

BAB II

TINJAUAN PUSTAKA

2.1. Animasi 3D

3D adalah salah satu perkembangan dari industri animasi. Bernhardt (2015) berkata bahwa Animasi 3D membahas mengenai ruang atau tempat dan teknologi lah yang menjadi penyebab kemunculan animasi 3D. Teknologi komputer di bidang grafik dan CG membuat adanya berbagai metode serta teknik dalam pembuatan animasi. Animasi 3D pun menjalani proses yaitu *pre-production*, *production*, dan *post-production* (Beane, 2012).

Pre-production merupakan tahap perencanaan dari pembuatan sebuah animasi 3D. Tahap ini menjadi penentuan lancar atau tidaknya produksi serta selesai atau tidaknya sebuah proyek. Tahap *production* merupakan tahap eksekusi ide, riset, serta desain dari tahap *pre-production* menjadi animasi 3D. Proses dalam tahap ini adalah *modelling*, *texturing*, *rigging*, *layouting*, *animating*, *VFX*, *lighting* dan *rendering*. *Post-production* merupakan tahap dimana hasil *production* digabungkan dan dikoreksi sebelum menjadi hasil final (Beane, 2012).

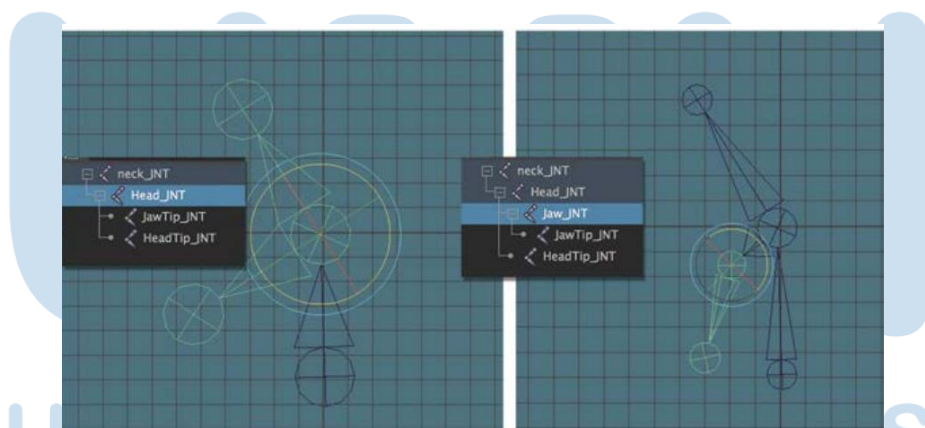
2.2. Rigging

Rigging adalah 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 paling teknis dalam pipeline produksi animasi 3D (hlm. 43). Pada dasarnya, proses *rigging* merupakan proses pemberian kerangka atau armatur

seperti tulang pada tubuh manusia. Tubuh manusia bergerak karena adanya pergerakan otot yang menggerakkan rangka atau tulang sehingga terjadinya pergerakan bagian tubuh manusia. Pada animasi 3D, geometri bergerak karena adanya pergerakan *joint* yang membuat terjadinya deformasi pada geometri.

2.2.1. Joints

Joints merupakan sebuah kerangka seperti tulang pada manusia. Dalam animasi 3D, ketika kita menggerakkan *joint* maka geometri yang terpengaruh oleh *joint* akan mengalami perubahan (Derakhshani, 2015, hlm 352). Posisi *joints* adalah posisi sendi pada objek dimana geometri akan mengalami deformasi ketika terjadi pergerakan. Dalam merancang *rig*, hierarki dari *joint* merupakan suatu hal yang penting. King (2015) membahas bahwa hierarki memudahkan *animator* dalam menggerakkan beberapa objek sekaligus karena adanya tingkatan hierarki yang mempengaruhi anggota di bawahnya (hlm. 94).

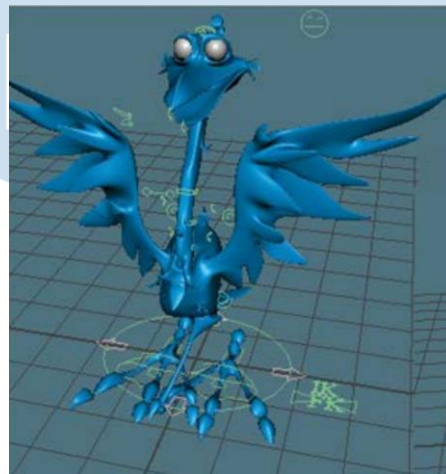


Gambar 2.1. *Joints*

(Rig It Right, O'Hailey, 2019, hlm 72)

2.2.2. Controller

Controller adalah sebuah *user interface* dari sistem *rigging* (O’Hailey, 2019, hlm 36). *Controller* berguna untuk memudahkan bagian yang bisa untuk dianimasikan. Dalam prakteknya *controller* terhubung dengan suatu atribut sehingga memungkinkan bagi sebuah *controller* untuk menggerakkan bagian pada suatu karakter. O’Hailey berkata bahwa salah satu faktor dalam merancang sebuah *rig* adalah *happy math* yang artinya sebuah *controller* yang memiliki nilai 0 pada atributnya (kecuali skala). Aplikasi *happy math* ini membuat karakter kembali ke posisi awal ketika atribut pada *controller* -nya bernilai 0.



