



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

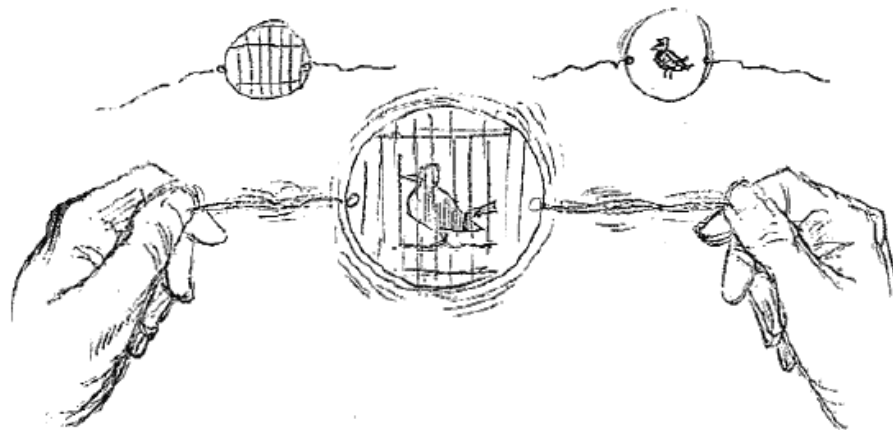
2.1. Animasi

Animasi memiliki arti jiwa atau hidup. Animasi merupakan rangkaian gambar yang disusun secara berurutan dan menciptakan ilusi berupa gerakan. Menurut Chong (2008) animasi memiliki hubungan dengan realisme. Realisme dalam hal ini artinya menakutkan penonton untuk mempercayai segala fantasi atau imajinasi yang tercipta dalam dunia animasi. Secara tidak langsung animasi memberikan hidup pada objek dan membuat objek tersebut *believable* ketika dilihat penonton (hlm 22-23).

Konsep dari animasi sudah ada jauh sebelum adanya televisi dan film. Dalam sejarahnya, dari dulu manusia berusaha menciptakan ilusi gerakan ini dengan berbagai cara dari gambar hewan di dinding gua yang memiliki kaki lebih dari 4 sampai penggambaran figur dewa-dewi yang sedang melakukan suatu gerakan (Williams, 2009, hlm 12). Pergerakan ilusi bisa terjadi karena adanya prinsip *The Persistence of Vision*. *The Persistence of Vision* merupakan sebuah prinsip dimana mata manusia dapat menyimpan dalam waktu singkat gambar yang manusia lihat. Gambar berikutnya yang dilihat oleh mata manusia akan menimpa gambar yang sebelumnya telah tersimpan sehingga menciptakan ilusi gerakan (Williams, 2009, hlm 13).

Dari prinsip *The Persistence of Vision* animasi berkembang menjadi mainan ilusi optik diantaranya *Thaumatrope*, *Phenakistoscope*, *Praxinoscope* sampai pada

ke *The Flipper Book* (Williams, 2009, hlm 14-15). Selanjutnya perkembangan animasi mulai mengarah ke media yang lebih serius yaitu film. Contoh film-film ialah *Humorous Phases of Funny Faces* karya Stuart Blackton, *Little Nemo in Slumberland* dan *Gertie the Dinosaur* karya Winsor McCay sampai teknik *rotoscope* yang ditemukan oleh Max Fleischer yang menjadi cikal bakal dari teknik *motion capture* (Menache, 2011, hlm 3). Pada awalnya animasi dalam media film sebagai bentuk kesenian dan bukan komoditi dagang. Munculnya film-film animasi ini membuat animasi sebagai industri lahir dan menghasilkan berbagai perkembangan baru dalam industri film animasi.



Gambar 2. 1. Ilustrasi Permainan Ilusi

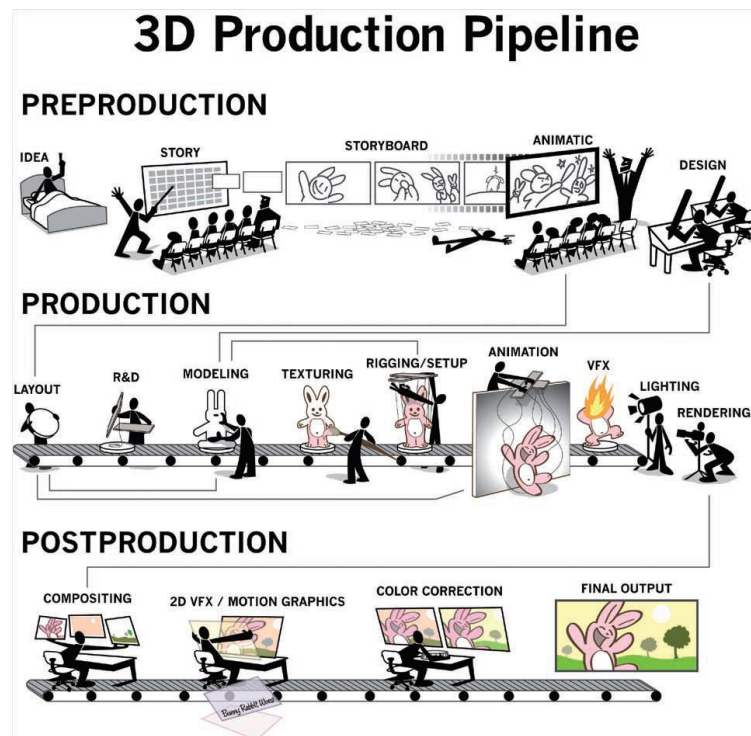
(Animator's Survival Kit, William, 2002, hlm 13)

