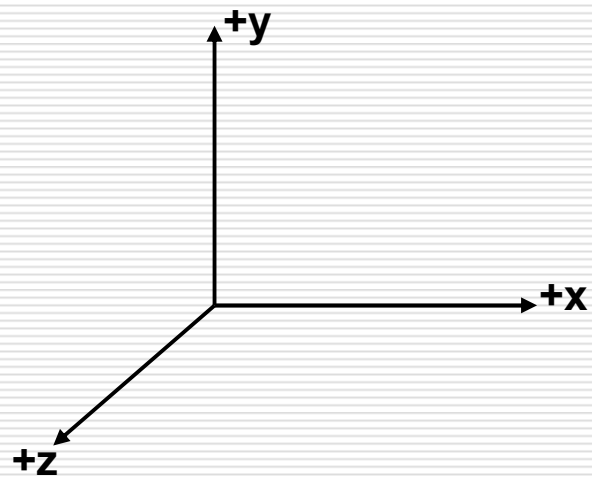


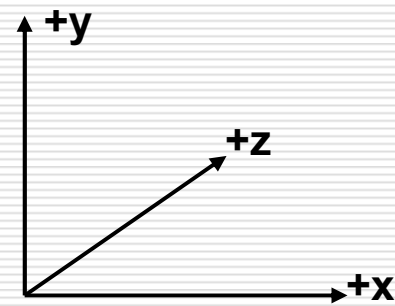
Pendahuluan

- Grafika Komputer dalam aplikasinya terbagi menjadi 2 :
 - Grafika 2D
 - Grafika 3D
 - Aplikasi 2D banyak dipakai dalam pembuatan grafik, peta, kreasi 2D yang banyak membantu pemakai dalam membuat visualisasi.
 - Grafika 2D memiliki kekurangan, yaitu : ketidakmampuannya untuk merepresentasikan objek 3D. Kekurangan ini sangat dirasakan terutama dalam bidang desain, dimana kebanyakan desainer membuat barang yang ada dalam dunia nyata yang berdimensi 3.
 - Grafika 3D memiliki kemampuan untuk membuat visualisasi dari sebuah benda yang nyata yang dapat dilihat dari berbagai sudut pandang. Hal inilah yang membuat grafika 3D banyak dipakai terutama dalam bidang desain dari sebuah produk.
-

Sistem Koordinat 3D



Right-handed



Left-handed

Primitif 3D

- Dalam dunia 3D terdapat beberapa primitif seperti :
 - Titik (*point*)
 - Garis (*line*)
 - Bidang/Permukaan (*plane/surface*)
 - Bola (*sphere*)
 - Kubus(*cube*)
 - Silinder (*cylinder*)
 - Kerucut (*cone*)
 - Cincin (*torus*)
 - dll
-

Primitif 3D

□ Titik

- Posisi sebuah titik dalam grafika 3D diekspresikan dengan (x,y,z)

□ Garis

- Sebuah garis dibentuk dengan mendeskripsikan dua buah titik, yaitu (x_1,y_1,z_1) dan (x_2,y_2,z_2) yang sebagai ujung dari sebuah garis.
- Sebuah garis dalam grafika 3D dapat diekspresikan dengan sepasang persamaan, yaitu :

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

$$\frac{z - z_1}{x - x_1} = \frac{z_2 - z_1}{x_2 - x_1}$$

$$\frac{z - z_1}{x - x_1} = \frac{z_2 - z_1}{x_2 - x_1}$$

$$\frac{z - z_1}{x - x_1} = \frac{z_2 - z_1}{x_2 - x_1}$$

Primitif 3D

□ Bidang

- Pada grafika 3D, terdapat sebuah geometri yang sangat penting, yaitu bidang datar (*plane*). Sebuah bidang datar pada grafika 3D dispesifikasikan dengan sebuah persamaan, yaitu :

$$Ax + By + Cz + D = 0$$

Representasi Object 3D

- Untuk merepresentasikan object 3D :
 - Persamaan Geometri
 - Constructive Solid Geometry (CSG)
 - Kurva & Permukaan Bezier
 - Lathe Object
 - Fractal
-

Representasi Object 3D

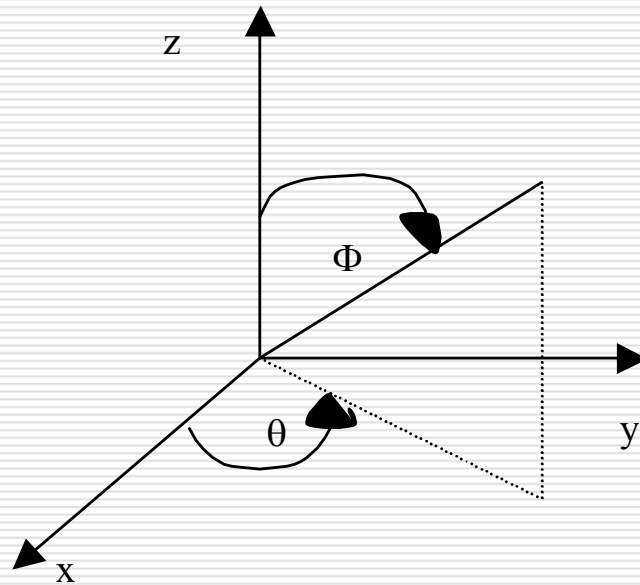
□ Dengan Persamaan Geometri

- Suatu object 3D dapat direpresentasikan langsung dengan menggunakan persamaan geometri dari object tersebut.
- Misalkan : untuk membangun sebuah bola, maka bisa dengan menggunakan rumus :

$$X^2 + Y^2 + Z^2 = R^2$$

atau dengan rumus :

Representasi Object 3D



- $x = r \cdot \sin\Phi \cdot \sin\theta$; $0 \leq \Phi \leq 2\pi$
 - $y = r \cdot \sin\Phi \cdot \cos\theta$; $-\pi \leq \theta \leq \pi$
 - $z = r \cdot \cos\Phi$
-

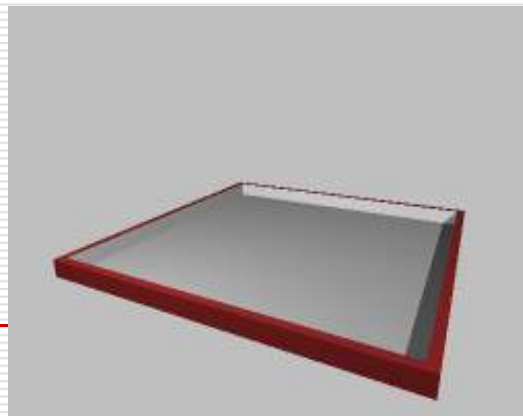
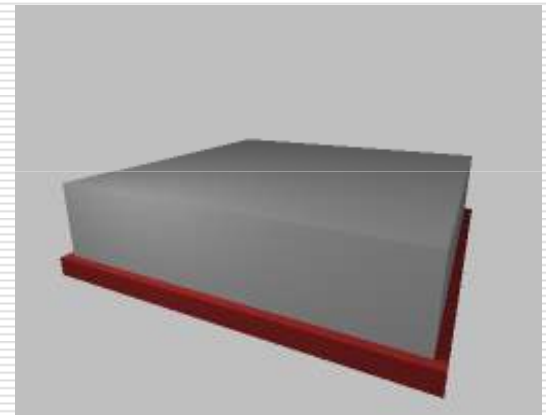
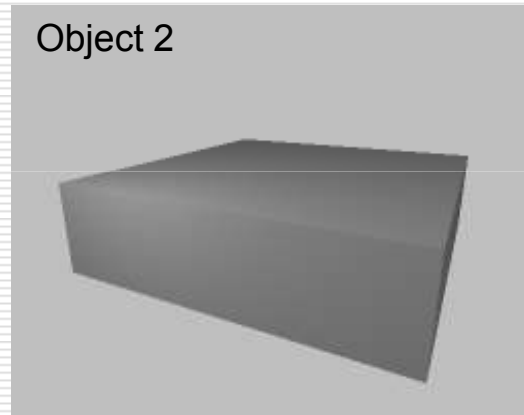
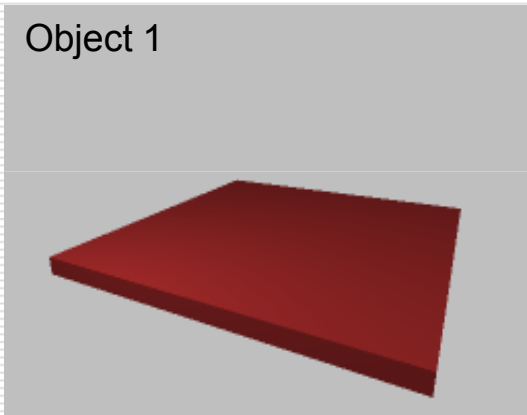
Representasi Object 3D

- Dengan Constructive Solid Geometry (CSG)
 - CSG adalah suatu cara membentuk object dengan jalan menggabungkan atau memotong(mengurangi) dari beberapa object primitif 3D.
 - CSG dalam POV-Ray melibatkan :
 - difference
 - union
 - intersect
-

