



### **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. Animasi 3D

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 ketiga sumber diatas dapat disimpulkan bahwa animasi adalah rangkaian gambar diam atau statis yang dapat menghasilkan ilusi gerakan.

Proses produksi animasi 3D terdiri dari beberapa tahap. Menurut Parent (2009), proses produksi animasi 3D terdiri dari 4 tahap utama:

1. *Modeling*
2. *Animation*
3. *Texturing & Lighting*
4. *Rendering*

*Modeling* adalah tahap paling awal dalam produksi animasi 3D. *Modeling* mengacu pada pembuatan model, seperti karakter dan latar belakang. Menurut Beane (2012), model adalah representasi permukaan geometris sebuah objek yang dapat dilihat dan digerakkan di dalam *software* 3D. Ada banyak teknik untuk membuat suatu model 3D. Model dapat dibuat dari awal dengan *software* 3D

seperti Autodesk Maya atau 3ds Max. Model juga dapat dipahat dan diukir secara digital dengan Mudbox atau Zbrush. Proses ini membutuhkan waktu yang cukup lama dan juga membutuhkan perhatian lebih untuk membuat detail model agar terlihat seperti aslinya.

Animasi adalah tahap setelah proses *modeling* selesai. Setelah model siap, model tersebut perlu dianimasikan sesuai *storyboard* yang ada. Menurut Beane (2012), ada 3 tipe animasi dalam media 3D, yaitu *keyframed animation*, *motion capture*, dan *procedural animation*. Animasi adalah aspek penting dalam *workflow* 3D. Walaupun model sudah cukup bagus, namun jika animasinya kurang, dapat menurunkan kualitas suatu animasi 3D secara keseluruhan.

Menurut Parent (2009), biasanya *texturing* dan *lighting* dilakukan setelah animasi model selesai. Namun tahap ini dapat juga dilakukan sebelum animasi. *Texturing* adalah proses memberi warna, tekstur dan material ke dalam model. Tujuan *texturing* adalah agar permukaan model dapat terlihat seperti desain yang ada di dalam *concept art* atau seperti objek aslinya di dunia nyata. *Lighting* atau pencahayaan adalah proses memberi cahaya dalam suatu adegan animasi 3D. Pencahayaan dalam aplikasi 3D mirip dengan pencahayaan di dunia nyata. Di dalam aplikasi 3D terdapat banyak jenis cahaya seperti *spotlight*, atau cahaya matahari.

Setelah proses *texturing* dan *lighting* selesai, maka *scene* animasi 3D perlu di-*render* untuk melihat hasilnya. *Rendering* adalah tahap terakhir dari proses produksi. *Rendering* adalah proses mentranslasikan suatu konten 3D ke sebuah

gambar 2D. Setelah tahap ini selesai, tahap berikutnya adalah *post production* atau pasca produksi, dimana hasil *rendering* akan melalui proses *compositing* untuk menjadi sebuah animasi 3D.

## 2.2. *Rigging* dalam Animasi

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 sistem pada suatu karakter yang terdiri dari kumpulan kontrol untuk mempermudah animasi. Proses *rigging* dilakukan sebelum tahap animasi dimulai.

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:

1. Tipe gerakan yang akan dilakukan karakter

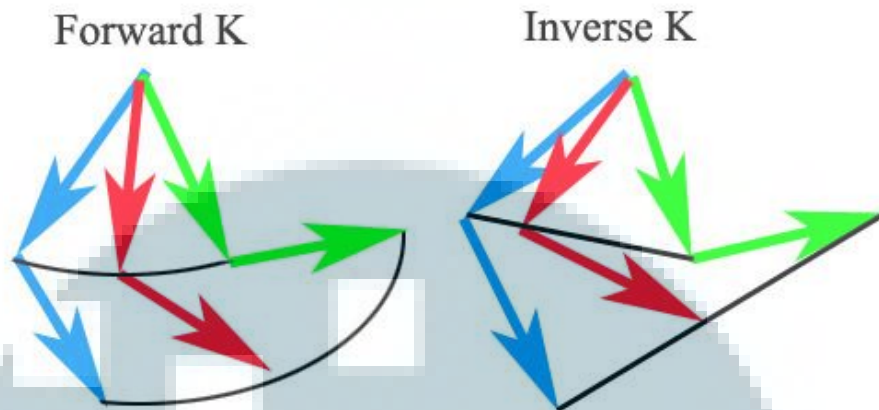
2. Penampilan karakter
3. Bagian tubuh yang perlu bergerak
4. Bagian tubuh khusus yang lain

### 2.2.1. Mekanika *Rigging*

#### 1. *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 object* dapat mempengaruhi gerakan *parent object* tersebut, dengan menggunakan formula *kinematics*.

*Inverse Kinematics* atau IK adalah salah satu dari dua mekanika *kinematics*. Bedanya adalah IK mengatur dan mempengaruhi semua gerakan objek dalam suatu sistem ketika objek terakhir atau objek *child* dalam hierarki sistem tersebut digerakkan. Dengan IK, kita dapat menggerakkan suatu sistem objek dengan menggerakkan objek turutan terakhirnya saja. Sedangkan *Forward Kinematics* atau FK, menyebabkan objek paling bawah di dalam struktur hierarki bergerak jika objek *parent* digerakkan. Ketika kita menggerakkan suatu hierarki, objek *child* ikut bergerak dengan objek *parent*, namun objek *child* tersebut juga dapat digerakkan secara bebas tanpa pengaruh *parent*.



Gambar 2.1. *Kinematics*  
 ([http://udn.epicgames.com/Three/rsrc/Three/AnimationNodes/FK\\_VS\\_IK.jpg](http://udn.epicgames.com/Three/rsrc/Three/AnimationNodes/FK_VS_IK.jpg))

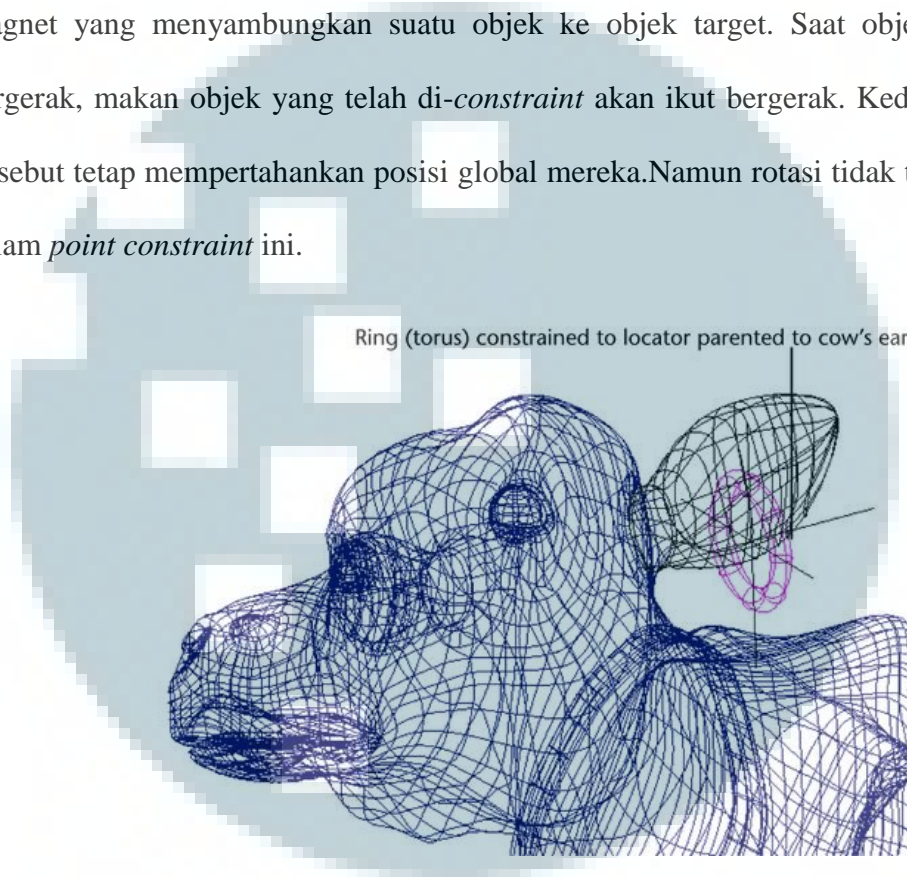
## 2. *Constraints*

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)

**a. *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.

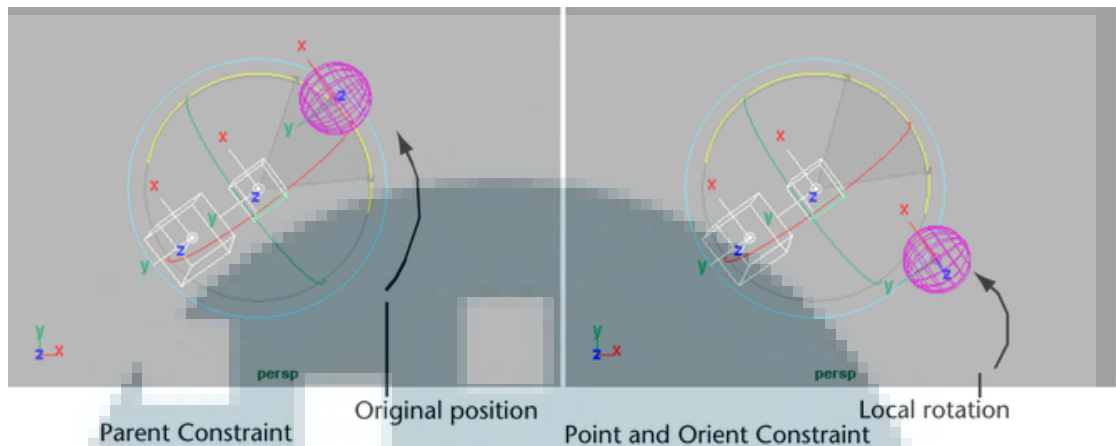


Gambar 2.2. *Point Constraint*

([http://download.autodesk.com/us/maya/2011help/images/MED/Sherlock/CharSetup/comp\\_pointConstCowRing.png](http://download.autodesk.com/us/maya/2011help/images/MED/Sherlock/CharSetup/comp_pointConstCowRing.png))

**b. *Parent Constraint***

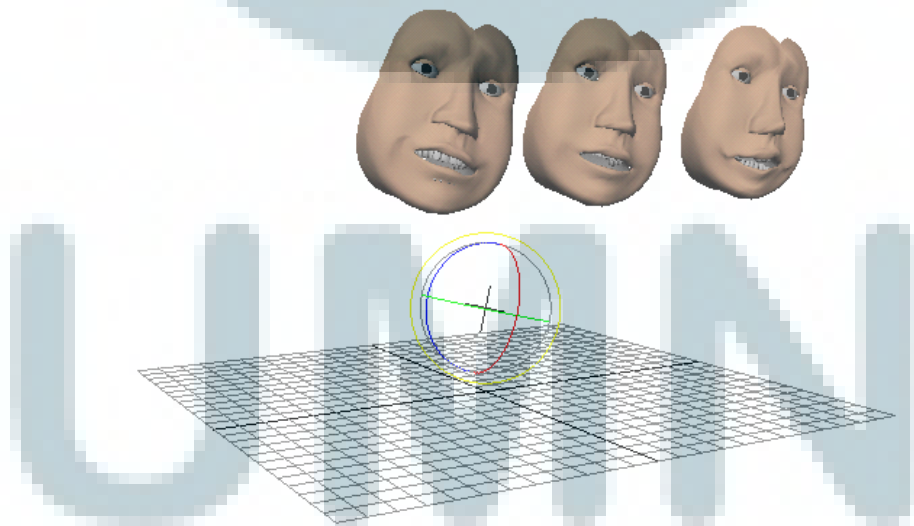
*Parent constraint* menyebabkan suatu constrained object bergerak bersamaan dengan objek penggerak atau pengendalinya, seperti hubungan *parent-child*. Bedanya, *parent constraint* membuat hubungan dengan atribut, sedangkan *parent-child* menggunakan hubungan hierarki. Satu objek dapat diberi *parent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



Gambar 2.3. *Parent & Orient Constraint*  
 ([http://download.autodesk.com/global/docs/maya2014/en\\_us/images/comp\\_PointOrient.png](http://download.autodesk.com/global/docs/maya2014/en_us/images/comp_PointOrient.png))

c. ***Orient Constraint***

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

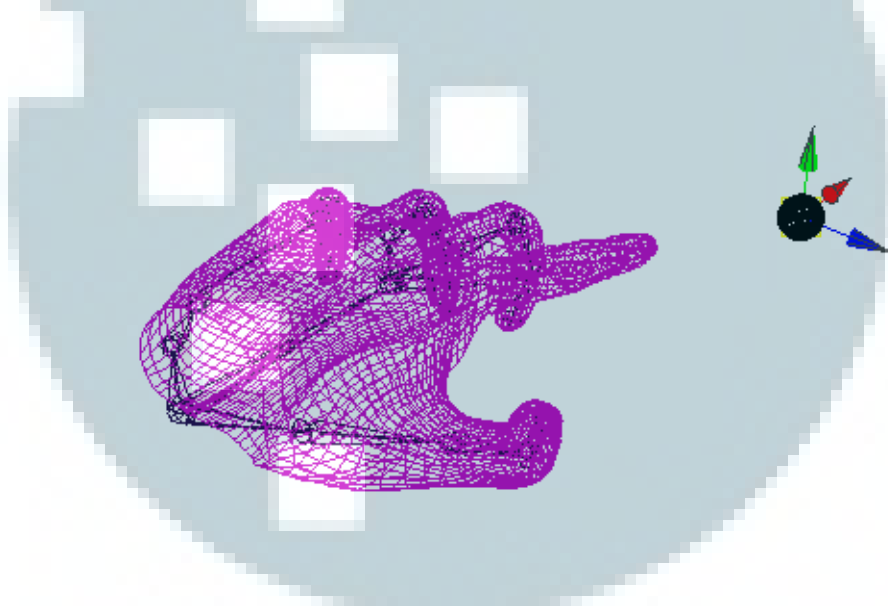


Gambar 2.4. Objek dengan *Orient Constraint*  
 ([http://download.autodesk.com/global/docs/maya2014/en\\_us/images/GUID-139D266D-FD4A-4358-BDA5-49F842A4DD83-low.png](http://download.autodesk.com/global/docs/maya2014/en_us/images/GUID-139D266D-FD4A-4358-BDA5-49F842A4DD83-low.png))



#### d. *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 penggerakannya. Biasanya digunakan untuk membuat mata mengikuti gerakan suatu objek.

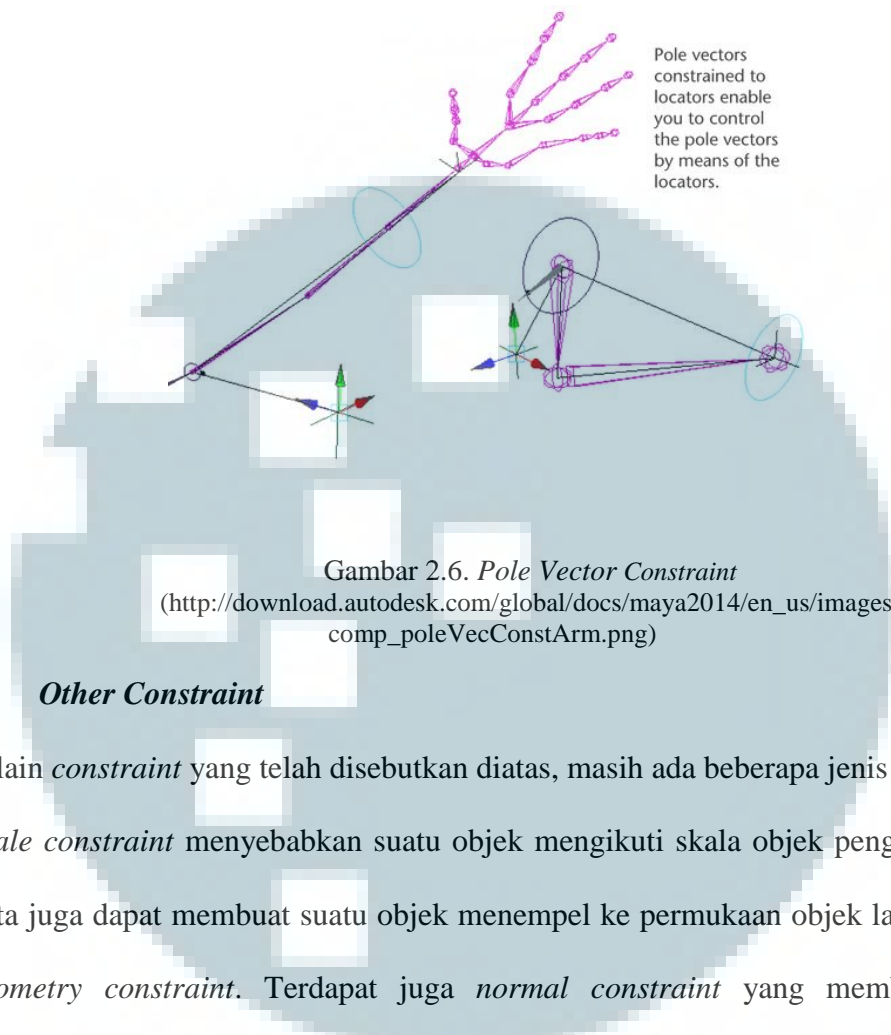


Gambar 2.5. *Aim Constraint*

([http://download.autodesk.com/global/docs/maya2014/en\\_us/images/GUID-3DA6BF4B-25CC-4356-8C18-8C2FBB063E23-low.png](http://download.autodesk.com/global/docs/maya2014/en_us/images/GUID-3DA6BF4B-25CC-4356-8C18-8C2FBB063E23-low.png))

#### e. *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.



Gambar 2.6. *Pole Vector Constraint*  
 ([http://download.autodesk.com/global/docs/maya2014/en\\_us/images/comp\\_poleVecConstArm.png](http://download.autodesk.com/global/docs/maya2014/en_us/images/comp_poleVecConstArm.png))

f. ***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 *geometry constraint*. 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 nilai arahnya diambil dari satu *poly* atau *vertex*.

2.2.2. **Jenis Rigging**

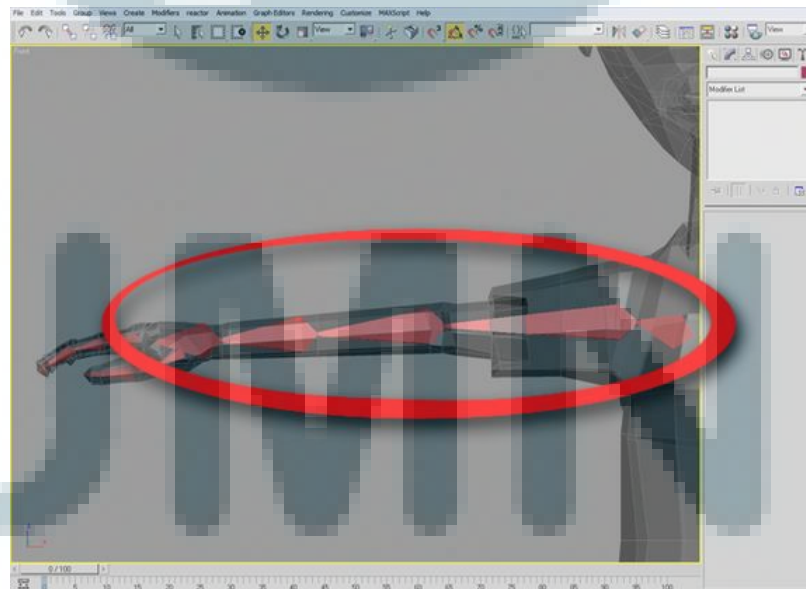
1. ***Parent and Child***

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. *Bones and Joints*

*Joints*, atau lebih tepatnya *joint deformer*, 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*.



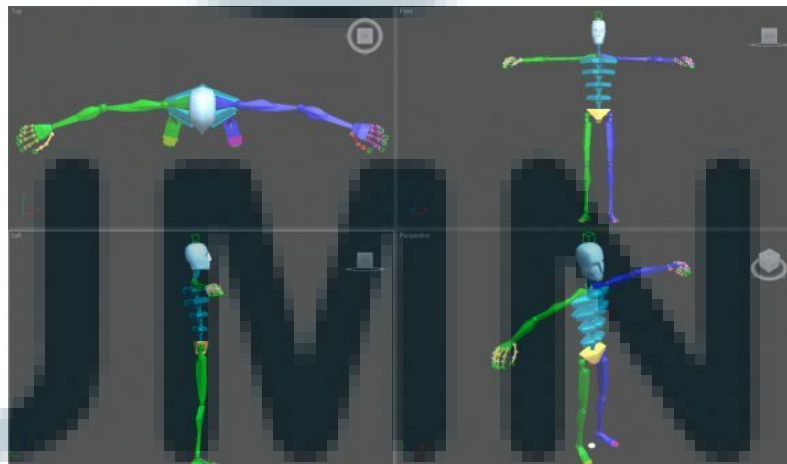
Gambar 2.7. *Bones*

([http://www.creativecrash.com/tutorialimages/318/tutorial\\_04.jpg](http://www.creativecrash.com/tutorialimages/318/tutorial_04.jpg))

### 3. *Skeletons*

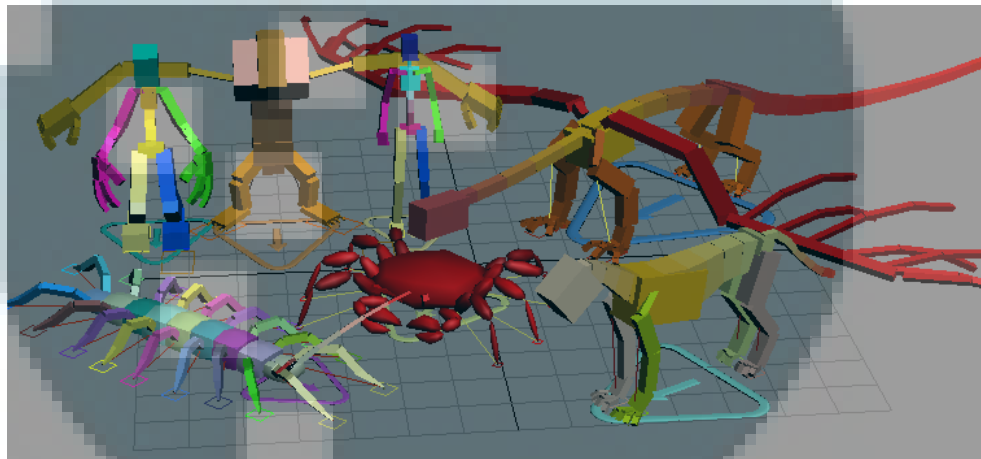
Dengan *joints*, kita dapat membuat hierarki *joints* yang kompleks seperti struktur rangka yang terdiri dari tangan, kaki, kepala, dan badan. Hierarki seperti ini disebut *skeleton*, atau *skeleton rig*. Hal ini biasanya dibutuhkan dalam model yang rumit seperti robot, manusia atau binatang. Beberapa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

*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). IK dalam *biped* juga telah diatur, misalnya dalam struktur tangan dan kaki.



Gambar 2.8. *Biped*  
(<http://www.sharecg.com/images/medium/60807.jpg>)

*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.



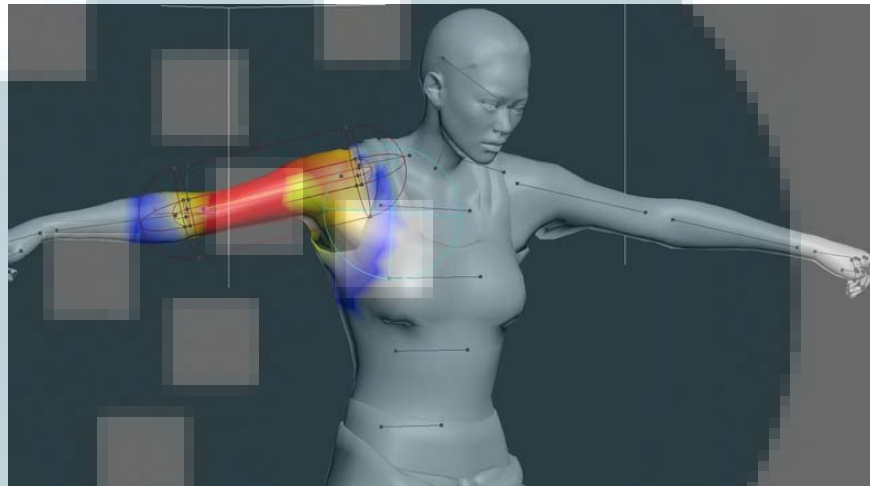
Gambar 2.9. Contoh *rig* yang tersedia dalam *CAT*  
([http://download.autodesk.com/esd/3dsmax/cat-help-2010/images/MED/CAT/English/intro\\_many\\_rigs.png](http://download.autodesk.com/esd/3dsmax/cat-help-2010/images/MED/CAT/English/intro_many_rigs.png))

### 2.3. *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 2.10. *Weight value*

([http://4.bp.blogspot.com/-](http://4.bp.blogspot.com/-QoNyMQcdJxU/T52KUEo5a_I/AAAAAAAAAPE/1zPA2OkuMCw/s1600/Skinning-weights.jpg)

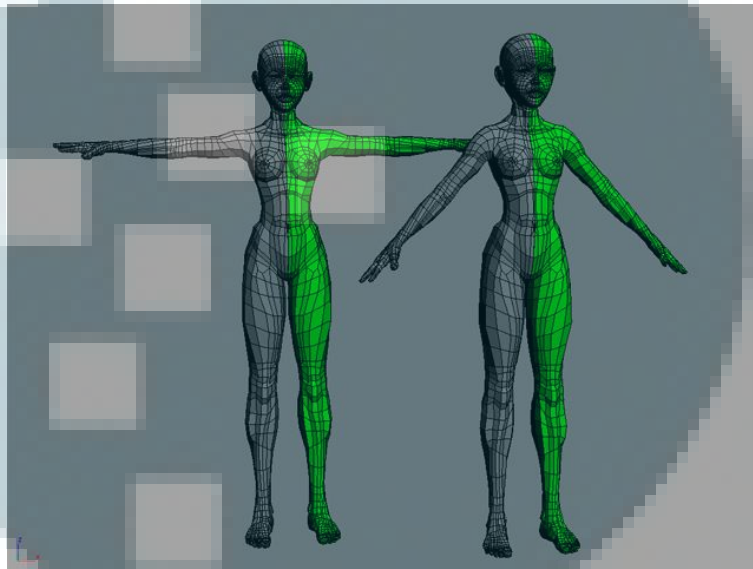
[QoNyMQcdJxU/T52KUEo5a\\_I/AAAAAAAAAPE/1zPA2OkuMCw/s1600/Skinning-weights.jpg](http://4.bp.blogspot.com/-QoNyMQcdJxU/T52KUEo5a_I/AAAAAAAAAPE/1zPA2OkuMCw/s1600/Skinning-weights.jpg))

#### 2.4. Pose dalam *Rigging*

Dalam membuat model karakter untuk di *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*.

*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. Pose ini memudahkan proses *rigging* karena topologi model pada pose ini biasanya relatif lebih benar.