Salah satu perkembangannya dapat dilihat dari tokoh Felix The Cat yang mampu menghadirkan sifat serta *personality* –nya kepada penonton. Kesuksesan tokoh Felix ini diikuti oleh Disney dengan menghadirkan Mickey Mouse dalam

film *Steamboat Willie*. Disney memainkan peran penting dalam perkembangan animasi di dunia. *Steamboat Willie* yang muncul pada tahun 1928 merupakan film animasi suara pertama di dunia. Tahun 1932 Disney menghadirkan *Flowers and Trees* film animasi berwarna pertama dan setahun setelahnya Disney memunculkan *Three Little Pigs* yang merupakan animasi dengan pengembangan *personality* yang lebih dalam pada karakter-karakternya. 4 tahun berikutnya Disney menciptakan *Snow White and the Seven Dwarfs* yang menjadi film animasi panjang pertama di dunia. Munculnya film *Snow White and the Seven Dwarfs* ini melahirkan era keemasan bagi industri animasi (Williams, 2009. Hlm 17-20).

2.1.1. Animasi 3D

3d merupakan salah satu perkembangan dari dunia animasi. Menurut Bernhardt (2015) Animasi 3D berbicara soal ruang atau *space* (hlm. xvii). Animasi 3D muncul karena adanya perkembangan teknologi terutama dalam bidang *computer graphic* atau CG. Perkembangan teknologi dalam dunia animasi 3d membuat adanya metode serta teknik-teknik baru dalam industri animasi 3d. Dalam industri sendiri, animasi 3D memiliki peran besar pada 3 bidang industri yaitu industri hiburan, keilmuan dan bidang lain seperti *augmented reality*. Dalam proses pembuatannya ada 3 tahapan dalam proses pembuatan animasi 3d yaitu *preproduction*, *production*, dan *postproduction* (Beane, 2012).



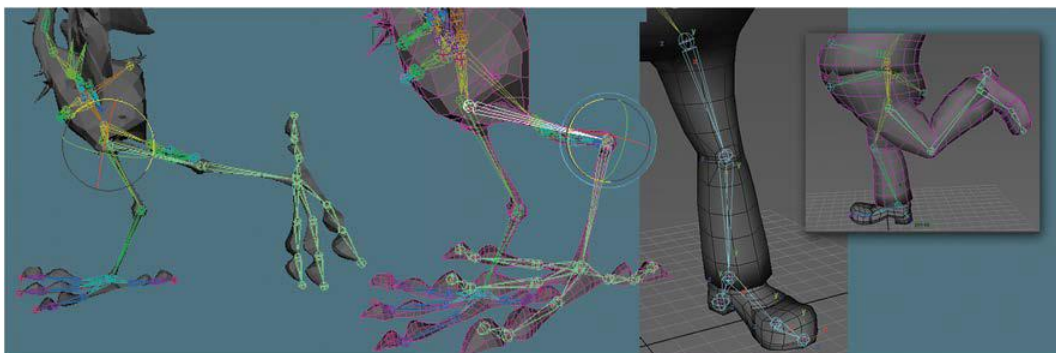
Gambar 2. 2. Rangkaian Alur Produksi Animasi 3D

(3d Animation Essentials, Beane, 2012, hlm 23)

Preproduction adalah tahap perencanaan dari pembuatan animasi 3d. Tahap perencanaan ini menjadi tolak ukur lancar atau tidaknya produksi animasi serta selesai atau tidaknya proyek tersebut. Tahap *production* adalah tahap memproses ide, riset, serta desain dari tahap *preproduction* menjadi film animasi 3d. Proses dalam tahap ini berupa *modelling*, *texturing*, *rigging*, *layouting*, *animating*, *VFX*, *lighting* dan *rendering*. Hasil dari tahap *production* akan diteruskan dalam tahap *postproduction*. *Postproduction* adalah tahapan dimana hasil render dari film animasi digabungkan atau di *compositing* dan dikoreksi sebelum menjadi hasil final (Beane, 2012).

2.2. Rigging

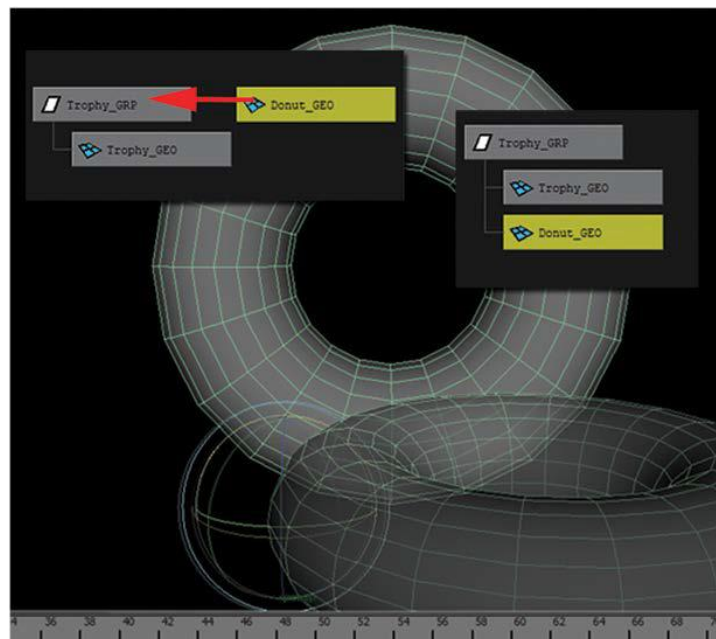
Rigging merupakan proses perancangan *control system* (Beane, 2012, hlm 177) dimana *control system* ini meliputi pemasangan *joints* dan pembuatan *controller* untuk mengontrol *joints* (Watkins, 2012, hlm 320). Menurut Assaf (2016) *Rigging* merupakan tahapan terteknis dalam *pipeline* produksi animasi 3D (hlm. 43). Dengan adanya *control system* ini membuat sebuah geometri --- baik itu karakter atau objek lainnya --- dapat bergerak dan juga memudahkan animator untuk menganimasikan geometri tersebut. Pada dasarnya, proses *rigging* merupakan proses pemberian kerangka atau armatur seperti tulang pada tubuh manusia. Tubuh manusia bergerak karena adanya pergerakan otot yang menggerakkan kerangka atau tulang sehingga terjadinya pergerakan bagian tubuh manusia. Pada animasi 3D, geometri bergerak karena adanya pergerakan *joint* yang membuat terjadinya deformasi pada geometri.



Gambar 2. 3. Rigging Dalam Produksi Animasi 3D

(Rig It Right, O'Hailey, 2019, hlm 118)

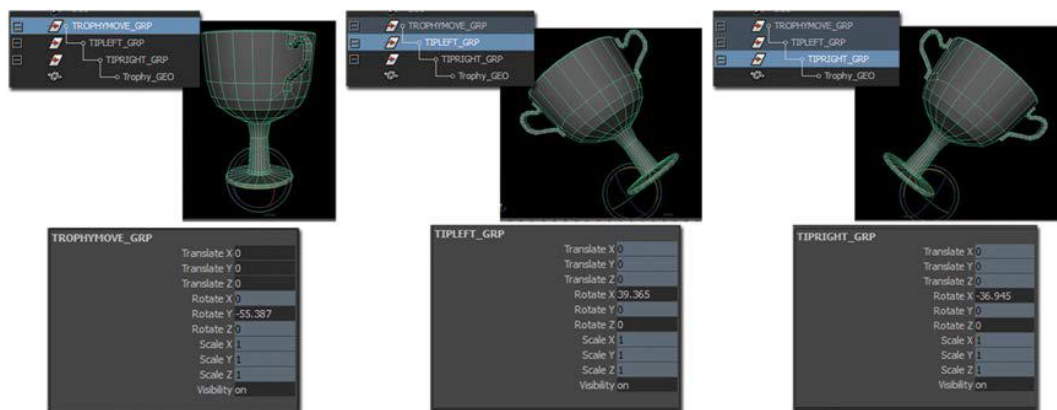
Dalam melakukan perancangan *rigging*, O’Hailey (2019) memberikan beberapa aturan dalam membuat sebuah *rig*. Pertama, jangan menganimasikan geometri. Dalam hal ini, geometri merupakan sebuah komponen yang memberikan informasi mengenai bentuk serta posisi dari objek yang akan dianimasikan. Biasanya sebuah objek akan di-*freeze transform* dalam *delete history* sehingga atribut yang ada di dalam geometri akan menjadi 0 sehingga semua deformasi atau perubahan yang terjadi pada geometri akan di-*set* menjadi *default* bagi geometri. Dalam hal ini, prinsip jangan menganimasikan geometri dapat berguna apabila objek yang akan di-*rig* berubah, *rigger* tetap mampu meng-*update rig*-nya tanpa harus membuat *rig* tersebut dari awal lagi (hlm. 8).



Gambar 2. 4. Mengubah Objek Geometri Pada *Rig* Yang Sama

(Rig It Right, O’Hailey, 2019, hlm 9)

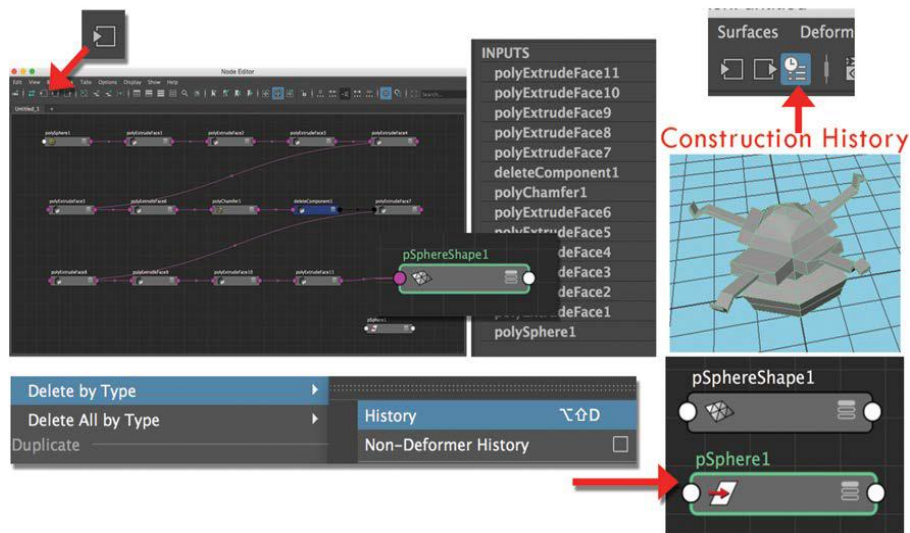
Kedua, buat *controller* yang masuk akan bagi animator. Dalam merancang sebuah *rig*, *rigger* tidak hanya merancang *control system* yang akan menggerakkan geometri dari sebuah karakter tetapi *rigger* juga mendesain bagaimana sebuah *controller* dapat dibaca dan dipakai secara mudah oleh seorang animator. Maka dari itu, beberapa hal yang dapat diperhatikan dalam membuat sebuah *controller* adalah mengunci atribut yang tidak terpakai dalam sebuah *controller*. Hal ini akan memberikan sebuah batasan bagi seorang animator. Disisi lain, batasan ini akan memudahkan seorang animator untuk mengetahui bagaimana cara mengoperasikan *controller* pada sebuah karakter. (O’Hailey, 2019, hlm. 36).



Gambar 2. 5. Lock Attribute

(Rig It Right, O’Hailey, 2019, hlm 11)

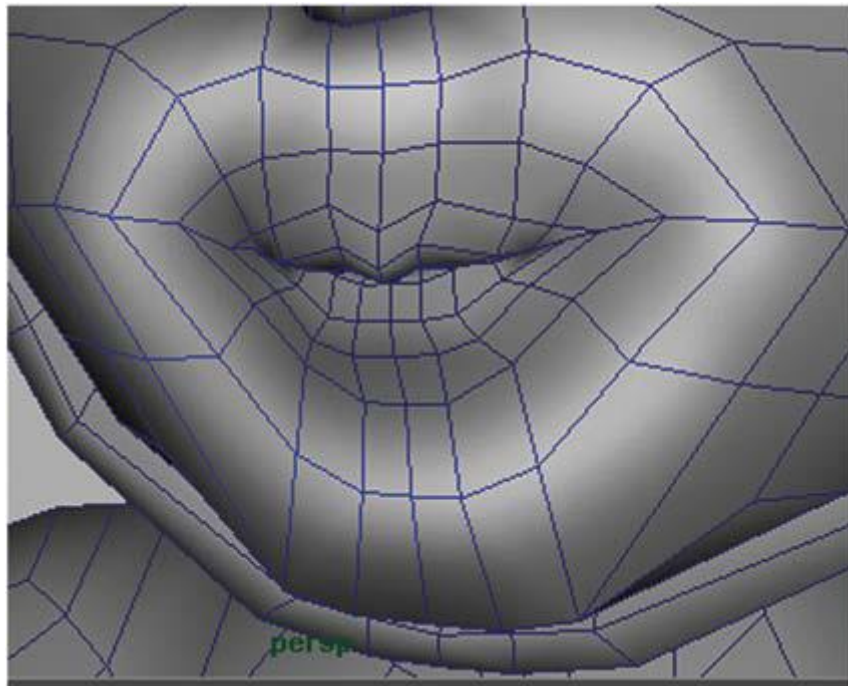
Ketiga ialah *happy math* dan *happy history*. Seperti yang sudah disinggung sebelumnya mengenai *freeze transform* dan *delete history*, kedua hal tersebut mempermudah dari segi operasional dan teknis dalam sebuah animasi. Dengan *freeze transform*, atribut sebuah objek akan bernilai 0 sehingga membuat posisi serta bentuknya yang sekarang menjadi *default*. Sementara itu *delete history* akan menghapus riwayat dari proses apa saja yang sudah dilakukan atau terjadi pada sebuah objek. Penghapusan riwayat ini bertujuan agar *software Maya* dapat lebih cepat untuk membaca dan memproses objek tersebut. Apabila *delete history* belum dilakukan maka *Maya* akan membaca riwayat perubahan atau translasi dari objek satu per satu dan akan membuat secara proses lebih lama (O’Hailey, 2019, hlm. xxiv)



Gambar 2. 6. *Delete History*

(Rig It Right, O’Hailey, 2019, hlm xxii)

Terakhir ialah *edgeloop*. *Edgeloop* merupakan garis atau *edges* yang saling berhubungan dan membentuk sebuah lingkaran. Dalam hal *rigging*, *edgeloop* menjadi penting karena pada dasarnya *rigging* merupakan konsep dimana *joint* akan mempengaruhi sebuah *vertex* pada geometri. Maka dari itu, penerapan *edgeloop* yang baik adalah sesuai dengan kontur dari permukaan objek. Contohnya pada bagian mulut pada gambar () dimana *edgeloop* mengitari bagi mulut yang nantinya akan menghasilkan deformasi yang natural seperti layaknya otot berkontraksi.

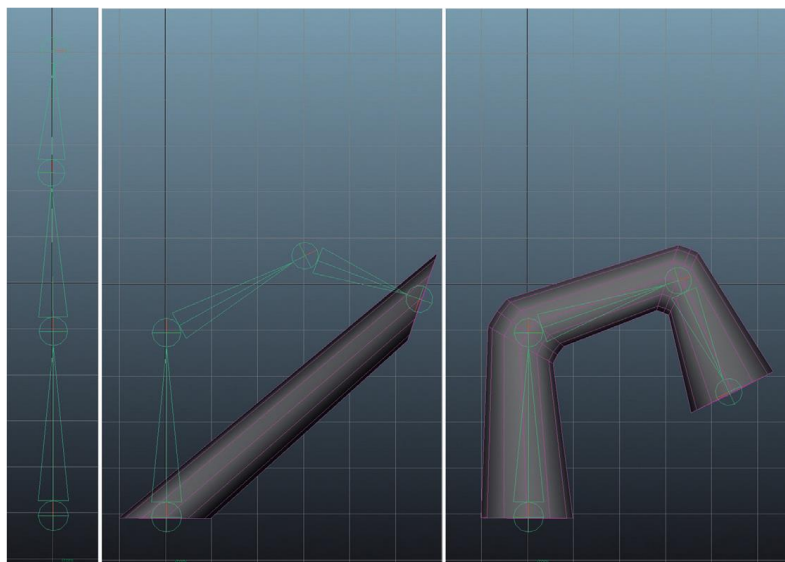


Gambar 2. 7. *Edgeloop*

(Rig It Right, O'Hailey, 2019, hlm 32)

2.2.1. Joints

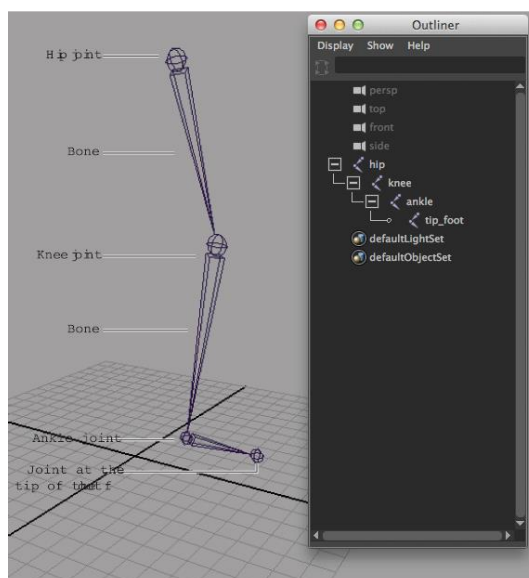
Joints merupakan sebuah armatur atau kerangka seperti tulang pada manusia. Dalam animasi 3D, ketika *joint* mengalami pergerakan maka geometri yang terpengaruh oleh *joint* juga akan mengalami perubahan (Derakhshani, 2015, hlm 352). Pada prinsipnya, posisi *joints* merupakan posisi persendian pada objek dimana geometri akan mengalami deformasi ketika terjadi pergerakan. Deformasi bisa terjadi karena *joints* berinteraksi dengan *vertices* pada geometri (Watkins, 2012, hlm 320). Pada gambar(2.4) memperlihatkan bagaimana *joint* tidak akan berfungsi apabila jumlah *vertices* pada geometri tidak cukup untuk menciptakan sebuah deformasi.



Gambar 2.8. Hubungan *Joints* dengan Geometri

(Getting Started in 3D with Maya. Create A Project From Start to Finish - Model, Texture, Rig, Animate, and Render in Maya, Watkins, 2012, hlm.322)

Dalam merancang sebuah *rig*, tingkatan atau hierarki dari *joint* merupakan suatu hal yang penting. King (2015) mengatakan hierarki memudahkan animator untuk mengerjakan beberapa objek sekaligus karena adanya hierarki yang mempengaruhi anggota dibawahnya (hlm. 94). Hierarki *joint* dalam perancangan *rig* ini memudahkan animator untuk menentukan relasi *parent-child* pada sebuah *joint*. *Parent* merupakan *joint* yang akan mempengaruhi *joint* lainnya ketika digerakan sementara *child* merupakan *joint* yang akan ikut bergerak ketika *joint* yang merupakan *parent* bergerak. Akan tetapi ketika *child* bergerak sendiri, *parent* tidak akan ikut terpengaruh karena posisi *parent* berada di atasnya. Dalam sebuah hierarki *joint*, terdapat yang disebut *root joint*. *Root joint* merupakan sebuah *parent* yang akan mempengaruhi semua *joint* yang berada dibawah hierarkinya (Watkins, 2012, hlm 323).

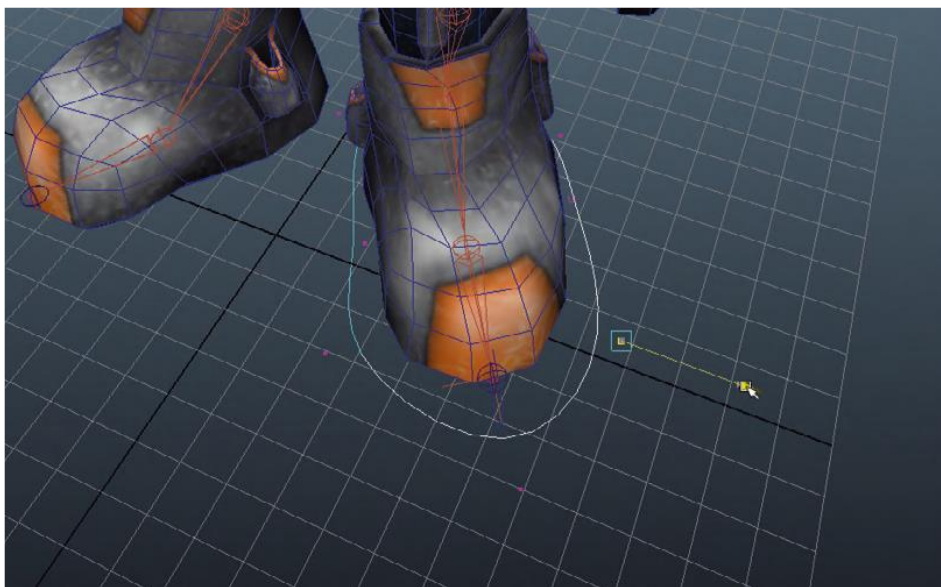


Gambar 2.9. Hierarki *Joints* Pada *Outliner*

(Introducing Autodesk Maya 2016, Derakhshani, 2015, hlm.353)

2.2.2. Controller

Controller merupakan sebuah *user interface* dari sistem *rigging* (O’Hailey, 2019, hlm 36). *Controller* memudahkan bagian yang bisa untuk dianimasikan ataupun memudahkan interaksi animator dengan *rig*. Dalam prakteknya *controller* terhubung dengan suatu atribut sehingga memungkinkan bagi sebuah *controller* untuk menggerakkan bagian pada suatu karakter. Menurut O’Hailey salah satu factor penting dalam merancang sebuah *rig* adalah *happy math* yang artinya sebuah *controller* memiliki nilai 0 pada atributnya (kecuali *scale*). Penerapan *happy math* ini membuat sebuah karakter dapat kembali ke posisi awal ketika atribut pada *controller* -nya bernilai 0.

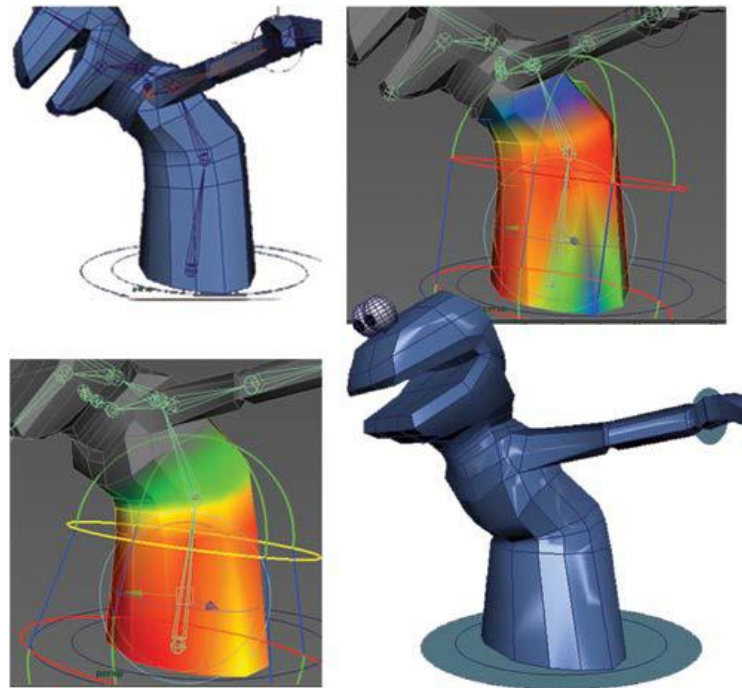


Gambar 2.10. Pemasangan Controller

(Getting Started in 3D with Maya. Create A Project from Start to Finish - Model, Texture, Rig, Animate, and Render in Maya, Watkins, 2012, hlm.344)

2.2.3. Skinning

Skinning merupakan proses penyatuan geometri dengan *joints*. Hal ini membuat *joint* memiliki pengaruh terhadap geometri ketika *joint* digerakan. Proses skinning juga menentukan batasan-batasan *joint* dalam mempengaruhi *vertices* pada sebuah geometri. Dalam proses *skinning* terdapat 2 bagian penting didalamnya yaitu *Bind Skin* dan *Paint Skin Weight*. *Bind Skin* merupakan tahapan menempelkan geometri kepada *joint*. *Paint Skin Weight* adalah tahapan penentuan sampai sejauh mana *joint* mempengaruhi suatu geometri.



Gambar 2. 11. *Skinning*

(Rig It Right, O'Hailey, 2019, hlm 83)

2.3. Relasi

Dalam perancangan sebuah *rig*, relasi merupakan hal yang penting (Derakhshani, 2015, hlm 387). Relasi menghubungkan sebuah objek dengan objek lainnya. Pada prakteknya, pemakaian konsep relasi ini dapat terlihat dalam pembuatan *controller*. *Controller* memiliki hubungan dengan sebuah atribut pada suatu *joint* sehingga ketika *controller* digerakan *joint* yang terhubung dengannya ikut bergerak. Beberapa konsep dari relasi ini terdapat pada *constraint* dan *set-driver key*.

2.3.1. Constraints

Constraints merupakan relasi antara *source* objek dengan *target* objek lainnya. Relasi tersebut meliputi beberapa atribut sederhana seperti *translate*, *rotate* dan *scale* (Derakhshani, 2015, hlm 387). *Source* objek adalah objek yang mengikuti atribut dari *target* objek. Sehingga ketika sebuah *target* objek bergerak atau berubah atributnya, *source* objek akan mengikutinya. Berikut merupakan beberapa jenis dari *constraint*:

1. *Point Constraint*

Point constraint adalah relasi yang menghubungkan atribut *translate* antara *source* objek dengan *target* sehingga ketika *target* bergerak maka *source* objek akan mengikuti pergerakan *target*.

2. *Orient Constraint*

Orient constraint menghubungkan atribut *rotation* pada 2 objek atau lebih. Hal ini membuat beberapa objek dapat berputar searah secara bersamaan.

3. *Scale Constraint*

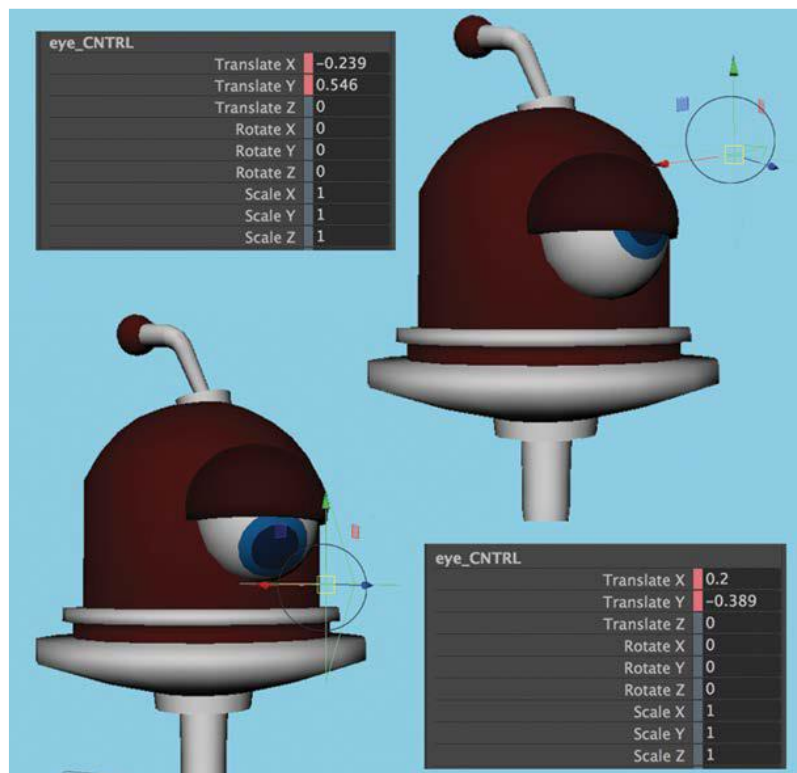
Constraint yang menghubungkan atribut *scale* pada 2 objek atau lebih.

4. Aim Constraint

Aim constraint adalah *constraint* yang menghubungkan atribut rotasi dari *source* objek dengan atribut *translate* dari *target* objek. Hasil dari *constraint* ini membuat *source* objek berputar atau berotasi sesuai dengan posisi dari *target* objek.

5. Parent Constraint

Parent constraint adalah *constraint* yang menghubungkan atribut *rotate* serta *translate* pada 2 objek atau lebih.



Gambar 2. 12. Aim Constraint

(Rig It Right, O'Hailey, 2019, hlm 53)

2.3.2. Set-Driven Keys

Set-driven keys (SDK) merupakan sebuah relasi antara objek seperti pada *constraint*. Perbedaan dari *constraint* dengan SDK adalah SDK dapat menghubungkan atribut tertentu atau *custom* atribut tidak seperti *constraint* yang menghubungkan atribut sederhana seperti *translate*, *rotate* dan *scale*. Maka dari itu *set-driven keys* memungkinkan seorang *rigger* untuk menciptakan sebuah relasi yang menggerakkan suatu bagian atau fitur khusus pada sebuah karakter. Dalam perancangannya, perlu dibuatnya sebuah atribut khusus yang akan menggerakkan bagian pada karakter.



Gambar 2. 13. Set-Driven Key

(Rig It Right, O'Hailey, 2019, hlm 165)

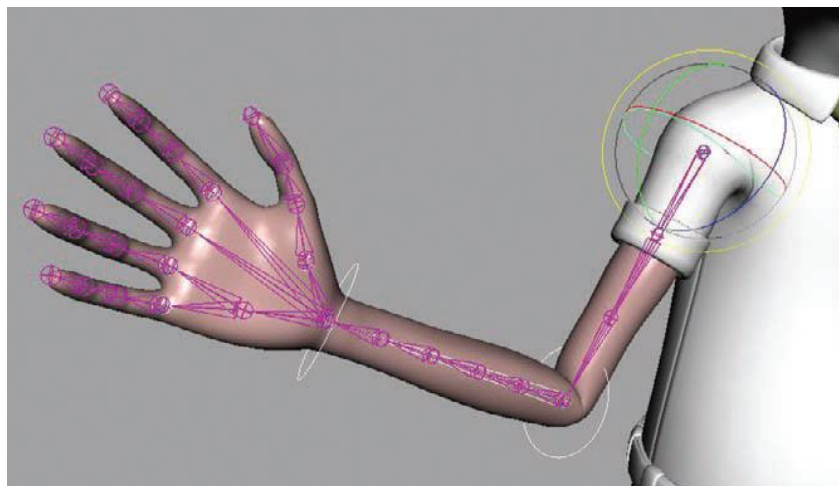
2.4. Kinematik

Menurut Cabrera, kinematik merupakan ilmu yang mempelajari pergerakan. Spesifiknya pergerakan yang terjadi tanpa adanya gaya eksternal yang mempengaruhi pergerakan tersebut. Jadi kinematik mempelajari pergerakan alami suatu objek. Selain pergerakan, kinematik juga

mempelajari bagaimana sesuatu bergerak (hlm 243). Dalam animasi 3D, kinematik merupakan metode untuk menggerakkan *joint* atau *skeletal system* pada suatu objek (Derakhshani, 2015, hlm 352). Watkins (2012) mengatakan bahwa kinematik adalah cara bagaimana animator menggerakkan sebuah objek (hlm 329). Terdapat 2 metode dalam penggerakkan *skeletal system* dalam animasi 3D yaitu *Inverse Kinematic* (IK) dan *Forward Kinematic* (FK).

2.4.1. Forward Kinematic (FK)

Forward Kinematic (FK) merupakan pergerakan yang berhubungan dengan hierarki dari *joints* pada suatu objek (Cabrera, 2008, hlm 219). Dalam prosesnya, ketika *joint* yang hierarkinya berada di atas berotasi maka posisi dari *joint-joint* yang hierarkinya berada di bawah *joint* tersebut akan berubah (Derakhshani, 2015, hlm 352). Sistem FK membuat animator harus menggerakkan banyak *joint* untuk menciptakan sebuah pose. Salah satu keunggulan dari sistem FK ini ialah pergerakan tubuh yang tercipta mengikuti lengkungan dari pergerakan tubuh manusia (Watkins, 2012, hlm 329).

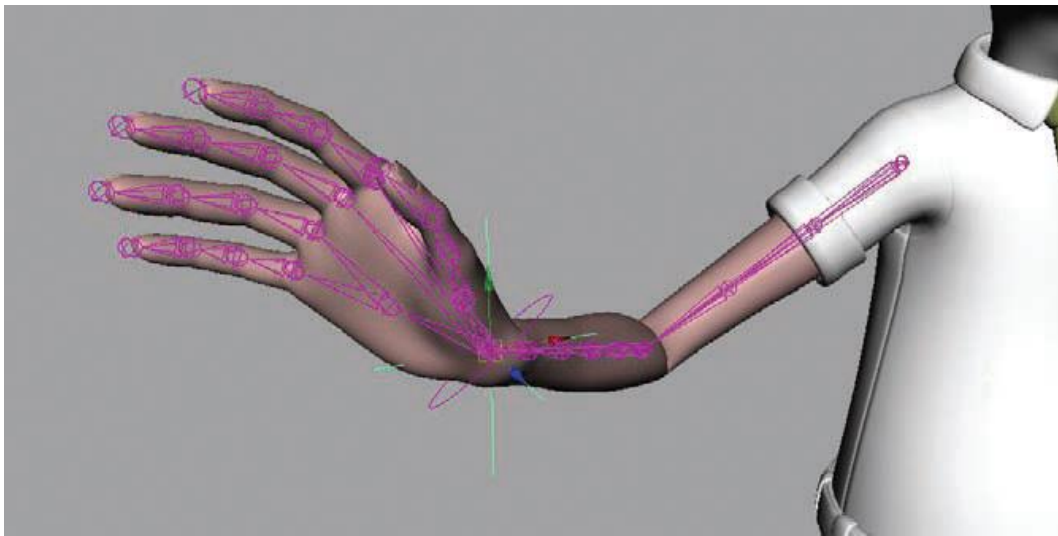


Gambar 2. 14. Forward Kinematic

(3d Animation Essentials, Beane, 2012, hlm 185)

2.4.2. Inverse Kinematic (IK)

Inverse Kinematic (IK) adalah sistem yang membalikan hierarki pada *joint*. Dalam sistem IK, terdapat *IK handle* dan *root segment*. *IK handle* biasanya dipasangkan pada ujung dari *joint system* seperti pada pergelangan tangan atau kaki. *Root segment* merupakan batas sampai dimana sistem IK ini berpengaruh misalnya pada bahu atau pangkal paha (Derakhshani, 2015, hlm 352). Ketika *IK handle* digerakan, maka *joint* yang hierarkinya berada di atas *handle* akan ikut bergerak dan pengaruhnya akan berhenti pada *joint* yang menjadi *root segment* dari sistem IK (Cabrera, 2008, hlm 220). Sistem IK biasanya digunakan pada bagian tubuh yang langsung bersentuhan dengan objek lain seperti kaki dimana selalu bersentuhan dengan tanah.



Gambar 2. 15. Inverse Kinematic

(3d Animation Essentials, Beane, 2012, hlm 185)

2.5. Transformasi

Transformasi merupakan suatu proses perubahan. Menurut Singh, dkk (2009) transformasi adalah proses perubahan keadaan yang menciptakan atau menambah fungsi dari suatu objek. Transformasi membuat keadaan objek berubah dari keadaan A menjadi B dan merubah cara kerja atau fungsi dari objek itu sendiri. Pada Gambar 2.16 memperlihatkan sebuah mainan mobil yang dapat bertransformasi menjadi robot *humanoid*. Pada kedua keadaan ini cara mainan ini dimainkan menjadi berbeda. Pada bentuk mobil mainan akan dimainkan dengan didorong dan akan bergerak layaknya mobil pada umumnya sementara pada bentuk robot *humanoid* mainan akan dimainkan dengan lebih fleksibel seperti memainkan figurin manusia.



Gambar 2. 16. Contoh Transformasi

(Innovations in Design Through Transformation: A Fundamental Study of Transformation Principles, Singh, 2009, hlm 2)

Pada *software Autodesk Maya*, transformasi mempunyai arti yaitu mengubah posisi dan mengubah bentuk. Terdapat 2 jenis transformasi dalam *Maya* yaitu *transformation objects* dan *transformation components*. *Transformation objects* adalah transformasi yang mengubah posisi sebuah objek terhadap lingkungannya. Pada umumnya *transformation objects* mengubah atribut *translate*, *rotate* dan *scale* sebuah objek. Disisi lain *transformation components* merupakan transformasi yang mengubah bentuk dari sebuah objek. Disini bagian-bagian sebuah objek seperti *vertice*, *edge*, dan *face* akan diubah sedemikian rupa sehingga menciptakan bentuk baru dari sebuah objek.