Representasi Object 3D

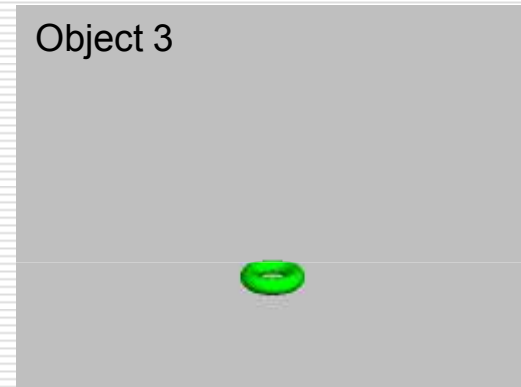
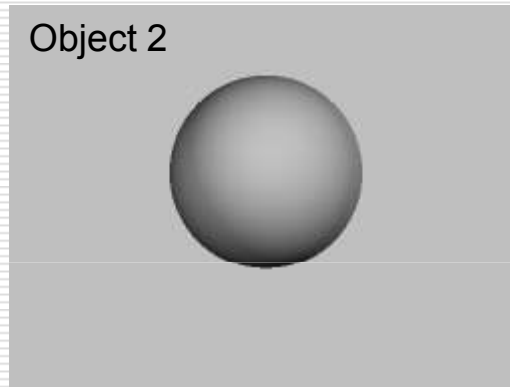
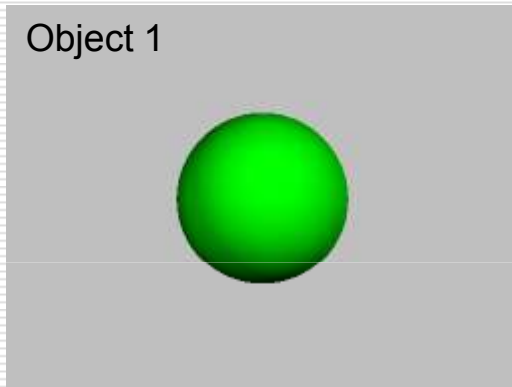
□ Dengan CSG – cont.

■ Contoh 1:

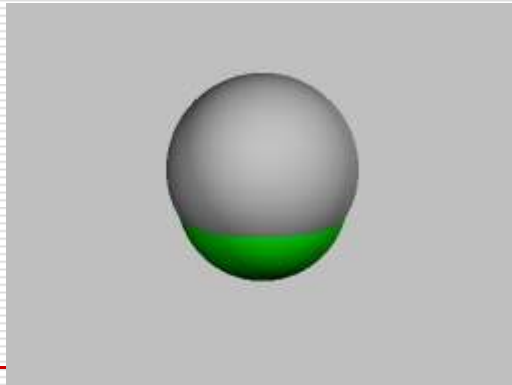


Representasi Object 3D

□ Dengan CSG – cont.



Object 5 = Object 1 – Object 2



Object 6 = Object 5 + Object 3



Representasi Object 3D

□ Kurva & Permukaan Bezier

$$x(t) = \sum_{i=0}^N x_i \cdot B_{i,N}(t)$$

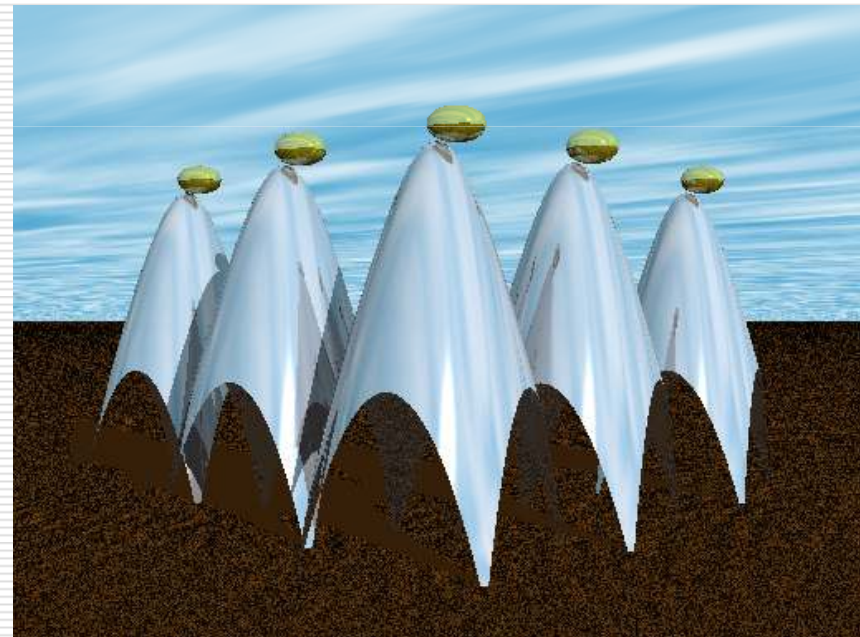
$$y(t) = \sum_{i=0}^N y_i \cdot B_{i,N}(t)$$

