



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, 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

### TINJAUAN PUSTAKA

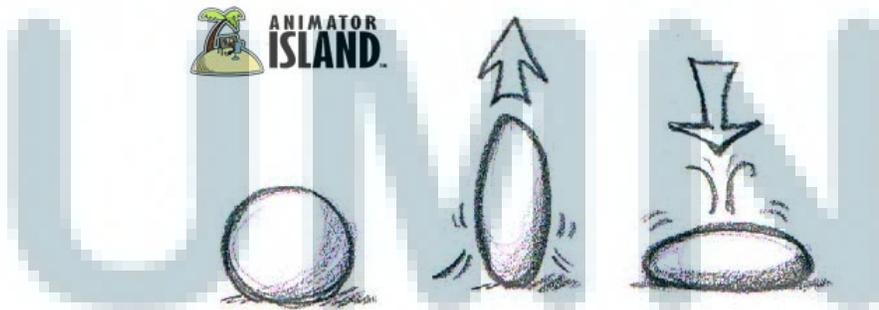
#### 2.1. Pengertian Animasi

Menurut Dobson (2009), animasi adalah gerakan yang digambar, benda mati yang “dihidupkan” melalui model dan gambar, dan film yang bukan *live-action*. Steven Withrow dalam bukunya “Secrets of Digital Animation” mengatakan bahwa animasi adalah seni untuk “menghidupkan” gambar diam atau statis. Dari kedua sumber tersebut dapat disimpulkan bahwa animasi adalah rangkaian gambar diam atau statis yang dapat menghasilkan ilusi gerakan.

Selain itu, Montgomery (2012) mengatakan dalam animasi ada 12 prinsip yang harus selalu diingat sebagai pembuat animasi.

##### 2.1.1. *Squash and Stretch*

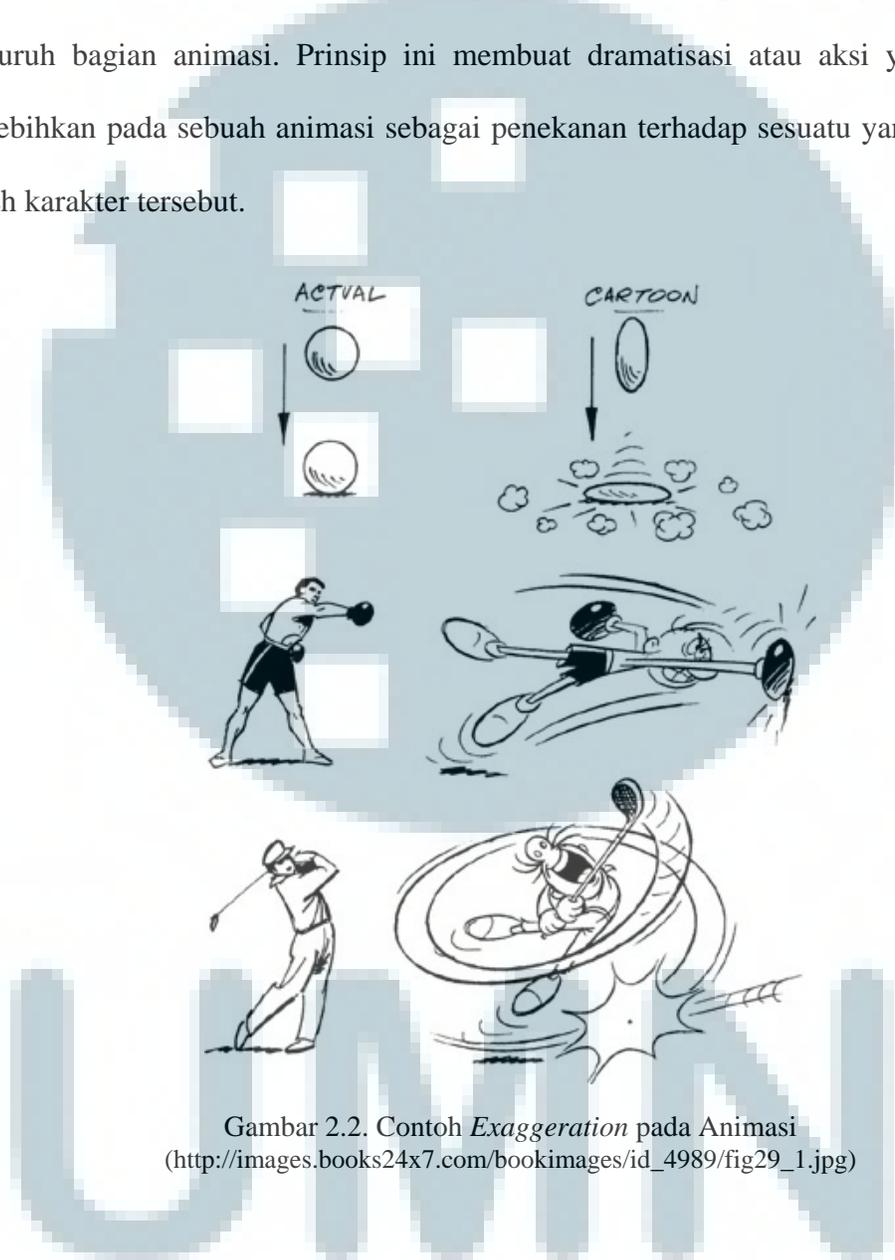
Prinsip ini diperlukan saat melakukan animasi perubahan volume atau bentuk pada objek organik.



Gambar 2.1. Perubahan Volume pada Bola  
(<http://www.animatorisland.com/wp-content/uploads/2012/06/sandsBall.png>)

### 2.1.2. *Exaggeration*

Prinsip ini adalah prinsip umum dalam penganimasian yang harus diaplikasikan pada seluruh bagian animasi. Prinsip ini membuat dramatisasi atau aksi yang sengaja dilebihkan pada sebuah animasi sebagai penekanan terhadap sesuatu yang dilakukan oleh karakter tersebut.



Gambar 2.2. Contoh *Exaggeration* pada Animasi  
([http://images.books24x7.com/bookimages/id\\_4989/fig29\\_1.jpg](http://images.books24x7.com/bookimages/id_4989/fig29_1.jpg))

### 2.1.3. *Appeal*

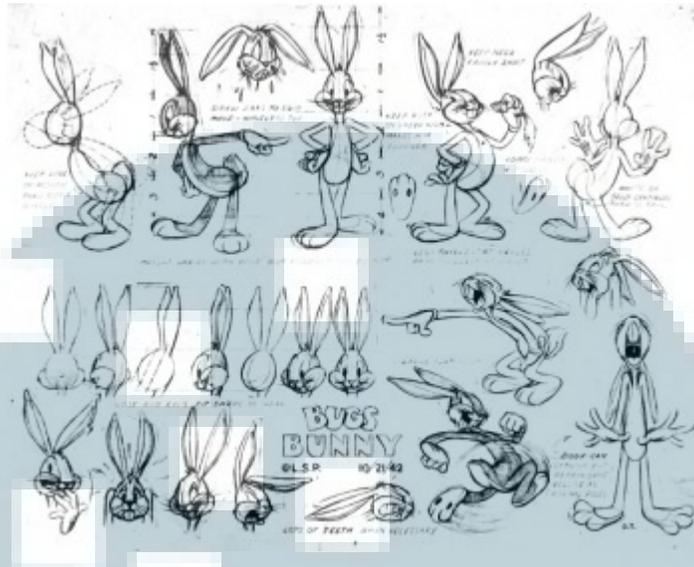
Prinsip ini ditujukan kepada daya tarik karakter dalam animasi. Setiap karakter dalam animasi harus memiliki daya tarik tersendiri, sehingga menjadi sesuatu yang dapat membedakan tiap karakter.



Gambar 2.3. Tokoh-Tokoh dalam Film Animasi  
(<http://craigbowman.com/wp-content/uploads/2010/08/Appeal.jpg>)

### 2.1.4. *Solid Drawing and Design*

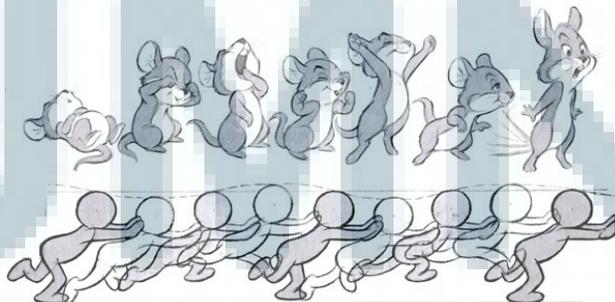
Keterampilan seni tradisional seperti menggambar, mengukir, dan mendesain sangat diperlukan dalam animasi digital. Tanpa kemampuan untuk membuat fondasi yang baik dan bentuk yang meyakinkan, animasi yang dibuat tidak akan mendapat hasil yang memuaskan.



Gambar 2.4. Sketsa Tokoh Bugs Bunny  
 ([http://www.michaelspornanimation.com/splog/wp-content/K/\\_Bugs4.jpg](http://www.michaelspornanimation.com/splog/wp-content/K/_Bugs4.jpg))

### 2.1.5. *Pose to Pose and Straight Ahead*

*Pose to Pose* dan *Straight Ahead* adalah 2 metode yang dapat digunakan dalam membuat animasi. *Pose to Pose* adalah metode animasi saat animator membuat pose utama karakter sebelum membuat pose perantara atau *in-between*. *Straight Ahead* adalah metode animasi yang melakukan penganimasian secara langsung tanpa membuat pose utama atau pose kunci.



Gambar 2.5. Contoh *Frame* Pergerakan Animasi  
 (<http://www.larabank.com/flash/classwork/show/139.jpg>)

### 2.1.6. *Secondary Action*

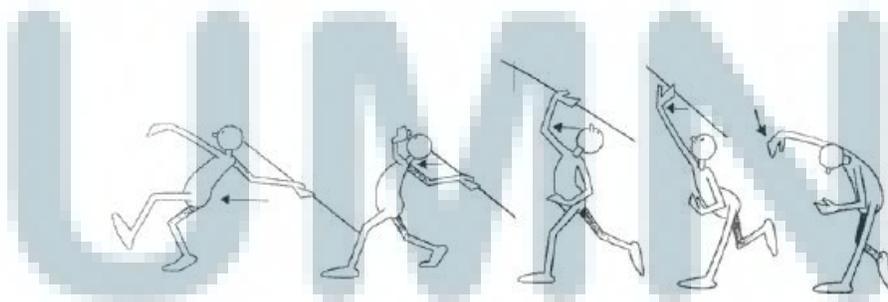
*Secondary Action* adalah gerakan tambahan yang mendukung gerakan utama pada animasi.



Gambar 2.6. Gerakan Ekor pada Penganimasian Tupai  
(<http://4.bp.blogspot.com/-ZC57kKjldyg/TsPxCr2twcI/AAAAAAAAAE4/0S8tPz6RrYk/s640/144c.jpg>)

### 2.1.7. *Follow-Through and Overlap*

Objek tidak selalu bergerak secara bersamaan, melainkan bergerak secara mengikut atau berlebih. Contohnya seperti saat mengayunkan pemukul bola baseball, badan akan mengikuti pergerakan tangan sesuai dengan arah ayunan yang dilakukan.

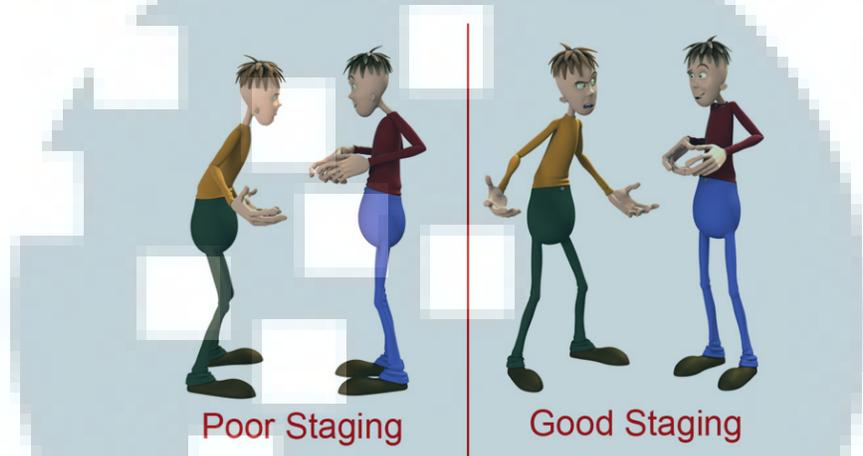


Gambar 2.7. Contoh *Follow-Through* dan *Overlap*  
(<http://www.animationbrain.com/2D/35principle.JPG>)

### 2.1.8. *Staging*

Staging adalah penempatan pandangan yang akan ditunjukkan oleh pembuat animasi.

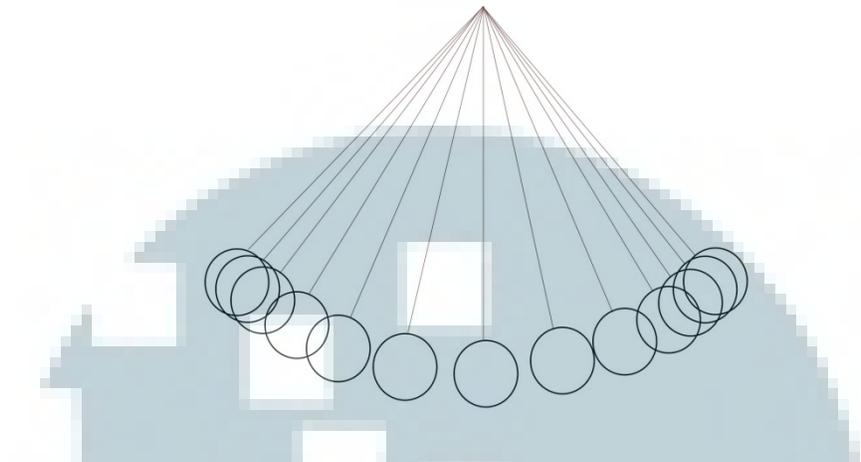
Prinsip ini sama dengan penempatan kamera atau karakter pada saat pembuatan film.



Gambar 2.8. Contoh *Staging* pada Animasi  
(<https://software.intel.com/sites/default/files/m/d/4/1/d/8/character-ani-2.jpg>)

### 2.1.9. *Ease-in and Ease-out*

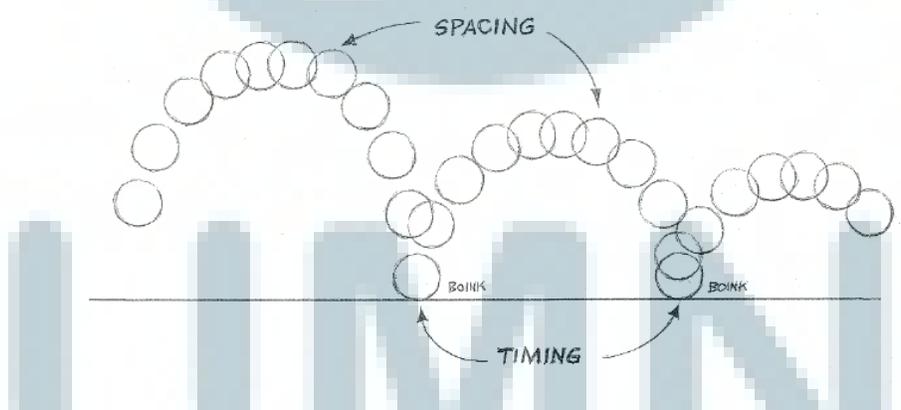
Objek dan manusia tidak bergerak pada kecepatan yang tetap. Ada kecepatan saat memulai pergerakan yang semakin lama semakin bertambah kecepatannya dan ada juga saat akan mengakhiri pergerakan yang semakin lama semakin menurun kecepatan dari pergerakan tersebut.



Gambar 2.9. Perbedaan Kecepatan pada Pendulum yang Bergerak  
(<https://amford5.files.wordpress.com/2012/03/slow-in-and-out.jpg>)

### 2.1.10. *Timing and Spacing*

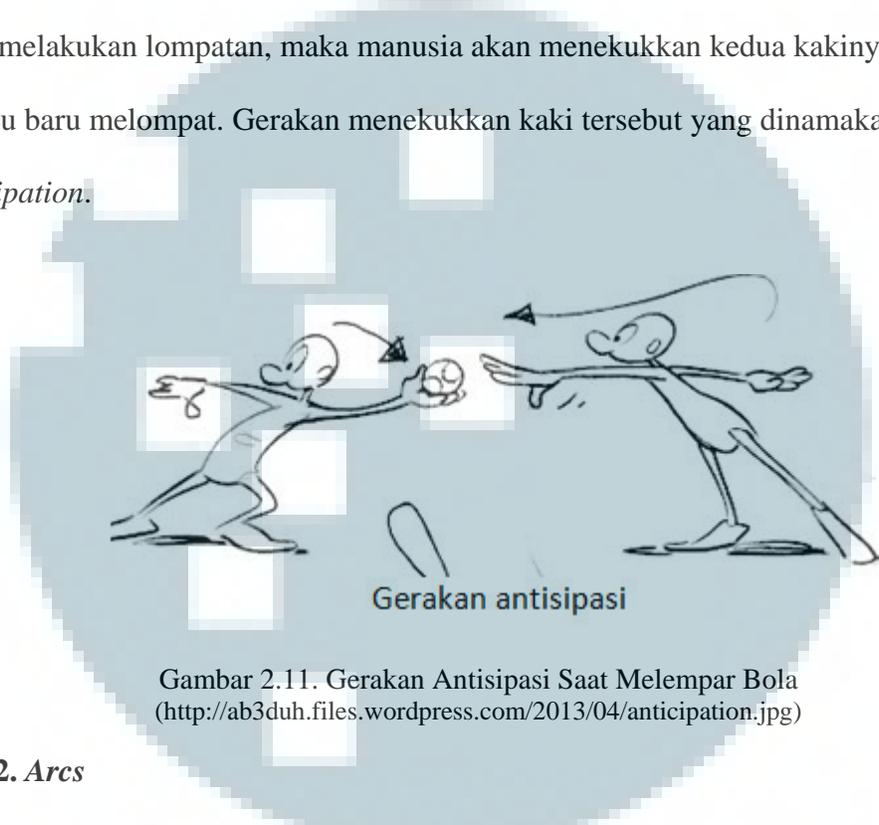
Salah satu kunci penting dalam membuat animasi adalah pengaturan dalam ketepatan waktu saat mengerjakan animasi.



Gambar 2.10. Contoh *Timing* dan *Spacing* pada Gerakan Bola  
(<http://road2animate.files.wordpress.com/2010/04/timing-and-spacing.png>)

### 2.1.11. Anticipation

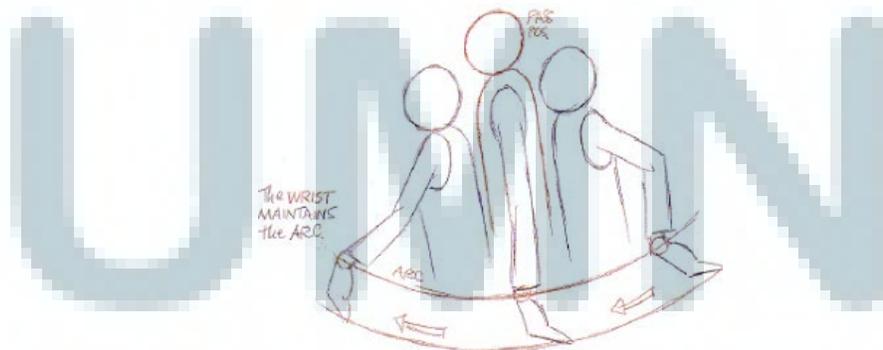
*Anticipation* adalah gerakan persiapan saat gerakan utama akan dimulai. Seperti saat akan melakukan lompatan, maka manusia akan menekukkan kedua kakinya terlebih dahulu baru melompat. Gerakan menekukkan kaki tersebut yang dinamakan *anticipation*.



Gambar 2.11. Gerakan Antisipasi Saat Melempar Bola  
(<http://ab3duh.files.wordpress.com/2013/04/anticipation.jpg>)

### 2.1.12. Arcs

Arcs adalah gerakan yang secara alami terjadi disebabkan oleh gravitasi, massa, dan sendi tulang yang bergerak melengkung.



Gambar 2.12. Contoh Gerakan Arc atau Melengkung  
([http://4.bp.blogspot.com/\\_fd868Dexycw/TOLZ2aGCdCI/AAAAAAAAAQo/nTXA2GoQU5Q/s320/arc.png](http://4.bp.blogspot.com/_fd868Dexycw/TOLZ2aGCdCI/AAAAAAAAAQo/nTXA2GoQU5Q/s320/arc.png))

## 2.2. Pengertian *Rigging*

Jika didefinisikan secara umum, *rigging* adalah pembuatan kerangka untuk model karakter dalam animasi 3D. Menurut Allen (2011), *rigging* adalah proses memperlengkapi suatu karakter dengan kumpulan kontrol yang membuat karakter tersebut mudah untuk dianimasikan. Kumpulan kontrol ini dapat terdiri dari persendian, tuas atau bahkan seleksi karakter terpisah. Sebuah animasi karakter sangat tergantung pada *rig* yang mengontrol modelnya. Kelly Murdock dalam bukunya “3ds Max 2012 Bible” (2011) mengatakan bahwa *rigging* adalah proses membuat struktur kerangka dan menegaskan batasan gerakan kerangka tersebut.

Dari sumber diatas, dapat disimpulkan bahwa *rigging* adalah proses membuat kerangka pada suatu karakter yang terdiri dari kumpulan kontrol untuk mempermudah animasi.

Untuk membuat sebuah *rig*, prosesnya dapat memakan waktu yang cukup lama. Namun waktu yang diluangkan untuk merencanakan *rig* sebelum tahap pembuatan dapat menghemat waktu saat memulai proses *rigging*. Dengan perencanaan yang baik, kita dapat memprediksi kebutuhan *rig* kita sebelum membuatnya. Allen dan Murdock (2008) mengatakan bahwa dalam tahap perencanaan, hal yang perlu diperhatikan adalah:

- Tipe gerakan yang akan dilakukan karakter
- Penampilan karakter

- Bagian tubuh yang perlu bergerak
- Bagian tubuh khusus yang lain

### 2.3. *Kinematics*

Menurut Murdock (2011), *kinematics* adalah suatu mekanika yang berhubungan dengan gerakan dari suatu sistem objek. Di dalam 3ds Max, istilah sistem objek adalah kumpulan objek yang saling terhubung dengan *link*. Setelah suatu sistem objek telah dibuat dan *parameter link* telah diatur, segala gerakan objek dibawah naungan *parent* objek dapat mempengaruhi gerakan *parent* objek tersebut, dengan menggunakan formula *kinematics*.

*Inverse Kinematics* atau IK adalah mekanika yang mirip dengan *kinematics*. Bedanya adalah IK mengatur dan mempengaruhi semua gerakan objek dalam suatu sistem ketika objek terakhir dalam hierarki sistem tersebut digerakkan. Dengan IK, kita dapat menggerakkan suatu sistem objek dengan menggerakkan objek turutan terakhirnya saja.

Pada hierarki sistem objek yang menggunakan *Forward Kinematics* atau FK, ketika objek pertama digerakkan posisi atau rotasinya, maka objek yang terhubung dengan objek pertama akan mengikuti gerakan yang dilakukan oleh objek pertama dalam hierarki sistem tersebut.

## 2.4. *Constraints*

Menurut Allen dan Murdock (2008), *constraints* adalah salah satu cara untuk mengontrol gerakan karakter secara otomatis. *Constraints* mengikat suatu animasi dengan cara tertentu. Dengan *constraints*, kaki karakter dapat dibuat agar tidak menembus tanah atau lantai.

Saat membuat *rig*, sangat disarankan untuk menentukan dua benda yang bergerak atau berputar secara bersamaan. Kadangkala hal ini dapat dicapai dengan hubungan hierarki. Ketika hubungan ini tidak memungkinkan, kita bisa menggunakan hubungan *constraint* untuk membuat suatu objek bergerak dan berputar secara bersamaan.

Objek yang telah diberi *constraint*, atau *constrained object*, bergerak, berputar, atau mengikuti objek targetnya. *Constraint* yang biasa digunakan untuk proses *rigging* adalah *point*, *orient*, *parent*, dan *pole constraint*. (Flaxman, 2008)

### 2.4.1. *Point Constraint*

Menurut Flaxman (2008), *point constraint* dapat diumpamakan seperti titik magnet yang menyambungkan suatu objek ke objek target. Saat objek target bergerak, maka objek yang telah di-*constraint* akan ikut bergerak. Kedua objek tersebut tetap mempertahankan posisi global mereka. Namun rotasi tidak termasuk dalam *point constraint* ini.

#### **2.4.2. Parent Constraint**

*Parent constraint* menyebabkan suatu *constrained object* bergerak bersamaan dengan objek penggerak atau pengendalinya, seperti hubungan *parent-child*. Bedanya, *parent constraint* membuat hubungan dengan atribut, sedangkan *parent-child* menggunakan hubungan hierarki. Satu objek dapat diberi *parent constraint* ke banyak objek penggerak (Allen dan Murdock, 2008).

#### **2.4.3. Orient Constraint**

*Orient constraint* memperbolehkan *constrained objects* untuk bergerak secara independen satu sama lain, namun orientasi di *world space* diberi kontrol sehingga tiap objek bergerak ke arah yang sama. Singkatnya, *orient constraint* membuat orientasi suatu objek mengikuti orientasi objek lainya (Flaxman, 2008).

#### **2.4.4. Aim Constraint**

*Aim constraint* menyebabkan *constrained objects* mengarah ke objek yang ditandai. Ini berbeda dengan *orient constraint* dimana setiap objek mengarah ke suatu objek yang spesifik. *Aim constraint* membatasi rotasi objek yang digerakkan sehingga objek tersebut selalu mengarah ke objek penggeraknya. Biasanya digunakan untuk membuat mata mengikuti gerakan suatu objek.

#### **2.4.5. Pole Vector Constraint**

*Pole vector constraint* menetapkan suatu vektor sebagai dasar bending atau pembengkokan suatu IK. Ketika menggunakan *constraint* ini, vektor biasanya ditandai

dengan node ketiga, yang berhubungan dengan awal dan akhir suatu IK *chain* atau urutan IK. Vektor tersebut mengontrol arah lekukan sendi dengan IK *Chain* tersebut.

#### **2.4.6. Other Constraint**

Selain *constraint* yang telah disebutkan diatas, masih ada beberapa jenis yang lain. *Scale constraint* menyebabkan suatu objek mengikuti skala objek penggerakannya. Kita juga dapat membuat suatu objek menempel ke permukaan objek lain dengan *geometryconstraint*. Terdapat juga *normal constraint* yang membuat arah orientasi suatu objek mengarah ke *normal* permukaan objek lain. *Normal* tersebut tegak lurus dengan permukaan objek penggerak. Biasanya *value* arahnya diambil dari satu *poly* atau *vertex*.

### **2.5. Jenis Rigging**

#### **2.5.1. Parent and Child**

Salah satu bagian dasar dari *rigging* adalah hierarki, dengan kata lain, bagian model mana saja yang dapat mengontrol atau menggerakkan bagian lain. Di dalam suatu hierarki, objek yang menggerakkan atau mengontrol disebut *parent*, dan objek yang dikontrol disebut *child* (Chopine, 2011). Di dalam penggambaran hierarki, *parent* berada diatas *child*. Objek *child* juga bisa menjadi *parent* dari objek yang lain.

#### **2.5.2. Bones and Joints**

*Joints*, atau lebih tepatnya *joint deformaters*, adalah bagian daerah dimana suatu model dapat dibengkokkan, dilekukan atau diputar (Chopine, 2011). Di dalam animasi 3D,

*joints* adalah salah satu teknik dasar dimana semua aplikasi 3D memiliki cara pembuatan *joints* yang hampir sama. Setelah mengaktifkan alat untuk membuat *joints*, sebuah *joint* akan dibuat dimana kita mengklik. Jika kita mengklik lagi selagi alat *joints* masih aktif, maka sebuah bone dan *joints* akan dibuat. *Joint* kedua ini akan jadi *child* dari *joint* pertama. *Bone* adalah penghubung visual antar *joints* dan menandakan *joint* mana yang menjadi *parent* atau *child*.

### 2.5.3. *Skeletons*

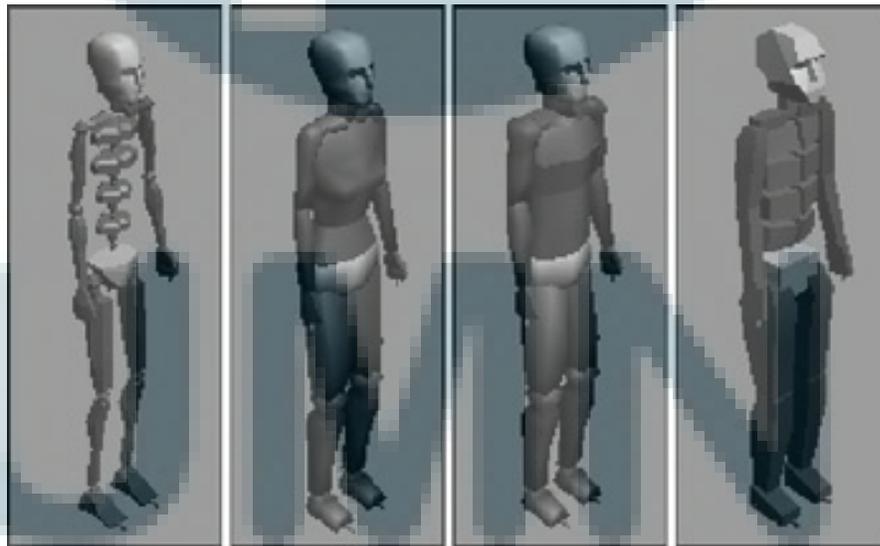
Dengan *joints*, kita dapat membuat hierarki *joints* yang kompleks seperti struktur rangka yang terdiri dari tangan, kaki, kepala, dan badan. Hierarki seperti ini disebut *skeleton*, atau *skeleton rig*. Hal ini biasanya dibutuhkan dalam model yang rumit seperti robot, manusia atau binatang. Beberapa aplikasi 3D menyediakan *skeleton rig* standar untuk memudahkan proses *rigging*. Menurut Jones (2012), di dalam software 3ds Max terdapat 2 jenis *skeleton rig* yaitu *biped* dan *CAT*.

#### 1. *Biped*

Dalam 3ds Max, *Biped* adalah struktur *humanoid* atau manusia standar dari program tersebut (Derakhshani, 2012). *Biped* berbeda dengan sistem *bones*, *biped* disajikan utuh dalam persentase model tubuh manusia sehingga animator tidak perlu membuat susunan tulangnya lagi terlebih dahulu (Ramadhan, 2006).

FK dan IK dalam *biped* juga telah diatur, misalnya dalam struktur tangan hingga lengan atas dan kaki hingga paha. Contoh sistem FK bila bagian paha digerakkan maka bagian betis dan telapak kaki akan mengikuti pergerakan yang dilakukan oleh bagian paha dan bila bagian lengan atas digerakkan maka bagian lengan tengah hingga telapak tangan beserta dengan jari-jarinya akan mengikuti pergerakan juga. Selain FK, *biped* juga memiliki sistem IK seperti pada bagian telapak tangan dan bagian kaki juga. Contohnya bila telapak tangan digerakkan maka bagian lengan dan lengan atas akan berotasi sesuai dengan pergerakan dari telapak tangan ( Bousquet, 2006).

Dalam sistem *biped*, ada 4 macam tipe struktur tulang *biped* yaitu *Skeleton*, *Female*, *Male*, dan *Classic*.



Gambar 2.13. Tipe *Biped* dari Kiri yaitu *Skeleton*, *Female*, *Male*, dan *Classic*.  
(<http://flylib.com/books/4/68/1/html/2/images/fig42-2.jpg>)

*Skeleton* mempresentasikan *biped* sesuai dengan struktur tulang, *Male* dan *Female* mempresentasikan *biped* sesuai dengan gendernya, sedangkan *Classic* adalah bentuk *biped* yang berbentuk standar.

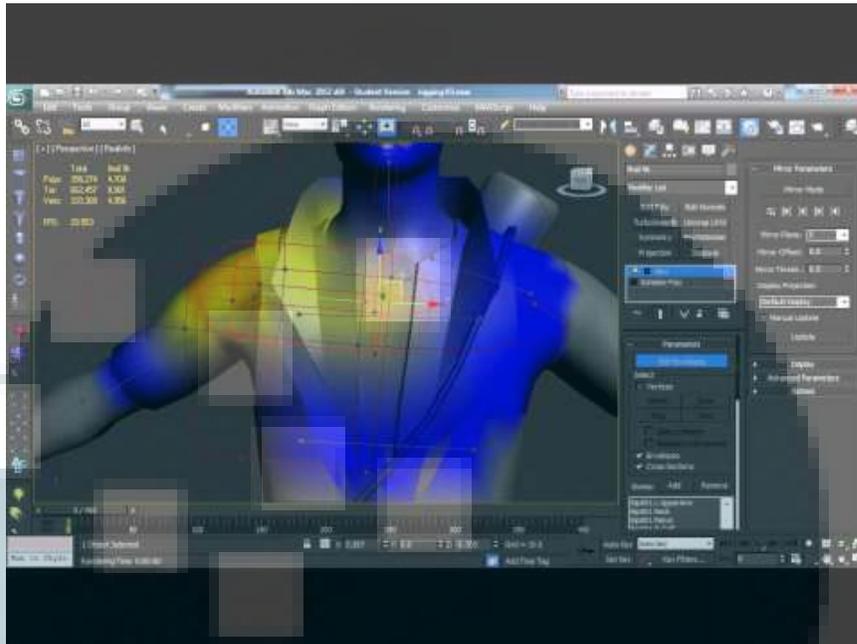
## 2. CAT

CAT adalah singkatan dari Character Animation Toolkit, yaitu salah satu fasilitas *rigging* yang terdapat dalam 3ds Max. Mirip seperti *Biped*, CAT adalah salah satu *predefined rig*. CAT memfasilitasi *rigging*, animasi nonlinear, *animation layering*, *motion capture* dan juga simulasi otot (Jones, 2012). CAT menyediakan beberapa bentuk *predefined rig*, tidak seperti *biped* yang hanya menyediakan struktur bentuk *humanoid* atau manusia.

### 2.6. Skinning

Setelah membuat dan memposisikan suatu *rig*, *rig* tersebut tidak langsung menggerakkan model 3D sampai kita menghubungkannya ke model tersebut. Proses menghubungkan *rig* ke model disebut *skinning*. Menurut Bousquet (2011), *skinning* adalah proses menghubungkan sebuah model karakter dengan *skeleton rig*.

*Skinning* bekerja dengan menghubungkan tiap *bones* dengan *vertex* tertentu di sebuah model. Di dalam aplikasi 3ds Max, digunakan *modifier Skin* untuk melakukan proses *skinning*. Jika suatu *vertex* dipengaruhi oleh lebih dari satu *bone*, maka pengaruh *bone* di *vertex* tersebut dipengaruhi oleh *weight value*. Seluruh nilai *weight vertex* dapat bertambah hingga 1.0, dan dengan nilai *weight* 1.0 berarti *bone* memiliki pengaruh penuh terhadap *vertex* tersebut.

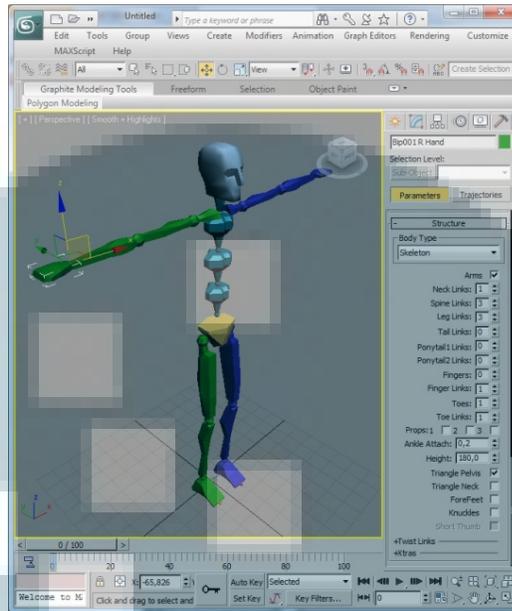


Gambar 2.14. Contoh *Skinning*  
([http://i4.ytimg.com/vi/fpU1yqsT8\\_A/0.jpg](http://i4.ytimg.com/vi/fpU1yqsT8_A/0.jpg))

## 2.7. Pose

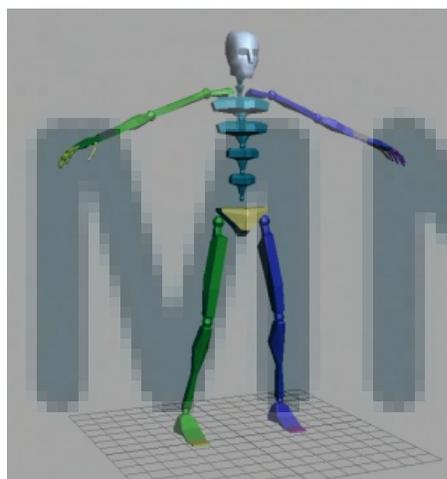
Dalam membuat model karakter untuk yang akan diberi sistem *rig*, salah satu hal yang perlu diperhatikan adalah ketika membuat pose *default* karakter. Menurut Allen & Murdock (2008), ada 2 pose umum, yaitu *T-pose* standar dan *relaxed T-Pose*.

*T-pose* standar adalah pose dimana karakter menghadap ke depan dengan lengan mengarah 90 derajat ke samping seperti huruf T. Kaki karakter juga mengarah ke samping sesuai lebar bahu. Walaupun ini adalah salah satu pose umum dalam banyak karakter, pose ini tidak natural dan dapat menyebabkan masalah dalam proses *rigging*.



Gambar 2.15. Contoh T-Pose  
<http://www.vali.de/wp-content/uploads/motionbuilderF.jpg>

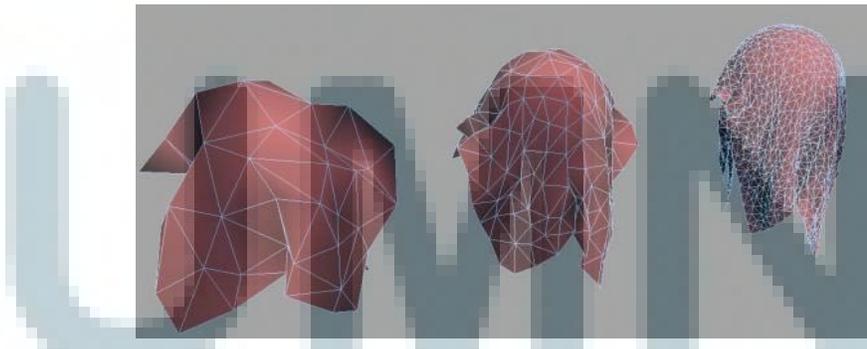
*Relaxed T-pose* adalah pose yang mirip dengan T-pose standar. Dalam relaxed T-pose, lengan karakter diturunkan 45 derajat. Dengan menempatkan lengan dengan sudut 45 derajat, lengan akan berada di area dimana mayoritas gerakan akan terjadi.



Gambar 2.16. Contoh *Relaxed T-Pose*  
[http://hewiki.heroengine.com/images/thumb/c/c6/Weapon\\_rig1.jpg/300px-Weapon\\_rig1.jpg](http://hewiki.heroengine.com/images/thumb/c/c6/Weapon_rig1.jpg/300px-Weapon_rig1.jpg)

## 2.8. Cloth

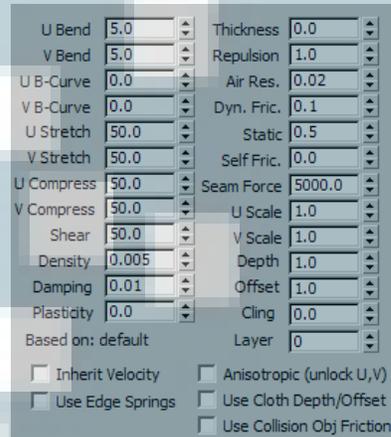
Pada proses pembuatan *rigging*, ada sistem modifikasi yang disebut dengan *cloth*. Menurut Murdock (2011), *cloth* dapat diaplikasikan ke objek yang akan dijadikan menyerupai kain. Objek yang dijadikan kain atau diberi sistem *cloth* disebut *cloth object*. Objek yang dijadikan *cloth object* diberi karakteristik yang membuatnya bereaksi dengan faktor penggerak lain seperti gravitasi, angin, atau objek lainnya. Setelah semua objek, penggerak, dan *cloth object* telah dibuat untuk proses animasi, simulasi dapat dilakukan untuk melihat bagaimana reaksi gerakan *cloth object* tersebut berinteraksi dengan faktor-faktor penggerak yang ada pada animasi yang dibuat. Pada saat membuat *cloth object*, resolusi dari objek tersebut harus mencukupi untuk simulasi agar dapat melakukan gerak lekukan seperti kain atau bahan lentur lainnya. Dari gambar berikut akan diperlihatkan bagaimana pengaruh resolusi terhadap gerak lekuk pada *cloth object*.



Gambar 2.17. Pengaruh Resolusi Pada Gerakan Lekuk *Cloth Object*.

(<http://docs.autodesk.com/3DSMAX/16/ENU/3ds-Max-Help/images/GUID-3A427A85-5EF7-47F4-8596-CB25366BCEB6-low.png>)

Intensitas gerak lekuk *cloth object* dapat ditentukan dari konfigurasi properti sistem *cloth*. Berikut adalah konfigurasi yang dapat diatur dalam sistem *cloth* terhadap *cloth object*.

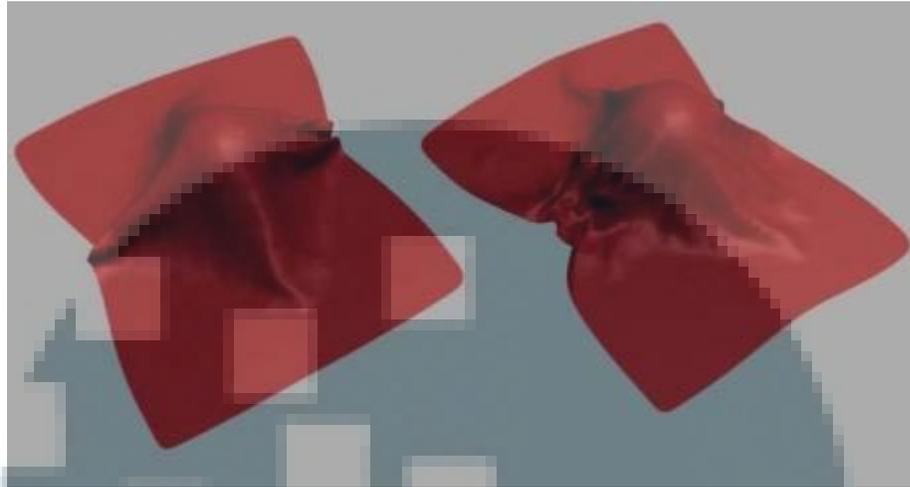


Gambar 2.18. Konfigurasi pada sistem *cloth*

(<http://docs.autodesk.com/3DSMAX/15/ENU/3ds-Max-Help/images/GUID-A2A4D56B-1550-42BD-84EC-D9512AB556FB-low.png>)

### 2.8.1. *U Bend / V Bend*

Konfigurasi ini menentukan lekukkan yang akan terjadi pada *cloth object*. Semakin tinggi *value* konfigurasi ini, maka *cloth object* akan semakin sulit untuk melakukan lekukkan saat disimulasi.



Gambar 2.19. *Value Bend* Objek Kiri Lebih Tinggi Dibandingkan Objek Kanan.  
(<http://docs.autodesk.com/3DSMAX/15/ENU/3ds-Max-Help/images/GUID-4CBCE86D-4881-4159-8271-AF3DCAA0C508-low.png>)

### 2.8.2 *U-B Curve / V-B Curve*

Konfigurasi ini mengatur lekukkan *cloth object* saat melipat. *Value 0* pada konfigurasi akan membuat resistensi lekuk *cloth object* menjadi konstan atau tidak berlaku sehingga *cloth object* dapat melekok. *Value 1* akan membuat *cloth object* menjadi sangat resistan dengan lekukkan. *Value* konfigurasi ini dinaikkan agar membuat *cloth object* tidak saling menumpuk saat melipat.

### 2.8.3. *U Stretch / V Stretch*

Konfigurasi ini mengatur peregangan pada *cloth object*. Semakin tinggi *value* konfigurasi ini maka *cloth object* akan semakin kaku dan semakin rendah *value* konfigurasi ini akan membuat *cloth object* menjadi seperti karet yang renggang.

#### **2.8.4. *U Compress / V Compress***

Konfigurasi berkebalikkan dari konfigurasi *stretch*. Konfigurasi ini mengatur kerapatan dari *cloth object*. Konfigurasi ini bisa membuat *cloth object* menjadi kusut atau mengkerut bila diberi *value* yang tinggi, sedangkan *value* yang rendah dapat membuat *cloth object* merapat dengan baik.

#### **2.8.5. *Shear***

Konfigurasi *shear* ini menentukan bagaimana tiap *polygon* dapat berubah bentuk saat simulasi *cloth object*. Konfigurasi ini membuat sisi-sisi *polygon* dapat meregang. Konfigurasi *shear* ini hanya berlaku untuk *polygon* dari *cloth object* yang tidak menempel pada objek yang menjadi pengikat atau menempel pada *cloth object*.

#### **2.8.6. *Density***

Konfigurasi *density* ini mengatur massa atau berat pada *cloth object* per sentimeter kubik. Semakin tinggi *value* konfigurasi ini maka *cloth object* akan menjadi seperti kain celana *jeans*, sedangkan jika *value density* diturunkan maka *cloth object* akan menjadi seperti kapas atau kain sutra.

#### **2.8.7. *Damping***

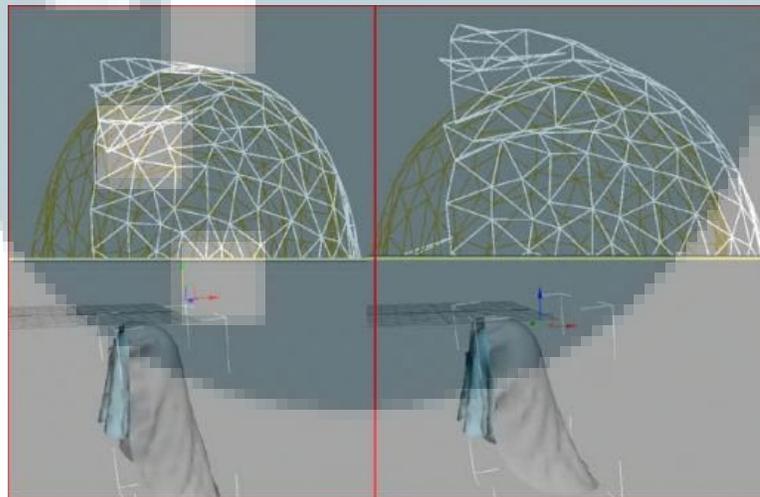
*Damping* menentukan cepat dan lambatnya reaksi *cloth object* pada saat pergerakan simulasi. *Value* yang tinggi pada konfigurasi ini akan membuat *cloth object* terlihat bergerak dalam keadaan basah, sedangkan pada *value* yang rendah *cloth object* akan terlihat lebih reaktif saat simulasi.

### 2.8.8. *Plasticity*

*Plasticity* cenderung membuat *cloth object* untuk menjaga perubahan bentuknya. Pada *value* 100 untuk konfigurasi *plasticity*, *cloth object* tidak akan mengubah bentuk antar *polygon* saat simulasi.

### 2.8.9. *Thickness*

*Thickness* mengatur ketebalan *cloth object* secara virtual. Pada *value* yang tinggi, *thickness* membuat *polygon* pada *cloth object* menjadi lebih terpisah.



Gambar 2.20. *Value Thickness* Objek Kiri 0, *Value Thickness* Objek Kanan 9.  
(<http://docs.autodesk.com/3DSMAX/15/ENU/3ds-Max-Help/images/GUID-8EC47604-BD62-4363-AFCE-D3AE6FBB1EA0-low.png>)

### 2.8.10. *Repulsion*

*Repulsion* menentukan besarnya kekuatan untuk menahan *cloth object* lain. Konfigurasi ini menjaga agar *cloth object* tidak saling menyatu saat melekok pada gerakan simulasi.

### **2.8.11. Air Res. (Air Resistance)**

Konfigurasi ini menentukan ketahanan *cloth object* dari tekanan udara. *Value* konfigurasi *air resistance* yang tinggi berguna untuk tekstur kain yang dijahit dengan rapat.

### **2.8.12. Dyn. Fric. (Dynamic Friction)**

*Dynamic friction* mengatur gesekan pada saat *cloth object* bergesekkan dengan objek lain. Semakin besar *value* konfigurasi ini, maka *cloth object* akan semakin sulit meluncur terhadap objek lain. *Value* konfigurasi yang rendah akan membuat *cloth object* lebih mudah meluncur terhadap objek lain.

### **2.8.13. Static Fric. (Static Friction)**

*Static friction* mengatur pergerakan *cloth object* pada saat keadaan diam. Gerakan-gerakan ini berupa gerakan kecil seperti kain yang terkena angin kecil.

### **2.8.14. Self Fric. (Self Friction)**

Konfigurasi ini mengatur gesekan antar *polygon* dari *cloth object* itu sendiri. Semakin tinggi *value* konfigurasi ini akan memberi gesekan lebih diantara *polygon* dari *cloth object* saat bersentuhan.

### **2.8.15. Seam Force**

Konfigurasi ini mengatur kekuatan sambungan pada *polygon* dari *cloth object*.

### **2.8.16. U Scale / V Scale**

*U Scale* dan *V Scale* mengatur seberapa banyak *cloth object* melebar dan menyusut.

### **2.8.17. Depth**

*Depth* mengatur kedalaman *polygon* dari *cloth object* saat bertumbukkan satu sama lain.

### **2.8.18. Offset**

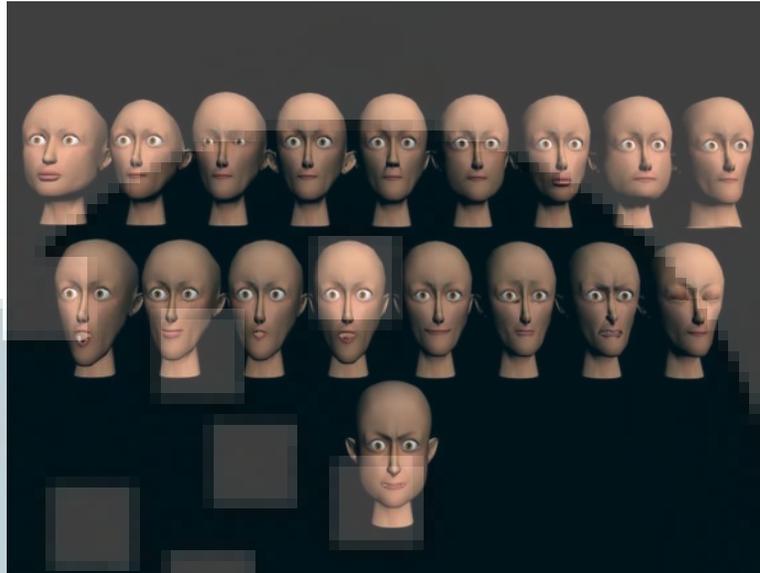
Konfigurasi ini menentukan jarak antara *cloth object* dengan objek yang ditempel oleh *cloth object*. Semakin kecil *value* konfigurasi ini, maka jarak antara objek dengan *cloth object* akan semakin dekat dan *value* konfigurasi tinggi maka *cloth object* akan terlihat melayang.

### **2.8.19. Cling**

*Cling* menentukan seberapa berpengaruh *cloth object* saat berinteraksi dengan objek lainnya. *Cling* berpengaruh pada efek-efek yang ingin dilakukan pada *cloth object*.

## **2.9. Morpher**

Maestri (2008) mengatakan *morpher* adalah *modifier* yang ada pada *software 3ds Max* yang berfungsi untuk mengubah bentuk objek agar sama dengan objek lainnya.



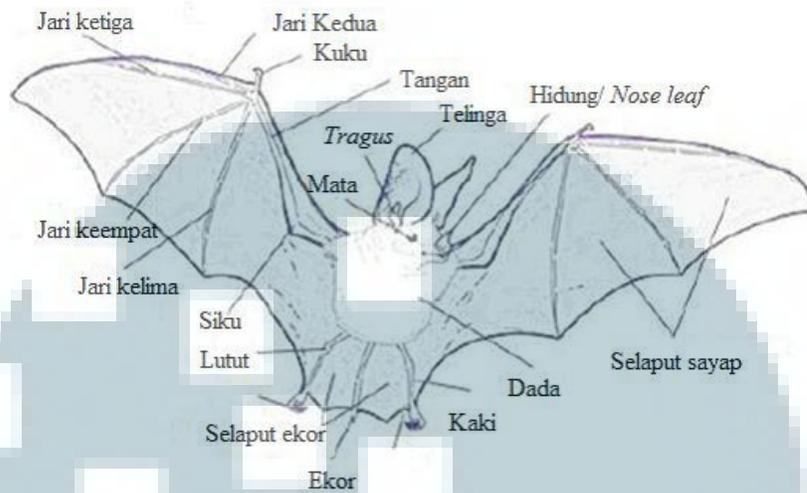
Gambar 2.21. Bentuk-Bentuk Morpher  
([http://previewcf.turbosquid.com/Preview/2014/05/26\\_\\_08\\_46\\_12/head%20creation3.jpgc43653a8-333b-4f07-88d1-ad9a6e71e514Larger.jpg](http://previewcf.turbosquid.com/Preview/2014/05/26__08_46_12/head%20creation3.jpgc43653a8-333b-4f07-88d1-ad9a6e71e514Larger.jpg))

### 2.10. *Reaction Manager*

Taylor dan Francis (2012) mengatakan *reaction manager* adalah alat yang memungkinkan untuk mengubah dan mengatur jalannya pergerakan animasi dan menghubungkannya dengan pergerakan objek lain.

### 2.11. Anatomi Kelelawar

Feldhammer (2007) mengatakan kelelawar merupakan salah satu hewan yang unik, karena kelelawar merupakan satu-satunya mamalia yang dapat terbang. Walaupun dapat terbang, struktur tulang kelelawar lebih mirip dengan struktur tulang manusia dibandingkan burung.

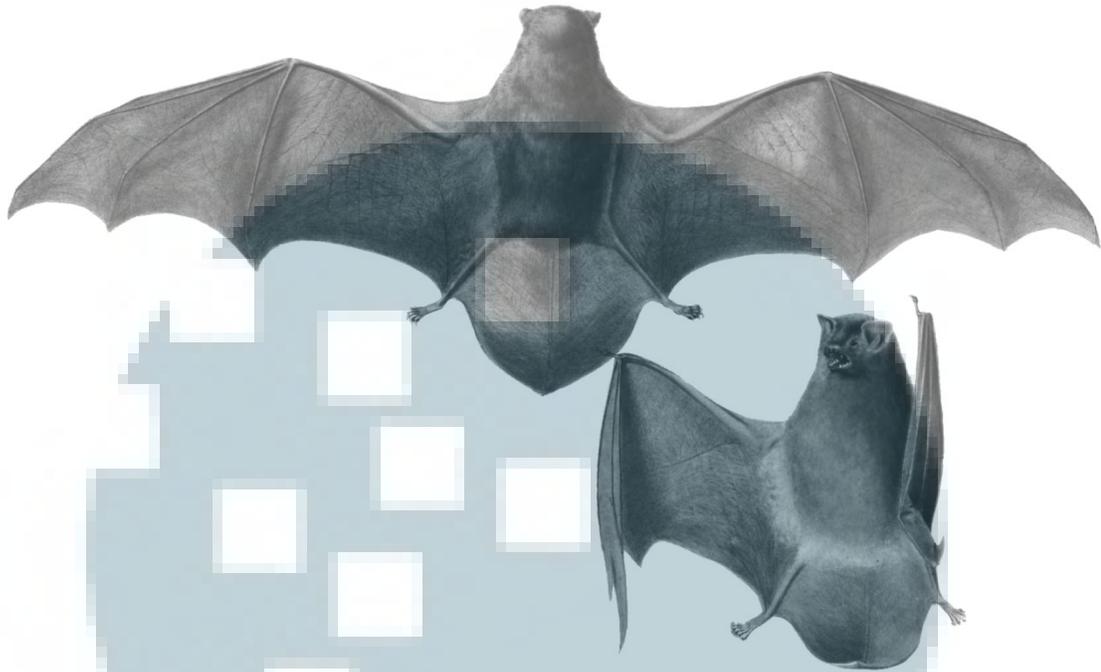


Gambar 2.22. Anatomi Fisik Kelelawar dan Keterangan Tiap Bagian Tubuhnya  
 ([http://www.cawildlife911.org/\\_images/bat\\_diagram.jpg](http://www.cawildlife911.org/_images/bat_diagram.jpg))

### 2.11.1 Sayap dan Jari-Jari Kelelawar

Hermanson (1998) mengatakan bahwa kelelawar merupakan salah satu anggota mamalia yang termasuk ke dalam ordo Chiroptera yang berarti mempunyai “sayap tangan”, karena kaki depan atau biasa disebut tangan bermodifikasi sebagai sayap. Akan tetapi sayap kelelawar memiliki bentuk dan struktur yang berbeda dengan sayap burung.