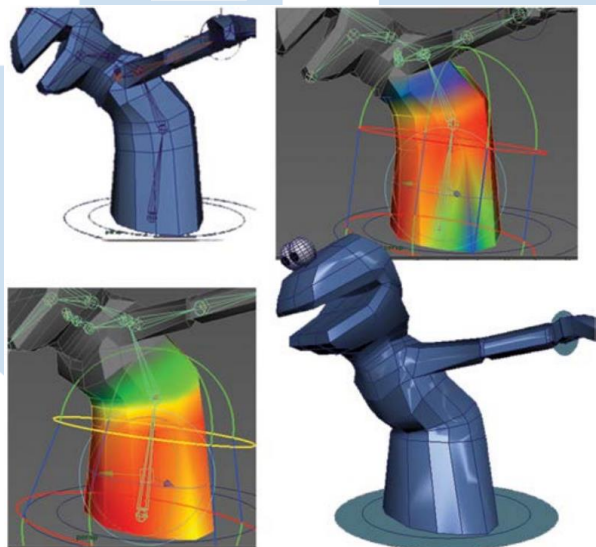
Gambar 2.2. Controller

(Rig It Right, O’Hailey, 2019, hlm 216)

2.2.3. Skinning

Skinning adalah proses penyatuan geometri dengan *joints*. proses ini membuat *joint* berpengaruh 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 2.3. Skinning
(Rig It Right, O’Hailey, 2019, hlm 83)

2.3. Relasi

Dalam perancangan sebuah *rig*, relasi adalah hal yang penting (Derakhshani, 2015, hlm 387). Relasi menghubungkan sebuah objek dengan objek lainnya. Pada prakteknya, pemakaian konsep relasi dipakai dalam pembuatan *controller*. *Controller* mempunyai hubungan dengan sebuah atribut pada suatu *joint* sehingga

joint yang terhubung dengannya bergerak ketika *controller* digerakan. 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

Scale constraint adalah fungsi *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.

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 2.4 *Set-Driven Key*

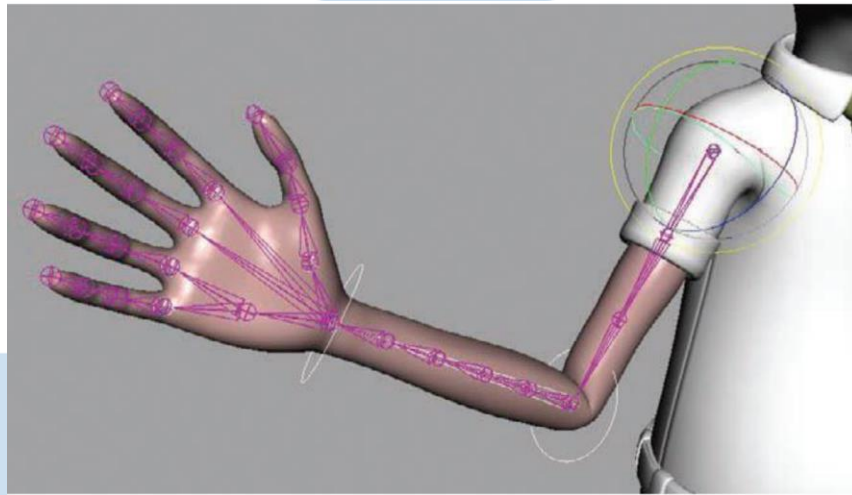
(Rig It Right, O'Hailey, 2019, hlm 165)

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 pergerakan *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 adalah pergerakan tubuh yang tercipta mengikuti lengkungan dari pergerakan tubuh manusia (Watkins, 2012, hlm 329).



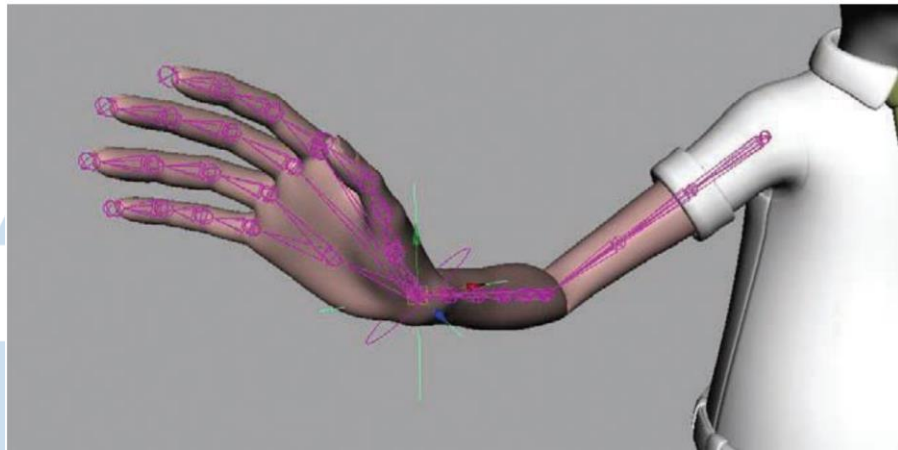
Gambar 2.5. Forward Kinematic

(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 dipasang 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 terhenti 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.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 2.6. Forward Kinematic

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

UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA