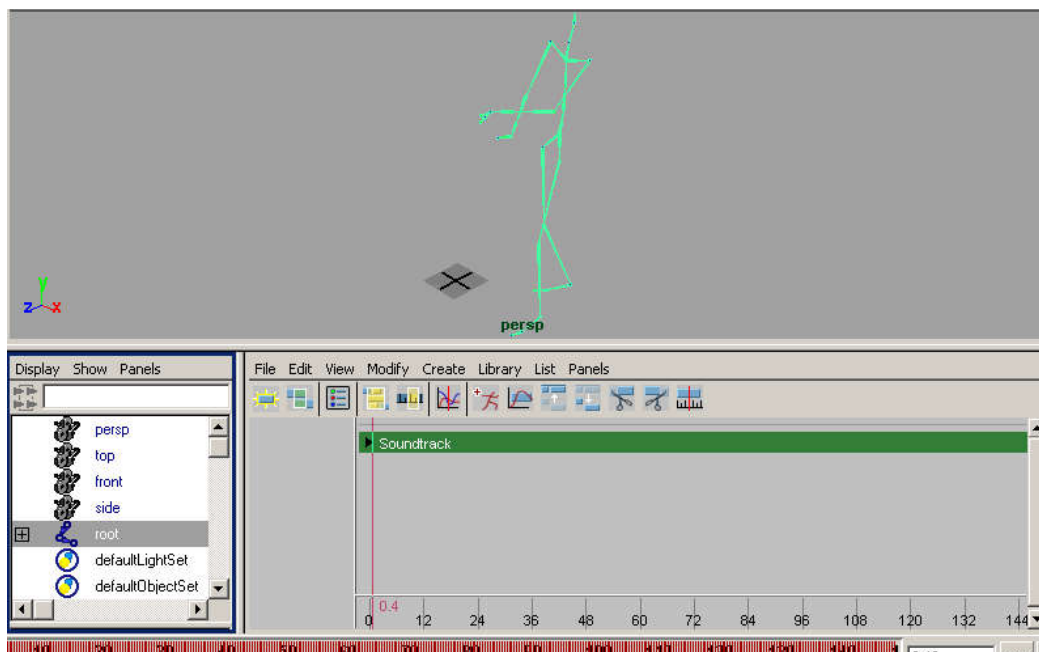


2. pada Outliner menu, pilih Show > Objects > Clips.



Setelah tulang dan animasi telah terimport dengan baik

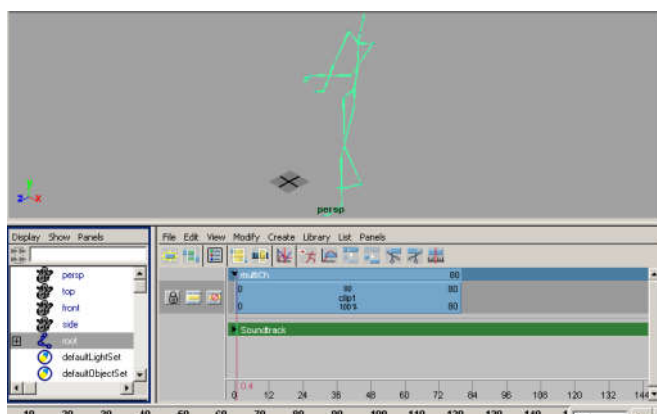




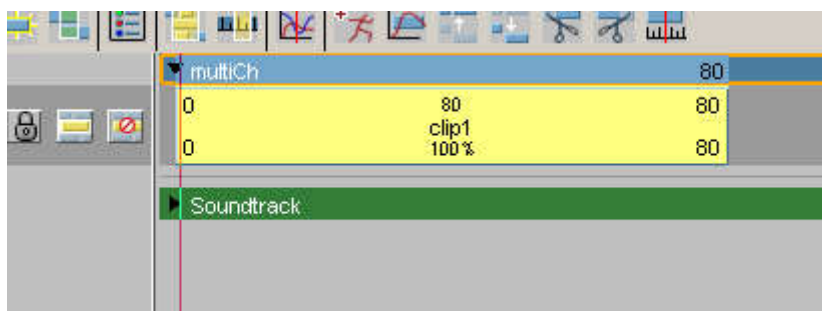
Pada outliner Klik root



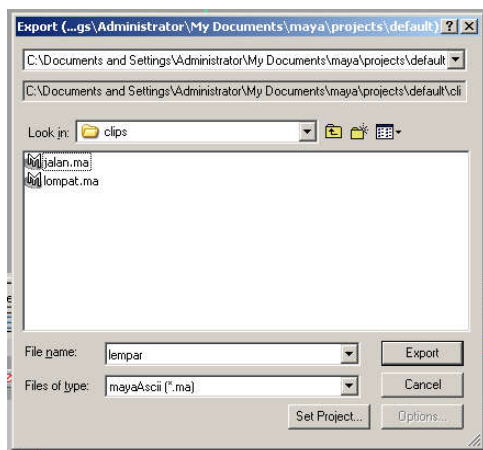
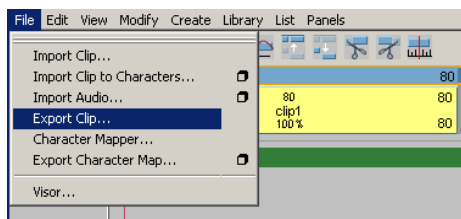
Tekan createclip



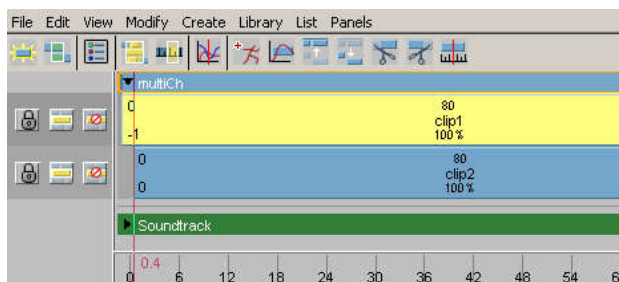
Klip akan muncul



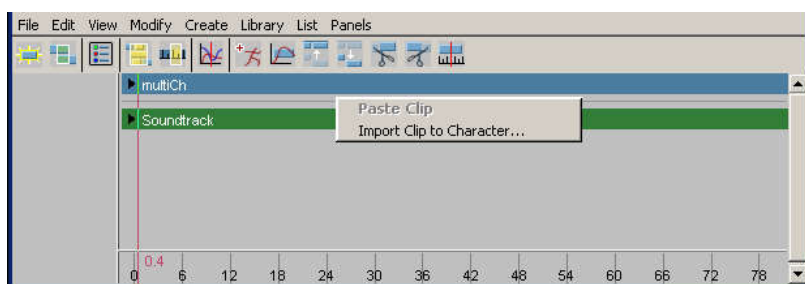
Clip yang sudah ada tinggal kita export dengan menekan file> export clip



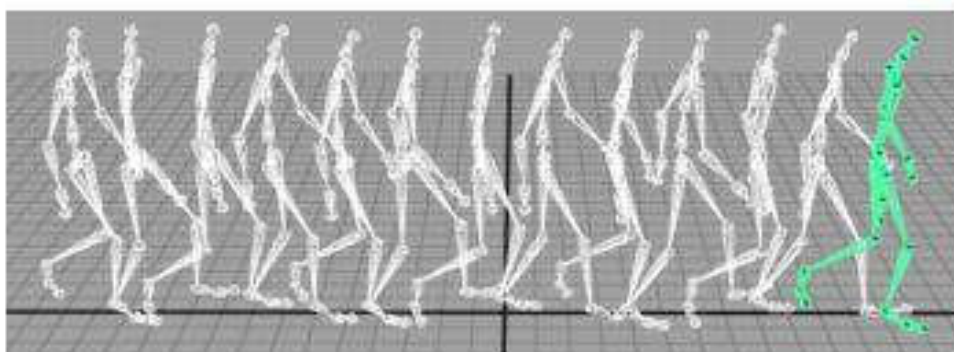
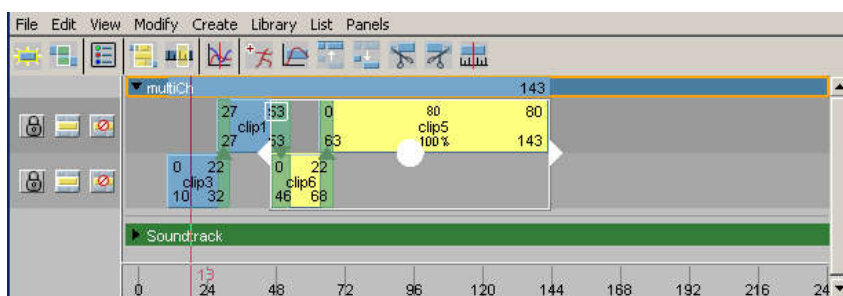
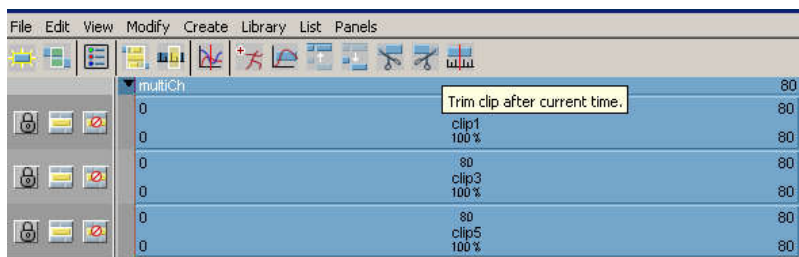
Duplicate clip



Delete clip dan klik kanan pada multich import clip



Ulangi beberapa kali untuk mengimport klip



9.7 Mengatur Arah Gerakan Pada Clip

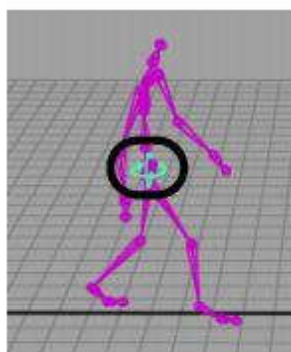
Motion capture data dapat digunakan dalam berbagai cara. Anda dapat mengarahkan gerakan karakter menggunakan data motion capture yang sama. *Redirect tool* memungkinkan Anda untuk mengubah animasi asli untuk karakter dengan memodifikasi path atau orientasi setiap saat selama animasi. Anda dapat mengubah awal atau akhir titik animasi karakter ini dengan mengarahkan posisi awal karakter atau dengan mengalihkan gerak selama animasi. Anda dapat mengarahkan karakter dalam urutan animasi yang ada dalam adegan atau karakter yang terkait dengan klip dalam Trax Editor .

Berikut ini contoh mengarahkan bone untuk membelok ketika berjalan.

1. Pada Time Slider, geser current time indicator ke frame 36.
2. Pilih skeleton .
3. pada main menu, pilih [Character > Redirect > !\[\]\(13b6bdd0ca077c333d50231f1443cb1d_img.jpg\)](#).
4. pada Redirect Options Window atur Redirection Type ke Rotation only dan click Redirect button.



Sebuah Rotation redirection control muncul diantara pinggul skeleton, yang merupakan root joint.

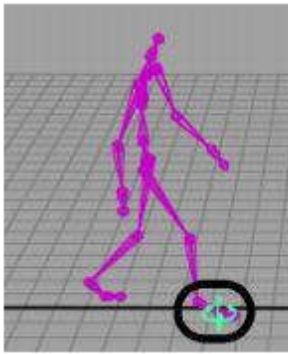


Memindahkan redirection control

Pada Toolbox, pilih Move tool.

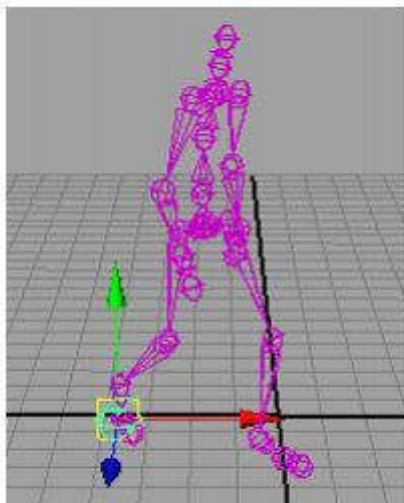
The Move tool manipulator muncul.

2. Tekan tombol v dan click-drag redirection control pada lokasi tulang yang diinginkan.

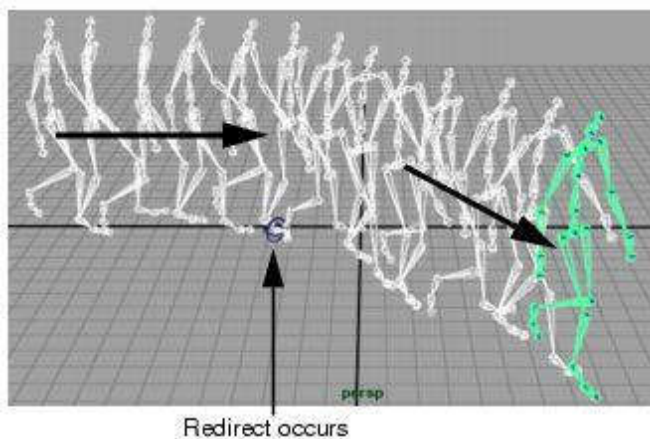


Membuat keyframe pada redirection control

1. Dalam kondisi redirection control terpilih, tekan s untuk menset keyframe.
 Anan tercipta garis merah pada frame 36 menunjukkan terbentuknya keyframe.
 Pada Time Slider, drag ke frame 57.
3. Pada Channel Box, dengan redirection control masih terpilih, atur Rotate Y attribute ke -45.



5. Dengan redirection control masih terpilih, tekan s untuk membuat keyframe.
 Hasilnya adalah tulang akan berbelok saat sampai pada redirection control keyframe, dan terus berjalan pada arah yang baru dan baru berhenti saat animasi berakhir.



Memasukkan klip sumber

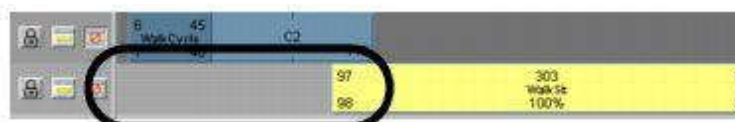
1. Pada Trax Editor, pilih Library > Insert Clip. Pilih WalkSitSource pada drop-down list. Duplikasi clip akan dibuat, dan dimasukkan pada new track.
2. Pada Trax Editor, pilih View > Frame All.
3. Pada Trax control area, klik Mute button untuk track yang berisi WalkCycle clip.



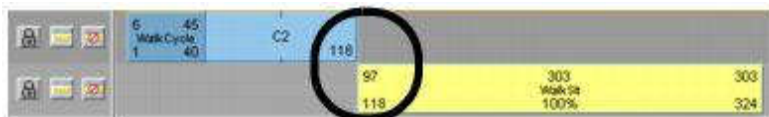
4. Geser current time indicator ke frame 98.
6. Click pada the WalkSit clip di track .
7. Pada Trax tool bar, klik Trim Clip Before Current Time icon untuk memotong klip sebelum frame 98.



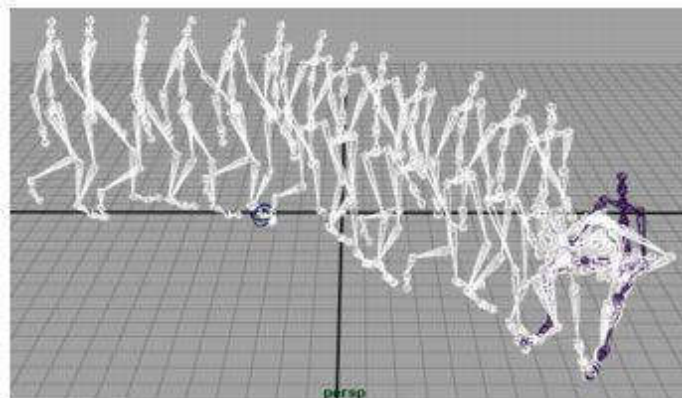
The WalkSit clip akan terpotong dan bagian sebelum frame 98 akan dihapus.



7. Klik Mute button pada track berisi WalkCycle clip untuk mengaktifkan kembali track.
8. Sesuaikan WalkSit clip agar Frame sesuai.



10. Klik play pada Time Slider playback controls untuk play back animasinya.

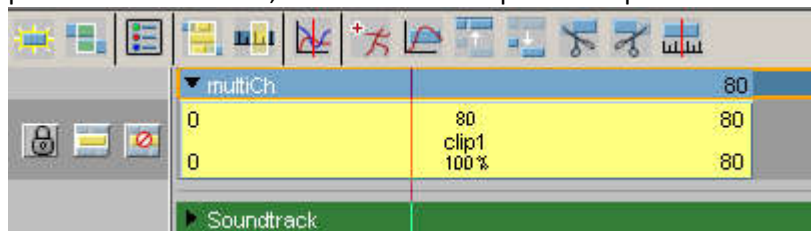


11. tekan stop pada the playback controls untuk menstop playback

sebuah loncatan akan terjadi diantara dua klip loncatan itu terjadi Karena pose skeleton tidak cocok dengan ahit klip pertama dan awal klip kedua , untuk memperbaiki loncatan ini maka kita bisa memperpanjang clip.

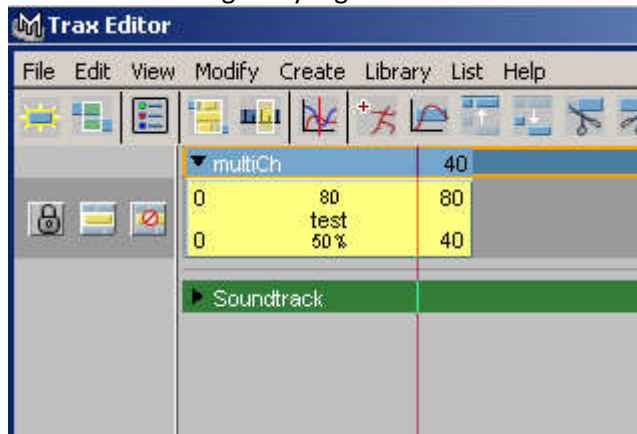
9.8 Memperpanjang durasi clip

1. pada Trax Editor, double-click pada clip in the pada bagian tengah



3. jika anda double click pada tulisan clip1 maka anda akan merename clip tersebut.
4. Jika anda medouble click angka100% maka anda akan memperpang/ emperpendek durasi klip. Jika anda isikan 200 maka animasi akan melambat tetapi clip juga dua kali lebih lama jika

anda isikan angka yang lebih kecil maka animasi juga akan semakin cepat



5. Press stop on the Time Slider's playback control.

9.9 Mengedit Sebagian Gerakan Dari Motion Capture.

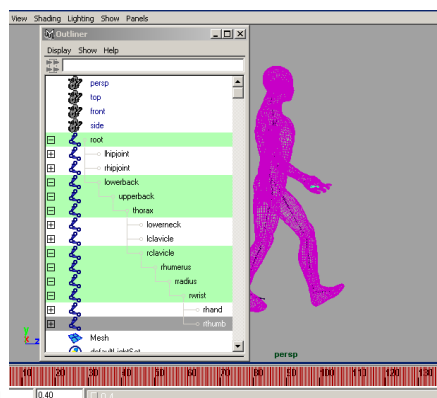
Saat kita menggunakan motion capture mungkin kita perlu membuang sebagian animasi dari motion capture tersebut, dan memanfaatkan sebagian lagi. Untuk itu perlu dilakukan pengeditan dari motion capture.

Untuk melakukan pengeditan kita harus sudah memiliki karakter yang sudah di animasikan dengan motion capture dahulu. Dalam contoh gambar obyek sudah memiliki animasi jalan dengan tangan melambai natural. Karena ingin merubah gerakan tangan tersebut menjadi kedepan maka kita harus menghilangkan keyframe dari gerakan tangan tersebut



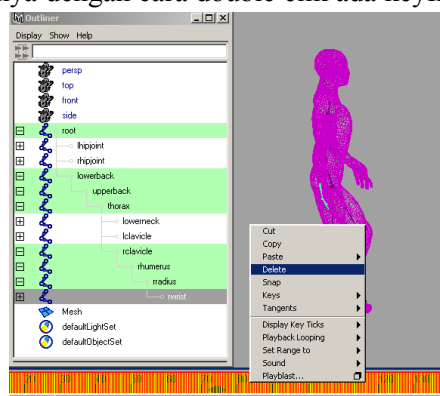
Menghilangkannya dengan menggunakan outliner dan dipilih tulang yang ingin dihilangkan animasinya. Dimulai dari bagian bagian paling ujung terlebih dahulu sehingga jika ingin menghilangkan animasi tangan maka kita mulai dari tulang jari.

Dari gambar diatas bisa kita lihat saat kita memilih tulang jari kanan terdapat banyak keyframe, ini



harus kita hilangkan

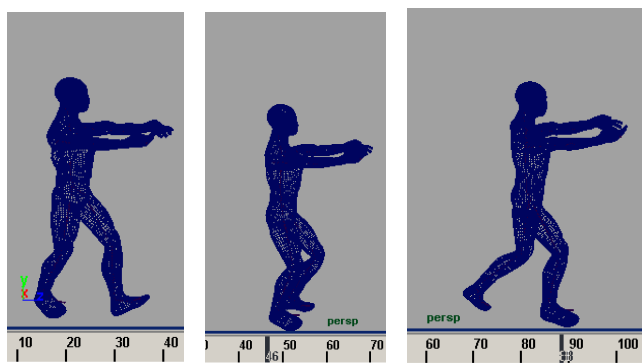
Untuk menghilangkannya dengan cara double klik ada keyframe sehingga semua berwarna merah lalu



klik kanan pilih delete.

Ulangi semua untuk tulang tangan.

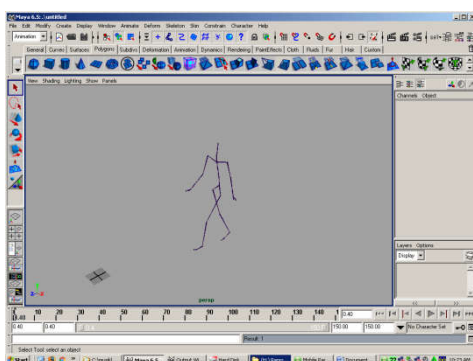
Setelah semua keyframe ada tangan dihapus maka didapatkan animasi jalan dengan tangan tidak bergerak. Kita gerakkan tangan sesuai dengan keinginan kita menggunakan animasi keyframe.



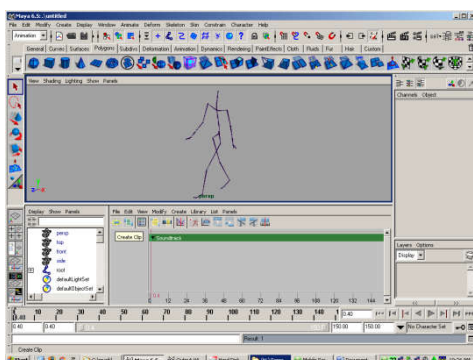
9.10 Menggabungkan Beberapa Motion Capture

Pada materi sebelumnya lebih banyak membahas penggunaan motion capture untuk satu obyek kerangka. Pada kali ini akan dibahas membuat animasi motion capture dengan beberapa kerangka yang dapat bergerak sesuai dengan clip yang sudah di buat dan di save sebelumnya.

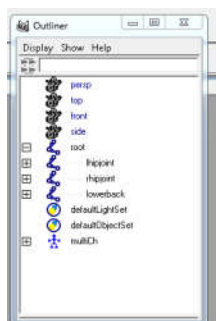
Untuk memulainya maka dibutuhkan adanya kerangka yang akan diduplikasi menjadi beberapa kerangka untuk itu dilakukan tahap Import dan memberi animasi motion capture melalui import animasi mocap yang sudah dibahas sebelumnya.



Lakukan create clip

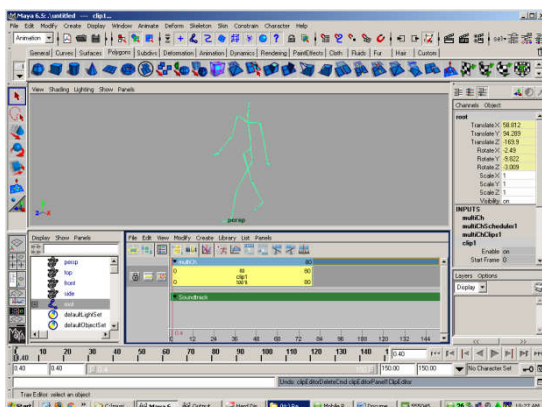


dari proses create clip ini akan diperoleh character map yang bisa dilihat di outliner



character map ini yang kita butuhkan

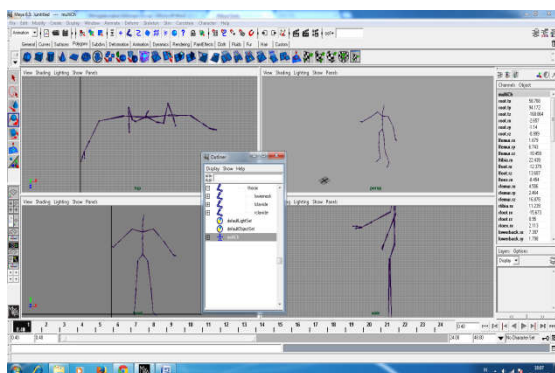
export clip agar animasi yang ada tidak terbuang percuma dan bisa dipakai lagi.



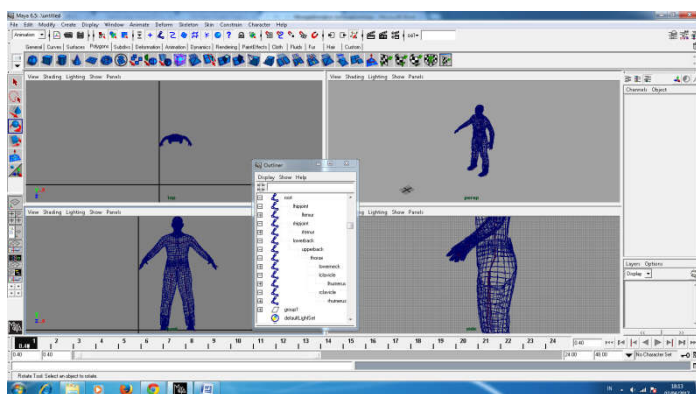
delete clip

Sehingga kita mendapatkan obyek plus character map tanpa animasi

Obyek kerangka yang sudah memiliki character map sesuai dengan motion capture ini kita tata kembali agar seperti bentuk awal sebelum diberi animasi



simpan sebagai master.mb

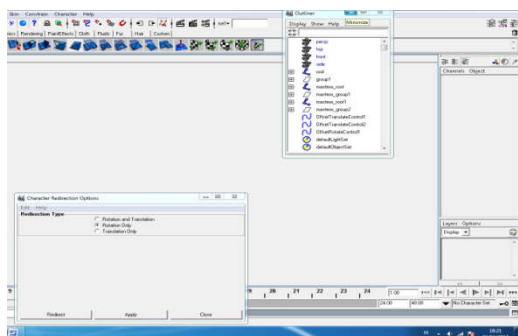


Buka scene baru

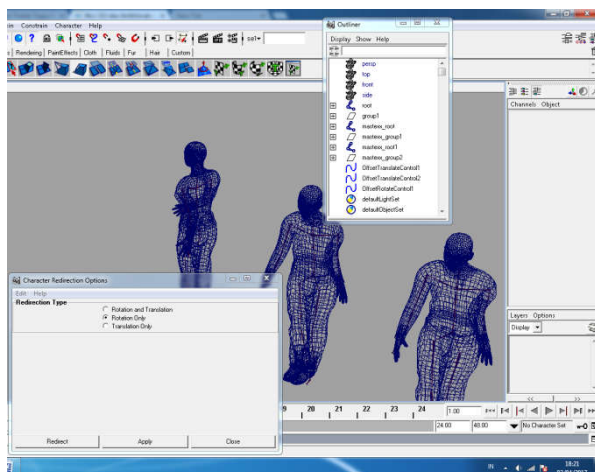
Import master .mb .tujuan import untuk mendapatkan obyek dari master .mb. kita bisa mengimpor dari satu file beberapa kali.

Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

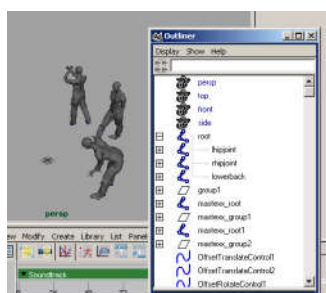
Jika anda melakukan import master beberapa kali, meskipun obyeknya sama tetapi saat dilihat di outliner akan terlihat beberapa, dalam contoh ini dilakukan import 3x sehingga ada tiga root joint dan 3 karakter set berbeda



Geser ketiga obyek tersebut dengan redirect



Import clip yang berbeda untuk ketiga obyek tersebut



Dengan clip yang berbeda beda anda akan dapat membuat animasi orang banyak dengan hanya menggunakan satu kerangka saja.

BAB 10 MENGGUNAKAN MOTION CAPTURE DENGAN MAKEHUMAN DAN BLENDER

10.1. open source software Blender

Blender adalah paket perangkat lunak pemodelan dan animasi 3D yang kuat yang tersedia untuk Windows, Macintosh, dan Linux. Seperti paket serupa lainnya seperti Lightwave, 3D Studio Max, dan Maya, Blender menawarkan berbagai alat pemodelan, animasi, dan rendering. Ia juga memiliki sejumlah fitur pembedanya sendiri, termasuk portabilitas lintas platform yang sangat baik, kemampuan untuk menjalankan skrip dalam bahasa pemrograman Python, antarmuka pengguna yang unik dan intuitif untuk alur kerja yang efisien, dan fleksibilitas luar biasa dalam mengimpor dan mengekspor file, adegan, dan objek untuk digunakan dengan program lain, termasuk beragam pelacak sinar berkualitas tinggi. Ini memiliki simulator fisika canggih, dan yang baru, secepat kilat penerapan UV unwrapping. Ini fitur saja sudah cukup membuat Blender menjadi pesaing kuat di dunia alat 3D,

tetapi fitur pembeda tunggal terbesarnya adalah gratis. Itu bukan hanya untuk mengatakan bahwa seseorang memberikan sampel gratis yang dapat Anda gunakan beberapa kapasitas tanpa membayar. Blender dilisensikan di bawah Lisensi Publik GNU, lisensi terdepan untuk perangkat lunak sumber terbuka. Ini berarti bahwa kode sumber Blender bebas tersedia bagi siapa saja untuk mengunduh, menggunakan, menyalin, mengubah, dan mendistribusikan untuk tujuan apa pun, asalkan mereka mematuhi pedoman yang ditetapkan dalam GPL. Pedoman ini mensyaratkan perubahan itu dibuat untuk kode diidentifikasi secara eksplisit dan kode yang dirilis yang dihasilkan tetap terbuka dan tersedia secara bebas. Singkatnya, Blender benar-benar gratis, dan lisensi dirancang untuk membuatnya yakin itu tetap gratis untuk selamanya

10.2 Menggunakan Motion Capture pada obyek Makehuman Secara Langsung

berikut ini cara menggunakan motion capture bvh kedalam obyek yang dibuat dengan makehuman

langkah pertama persiapkan obyek dengan memanfaatkan makehuman, obyek yang dibuat dengan menggunakan pilihan pose Tpose pose berbentuk T yang sangat cocok dengan obek bvh.



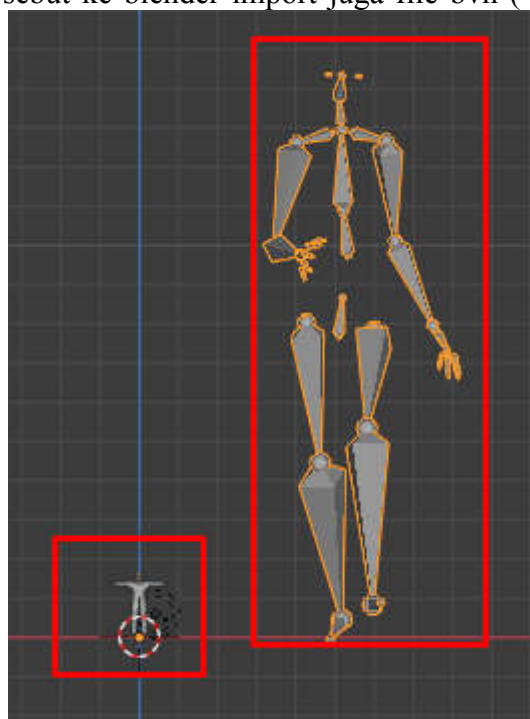
basic dari makehuman

kita atur pada posisi pose/animate lalu pilih pose lalu pilih Tpose



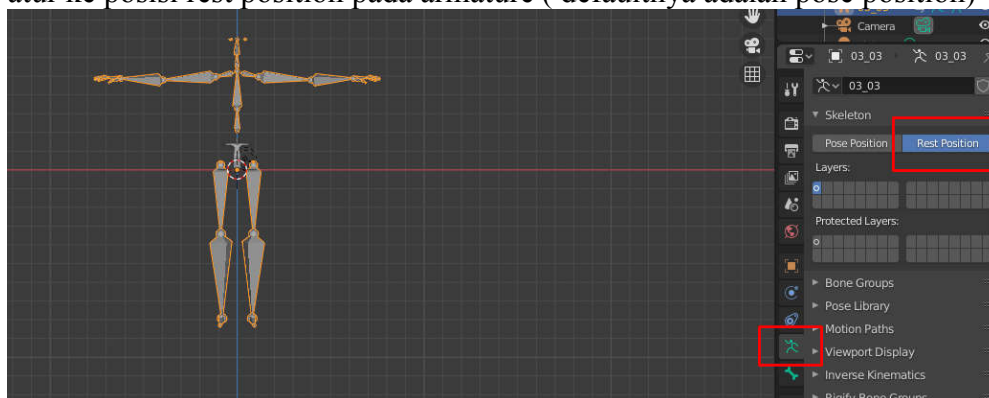
obyek yang kita pilih ini akan memiliki tangan terentang dan kaki rapat. setelah itu export ke MHX2, obyek ini tidak perlu disisipkan tulang dari makehuman.

import file mhx2 tersebut ke blender import juga file bvh (motion capture) hasilnya akan



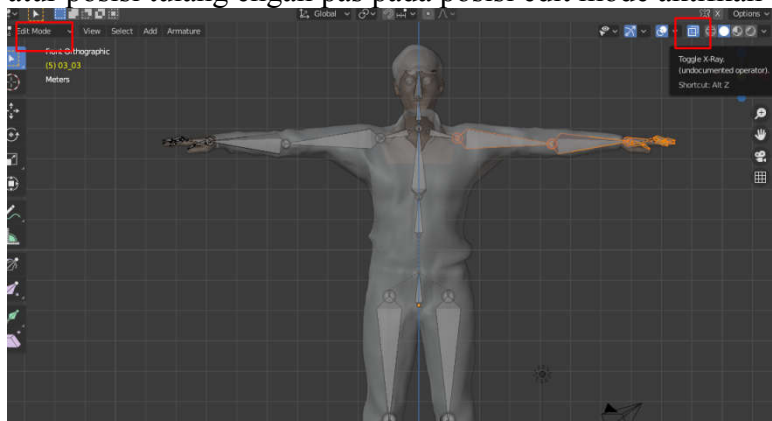
bvhnya terlalu besar.

atur ke posisi rest position pada armature (defaultnya adalah pose position)

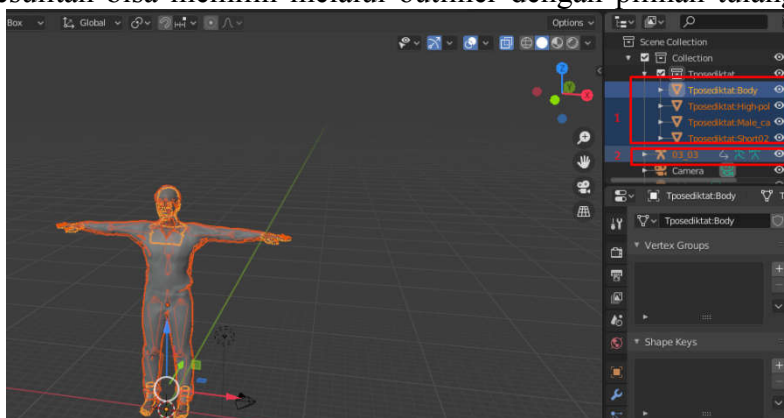


agar ukurannya sama dan lakukan langkah seperti rigging yaitu posisikan tulang berada ditengah obyek

atur posisi tulang engan pas pada posisi edit mode aktifkan toggle xray agar kelihatan semua

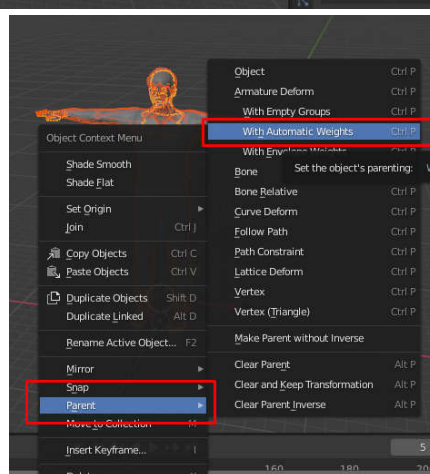


setelah selesai kita lakukan parenting dengan seleksi seluruh obyek manusianya , tekan shift lalu pilih tulang , jika kesulitan bisa memilih melalui outliner dengan pilihan tulang pada



pilihan yang terakhir.

klik



kanan pilih parent dan pilih with automatic weight.

dari sini bisa kita lihat bahwa langkahnya sama dengan pelajaran sebelumnya yaitu saat menempelkan tulang yang dibuat dengan rigify atau tulang yang dibuat dengan manual dengan daging manusianya, pada dasarnya motion capture format bvh adalah tulang armature yang sudah memiliki animasi.

sekarang pilih tulang nya pindah ke pose mode lalu pada bagian armature pilih ke pose position anda lihat daging sudah mengikuti animasi tulang , lalu tekan play untuk melihat



animasinya.

10.3 Memanfaatkan Motion Capture Obyek Makehuman Pada Blender Dengan Addons

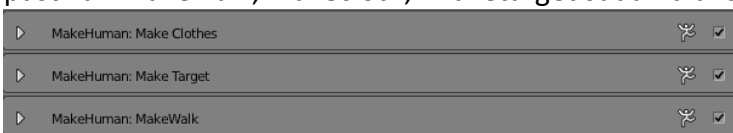
memanfaatkan fitur bantuan yang sudah tersedia sangat membantu kita dalam membuat animasi dengan cepat. dengan memanfaatkan makehuman dan addons pada blender sangat mempermudah pengerjaan dengan menghilangkan langkah rigging manual. karena blender mengalami perubahan yang sangat mendasar dari versi 2.79 kebawah dengan versi 2.8 keatas maka perlakuan dalam penggunaan makehuman pada blender juga sangat berbeda berikut akan dibagas keduanya.

10.3.1 untuk versi 2.79 kebawah

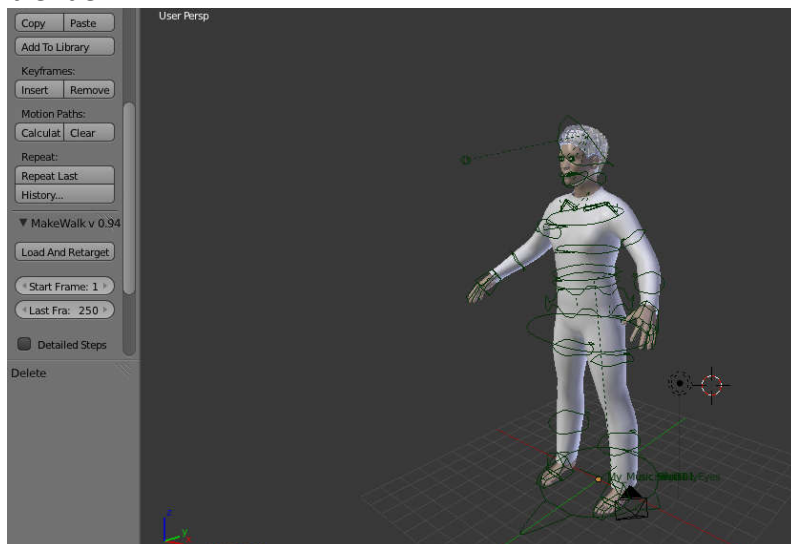
jika kita menggunakan blender 2.79 kebawah, dengan makehuman akan lebih mudah dengan memanfaatkan makewalk, sayangnya fitur ini tidak jalan pada versi 2.8 keatas

langkahnya:

Kembali ke blender, pastikan blender anda sudah siap untuk membuka file bvh dengan memanfaatkan make human, yaitu dengan melakukan langkah pengaktifan makehuman addons , silahkan lihat pada bagian pemanfaatan makehuman pada bab sebelumnya, pastikan makewalk, makecloth, maketarget sudah diaktifkan



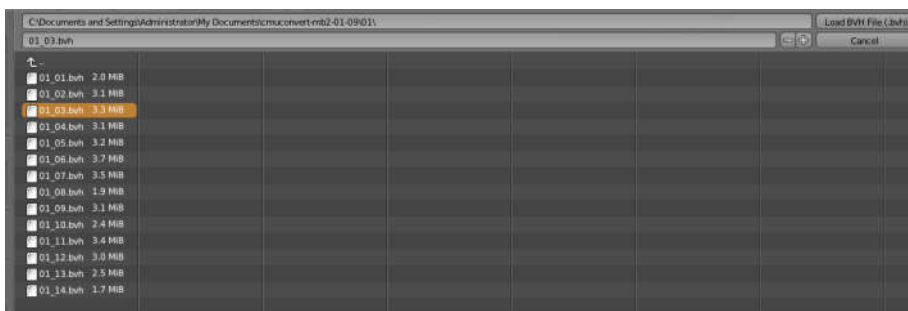
Untuk pemanfaatannya sangat mudah yaitu cukup mengimport file makehuman kedalam blender.



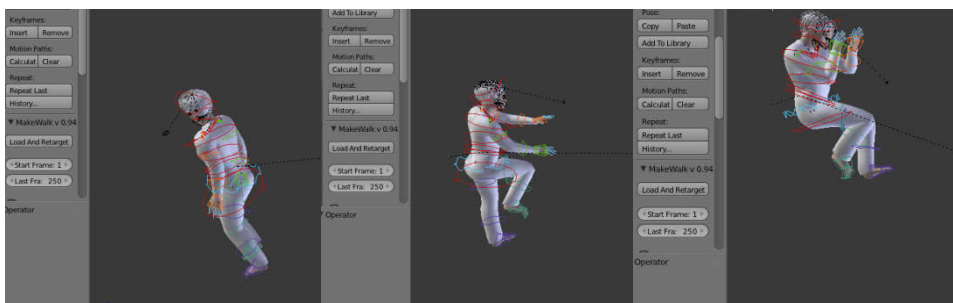
Perhatikan pada bagian samping terdapat fitur makewalk.

Klik pada load bvh files

Pilih bvh yang anda inginkan lalu load bvh file



Hasilnya animasi yang realistik sudah terbentuk dengan mudah dan cepat



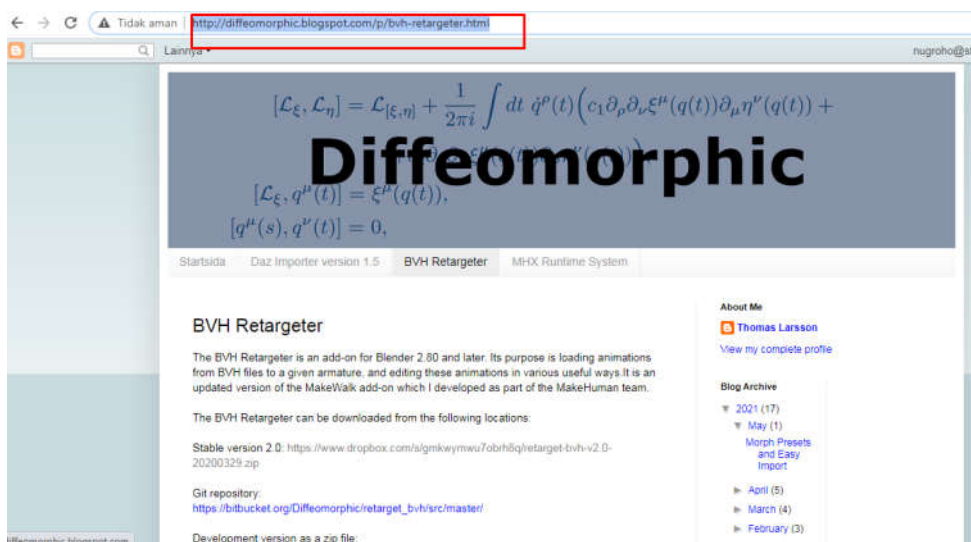
sangat mudah sekali jika menggunakan makewalk

10.3.2 pada blender 2.8 keatas

pada blender 2.8 keatas addons masih belum tersedia didalam programnya, jadi harus kita install dulu

langkahnya

kunjungi halaman <http://diffeomorphic.blogspot.com/p/bvh-retargeter.html>

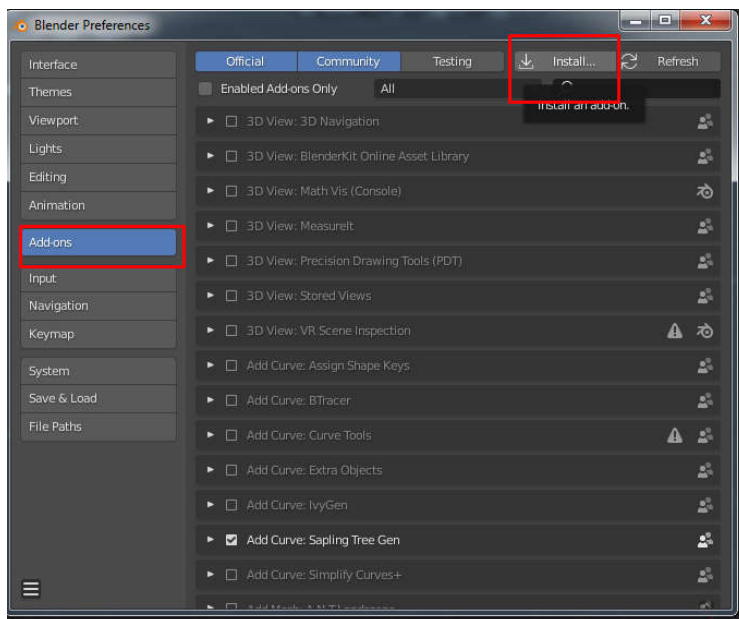


download pluginnya disini

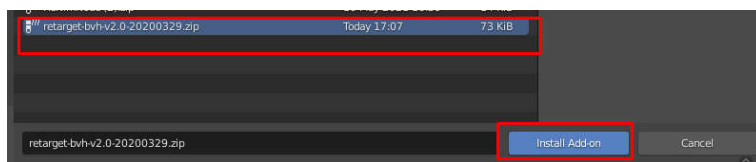
<https://www.dropbox.com/s/gmkwymwu7obr8q/retarget-bvh-v2.0-20200329.zip>

setelah didownload instal puginja dengan cara edit preference dan tekan addons dan klik install

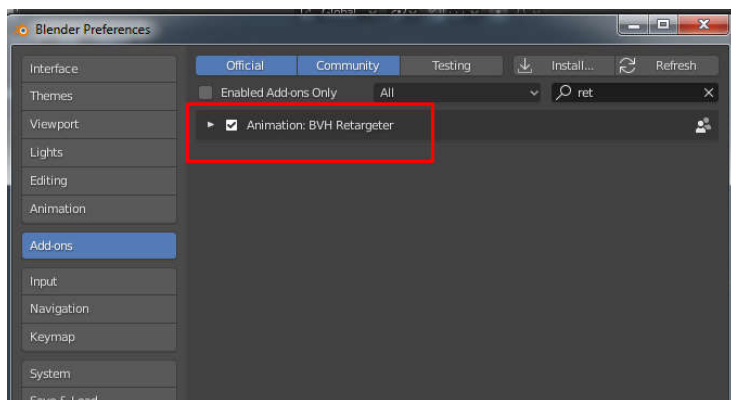
Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)



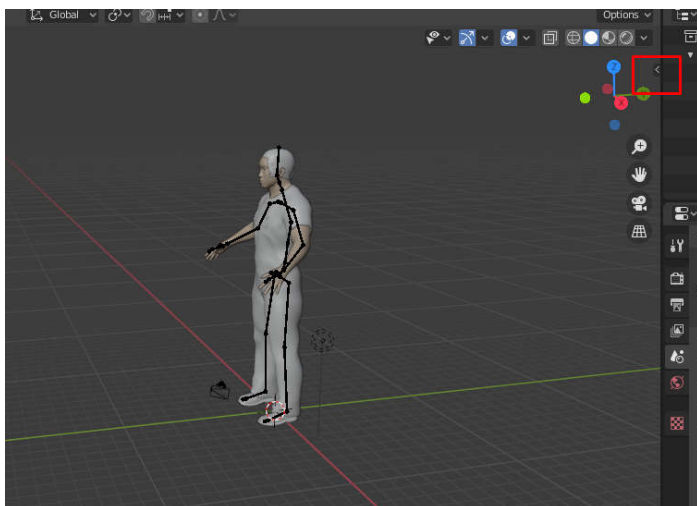
pilih hasil downloadnya dan tekan install



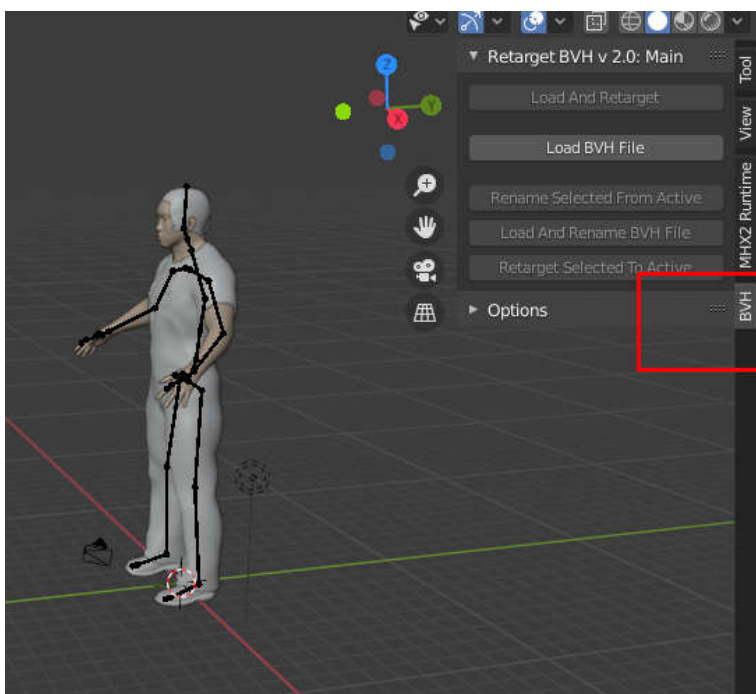
setelah itu aktifkan BVH retargeter



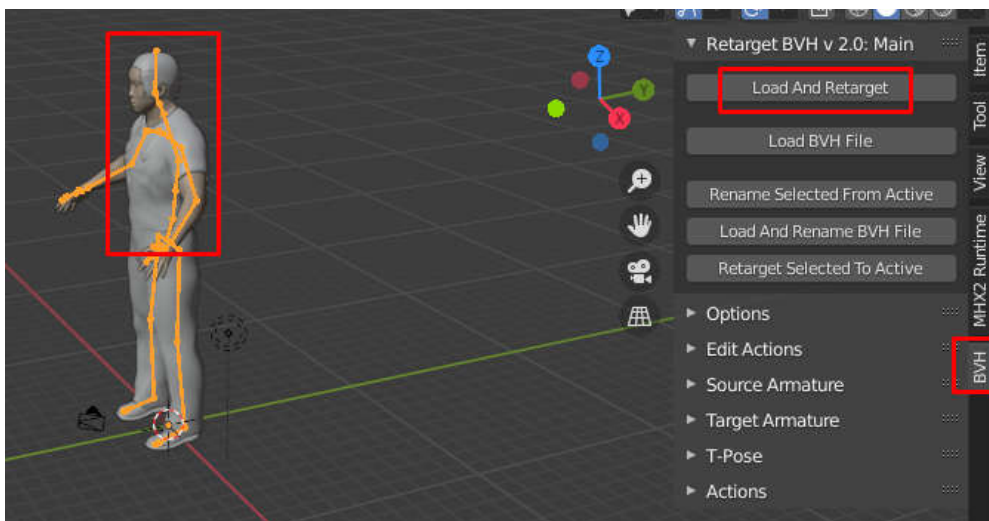
import obyek makehuman lalu klik < di pojok



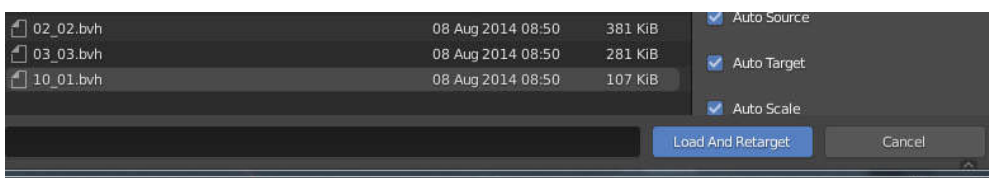
pilih pada tab BVH



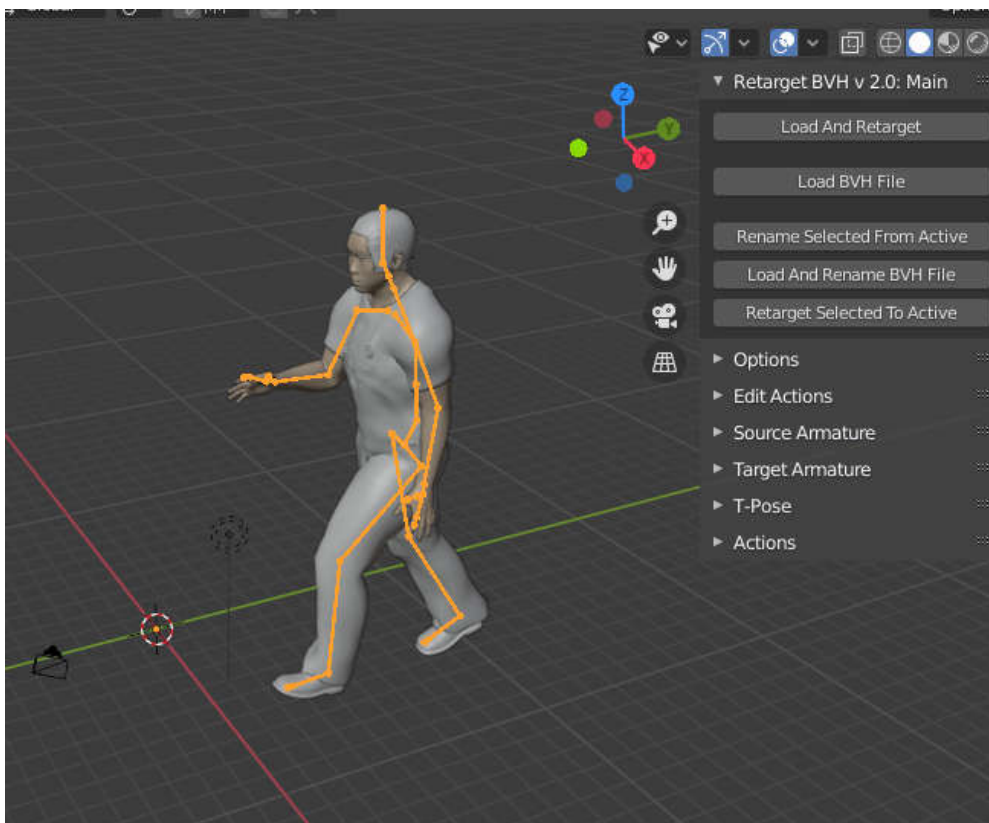
pilih tulang pada object mode maka tombol load and retarget akan aktif



pilih file bvh yang anda inginkan. klik load and retarget



hasilnya animasi tulang BVH akan langsung masuk ke obyek tinggal tes play maka obyek sudah bergerak sesuai gerakan bvh.



sama mudahnya dengan makewalk , yang jelas jauh lebih cepat dan lebih tepat gerakannya dibanding targetting bvh secara manual.