$$z(t) = \sum_{i=0}^N z_i \cdot B_{i,N}(t)$$

dimana :

$$B_{i,N} = C(N, i) \cdot (1-t)^{N-i} t^i$$

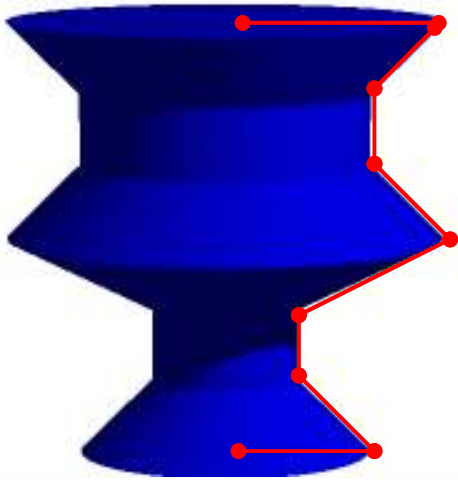
$$C(N, i) = \frac{N!}{i! \cdot (N-i)!}$$



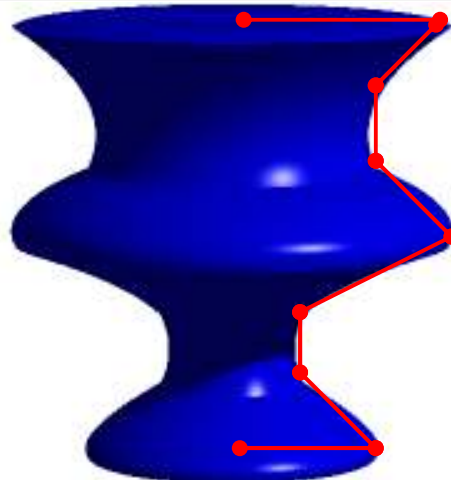
Representasi Object 3D

□ Lathe Object

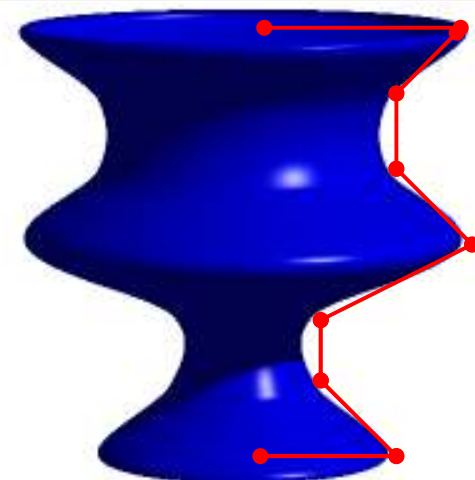
Linear Spline



Quadratic Spline



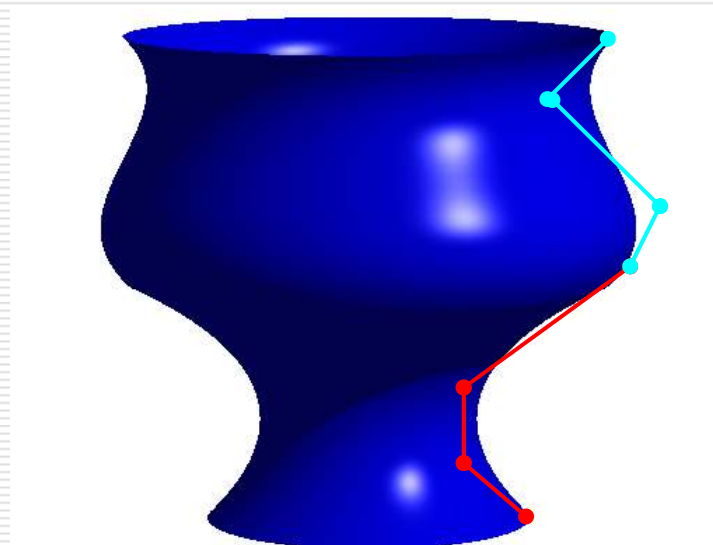
Cubic Spline



Representasi Object 3D

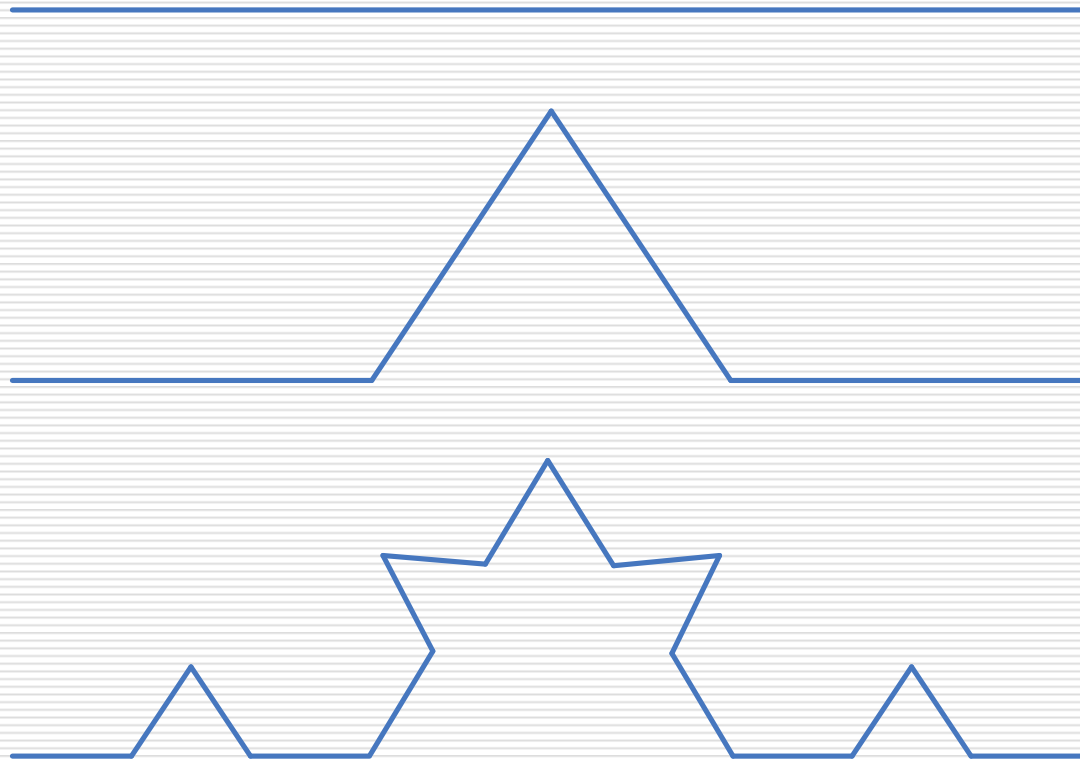
□ Lathe Object - cont

Bezier Spline



Representasi Object 3D

- Dengan Fractal



Rendering

- **Rendering** → Proses untuk menghasilkan sebuah citra 2D dari data 3D.
 - Proses ini bertujuan untuk memberikan visualisasi pada user mengenai data 3D tersebut melalui monitor atau pencetak yang hanya dapat menampilkan data 2D
 - Metode rendering yang paling sederhana dalam grafika 3D :
 - ***Wireframe rendering***
 - ***Hidden Line Rendering***
 - ***Shaded Rendering***
-

Rendering

Wireframe rendering

- Yaitu Objek 3D dideskripsikan sebagai objek tanpa permukaan.
 - Pada wireframe rendering, sebuah objek dibentuk hanya terlihat garis-garis yang menggambarkan sisi-sisi edges dari sebuah objek.
 - Metode ini dapat dilakukan oleh sebuah komputer dengan sangat cepat, hanya kelemahannya adalah tidak adanya permukaan, sehingga sebuah objek terlihat transparent. Sehingga sering terjadi kesalahpahaman antara sisi depan dan sisi belakang dari sebuah objek.
-

