

BAB II

LANDASAN TEORI

2.1. Pemodelan Objek 3D.

Pemodelan objek 3D adalah suatu metode untuk menggambarkan keadaan suatu objek 3D dalam bentuk geometri dan topologi objek 3D sehingga dapat dianalisa dan diolah lebih lanjut. Objek dalam dunia 3D berbentuk polihedron, yang dimaksud polihedron adalah gambaran objek mempunyai ruang 3 dimensi (panjang, lebar, tinggi) dimana pada sisi-sisi permukaan dibentuk oleh poligon. Poligon dibentuk dari kumpulan *vertex*, yaitu titik koordinat yang terhubungkan oleh setiap 2 garis yang saling bertemu. Contoh polihedron adalah balok, piramida, tabung. Sedangkan contoh poligon adalah bujursangkar, kotak, segitiga. (Fox, 1995).

Tujuan dari pemodelan objek 3D adalah :

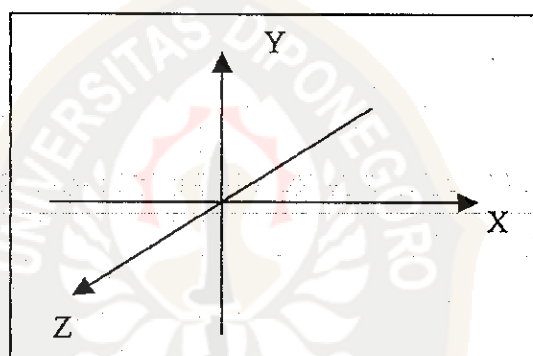
- a. Model objek dan informasi tentang lokasi pengamat akan bersama-sama digunakan untuk mensintesis citra dari objek yang dimaksud.
- b. Digunakan untuk memodifikasi dan menganalisa objek-objek yang akan ditampilkan.

Sehingga pada akhirnya dapat ditentukan koordinat titik yang membentuk objek tersebut. Informasi yang diperlukan dalam pemodelan objek 3D adalah geometri dan topologi objek. Geometri objek 3D berhubungan dengan ukuran, misalnya : lokasi titik atau ukuran objek. Topologi digunakan untuk menghubungkan titik-titik koordinat objek sehingga dapat membentuk suatu poligon, kemudian bagaimana

poligon yang terbentuk disusun untuk membentuk objek (polihedron) yang dimaksud. Sehingga diperlukan sistem koordinat untuk membentuk model objek. (Insap, 1996).

2.1.1. Sistem Koordinat Cartesian.

Sistem koordinat Cartesian yang digunakan dalam penulisan ini meliputi 3 sumbu koordinat (X,Y,Z). Sumbu X merupakan sumbu horizontal, sumbu Y sebagai sumbu vertikal, dan sumbu Z sebagai sumbu kedalaman, seperti Gambar 2.1. berikut :



Gambar 2.1. Sistem koordinat Cartesian 3D.

Objek benda akan terletak pada bidang XYZ, setiap objek tersusun dari titik-titik koordinat yang mempunyai koordinat (x, y, z) .

2.2. Proyeksi

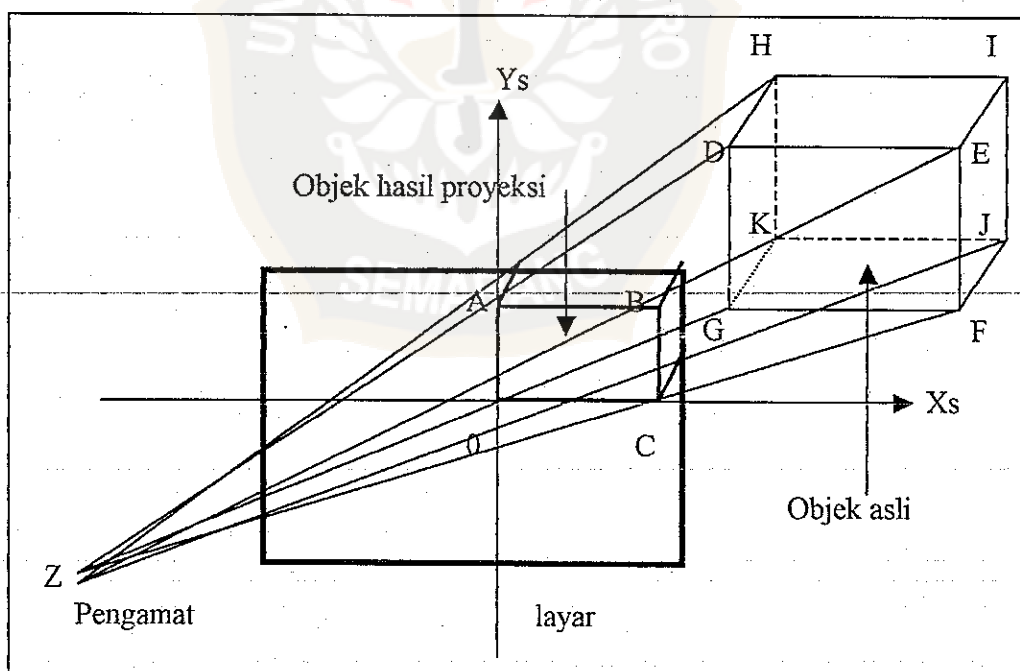
Suatu objek 3D apabila disinari sinar dari arah tertentu akan membentuk bayangannya pada suatu permukaan. Dengan menggunakan prinsip penyinaran tersebut maka dapat dihasilkan suatu proyeksi 2D. Langkah untuk menghasilkan objek 2D dari hasil proyeksi 3D adalah :

- a. Proyeksikan setiap titik dari objek 3D ke permukaan gambar 2D.

b. Gambar garis untuk menghubungkan sepasang titik.

2.2.1. Proyeksi Paralel.

Arah dari sinar diasumsikan tegak lurus terhadap suatu objek 3 dimensi (sejajar sumbu Z). Sehingga sinar jatuh pada suatu bidang 2 dimensi, misal bidang XY , sehingga akan timbul bayangan pada bidang XY . Bidang XY disebut sebagai bidang pandang atau bidang proyeksi. Karena arah jatuh sinar (proyektor) tegak lurus dengan bidang proyeksi, maka proyeksi paralel ini disebut juga sebagai proyeksi orthogonal, misal koordinat 2D pada bidang pandang yang dihasilkan adalah (X_s, Y_s) . Dimana (X_s, Y_s) adalah koordinat (x, y) dari objek 3D.



Gambar 2.2. Proyeksi paralel/orthogonal secara geometris.

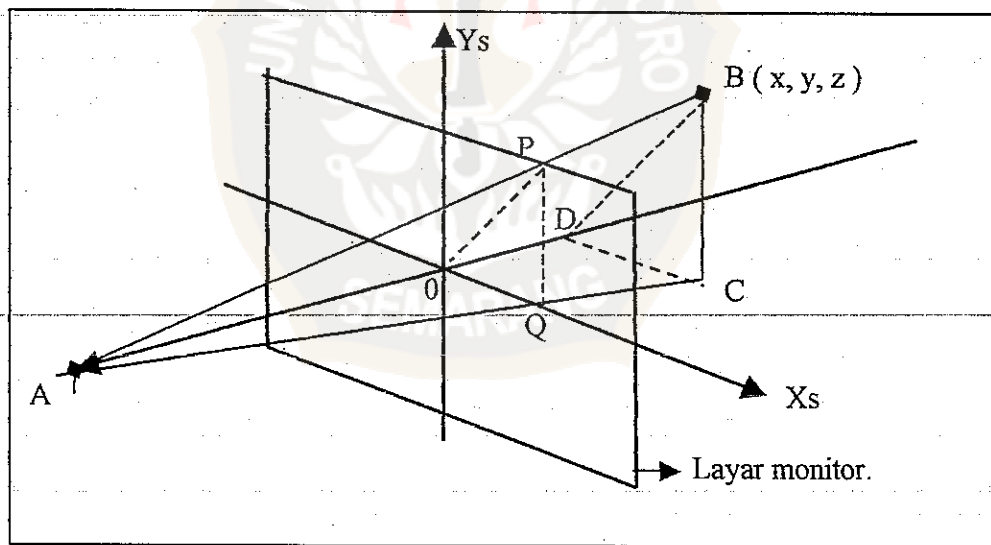
Perhatikan Gambar 2.2., objek 3D berbentuk balok DEFGHIJK diproyeksikan pada suatu bidang 2D datar yang terletak pada sumbu XY . Hasil proyeksi yang

diperoleh adalah objek 2D, ABCO. Koordinat (0,0) layar berada ditengah layar, karena dalam penulisan ini resolusi monitor yang digunakan adalah 320 x 200 pixel. Maka koordinat (0,0) berada pada koordinat layar (160, 100).

2.2.2. Proyeksi Perspektif.

Arah sinar pada objek diasumsikan berasal dari posisi/ tempat yang tidak ditentukan (bebas). Proyeksi perspektif berbeda dengan proyeksi parallel, dimana arah sinar diasumsikan tegak lurus dengan objek dan bidang pandang.

Contoh proyeksi perspektif dari suatu titik $B(x,y,z)$ pada bidang pandang XY, dan mata terletak pada AO sumbu Z. Panjang garis AO disebut jarak mata (*eye distance*).



Gambar 2.3. Proyeksi perspektif secara geometris.

Koordinat objek terletak pada $B(x,y,z)$ yang berhubungan dengan titik A (mata pengamat) oleh garis AO sebagai jarak pandang. Titik B akan menembus bidang pandang, yang diasumsikan layar monitor pada titik P (x,y,z) . Garis DC merupakan panjang objek terhadap sumbu X bidang pandang, garis BC merupakan tinggi objek

terhadap sumbu Y, dan panjang terhadap sumbu Z diwakili oleh garis DO. Pada layar monitor garis OQ merupakan koordinat X dari hasil proyeksi, dan garis PQ sebagai koordinat Y hasil proyeksi.

Panjang garis OQ dan PQ dapat diketahui dengan menggunakan dalil segitiga.

Perhatikan segitiga AOQ dan ADC, maka :

$$\frac{OQ}{DC} = \frac{AO}{AD}$$

maka :

$$OQ = \frac{AO \cdot DC}{AD}$$

Dimana : OQ = koordinat X hasil proyeksi dari objek.

AO = m = jarak antara mata dengan monitor, misal m pixel.

DC = koordinat X tiga dimensi dari objek.

AD = koordinat Z tiga dimensi dari objek.

Jadi dari persamaan OQ diatas dapat diasumsikan sebagai :

$$X_s = \frac{m \cdot x}{z}$$

Panjang PQ dapat diketahui melalui segitiga APQ dan ABC, maka :

$$\frac{PQ}{BC} = \frac{AO}{AD}$$

sehingga :

$$PQ = \frac{AO \cdot BC}{AD}$$

dimana : PQ = koordinat Y hasil proyeksi dari objek.

AO = m = jarak antara mata dan monitor, misal m pixel.

BC = koordinat Y tiga dimensi dari objek.

AD = koordinat Z tiga dimensi dari objek.

Jadi dari persamaan PQ diatas dapat diasumsikan sebagai :

$$Y_s = \frac{m \cdot y}{z}$$

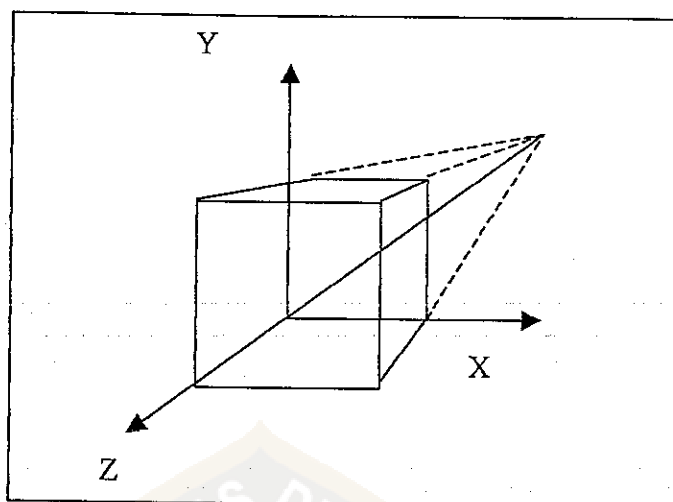
Dari hasil proyeksi, maka akan dapat disimpulkan bahwa :

1. Semakin jauh objek dari pengamat, hasil proyeksi akan semakin kecil, dan sebaliknya.
2. Jika suatu titik (x,y,z) terletak tepat pada bidang pandang, hasil proyeksinya adalah titik itu sendiri.
3. Hasil proyeksi sebuah garis lurus adalah garis lurus, karena hasil proyeksinya dapat diperoleh dengan memproyeksikan kedua titik ujung garis.

2.2.3. Jenis Proyeksi Perspektif.

Didasarkan pada banyaknya titik lenyap (*vanishing point*) yang dapat digunakan untuk menggambarkan proyeksi perpektif. Pada penulisan tugas akhir ini, jenis proyeksi perspektif yang digunakan adalah proyeksi perspektif 1 titik dan proyeksi perspektif 2 titik :

- A. Proyeksi perspektif 1 titik adalah proyeksi perspektif yang mempunyai 1 titik lenyap, yang dapat terletak pada salah satu sumbu koordinat. Seperti terlihat pada contoh Gambar 2.4.

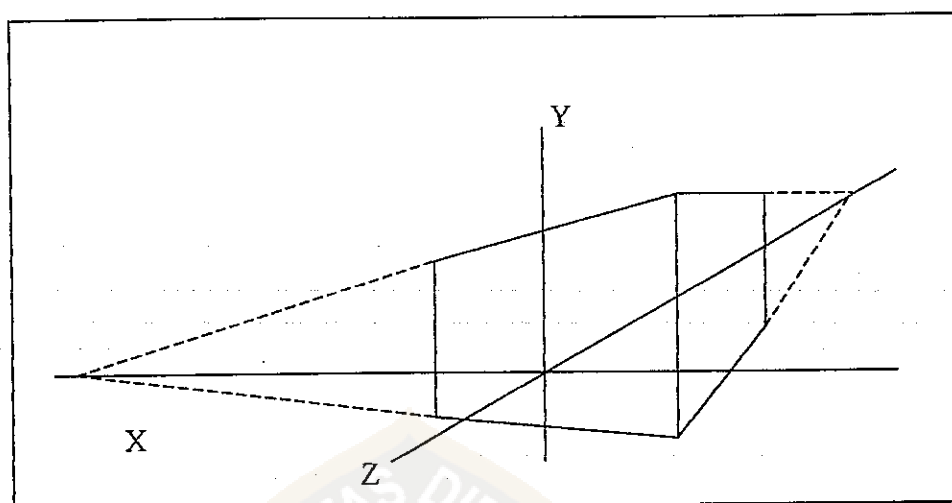


Gambar 2.4. Proyeksi perpektif 1 titik.

Pada Gambar 2.4. titik lenyap objek terletak pada sumbu Z. Mata pengamat terletak sejajar pada sumbu Z, sehingga semua garis yang sejajar dengan sumbu Z, atau yang menembus bidang pandang, akan bertemu pada titik lenyap. Sedangkan garis yang lain adalah garis tegak dan mendatar terhadap bidang pandang.

B. Proyeksi Perspektif 2 Titik (proyeksi angular)

Semua garis sejajar yang mengarah pada kedua titik lenyap, akan bertemu pada titik lenyapnya masing-masing. Sedangkan garis yang tidak berubah adalah garis yang sejajar terhadap sumbu yang titik lenyapnya tidak berada pada sumbu tersebut.



Gambar 2.5. Proyeksi perspektif 2 titik.

Titik lenyap pada Gambar 2.5. adalah sumbu X dan sumbu Z, garis yang sejajar sumbu Y tetap, karena tegak lurus dan berada didepan pengamat.

2.3 Matriks

Definisi matriks adalah susunan segi empat siku-siku, bilangan-bilangan dalam susunan tersebut dinamakan entri dalam matriks.

Ukuran matriks dijelaskan dengan menyatakan banyaknya baris (garis horizontal) dan banyaknya kolom (garis vertikal). Sebuah matriks dengan n baris dan n kolom disebut matriks kuadrat berorde n , dan entri-entri misal : $a_{11}, a_{22}, \dots, a_{nn}$ berada pada diagonal utama.

Pada penulisan perancangan game 3D simulator ini akan digunakan matriks kuadrat berorde 4, sebagai orde matriks transformasi. Sedangkan untuk mendefinisikan koordinat objek digunakan matriks berorde 1×4 .

Misal : matriks A merupakan matriks kuadrat berorde 4, B adalah matriks orde 1 x 4 yang mendefinisikan koordinat objek di titik B (a_{11} , a_{12} , a_{13})

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} ; \quad B = [a_{11} \quad a_{12} \quad a_{13} \quad 1]$$

2.3.1. Invers Matriks.

Invers matriks adalah sebuah matriks yang jika dikalikan dengan matriks dirinya sendiri akan menjadi matriks identitas. Misal : [A] adalah matriks kuadrat berorde 4, [A^{-1}] adalah invers matriks [A], dan [I] adalah matriks identitas, maka :

$$[A^{-1}] \times [A] = [I]$$

Matriks identitas yang digunakan adalah matriks kuadrat berorde 4. Misal : [I] adalah matriks identitas.

$$[I] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.4. Transformasi Objek 3D.

Transformasi geometris pada objek 3D digunakan untuk memperlihatkan realita dari objek yang ditinjau. Aspek penting dalam penyajian transformasi objek 3D, adalah :

1. Sembarang transformasi dapat disajikan ke dalam matriks transformasi.

2. Transformasi yang lebih rumit dapat dinyatakan oleh sebuah matriks transformasi yang merupakan gabungan dari matriks transformasi dari transformasi dasar.

Berdasarkan sistem koordinat yang digunakan, setiap titik objek 3D ditentukan oleh 3 posisi, yaitu : posisi terhadap sumbu X, posisi terhadap sumbu Y, posisi terhadap sumbu Z. Misal : sebuah titik sembarang Q dapat dinyatakan sebagai (q_x, q_y, q_z)

2.4.1. Dilatasi

Dilatasi adalah penggerakan sebuah objek ke lokasi baru dengan menambahkan suatu nilai konstanta untuk setiap titik koordinat yang terdefinisi dalam objek tersebut.

Jika P (p_x, p_y, p_z) adalah posisi titik asal, Q (q_x, q_y, q_z) adalah posisi titik setelah digeser, I adalah matriks identitas, dan tr_x, tr_y, tr_z merupakan nilai konstanta yang menunjukkan besarnya penggeseran pada setiap sumbu koordinat, maka hasil penggeseran dapat dinyatakan sebagai :

$$(q_x, q_y, q_z) = (p_x + tr_x, p_y + tr_y, p_z + tr_z)$$

Persamaan tersebut dapat ditulis dalam bentuk matriks, sebagai :

$$[q_x \ q_y \ q_z \ 1] = [p_x \ p_y \ p_z \ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ tr_x & tr_y & tr_z & 1 \end{bmatrix}$$

2.4.2. Penskalaan

Penskalaan dilakukan dengan melakukan pengalihan setiap titik koordinat dari objek dengan faktor pengali, faktor pengali dapat berbeda pada setiap elemennya.

Hasil dari proses penskalaan adalah perubahan ukuran dari objek, karena dipengaruhi hasil perkalian dari setiap titik koordinatnya dengan faktor pengali.

Jika $Q (q_x, q_y, q_z)$ adalah posisi setelah penskalaan, $P (p_x, p_y, p_z)$ adalah posisi awal, s_x, s_y, s_z adalah matriks transformasi (faktor pengali) pada setiap sumbu koordinat.

Maka penskalaan dapat ditulis sebagai :

$$(q_x, q_y, q_z) = (s_x \cdot p_x, s_y \cdot p_y, s_z \cdot p_z)$$

Dapat dinyatakan dalam bentuk matriks, sebagai :

$$[q_x \ q_y \ q_z \ 1] = [p_x \ p_y \ p_z \ 1] \cdot \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.4.3. Rotasi.

Rotasi adalah perubahan dari suatu koordinat objek ke dalam kedudukan baru dengan menggerakkan seluruh titik koordinat yang didefinisikan pada bentuk awal dengan suatu besaran sudut pada suatu sumbu putar.

Jika $Q (q_x, q_y, q_z)$ adalah posisi setelah rotasi pada suatu sumbu putar, $P (p_x, p_y, p_z)$ adalah posisi awal sebelum dilakukan rotasi, dan R adalah matriks rotasi pada suatu sumbu putar. Sistem koordinat 3D mempunyai tiga buah sumbu putar, maka rotasi tiap sumbu bisa dituliskan sebagai berikut, dengan θ menunjukkan sudut putar.

A. Rotasi terhadap sumbu X.

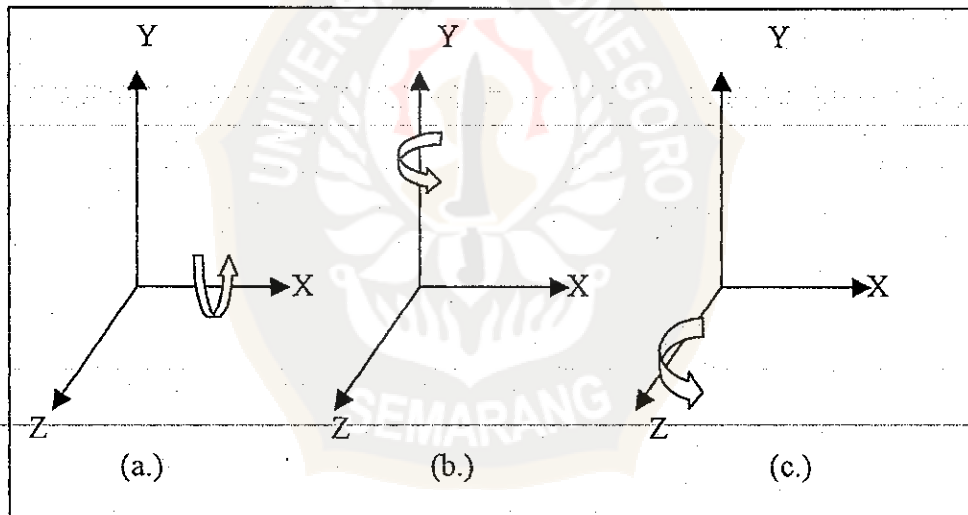
$$[q_x \ q_y \ q_z \ 1] = [p_x \ p_y \ p_z \ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

B. Rotasi terhadap sumbu Y.

$$[q_x \ q_y \ q_z \ 1] = [p_x \ p_y \ p_z \ 1] \cdot \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

C. Rotasi terhadap sumbu Z.

$$[q_x \ q_y \ q_z \ 1] = [p_x \ p_y \ p_z \ 1] \cdot \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

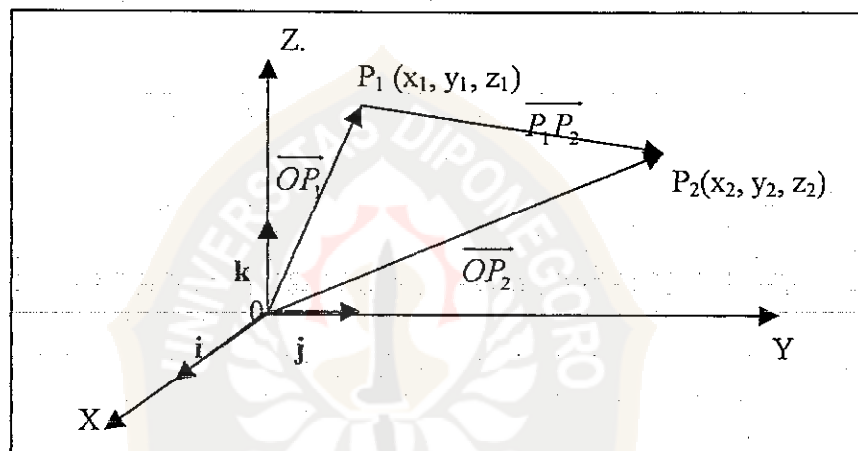


Gambar 2.6. Rotasi pada tiap sumbu putar koordinat.

Rotasi dari suatu objek menghasilkan kedudukan baru suatu objek tergantung dari besar sudut putar (θ), tetapi bentuk dari objek tidak mengalami perubahan. Pada Gambar 2.6.(a.) menunjukkan rotasi pada sumbu X, Gambar 2.6.(b.) merupakan rotasi pada sumbu Y, dan Gambar 2.6.(c.) merupakan rotasi pada sumbu Z.

2.5. Vektor.

Secara geometris vektor dapat dinyatakan sebagai segmen-segmen garis terarah di ruang 2 dimensi atau ruang 3 dimensi, arah panah menentukan arah vektor dan panjang vektor menyatakan besarnya. Ekor panah dinamakan titik awal (*initial point*) dari vektor, dan ujung panah dinamakan titik terminal (*terminal point*). Pada penulisan tugas akhir ini, vektor yang digunakan adalah vektor pada ruang 3 dimensi.



Gambar 2.7. Vektor pada ruang 3 dimensi

Perhatikan Gambar 2.7., terdapat titik $P_1(x_1, y_1, z_1)$ dan $P_2(x_2, y_2, z_2)$ pada bidang XYZ, maka vektor yang dihasilkan adalah :

$$\overline{OP_1} = \mathbf{u} = x_1 \mathbf{i} + y_1 \mathbf{j} + z_1 \mathbf{k} \quad \text{dan} \quad \overline{OP_2} = \mathbf{v} = x_2 \mathbf{i} + y_2 \mathbf{j} + z_2 \mathbf{k}.$$

\mathbf{i} , \mathbf{j} , \mathbf{k} adalah suatu vektor yang terletak sepanjang sumbu koordinat (Gambar 2.7.)

dengan arah vektor berturut-turut pada arah sumbu X, sumbu Y, dan sumbu Z.

2.5.1. Panjang Vektor

Panjang vektor disebut juga norma vektor, perhatikan kembali Gambar 2.8., misal : panjang suatu titik $P_1(x_1, y_1, z_1)$ dari titik $(0,0,0)$ didefinisikan sebagai berikut.

$$|\overline{OP_1}| = |\mathbf{u}| = \sqrt{x_1^2 + y_1^2 + z_1^2}$$

2.5.2. Penambahan Vektor

Penambahan 2 vektor dilakukan dengan operasi penambahan pada masing-masing komponen vektor. Perhatikan kembali Gambar 2.8., misal : vektor $\mathbf{u} = x_1 \mathbf{i} + y_1 \mathbf{j} + z_1 \mathbf{k}$ akan ditambahkan dengan vektor $\mathbf{v} = x_2 \mathbf{i} + y_2 \mathbf{j} + z_2 \mathbf{k}$, maka :

$$\mathbf{u} + \mathbf{v} = (x_1 + x_2) \mathbf{i} + (y_1 + y_2) \mathbf{j} + (z_1 + z_2) \mathbf{k}$$

2.5.3. Pengurangan Vektor.

Pengurangan 2 vektor dilakukan dengan operasi pengurangan pada masing-masing komponen vektor. Perhatikan kembali Gambar 2.7., misal : vektor $\mathbf{u} = x_1 \mathbf{i} + y_1 \mathbf{j} + z_1 \mathbf{k}$ akan ditambahkan dengan vektor $\mathbf{v} = x_2 \mathbf{i} + y_2 \mathbf{j} + z_2 \mathbf{k}$, maka :

$$\mathbf{u} - \mathbf{v} = (x_1 - x_2) \mathbf{i} + (y_1 - y_2) \mathbf{j} + (z_1 - z_2) \mathbf{k}$$

2.5.4. Unit Vektor.

Unit vektor dari suatu vektor adalah vektor lain yang memiliki panjang vektor = 1, tetapi arahnya tetap sama. Setiap vektor mempunyai unit vektor, untuk mencari unit vektor adalah sebagai berikut.

Misal : vektor $\mathbf{u} = x_1 \mathbf{i} + y_1 \mathbf{j} + z_1 \mathbf{k}$, unit vektor dari vektor \mathbf{u} adalah :

$$\text{Unit vektor } \mathbf{u} = \frac{x_1}{\sqrt{x_1^2 + y_1^2 + z_1^2}} \mathbf{i} + \frac{y_1}{\sqrt{x_1^2 + y_1^2 + z_1^2}} \mathbf{j} + \frac{z_1}{\sqrt{x_1^2 + y_1^2 + z_1^2}} \mathbf{k}$$

2.5.5. Hasil Kali Titik (*Dot Product*)

Perhatikan kembali Gambar 2.7., jika \mathbf{u} dan \mathbf{v} adalah vektor di ruang 3 dimensi, dan θ adalah sudut yang dibentuk antara \mathbf{u} dan \mathbf{v} , maka hasil kali titik didefinisikan oleh

$$\mathbf{u} \cdot \mathbf{v} = \begin{cases} |\mathbf{u}| \cdot |\mathbf{v}| \cdot \cos \theta, & \text{jika } \mathbf{u} \neq 0 \text{ dan } \mathbf{v} \neq 0 \\ 0 & \text{jika } \mathbf{u} = 0 \text{ atau } \mathbf{v} = 0, \text{ atau } \cos \theta = 0. \end{cases}$$

$$\text{maka, jika } \mathbf{u} \neq 0 \text{ dan } \mathbf{v} \neq 0, \cos \theta = \frac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{u}| \cdot |\mathbf{v}|}$$

$$\theta = \cos^{-1} \left(\frac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{u}| \cdot |\mathbf{v}|} \right)$$

Dimana, $|\mathbf{u}|$ = norma vektor $\mathbf{u} (x_1, y_1, z_1)$

$|\mathbf{v}|$ = norma vektor $\mathbf{v} (x_2, y_2, z_2)$

2.5.6. Hasil Kali Silang (*Cross Product*)

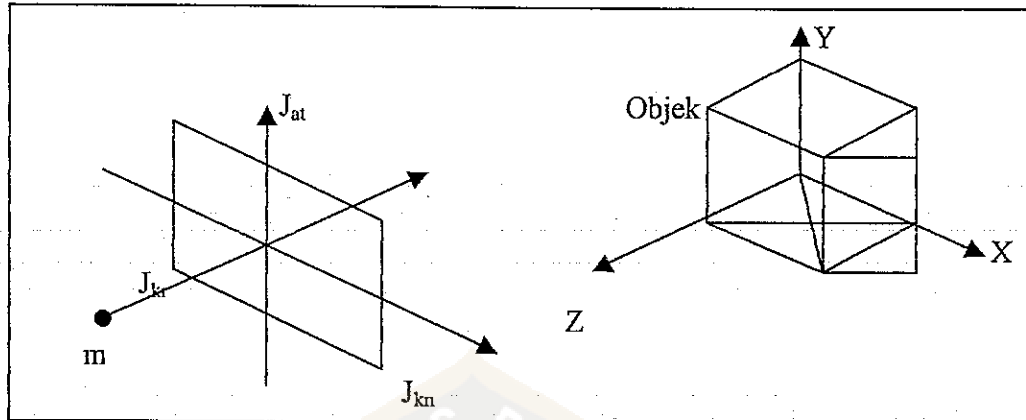
Perhatikan kembali Gambar 2.7., jika $\mathbf{u} (x_1, y_1, z_1)$ dan $\mathbf{v} (x_2, y_2, z_2)$ adalah vektor di ruang 3 dimensi, maka hasil kali silang didefinisikan oleh

$$\mathbf{u} \times \mathbf{v} = (y_1 \cdot z_2 - z_1 \cdot y_2, z_1 \cdot x_2 - x_1 \cdot z_2, x_1 \cdot y_2 - y_1 \cdot x_2)$$

2.6. Pembatasan Bidang Pandang.

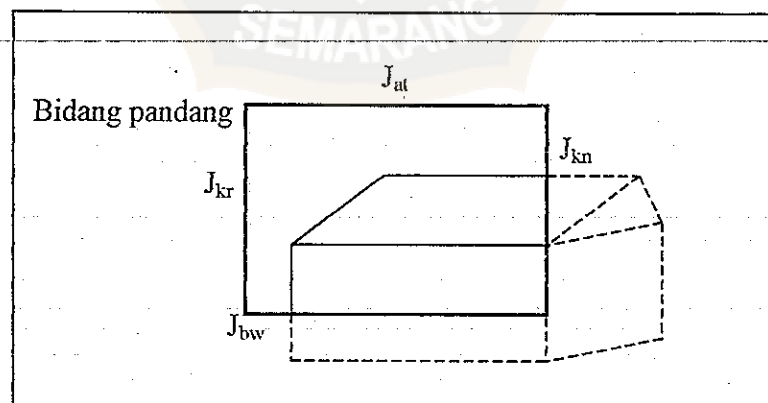
Batas-batas jendela dan posisi mata pengamat ditentukan dalam sistem koordinat pandang. Jendela terletak pada bidang pandang, dan batas-batasnya dinyatakan sebagai $(J_{kr}, J_{at}, J_{kn}, J_{bw})$. Dari batas jendela tersebut dapat dilihat bahwa pada arah mendatar, jendela mempunyai batas dari vertikal kiri = J_{kr} sampai vertikal kanan = J_{kn} , sedangkan pada arah tegak, jendela mempunyai batas horizontal atas = J_{at}

sampai horizontal bawah = J_{bw} . Posisi mata pengamat, $m = (m_x, m_y, m_z)$ dapat berada pada sembarang tempat di bidang pandang.



Gambar 2.8. Posisi jendela dan mata pengamat

Objek pada Gambar 2.8. bersifat 3 dimensi, kemudian diproyeksikan pada bidang pandang yang bersifat 2 dimensi. Sehingga koordinat objek pada proyeksi di bidang pandang akan bersifat 2 dimensi. Koordinat objek yang terlihat hanya koordinat yang berada di depan bidang pandang.



Gambar 2.9. Objek pada suatu bidang pandang

Pada Gambar 2.9. objek telah diproyeksi pada bidang pandang 2 dimensi. Objek yang terlihat hanya sebagian (garis tidak putus), objek yang diluar daerah bidang pandang tidak akan terlihat (garis putus).

2.7. Mode Grafik VGA .

Pada penulisan ini mode grafik VGA mempunyai resolusi 320 x 200 pixel, 256 warna. Pada prinsipnya hasil tampilan pada monitor berasal dari rekaman memori dari RAM (*Random Access Memory*) yang dihasilkan dari *monitor card* pada komputer. Monitor card akan mentransfer gambar yang dihasilkannya pada suatu alokasi alamat memori yang digunakannya pada monitor. Alamat memori tersebut disebut sebagai halaman bayangan (*visual page*).

2.7.1. Alokasi Pixel pada Memori Layar.

Pemindahan dari mode grafik dapat menggunakan *interrupt DOS service 10h*, dengan alamat halaman bayangan dimulai dari \$A000:0 sampai \$A000:63999 atau pada bilangan heksadesimal di \$A000:00000 sampai \$A000:F9FFh.

Setiap *pixel* yang digunakan disimpan dalam 1 *offset* (=1 byte). Sehingga jumlah memori yang digunakan adalah 64000 byte, sebagai latar belakang standar, karena resolusi yang digunakan 320 x 200 pixel, maka tempat penampungan pixel sejumlah $320 \times 200 = 64000$.

0	1	2	318	319
								
....						633998	63999

Gambar 2.10. Sistem alokasi pixel pada monitor VGA.

Gambar 2.10. menggambarkan jumlah pixel horizontal adalah 320 dan jumlah pixel vertikal 200, dengan sistem penyimpanan VGA yang terdiri 64000 offset dan menampung 64000 byte.

Sebelum dimunculkan pada layar monitor, memori yang digunakan untuk menampung proses sementara, disebut memori *buffer*. Memori buffer merupakan memori sementara untuk menampung proses yang belum saatnya dimunculkan pada layar monitor. Kapasitas memori buffer disesuaikan dengan besar memori yang dibutuhkan dalam suatu proses.

2.7.2. Aplikasi Warna pada VGA (*Video Graphic Accelerator*).

Warna yang dihasilkan merupakan campuran warna primer RGB (*Red, Green, Blue*), dengan batasan intensitas setiap warna primer 0 – 63. Intensitas warna menunjukkan sifat warna dari yang paling terang sampai yang paling gelap. Daftar warna pada mode VGA berjumlah 256 dari kombinasi RGB, sedangkan jumlah warna maksimal yang dihasilkan adalah 262144 warna. Jumlah total warna berasal dari :

$$\text{Total warna} = R * G * B = 64 * 64 * 64 = 262144$$

Sistem pewarnaan ini berfungsi untuk memberikan warna pada objek dan latar belakang.

2.8. Pemrograman Turbo Pascal untuk Grafik

Program animasi unit grafik harus mampu menampung dan menjalankan ribuan looping dan pergerakan dalam setiap satuan waktu secara bersamaan. Unit grafik yang akan digunakan dalam penulisan ini adalah VGACAD.TPU, yang mampu menangani operasi grafik, pergerakan objek, dan manipulasi matematis yang lebih kompleks (Nalwan, 1996).

VGACAD.TPU merupakan pengembangan unit GRAPH.TPU yang merupakan salah satu unit grafik standar Turbo Pascal 7.0.

Agar proses animasi dapat berjalan maka layar monitor harus bersifat grafik VGA, layar monitor komputer pada awalnya bersifat mode teks sehingga harus diubah menjadi mode grafik VGA. Perpindahan dari mode layar yang berbasis teks (mode standar monitor PC) ke mode grafik dipergunakan perintah `OpenVga256`.

Untuk menutup mode grafik dan kembali ke mode teks dipergunakan perintah `CloseVga256`.