BAB 11 MOTION CAPTURE BERBASIS AI

11.1 Machine Learning: mengubah permainan Motion Capture

Penggunaan Motion Capture (MoCap) sangatlah penting di dunia animasi 3D dan pengembangan efek film profesional. Selama bertahun-tahun, teknik ini telah membuat para aktor mengenakan baju khusus yang biasanya berbahan spandeks untuk profesional dan melukis titik-titik di wajah mereka, semuanya ditujukan untuk menangkap ekspresi dan gerakan mereka sehingga gerakan mereka dapat membuat hidup obyek animasi 3D mereka. Namun sekarang, berkat machine learning, proses ini dapat berubah ke proses yang lebih nyaman.

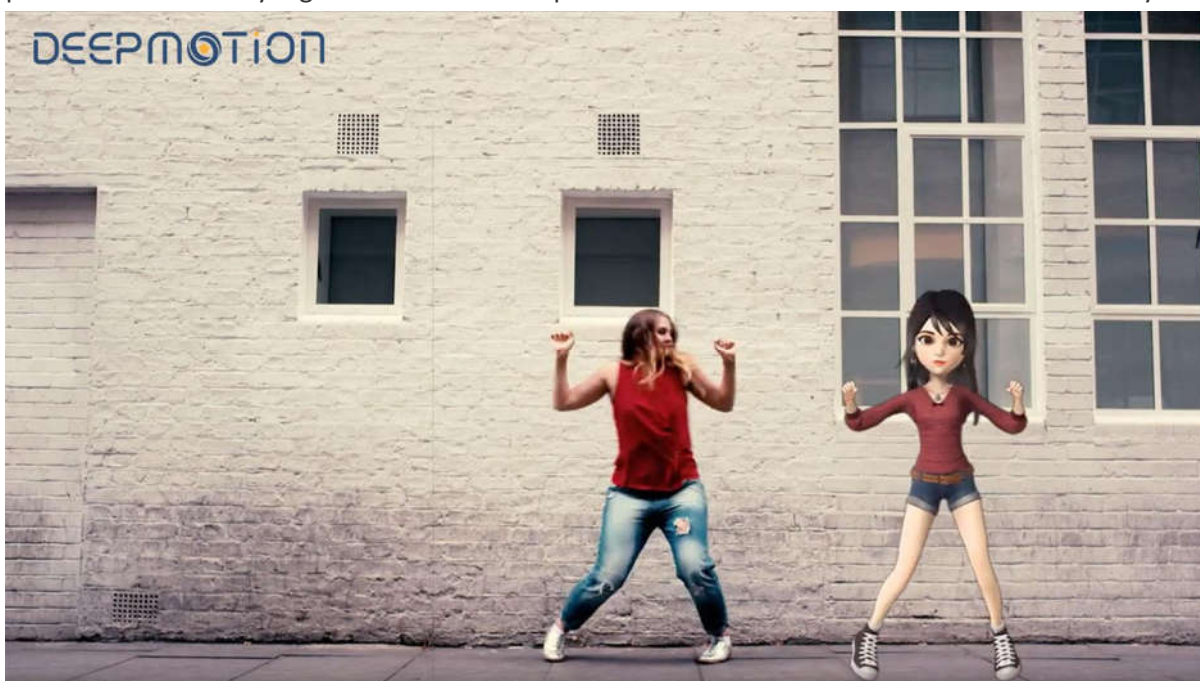
Di era AI, semakin banyak aplikasi computer vision dan machine learning (ML) yang membutuhkan estimasi 2D human pose sebagai input informasi. Ini juga melibatkan tugas selanjutnya dalam pengenalan gambar dan analitik video berbasis AI. Estimasi pose tunggal dan multi-orang adalah tugas komputer vision yang penting dan dapat digunakan dalam domain yang berbeda, seperti pengenalan tindakan, keamanan, olahraga, dan lainnya. Estimasi Pose masih merupakan teknologi komputer vision yang cukup baru. Namun, dalam beberapa tahun terakhir, akurasi estimasi pose manusia mencapai terobosan besar dengan munculnya Convolutional Neural Networks (CNNs)

Tujuan dari machine learning adalah untuk mengajarkan sistem komputer untuk membuat keputusan dari kumpulan data besar. Teknik ini telah membuktikan dirinya untuk memecahkan masalah yang lebih sulit dari industri VFX, membantu seniman mencapai hasil yang sebelumnya tidak dapat mereka lakukan dengan cara yang lebih cepat dan lebih efektif.

Dengan pemikiran ini, menjadi lebih mudah untuk melihat mengapa teknik machine learning diterapkan pada motion capture. Dimana sebelumnya MoCap melibatkan tugas-tugas yang lebih membosankan, seperti seniman mengecat head rig, penggunaan machine learning berjanji untuk membuat proses lebih efisien dan meskipun ini adalah konsep yang relatif baru, ada sejumlah perusahaan VFX yang beralih ke machine learning dan mengungkap kemungkinannya.



Salah satu perusahaan yang memelopori penggunaan machine learning ini, bersama dengan kecerdasan buatan, adalah DeepMotion. Bersama-sama, kedua rangkaian teknologi ini bertujuan untuk memecahkan masalah yang dapat terjadi saat menggunakan MoCap—seperti peralatan ekstensif yang sering dibutuhkan—alih-alih menganjurkan penggunaan motion capture tanpa penanda, memungkinkan animasi yang cerdas dan realistis. Mereka juga lebih lanjut memungkinkan pembuatan karakter yang lebih interaktif dan perluasan saluran animasi dan data simulasi lainnya.

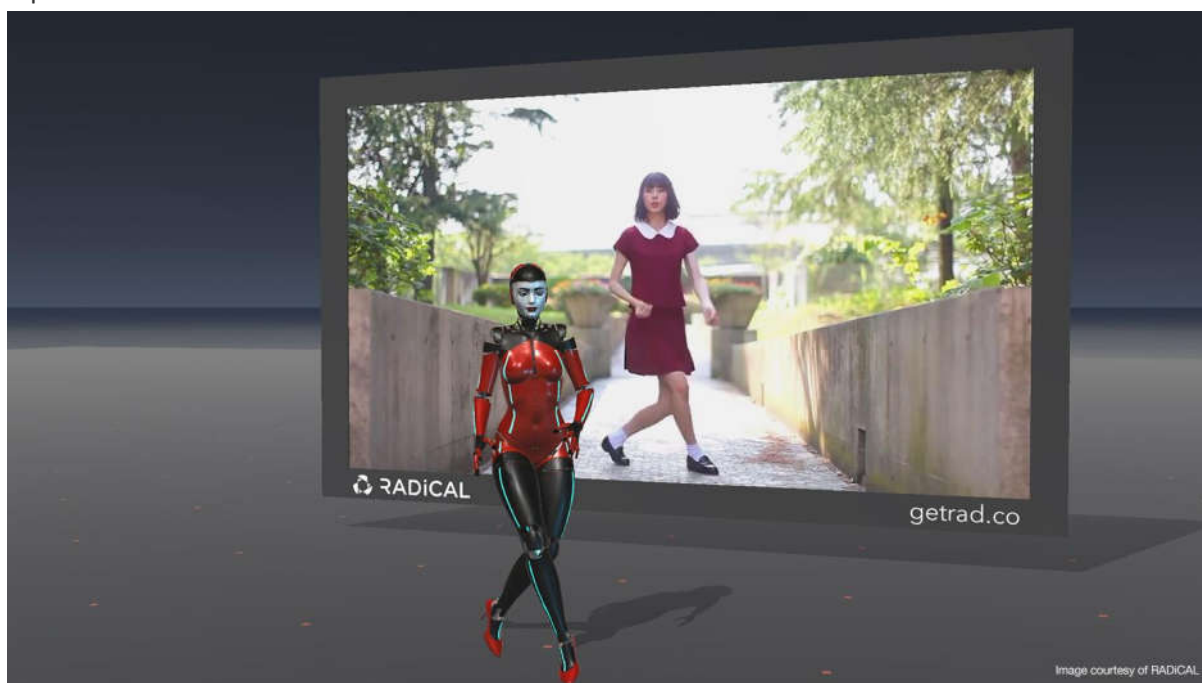


RADiCAL adalah perusahaan lain yang menggunakan machine learning dan AI untuk meningkatkan motion capture. Namun, fokus mereka adalah mendeteksi dan merekonstruksi

gerakan manusia 3D dari konten 2D, semuanya tanpa menggunakan setelan dan perangkat keras MoCap konvensional. Terlebih lagi, tim ini juga membuat kemajuan dengan penggunaan cloud di bidang motion capture.

Dengan mengirimkan dan mengunggah video seseorang ke sistem cloud mereka, RADiCAL dapat mengirim kembali animasi gerakan yang diambil dari video tersebut. Sekarang, tentu saja, untuk lebih banyak konten amatir, ini menjanjikan gerakan realistis dan pembuatan animasi. Tapi untuk blockbuster atau acara TV, ini hanya menggores permukaan dari apa yang mungkin terjadi di masa depan.

Sementara pekerjaan perusahaan ini masih dalam pengembangan, kemungkinan yang mereka hadirkan untuk industri VFX sangatlah besar. Tidak hanya memungkinkan metode motion capture yang lebih dinamis, tetapi juga pengurangan jumlah peralatan yang diperlukan untuk melakukannya. Selain itu, teknik pengambilan dengan resolusi lebih tinggi dan algoritme pemrosesan yang canggih ini cenderung meningkatkan kualitas pekerjaan yang dihasilkan dan hasil yang dapat diprediksi.



Seperti yang disebutkan sebelumnya, motion capture tetap menjadi salah satu sumber data gerak yang lebih populer, biasanya ini membutuhkan banyak peralatan. Ini adalah salah satu alasan utama telah terjadi pergeseran ke teknologi motion capture tanpa penanda, bahkan muncul di film-film terkenal seperti *The Irishman* dan penggunaan teknologi deepfake.

Dan meskipun ini adalah penggunaan teknik yang berhasil, ini membutuhkan pengaturan kamera tugas berat dan kerja keras dari tim VFX. Namun, penelitian dari University of California mengeksplorasi penerapan pembelajaran penguatan mendalam sebagai cara untuk menangkap data gerak. Cabang machine learning khusus ini berfokus pada pelatihan model AI untuk belajar dari pengalaman masa lalu dan bertujuan untuk mengurangi penggunaan peralatan secara intensif.

Dengan menggunakan konsep tersebut, tim peneliti menemukan bahwa model AI dapat belajar dari sejumlah materi referensi yang telah diberikan kepada mereka, termasuk sumber yang lebih terbatas seperti video YouTube. Ini membuka kemungkinan untuk desain karakter yang lebih

cepat dan lebih mudah, di samping cara yang lebih efisien untuk membersihkan noise gaya analog—semua berdasarkan apa yang dipelajari model tentang cara tubuh manusia bergerak.

Bahkan, de-noising data MoCap menjadi lebih banyak digunakan dalam industri ini, dengan digunakan untuk membuat Thanos di *Avengers: Infinity War*. Seperti yang jelas bagi para penggemar bermata elang itu, tampilan keseluruhan Thanos berubah dari saat ia pertama kali terlihat di *The Avengers*, hingga *Perang Infinity*. Ini sebagian besar disebabkan oleh perbedaan dalam cara Digital Domain menggunakan motion capture untuk membuat karakter.

Dalam *Infinity War*, tim ingin melihat lebih banyak ekspresi Josh Brolin di Thanos untuk menciptakan karakter yang realistis dan dapat dipercaya. Untuk melakukan ini, Digital Domain menggunakan perangkat lunak machine learning baru yang disebut Masquerade. Titik-titik pelacakan diletakkan di wajah Brolin, seperti yang biasa terjadi pada MoCap, dan kemudian direkam oleh sepasang kamera HD. Rekaman ini kemudian dikirim melalui algoritme machine learning yang berisi basis data besar pemindaian wajah beresolusi tinggi dari wajah Brolin.

Masquerade kemudian mengambil mesh beresolusi rendah ini dan menemukan solusi terbaik untuk bentuk wajah beresolusi tinggi. Setelah tim mendapatkan solusinya, mereka dapat menyesuaikannya namun diperlukan agar terlihat tepat. Ini terbukti menjadi penghemat waktu yang sangat besar bagi para seniman yang mengerjakan proyek karena itu berarti mereka tidak perlu menyesuaikan setiap inci CG Thanos secara manual, yang seringkali merupakan tugas yang agak membosankan.



Salah satu tantangan utama machine learning adalah potensinya untuk menyingkirkan artis dari proses kreatif dan teknis produksi. Namun, saat kita melihat bagaimana machine learning bekerja bersama dengan motion capture, jelas bahwa keduanya tidak t mematuhi ini.

Sebaliknya, mereka bekerja bersama secara harmonis — sesuatu yang dengan senang hati dialami oleh Digital Domain dalam pembuatan tim Thanos. Dalam hal ini, machine learning meningkatkan cara seniman bekerja dengan motion capture, memungkinkan mereka mencapai hasil

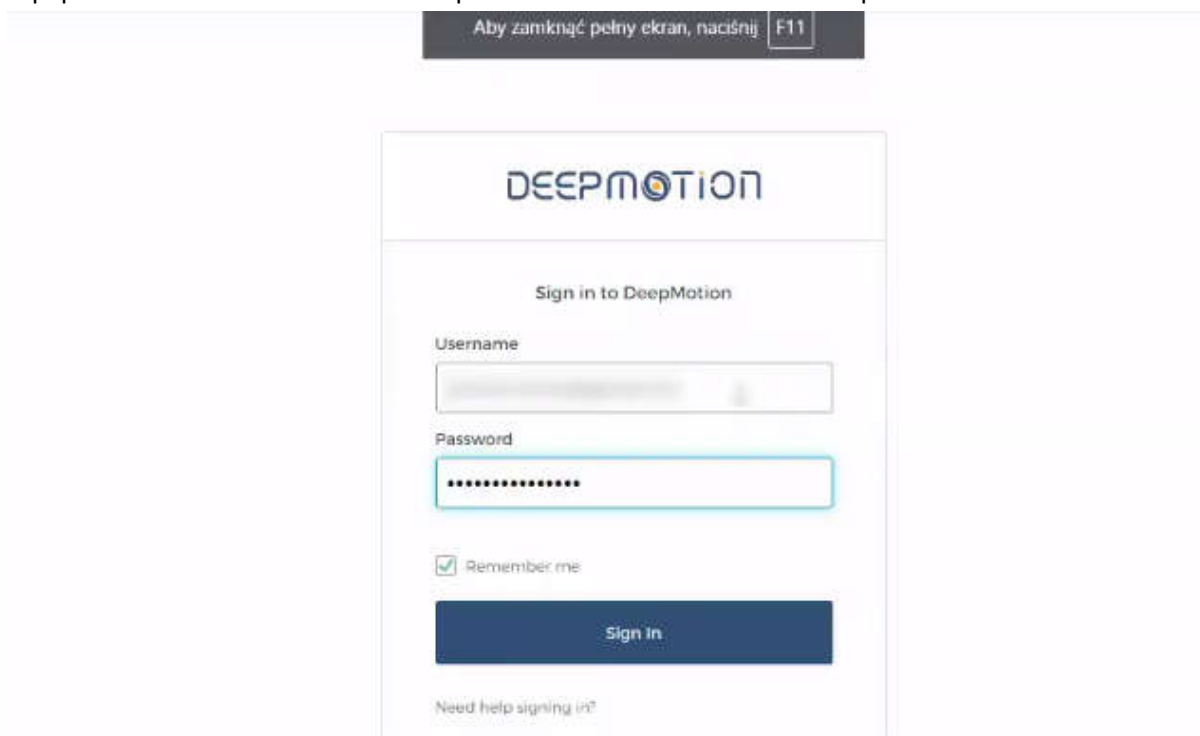
yang diinginkan dengan lebih cepat dan lebih efisien. Tampaknya ketika machine learning terus berkembang, tidak hanya akan membuka jalan bagi penggunaan motion capture yang lebih maju dan dinamis, tetapi juga meluas lebih jauh ke area lain dari industri VFX dan terus meningkatkan kualitas hidup artis.

11.2 ai motion capture dengan deep motion

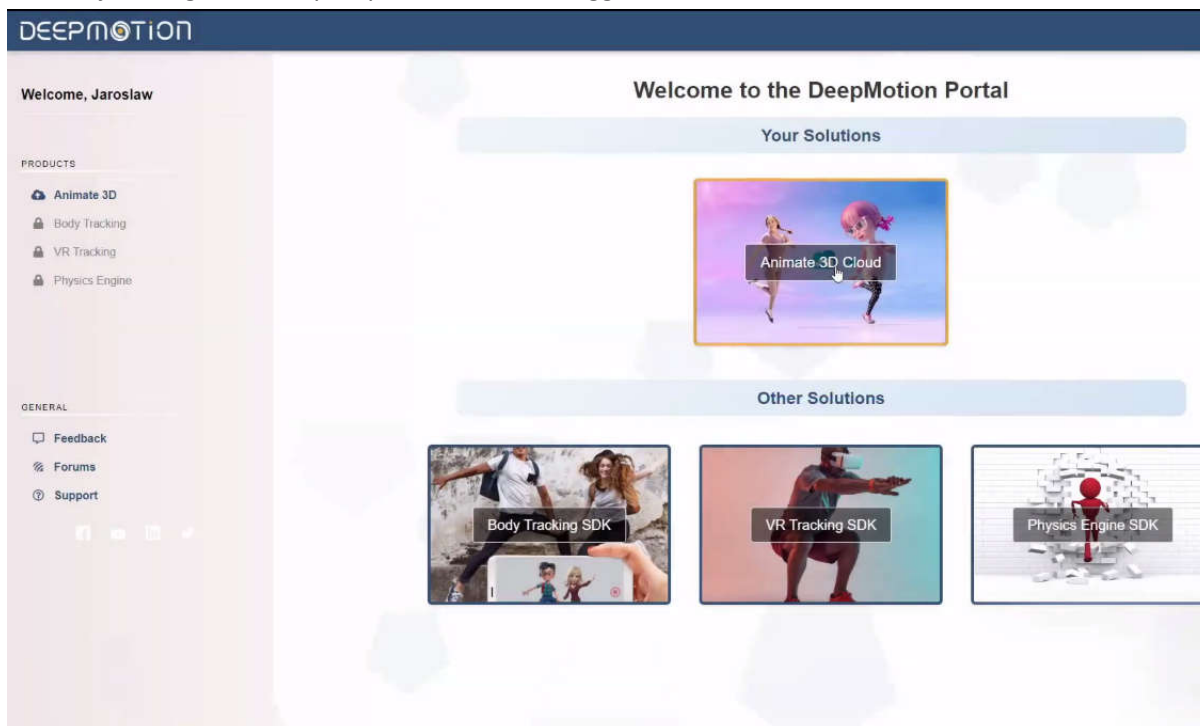
dalam materi ini dibahas cara membuat motion capture otomatis dengan deep motion



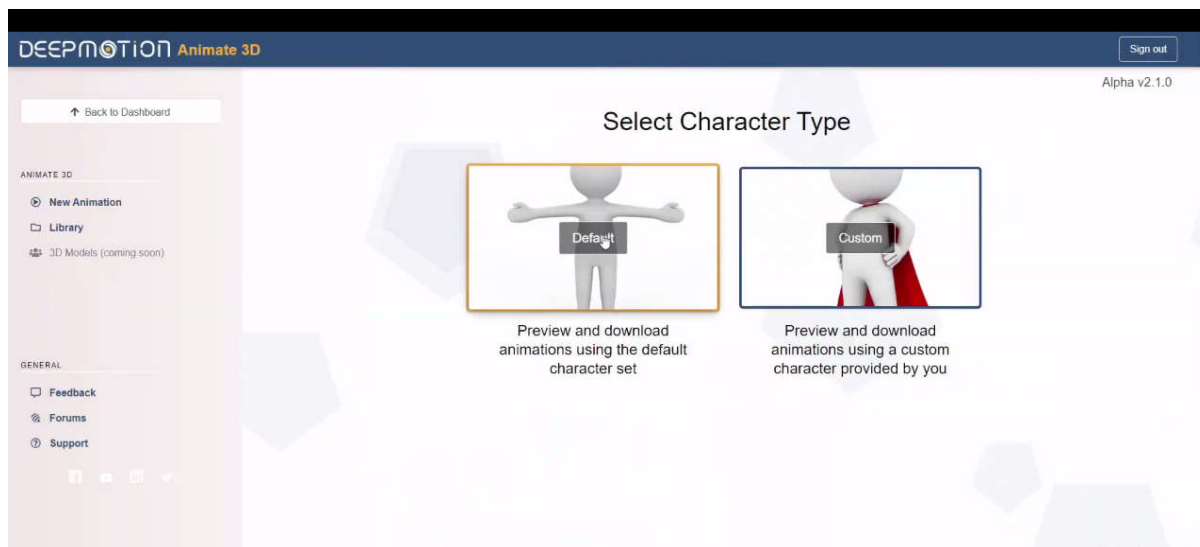
tapi pertama-tama mari kita lihat deep motion ini adalah anda harus dapat mendaftar dan masuk



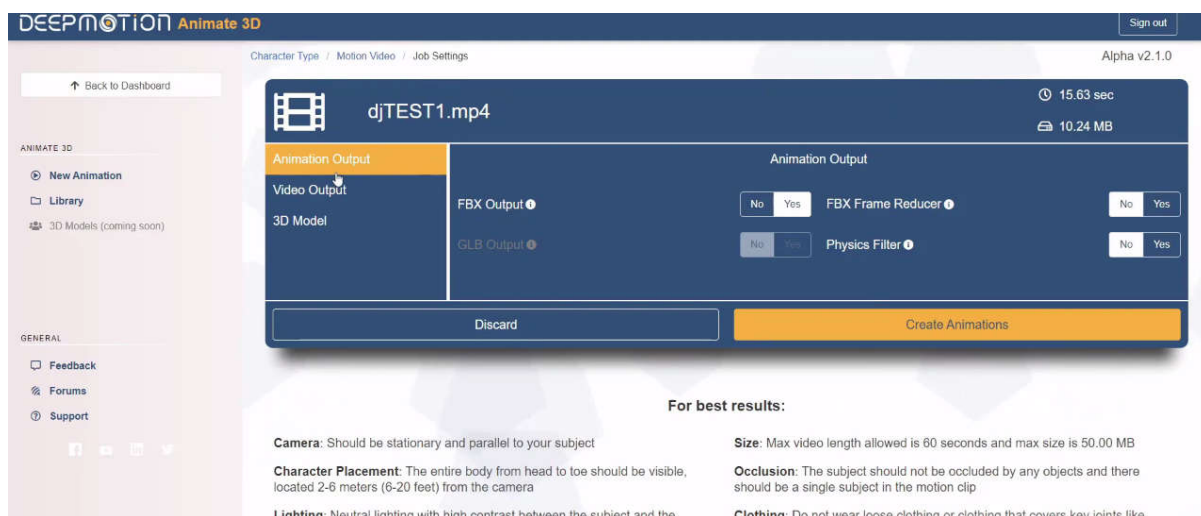
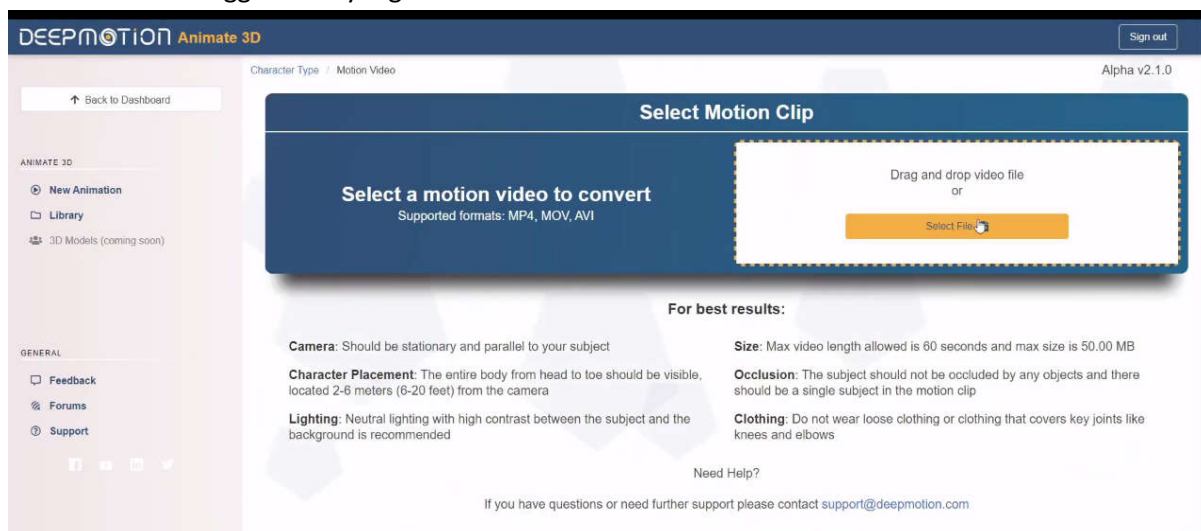
untuk uji coba gratis cukup siapkan video dan unggah di animate 3d cloud



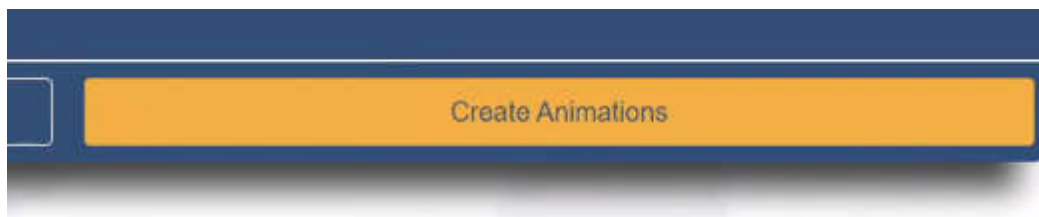
pilih file



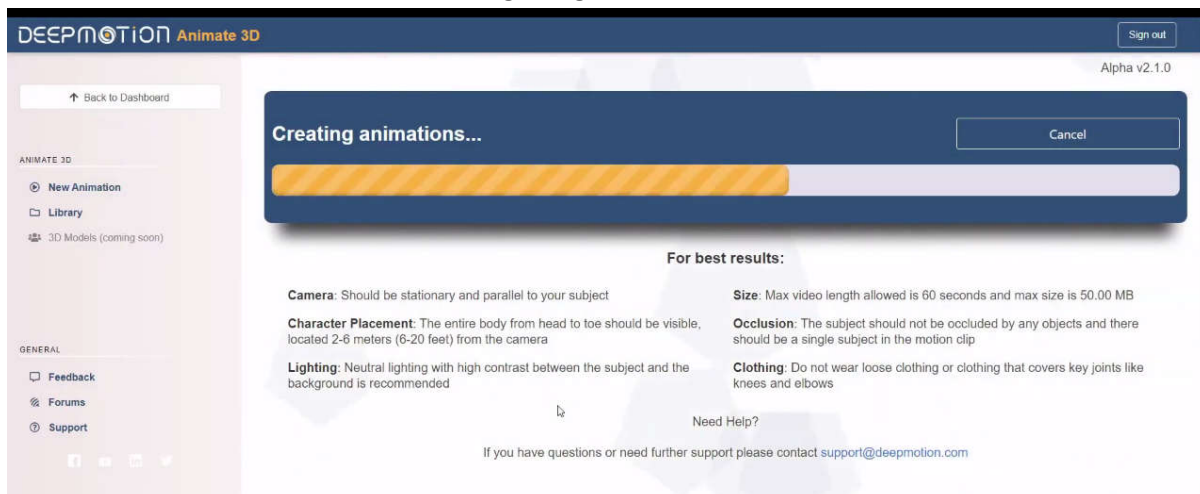
dan unggah video Anda sekarang. ada beberapa opsi itu Anda dapat mengatur sesuai keinginan, di sini kita akan menggunakan yang default



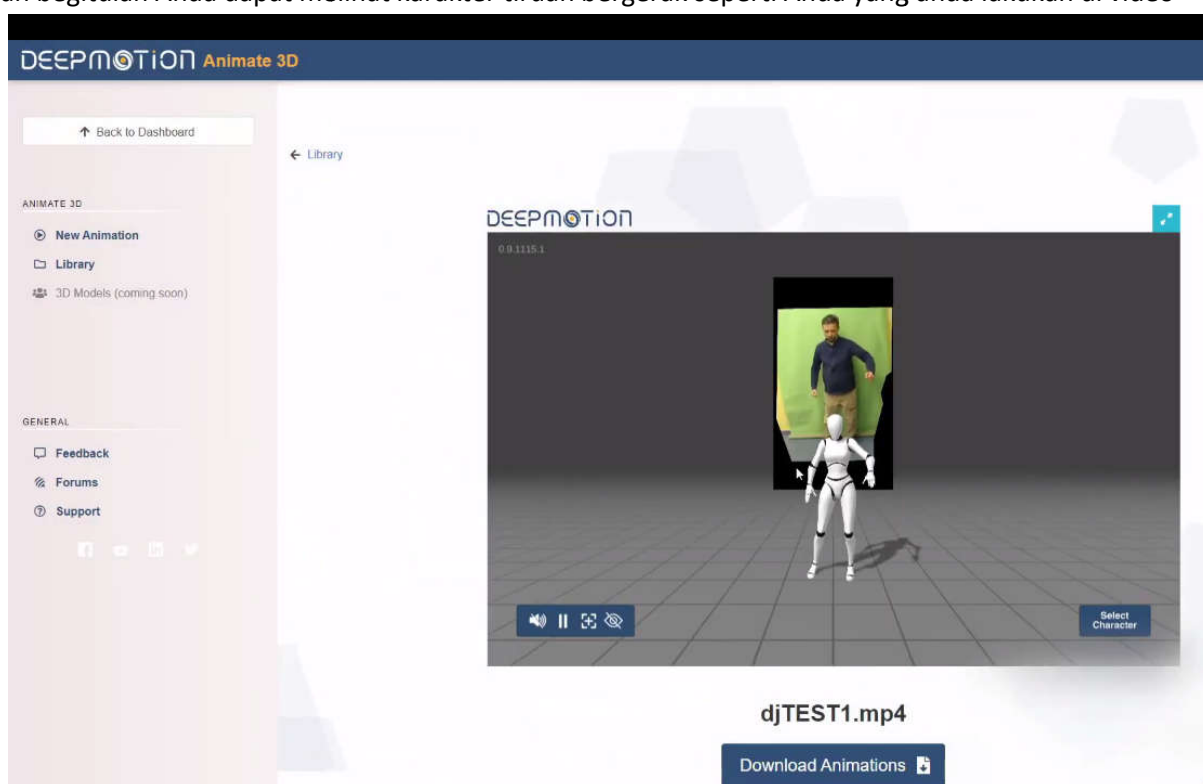
dan klik create animation



animasi akan dibuat secara otomatis dengan algoritma ai



dan begitulah Anda dapat melihat karakter tiruan bergerak seperti Anda yang anda lakukan di video



Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

hasilnya biasanya cukup bagus kecuali jika Anda mengambil karakter yang sangat berbeda dari Anda misalnya membuat obyek karakter yang gendut sementara anda sangat kurus ,sehingga Anda dapat membuat



beberapa artefak tetapi pada umumnya deep motion mampu melakukan motion capture yang baik membuat setidaknya gerakan dasar ada beberapa tips untuk membuat video Anda mendapatkan hasil terbaik dari algoritme ini dan ada batasan ukuran video yang Anda unggah

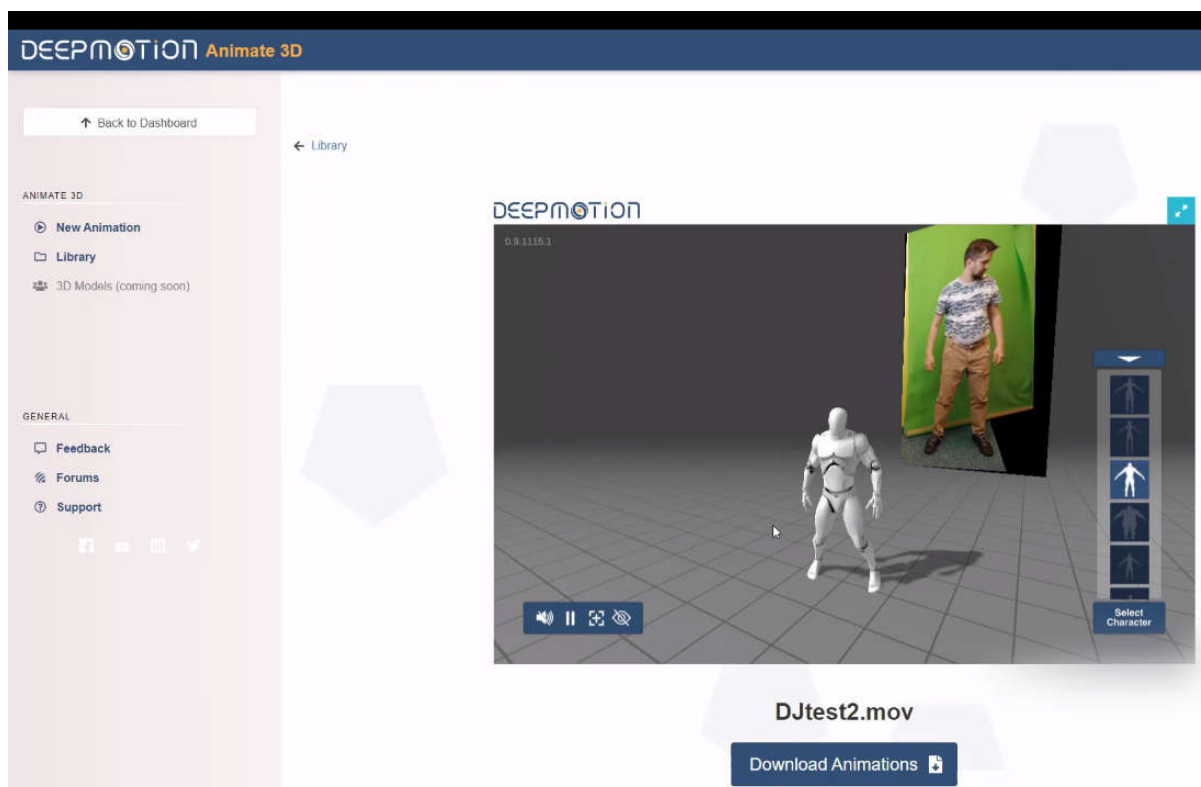
For best results:

<p>Camera: Should be stationary and parallel to your subject</p> <p>Character Placement: The entire body from head to toe should be visible, located 2-6 meters (6-20 feet) from the camera</p> <p>Lighting: Neutral lighting with high contrast between the subject and the background is recommended</p>	<p>Size: Max video length allowed is 60 seconds and max size is 50.00 MB</p> <p>Occlusion: The subject should not be occluded by any objects and there should be a single subject in the motion clip</p> <p>Clothing: Do not wear loose clothing or clothing that covers key joints like knees and elbows</p>
---	--

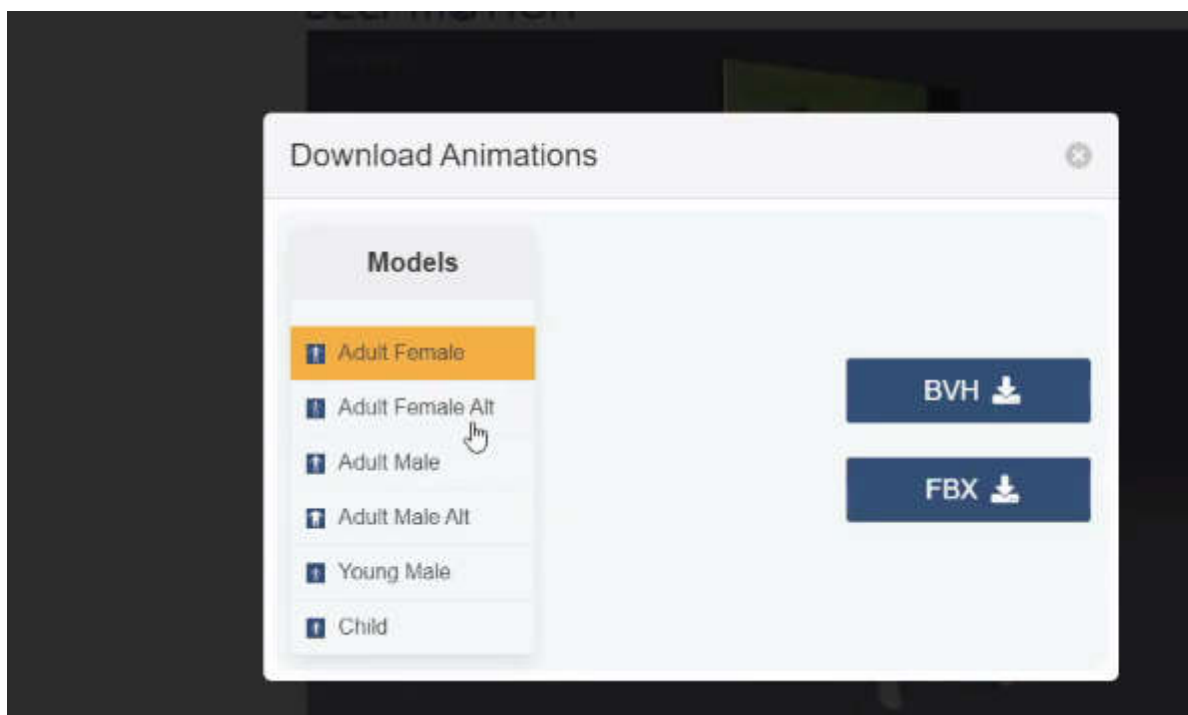
Need Help?

If you have questions or need further support please contact support@deepmotion.com

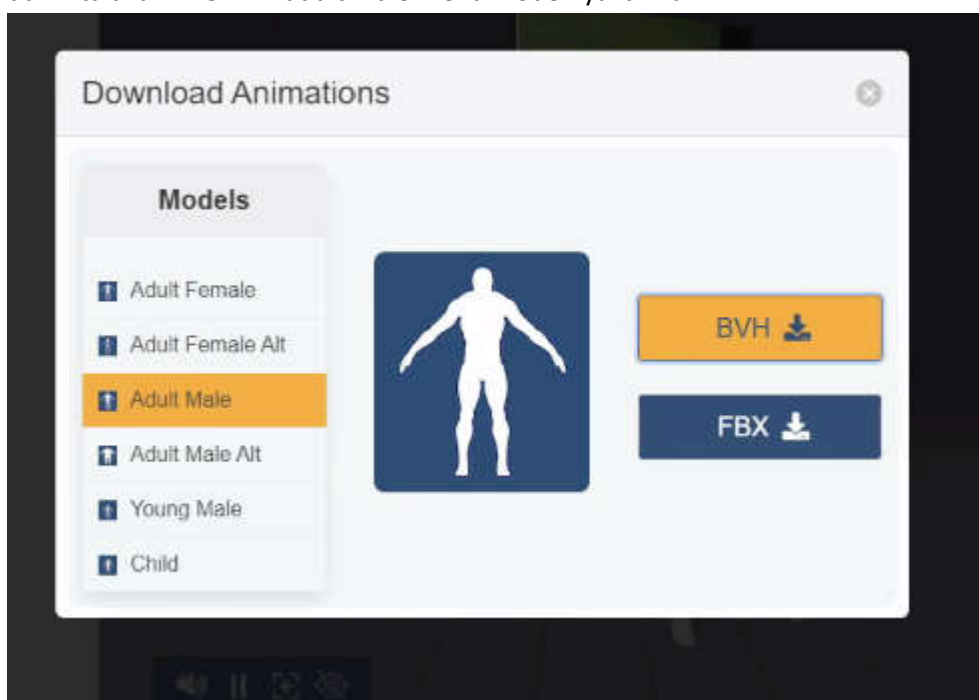
jadi di sini kita mengunggah versi dengan tangkapan diri skita yang mengikuti aturan diatas,maka hasilnya sedikit lebih baik dan mesh mampu bergerak cukup lancar sekarang



setelah selesai kita cukup mengklik unduh animasi

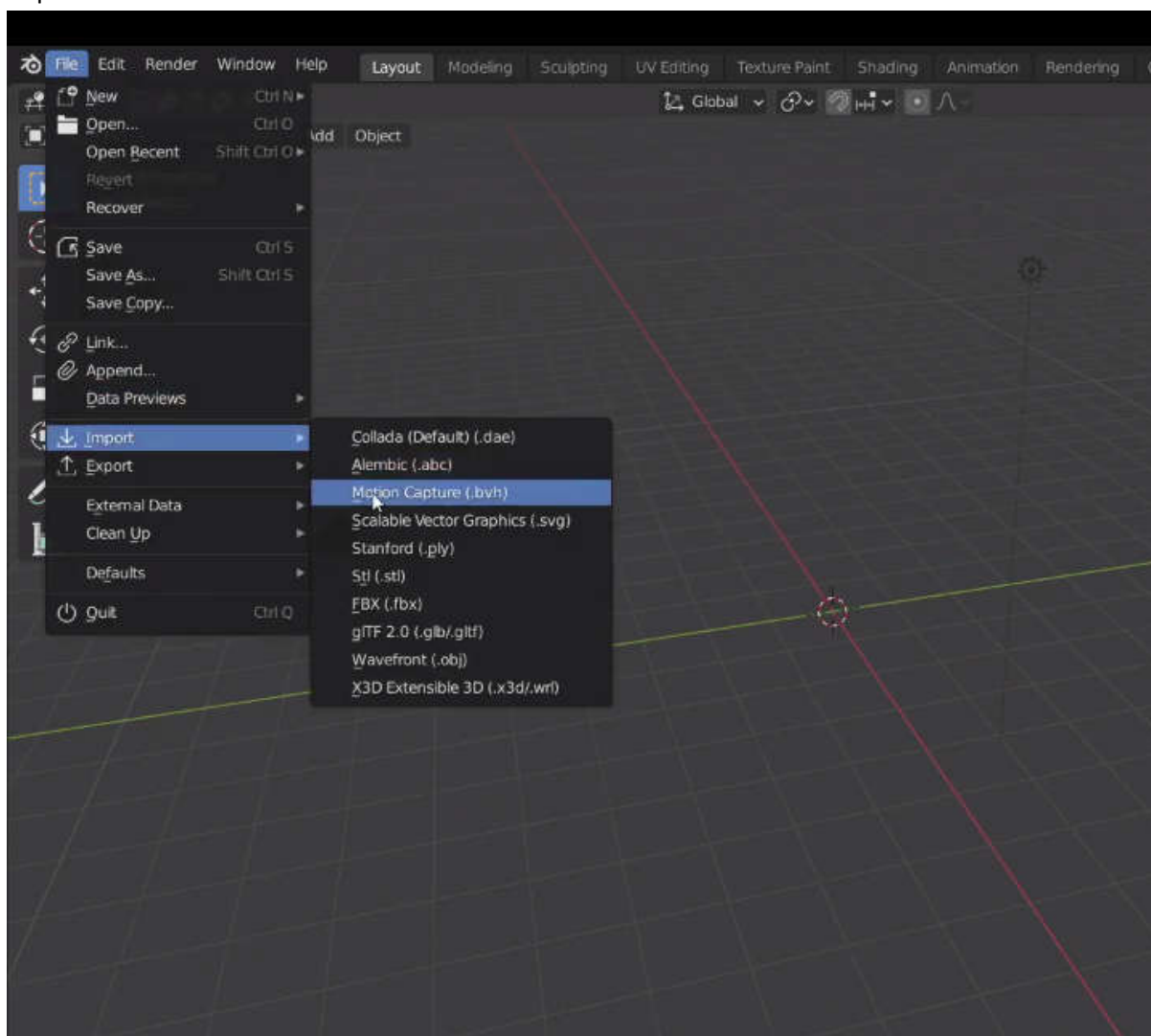


dan kita akan memilih adult male krena modelnya laki laki

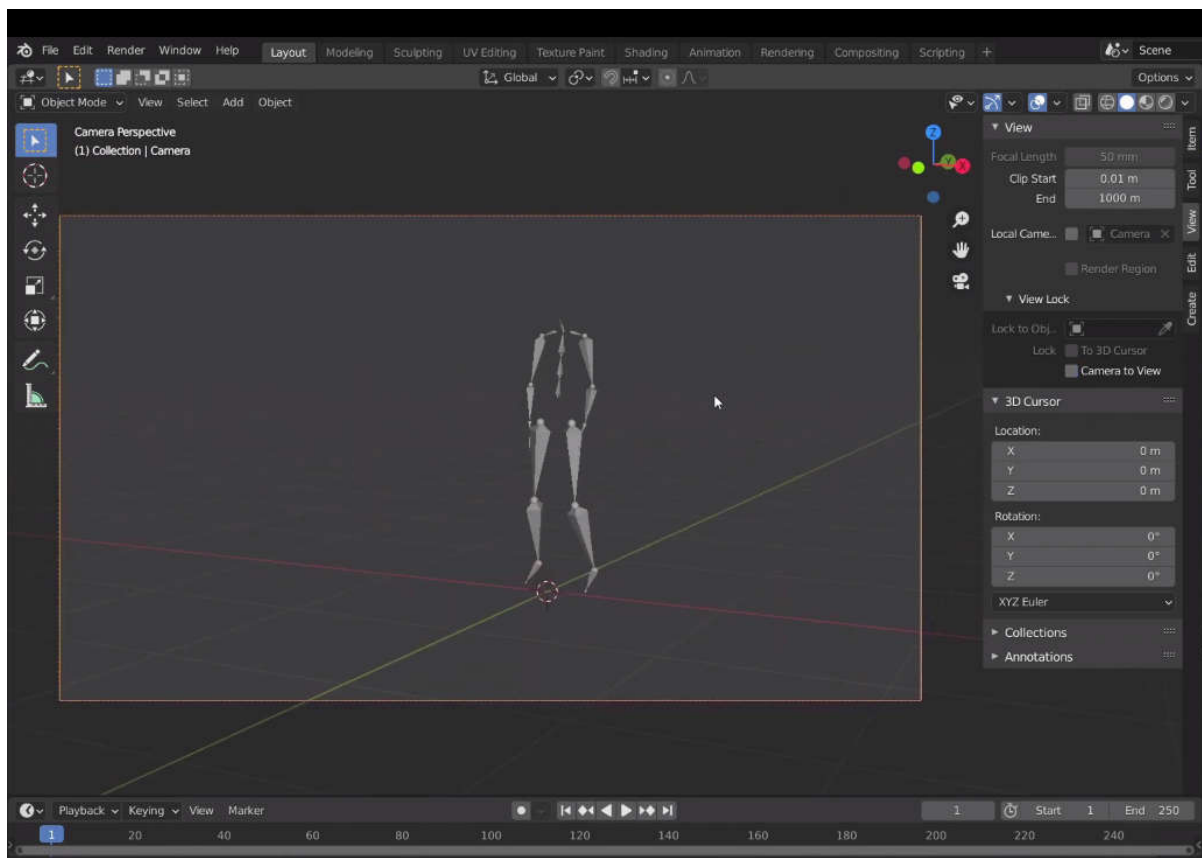


dan kita akan memilih file bvh yang sepertinya paling cocok untuk kita impor ke blender

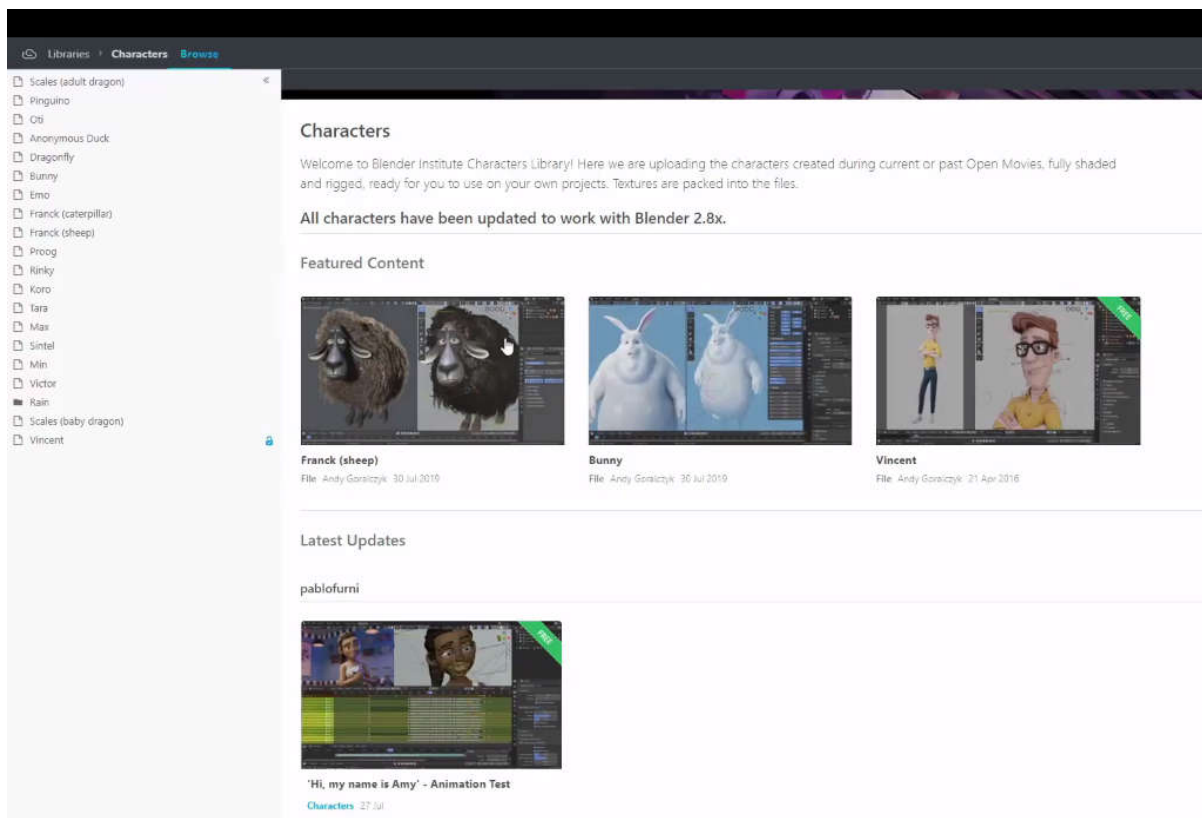
impor bvh ke blender



letakkan di adegan kita cukup skalakan dengan tepat dan di sana kita memiliki kerangka yang bergerak sesuai yang ada di deepmotion

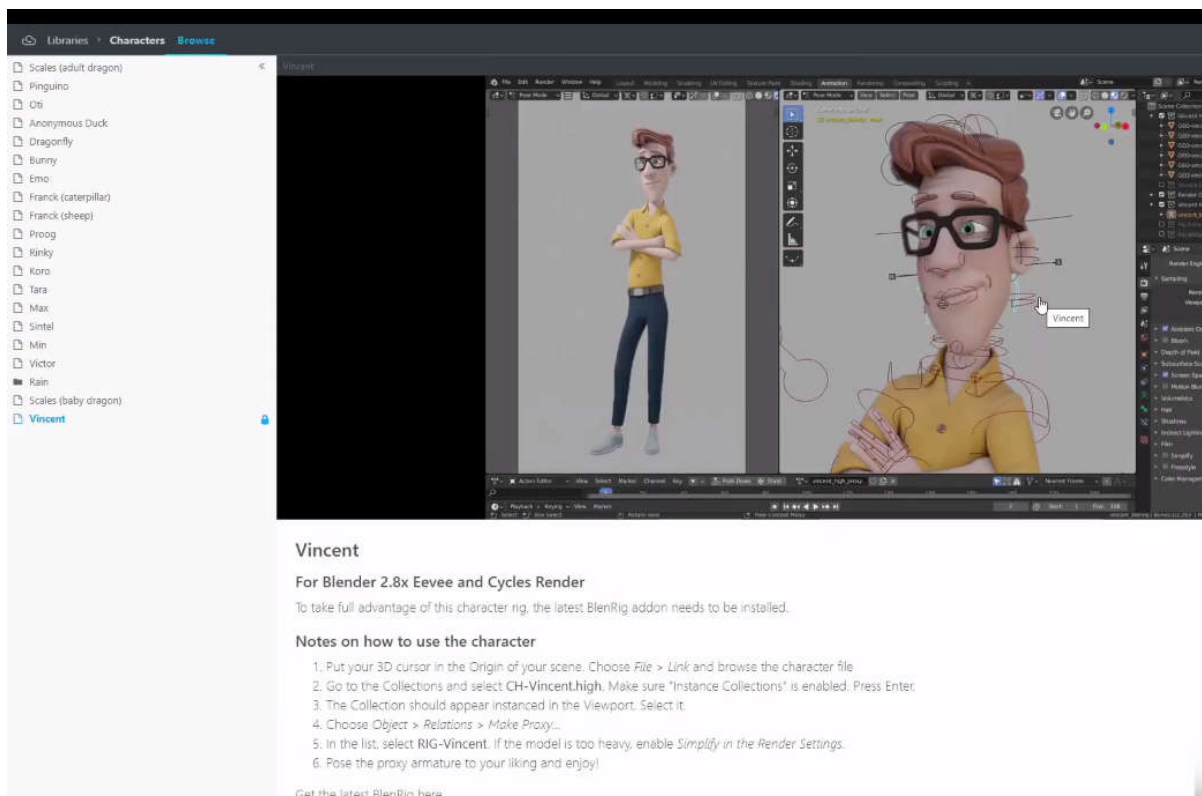


sekarang mari kita pergi dan tautkan armature siap itu ke karakter siap pakai yang akan kita unduh dari blender cloud



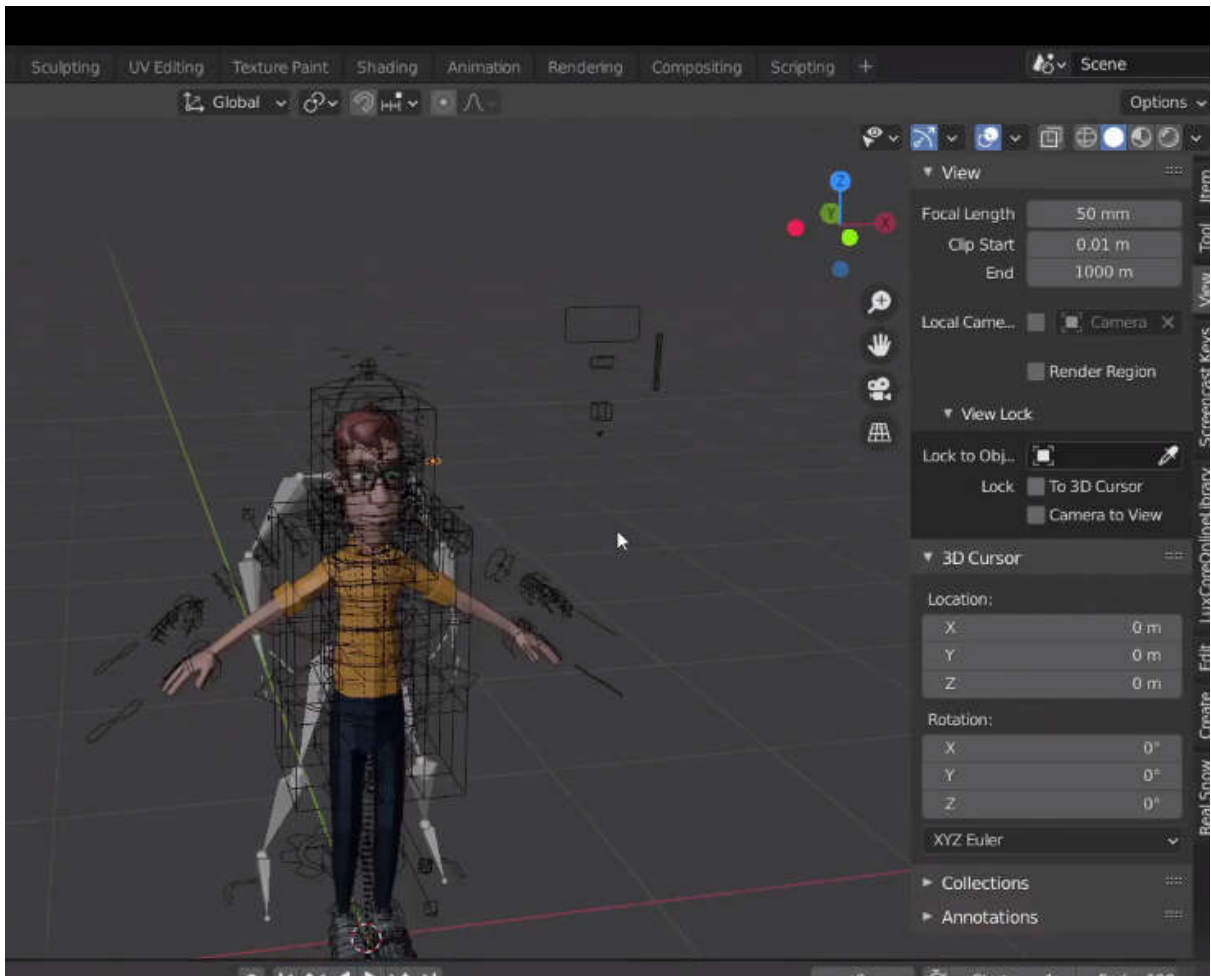
kita akan menggunakan vincent

Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

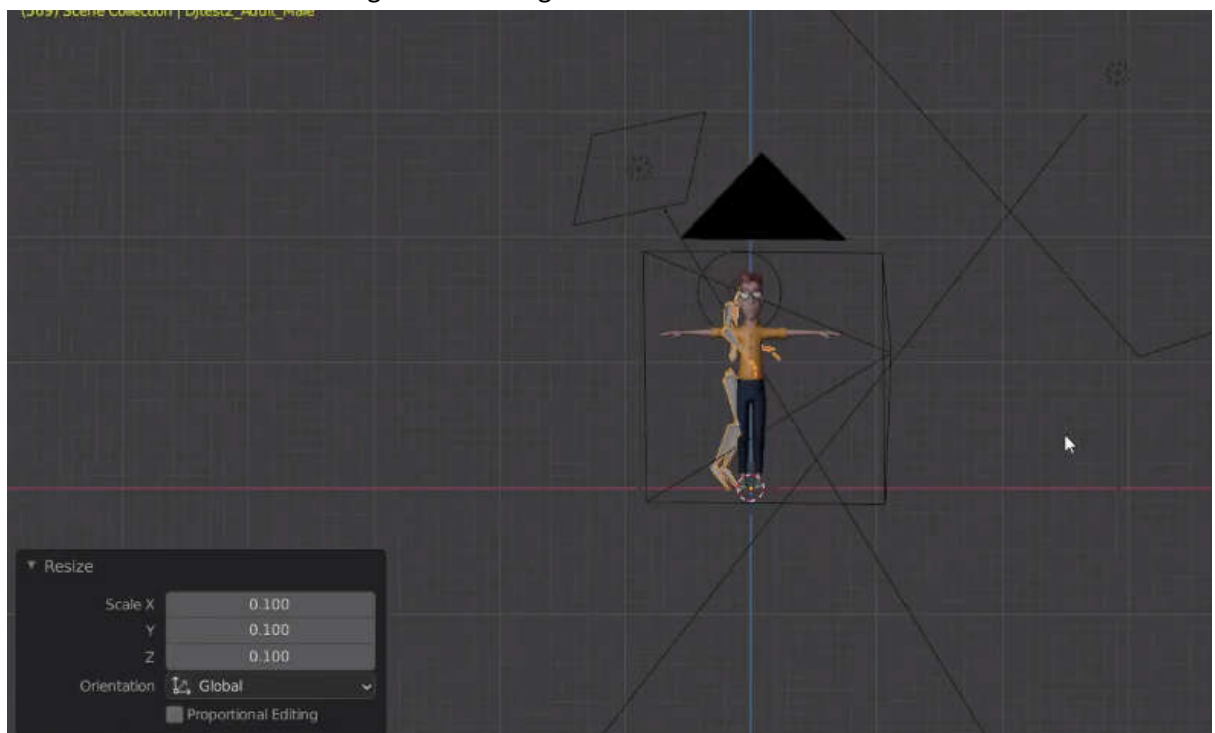


vincent itu adalah karakter rig terperinci, jadi kita baru saja menyingkirkan seluruh rig menyisakan bagian mata saja dan menyederhanakan mesh hanya untuk membuatnya hal-hal yang lebih

sederhana

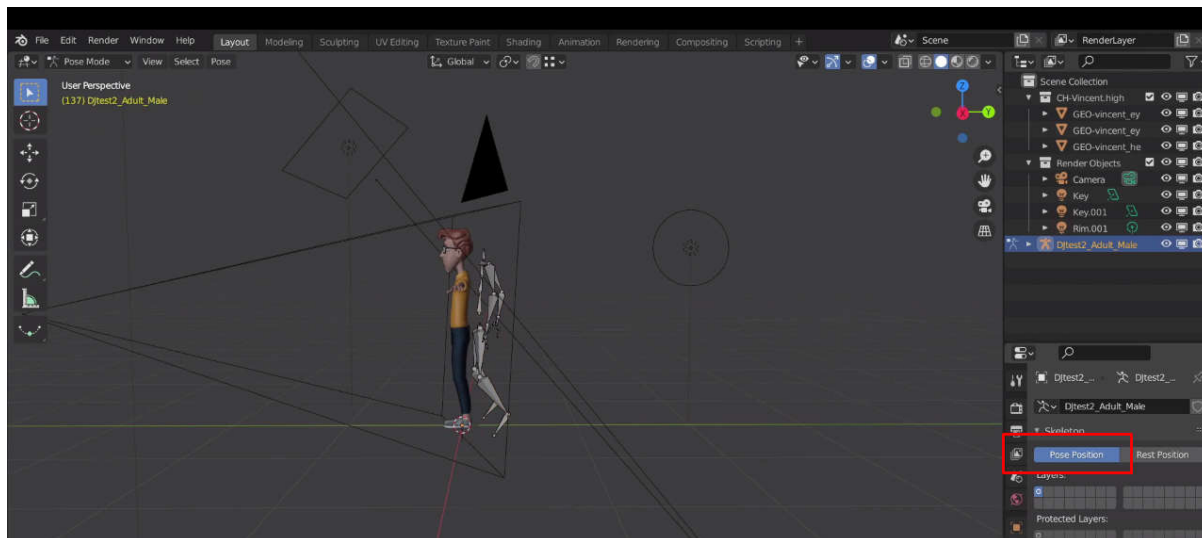


mari sesuaikan skala armature agar sesuai dengan vincent

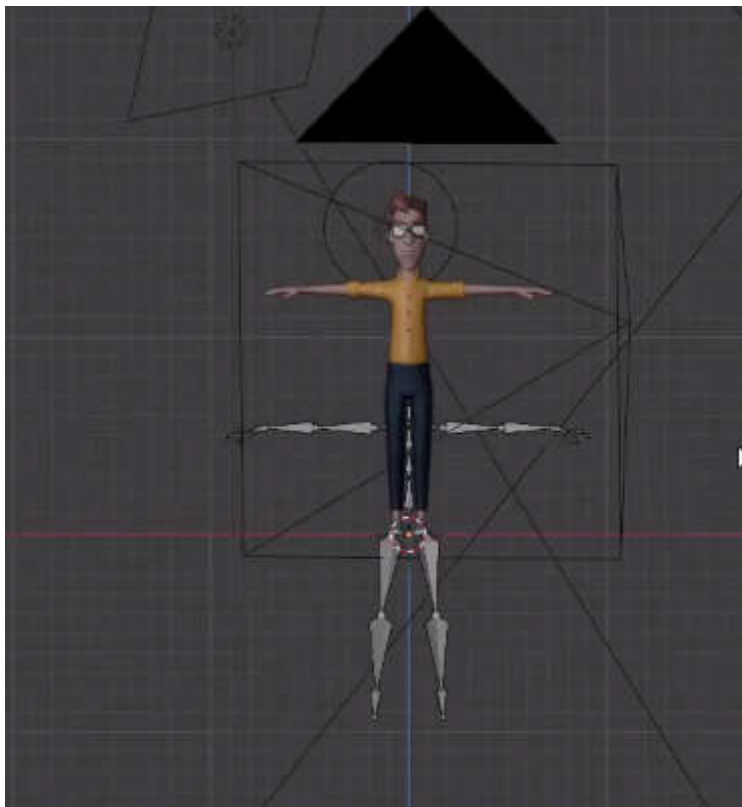


Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

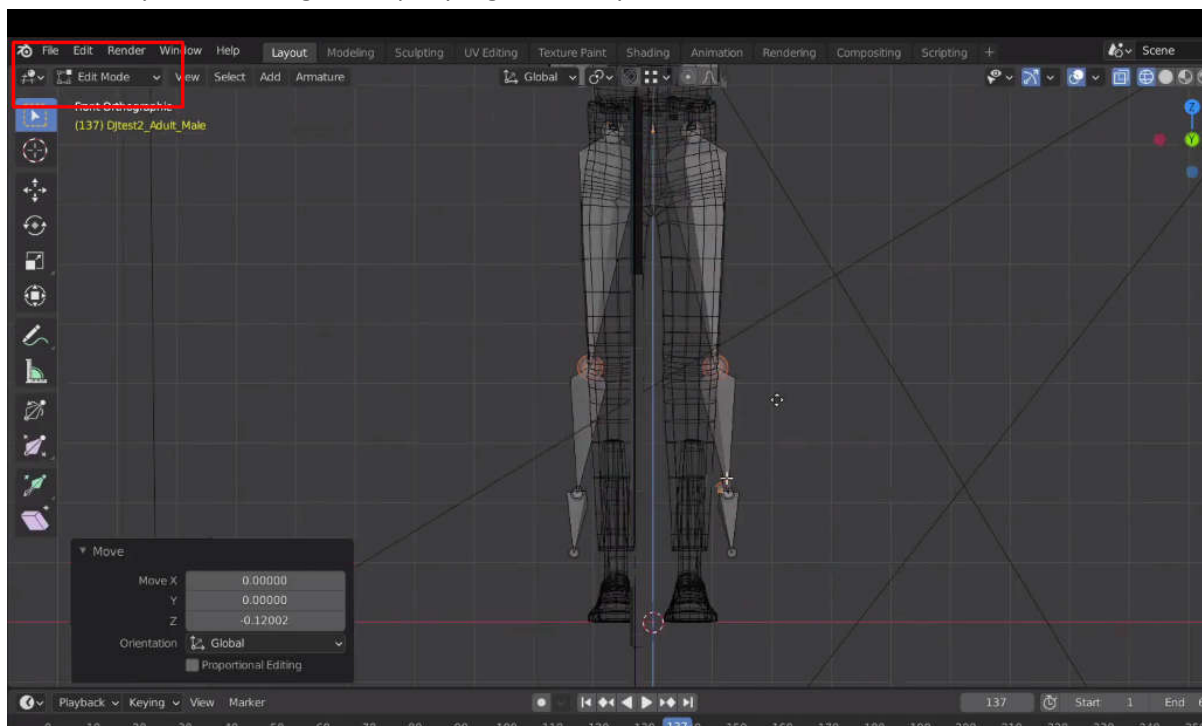
dan mari kita letakkan di dekat vincent. tetapi Anda dapat melihat posenya sangat berbeda jadi mari kita pastikan dalam mode pose bahwa armature disetel ke posisi rest yang merupakan T pose



dan sekarang mari kita letakkan armature dan scale itu sesuai agar sesuai dengan mesh

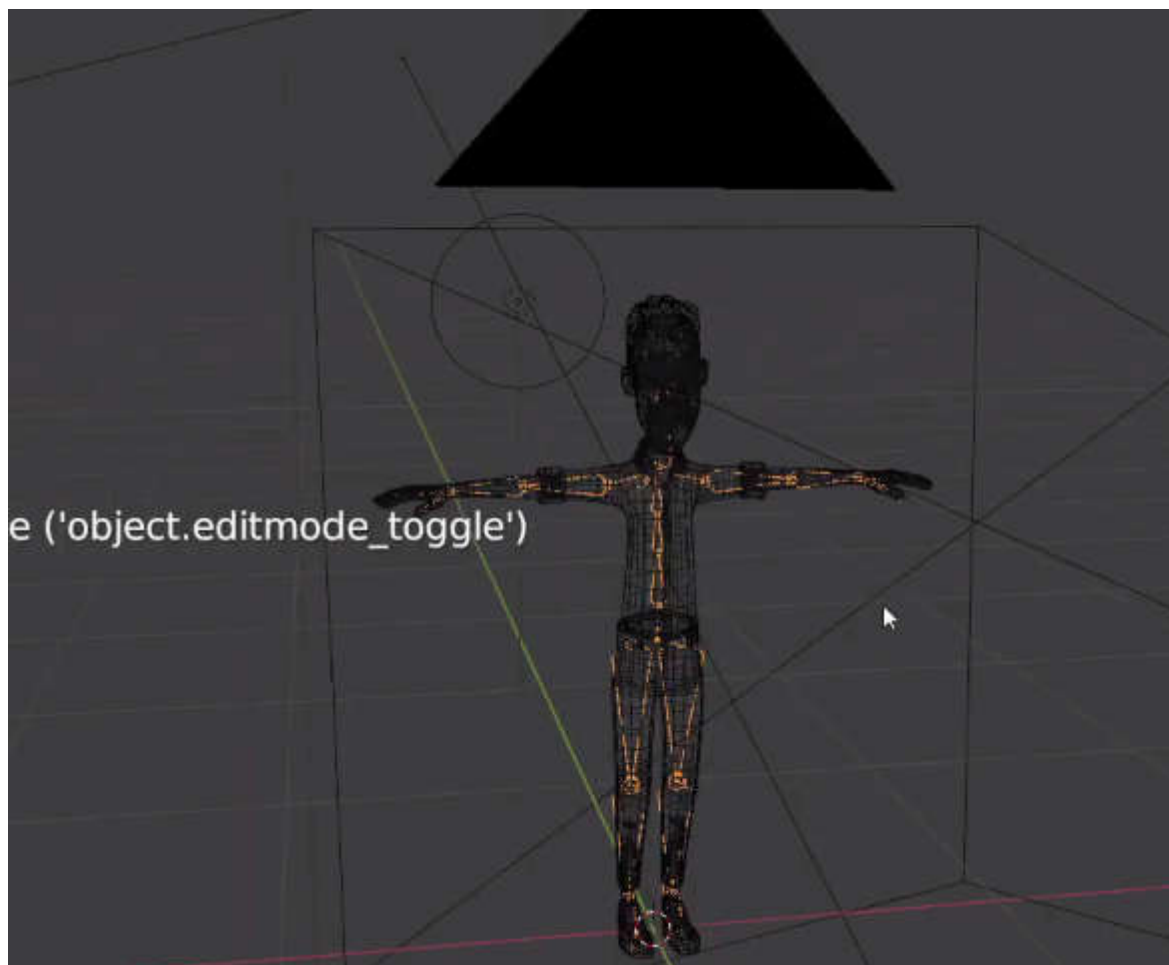


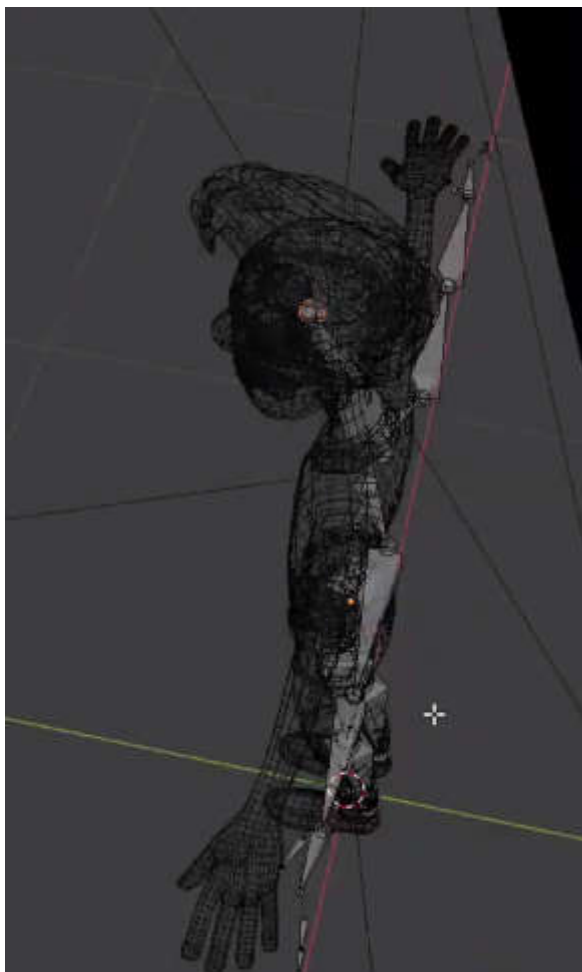
sekarang kita akan pindah ke mode edit dan hanya memindahkan sambungan kunci dan membuatnya sesuai dengan tempat yang seharusnya



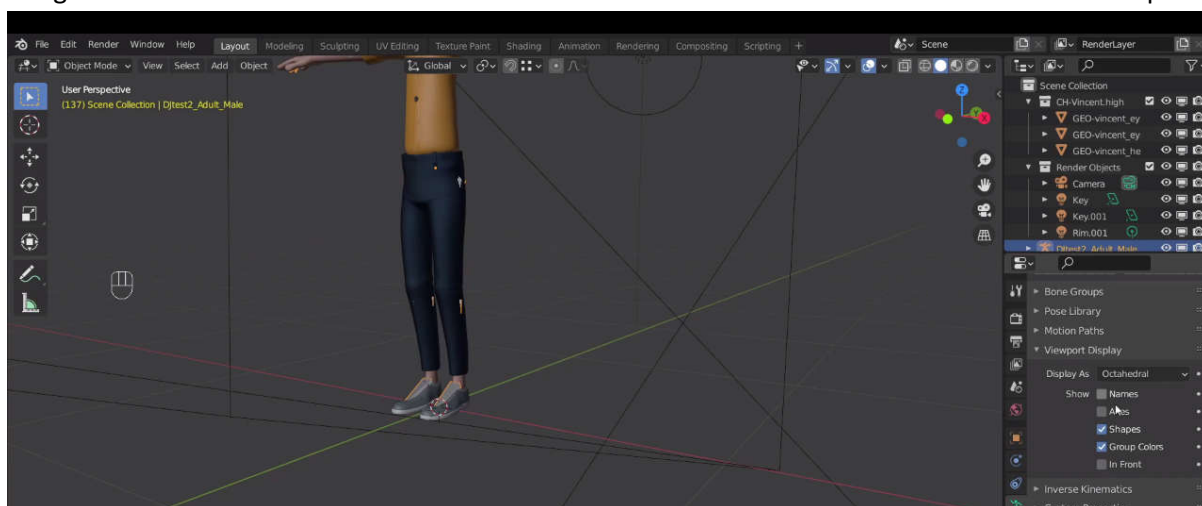
2:18

luangkan waktu Anda dan buat seakurat mungkin karena itu akan memengaruhi kualitas keseluruhan



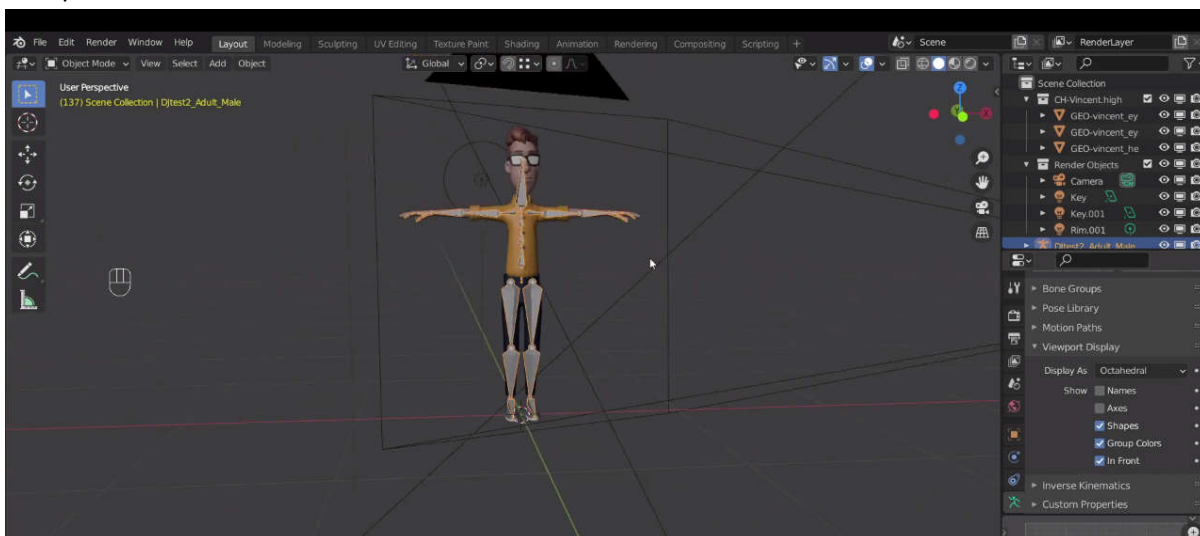


animasi Anda pada akhirnya, tetapi pastikan Anda mencentang semua tiga dimensinya jadi buka tampilan depan, tampilan samping, dan tampilan atas, dan pastikan armature cocok dengan mesh dengan sempurna

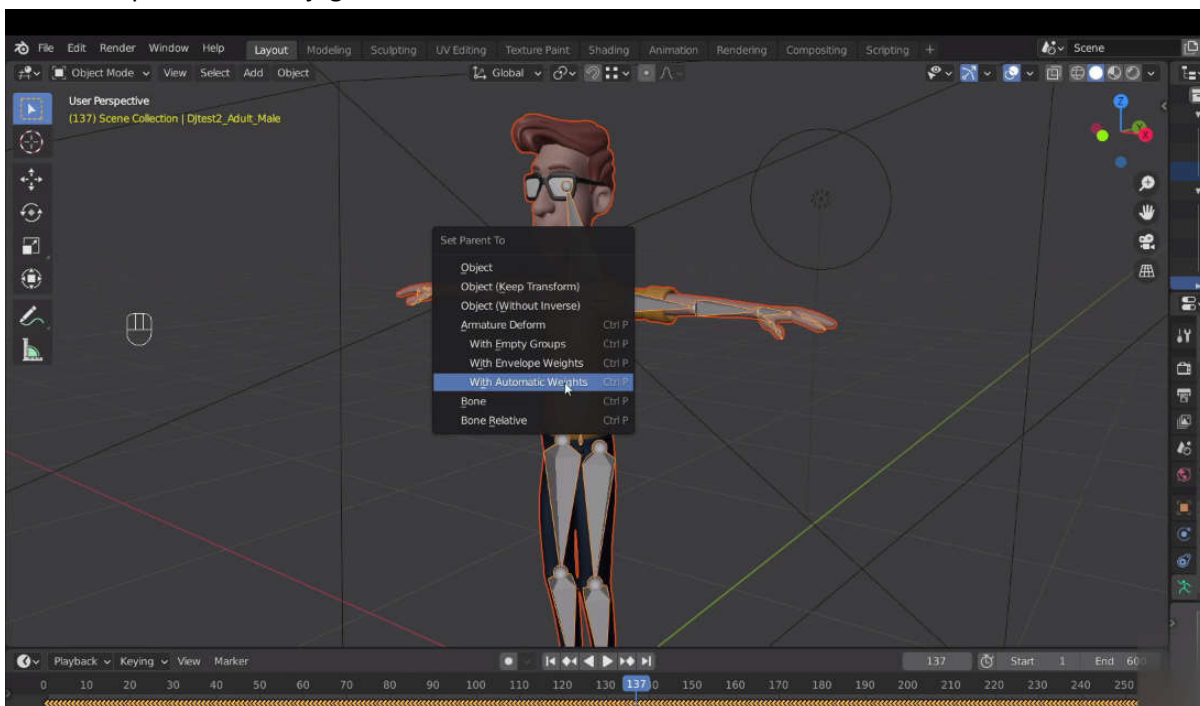


setelah selesai kita dapat memastikan bahwa armature terlihat sepanjang waktu jadi di tampilan

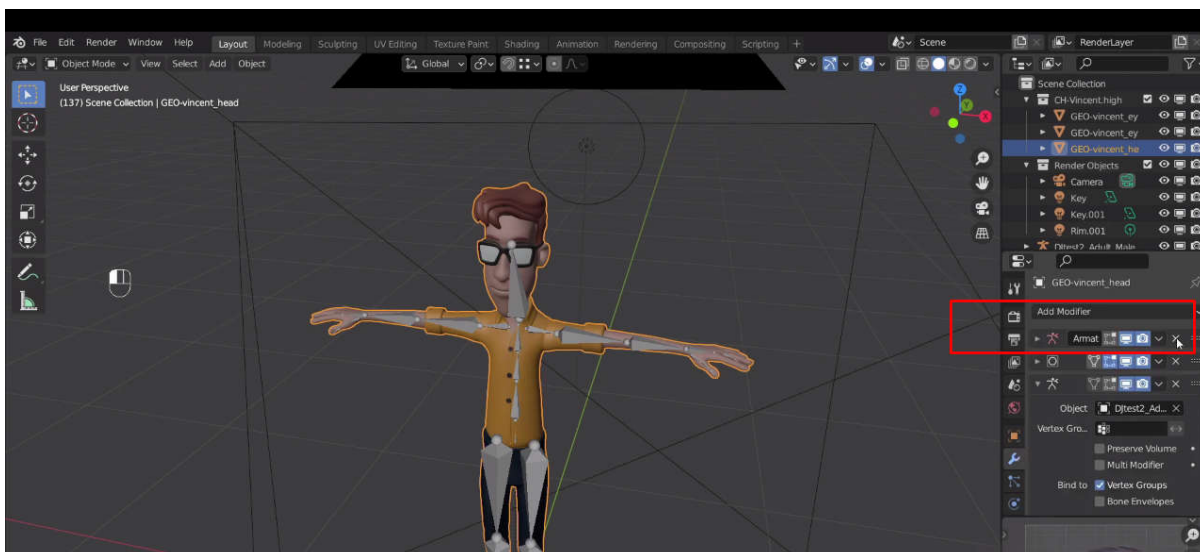
viewport



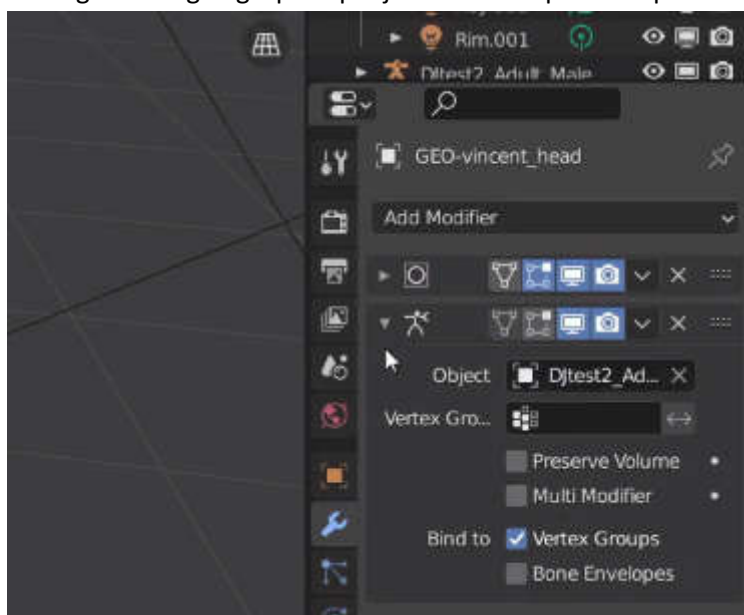
ikat mesh dengan armature dan untuk melakukan itu Anda harus memilih mesh terlebih dahulu dan kemudian pilih armature juga



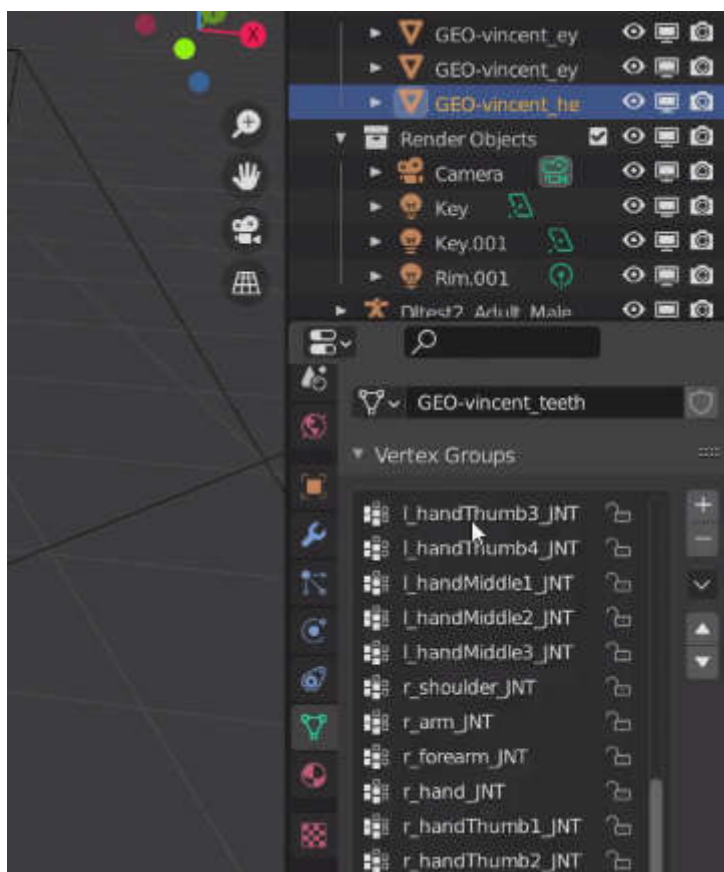
dan pilih deformasi armature dengan with automatic weight apa yang dilakukannya itu memberi mesh pengubah armature oh ini yang lama mari kita singkirkan armature



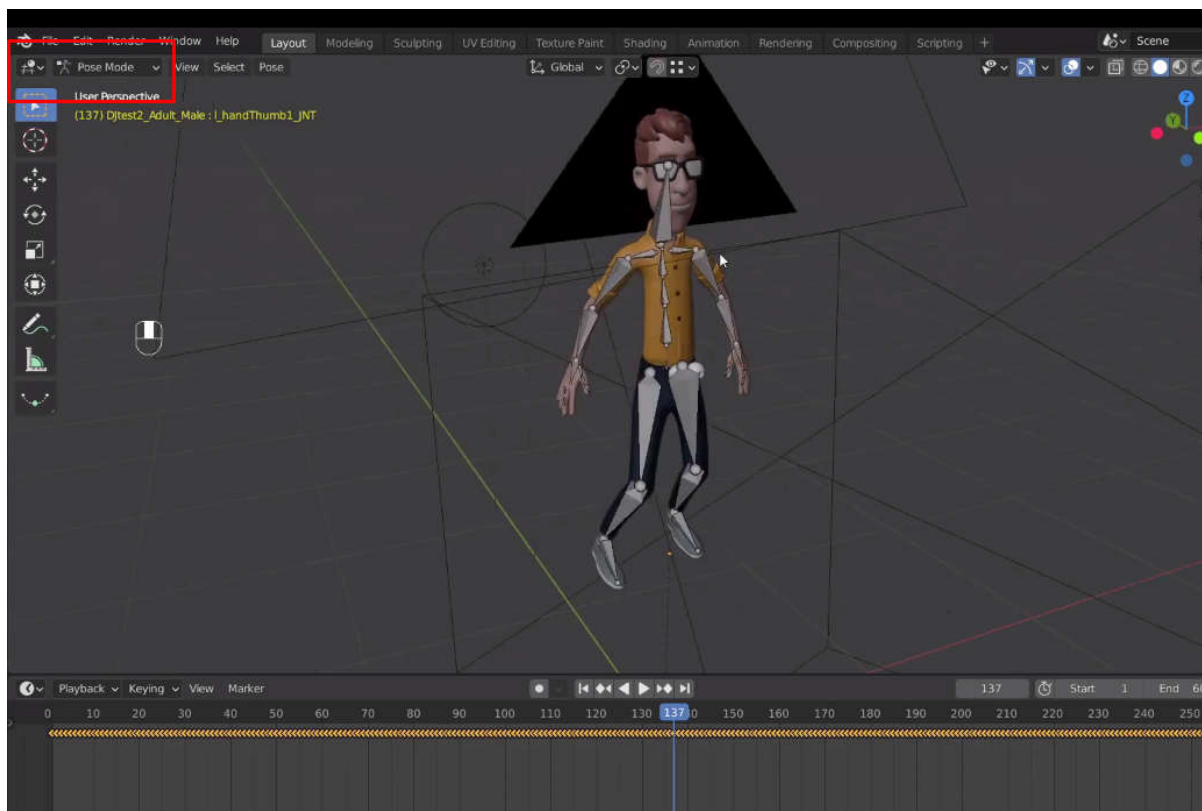
itu mengikat dengan grup simpul jadi mari kita periksa opsi mesh di sini dan grup simpul Anda



dapat melihat ada banyak grup di sana untuk setiap tulang armature sekarang jika Anda memilih



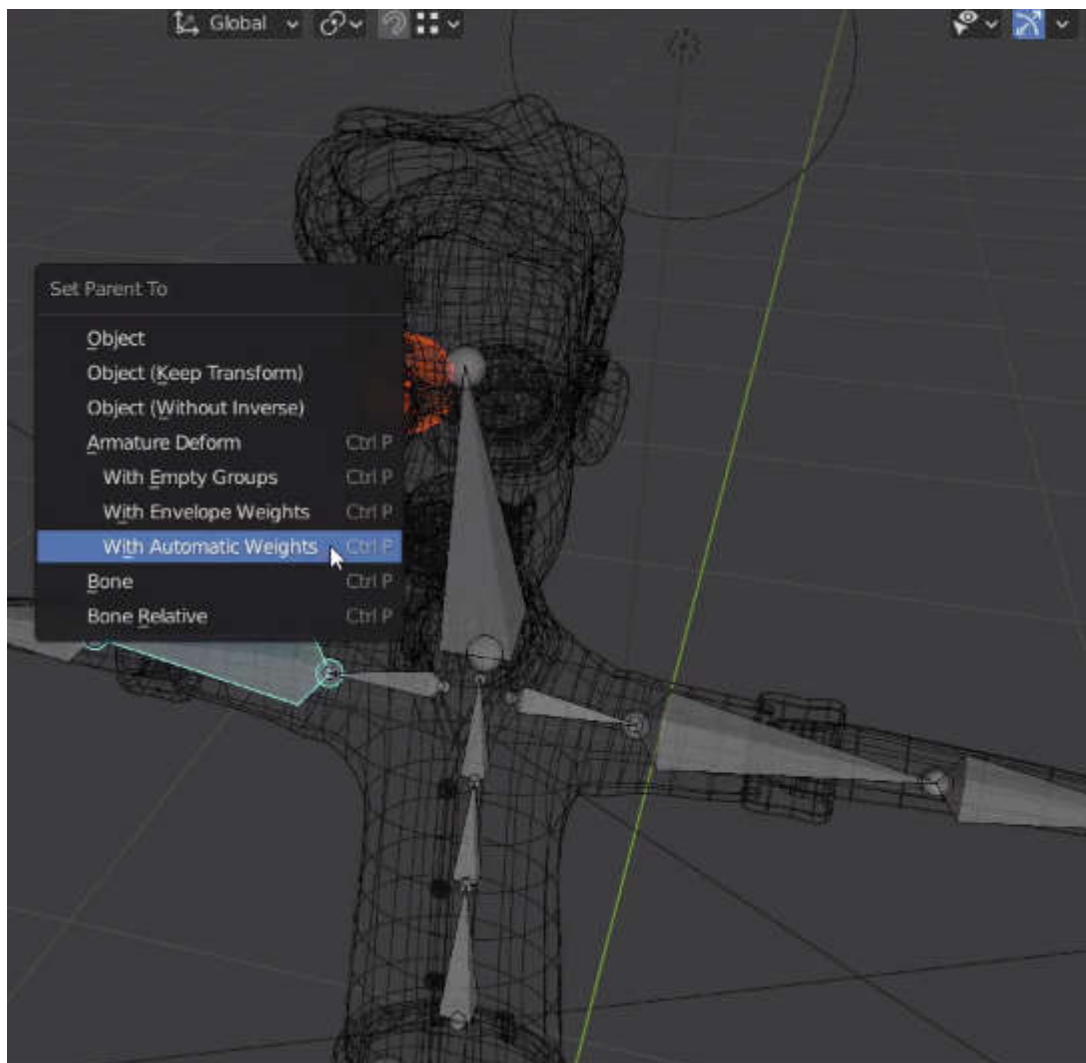
armature dan pergi ke posisi pose



Anda dapat melihat bahwa animasi diterapkan ke armature juga diterapkan pada mesh Anda dapat melihat hanya bola mata yang tersisa

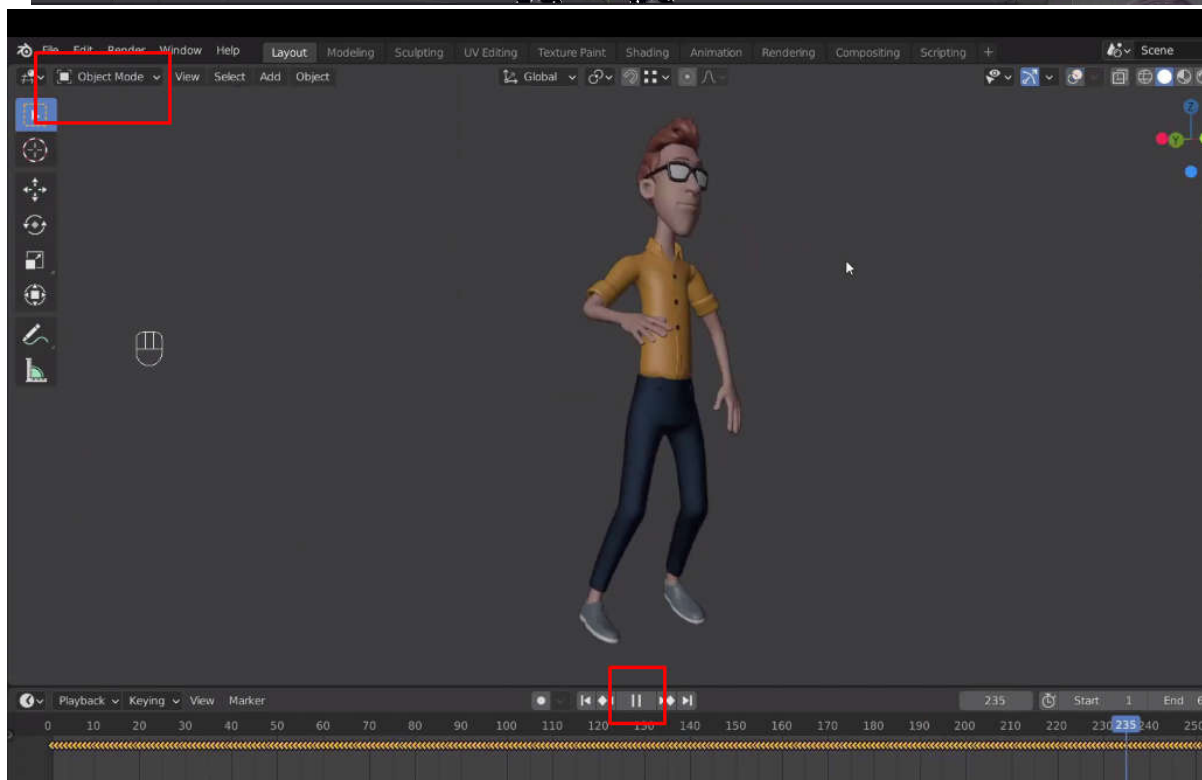
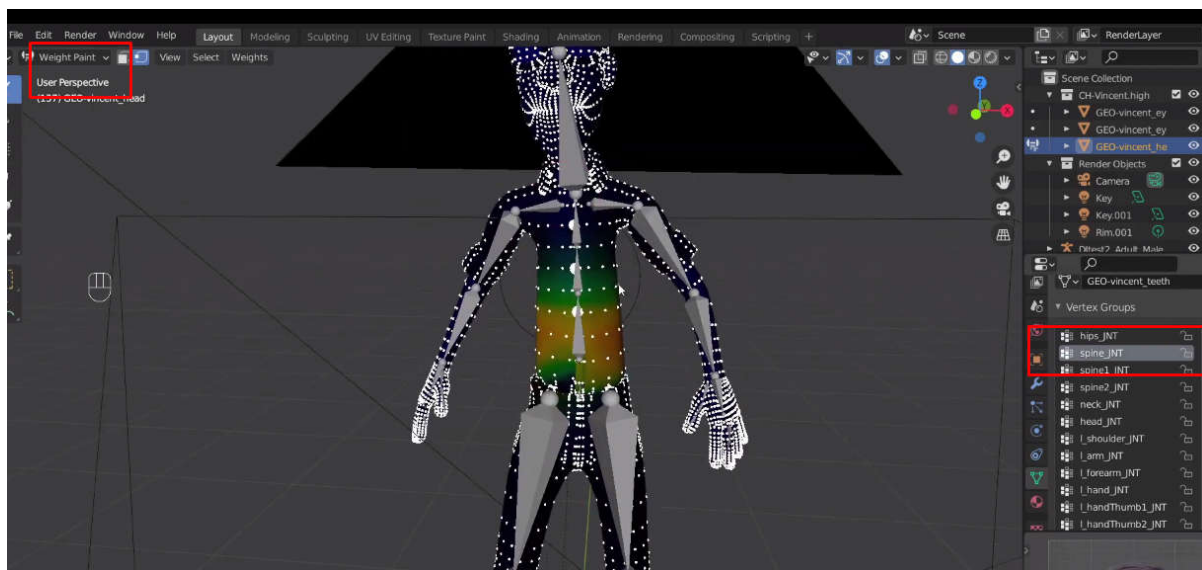


karena mereka adalah mesh yang terpisah jadi mari kita ulangi seluruh proses ini ke bola mata



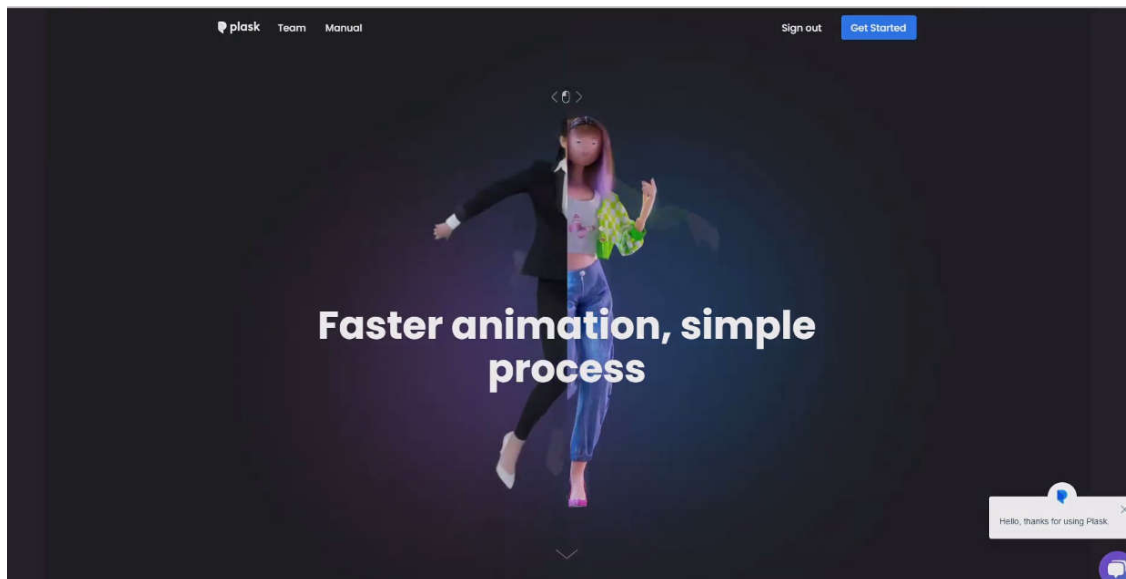
juga hanya mengambil bola mata dan armature dan parenting itu dengan cara otomatis dan bola mata mengikuti armature.

juga jika Anda melihat mesh dalam mode weight paint cukup pilih beberapa grup simpul Anda dapat melihat bahwa bobot sudah ada secara otomatis ditetapkan jadi jika ada yang salah dengan otomatis bobot ini Anda selalu dapat pergi ke mode weight paint dan hanya melakukan perbaikan seperlunya.

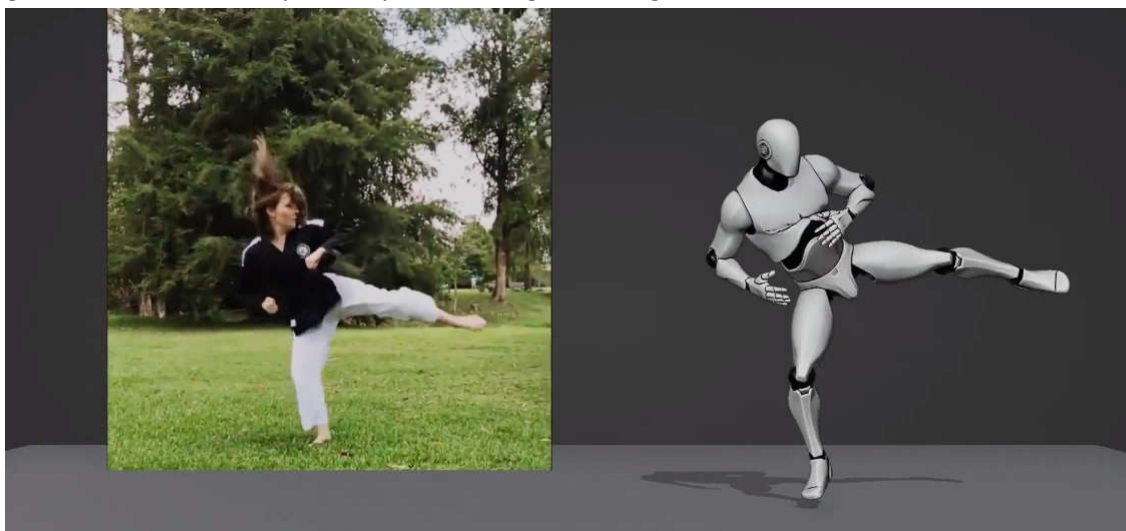


animasi sudah berjalan seperti yang kita inginkan

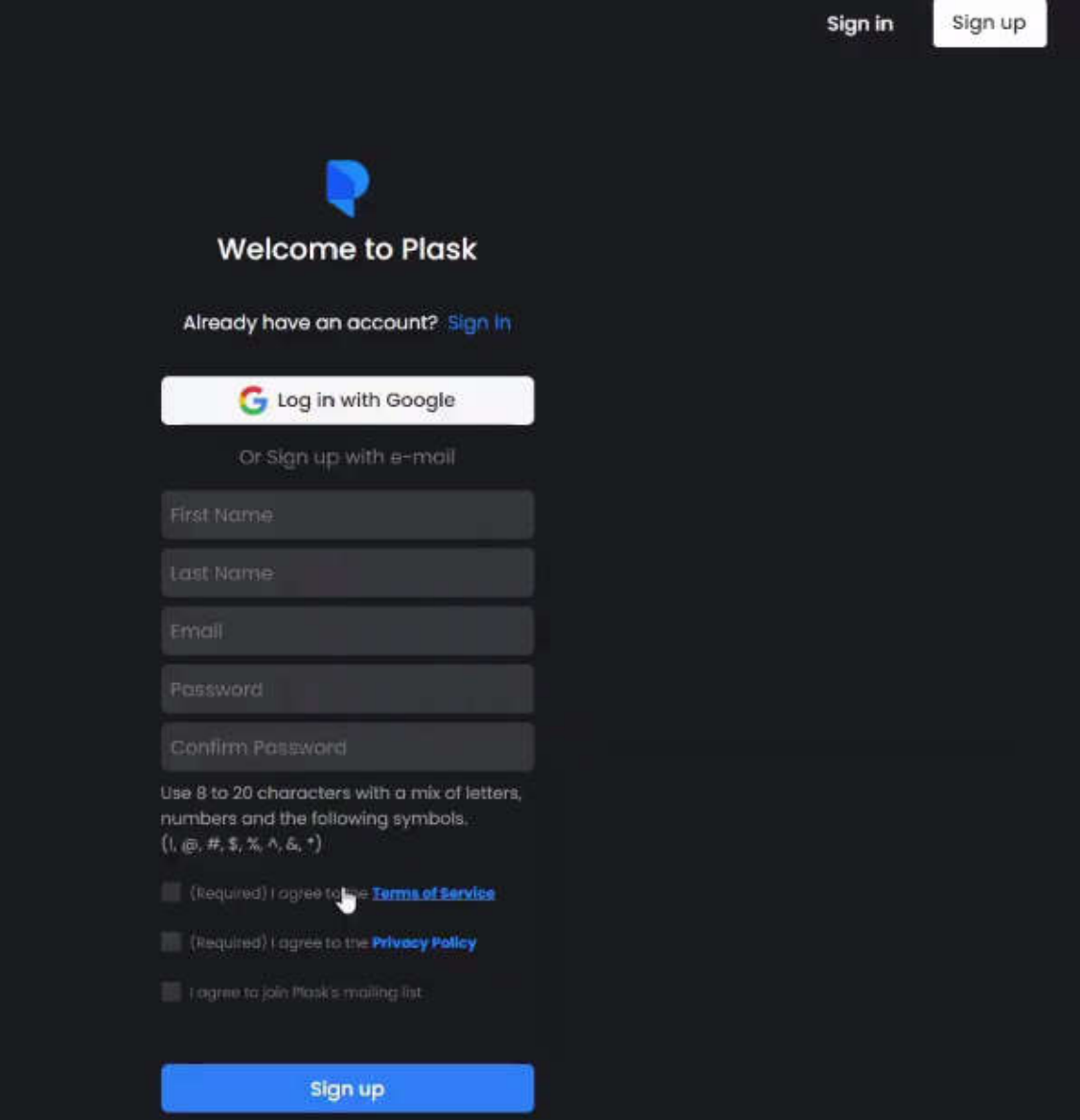
11.3 AI motion capture dengan PLASK.AI



plask.ai ini adalah motion capture yang berbasis ai yang memungkinkan Anda membuat motion capture dari rekaman video sederhana dan gratis. ada aplikasi lain yang dapat mengekstrak gerakan dari video tetapi berbayar atau sangat sulit digunakan



Anda harus pergi ke plask.ai dan Anda harus membuat akun Anda, jadi Anda harus menekan daftar



Sign in Sign up

Welcome to Plask

Already have an account? [Sign in](#)

Log in with Google

Or Sign up with e-mail

First Name

Last Name

Email

Password

Confirm Password

Use 8 to 20 characters with a mix of letters, numbers and the following symbols: (! , @ , # , \$, % , ^ , & , *)

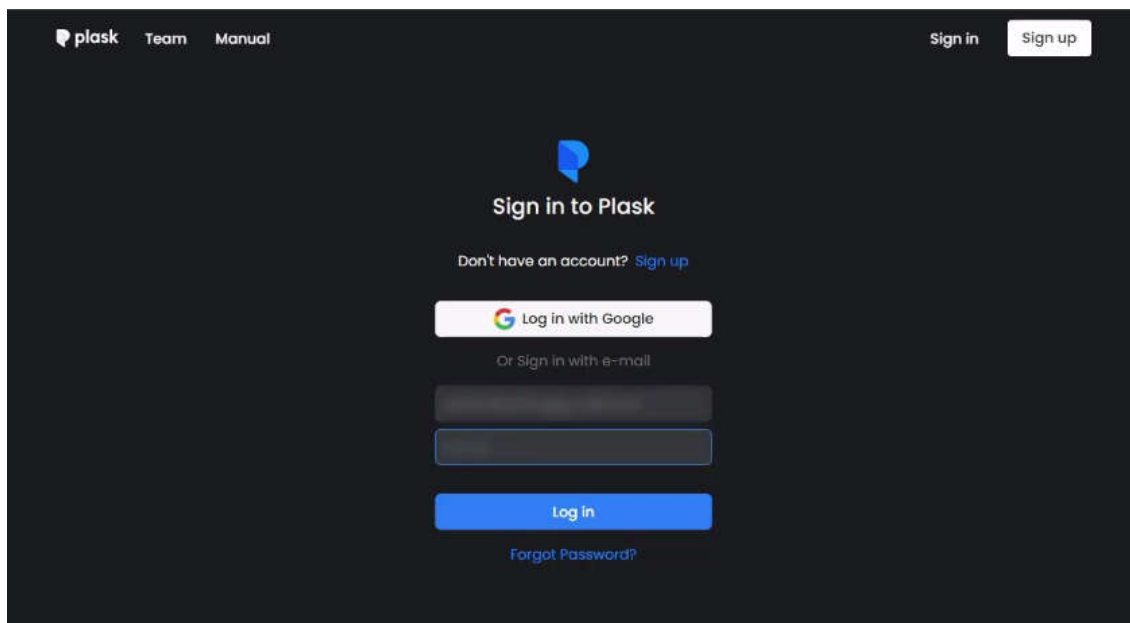
(Required) I agree to the [Terms of Service](#)

(Required) I agree to the [Privacy Policy](#)

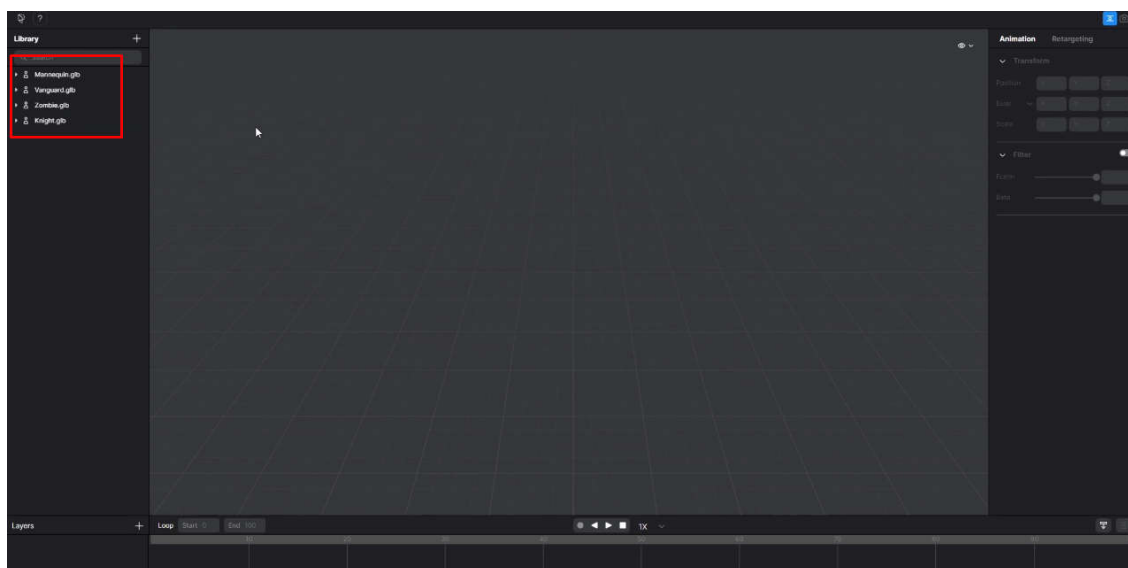
I agree to join Plask's mailing list

Sign up

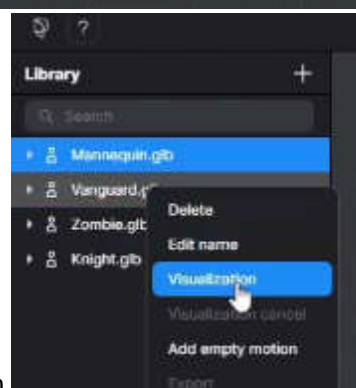
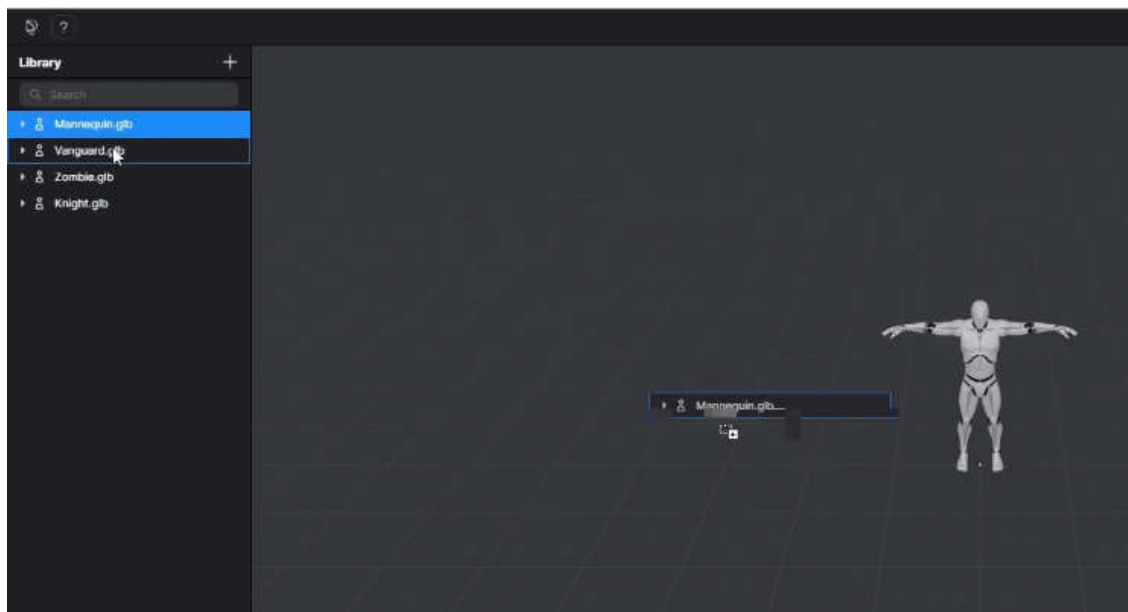
dan
setelah Anda membuat akun, Anda dapat masuk



setelah Anda sign in, Anda akan melihat get started klik dan itu akan membawa Anda ke aplikasi dan akan membutuhkan beberapa saat untuk memuat beberapa konten defaultnya dan aplikasi siap digunakan jadi untuk menampilkan karakter dalam tampilan 3d Anda



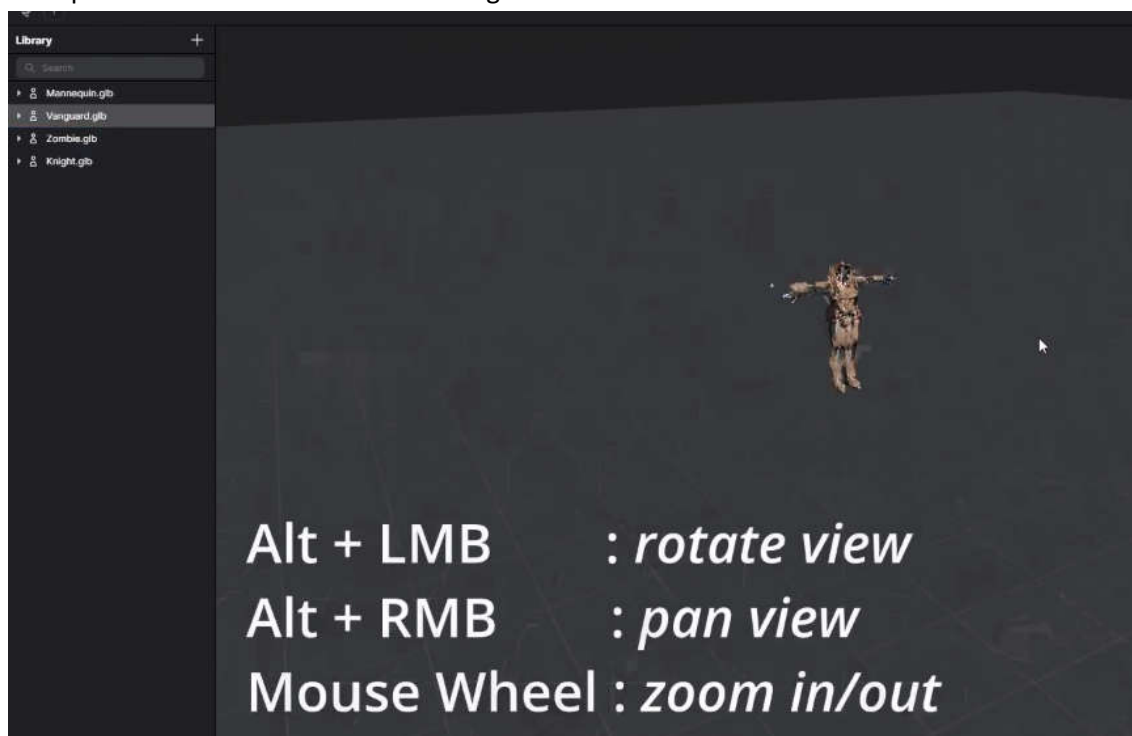
cukup menarik dan melepasnya ke tampilan



atau Anda dapat mengklik kanan dan memilih visualization

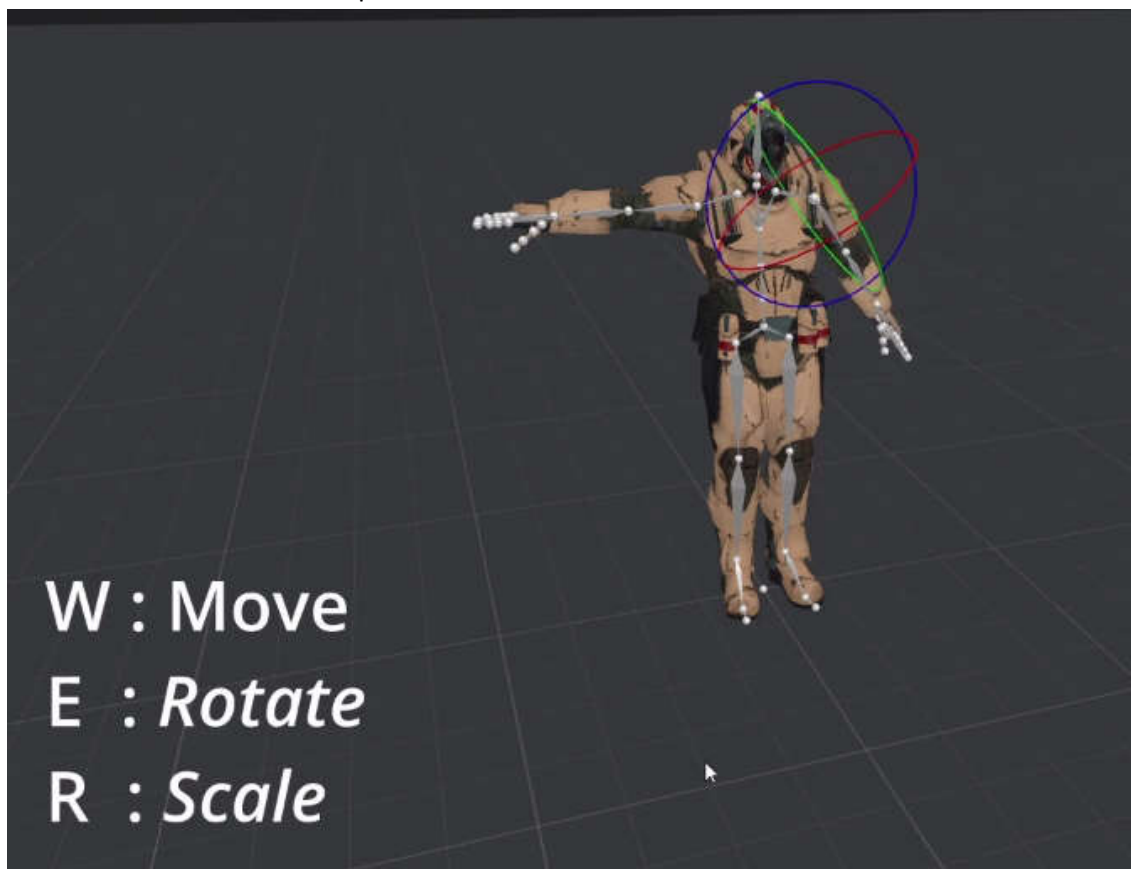
Anda dapat mengorbit sekitar menggunakan tombol alt dan kiri mouse Anda dapat menggeser tampilan menggunakan alt dan kanan tombol mouse dan Anda dapat memperbesar dan

memperkecil dengan roda mouse



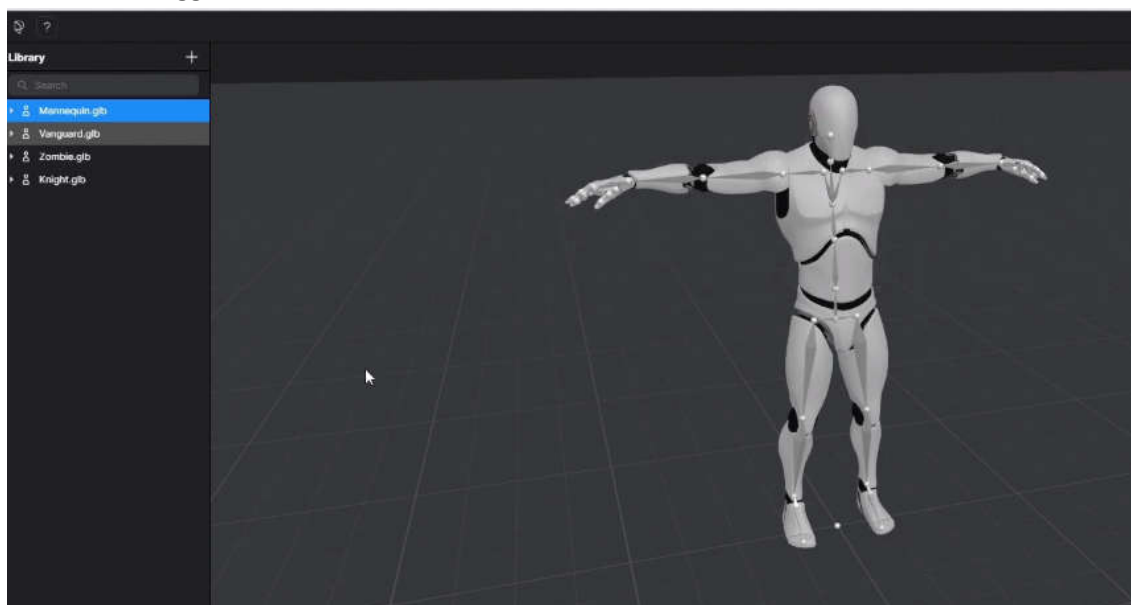
lalu secara teknis Anda dapat memilih joint dan dengan pintasan standar w Anda dapat beralih antara gerakan memutar dan skala dan Anda dapat memanipulasi joint sambungan dan menjeda

karakter dan Anda bahkan dapat



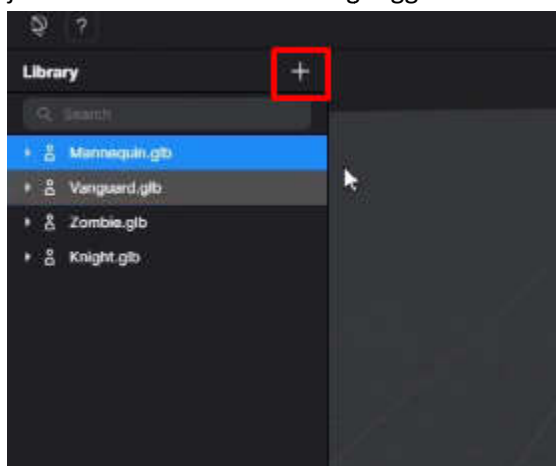
merekam keyframe

kita akan menggunakan manekin

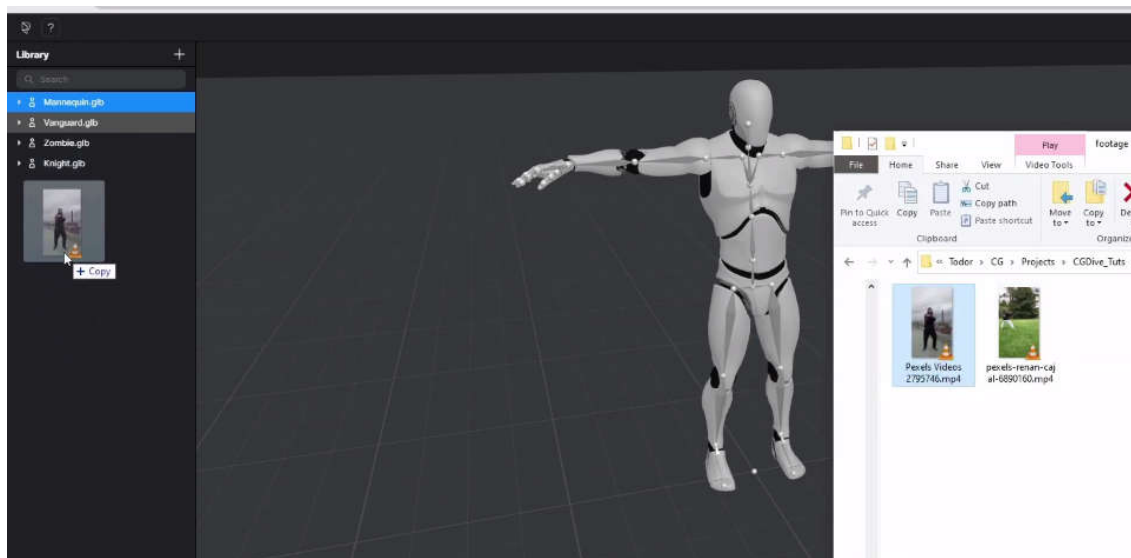


dan dengan itu kita siap untuk coba fitur ai mocap

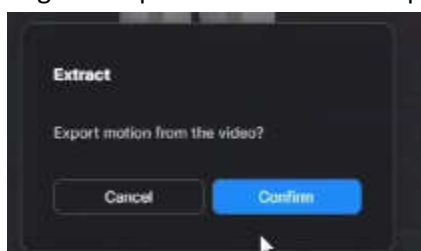
jadi ada dua cara untuk mengunggah video Anda dapat menekan tanda tambah di sini



dan menavigasi ke folder tempat Anda memiliki beberapa clip atau Anda cukup menarik dan melepaskan rekaman ke bagian library

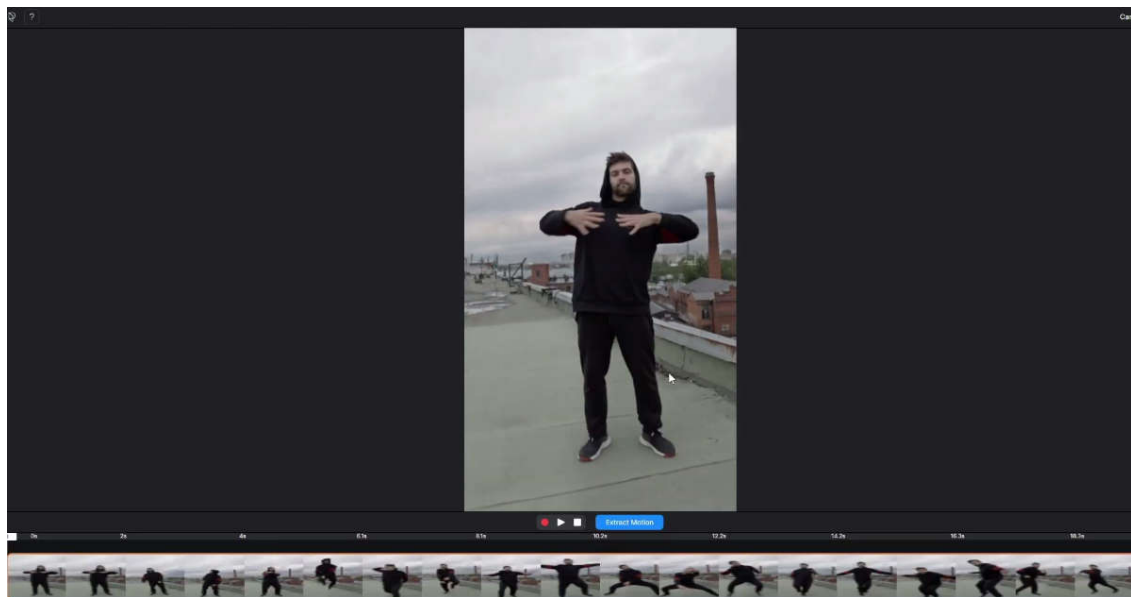


bagaimanapun Anda akan mendapatkan pesan ini

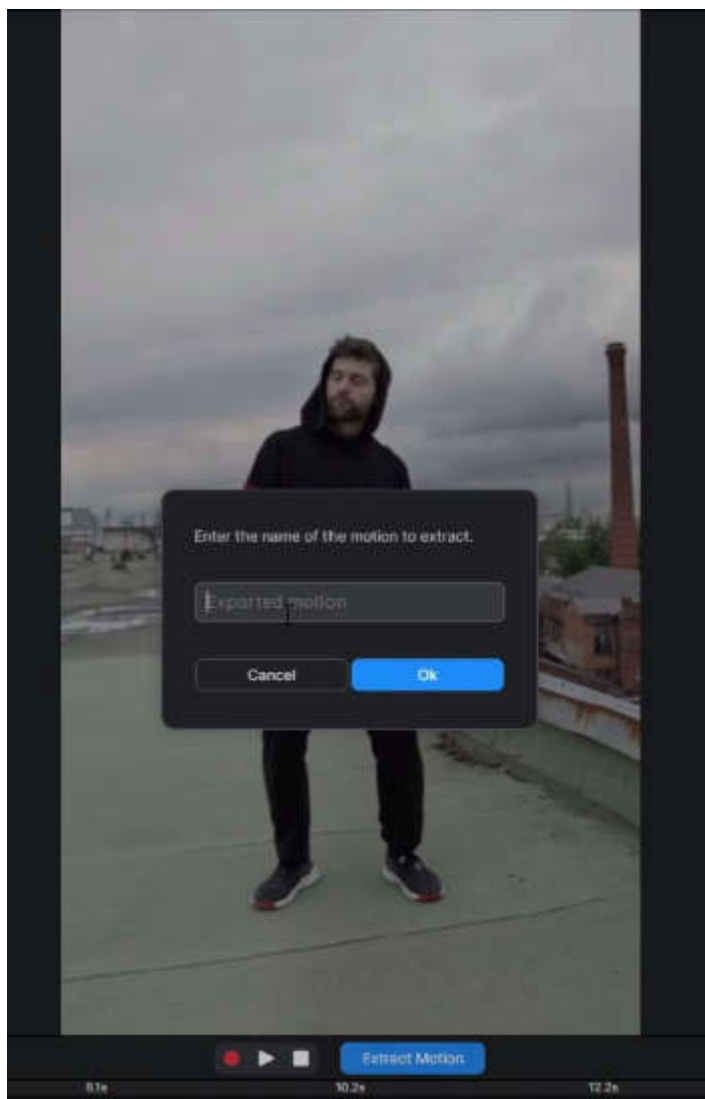


klik konfirm

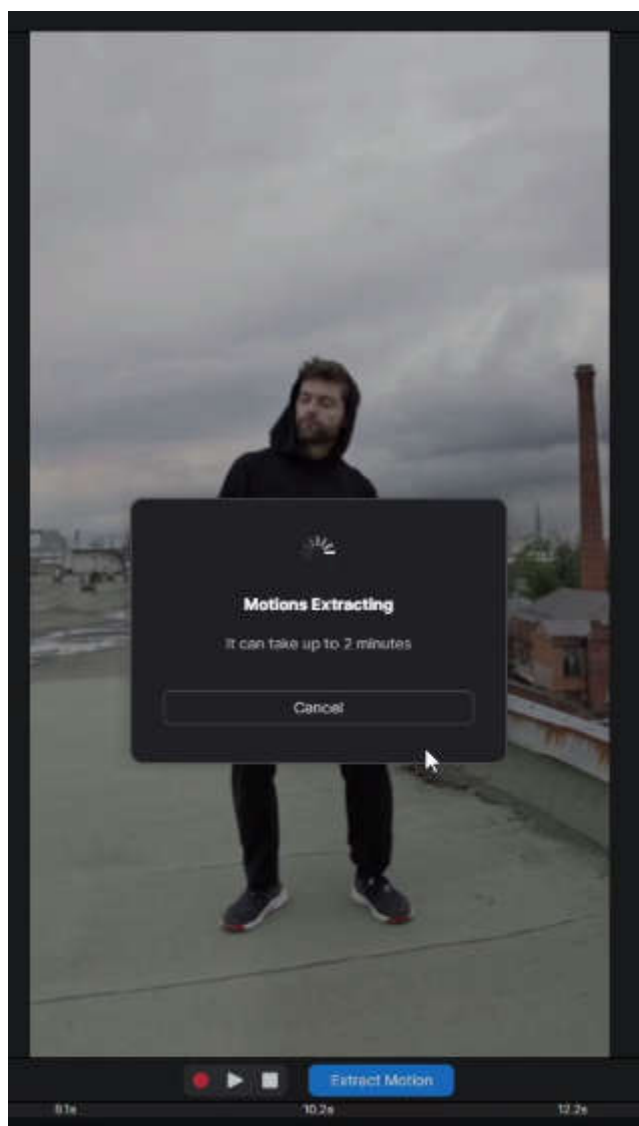
plask.ai akan dengan cepat memproses video dan dari sini prosesnya hampir otomatis Anda dapat memutar rekamannya



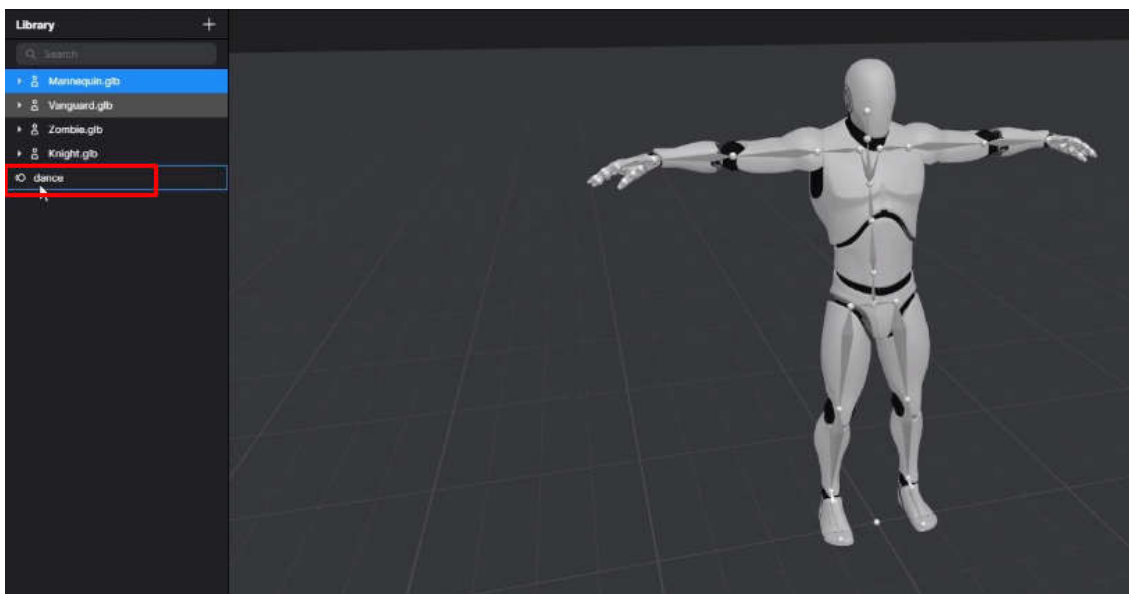
lihat apa yang Anda miliki dan hal penting lainnya yang dapat Anda adalah menentukan awal dan akhir dari tindakan yang Anda ingin lakukan cukup seret awal akhir seperti ini dan ketika Anda puas dengan



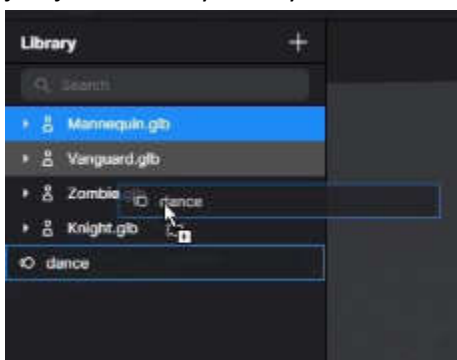
cukup klik ekstrak motion dan tekan ok lalu Anda akan memiliki untuk menunggu sementara plak memproses gerakan jelas semakin lama gerakan semakin lama waktu yang dibutuhkan saat plask.ai melakukan



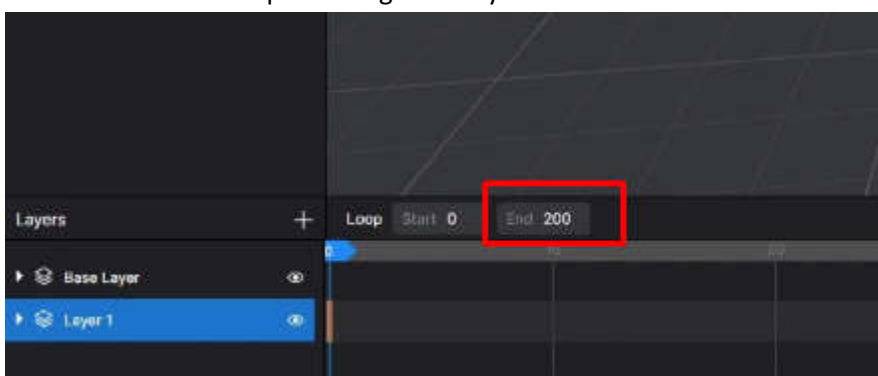
ketika selesai Anda akan kembali ke antarmuka awal dalam gerakan yang diekstraksi akan ada di library Anda



jadi jika kita hanya menyeret dan melepaskan gerakan



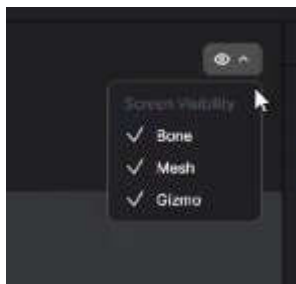
ke manekin tindakan akan diterapkan jika Anda ingin menggunakan tindakan pada cukup seret dan lepas dan itu akan benar-benar muncul di viewport dan kemudian jika kita ingin beralih kembali ke awal kita dapat memperluas tindakan yang dimiliki. dan klik kanan pada dance dan kita akan memilih visualization kita dapat mengedit keyframe akhir karena ini lebih panjang dari 100 frame



jadi inilah kualitas yang kita miliki
gerakan mocap akan bergantung pada banyak faktor.

plask.ai menghasilkan hasil yang serupa jika tidak lebih unggul, hasilnya memang tidak sempurna. yang Anda akan lakukan dengan plask.ai mengimpor beberapa video dan mengubahnya menjadi data mocap.

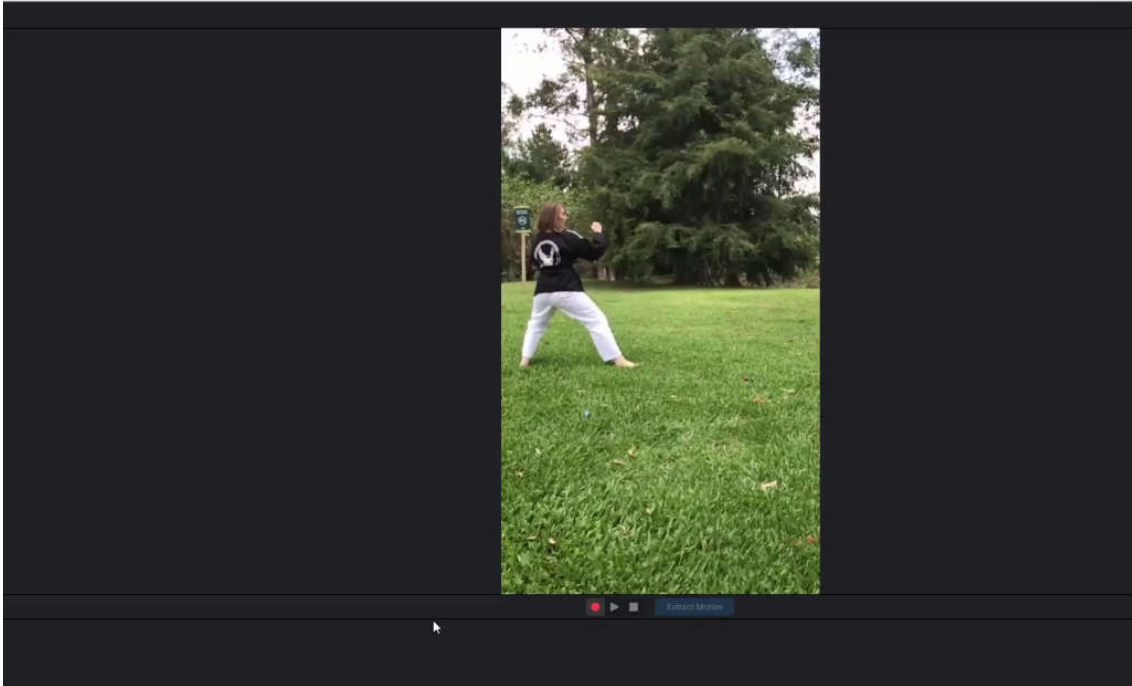
aplikasi ini sebenarnya memiliki lebih banyak fitur bahkan memiliki lapisan animasi di sini dan tampaknya berfungsi kita dapat menambahkan satu lagi misalnya dapat mengedit tindakan dapat memasukkan keyframe tetapi secara keseluruhan fitur animasi dan Anda tidak dapat menambahkan ik misalnya, tidak ada fungsi undo yang mungkin Anda perhatikan di antarmuka adalah Anda bisa nonaktifkan tampilan tulang mesh dan gizmos



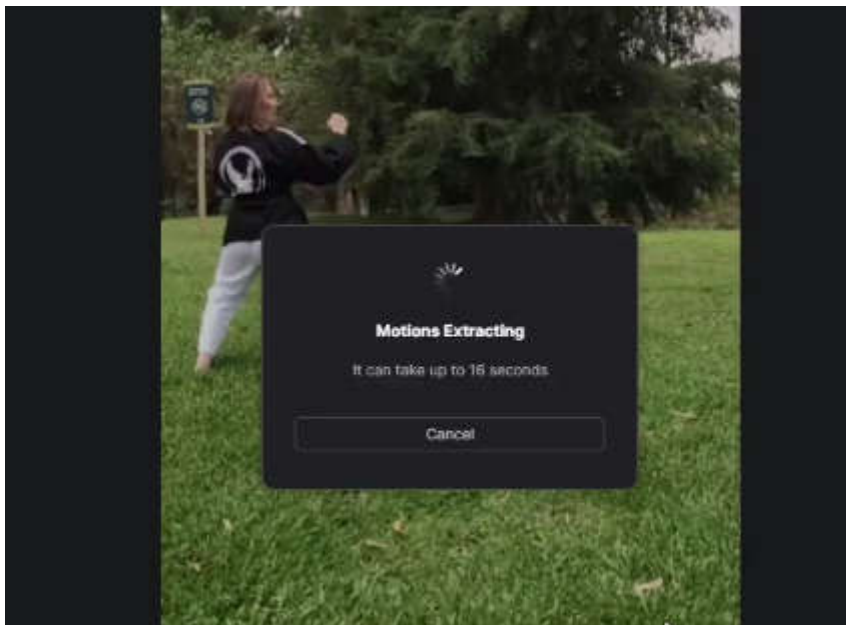
plask.ai memiliki timeline sederhana kita dapat memainkan kita dapat memutarinya secara terbalik sesuatu hal lain yang perlu diingat adalah bahwa Anda hanya dapat memiliki satu karakter di dalam status aplikasi saat ini jadi jika kita menggunakan zombie, manekin akan berganti zombie jika kita membawa mereka kembali manekin zombie akan pergi jadi jangan berharap untuk menganimasikan adegan kompleks di dalam plask.ai

Dan juga sangat penting untuk mengetahui bahwa plask ai tidak akan menyelamatkan Anda file jadi jika kita muat ulang halaman ini aksi tarian ini yang kita ekstrak akan hilang jadi kita tidak akan memuat ulang aplikasi ,jadi harus diingat hanya memuat ulang aplikasi setelah Anda memiliki beberapa data mocap Anda.