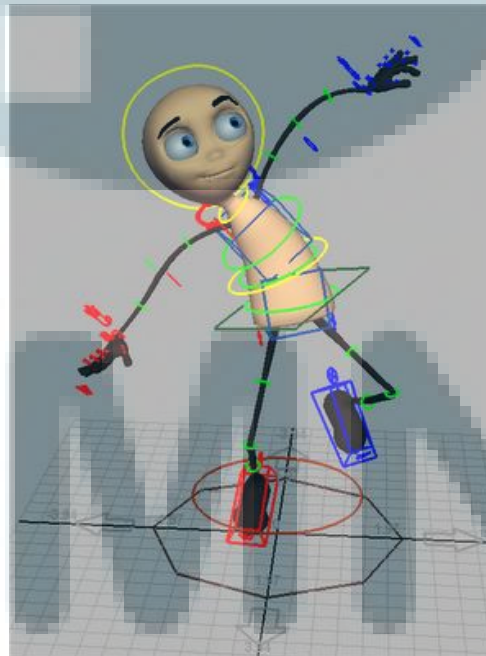
Gambar 2.11. *T-pose* standard dan *relaxed T-pose*  
([http://www.3dm3.com/tutorials/female\\_body/step35.jpg](http://www.3dm3.com/tutorials/female_body/step35.jpg))

## 2.5. *Controller*

Allen & Murdock (2008) mengatakan bahwa sistem kontrol adalah salah satu bagian penting dalam pembuatan *rig*. Ketika sebuah *rig* terdiri dari banyak bagian kontrol yang kompleks, biasanya dibuat suatu tampilan sistem kontrol yang memudahkan para animator menganimasikan *rig* tersebut. Sangatlah penting untuk membuat *custom controller* yang mudah dimengerti.

Menurut Lee (2001), *Wire Parameter* adalah salah satu *tool* 3ds Max yang dapat digunakan untuk membuat *custom controller*. *Wire Parameter* memungkinkan pengguna untuk *wiring* atau menghubungkan, nilai dari *parameter* suatu objek dengan *parameter* objek lain. *Wire Parameter* dapat menghasilkan koneksi satu arah maupun dua arah, sehingga memungkinkan untuk membuat suatu hubungan yang kompleks antar *parameter*.

*Reaction Manager* adalah salah satu *tool* lain dalam 3ds Max untuk membuat *custom controller*. *Reaction Manager* memungkinkan suatu objek mempengaruhi keadaan objek lain, dengan menggunakan hubungan *master*, *slave*, dan *state* (Murdock, 2011). Satu *master* dapat mengendalikan beberapa parameter *slave*.



Gambar 2.12. Contoh *custom controller*

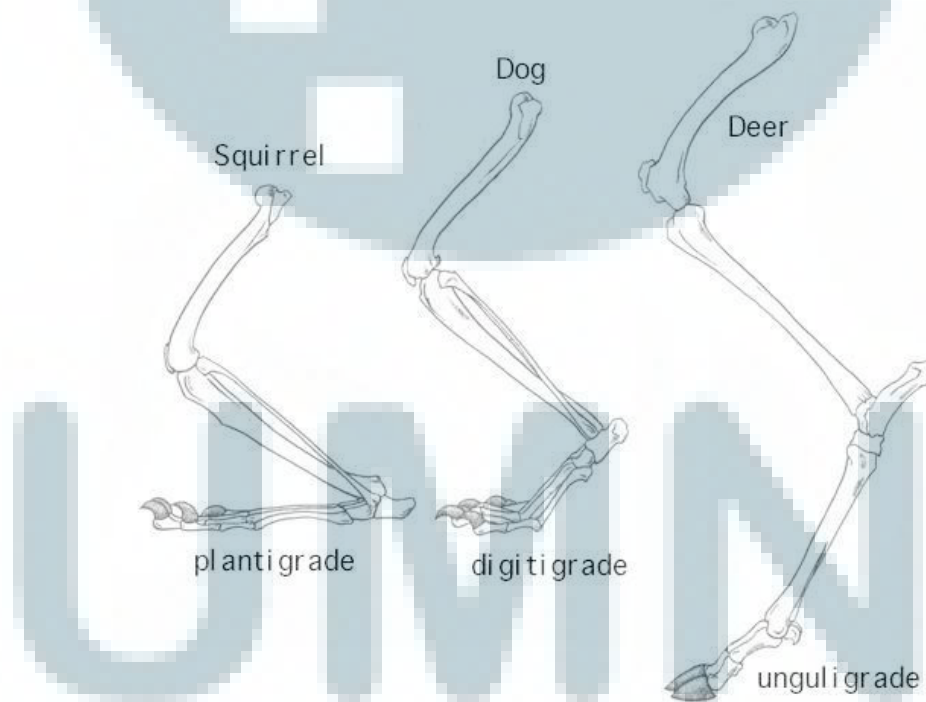
([http://1.bp.blogspot.com/-](http://1.bp.blogspot.com/-ImAbguaG5NM/UAt4Zd79gpI/AAAAAAAAASpE/9FWW76ibYU4/s400/rigging_tips_004.png)

[ImAbguaG5NM/UAt4Zd79gpI/AAAAAAAAASpE/9FWW76ibYU4/s400/rigging\\_tips\\_004.png](http://1.bp.blogspot.com/-ImAbguaG5NM/UAt4Zd79gpI/AAAAAAAAASpE/9FWW76ibYU4/s400/rigging_tips_004.png))



## 2.6. Anatomi dan *Rigging*

Menurut Flaxman (2008), desain sebuah *rigging* didasarkan dari anatomi dan gerakan karakter. Sebelum memulai *rigging*, bentuk dan gerakan harus dipelajari terlebih dahulu. Pengetahuan tentang anatomi termasuk hal yang penting dalam membuat *rigging* untuk sebuah karakter. Segala hal dalam *rigging*, mulai dari penempatan *joint* sampai *weight value*, perlu mempertimbangkan anatomi. Struktur sistem gerak termasuk dalam klasifikasi anatomi. Menurut Searfoss (1995), hewan bertulang belakang memiliki 3 tipe *locomotion* atau sistem gerak, yang masing-masing berhubungan dengan bentuk dan daerah kaki yang bersentuhan dengan tanah, yaitu *unguligrade*, *digitigrade* dan *plantigrade*.



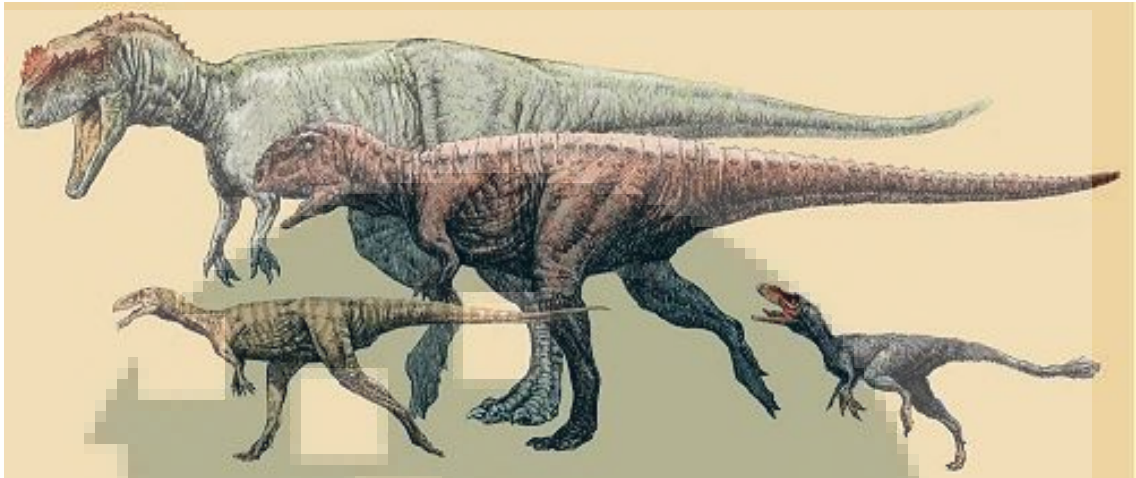
Gambar 2.13. *Locomotion* dalam hewan bertulang belakang  
(<http://people.eku.edu/ritchisong/plantigrade.jpg>)

### 2.6.1. Anatomi Dinosaurus

Dinosaurus adalah salah satu contoh hewan *vertebrata*, atau hewan bertulang belakang. Untuk pembagian tulang belakang atau *vertebrae* terbagi menjadi *cervical*, *dorsal*, *sacral* dan *caudal*. Menurut Brusatte (2012), sumber informasi paling penting dalam postur tubuh dan sistem gerak dinosaurus berasal dari komponen fosil makhluk tersebut, yaitu tulang dinosaurus itu sendiri. Jika suatu kerangka cukup lengkap, dapat ditentukan apakah suatu dinosaurus termasuk tipe *bipedal* atau *quadrupedal*. Cara suatu tulang berartikulasi dapat memberikan informasi jajaran gerakan bagian-bagian tertentu dari suatu rangka.

Menurut Martin (2009), dasar dari anatomi dinosaurus adalah tulang panggul. Dinosaurus terbagi dalam 2 *sub-clade*, yaitu *Saurischia* dan *Ornithischia* berdasarkan perbedaan struktur tulang panggul. *Saurischia* terbagi lagi ke dalam 2 kelas yaitu *theropoda* dan *sauropodomorpha*. Sebagian besar dinosaurus memiliki struktur kaki *digitigrade*, dimana tulang metatarsal tidak menyentuh tanah melainkan ruas jarinya.

UMMN



Gambar 2.14. Dinosaurus kelas *theropoda*  
([http://www.dinosaurcentral.com/images/panopies/Theropod\\_panoply.jpg](http://www.dinosaurcentral.com/images/panopies/Theropod_panoply.jpg))

Menurut Brusatte (2012), semua dinosaurus *theropod* memiliki struktur tubuh yang hampir sama, yaitu:

1. *Bipedal* atau berjalan dengan 2 kaki
2. Berjalan dengan jari kaki mereka, dengan 3 jari yang menyentuh tanah

Menurut Kaiser (2007), tubuh dinosaurus yang berjalan dengan dua kaki atau *bipedal*, berfungsi sebagai modul daya penggerak tunggal, karena *forelimb* atau organ lambai depan mereka meninggalkan peran dalam sistem gerak dan digantikan dengan fungsi menangkap mangsa. Ketika seekor dinosaurus berjalan, ia harus mengalihkan berat badan ke kaki dan akan tergoyang dari sisi ke sisi sambil berjalan ke depan. Untuk menjaga keseimbangannya, ia harus melakukan gerakan kompensasi dengan kepala dan ekornya.

## 2.7. Sendi

Menurut Premkumar (2004), persendian diklasifikasikan sesuai struktur dan fungsi gerakannya. Secara fungsi, sendi terbagi menjadi 3, yaitu:

- Synarthrosis
- Amphiarthrosis
- Diarthrosis

Menurut Hamill (2006), sebuah potensi gerakan dari suatu bagian tubuh ditentukan oleh struktur dan fungsi sendi *synovial* atau *diarthrosis*. Sendi diarthrosis atau *diarthrodial joint* adalah sendi yang bebas digerakkan. Sendi diarthrodial memungkinkan artikulasi yang bergesekan rendah. Menurut Alcamo (2004), ada 6 tipe sendi diarthrosis, yaitu *gliding*, *hinge*, *pivot*, *ellipsoid*, *saddle* dan *ball-and-socket joint*.

### 1. *Gliding Joint*


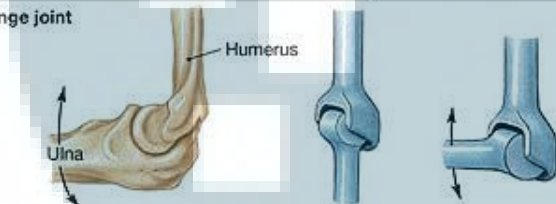
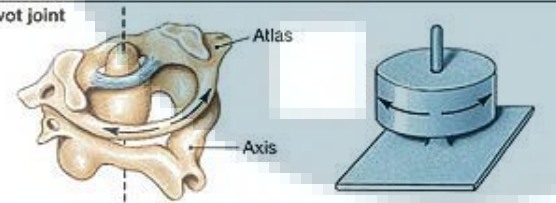
*Gliding joint* atau sendi luncur adalah sendi yang memungkinkan gerakan non-aksial atau rotasi pada satu bidang datar. Sendi ini disebut juga *plane joint*. Contohnya adalah tulang pergelangan kaki.

### 2. *Hinge Joint*

*Hinge joint* atau sendi engsel adalah sendi yang hanya memungkinkan gerakan satu poros pada satu bidang. Dengan kata lain, gerakan secara satu arah, seperti pintu yang memiliki engsel. Contohnya adalah siku dan lutut

### 3. *Pivot Joint*

*Pivot joint* atau sendi putar adalah sendi yang hanya memungkinkan gerakan rotasi satu poros pada satu bidang. Contohnya adalah hubungan tulang kepala dengan tulang belakang.

Types of Synovial Joints		Movement	Examples
<b>Gliding joint</b> 		Slight nonaxial or multiaxial	<ul style="list-style-type: none"> <li>• Acromioclavicular and clavicularosternal joints</li> <li>• Intercarpal and intertarsal joints</li> <li>• Vertebrocostal joints</li> <li>• Sacroiliac joints</li> </ul>
<b>Hinge joint</b> 		Monaxial	<ul style="list-style-type: none"> <li>• Elbow joint</li> <li>• Knee joint</li> <li>• Ankle joint</li> <li>• Interphalangeal joint</li> </ul>
<b>Pivot joint</b> 		Monaxial (rotation)	<ul style="list-style-type: none"> <li>• Atlas/axis</li> <li>• Proximal radioulnar joint</li> </ul>

Copyright © 2004 Pearson Education, Inc., publishing as Benjamin Cummings.

Gambar 2.15. Gliding, pivot dan hinge joint  
[http://www.as.miami.edu/chemistry/2008-1-MDC/2085/Chap9-New/Chapter%209\\_files/image009.jpg](http://www.as.miami.edu/chemistry/2008-1-MDC/2085/Chap9-New/Chapter%209_files/image009.jpg)

### 4. *Ball-and-socket Joint*

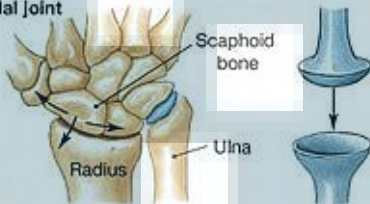
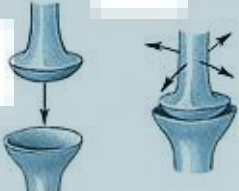
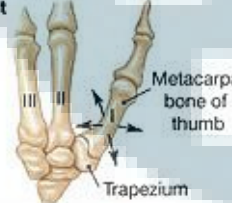
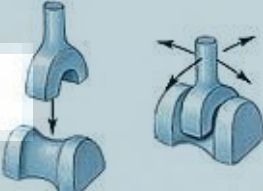
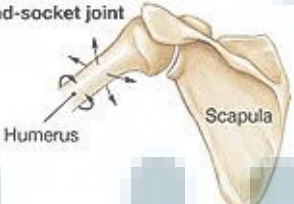

*Ball-and-socket joint* atau sendi peluru adalah sendi yang memungkinkan gerakan dalam tiga bidang, atau segala arah. Contohnya adalah bahu.

### 5. *Ellipsoid Joint*

*Ellipsoid joint* adalah sendi yang hanya memungkinkan gerakan pada dua bidang dengan dua sumbu

## 6. Saddle Joint

Saddle *joint* atau sendi pelana adalah sendi yang memungkinkan gerakan pada dua bidang datar. Secara fungsi sendi ini mirip seperti sendi *ellipsoid* namun sedikit lebih bebas. Contohnya adalah hubungan jempol dengan telapak tangan.

Types of Synovial Joints		Movement	Examples
<b>Ellipsoidal joint</b> 		Biaxial	<ul style="list-style-type: none"> <li>• Radiocarpal joint</li> <li>• Metacarpophalangeal joints 2-5</li> <li>• Metatarsophalangeal joints</li> </ul>
<b>Saddle joint</b> 		Biaxial	<ul style="list-style-type: none"> <li>• First carpometacarpal joint</li> </ul>
<b>Ball-and-socket joint</b> 		Triaxial	<ul style="list-style-type: none"> <li>• Shoulder joint</li> <li>• Hip joint</li> </ul>

Copyright © 2004 Pearson Education, Inc., publishing as Benjamin Cummings.

Gambar 2.16. Ellipsoid, Saddle dan Ball-and socket joint  
 ([http://www.as.miami.edu/chemistry/2008-1-MDC/2085/Chap9-New/Chapter%209\\_files/image010.jpg](http://www.as.miami.edu/chemistry/2008-1-MDC/2085/Chap9-New/Chapter%209_files/image010.jpg))