Rendering

Hidden Line Rendering

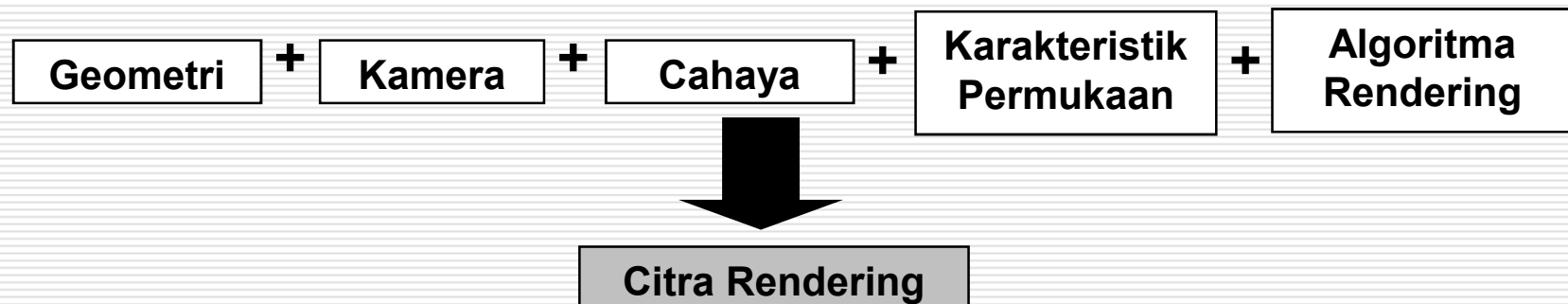
- Metode ini menggunakan fakta bahwa dalam sebuah objek, terdapat permukaan yang tidak terlihat atau permukaan yang tertutup oleh permukaan lainnya.
 - Dengan metode ini, sebuah objek masih direpresentasikan dengan garis-garis yang mewakili sisi dari objek, tapi beberapa garis tidak terlihat karena adanya permukaan yang menghalanginya. Metode ini lebih lambat dari wireframe rendering, tapi masih dikatakan relatif cepat. Kelemahan metode ini adalah tidak terlihatnya karakteristik permukaan dari objek tersebut, seperti warna, kilauan (shininess), tekstur, pencahayaan, dll.
-

Rendering

Shaded Rendering

- ❑ Pada metode ini, komputer diharuskan untuk melakukan berbagai perhitungan baik pencahayaan, karakteristik permukaan, shadow casting, dll.
 - ❑ Metode ini menghasilkan citra yang sangat realistik, tetapi kelemahannya adalah lama waktu rendering yang dibutuhkan.
-

Rendering



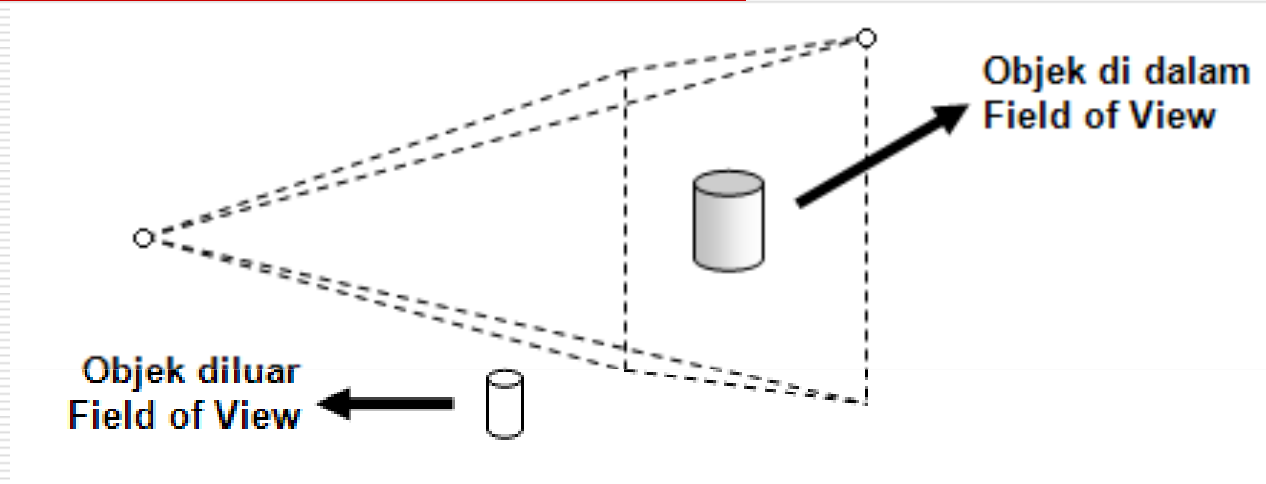
- Secara umum, proses untuk menghasilkan *rendering* dua dimensi dari objek-objek 3D melibatkan 5 komponen utama :
 - Geometri
 - Kamera
 - Cahaya
 - Karakteristik Permukaan
 - Algoritma Rendering
-

Kamera

- Dalam grafika 3D, sudut pandang (*point of view*) adalah bagian dari kamera. Kamera dalam grafika 3D biasanya tidak didefinisikan secara fisik, namun hanya untuk menentukan sudut pandang kita pada sebuah *world*, sehingga sering disebut *virtual camera*.

 - Sebuah kamera dipengaruhi oleh dua buah faktor penting.
 - **Faktor pertama** adalah lokasi (*camera location*). Lokasi sebuah kamera ditentukan dengan sebuah titik (x,y,z) .
 - **Faktor kedua** adalah arah pandang kamera. Arah pandang kamera ditunjukkan dengan sebuah sistem yang disebut sistem koordinat acuan pandang atau sistem (U,N,V) . Arah pandang kamera sangat penting dalam membuat sebuah citra, karena letak dan arah pandang kamera menentukan apa yang terlihat oleh sebuah kamera. Penentuan apa yang dilihat oleh kamera biasanya ditentukan dengan sebuah titik (x,y,z) yang disebut *camera interest*.
-

Kamera



- Pada kamera, dikenal *field of view* yaitu daerah yang terlihat oleh sebuah kamera. Field of view pada grafika 3D berbentuk piramid, karena layar monitor sebuah komputer berbentuk segiempat. Objek-objek yang berada dalam field of view ini akan terlihat dari layar monitor, sedang objek-objek yang berada di luar field of view ini tidak terlihat pada layar monitor. Field of view ini sangat penting dalam pemilihan objek yang akan diproses dalam rendering. Objek-objek diluar field of view biasanya tidak akan diperhitungkan, sehingga perhitungan dalam proses rendering, tidak perlu dilakukan pada seluruh objek.
-

Cahaya

- Sumber cahaya pada grafika 3D merupakan sebuah objek yang penting, karena dengan cahaya ini sebuah *world* dapat terlihat dan dapat dilakukan proses *rendering*. Sumber cahaya ini juga membuat sebuah *world* menjadi lebih realistis dengan adanya bayangan dari objek-objek 3D yang ada.

 - Sebuah sumber cahaya memiliki jenis. Pada grafika 3D dikenal beberapa macam sumber cahaya, yaitu :
 - ***Point light***
 - ***Spotlight***
 - ***Ambient light***
 - ***Area light***
 - ***Directional light***
 - ***Parallel point***
-

Cahaya

- ***point light***
 - memancar ke segala arah, namun intensitas cahaya yang diterima objek bergantung dari posisi sumber cahaya. Tipe ini mirip seperti lampu pijar dalam dunia nyata.
 - ***spotlight***
 - memancarkan cahaya ke daerah tertentu dalam bentuk kerucut. Sumber cahaya terletak pada puncak kerucut. Hanya objek-objek yang terletak pada daerah kerucut tersebut yang akan nampak.
 - ***ambient light***
 - cahaya latar/alam. Cahaya ini diterima dengan intensitas yang sama oleh setiap permukaan pada benda. Cahaya latar tersebut dimodelkan mengikuti apa yang terjadi di alam, diaman dalam keadaan tanpa sumber cahaya sekalipun, benda masih dapat dilihat.
 - ***area light***
 - ***directional light***
 - memancarkan cahaya dengan intensitas sama ke suatu arah tertentu. Letak tidak mempengaruhi intensitas cahayanya. Tipe ini dapat menimbulkan efek seolah-olah sumber cahaya berada sangat jauh dari objek
 - ***parallel point***
 - sama dengan directional, hanya pencahayaan ini memiliki arah dan posisi.
-

Cahaya

- Model dari pencahayaan, dipakai untuk menghitung intensitas dari cahaya yang terlihat dari setiap posisi pada setiap permukaan benda yang terlihat oleh kamera. Ketika melihat sebuah benda, terlihat cahaya yang dipantulkan dari permukaan benda, dimana cahaya ini merupakan integrasi dari sumber-sumber cahaya serta cahaya yang berasal dari pantulan cahaya permukaan-permukaan yang lain. Karena itu benda-benda yang tidak langsung menerima cahaya dari sumber cahaya, masih mungkin terlihat bila menerima cahaya pantulan yang cukup dari benda didekatnya.
 - Model sederhana dari sumber cahaya adalah sebuah titik sumber, dimana dari titik ini cahaya dipancarkan. Perhitungan pencahayaan bergantung pada sifat dari permukaan yang terkena cahaya, kondisi dari cahaya latar serta spesifikasi sumber cahaya.
-

Cahaya

- Semua sumber cahaya dimodelkan sebagai sumber titik yang dispesifikasikan dengan :
 - **Lokasi** → Lokasi (x,y,z) dari sebuah sumber cahaya akan menentukan pengaruhnya terhadap sebuah objek.
 - **Intensitas** → Intensitas cahaya menyatakan kekuatan cahaya yang dipancarkan oleh sebuah sumber cahaya. Parameter ini merupakan angka, yang biasanya makin besar nilainya, makin terang sumber cahaya tersebut.
 - **Warna** → Warna cahaya dari sumber ini akan mempengaruhi warna dari sebuah objek, jadi selain warna objek tersebut warna cahaya yang jatuh pada objek tersebut akan mempengaruhi warna pada *rendering*. Warna cahaya ini biasanya terdiri dari 3 warna dasar grafika komputer, yaitu: merah, hijau, biru atau lebih dikenal dengan RGB.
-

Karakteristik Permukaan

- Karakteristik permukaan dari sebuah objek adalah sifat dari permukaan sebuah objek.
 - Karakteristik permukaan ini meliputi
 - Warna
 - Tekstur
 - Sifat permukaan, seperti kekasaran (*roughness*), refleksifitas, *diffuseness* (jumlah cahaya yang dipantulkan oleh objek), transparansi, dan lain-lain.
 - Parameter Warna dalam karakteristik permukaan direpresentasikan dengan tiga warna dasar, yaitu RGB. Saat *rendering*, warna pada sebuah objek tergantung dari warna dalam karakteristik permukaan dan warna cahaya yang mengenainya. Jadi citra hasil *rendering* mungkin akan memiliki warna yang sedikit berbeda dengan warna objek tersebut.
-

Karakteristik Permukaan

- ❑ **Parameter tekstur** direpresentasikan dengan sebuah nama file. File ini akan menjadi tekstur pada permukaan objek tersebut. Selain itu juga ada beberapa parameter dalam tekstur yang berguna untuk menentukan letak tekstur pada sebuah objek, sifat tekstur, perulangan tekstur, dan lain-lain.
 - ❑ **Sifat Permukaan**, seperti *diffuseness*, reflektivitas, dan lain-lain direpresentasikan dengan sebuah nilai. Nilai ini menentukan sifat dari parameter-parameter tersebut. Misalnya pada *roughness*, makin besar nilai parameternya, makin kasar objek tersebut.
-

Algoritma Rendering

- **Algoritma Rendering** adalah prosedur yang digunakan oleh suatu program untuk mengerjakan perhitungan untuk menghasilkan citra 2D dari data 3D.

 - Kebanyakan algoritma rendering yang ada saat ini menggunakan pendekatan yang disebut *scan-line rendering* → berarti program melihat dari setiap pixel, satu per satu, secara horizontal dan menghitung warna di pixel tersebut.

 - Saat ini dikenal 3 algoritma :
 - *Ray-Casting*
 - *Ray-Tracing*
 - *Radiosity*
-

Algoritma Rendering

Ray-Casting	Ray-Tracing	Radiosity
<ul style="list-style-type: none"> Menembakkan sinar untuk mengetahui warna dari cell 	<ul style="list-style-type: none"> Menelusuri sinar secara terbalik untuk mengetahui warna dari sebuah pixel 	<ul style="list-style-type: none"> Membagi bidang menjadi bidang yang lebih kecil untuk menentukan warna
<ul style="list-style-type: none"> Menggunakan metode sampling untuk menampilkan hasil 	<ul style="list-style-type: none"> Tidak menggunakan sampling 	<ul style="list-style-type: none"> Tidak menggunakan sampling
<ul style="list-style-type: none"> Biasanya dimodifikasi sesuai kebutuhan 	<ul style="list-style-type: none"> Tidak dapat dimodifikasi 	<ul style="list-style-type: none"> Tidak dapat dimodifikasi
<ul style="list-style-type: none"> Memiliki konstrain geometrik tertentu 	<ul style="list-style-type: none"> Tidak Memiliki konstrain geometrik tertentu 	<ul style="list-style-type: none"> Tidak Memiliki konstrain geometrik tertentu
<ul style="list-style-type: none"> Waktu rendering cepat 	<ul style="list-style-type: none"> Waktu rendering lambat 	<ul style="list-style-type: none"> Waktu rendering sangat lambat
<ul style="list-style-type: none"> Digunakan untuk visuali-sasi secara cepat 	<ul style="list-style-type: none"> Digunakan untuk visualiasi hasil akhir 	<ul style="list-style-type: none"> Digunakan untuk visualiasi hasil akhir
<ul style="list-style-type: none"> Dapat digunakan untuk out-door dan in-door scene 	<ul style="list-style-type: none"> Dapat digunakan untuk out-door dan in-door scene 	<ul style="list-style-type: none"> Biasanya digunakan untuk in-door scene
<ul style="list-style-type: none"> Hasil rendering kadang terlihat kotak-kotak 	<ul style="list-style-type: none"> Hasil rendering sangat realistik 	<ul style="list-style-type: none"> Hasil rendering sangat realistik