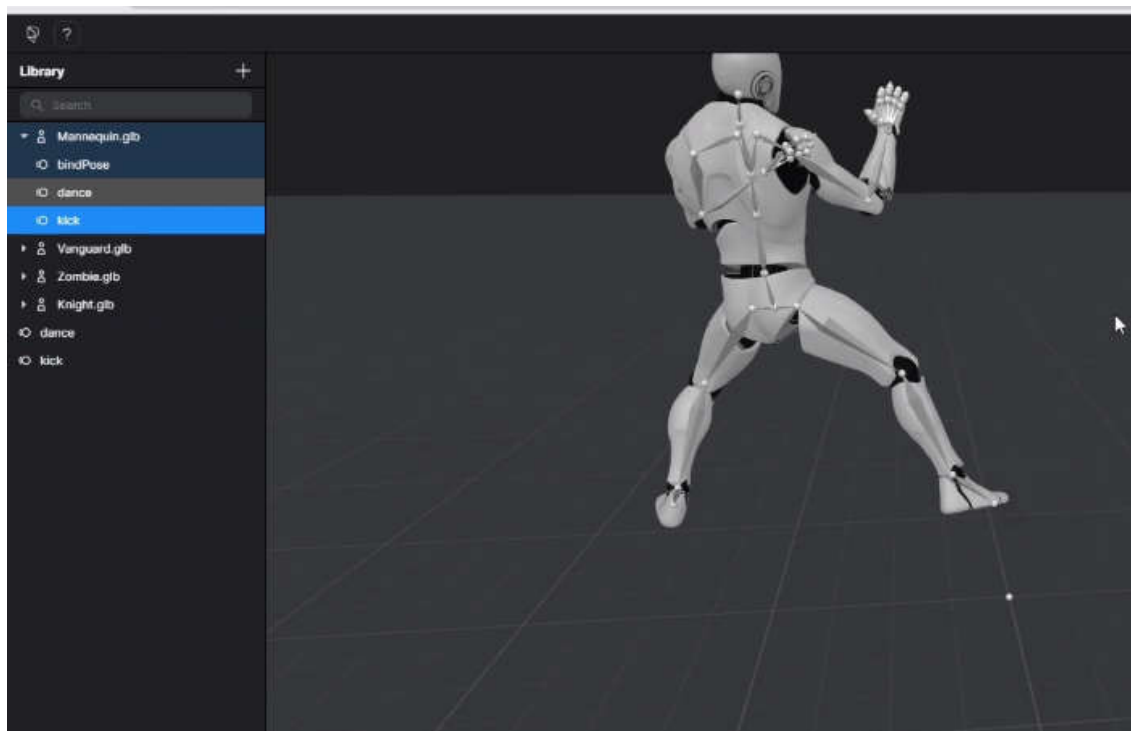
mungkin ingin mengekspornya dari plask.ai ke 3d lain hingga mesin game atau apa pun yang Anda sukai dan Anda dapat mengekspor dengan bebas menggunakan fbx ,gltf dan bahkan bvh untuk blender yang kita akan, kita menguji fpx dan gltf dan keduanya baik-baik saja. gltf mungkin sebenarnya memiliki beberapa keuntungan tetapi kita akan tetap menggunakan fbx di sini jadi untuk membuat hal-hal sedikit lebih menarik, kita hanya akan mengekstrak satu tindakan lagi dan lalu kita akan mengekspor serta melakukannya dengan sangat cepat



dan inilah tindakan tendangan baru kita



sampai ekspor karakter dan animasi

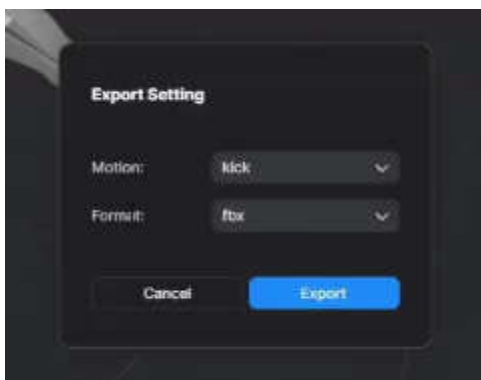


Anda dapat mengklik kanan pada karakter

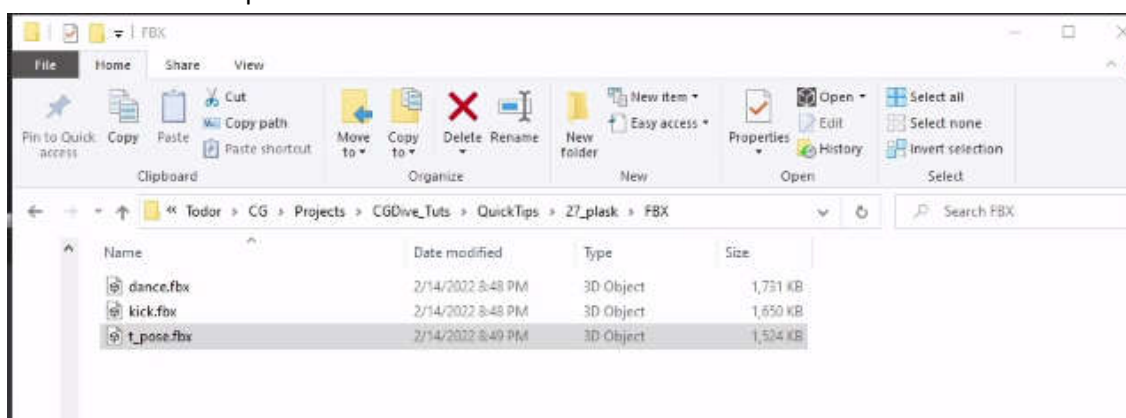


pilih ekspor

pilih gerakan dalam format file yang Anda sukai

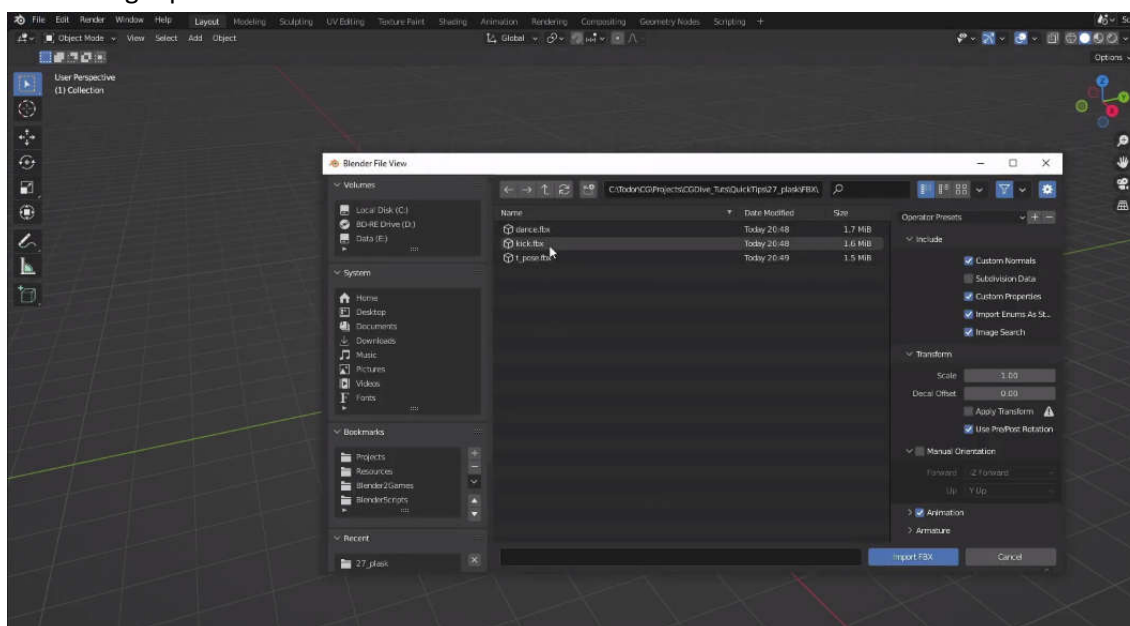


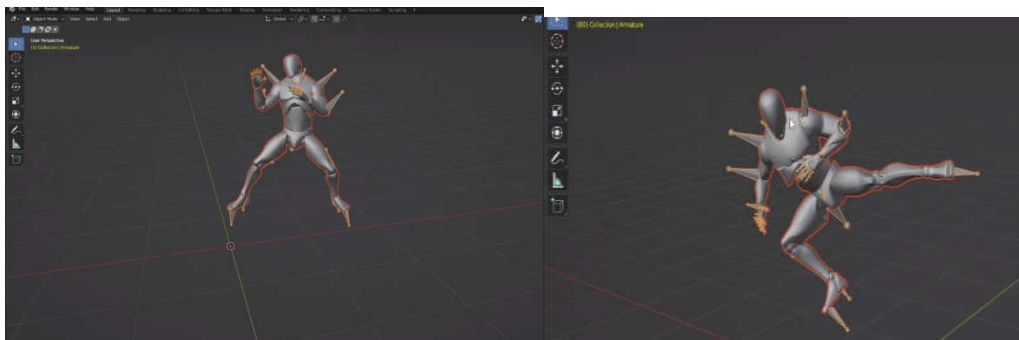
dan klik ekspor juga mengekspor aksi dance dan juga akan mengekspor bind pose. bind pose adalah t-pose pada dasarnya dan Anda mungkin memerlukannya, untuk blender, kita akan membutuhkan itu file akan memiliki penamaan sederhana ini



jadi sekarang Anda dapat melanjutkan untuk mengimpor file ke aplikasi 3d favorit Anda

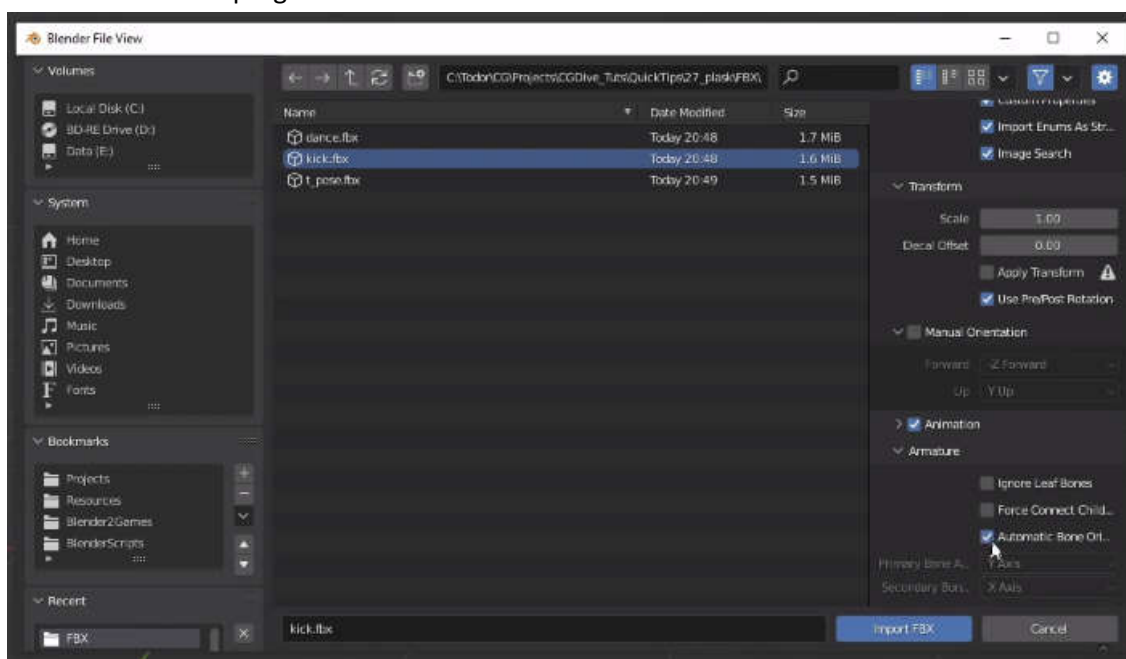
kita akan menunjukkan cara melakukannya di blender di sini kita menggunakan blender 3.01 dan kita akan mengimpor fbx





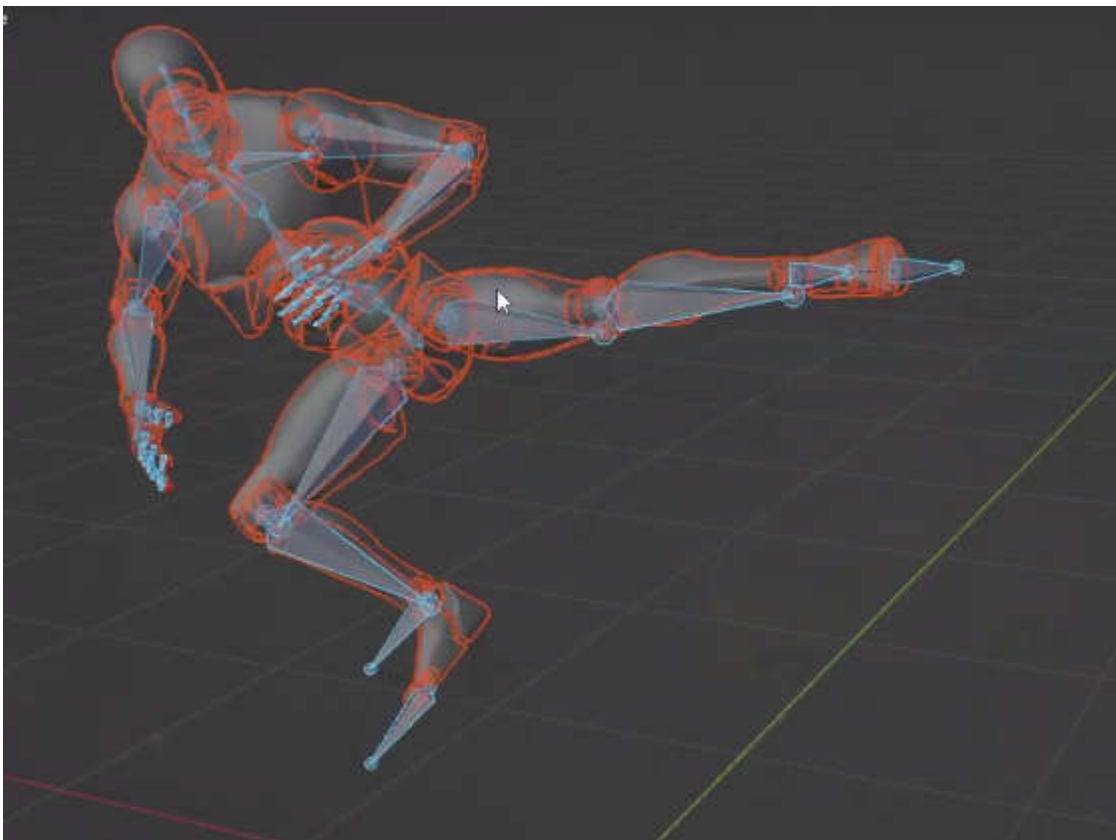
dan sekarang jika kita mengimpor salah satu tindakan dengan pengaturan default anda akan melihat bahwa itu berfungsi kecuali bahwa semua agak berorientasi aneh yang mana adalah masalah umum saat mengimpor file di blender

dan kali ini kita akan pergi ke armature dan

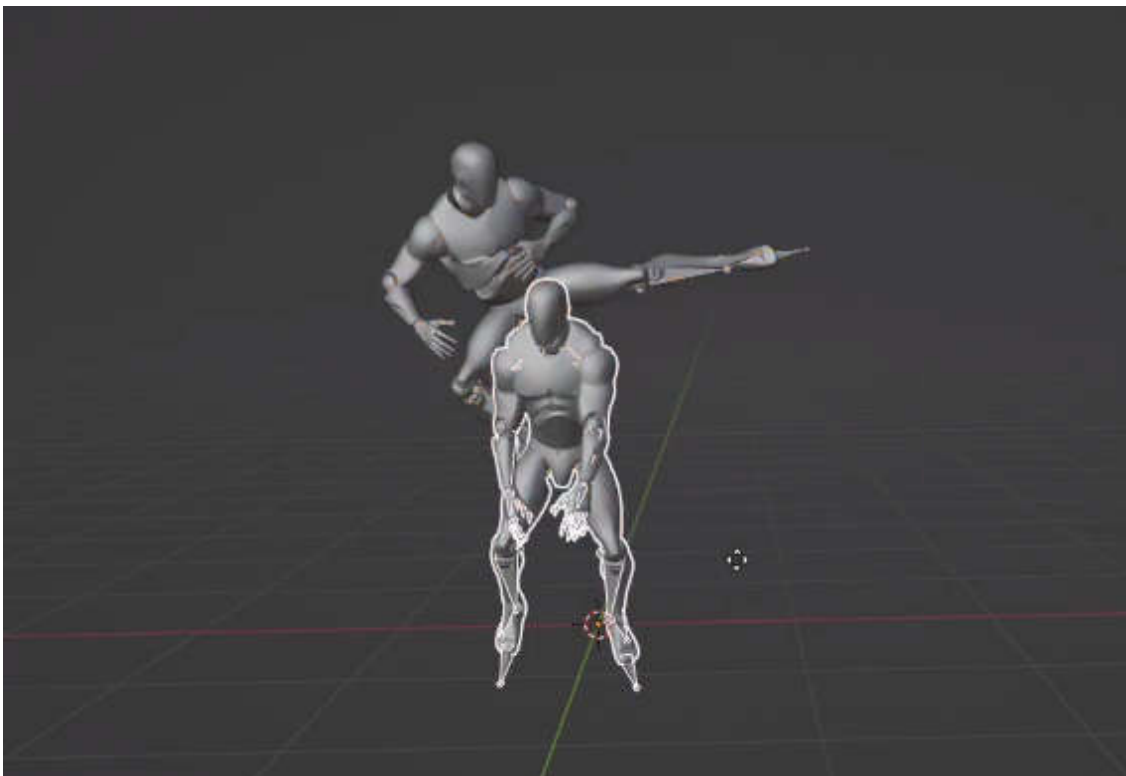


memilih orientasi tulang otomatis dan mengimpor fbx

jadi sekarang kita memiliki tindakan yang sama tetapi tulang kita lebih berorientasi



oke jadi kita dapat mengimpor aksi dance dengan cara yang sama

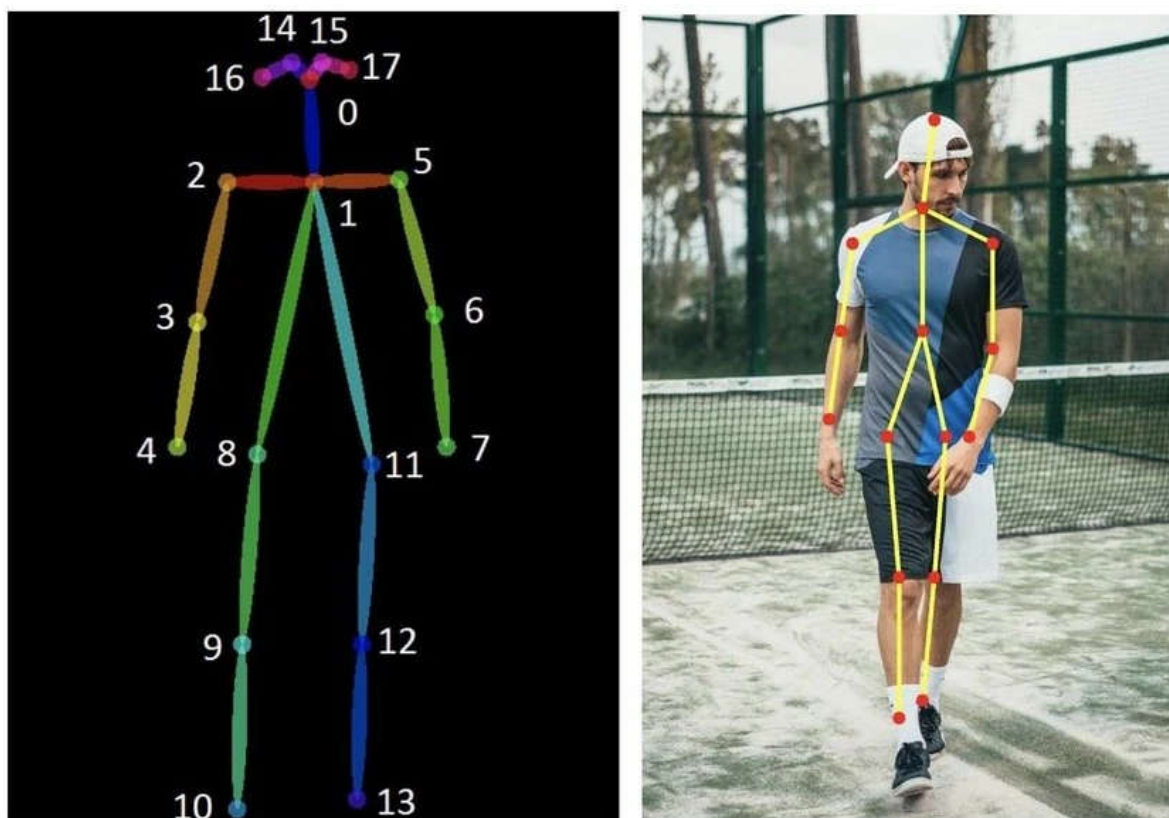


dan jika memiliki karakter ini dalam blender dengan animasi ini adalah semua yang Anda inginkan maka Anda sudah siap

Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

11.4 Motion capture dengan OpenPose

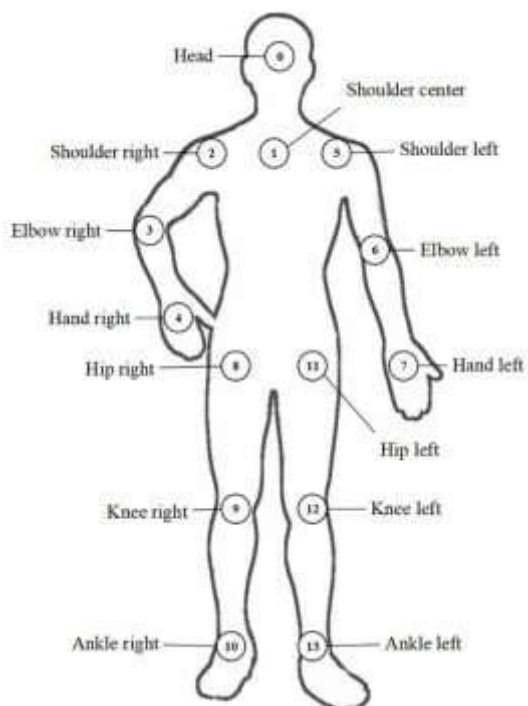
Apa itu OpenPose? OpenPose adalah perpustakaan deteksi pose manusia multi-orang secara real-time yang untuk pertama kalinya menunjukkan kemampuan untuk bersama-sama mendeteksi titik kunci tubuh, kaki, tangan, dan wajah manusia pada satu gambar. OpenPose mampu mendeteksi total 135 titik kunci. Metode ini menjadi pemenang COCO 2016 Keypoints Challenge dan populer karena kualitasnya dalam deteksi multi-orang.



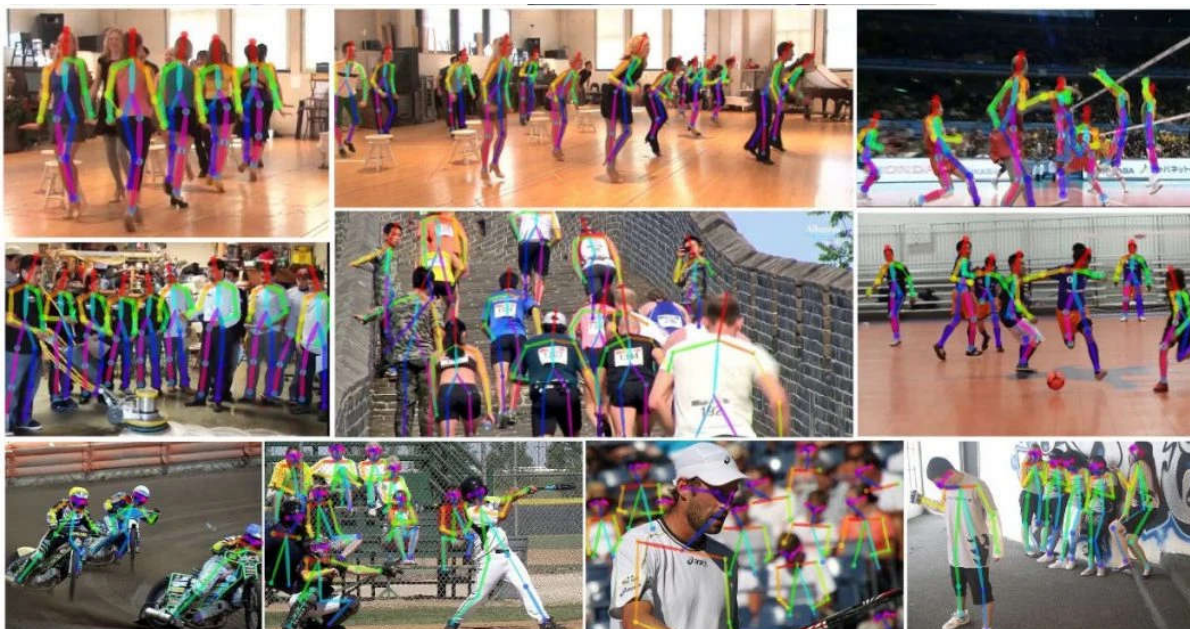
Teknik OpenPose diciptakan oleh Ginés Hidalgo, Yaser Sheikh, Zhe Cao, Yaadhav Raaj, Tomas Simon, Hanbyul Joo, dan Shih-En Wei. Namun, dikelola oleh Yaadhav Raaj dan Ginés Hidalgo.

Estimasi Pose dengan OpenPose

Kerangka pose manusia menunjukkan orientasi individu dalam format tertentu. Pada dasarnya, ini adalah sekumpulan titik data yang dapat dihubungkan untuk mendeskripsikan pose seseorang. Setiap titik data dalam kerangka juga bisa disebut bagian atau koordinat, atau titik. Koneksi yang relevan antara dua koordinat dikenal sebagai tungkai atau pasangan.



Namun, penting untuk dicatat bahwa tidak semua kombinasi titik data memunculkan pasangan yang relevan. Titik Kunci Pose Manusia Mengetahui orientasi seseorang membuka jalan bagi banyak aplikasi kehidupan nyata, banyak di antaranya dalam olahraga dan kebugaran. Banyak pendekatan estimasi pose manusia telah diusulkan selama bertahun-tahun. Teknik pertama biasanya memperkirakan pose satu individu dalam gambar yang terdiri dari satu orang. OpenPose memberikan pendekatan yang lebih efisien dan kuat yang memungkinkan penerapan estimasi pose pada gambar dengan pemandangan yang ramai.



Apa saja fitur OpenPose? library deteksi pose manusia OpenPose memiliki banyak fitur tetapi yang diberikan di bawah ini adalah beberapa yang paling luar biasa:

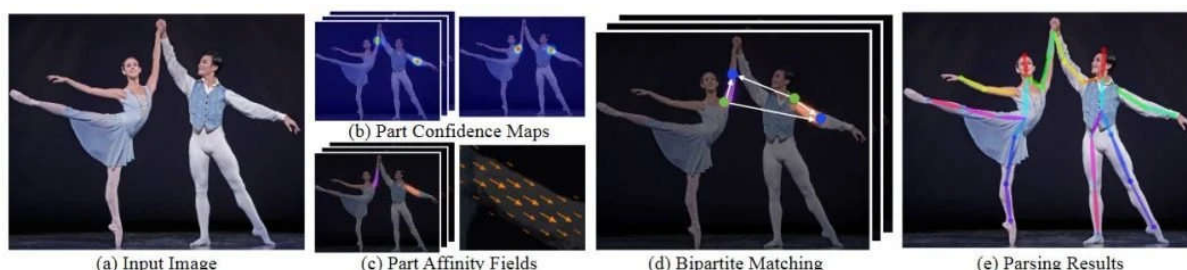
- Deteksi titik kunci orang tunggal 3D waktu nyata
- Triangulasi 3D dengan beberapa tampilan
- kamera Kompatibilitas kamera Flir
- Deteksi titik kunci multi orang 2D waktu nyata
- 15, 18 , 27-keypoint estimasi keypoint tubuh/kaki
- 21 estimasi keypoint tangan
- 70 estimasi keypoint wajah
- Pelacakan satu orang untuk mempercepat deteksi dan penghalusan visual
- Kotak peralatan kalibrasi untuk estimasi parameter kamera ekstrinsik, intrinsik, dan distorsi

Hardware dan Kamera

OpenPose mendukung input video dari aliran gambar, video, dan kamera webcam, kamera Flir/Point Grey, kamera IP (CCTV), dan sumber masukan khusus (seperti kamera kedalaman, kamera lensa stereo, dll.) Dari segi perangkat keras, OpenPose mendukung versi yang berbeda untuk Nvidia Komputasi GPU (CUDA), AMD GPU (OpenCL), dan non-GPU (CPU). Itu dapat dijalankan di Ubuntu, Windows, Mac, dan Nvidia Jetson TX2.

Cara Kerja OpenPose

Library OpenPose awalnya mengeluarkan fitur dari gambar menggunakan beberapa lapisan pertama. Fitur yang diekstraksi kemudian dimasukkan ke dalam dua divisi paralel lapisan jaringan konvolusional. Divisi pertama memprediksi satu set 18 peta kepercayaan — dengan masing-masing menunjukkan bagian tertentu dari kerangka pose manusia. Cabang berikutnya memprediksi kumpulan lain dari 38 Part Affinity Fields (PAFs) yang menunjukkan tingkat asosiasi antar bagian. Tahap selanjutnya digunakan untuk membersihkan prediksi yang dibuat oleh cabang. Dengan bantuan peta kepercayaan, grafik bipartit dibuat antara pasangan bagian. Melalui nilai PAF, tautan yang lebih lemah dipangkas dalam grafik bipartit. Sekarang, dengan menerapkan semua langkah yang diberikan, kerangka pose manusia dapat diperkirakan dan dialokasikan ke setiap orang dalam gambar.



Ikhtisar Pipeline

- a) seluruh gambar sebagai input
- b) CNN dua cabang untuk bersama-sama memprediksi peta kepercayaan untuk deteksi bagian tubuh
- c) memperkirakan bidang afinitas bagian untuk asosiasi bagian d) set pencocokan bipartit untuk mengasosiasikan kandidat bagian tubuh

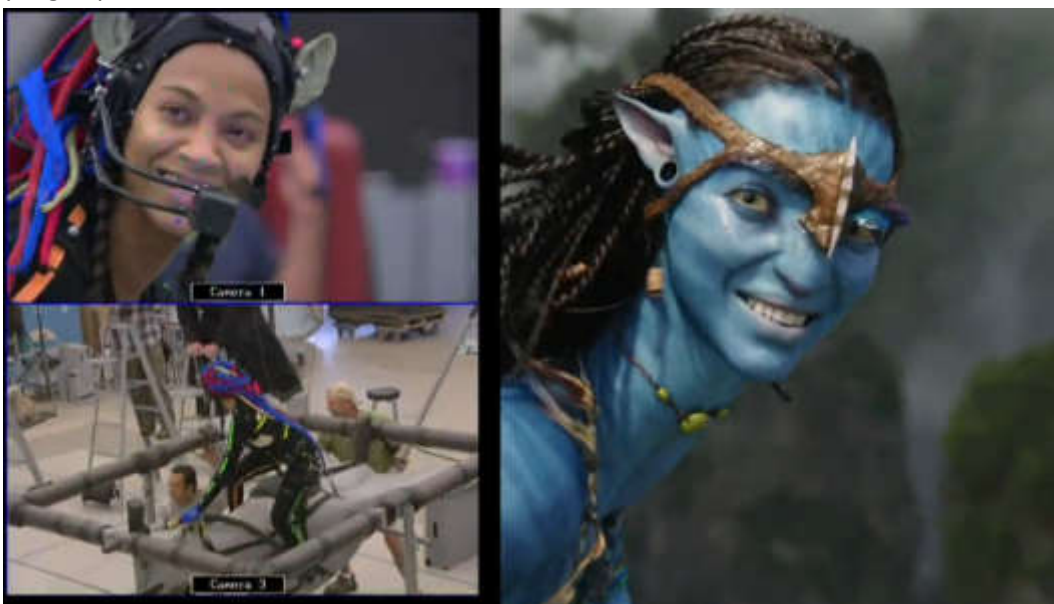
e) menyusunnya menjadi pose seluruh tubuh untuk semua orang dalam gambar

openpose merupakan program yang masih dalam tahap pengembangan, dan untuk menjalankannya dibutuhkan script yang membutuhkan kemampuan pemrograman , jadi tidak dibahas lebih lanjut dalam buku ini. hanya pada penggunaannya dalam facial motion capture yang akan dibahas pada bab selanjutnya.

bab 12 proses facial motion capture

12.1 facial motion capture

facial motion capture adalah proses penangkapan gerakan wajah untuk mendapatkan data mocap yang dapat dimanfaatkan dalam animasi karakter 3D .



ada banyak teknologi yang mampu menangkap facial motion capture dari yang peralatan khusus sampai dengan teknologi yang seadanya.

dalam bagian ini akan dibahas proses motion capture yang cukup memanfaatkan teknologi yang ada dengan menggunakan dua metode dengan blender saja dan menggunakan pemanfaatan openpose dan blender.

12.1 facial motion capture dengan blender saja

Metode ini ada cukup banyak , ada yang berupa plug in dan berbayar tapi yang kita bahas disini adalah metode yang digunakan oleh cgmatter <https://www.cgmatter.com/> dimana dia juga menyediakan script untuk menjalankan proses ini, tidak perlu plug in dan tidak berbayar

CGMATTER facial capture



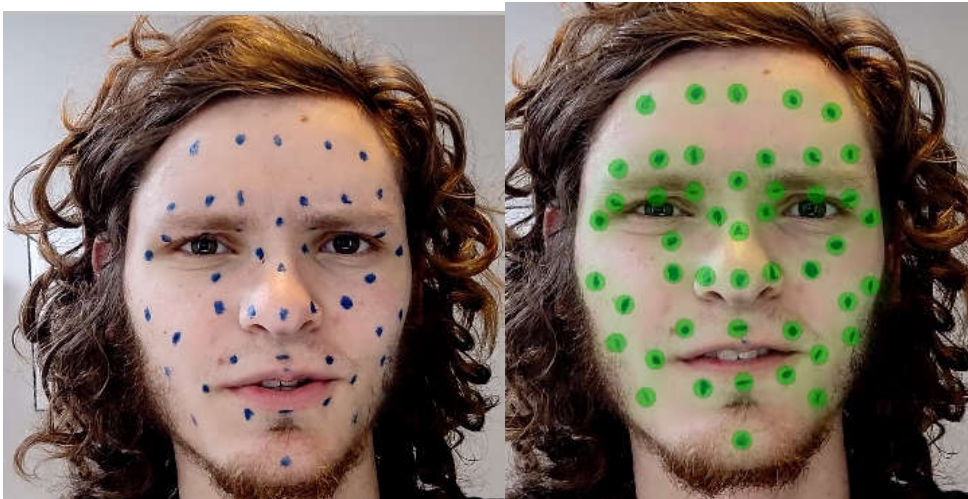
facial motion capture banyak cara dan metode dan peralatannya , dan dalam pembahasan ini untuk facial motion capture diambil cara yang yang termurah hanya mengandalkan kamera handpone ,



alat untuk penanda (ballpoint) dan software blender dengan menggunakan script open source. tanpa menggunakan plugin. metode ini mengandalkan pada deteksi titik pada wajah.

tahap satu adalah pengambilan ngambar

pertama tama wajah harus diberi marker , bisa menggunakan pensil rias untuk keamanan dan kemudahan menghilangkan penanda jika sudah selesai. lokasi marker adalah pada titik dimana wajah kita bergerak sangat banyak , dan semakin banyak markernya akan semakin baik

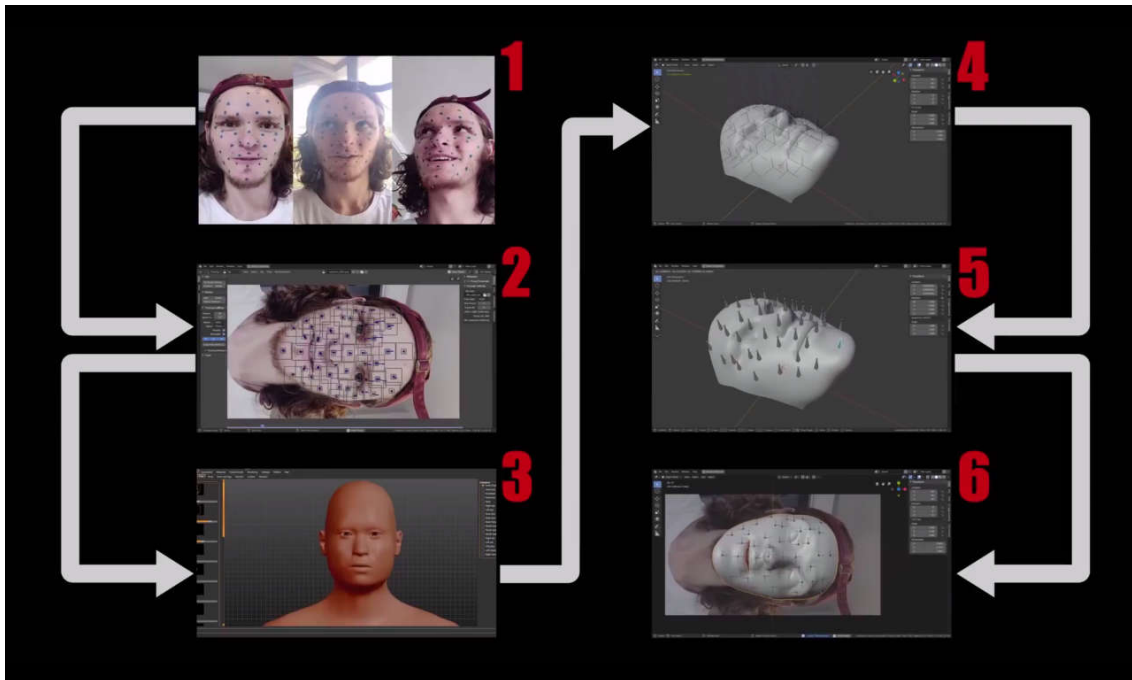


kali ini akan membicarakan tentang bagaimana blender program open source gratis dapat diubah menjadi solusi motion capture yang benar-benar gratis



dan untuk ini kita tidak akan membutuhkan add-on atau plugin apa pun, hanya menggunakan kamera ponsel dan pensil rias,

alur kerja yang harus kita lalui untuk mendapatkan hasil akhir. ada enam langkah dan penting untuk benar-benar memahami tujuan dari setiap langkah sehingga Anda bisa mengekstrapolasi teknik-teknik ini. Di luar kasus khusus pelacakan wajah dan setelah kita mempelajari seluruh alur kerja dengan melakukan setiap langkah secara manual, kita akan benar-benar dapat mengotomatiskan sebagian besar dari ini.



ada skrip Python dasar yang akan lakukan otomatisasi ini dan Anda dapat menggunakan dan mendistribusikan skrip secara bebas

#CGMatter Face Mocap Script

```
import bpy
```

```
iter = 1
```

```
#rename trackers
```

```
FaceTrackers = bpy.data.collections.new('FaceTrackers')
```

```
bpy.context.scene.collection.children.link(FaceTrackers)
```

```
for trackers in bpy.data.objects:
```

```
    if trackers.type != 'EMPTY':
```

```
        continue
```

```
    else:
```

```
        iter = iter + 1
```

```
        FaceTrackers.objects.link(trackers)
```

```
if FaceTrackers:
```

```
    for i, o in enumerate(FaceTrackers.objects):
```

```
        o.name = "Tracker%d" % (i+1)
```

```
#face mesh setup
```

```
bpy.context.scene.frame_set(1)
```

```
for obj in bpy.context.selected_objects:
```

```
    obj.name = "Head"
```

```
    obj.data.name = "Head"
```

```

bpy.ops.object.transform_apply()

#trackers with depth
for value in range(1,iter):
    selectname = "Tracker" + str(value)
    ob = bpy.data.objects[selectname]
    bpy.context.view_layer.objects.active = ob
    bpy.context.object.constraints["Follow Track"].depth_object = bpy.data.objects["Head"]

#add armatures to trackers
for value in range(1,iter):
    selectname = "Tracker" + str(value)
    ob = bpy.data.objects[selectname]
    bpy.context.view_layer.objects.active = ob
    bpy.ops.object.armature_add(enter_editmode=False, location=ob.matrix_world.translation)

#select armatures
for bones in bpy.data.objects:
    if bones.type != 'ARMATURE':
        continue
    else:
        bones.select_set(state=True)

#join armatures in correct order
bpy.context.view_layer.objects.active = ob = bpy.data.objects['Armature']
bpy.ops.object.join()
bpy.ops.object.transform_apply()

#rename bones in armature
for value in range(1,iter):
    if value == 1:
        bonename = "Bone"
    elif value > 1 and value < 11:
        bonename = "Bone.00" + str(value-1)
    elif value >= 11:
        bonename = "Bone.0" + str(value-1)
    bone = bpy.data.objects['Armature'].data.bones.get(bonename)
    bpy.data.objects['Armature'].data.bones[bonename].name = 'Bone' + str(value)

```

enam langkah yang perlu kita lalui

langkah pertama jelas perlu merekam video, kita akan membahas cara meletakkan pelacak di wajah Anda di tempat khusus Anda harus meletakkannya dan cara memfilmkan wajah Anda untuk mencegah masalah di kemudian hari dalam alur kerja .

langkah kedua kita akan melakukan pelacakan , kita akan membahas metode tercepat untuk gunakan dan cara menangani pelacakan penanda yang sulit dan ingat mulai saat ini kita pada akhirnya akan memiliki skrip Python yang mengotomatiskan sisa alur kerja.

pada langkah ketiga kita akan membuat mesh yang akan kita gunakan sebagai target kita data motion capture dan untuk tutorial ini kita akan menunjukkan cara membuat wajah Anda sendiri, tetapi tentu saja Anda dapat melakukannya dengan mesh apa pun yang Anda inginkan.

di langkah keempat kita akan menggunakan mesh yang kita buat untuk memberikan kedalaman pelacak kita dengan baik gunakan pelacak ini untuk meletakkan beberapa armature yang akan bertanggung jawab untuk mengubah bentuk wajah.

pada langkah kelima kita akan memasang mesh wajah menggunakan armature. ini akan memungkinkan kita mengubah bentuk geometri dengan tulang dan kita akan membahas bagaimana kita bisa men-tweak rig dengan weight paint. Akhirnya di urutan keenam langkah kita akan menautkan rig kita ke pelacak yang akan menganimasikan mesh kita, kita kemudian dapat memulai tindakan ini untuk meningkatkan kinerja pemutaran.

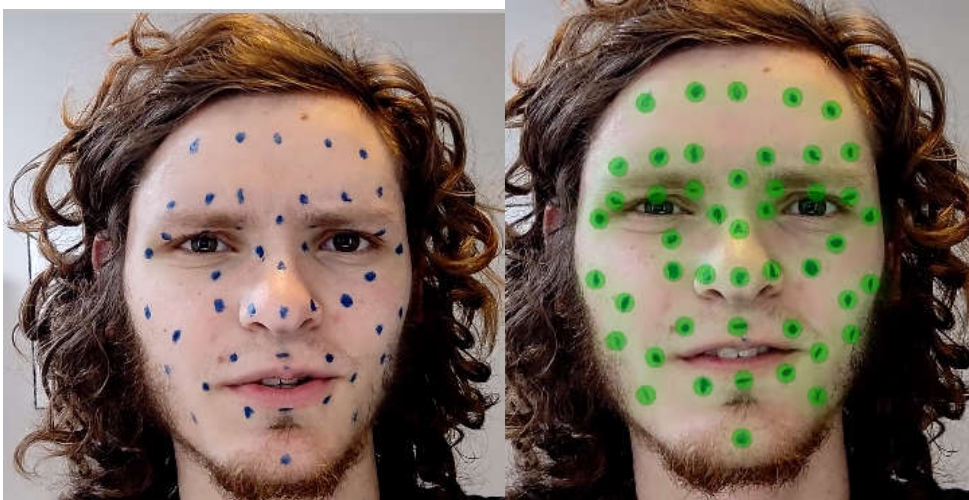


jelas hal pertama yang perlu kita lakukan gerakan menangkap untuk menambahkan penanda pelacakan berkualitas tinggi dengan cepat di sekitar wajah, kita menginginkan kulit dan ekspresi kita sehingga kita dapat menambahkan banyak penanda dengan tersebut.

Menggunakan spidol dan itu benar-benar berfungsi dengan sebagian besar warna kulit kecuali jika Anda memiliki kulit super gelap di mana yang hitam muncul, Anda dapat mencoba menggantinya dengan pewarna putih tentu saja kerugian besar untuk spidol dan whiteout adalah sangat sulit untuk dihilangkan dari kulit Anda juga memiliki fakta bahwa keduanya dianggap agak beracun.

Jadi yang kita sarankan adalah menemukan semacam pensil riasan gelap dan menggunakannya untuk membuat penanda tidak hanya tidak dimaksudkan dioleskan ke kulit tetapi

lebih mudah dibersihkan jadi bagi Anda yang tidak memiliki pensil riasan apa pun coba cari seseorang yang punya untuk dipinjam.

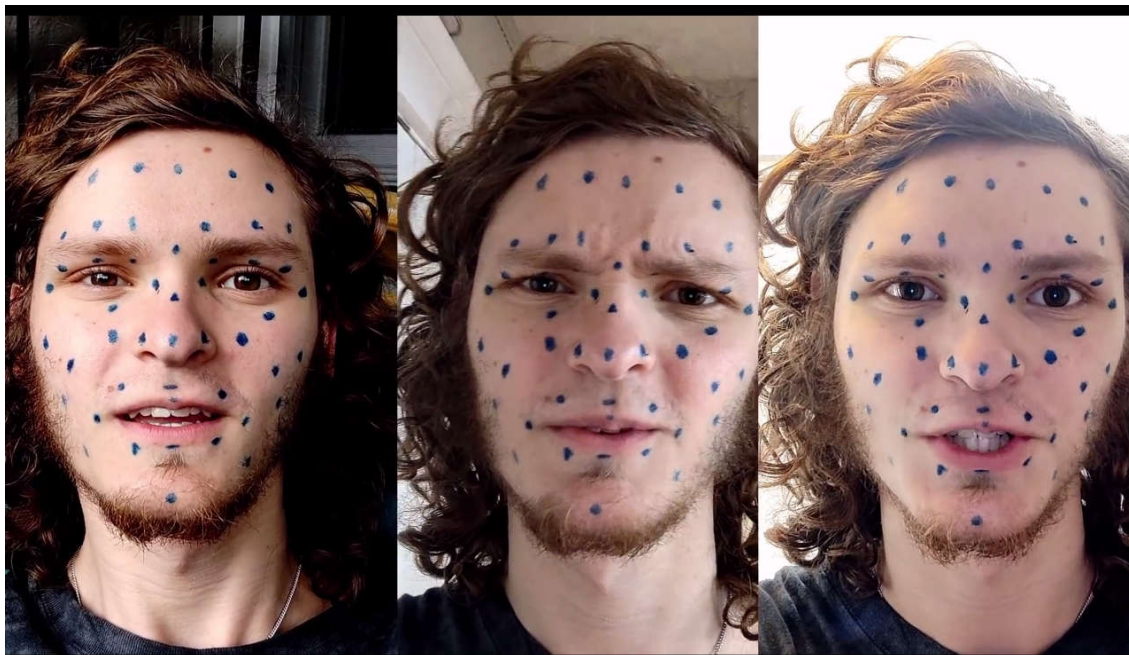


ada beberapa prinsip sederhana yang ingin kita ikuti pertama-tama kita ingin mengidentifikasi area wajah yang paling banyak berubah bentuk atau di tangkap secara umum ini adalah area seperti mulut mata dan alis dan di area ini kita ingin menerapkan lebih banyak penanda pelacakan dari biasanya, semakin tinggi kepadatan penanda kita, semakin banyak detail yang didapat dalam tangkapan kita.

kita menghindari penambahan penanda dan area yang tidak dapat dilihat kamera ini adalah area seperti di bawah hidung, sisi wajah dan area yang terhalang oleh rambut Anda jika tidak, kita dapat menambahkan penanda cukup banyak di tempat lain secara simetris. pastikan harus simetris jika tidak akan mendapatkan detail tambahan di satu sisi wajah.

dan sekarang kita telah menggambar penanda yang kita inginkan untuk memastikan kita sedang syuting g rekaman kita dengan benar. Cara teraman untuk melakukan ini adalah dengan bidikan langsung dengan kamera. dengan cara ini penanda kita tetap terlihat di seluruh bidikan sehingga kita bisa mendapatkan deformasi penuh dan kita juga harus menghindari gerakan kamera apa pun yang ditafsirkan sebagai bagian dari kinerja.

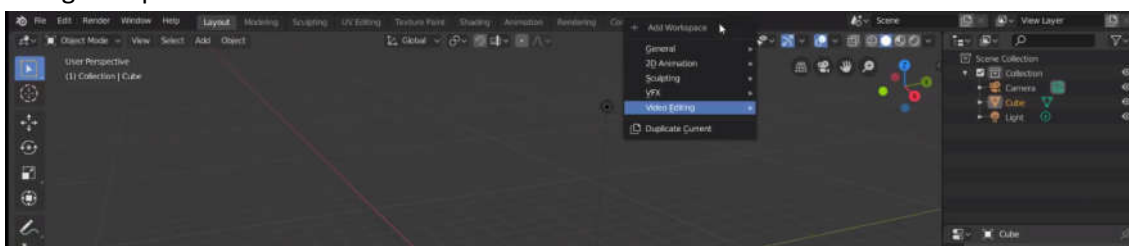
jenis bidikan yang buruk untuk motion capture adalah bidikan di mana beberapa pelacak kita terhalang dan juga bidikan di mana kita menambahkan gerakan melalui bentuk kamera dan jika kita mengikuti prinsip ini, dapat dilacak dengan sempurna.



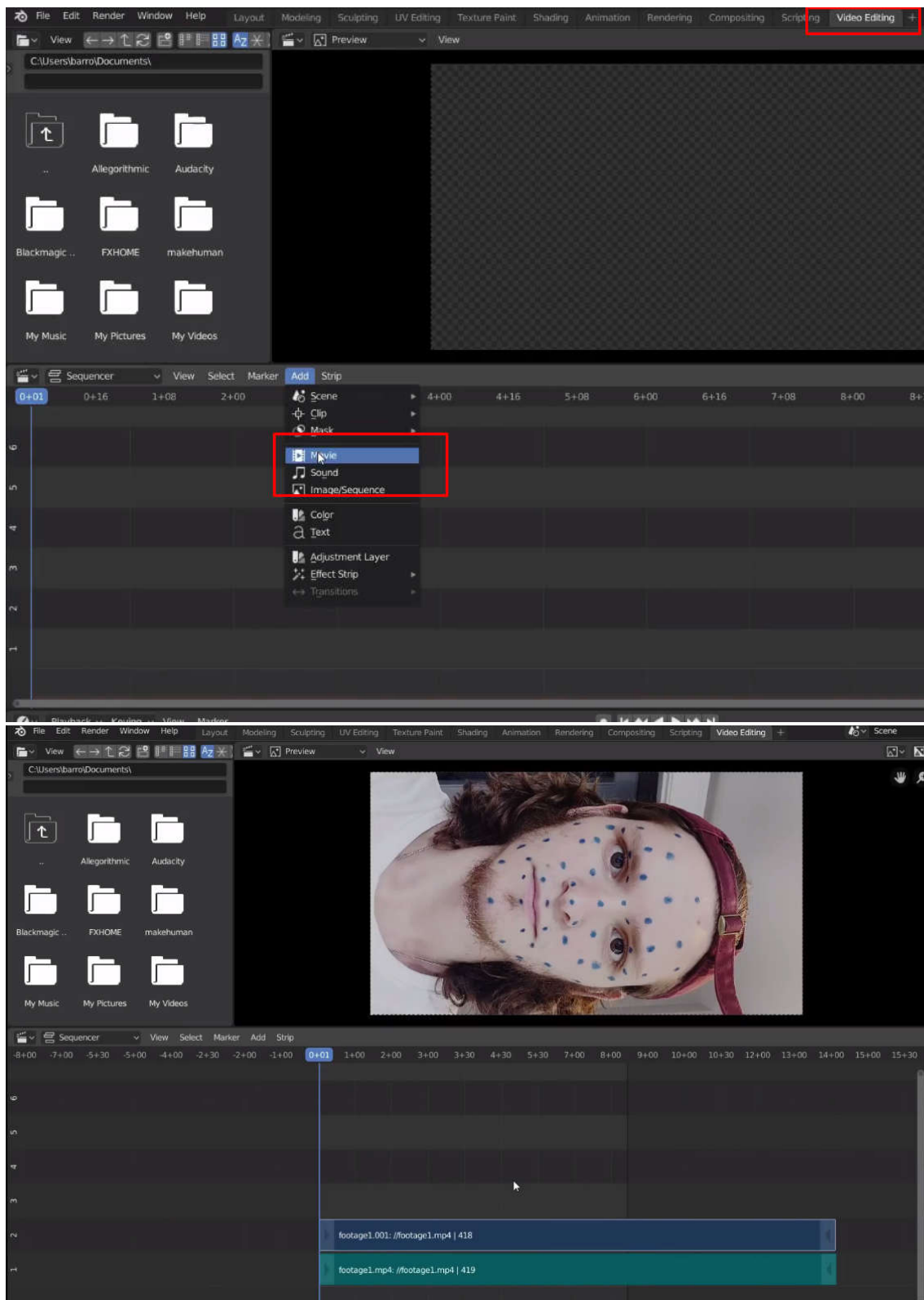
bidikan yang perhatikan bahwa kondisi pencahayaan tidak terlalu penting selama kita dapat melihat penanda dan bidikan bergerak juga baik selama wajah tetap di tengah.

jadi ini adalah hasil bidikan yang akan digunakan untuk motion capture dan jelas ketika kita memfilming ini di ponsel kita, kita akhirnya memfilmkannya secara vertikal. hasil bidikan ini nanti akan dapat kita dapat memanfaatkan untuk menjadi seperti bidikan horizontal yang mengisi frame dan hasil inilah yang akan kita lacak

sebelum kita melakukan pelacakan itu kita ingin mengonversi rekaman kita menjadi urutan gambar yang cenderung ditangani oleh blender dengan lebih baik, jadi dalam blender pindahlah ke ruang kerja pengeditan video di mana kita dapat melakukan konversi ini dan mengimpor rekaman sebagai strip film.

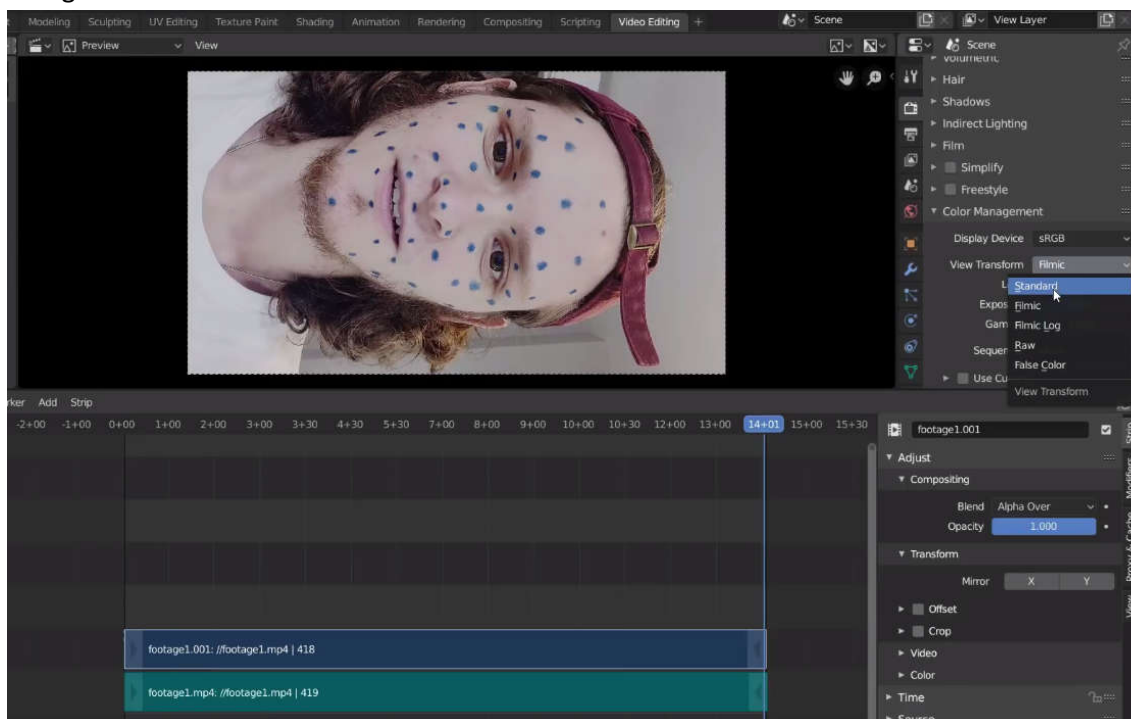


impor clip

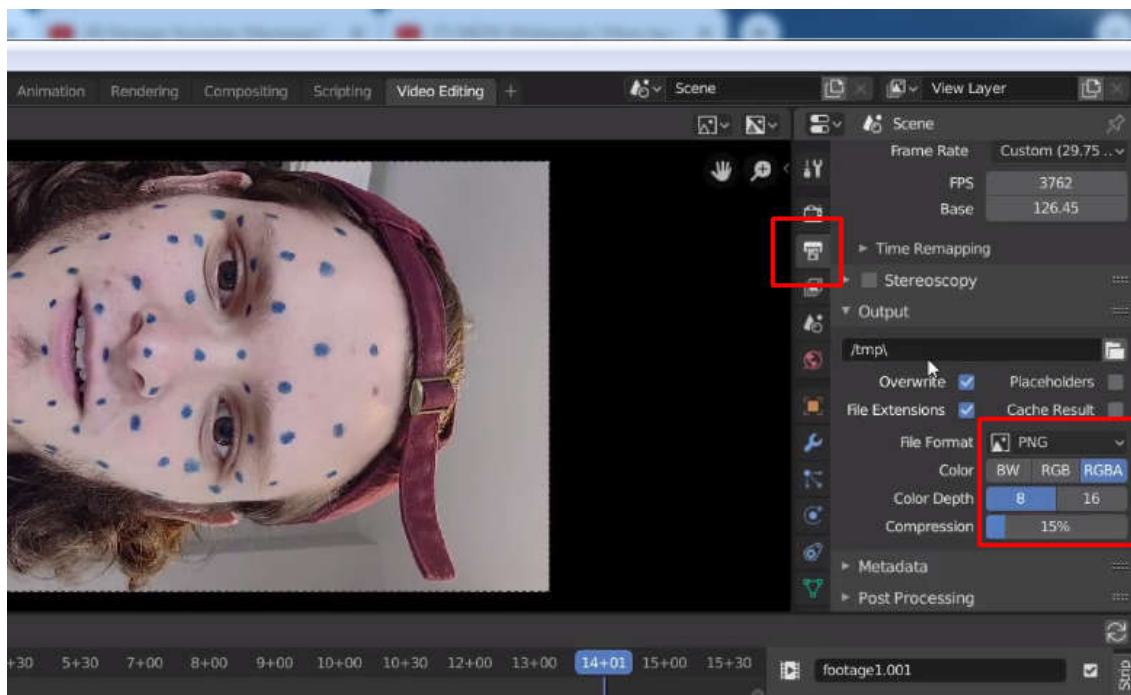


hal yang menyenangkan dari proses ini adalah blender itu akan secara otomatis mengubah kecepatan frame proyek kita, namun kita masih perlu mencocokkan titik akhir proyek kita dan untuk melakukan ini, temukan saja di mana strip film berakhir dan salin nilai frame saat ini ke penggeser titik akhir.

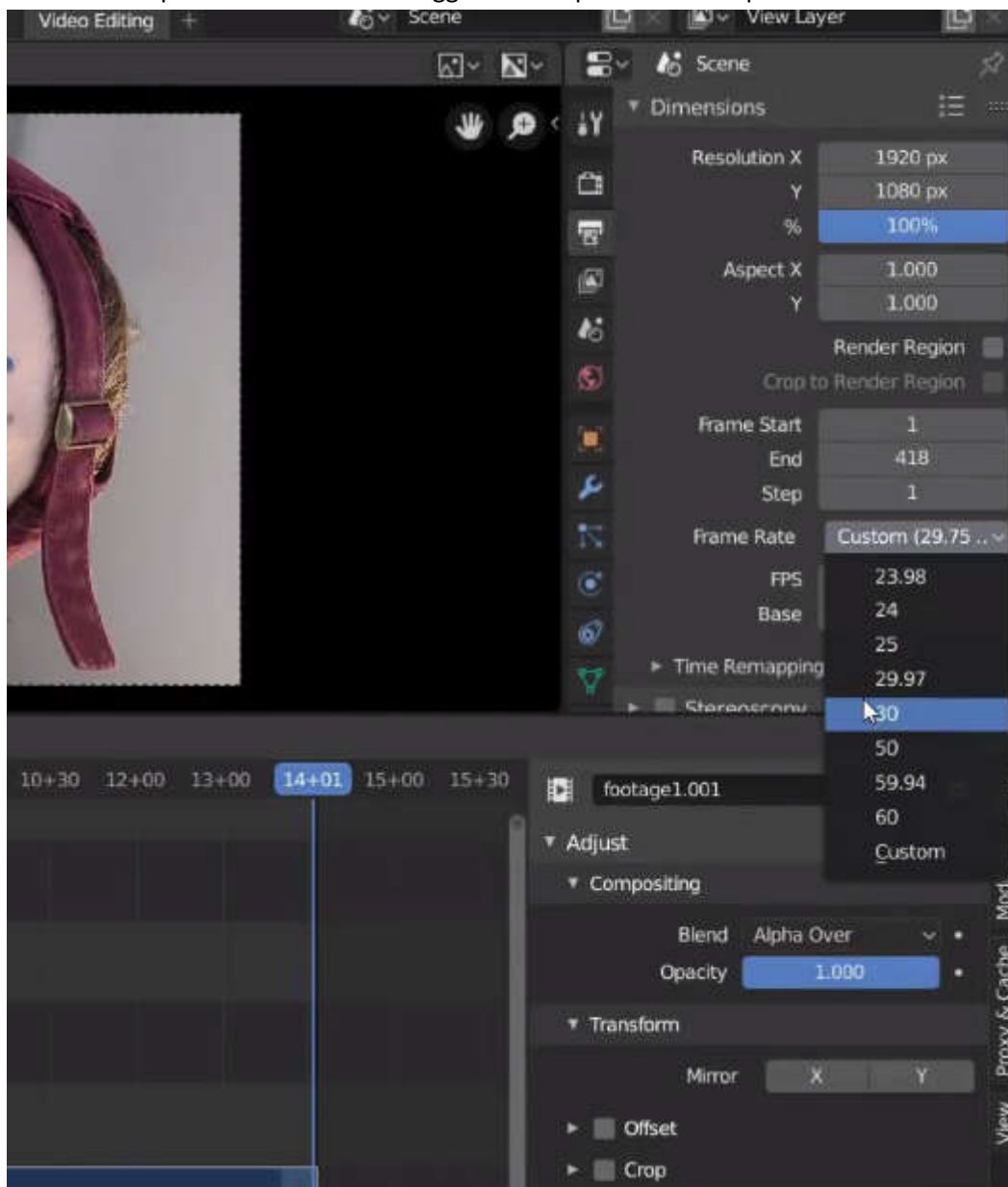
sekarang hal terakhir yang masih perlu kita perbaiki adalah warnanya dari rekaman kita yang jelas telah berubah dari aslinya, pengaturan yang dapat kita atur untuk memperbaikinya disebut transformasi tampilan dan terletak di tab render di bawah manajemen warna dan kita hanya ingin mengalihkan ini dari mode film kembali ke standar



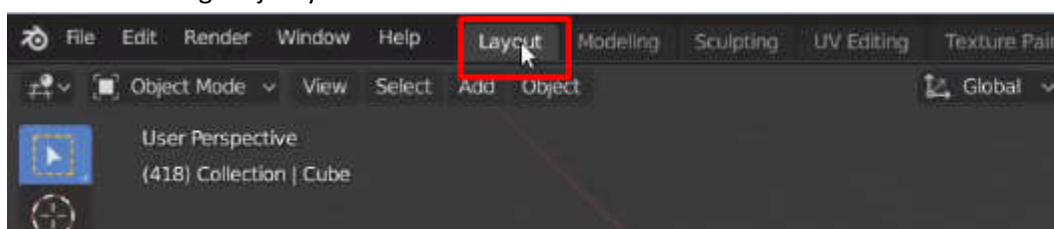
akhirnya untuk mengekspor ini sebagai urutan gambar kita bisa pergi ke tab output dan mengatur ini untuk merender sebagai urutan PNG dengan kompresi nol dan hanya mengatur jalur output untuk render kita



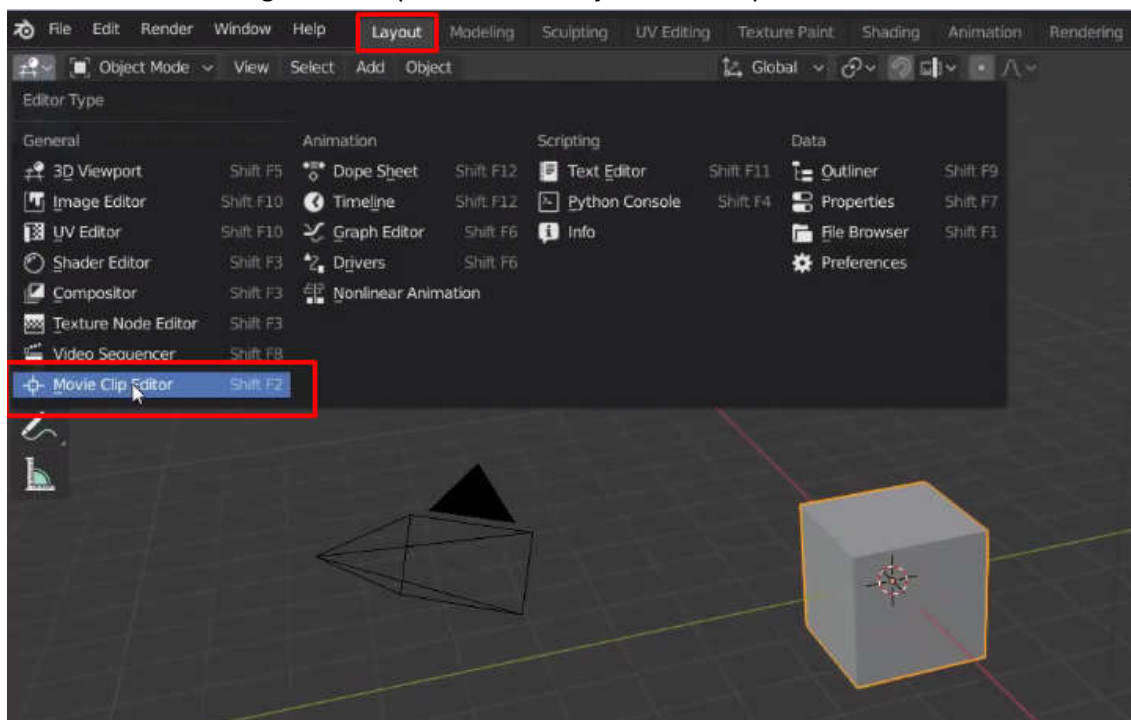
Sekarang untuk membuat gambar kita cukup tekan ctrl f12 dan tunggu blender memproses seluruh klip dan sekali lagi apa yang kita lakukan di sini adalah mengonversi file video menjadi urutan gambar yang kecepatanframennya tidak tergantung sehingga ketika render kita selesai kita dapat memilih kecepatan frame untuk menggunakan seperti 30 frame per detik



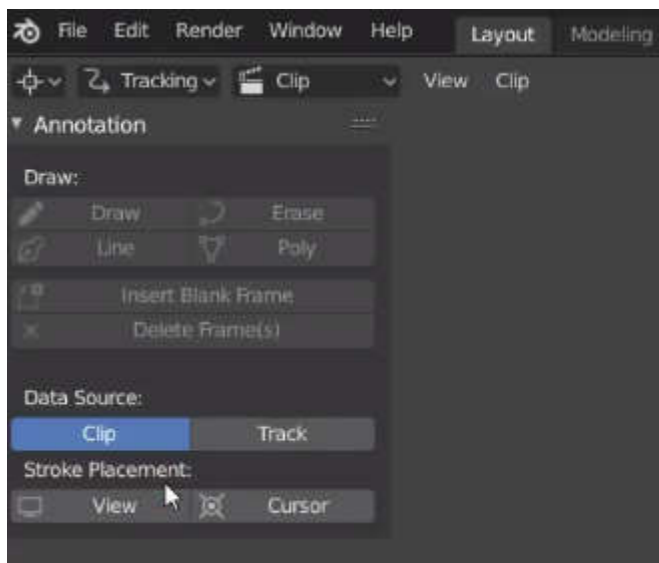
sekarang kita siap untuk beralih ke pelacakan yang berarti kita dapat membuang klip video kita dan kembali ke ruang kerja layout

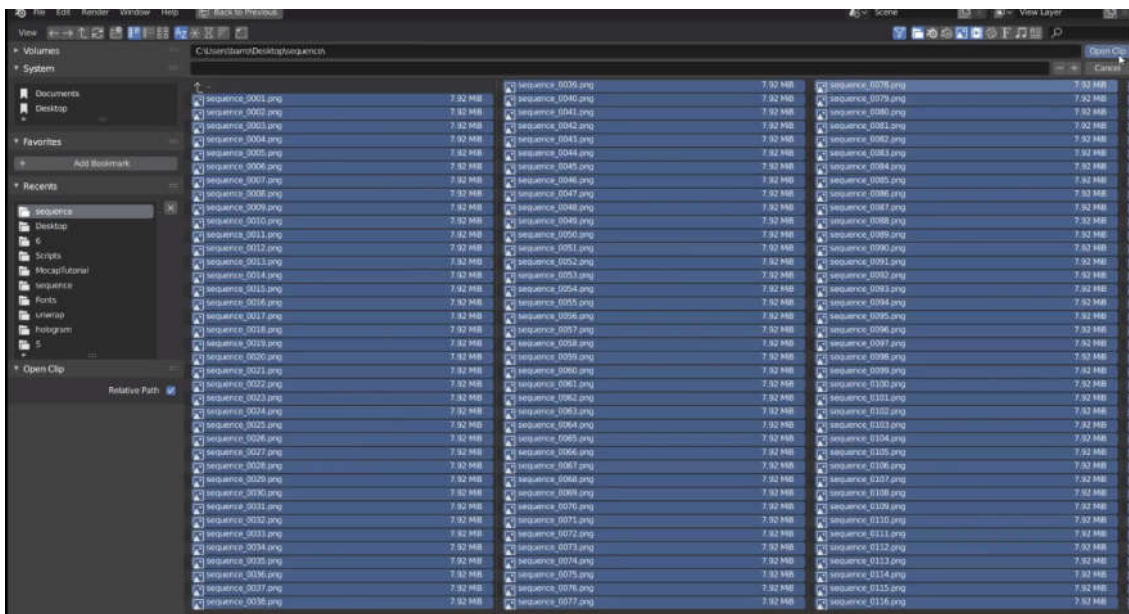


di sini kita harus mengubah viewport 3d kita menjadi editor klip video

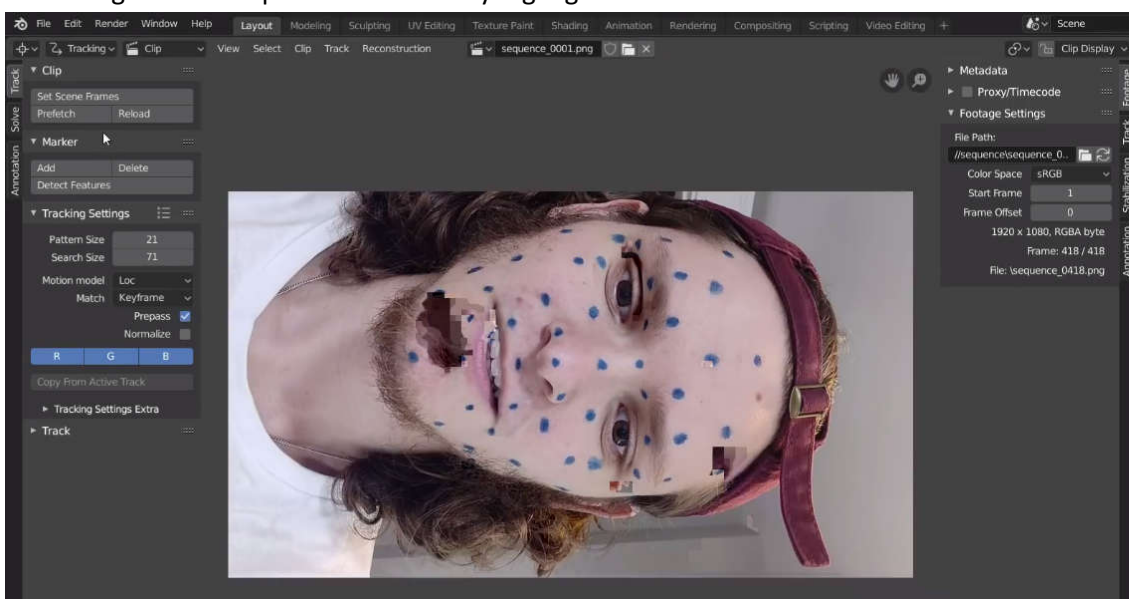


yaitu di mana kita melakukan pelacakan dan kemudian mengimpor dalam urutan gambar yang kita buat



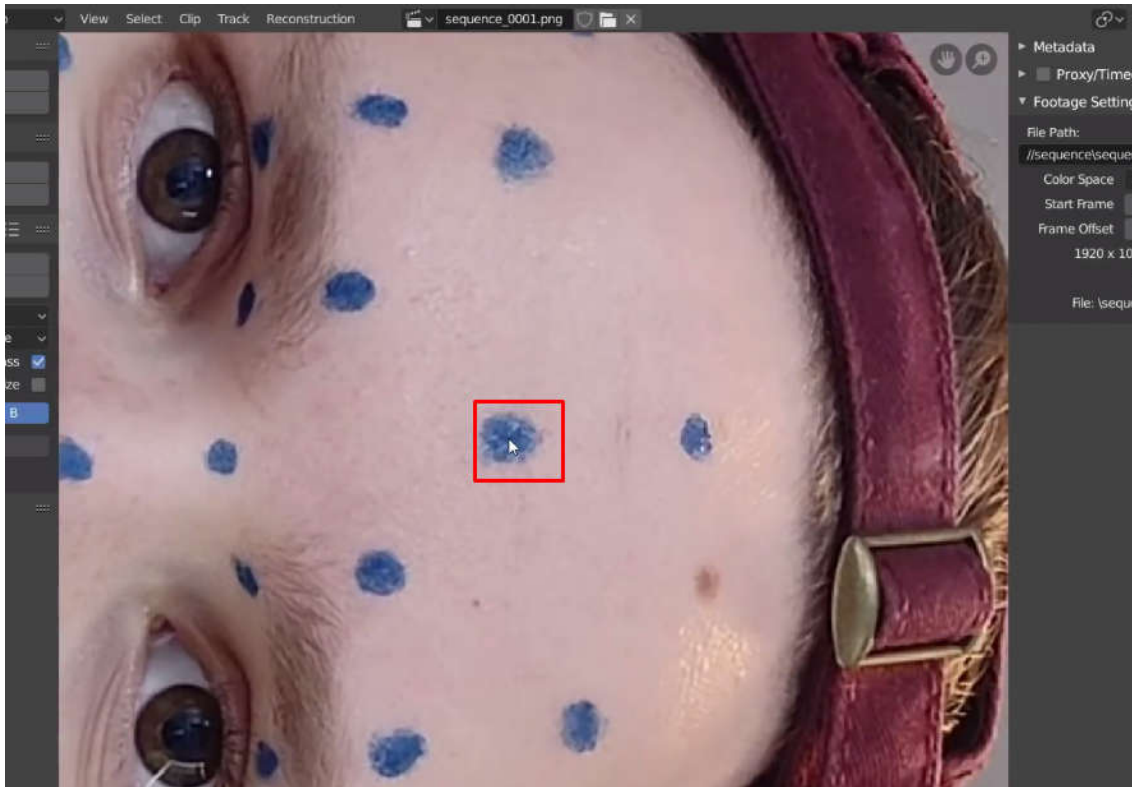


hal pertama yang ingin kita lakukan adalah menekan prefetch yang akan memuat seluruh urutan ke dalam memori dan mencegah kesalahan dari kelambatan di masa mendatang dan untuk mengidentifikasi penanda tertentu yang ingin kita lacak



kita akan menggunakan penanda ini di dahi tempat kita akan meletakkan pelacak dengan mengklik ctrl dan setelah kita memposisikan area pola di sekitar fitur kita, kita dapat melacak ini ke depan

dengan menekan ctrl T

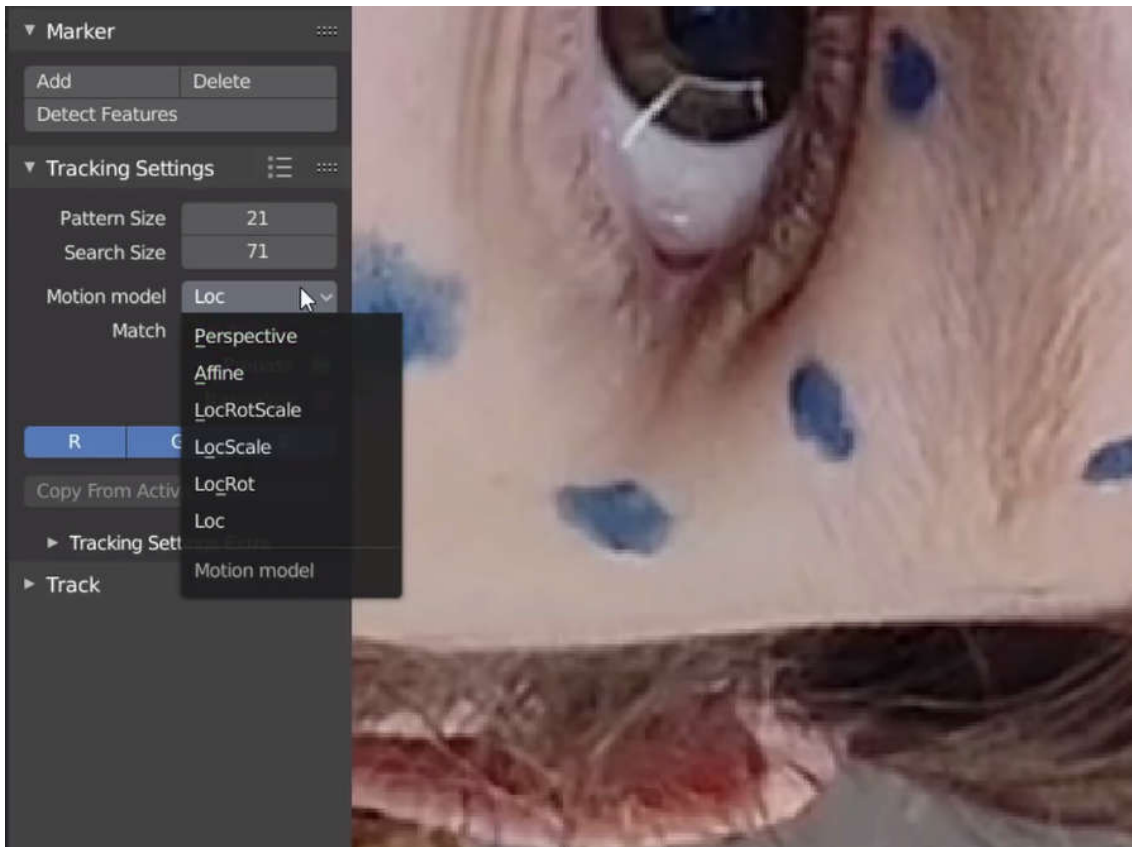


dan dalam hal ini semuanya bekerja dengan sempurna sehingga kita dapat mengunci pelacak kita dengan menekan tombol ini atau menggunakan pintasan ctrl L

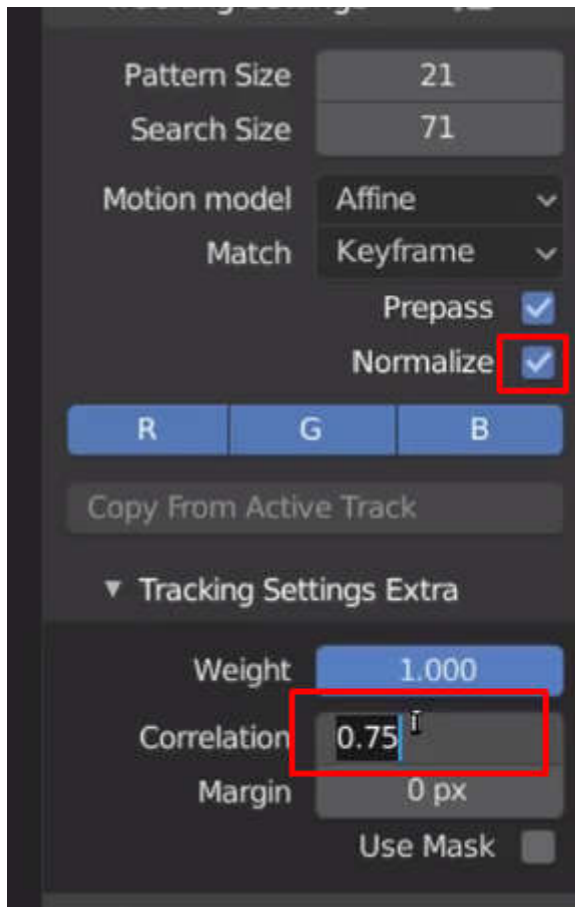


Ini ada beberapa pengaturan pelacak yang ingin kita ubah sedikit untuk membantu kita, pertama-tama kita perlu mengubah model gerakan menjadi sesuatu yang menangani distorsi.

dua model yang dapat melakukan ini adalah perspektive dan affine

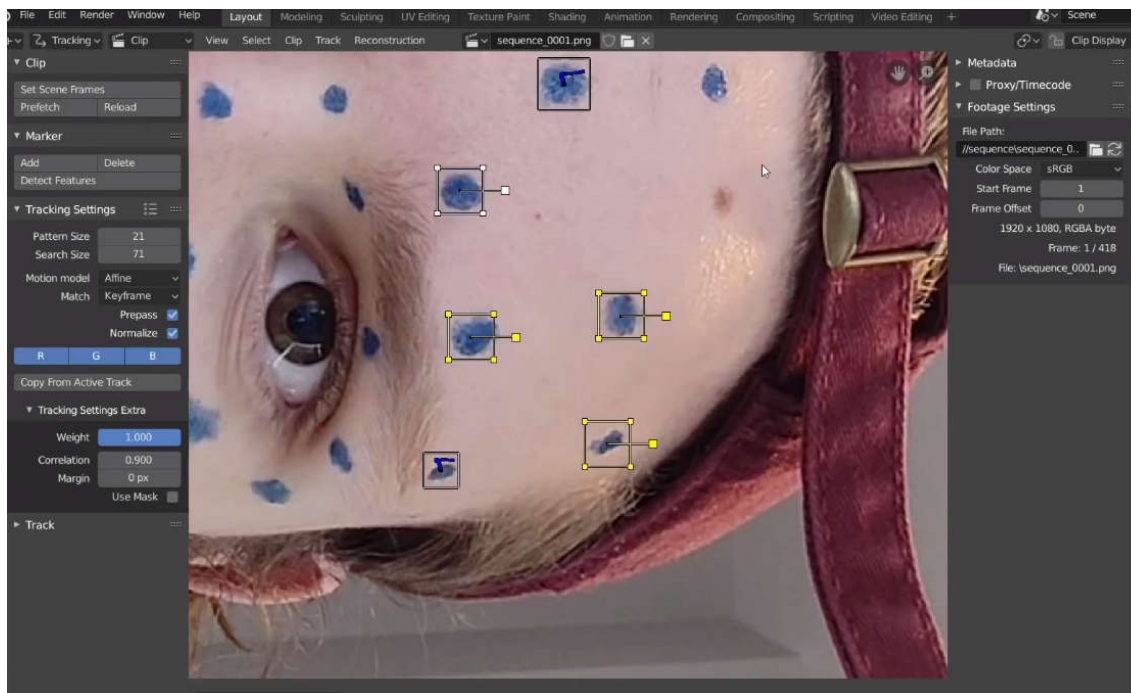


dan karena mereka deformasi bukanlah perubahan perspektif yang sebenarnya melainkan hanya melakukan penggeseran dan peregangan



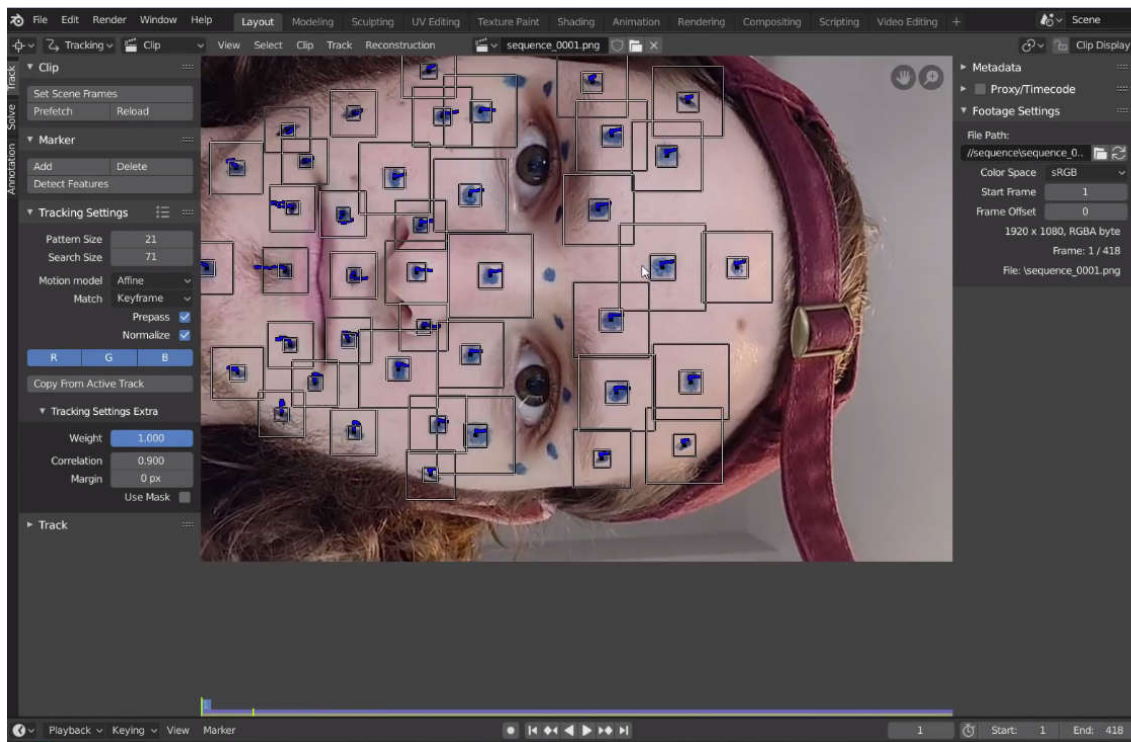
beberapa hal lain yang dapat kita ubah adalah mengaktifkan normalisasi untuk membuat pelacakan tidak berubah terhadap perubahan pencahayaan dan juga meningkatkan korelasi kita menjadi 0,9 yang berarti kita memerlukan kepercayaan 90 persen pada setiap frame atau jika tidak pelacakan akan berhenti

Sekarang setelah kita membuat perubahan ini, mari tambahkan pelacak kembali. dan buat lacak kata-kata jadi jelas parameter pelacakan ini jauh lebih baik daripada yang kita miliki sebelumnya dan kita dapat menggunakan ini untuk semua sisa penanda.

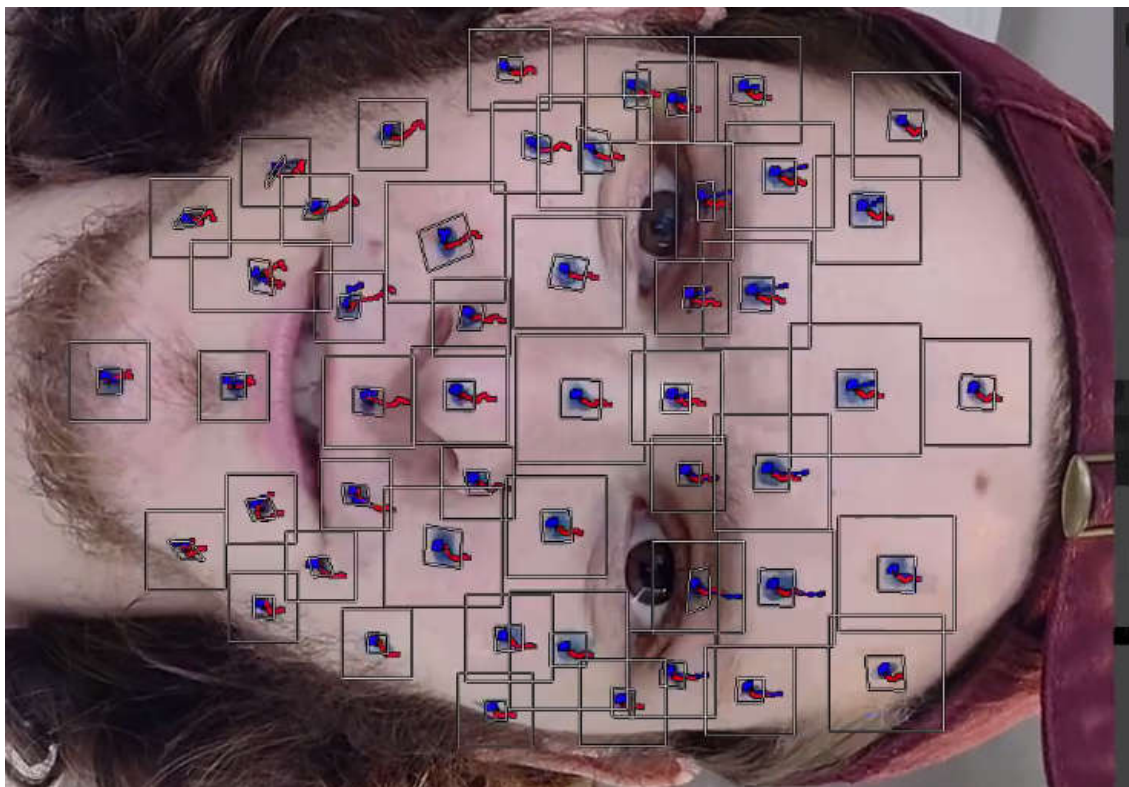


kali ini mari coba lacak sekelompok dari mereka yang dapat kita pilih dengan skala ke atas untuk menutupi fitur kita dan melacak ke depan dengan ctrl T sehingga meskipun kita melacak beberapa fitur sekaligus, kita masih mendapatkan hasil yang sangat cepat dan akurat yang dapat kita awasi

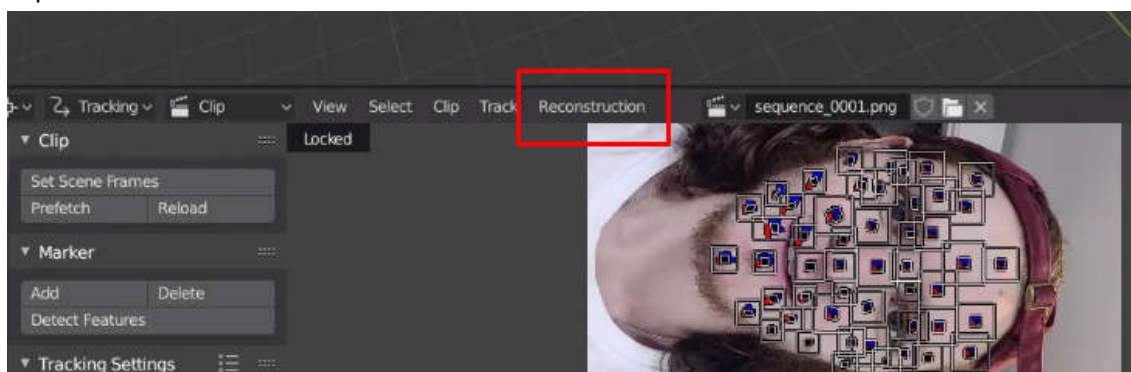
jadi mari kita kunci grup ini dengan ctrl L dan ulangi ini dengan cluster berikutnya



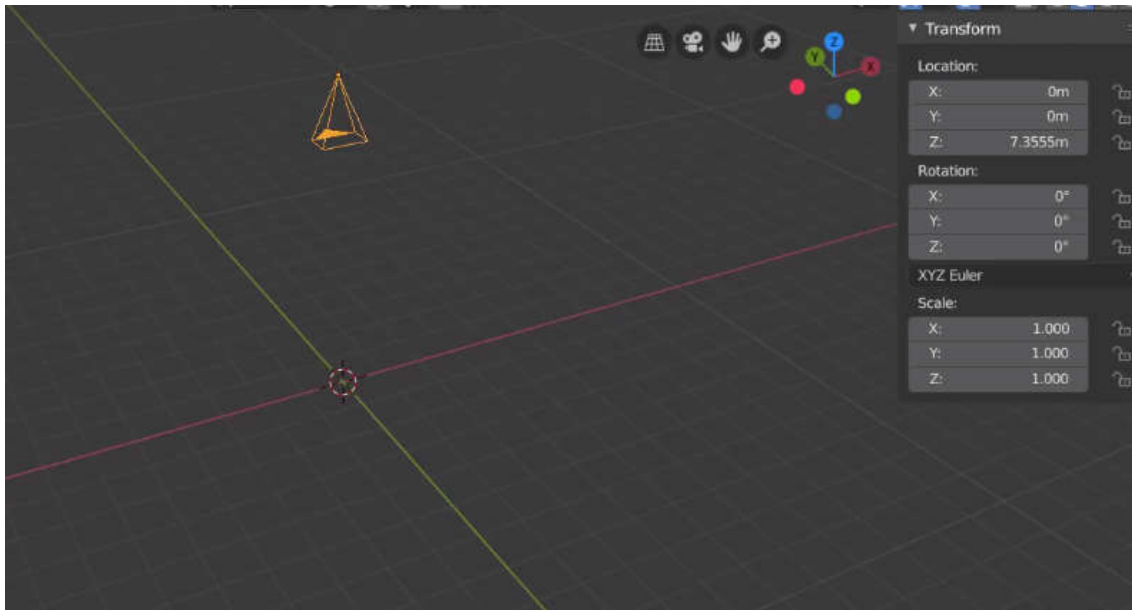
Anda dapat melihat bahwa saat melacak cluster ini kita benar-benar mendapatkan pelacak yang tidak bertahan sampai akhir



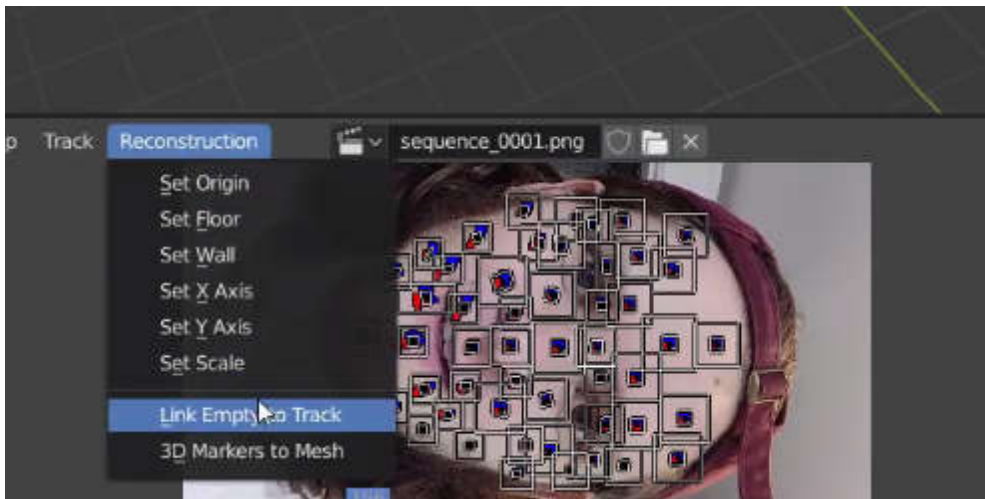
sekarang semuanya sudah siap dan mterorganisir, kita perlu memilih semua pelacak kita di editor klip film dan kemudian di bawah tautan hit rekonstruksi

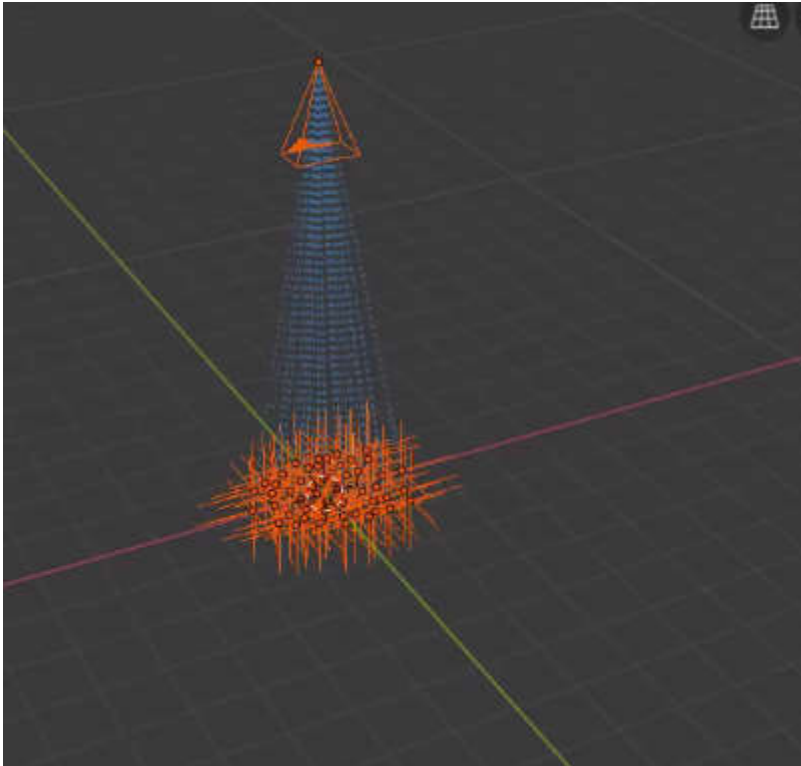


arahkan kamera ke z axis



select semua marker lalu tekan reconstruction dalu pilih link to empty track



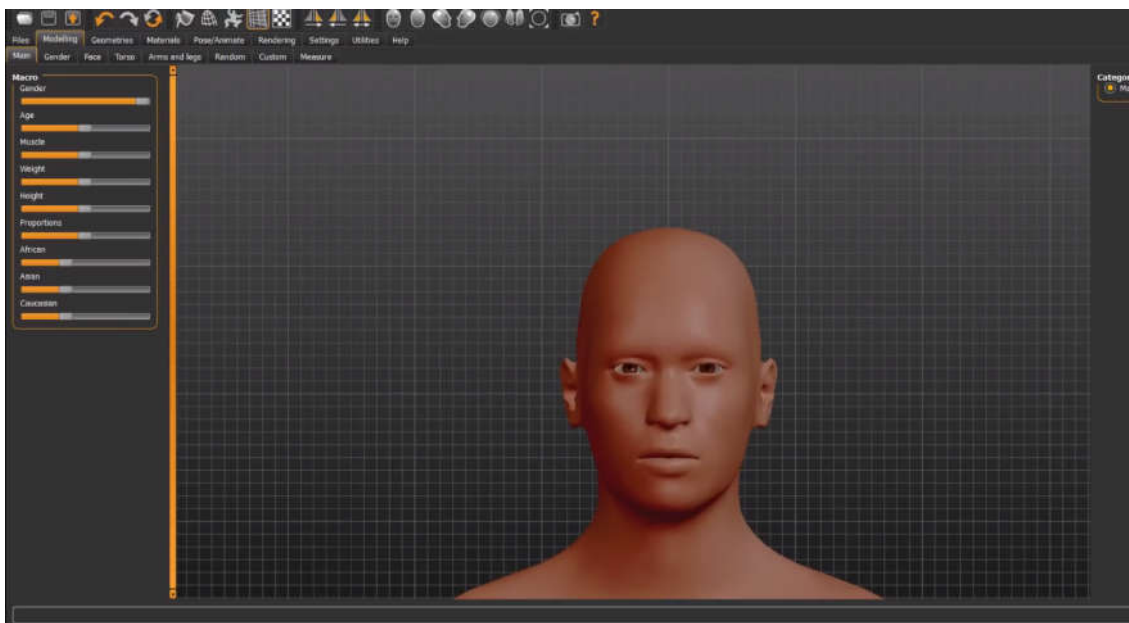


tentu saja kita perlu menambahkan beberapa informasi kedalaman. saat ini kita hanya memiliki dua dimensi yang sesuai dengan gerakan X&Y pelacak. kita dibutuhkan mesh wajah yang akan kita gunakan untuk menambahkan informasi kedalaman.

kita jelas dapat mengunduh model wajah gratis mempostingnya di situs-situs seperti turbosquid, kita tidak merekomendasikan mengunduh model adalah karena kita tidak memiliki kendali terutama masalah jika Anda ingin mesh terlihat seperti wajah Anda karena kemungkinan tidak ada model yang diposting akan cocok dengan itu .

opsi kedua

yang merupakan program sumber terbuka gratis yang disebut make human pada dasarnya ini adalah pembuat karakter 3d lengkap tetapi tentu saja kita hanya tertarik dengan parameter wajah jadi yang bisa kita lakukan adalah mentweak slider ini untuk membuat wajah kustom dan jika Anda secara khusus menginginkan mesh wajah Anda sendiri



makehuman memungkinkan Anda menambahkan gambar latar belakang yang dapat digunakan sebagai referensi dan saat kita selesai mengedit karakter kita bisa pergi ke tab ekspor dan mengirimkannya ke blender sebagai file obj dasar jadi hal yang menyenangkan tentang pendekatan ini adalah bahwa dengan waktu yang cukup kita dapat menghasilkan hampir semua wajah yang kita inginkan satu-satunya downside di sini adalah ini bisa jadi proses yang cukup lambat untuk membuat wajah Anda sendiri

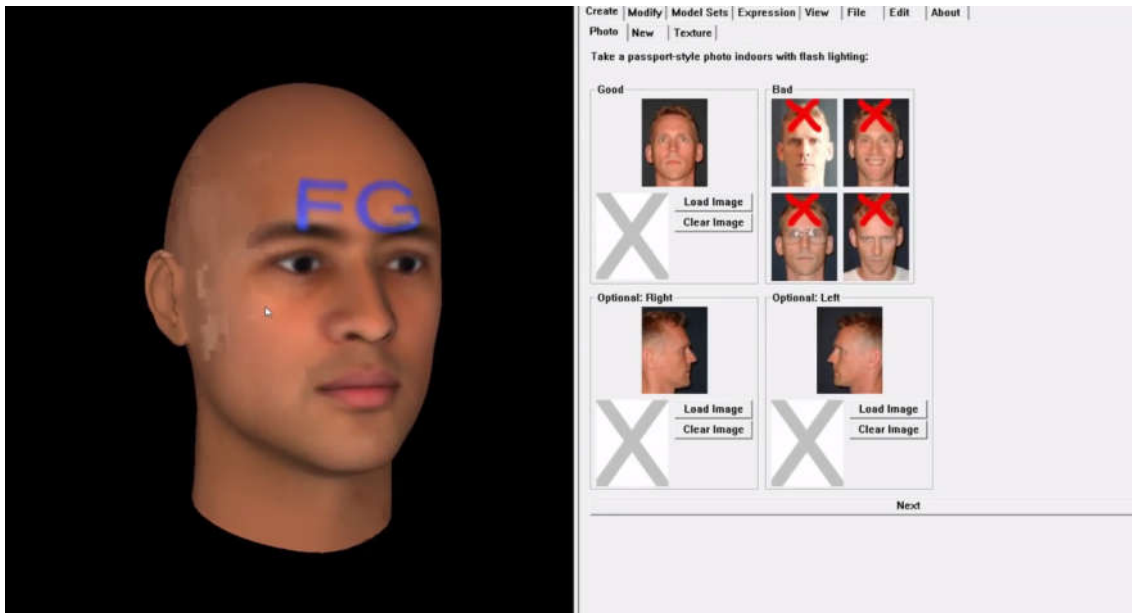
opsi ketiga kita adalah program yang disebut facegen modeler yang sangat mahal

Demo:	FREE
Core:	\$299
Pro:	\$699
Enterprise:	\$1,395

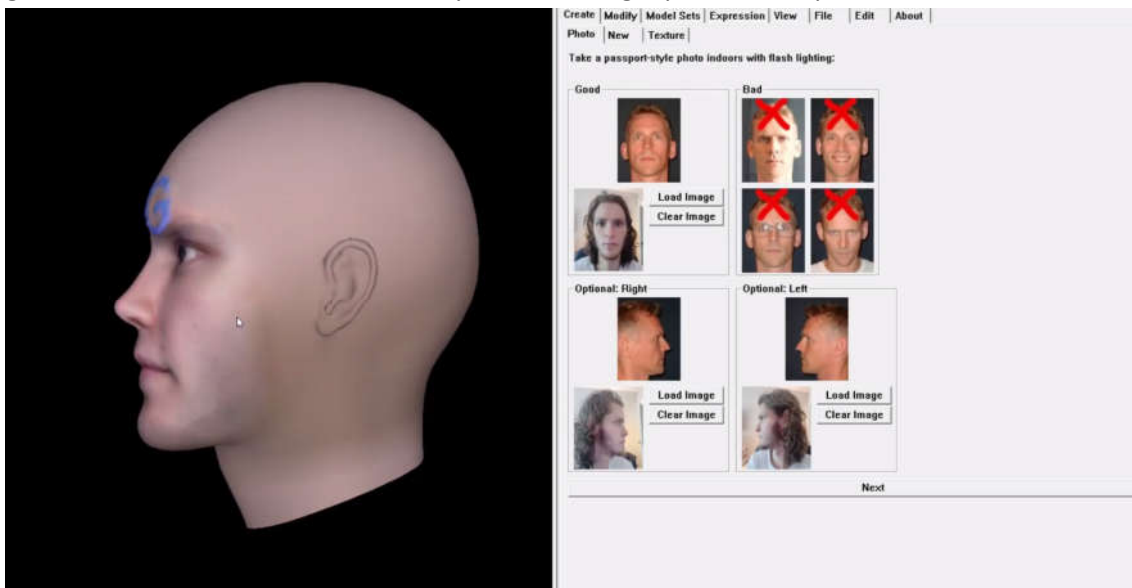
tetapi ternyata versi demo gratis sebenarnya memiliki semua alat yang kita butuhkan dan apa yang dapat dilakukan program ini adalah menghasilkan model 3d wajah Anda hanya dari foto yang menghadap ke depan dan 2 foto yang menghadap ke samping jadi ini sebenarnya akan menjadi metode yang kita gunakan untuk tutorial ini karena kita

ingin mesh target kita cocok sedekat mungkin.

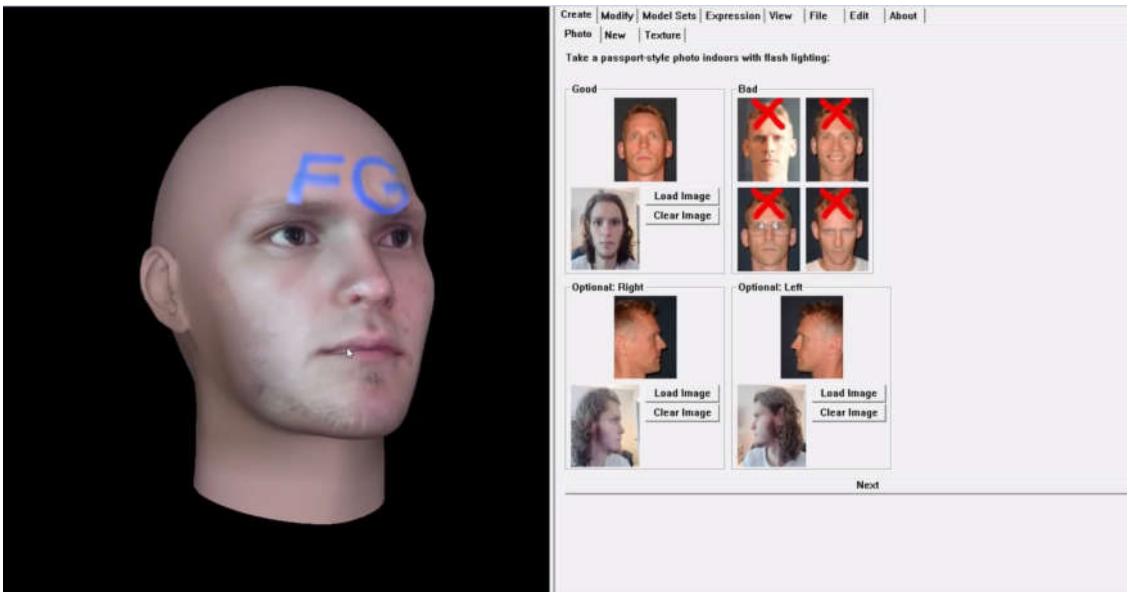
tampilan awal



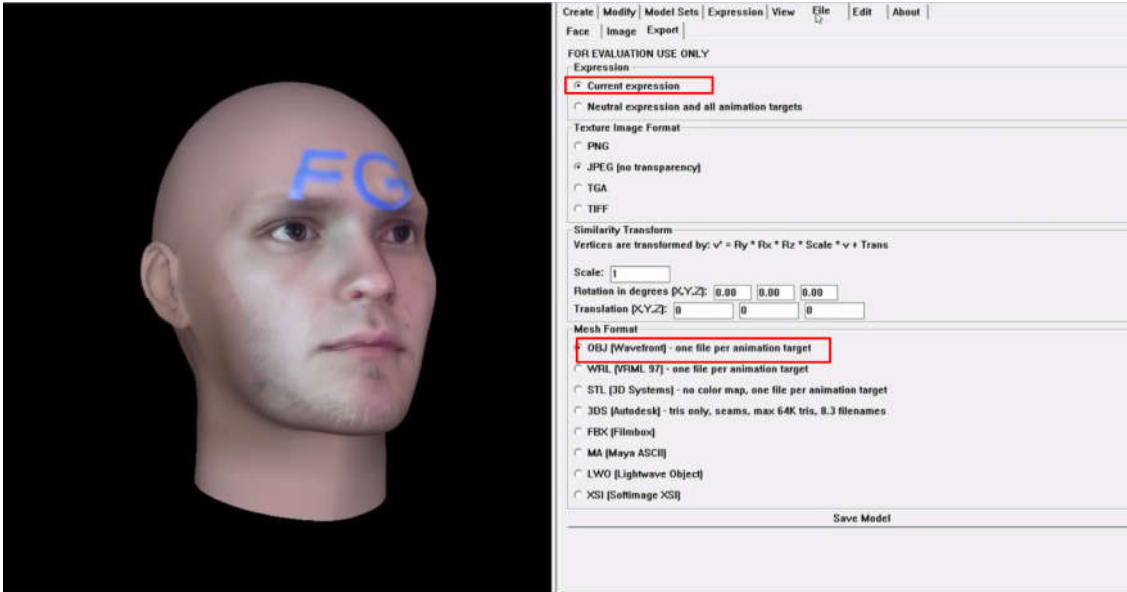
dan kita ingin memastikan kita berada di bagian foto di bawah tab buat di sinilah kita memuat 3 gambar referensi kita dan kita hanya akan mengimpor foto depan foto kanan dan foto kiri d



sekarang kita dapat menekan next untuk memulai perataan kita sebelum mencocokkan salah satu fitur pastikan gambar diputar tegak dan kemudian yang harus kita dan kita akhirnya mendapatkan model kepala khusus ini yang cukup akurat cocok dengan referensi sekarang

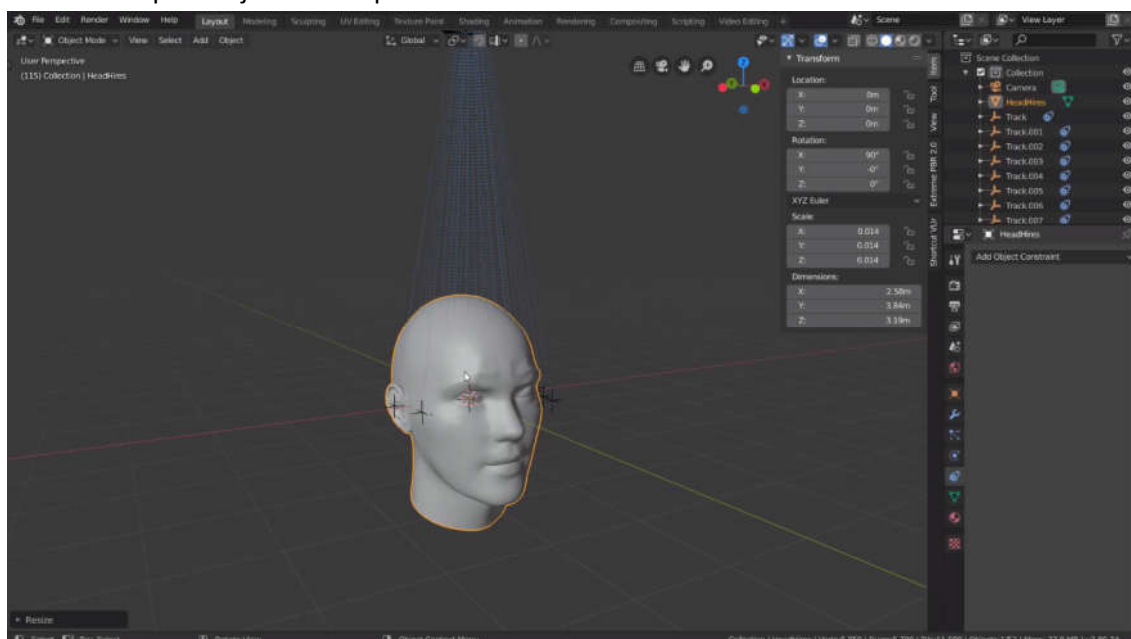


sesuatu yang mungkin Anda memperhatikan adalah logo FG wajah besar di dahi yang merupakan watermark yang selalu ditempatkan dalam versi demo ini benar-benar membuat tekstur yang dihasilkan tidak dapat digunakan tetapi untuk tujuan kita, kita bahkan tidak memerlukan tekstur ini untuk memulai langsung menuju ke tab file yang mana kita akan mengekspor mesh kita dan satu-satunya pengaturan yang benar-benar ingin kita perhatikan adalah ekspresi yang harus diatur ke ekspresi saat ini dan format mesh di mana kita hanya dapat menggunakan file obj dasar

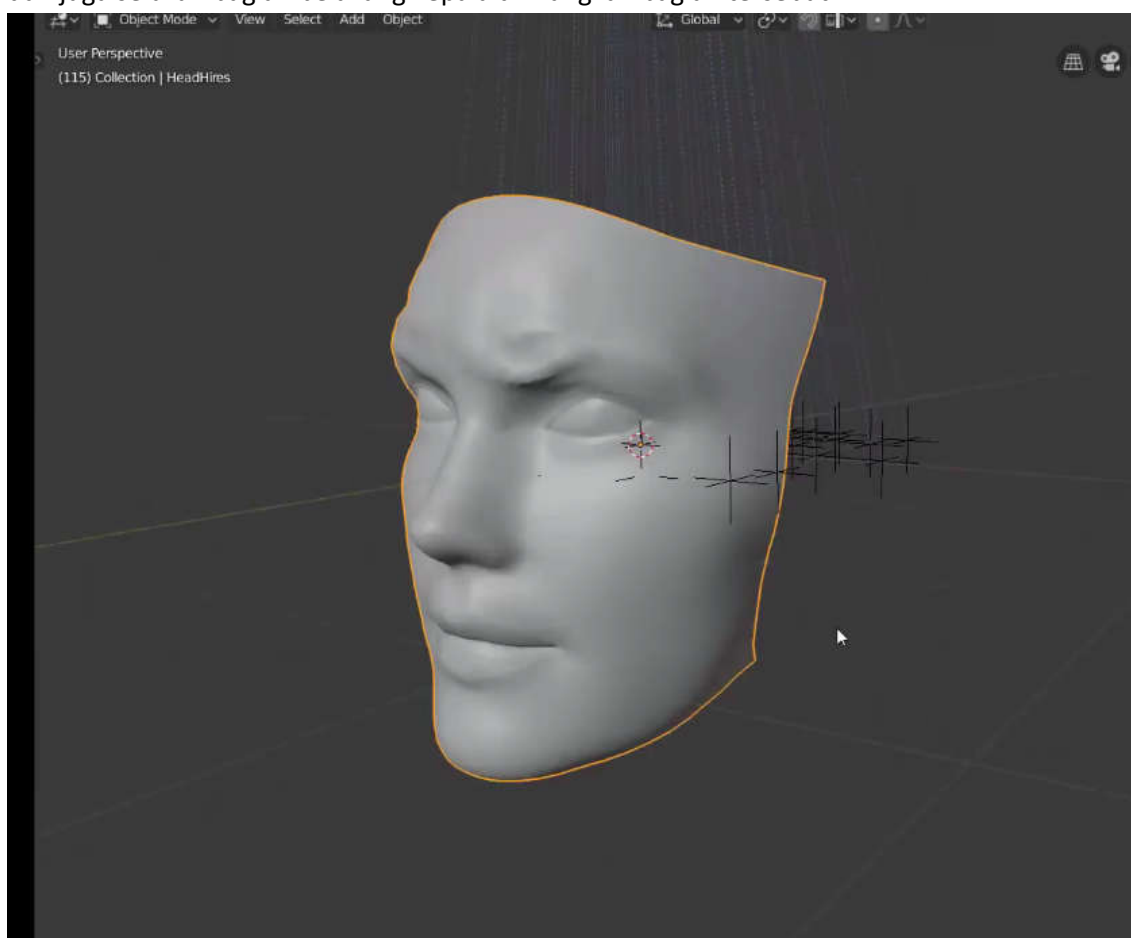


setelah semua pengaturan ekspor sudah siap, kita dapat menyimpan model ini ke direktori khusus dan apa yang akhirnya diekspor dari wajah adalah untuk file yang merupakan objek yang kita inginkan file material yang sesuai dan dua tekstur untuk kepala dan mouth

hasil dari import obj adalah seperti ini

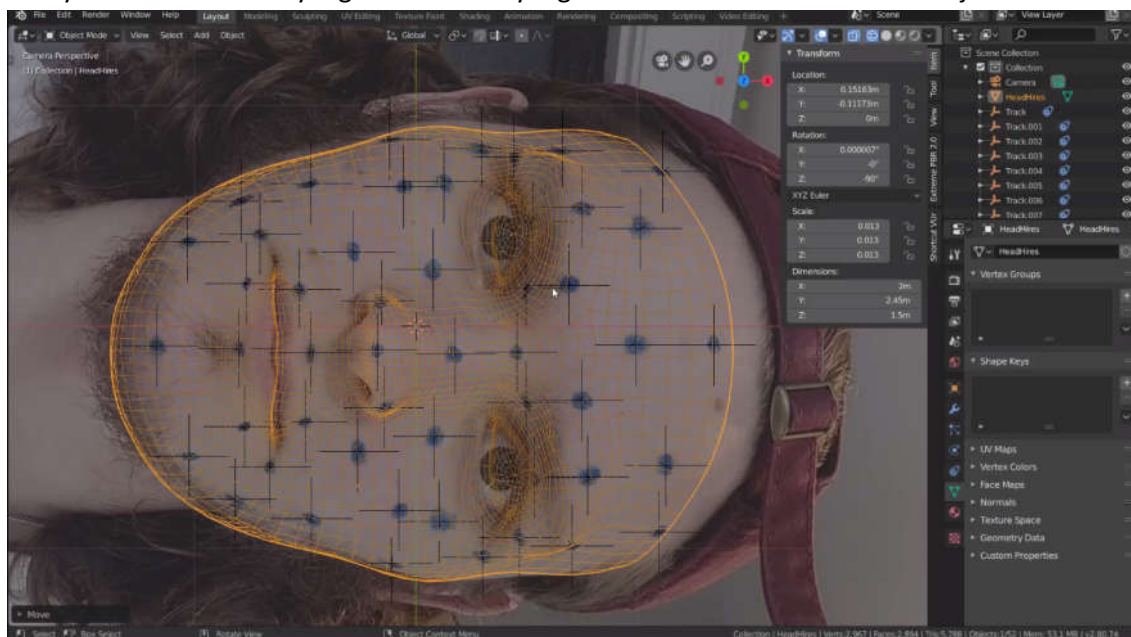


dan jelas kita memiliki banyak geometri tambahan di sini. yang kita tidak perlu adalah area seperti leher dan juga seluruh bagian belakang kepala dihilangkan bagian tersebut

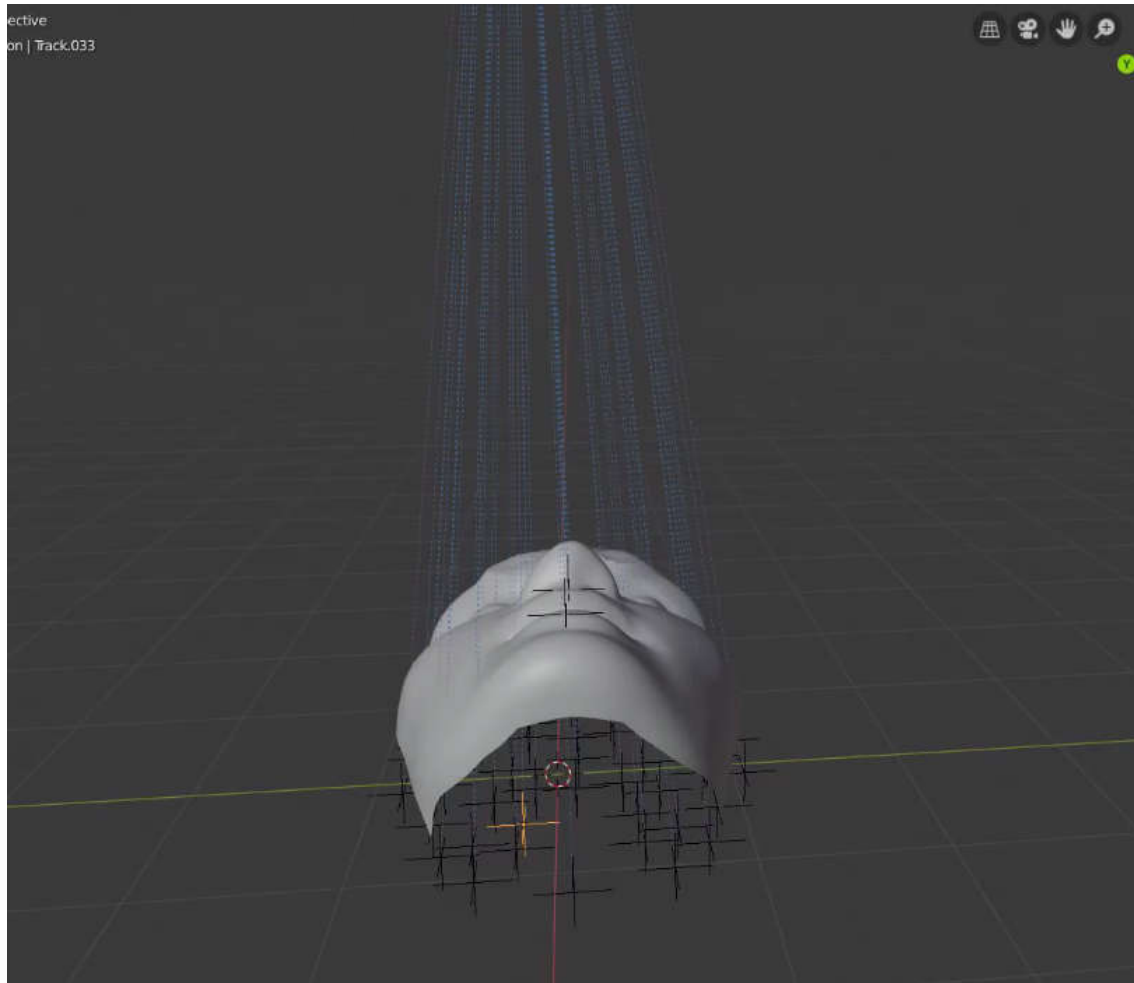


untuk memberikan kedalaman kosong, kita perlu

memastikan bahwa itu benar-benar selaras dengan rekaman kita akan melakukan penyalarsan ini dalam tampilan kamera beralih ke mode wireframe dengan menekan Z dan kemudian menyalarskan ini yang terbaik yang Anda bisa untuk menjadi lebih akurat

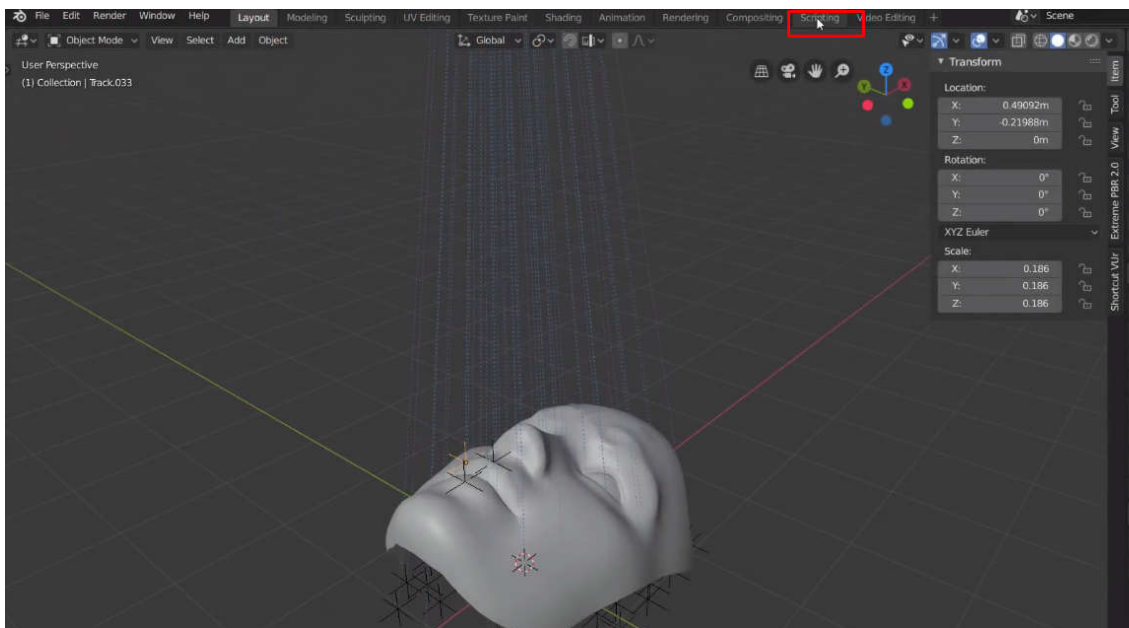


pastikan ini sudah teralign dengan baik

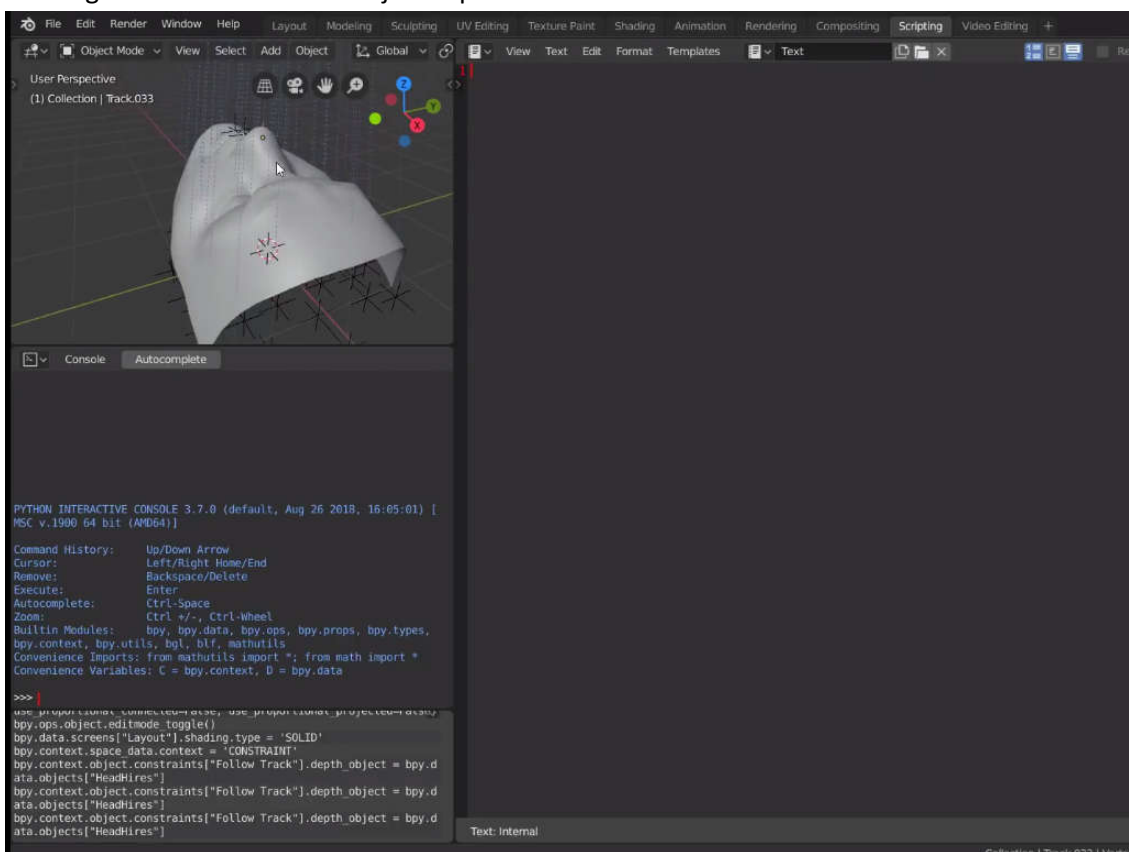


tentu saja ini membutuhkan waktu lama untuk melakukannya jadi kita ingin beberapa cara untuk mengotomatisasi ini dan pada titik ini kita dapat menggunakan yang disediakan skrip yang sudah disediakan.

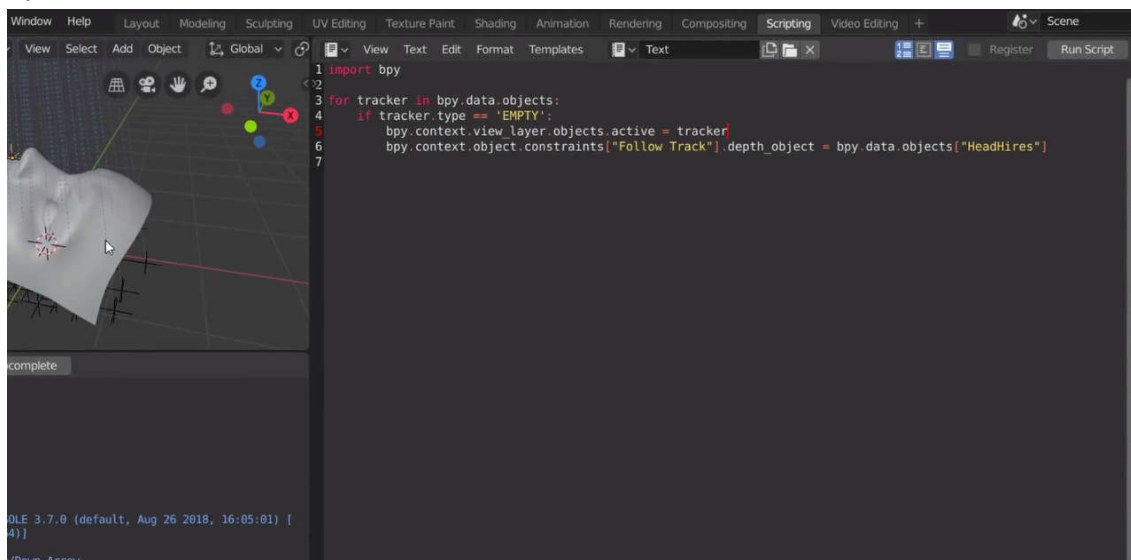
untuk melakukan ini pergi ke ruang kerja script



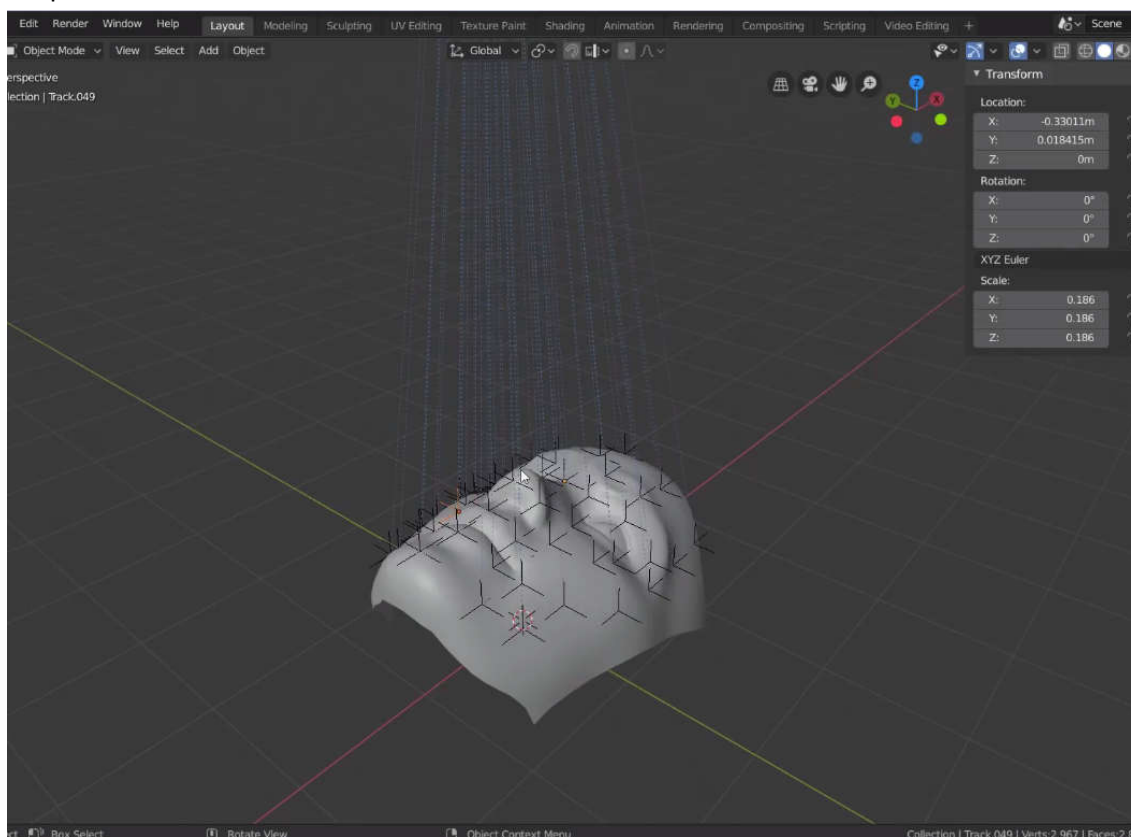
dan buat skrip baru dan bagi Anda yang tidak tahu cara membuat skrip apa pun jangan khawatir tentang ini karena ini akan menjadi super sederhana



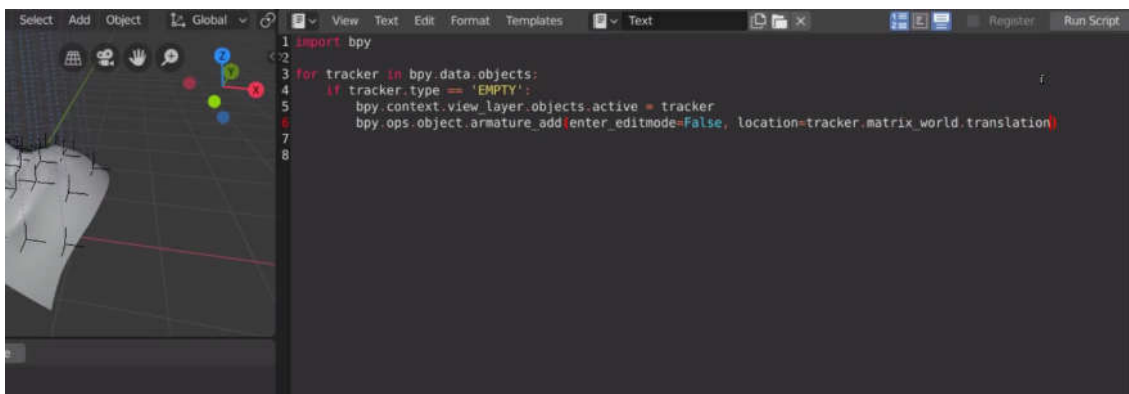
mulai dengan impor bpy sehingga kita dapat mereferensikan objek blender kita menggunakan Python



untuk memeriksa apakah skrip benar-benar berfungsi kita hanya perlu untuk menjalankan skrip

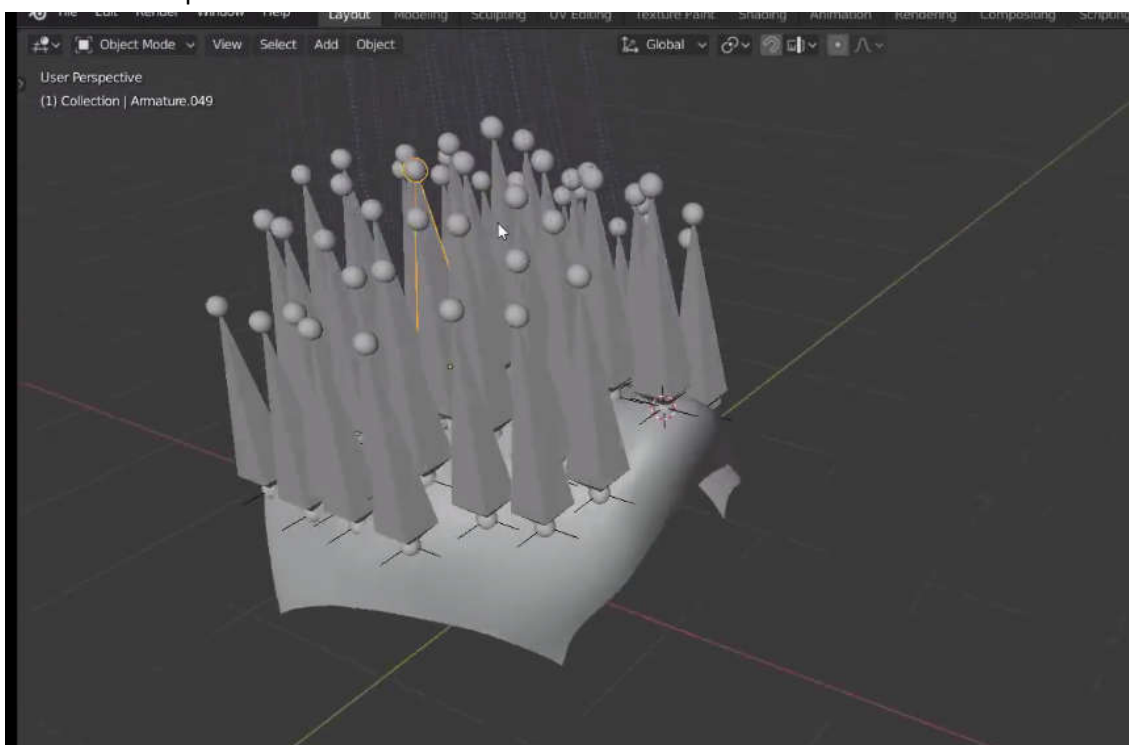


sekarang diproyeksikan pada mesh pada titik ini
buka skript sekali lagi

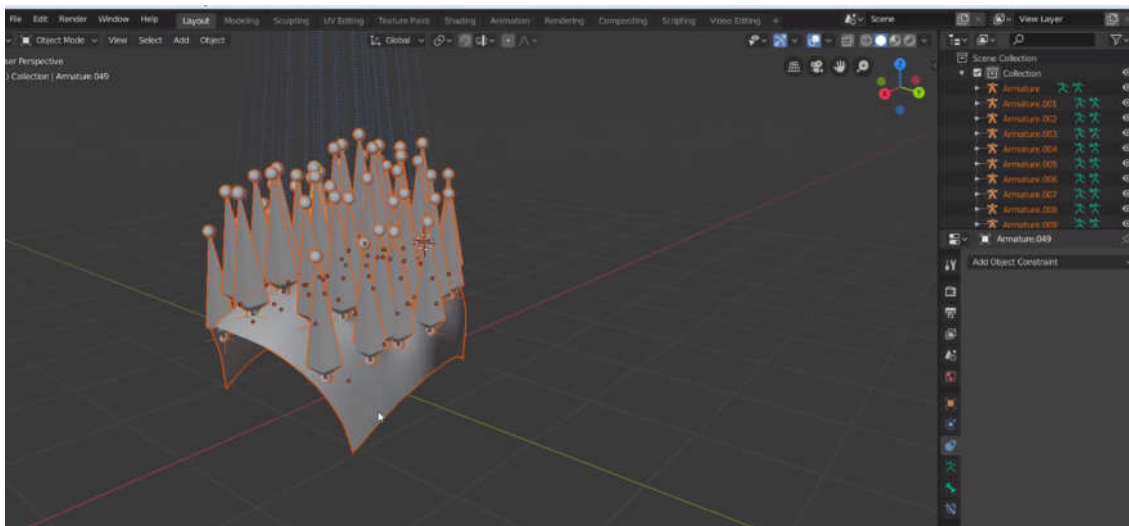


setelah itu jalankan script

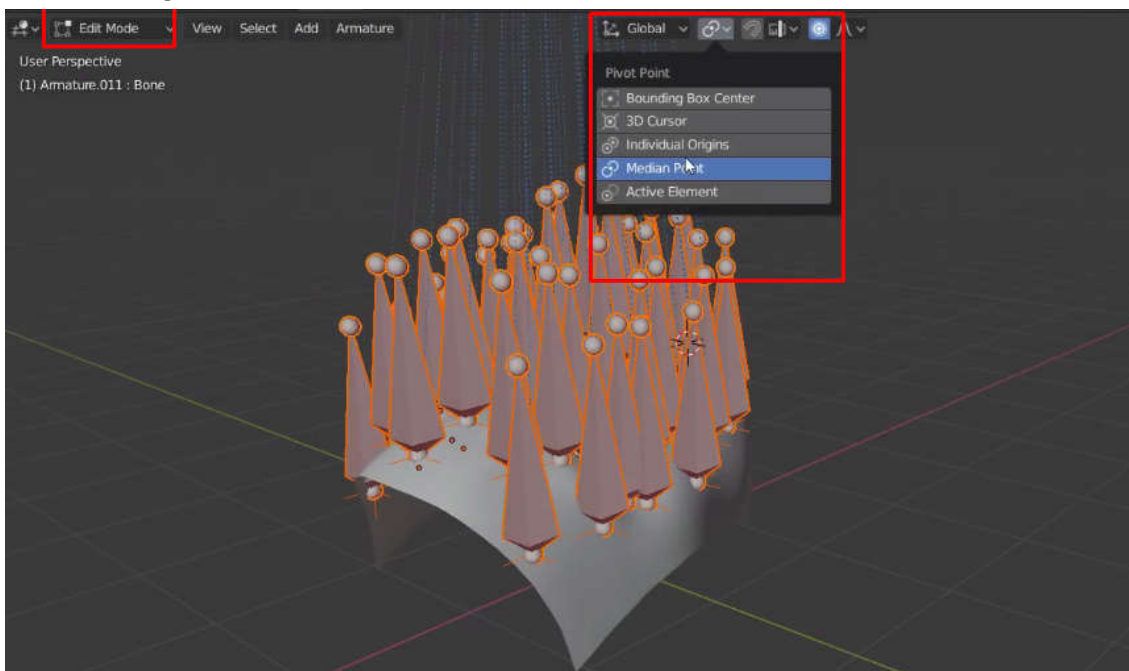
maka akan tercipta armature dalam mesh tersebut

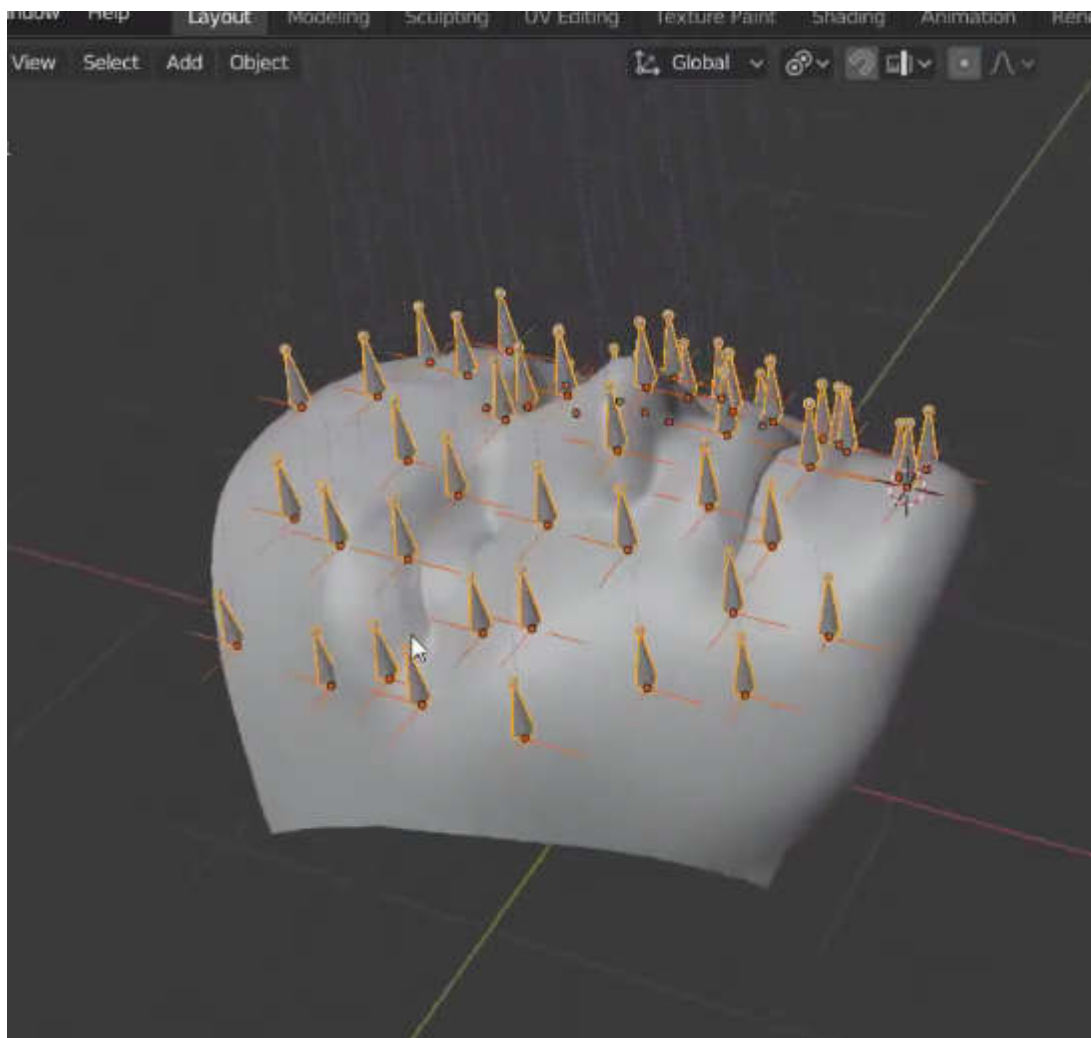


setelah itu select all armature

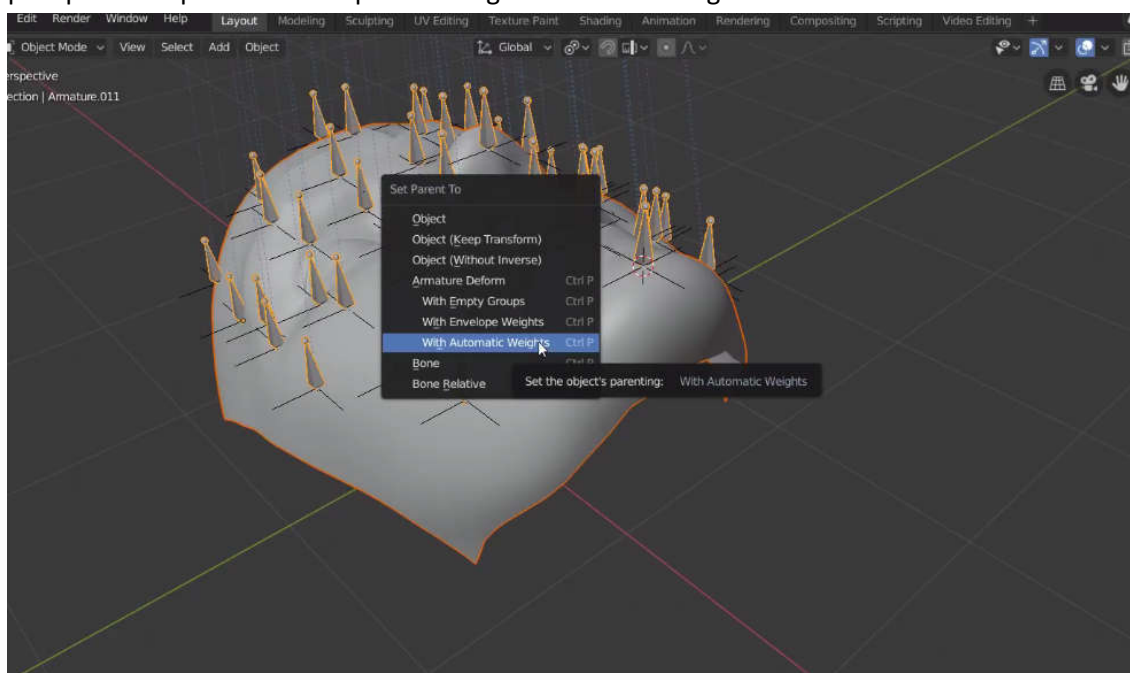


dan pilih joint setelah semua bone bergabung jadi satu bisa diskalakan dengan menggunakan individual origin



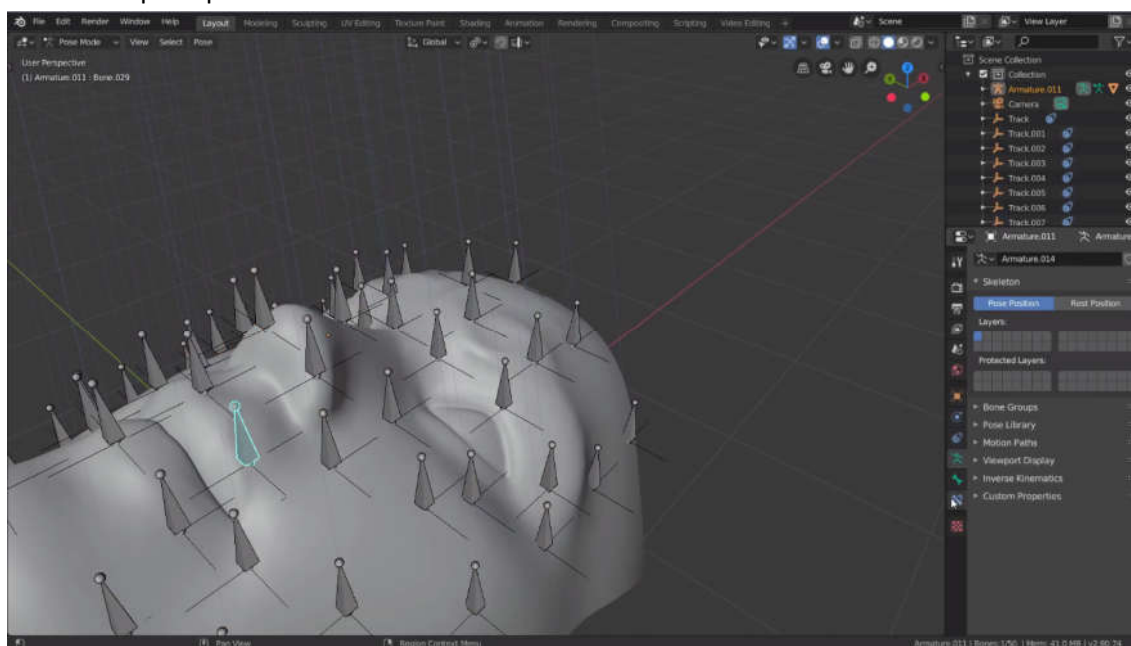


pilih parent > pilih bone lalu parent dengan automatic weight

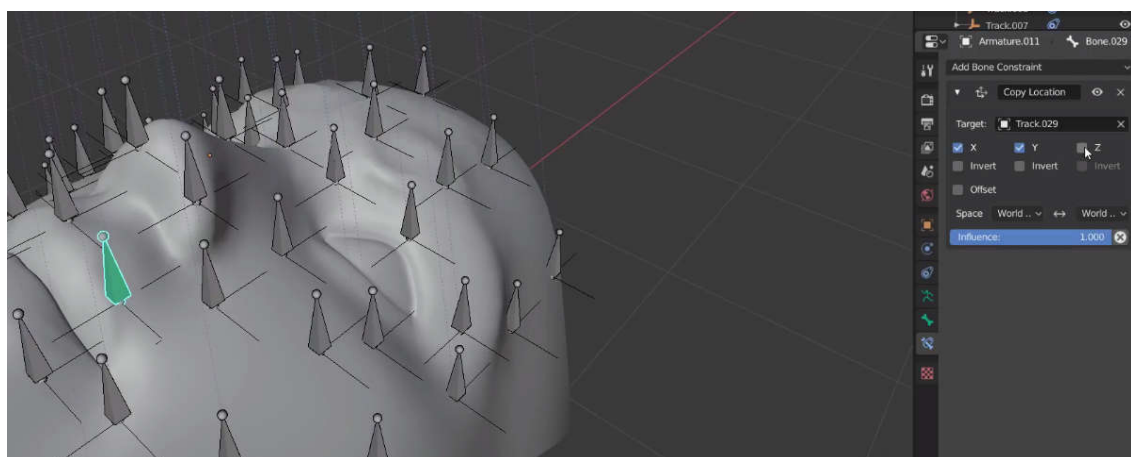
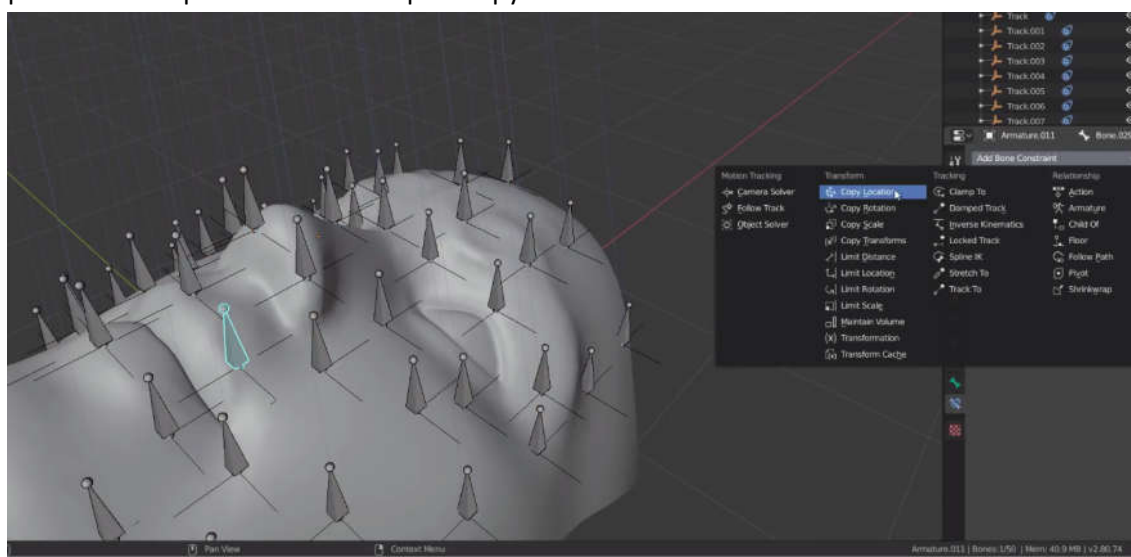


Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

setelah itu pada pose mode

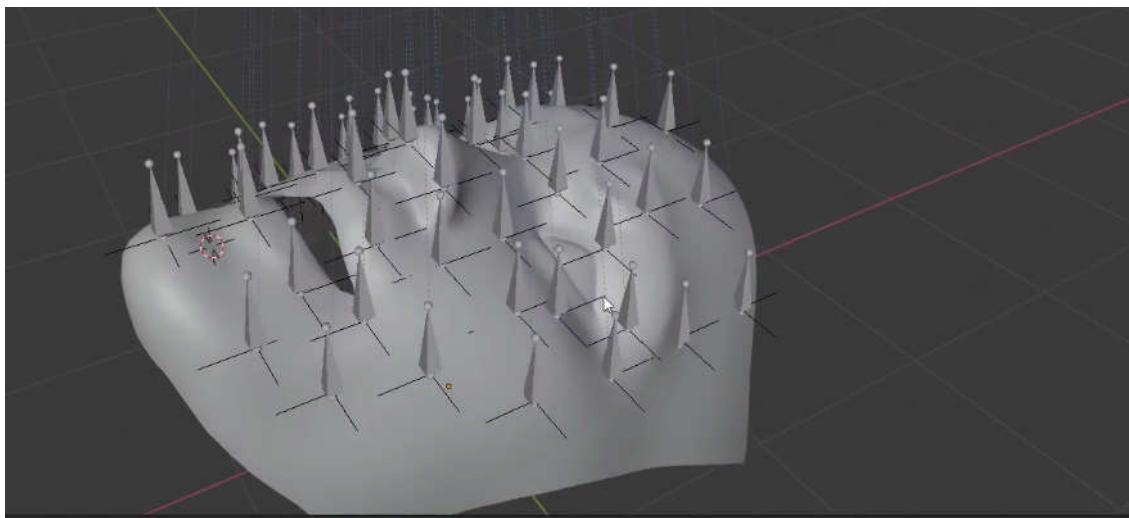


pilih bone lalu pada tab constraint pilih copy location



Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

lakukan untuk semua tulang
 sekarang gerakan wajah sudah dapat bergerak sesuai gerakan tulang yang menempel



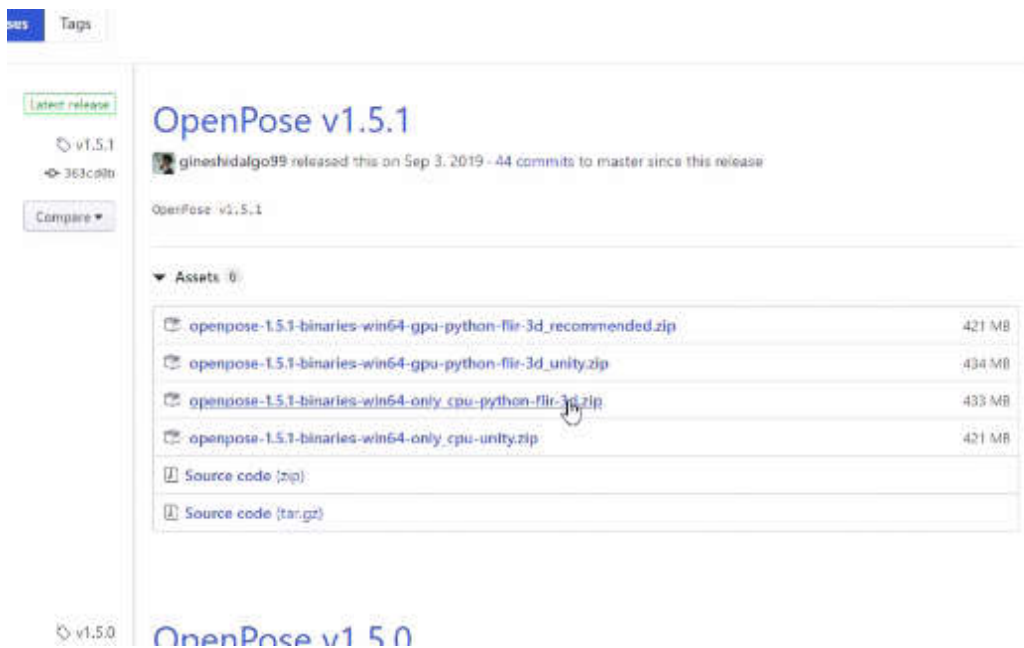
12.3 facial motion capture dengan openpose



openpose dapat digunakan untuk merubah wajah Anda dan menjadikan sebagai data motion capture dan kemudian jika Anda sudah memiliki data tersebut , Anda dapat membawanya ke dalam blender dan meletakkannya kedalam wajah karakter 3d .

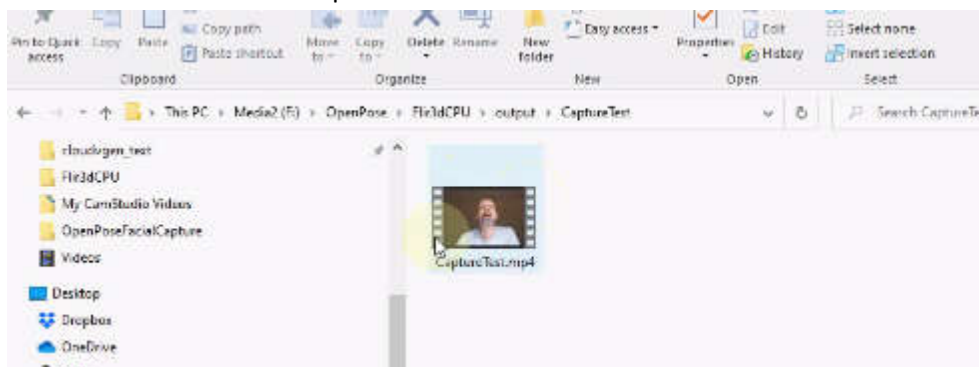
Dalam facial motion capture ini akan menggunakan metode yang dikembangkan oleh Nick Keeline <https://github.com/nkeeline> dia membuat script untuk menggabungkan antara openpose ke blender.

berikut ini adalah proses pembuatannya.

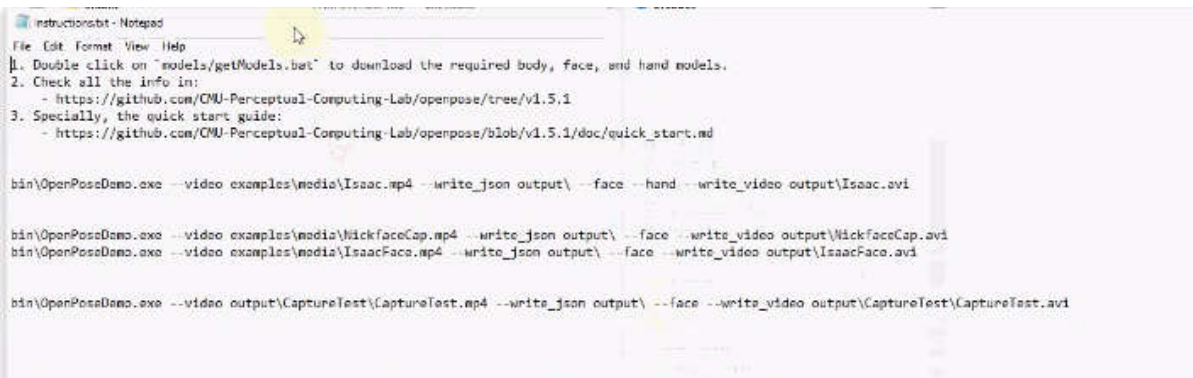


open pose adalah program gratis untuk penggunaan non-komersial dan juga sudah dikompilasi jadi Anda pergi ke halaman unduhan <https://github.com/CMU-Perceptual-Computing-Lab/openpose> jika Anda mengkliknya memiliki tautan unduhan untuk menginstal openpose

membuat folder bernama capturetest dan kita meletakkan



video mpeg-4 kita tentang tangkapan wajah yang dilakukan dengan ponsel yang akan kita lakukan adalah menjalankan video ini ke openpose dan membuat semua data untuk motion capture wajah,



lihat instruksi


```

F:\OpenPose>cd ..
F:\>cd Openpose
F:\OpenPose>cd bin3dcpu
F:\OpenPose\bin3dCPU>bin\OpenPoseDemo.exe --video output\CaptureTest\CaptureTest.mp4 --write_json output\CaptureTest --face --write_video output\CaptureTest\CaptureTest.avi
bin\OpenPoseDemo.exe --video output\CaptureTest\CaptureTest.mp4 --write_json output\CaptureTest --face --write_video output\CaptureTest\CaptureTest.avi

```

The Command Is:

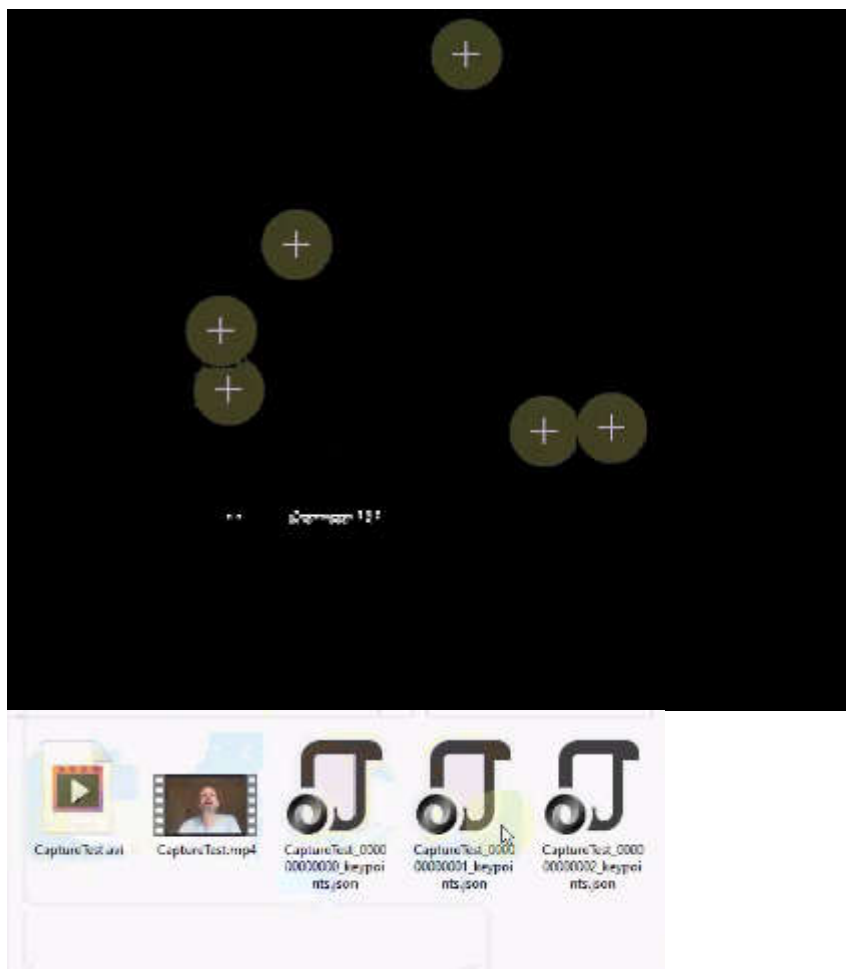
```

bin\OpenPoseDemo.exe --video output\CaptureTest\CaptureTest.mp4
--write_json output\CaptureTest
--face --write_video output\CaptureTest\CaptureTest.avi

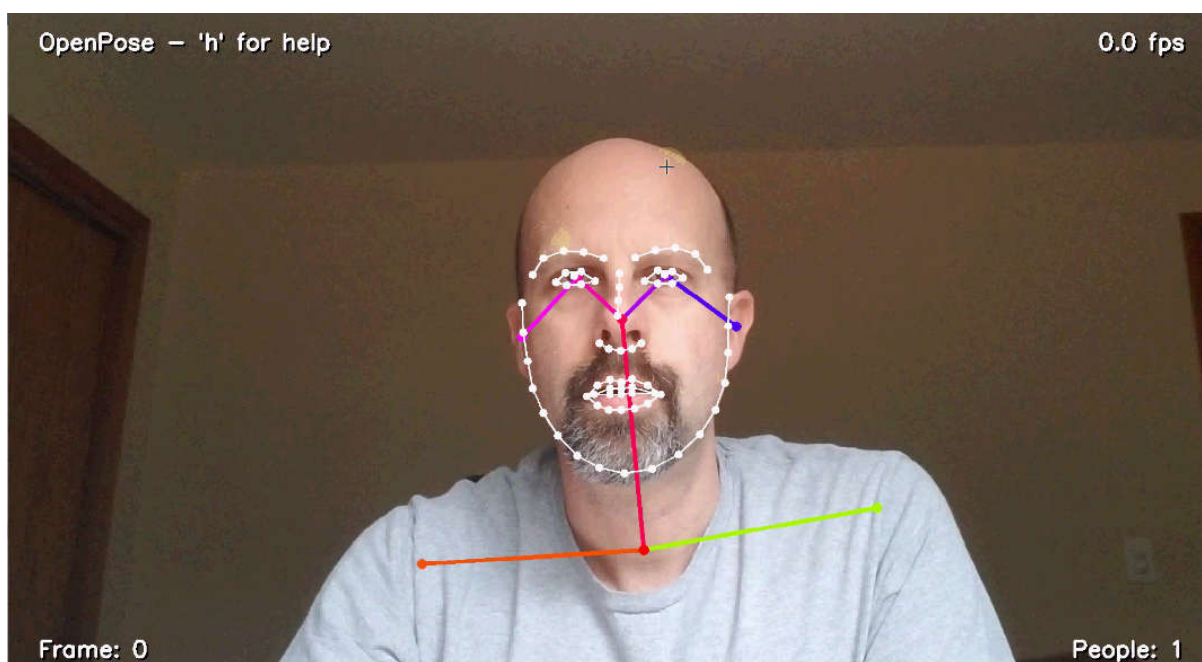
```

kita tekan ENTER itu akan mulai proses openpose perlu satu menit untuk menggambar frame pertama tetapi yang dilakukannya adalah menganalisis frame pertama dalam file dan menemukan semua titik wajah kita dan memplotnya ke layar dan mencari tahu di mana poin yang lain sedetik wajah kita akan muncul di





dapat anda lihat openpose membuat file AVI dan itu membuat JSON setiap frame



hasilnya seperti diatas.

Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

[Read the guide](#)

nkeeline / OpenPose-to-Blender-Facial-Capture-Transfer

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

This blender Python Script maps an OpenPose Facial Capture to a blender facial Rig [Edit](#)

Manage topics

7 commits 1 branch 0 packages 0 releases 1 contributor MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

File	Action	Time
Update README.md		Latest commit: yesterday
Facial_JSON.py	Add files via upload	yesterday
LICENSE	initial commit	yesterday
README.md	Update README.md	yesterday

README.md

OpenPose-to-Blender-Facial-Capture-Transfer

This blender Python Script maps an OpenPose Facial Capture to a blender facial Rig

Script Modifications

Open the script in Blender and modify lines as follows to make the script work on your rig:

jadi sekarang setelah kita menangkap gerakan wajah kita ke file JSON menggunakan openpose, kita dapat pergi dan mencoba dan membuatnya diimpor ke blender untuk melakukan itu.

gunakan skrip Python dan kita telah mempostingnya di sini di github <https://github.com/nkeeline/OpenPose-to-Blender-Facial-Capture-Transfer> , Anda dapat pergi dan ambil itu di halaman github untuk skrip nya

Product Solutions Open Source Pricing Search

nkeeline / OpenPose-to-Blender-Facial-Capture-Transfer Public

Code Issues 7 Pull requests Actions Projects Security Insights

master 1 branch 1 tag Go to file Code

nkeeline Update README.md 1c1cec7 on Jan 8 43 commits

File/Folder	Description	Time
BoneMapFiles	Update RigifyBoneMap.json	2 years ago
Examples/JSONExtraPersonIntoShot	making files for script install and minor change	2 years ago
Pictures	add pic	2 years ago
Source	up to ver 1.0.0 for release	2 years ago
Facial JSON-Auto Rig Pro.py	Adding Auto Rig Pro Version of Facial JSON File	2 years ago
Facial JSON.py	Update Facial JSON.py	3 years ago
LICENSE	Initial commit	3 years ago
OpenPoseToRig.py	making files for script install and minor change	2 years ago

ada deskripsi panjang tentang cara memodifikasi file untuk membuatnya berfungsi untuk rig Anda, Anda dapat mengunduh file Python dan membukanya di editor teks di dalam blender atau Anda cukup mengklik file tersebut memilihnya

```
#####
#####
#####
# OpenPose to BLender Script
# Source: https://github.com/nkeeline/OpenPose-to-Blender-Facial-Capture-Transfer/
# Version 1.3 Added eyelide jitter rejection, but keyframe blinks.
#####
#####
#####
```

```
import bpy
import json
import os.path
from os import path
import math
import mathutils
```

```
#####
#####
```

```
#####
# Setup variabes, change these to adapt the script to a rig.
#   Chages to the face rig needed for script to work:
#     Move Chine to be parented to lowerjaw instead of lowerFaceRig (in edit mode)
#     In edit mode scale all face bones in x and y to zero and zero out roll.
#####
#####
#####

FileIdentifier = "CaptureTest"
DestArmName = "Phillip"
#Go Into Edit mode and get delta x between earlobes
#we know the pixel distance between ears in open pse so knowing it for the character
#biveus the conervation between pixels and blender units.
EarLobeToEarLobedistanceInBlenderUnits = .17463
StartFrameNumber = 0
mouth_KeyFrame_Every_Nth_Frame = 3
eyes_KeyFrame_Every_Nth_Frame = 15
#number of frames to transfer, make really t
NumberOfFramesToTransfer = 1400
keyFrame = True
#eyelids are kind of inaccurate, so let's tie them together if
#your not going ot wink in the video, a little more natrualif true.
TieEyelidsTogether = True
#If the eyelid moves more than this amount of blender units, we keyframe,
#that way we catch fast blinks, but eliminate jitter in eyelid...
EyelidBlinkTHreshold = .01

#####
#####
#####
# Furntions for converting and manipulation of points.
#####
#####
#####

def combineRReliability(pt1, pt2):
    rel1 = pt1[2]
    rel2 = pt2[2]
    if rel1 < rel2:
        relOut = rel1
    else:
        relOut = rel2
```

```

return relOut

def GetPoint (Array, index):
    baseIndex = index*3
    x = float(Array[baseIndex])
    y = float(Array[baseIndex + 1])
    reliability = float(Array[baseIndex + 2])
    return [x,y,reliability]

    #this equation rotates the a point around a center, needed to take head tilt out of facial capture.
def rotatePoint(point, center, angle):
    #got equation from https://www.gamefromscratch.com/post/2012/11/24/GameDev-math-recipes-Rotating-one-point-around-another-point.aspx
    angle = math.radians(angle ) #Convert to radians
    rotatedX = math.cos(angle) * (point[0] - center[0]) - math.sin(angle) * (point[1]-center[1]) + center[0];
    rotatedY = math.sin(angle) * (point[0] - center[0]) + math.cos(angle) * (point[1] - center[1]) + center[1];
    return [rotatedX,rotatedY,point[2]]

def AverageTwoPoints(pt1, pt2):
    avgX = (pt1[0] + pt2[0])/2
    avgY = (pt1[1] + pt2[1])/2
    return [avgX, avgY,combineRReliability(pt1,pt2)]

def DifferenceBetweenPoint(pt1, pt2):
    diffX = (pt1[0] - pt2[0])
    diffY = (pt1[1] - pt2[1])
    return [diffX, diffY,combineRReliability(pt1,pt2)]

def InverseXandY(pt):
    x = pt[0]*-1
    y = pt[1]*-1
    return [x, y, pt[2]]

def InverseX(pt):
    x = pt[0]*-1
    y = pt[1]
    return [x, y, pt[2]]

def multiply(pt, val):
    x = pt[0]*val
    y = pt[1]*val

```

```
return [x, y, pt[2]]
```

```
#####
#####
#####
# This class get's data from open pose data provided and has a bunch of methods that
#get bone angles and position used to position a bone or get another bones rotation.
#####
#####
#####
```

```
class PersonJSONData:
```

```
def __init__(self, pose, face, EarLobeToEarLobedistanceInBlenderUnits, TieEyelidsTogether):
    self.poseStart = pose
    self.faceStart = face
    self.poseCurrent = pose
    self.faceCurrent = face
    self.TieEyelidsTogether = TieEyelidsTogether
    #distance is distance between ears
    self.distancBetweenEars = abs(GetPoint(pose, 18)[0] - GetPoint(pose, 18)[1])
    self.StartNosePosition = GetPoint(pose, 0)
    self.StartFaceNosePosition = GetPoint(face, 30)
    self.StartRightEyePosition = GetPoint(pose, 15)
    self.StartLeftEyePosition = GetPoint(pose, 16)
    #self.StartRightEyePosition = GetPoint(face, 68)
    #self.StartLeftEyePosition = GetPoint(face, 69)
    self.LastGoodHeadTilt = 0
    self.StartHeadTilt = self.getHeadTiltSideToSide()
    self.EarLobeToEarLobedistanceInBlenderUnits = EarLobeToEarLobedistanceInBlenderUnits
```

```
def SetCurrentPose(self, pose, face):
```

```
    self.poseCurrent = pose
    self.faceCurrent = face
```

```
    #given a point return it correct for head tilt at the current point in time.
```

```
def GetCurrentPointHeadTiltCorrected(self, Point):
```

```
    angle = self.getHeadTiltSideToSide()
    #point 8 is the nose we rotate around it by the tilt of the head
    PointCorrected = rotatePoint(GetPoint(self.faceCurrent, 8), Point,(angle*-1))
    return PointCorrected
```

```
    #this method returns a start point corrected for the head tilt at start, good for returning a point
```

```
    #In Space at start of capture corrected for head tilt at start of capture.
```

```

def GetStartPointHeadTiltCorrected(self, Point):
    angle = self.StartHeadTilt
    #point 8 is the nose we rotate around it by the tilt of the head
    PointCorrected = rotatePoint(GetPoint(self.faceStart, 8), Point,(angle*-1))
    return PointCorrected

def GetDistanceBetweenTwoCurrentPoints(self, Pnt1, Pnt2):
    x2 = Pnt2[0]
    x1 = Pnt1[0]
    y2 = Pnt2[1]
    y1 = Pnt1[1]
    dist = math.sqrt((x2 - x1)**2 + (y2 - y1)**2)
    return dist

def ConvertPixelPointToBlenderUnits(self, pnt):
    pntx = pnt[0] * (self.EarLobeToEarLobedistanceInBlenderUnits/self.distancBetweenEars)
    pnty = pnt[1] * (self.EarLobeToEarLobedistanceInBlenderUnits/self.distancBetweenEars)
    reliability = pnt[2]
    return [pntx,pnty,reliability]

def getHeadTiltSideToSide(self):
    #We get the tils by taking the angle difference between the left and right years
    leftEar = GetPoint(self.poseCurrent, 18)
    rightEar = GetPoint(self.poseCurrent, 17)
    deltaY = leftEar[1] - rightEar[1]
    deltaX = leftEar[0] - rightEar[0]
    if leftEar[2] < .1 or rightEar[2] < .1:
        tiltOut = self.LastGoodHeadTilt
    else:
        tilt = math.atan(deltaY/deltaX)
        tiltOut = math.degrees(tilt*-1)
        self.LastGoodHeadTilt = tiltOut
    return tiltOut

def getHeadTiltUpDown(self):
    #get the angle by using the nose vertical translation and approximating the distance
    #between the nose and the head pivot using the ear to ear distance from the start.
    CurrentNosePosition = GetPoint(self.poseCurrent, 0)
    deltaY = CurrentNosePosition[1] - self.StartNosePosition[1]
    tilt = math.atan(deltaY/self.distancBetweenEars)
    return math.degrees(tilt)

def getHeadTiltLeftRight(self):

```



```

#get the angle by using the nose horizontal translation and approximating the distance
#between the nose and the head pivot using the ear to ear distance from the start.
CurrentNosePosition = GetPoint(self.poseCurrent, 0)
deltaX = CurrentNosePosition[0] - self.StartNosePosition[0]
tilt = math.atan(deltaX/self.distancBetweenEars)
return math.degrees(tilt)

def getJawTiltUpDown(self):
    #To get the jaw tilt up down we rotate the point around the nose by the head tilt angle
    #Then we
    NoseToJawStartDist = self.GetDistanceBetweenTwoCurrentPoints(GetPoint(self.faceStart,
8),GetPoint(self.faceStart, 30))
    NoseToJawCurrentDist = self.GetDistanceBetweenTwoCurrentPoints(GetPoint(self.faceCurrent,
8),GetPoint(self.faceCurrent, 30))
    JawDelta = NoseToJawStartDist - NoseToJawCurrentDist
    #approximate the distance from the tip of your jaw to it's pivot by taking the distance between the
ears and subtracting an eighth
    PivottoEndOfJaw = self.distancBetweenEars - .125*self.distancBetweenEars
    Jawtilt = math.atan(JawDelta/PivottoEndOfJaw)
    return math.degrees(Jawtilt)

#This takes two points and averages them in the start frame and the current frame. It then
def getFacePosition(self, ptnum1, ptnum2, RelativeTo):
    ptAvgStart = AverageTwoPoints(GetPoint(self.faceStart, ptnum1),GetPoint(self.faceStart, ptnum2))
    ptAvgCurrent = AverageTwoPoints(GetPoint(self.faceCurrent, ptnum1),GetPoint(self.faceCurrent,
ptnum2))

    if RelativeTo == "lefteye":
        ptDiffCurrent = DifferenceBetweenPoint(ptAvgCurrent,GetPoint(self.poseCurrent, 16))
        #ptDiffCurrent = DifferenceBetweenPoint(ptAvgCurrent,GetPoint(self.faceCurrent, 69))
        ptDiffStart = DifferenceBetweenPoint(ptAvgStart,self.StartLeftEyePosition)
    elif RelativeTo == "righteye":
        ptDiffCurrent = DifferenceBetweenPoint(ptAvgCurrent,GetPoint(self.poseCurrent, 15))
        #ptDiffCurrent = DifferenceBetweenPoint(ptAvgCurrent,GetPoint(self.faceCurrent, 68))
        ptDiffStart = DifferenceBetweenPoint(ptAvgStart,self.StartRightEyePosition)
    else:
        ptDiffCurrent = DifferenceBetweenPoint(ptAvgCurrent,GetPoint(self.faceCurrent, 30))
        ptDiffStart = DifferenceBetweenPoint(ptAvgStart,self.StartFaceNosePosition)

    #StartPos = self.GetStartPointHeadTiltCorrected(ptDiffStart)
    #CurrentPos = self.GetCurrentPointHeadTiltCorrected(ptDiffCurrent)
    #DeltaPixels = DifferenceBetweenPoint(GetPoint(self.faceStart,
ptnum1),GetPoint(self.faceCurrent, ptnum1))

```

```

#DeltaPixels = DifferenceBetweenPoint(ptAvgStart, ptAvgCurrent)
DeltaPixels = DifferenceBetweenPoint(ptDiffStart, ptDiffCurrent)
#DeltaPixels = DifferenceBetweenPoint(StartPos,CurrentPos)
return DeltaPixels

def getLowerLipCenterPosition(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(57,66, "nose"))
    return multiply(output, correction)

def getLowerLipCenterLeftPosition(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(56,65, "nose"))
    return multiply(output, correction)

def getLowerLipCenterRightPosition(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(58,67, "nose"))
    return multiply(output, correction)

def getUpperLipCenterPosition(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(51,62, "nose"))
    return multiply(output, correction)

def getUpperLipCenterLeftPosition(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(52,63, "nose"))
    return multiply(output, correction)

def getUpperLipCenterRightPosition(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(50,61, "nose"))
    return multiply(output, correction)

def getUpperLipOuterLeftPosition(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(53,53, "nose"))
    return multiply(output, correction)

def getUpperLipOuterRightPosition(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(49,49, "nose"))
    return multiply(output, correction)

def getLowerLipOuterLeftPosition(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(55,55, "nose"))
    return multiply(output, correction)

def getLowerLipOuterRightPosition(self, correction):

```

```

output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(59,59, "nose"))
return multiply(output, correction)

def getLipLeftPosition(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(54,64, "nose"))
    return multiply(output, correction)

def getLipRightPosition(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(48,60, "nose"))
    return multiply(output, correction)

def getEyeBrowLeftOuter(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(25,26, "lefteye"))
    return multiply(output, correction)

def getEyeBrowLeftCenter(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(24,24, "lefteye"))
    return multiply(output, correction)

def getEyeBrowLeftInner(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(22,23, "lefteye"))
    return multiply(output, correction)

def getEyeBrowRightOuter(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(17,18, "righteye"))
    return multiply(output, correction)

def getEyeBrowRightCenter(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(19,19, "righteye"))
    return multiply(output, correction)

def getEyeBrowRightInner(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(20,21, "righteye"))
    return multiply(output, correction)

def getEyelidRight(self, correction):
    if self.TieEyelidsTogether:

```

```

    eyeR = self.getFacePosition(37,38, "righteye")
    eyeL = self.getFacePosition(43,44, "lefteye")
    output = self.ConvertPixelPointToBlenderUnits(AverageTwoPoints(eyeR,eyeL))
else:
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(37,38, "righteye"))
return multiply(output, correction)

```

```

def getEyelidLeft(self, correction):
    if self.TieEyelidsTogether:
        eyeR = self.getFacePosition(37,38, "righteye")
        eyeL = self.getFacePosition(43,44, "lefteye")
        output = self.ConvertPixelPointToBlenderUnits(AverageTwoPoints(eyeR,eyeL))
    else:
        output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(43,44, "lefteye"))
    return multiply(output, correction)

```

```

def getEyelidLowerRight(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(40,41, "righteye"))
    return multiply(output, correction)

```

```

def getEyelidLowerLeft(self, correction):
    output = self.ConvertPixelPointToBlenderUnits(self.getFacePosition(46,47, "lefteye"))
    return multiply(output, correction)

```

```

#####
#####
#####
# Code for iteration through frames and applying positions and angles to rig
#####
#####
#####

```

```

keepGoing = True
filenum = 0
FirstTime = True

```

```

print("")
print('Start of Everything')
print("")
DestArm = bpy.data.objects[DestArmName]

```

```

scene = bpy.context.scene
head_bone = DestArm.pose.bones["head"]
jaw_bone = DestArm.pose.bones["lowerJaw"]
lower_face = DestArm.pose.bones["lowerFaceRig"]
lower_lip_center = DestArm.pose.bones["LipLowerMiddle"]
lower_lip_center_l = DestArm.pose.bones["lLipLowerInner"]
lower_lip_center_r = DestArm.pose.bones["rLipLowerInner"]
upper_lip_center = DestArm.pose.bones["LipUpperMiddle"]
upper_lip_center_l = DestArm.pose.bones["lLipUpperInner"]
upper_lip_center_r = DestArm.pose.bones["rLipUpperInner"]

upper_lip_outer_r = DestArm.pose.bones["rLipUpperOuter"]
lower_lip_outer_r = DestArm.pose.bones["rLipLowerOuter"]
upper_lip_outer_l = DestArm.pose.bones["lLipUpperOuter"]
lower_lip_outer_l = DestArm.pose.bones["lLipLowerOuter"]

corner_lip_r = DestArm.pose.bones["rLipCorner"]
Corner_lip_l = DestArm.pose.bones["lLipCorner"]

eyebrow_outer_r = DestArm.pose.bones["rBrowOuter"]
eyebrow_center_r = DestArm.pose.bones["rBrowMid"]
eyebrow_inner_r = DestArm.pose.bones["rBrowInner"]

eyebrow_outer_l = DestArm.pose.bones["lBrowOuter"]
eyebrow_center_l = DestArm.pose.bones["lBrowMid"]
eyebrow_inner_l = DestArm.pose.bones["lBrowInner"]

eyelid_l = DestArm.pose.bones["Eyelid.l"]
eyelid_r = DestArm.pose.bones["Eyelid.r"]

eyelid_lower_l = DestArm.pose.bones["EyelidLower.l"]
eyelid_lower_r = DestArm.pose.bones["EyelidLower.r"]

#going to force Eyelid Keyframe if it's drastically different
PreviousEyelidValue = 0
CurrentFrame = 0

while keepGoing:
    numberString = str(filenum).zfill(12)
    filenum = filenum + 1
    mypath = bpy.path.abspath("//" + FileIdentifier + "_" + numberString + "_keypoints.json")

```

```
if path.exists(mypath):
```

```
    print("We are On File: " + str(filename))
```

```
    with open(mypath) as json_file:
```

```
        face1_dict = json.load(json_file)
```

```
        #face1 = '{"version":1.3,"people":[{"person_id":[-  
1],"pose_keypoints_2d":[880.441,483.283,0.877448,859.957,809.893,0.519232,509.674,777.567,0.3  
52327,0,0,0,0,0,1227.86,786.323,0.347931,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
59,415.538,0.896947,951.13,406.708,0.918684,698.036,468.529,0.858658,1045.25,447.943,0.87241  
2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
4,0.915689,716.382,513.514,0.889852,726.22,557.783,0.855176,740.976,598.773,0.825968,768.85,  
631.565,0.810219,801.642,661.078,0.833424,839.353,682.393,0.858697,886.901,687.312,0.848232,  
931.171,677.474,0.870025,967.242,649.601,0.884166,996.755,616.809,0.898078,1018.07,577.458,0  
.867331,1026.27,536.468,0.857222,1031.19,492.199,0.847157,1036.11,447.929,0.896024,1037.75,4  
05.299,0.8281,722.941,397.101,0.868592,745.895,375.787,0.908432,773.768,367.589,0.938036,808  
.2,367.589,0.91255,837.713,379.066,0.884488,895.099,370.868,0.846583,922.973,356.111,0.88807  
9,955.765,352.832,0.948471,985.278,356.111,0.902596,1008.23,377.426,0.933312,867.226,411.858  
,0.878828,868.866,438.092,0.921134,870.505,465.965,0.893925,870.505,492.199,0.882079,837.713  
,521.712,0.897178,854.109,524.991,0.945999,875.424,529.91,0.932919,893.46,523.351,0.929055,9  
13.135,516.793,0.900395,763.931,426.614,0.929163,781.967,416.777,0.934847,804.921,415.137,0.  
898184,826.236,428.254,0.905151,804.921,433.173,0.941087,781.967,433.173,0.943364,918.054,4  
20.056,0.959887,934.45,405.299,0.959568,957.405,402.02,0.963585,977.08,410.218,0.9434,960.68  
4,420.056,0.952746,937.729,421.696,0.940418,819.678,579.098,0.9055,842.632,570.9,0.90693,863.  
947,564.341,0.946204,880.343,567.621,0.923158,898.379,562.702,0.917559,919.694,565.981,0.944  
104,945.927,569.26,0.928293,921.333,590.575,0.944567,901.658,600.413,0.933296,883.622,602.05  
2,0.932713,865.587,602.052,0.951636,844.272,595.494,0.898185,829.515,579.098,0.921253,863.94  
7,579.098,0.947624,881.983,579.098,0.934409,900.018,577.458,0.946441,936.09,572.539,0.90652,  
900.018,577.458,0.937046,881.983,579.098,0.932065,863.947,579.098,0.967957,793.444,421.696,0  
.82772,945.927,408.579,0.919685],["hand_left_keypoints_2d":[],"hand_right_keypoints_2d":[],"pose  
_keypoints_3d":[],"face_keypoints_3d":[],"hand_left_keypoints_3d":[],"hand_right_keypoints_3d":[]  
}}}'
```

```
        #face1_dict = json.loads(face1)
```

```
        #print(face1_dict.keys())
```

```
        people = face1_dict['people']
```

```
        #print(people[0].keys())
```

```
        ##dict_keys(['person_id', 'pose_keypoints_2d', 'face_keypoints_2d', 'hand_left_keypoints_2d',  
'hand_right_keypoints_2d', 'pose_keypoints_3d', 'face_keypoints_3d', 'hand_left_keypoints_3d',  
'hand_right_keypoints_3d'])
```

```

#print(json.dumps(people[0]))
myPerson = people[0]

pose = myPerson['pose_keypoints_2d']
face = myPerson['face_keypoints_2d']

if FirstTime:
    p1 = PersonJSONData(pose, face, EarLobeToEarLobedistanceInBlenderUnits,
TieEyelidsTogether)
    FirstTime = False
else:
    p1.SetCurrentPose(pose, face)

#print(GetPoint(pose,0))
head_angle =
mathutils.Euler((math.radians(p1.getHeadTiltUpDown()),math.radians(p1.getHeadTiltLeftRight()),ma
th.radians(p1.getHeadTiltSideToSide()))), 'XYZ')

head_bone.rotation_mode = 'XYZ'
head_bone.rotation_euler = head_angle

jaw_bone.rotation_mode = 'XYZ'
jaw_bone.rotation_euler.z = math.radians(p1.getJawTiltUpDown())

lower_face.rotation_mode = 'XYZ'
lower_face.rotation_euler.x = math.radians(p1.getJawTiltUpDown())

Offset = p1.getLowerLipCenterPosition(1)
lower_lip_center.location.x = Offset[0]*-1
lower_lip_center.location.z = Offset[1]*-1

Offset = p1.getLowerLipCenterLeftPosition(1)
lower_lip_center_l.location.x = Offset[0]
lower_lip_center_l.location.z = Offset[1]

Offset = p1.getLowerLipCenterRightPosition(1)
lower_lip_center_r.location.x = Offset[0]
lower_lip_center_r.location.z = Offset[1]

Offset = p1.getUpperLipCenterPosition(1)
upper_lip_center.location.x = Offset[0]
upper_lip_center.location.z = Offset[1]

```

```

Offset = p1.getUpperLipCenterLeftPosition(1)
upper_lip_center_l.location.x = Offset[0]
upper_lip_center_l.location.z = Offset[1]

```

```

Offset = p1.getUpperLipCenterRightPosition(1)
upper_lip_center_r.location.x = Offset[0]
upper_lip_center_r.location.z = Offset[1]

```

```

Offset = p1.getUpperLipOuterRightPosition(1)
upper_lip_outer_r.location.x = Offset[0]
upper_lip_outer_r.location.z = Offset[1]

```

```

Offset = p1.getLowerLipOuterRightPosition(1)
lower_lip_outer_r.location.x = Offset[0]
lower_lip_outer_r.location.z = Offset[1]

```

```

Offset = p1.getUpperLipOuterLeftPosition(1)
upper_lip_outer_l.location.x = Offset[0]
upper_lip_outer_l.location.z = Offset[1]

```

```

Offset = p1.getLowerLipOuterLeftPosition(1)
lower_lip_outer_l.location.x = Offset[0]
lower_lip_outer_l.location.z = Offset[1]

```

```

Offset = p1.getLipRightPosition(1)
corner_lip_r.location.x = Offset[0]
corner_lip_r.location.z = Offset[1]

```

```

Offset = p1.getLipLeftPosition(1)
Corner_lip_l.location.x = Offset[0]
Corner_lip_l.location.z = Offset[1]

```

```

eyecorrection = 2
Offset = p1.getEyeBrowLeftOuter(1)
eyebrow_outer_r.location.x = Offset[0]
eyebrow_outer_r.location.z = Offset[1]

```

```

Offset = p1.getEyeBrowLeftCenter(1)

```



```
eyebrow_center_r.location.x = Offset[0]
eyebrow_center_r.location.z = Offset[1]
```

```
Offset = p1.getEyeBrowLeftInner(1)
eyebrow_inner_r.location.x = Offset[0]
eyebrow_inner_r.location.z = Offset[1]
```

```
Offset = p1.getEyeBrowRightOuter(1)
eyebrow_outer_l.location.x = Offset[0]
eyebrow_outer_l.location.z = Offset[1]
```

```
Offset = p1.getEyeBrowRightCenter(1)
eyebrow_center_l.location.x = Offset[0]
eyebrow_center_l.location.z = Offset[1]
```

```
Offset = p1.getEyeBrowRightInner(1)
eyebrow_inner_l.location.x = Offset[0]
eyebrow_inner_l.location.z = Offset[1]
```

```
Offset = p1.getEyelidLeft(10)
eyelid_l.location.z = Offset[1]
```

```
Offset = p1.getEyelidRight(10)
eyelid_r.location.z = Offset[1]
```

```
Offset = p1.getEyelidLowerLeft(1)
eyelid_lower_l.location.z = Offset[1]
```

```
Offset = p1.getEyelidLowerRight(1)
eyelid_lower_r.location.z = Offset[1]
```

```
#print(p1.getHeadTiltSideToSide())
#print(p1.getHeadTiltUpDown())
#print(p1.getHeadTiltLeftRight())
```

#keyfram if the eyelid changes a LOT. That way we catch blinks.

```
AmountTheEyelidChanged = abs(eyelid_l.location.z - PreviousEyelidValue)
```

```
if AmountTheEyelidChanged > EyelidBlinkTHreshold and keyFrame:
```

```
    eyelid_r.keyframe_insert(data_path='location',frame= StartFrameNumber + CurrentFrame)
```

```
    eyelid_l.keyframe_insert(data_path='location',frame= StartFrameNumber + CurrentFrame)
```

```
PreviousEyelidValue = eyelid_l.location.z
```

```

#if current frame is an exact multiple of Every Nth Frame
if (CurrentFrame % mouth_KeyFrame_Every_Nth_Frame == 0) and (keyFrame):
    head_bone.keyframe_insert(data_path='rotation_euler',frame= StartFrameNumber +
CurrentFrame)
    jaw_bone.keyframe_insert(data_path='rotation_euler',frame= StartFrameNumber +
CurrentFrame)
    lower_face.keyframe_insert(data_path='rotation_euler',frame= StartFrameNumber +
CurrentFrame)
    lower_lip_center.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)
    lower_lip_center_r.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)
    lower_lip_center_l.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)
    upper_lip_center.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)
    upper_lip_center_r.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)
    upper_lip_center_l.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)

    upper_lip_outer_r.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)
    lower_lip_outer_r.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)
    upper_lip_outer_l.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)
    lower_lip_outer_l.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)

    corner_lip_r.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)
    Corner_lip_l.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)

#if current frame is an exact multiple of Every Nth Frame
if (CurrentFrame % eyes_KeyFrame_Every_Nth_Frame == 0) and (keyFrame):
    eyebrow_outer_l.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)
    eyebrow_center_l.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)

```

```

    eyebrow_inner_l.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)

    eyebrow_outer_r.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)
    eyebrow_center_r.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)
    eyebrow_inner_r.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)

    eyelid_r.keyframe_insert(data_path='location',frame= StartFrameNumber + CurrentFrame)
    eyelid_l.keyframe_insert(data_path='location',frame= StartFrameNumber + CurrentFrame)

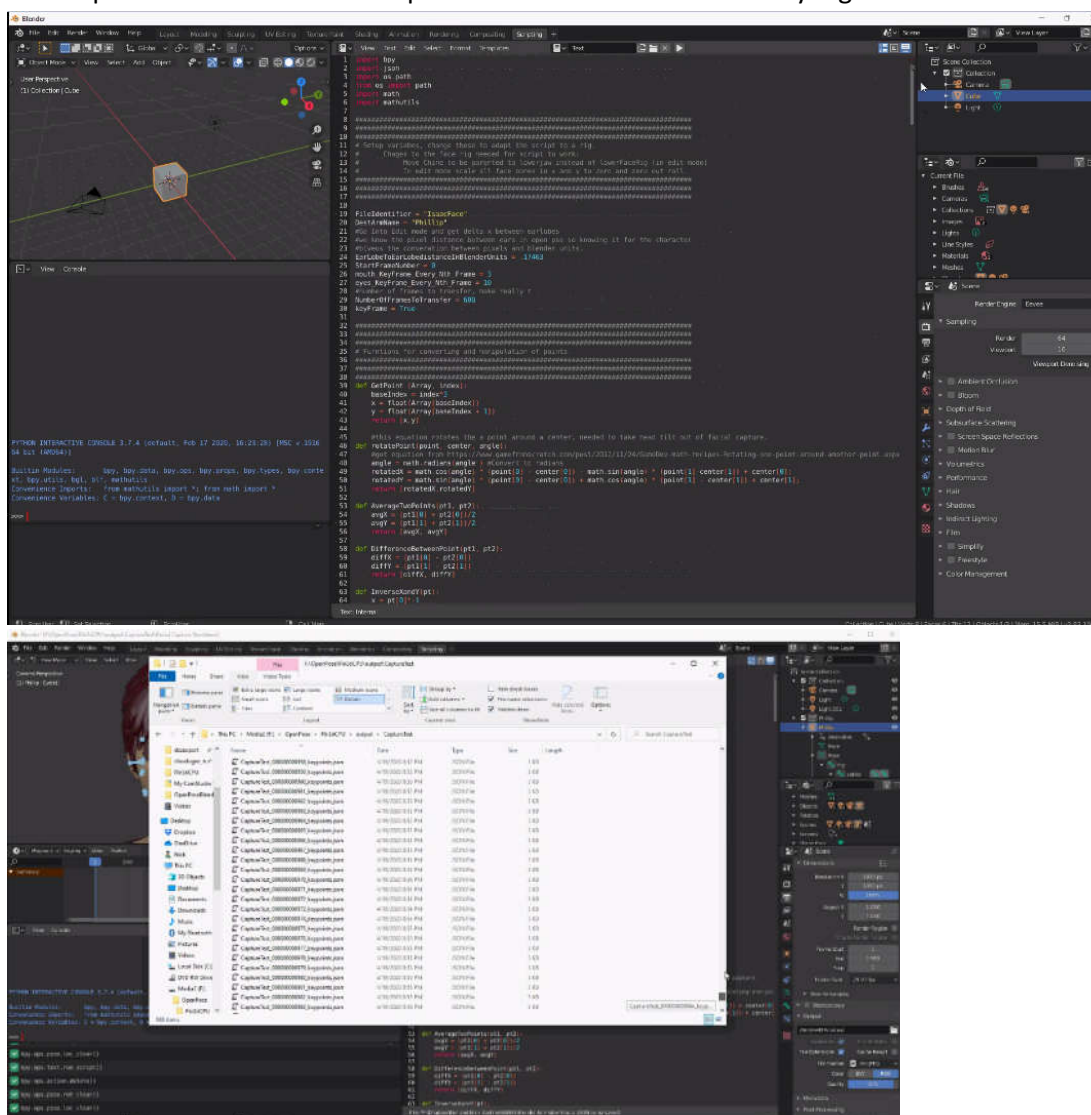
    eyelid_lower_r.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)
    eyelid_lower_l.keyframe_insert(data_path='location',frame= StartFrameNumber +
CurrentFrame)

CurrentFrame = CurrentFrame + 1
bpy.context.scene.frame_set(StartFrameNumber + CurrentFrame)
if CurrentFrame > NumberOfFramesToTransfer:
    keepGoing = False
else:
    keepGoing = False

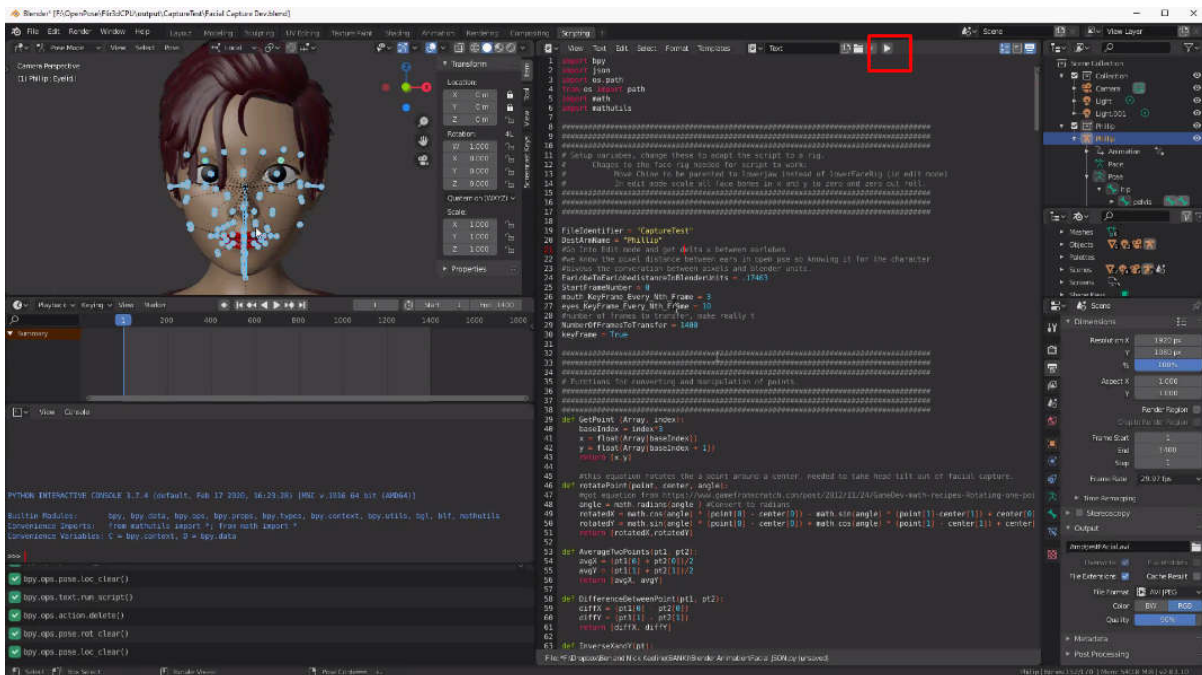
```

di dalam blender membuka file baru Anda dapat membuka skrip dan membuat teks baru dan pastekan di dalamnya Anda sekarang memiliki skrip yang dapat Anda modifikasi agar berfungsi dengan rig Anda dan setelah selesai dengan modifikasi, Anda dapat menekan tombol Jalankan untuk menjalankan skrip dan menerapkan file JSON ke rig Anda jadi sekarang bahwa kita telah selesai dengan pengambilan pose terbuka, kita memiliki direktori kita dengan dua file video kita, salah satu

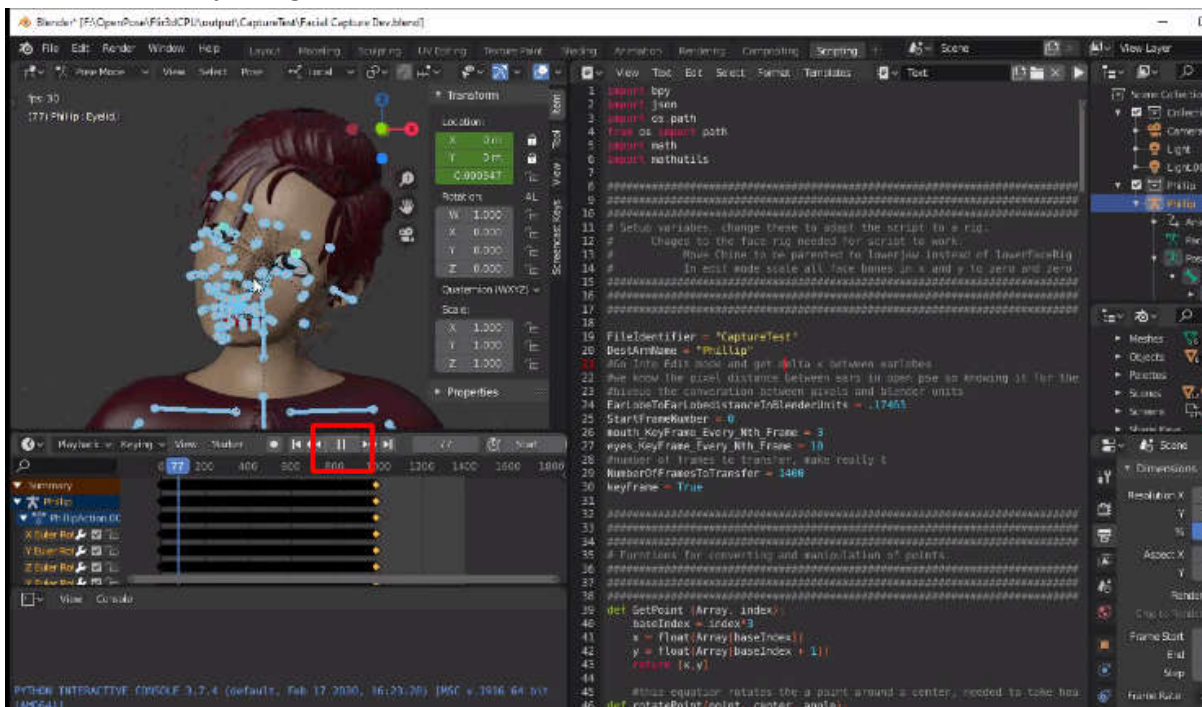
file input dan satu file output dan semua file JSON yang dihasilkan oleh openpose



Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)



animasi ini bekerja dengan baik



Motion capture dalam animasi karakter 3D (setiyo Adi Nugroho)

Daftar Pustaka

1. Lars Mündermann, Stefano Corazza and Thomas P Andriacch. **The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications**. Journal of NeuroEngineering and Rehabilitation 2006
2. Hyewon Song, Suwoong Heo, Jiwoo Kang, Sanghoon Lee, **3D Character Animation: A Brief Review** Journal of International Society for Simulation Surgery 2015
3. TONY MULLEN, **Introducing Character Animation with Blender** ,WILEY PUBLISHING, INC 2007
- 4.
5. Pedro Nogueira, **Motion Capture Fundamentals A Critical and Comparative Analysis on Real-World Applications**, Faculdade de Engenharia da Universidade do Porto e Novembro de 2011
6. Ashish Sharma, Mukesh Agarwal, Anima Sharma, Pankhuri Dhuria, **MOTION CAPTURE PROCESS, TECHNIQUES AND APPLICATIONS** , International Journal on Recent and Innovation Trends in Computing and Communication 2007.
7. M. Rahul, **Review on Motion Capture Technology**, Global journal of computer science and technology, 2018
8. Steve Roberts, **Character Animation in 3D: Use traditional drawing techniques to produce stunning CGI animation** Focal Press , 2004
9. Rick Parent, **Computer Animation Complete All-in-One: Learn Motion Capture, Characteristic, Point-Based, and Maya Winning Techniques**, Elsevier, 2010
10. Michael Meredith, Steve C. Maddock, **Motion Capture File Formats Explained**, 2001
11. Naoki Numaguchi, Atsushi Nakazawa, Takaaki Shiratori, Jessica K. Hodgins , **A Puppet Interface for Retrieval of Motion Capture Data**, Symposium on Computer Animation (2011)
12. <https://github.com/nkeeline/OpenPose-to-Blender-Facial-Capture-Transfer>
13. <https://www.cgmatter.com/>
14. <https://research.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html>



BIODATA PENULIS

Penulis adalah Dosen dari Universitas Sains dan Teknologi Komputer (Universitas STEKOM) di Program studi Komputer Grafis. yang mengajar dibidang animasi dan modeling 3 dimensi, video editing dan augmented reality, selain itu juga telah melakukan beberapa penelitian di bidang animasi dan modeling 3 Dimensi



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 978-623-8120-00-0 (PDF)