Pada sayap kelelawar tidak tumbuh bulu yang panjang seperti pada burung, melainkan ekstensi dari kulit di sela-sela jari kelelawar yang melebar dan tipis, selain itu kulit sayap kelelawar dapat meregang dan menyempit. Walaupun tipis kulit sayap kelelawar cukup kuat untuk menahan tekanan udara saat terbang (Feldhammer, 2007).

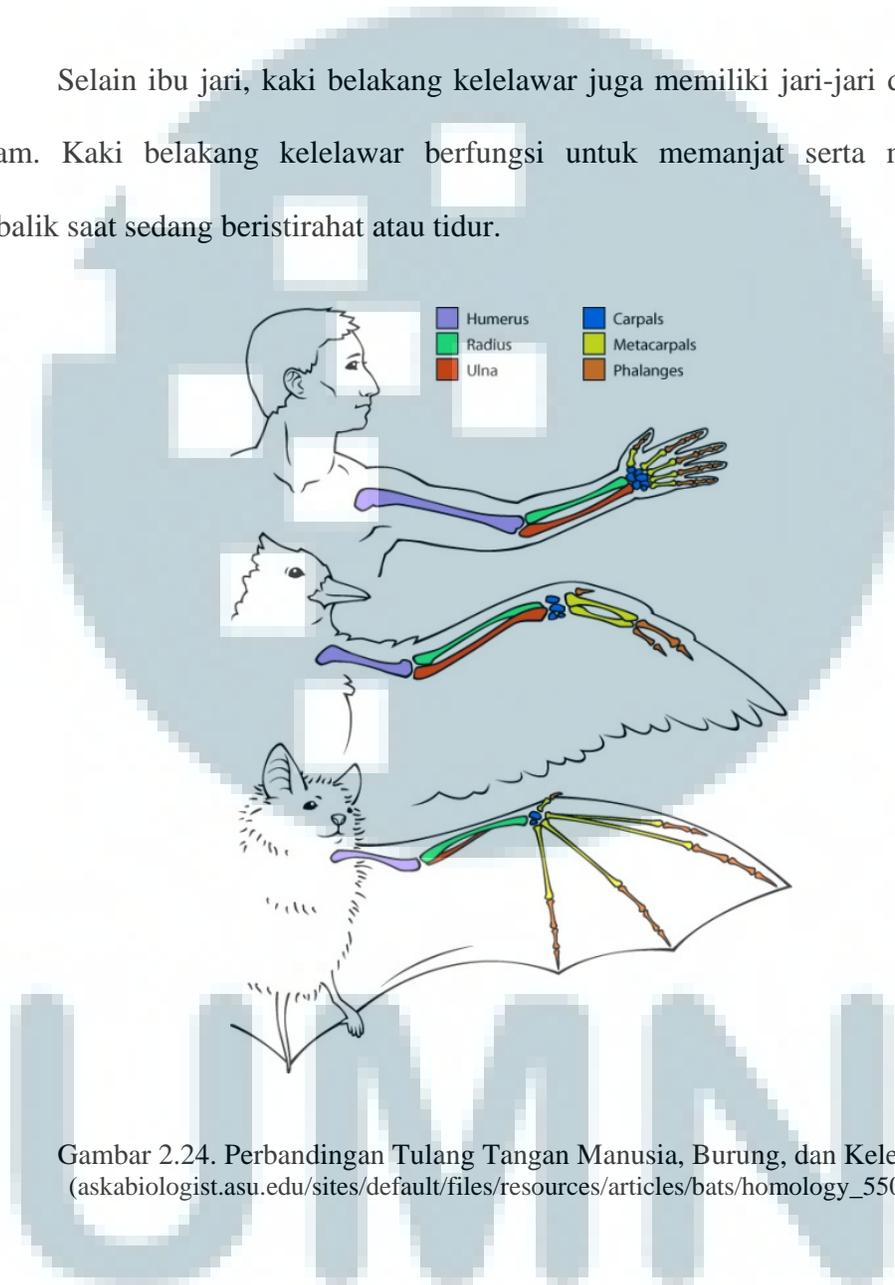


Gambar 2.23. Sayap Kelelawar Saat Melebar dan Menyempit.  
([upload.wikimedia.org/wikipedia/commons/d/d8/Miniopterus\\_schreibersii\\_dasythrix.jpg](http://upload.wikimedia.org/wikipedia/commons/d/d8/Miniopterus_schreibersii_dasythrix.jpg))

Thomas (1987) mengatakan bahwa kelelawar memiliki nama latin yang bermakna “sayap tangan” karena struktur tulang sayap yang dimiliki kelelawar mirip dengan tangan manusia yaitu memiliki 5 jari, hanya perbedaannya 4 jari yang dimiliki kelelawar sangat panjang. Walaupun struktur jari-jari sayap kelelawar mirip dengan manusia, jari sayap kelelawar merupakan tulang rawan yang tipis dengan kadar kalsium sedikit sehingga memungkinkan jari kelelawar dapat membengkok pada saat mengepakkan sayap. Berbeda dengan ibu jari kelelawar yang pendek dan tidak tersambung dengan kulit sayapnya, ibu jari kelelawar berukuran pendek dan memiliki cakar tajam yang berfungsi untuk mencengkeram atau memegang sehingga dapat digunakan untuk memanjat dan bergelantungan. Pada saat masih muda, kelelawar

menggunakan cakar pada ibu jarinya untuk menggantung pada induknya saat menyusui.

Selain ibu jari, kaki belakang kelelawar juga memiliki jari-jari dengan cakar tajam. Kaki belakang kelelawar berfungsi untuk memanjat serta menggantung terbalik saat sedang beristirahat atau tidur.



Gambar 2.24. Perbandingan Tulang Tangan Manusia, Burung, dan Kelelawar ([askabiologist.asu.edu/sites/default/files/resources/articles/bats/homology\\_550.jpg](http://askabiologist.asu.edu/sites/default/files/resources/articles/bats/homology_550.jpg))

### 2.11.2 Telinga Kelelawar dan Ekolokasi

Kelelawar juga memiliki kelebihan lain selain terbang yaitu kemampuan ekolokasinya. Aldridge (1987) mengatakan ekolokasi adalah kemampuan untuk mengetahui sebuah benda menggunakan pantulan suara. Feldhammer (2007) juga berpendapat bahwa kemampuan ekolokasi dapat memberi informasi yang tidak berbeda jauh dengan penglihatan. Dalam ekolokasi, kelelawar mengeluarkan suara dengan frekuensi dan intensitas yang tinggi lewat mulut atau hidungnya. Getaran suara yang dihasilkan terpancar dan bila mengenai benda akan terpantul kembali berbalik menuju saat getaran suara tersebut dipancarkan. Saat getaran suara terpantul, getaran suara diterima oleh kelelawar melalui telinga mereka. Hal ini sangat jelas terlihat dari bentuk kuping kelelawar yang relatif besar dan cekung ke dalam.

Telinga kelelawar terbentuk dari tulang rawan yang menyambung langsung dengan tengkorak kelelawar, tulang rawan telinga kelelawar memungkinkan telinga kelelawar dapat menekuk. Seperti terlihat pada beberapa jenis kelelawar yang ada pada gambar berikut.



Gambar 2.25. Contoh Jenis-Jenis Kelelawar

- 1 (<http://idahoptv.org/dialogue4kids/season1/bats/images/FPbat3.jpg>)
- 2 (<http://www.factzoo.com/sites/all/img/mammals/bats/long-eared-bat.jpg>)
- 3 ([http://upload.wikimedia.org/wikipedia/commons/c/ce/Plecotus\\_auritus01.jpg](http://upload.wikimedia.org/wikipedia/commons/c/ce/Plecotus_auritus01.jpg))