

## 2. STUDI LITERATUR

### 2.1 Animasi

Menurut Wells (2016), animasi pada dasarnya dapat diartikan sebagai membuat ilusi pergerakan benda mati. Namun seiring dengan perkembangan zaman, definisi animasi pun mengalami perubahan menjadi memberikan kehidupan kepada benda mati. Dalam industri animasi 3D, diperlukan sebuah sistem yang berisi sekumpulan orang yang mengerjakan tugasnya masing-masing untuk dapat membuat animasi 3D secara keseluruhan. Dalam *3D animation pipeline*, tiap pekerja akan meneruskan pekerjaan mereka kepada pekerja lain. Sehingga *3D artist* perlu mengetahui dan memahami beberapa tahap setelah pekerjaannya (Beane, 2012). Menurut Beane, tahapan produksi terdiri dari *layout, research and development, modeling, texturing, rigging, animation, 3D VFX*, dan *lighting/rendering*

### 2.2 Sistem Gerak Manusia

#### a. Tulang

Menurut Viandari (2019), kerangka manusia dapat dibagi menjadi 2, yaitu rangka aksial dan rangka apendikuler. Rangka aksial merupakan rangka yang menyusun sumbu tubuh, yang terdiri dari tulang kepala, tulang dada, tulang rusuk, dan tulang belakang. Sedangkan rangka apendikuler merupakan rangka alat gerak yang melengkapi rangka aksial.

#### b. Sendi

Sendi adalah hubungan antar tulang, baik yang dapat digerakkan maupun tidak dapat digerakkan (Tate, 2012). Berdasarkan pergerakannya, sendi dapat dibagi menjadi 3, sendi sinartrosis, sendi amfiartrosis, dan sendi diartrosis. Sendi sinartrosis merupakan sendi yang tidak dapat digerakan. Sendi amfiartrosis merupakan sendi yang memiliki gerakan terbatas. Sendi diartrosis merupakan sendi yang dapat bergerak secara leluasa.

### 2.3 Sistem *Rigging*

Menurut Beane (2012), *rigging* adalah proses membuat objek model agar dapat digerakan. *Rigger* bertanggung jawab atas pembuatan sistem *rigging* serta *controller* yang dapat memudahkan animator dalam proses animasi. Pembuatan sistem *rigging* didasari dengan pemikiran setiap benda yang bergerak memiliki sistem penggerakannya. O'Neill (2015) menambahkan, proses membuat *rig* karakter tidaklah mudah, terutama dalam *motion system*. Setelah pembuatan sistem *rigging* selesai pun, *rig* masih perlu diawasi hingga akhir produksi.

Dalam pembuatan sistem *rigging*, O'Hailey (2018) membuat 10 peraturan dasar dalam pembuatan sistem *rigging*. Beberapa di antaranya meliputi:

- Membuat *controller* yang mudah dipahami oleh animator.
- *Controls* dan *joints* harus bernilai nol.
- Menghapus *history* yang tidak diperlukan.
- Selalu memperhatikan peletakan *joints*, sudut yang diinginkan, dan orientasinya.
- Tidak boleh menge-freeze *blend shape*.

Dalam pembuatan *rig*, *rigger* juga harus memahami tentang 3 sistem yang terhubung di dalamnya (McLaughlin, Cutler, & Coleman, 2011):

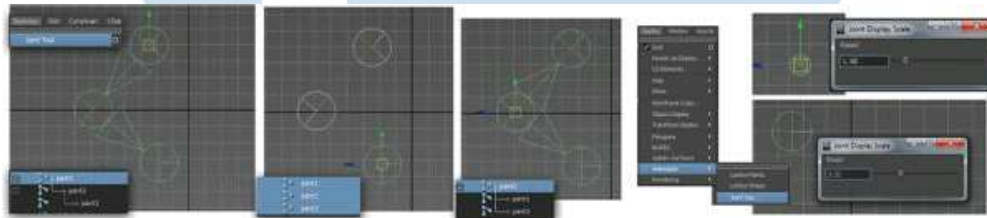
- *Motion system* -kombinasi dari *joints* yang membentuk sistem kerangka karakter.
- *Control system* -koneksi dari *controls* ke *joints* yang dapat menggerakkan *motion system*.
- *Deformation system* -pengaruh dari *motion system* terhadap sebuah model.

Mereka juga menegaskan, untuk menghasilkan *motion system* yang baik diperlukan pemahaman anatomi dan sistem gerak. Untuk menghasilkan *control system* yang baik, dibutuhkan pengetahuan interaksi animator dengan model karakter, serta kemampuan mendesain solusi dari masalah pergerakan yang kompleks. Sedangkan untuk membuat *deformation system*, diperlukan pemahaman anatomi dan *sculptor's touch* dalam mempertahankan bentuk dan detail model

selama bergerak (2011). Menurut Briggs (2021), sebelum memulai tahap *rigging*, lebih baik untuk menganalisis bagian mana dari objek yang bergerak (*rig*).

#### a. *Joints*

Sesuai dengan peraturan dasar O'Hailey (2018), *joints* hanya boleh di-*rotate* dan tidak boleh di-*translate*. Dalam kasus khusus, ujung paling dasar *joints* atau *root* dapat di-*rotate* dan *translate* tanpa mengganggu hasil akhir. Dalam gambar 2.1 dapat dilihat 2 macam *joints* yang memiliki *root* yang berbeda. Ketika *joints* di-*rotate* juga perlu diperhatikan bahwa anak-anak *joints* juga ikut ter-*rotate*. Beliau juga menegaskan, *joints* harus diletakan untuk dapat menyatu dengan *edge loops* model. Jika posisi *joints* kurang tepat, maka pergerakan *rig* tidak akan sesuai dengan yang diinginkan.

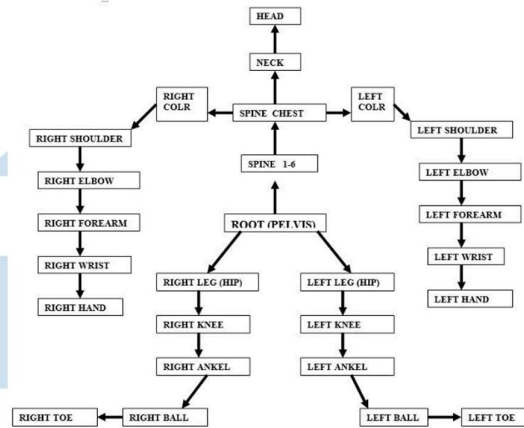


Gambar 2.1 Membuat *joints* dan *root*

(O'Hailey, 2018)

Bhati dan rekannya (2015) menegaskan bahwa proses *rigging* pertama kali dilakukan dengan membuat struktur hirarki dari kerangka. Setelah dibuat, *joints* harus diberi nama agar mudah untuk dibedakan. Sutiono (2012) mengatakan bahwa dalam penyusunan hirarki kerangka terpusat pada susunan kerangka di bawah perut karakter. Titik pusat tersebut berfungsi untuk menggabungkan *joints* kaki dan *joints* tulang punggung.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 2.2 Struktur hirarki kerangka dari karakter bipedal  
(Bhati dkk, 2015)

#### b. *Parent and Child*

Semua *rig* memiliki sistem hirarkinya sendiri, yaitu sistem *parent and child* (Beane, 2012). Ketika objek *parent* bergerak, objek *child* akan ikut bergerak. Sedangkan ketika objek *child* bergerak, objek *parent* tidak ikut bergerak. Sebuah objek *parent* bisa memiliki beberapa objek *children*, tetapi tidak sebaliknya.

#### c. *Constraints*

Menurut Beane (2012), *constraints* adalah sebuah penghubung suatu objek untuk mengendalikan objek lainnya. *Constraints* pada dasarnya terlihat sama dengan *parents*, tetapi *constraints* dapat memberi kontrol yang lebih banyak dan koneksi yang lebih baik. Beliau menyebutkan bahwa jenis-jenis *constraints* yang sering digunakan adalah *point*, *aim*, dan *orient*.

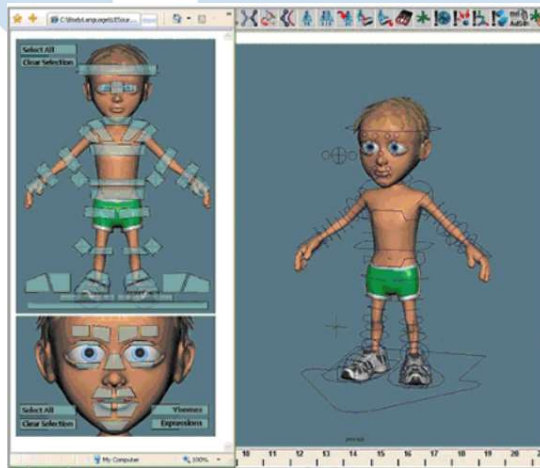
O'Hailey (2018) menambahkan *pole vector* sebagai salah satu jenis *constraints* yang sering dipakai dalam *software* Autodesk Maya terutama dalam kasus *inverse kinematics*. *Pole vector* menjadi acuan arah yang dituju bagi *inverse kinematics*.

d. *Forward Kinematics* (FK) dan *Inverse Kinematics* (IK)

Beane (2012) membagi pergerakan joints menjadi *forward kinematics* (FK) dan *inverse kinematics* (IK). FK bergerak mengikuti gerak struktur dan IK bergerak mempengaruhi struktur. Contoh penggunaannya adalah pada kasus gerak kaki. Saat kaki menapak, maka IK digunakan dan saat bergerak maka FK (Schleifer, 2006).

e. *Controller*

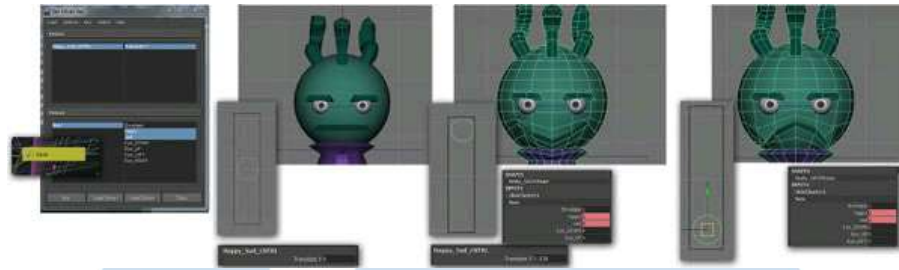
Berdasarkan jurnal Satriawan dan Apriyani (2012), fungsi utama *controller* adalah untuk menggerakkan *joints*. Biasanya, *controller* merupakan sebuah *curves* yang di-*constraints* kepada *joints*. Menurut Allen dan Murdock (2008), prinsip utama pembuatan *controller* adalah mempermudah kinerja animator. Oleh sebab itu, *Rigger* disarankan untuk membuat semua *controller* yang dibutuhkan oleh animator dan meletakkannya dekat dengan apa yang dikendalikan.



Gambar 2.3 *Controller* karakter humanoid

(Allen dan Murdock, 2008)

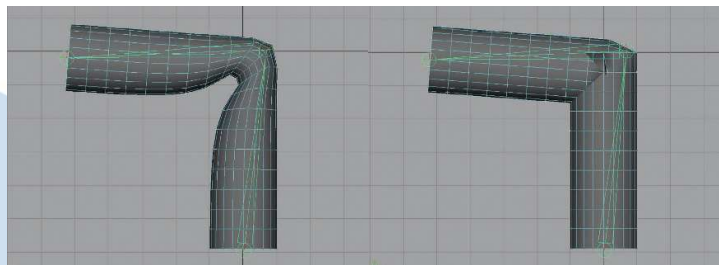
Dalam buku O'Hailey (2018), *set driven key* sering diaplikasikan kepada *controller* untuk menghubungkan sebuah atribut dari suatu objek ke atribut dari objek lain. Atribut yang diberi *set driven key*, dapat bersifat serupa maupun benar-benar berlawanan (Allen dan Murdock, 2008).



Gambar 2.4 Menghubungkan *controller* ekspresi dengan *blendshapes*  
(O'Hailey, 2018)

#### f. *Skinning*

Menurut O'Hailey (2018), *skinning* adalah proses untuk membuat *mesh* bergerak sesuai dengan kerangkanya. *Rigger* harus mengatur *vertex* mana saja yang akan bergerak dengan suatu *joints* dan seberapa banyak *weights* yang dibutuhkan untuk mendapatkan pergerakan yang tepat. Dalam kasus model yang simetris, *skin weight* dapat di-*mirror* agar proses pengerjaan menjadi lebih efisien. Berbeda dengan *skinning* biasa atau *smooth skinning*, *rigid skinning* menaruh antara 100% *value* atau 0% *value of weights* pada sebuah *vertex* (Beane, 2012).



Gambar 2.5 *Smooth skinning* (kiri) dan *Rigid skinning* (kanan)  
(Beane, 2012)

*Blendshapes* adalah sebuah deformasi yang dapat menghubungkan 2 objek bersama dan dapat membuat objek asli berubah bentuk menjadi kedua objek tersebut (Bane, 2012). Dalam kasus tertentu, seperti pembuatan *facial animation rig*, penggunaan *blendshapes* lebih disarankan daripada penggunaan *joints*, karena dapat mempercepat kinerja. Meskipun begitu, *blendshapes* harus dilakukan dengan teliti agar tidak ada *blendshapes* lain yang memengaruhi *vertex* yang sama

(O’Hailey, 2018). Dalam prosesnya, pembuatan *blendshapes* harus dilakukan sebelum pembuatan *skinning*, sehingga sistem *rigging* dapat menggerakkan model yang memiliki *blendshapes*. Biasanya *blendshapes* dibuat oleh seorang *modeler* agar deformasi yang terjadi sesuai dengan yang diharapkan. Sedangkan jika *blendshapes* dilakukan setelah *skinning*, maka opsi ‘*At Front of Chain*’ harus dipilih, agar urutan deformasi menjadi benar.



UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA