

Modul Praktikum 5

Proyeksi 3D

A. Tujuan

1. Mahasiswa mampu menjelaskan tentang 3D *viewing*;
2. Mahasiswa mampu menjelaskan tentang proyeksi parallel;
3. Mahasiswa mampu menjelaskan tentang proyeksi perspektif;
4. Mahasiswa mampu menerapkan 3D *viewing* dan proyeksi dengan OpenGL.

B. Dasar Teori

Alam fisik dalam persepsi manusia adalah sebuah ruang yang berformat 3 dimensi. Benda-benda yang ada di dalamnya umum direpresentasikan menggunakan format 3 dimensi: panjang, lebar, dan tinggi. Dalam matematika, 3 dimensi ini biasa dinyatakan dalam sistem koordinat kartesian. Koordinat kartesian 3 dimensi memiliki 3 bidang yang saling tegak lurus satu dengan yang lainnya. Tiap bidang memiliki sumbu yang koordinat yang biasa disebut sumbu x, y, dan z.

3 dimensi di OpenGL

OpenGL menggunakan matrix sebagai komponen dasar untuk menghasilkan tampilan pada layar. Semua matrix ini didefinisikan untuk dapat memproses operasi-operasi dalam 3 dimensi.

Jika pada pelajaran-pelajaran sebelumnya obyek dibuat dalam 2 dimensi, sebenarnya obyek-obyek tersebut adalah obyek 3 dimensi. Hanya saja dimensi ketiga diabaikan. Termasuk didalam konsep ini adalah transformasi. Transformasi selalu dilakukan dalam format 3 dimensi.

Proyeksi

Seringkali diperlukan untuk menggambarkan obyek 3 dimensi kedalam format 2 dimensi, contohnya: arsitek perlu menuangkan idenya tentang sebuah obyek bangunan 3 dimensi diatas kertas (2 dimensi). Contoh lain adalah penggambaran dunia OpenGL yang 3 dimensi ke layar monitor yang 2 dimensi. Perubahan format dari 3 dimensi menjadi 2 dimensi ini memerlukan proses/aturan khusus. Proses/aturan ini disebut proyeksi grafis.

Ada 2 jenis proyeksi:

Perspektif

Cara mata manusia dan kamera menangkap gambar obyek sekelilingnya. Obyek yang jauh terlihat kecil, obyek yang dekat terlihat besar. 2 garis sejajar akan terlihat menyatu di kejauhan.

Parallel

Garis proyeksi selalu sejajar baik di obyek 3 dimensi maupun di penggambaran 2 dimensinya. Jenis proyeksi ini digunakan oleh orang-orang teknik (Arsitek, teknik mesin, teknik sipil) dalam menggambar pekerjaannya.

Proyeksi di OpenGL

Dalam OpenGL, diperlukan beberapa perubahan setting agar sebuah aplikasi dapat menampilkan obyek 3 dimensi. Jika pada aplikasi-aplikasi sebelumnya perintah proyeksi yang digunakan adalah `gluOrtho2D(kiri, kanan, bawah, atas)` untuk menampilkan obyek 2 dimensi, perintah yang sama harus diganti agar aplikasi dapat menampilkan obyek 3 dimensi:

Untuk proyeksi parallel, gunakan `glOrtho(kiri, kanan, bawah, atas, dekat, jauh);`

Untuk proyeksi perspektif, gunakan `gluPerspective(fovy, aspek rasio, dekat, jauh);`

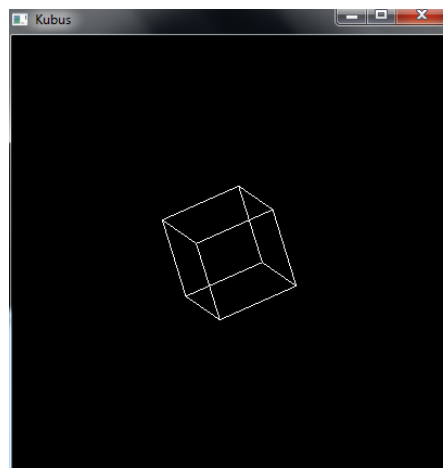
Coba masukkan *source code* pembuatan kubus 3d berikut :

```

Untitled1.rc  main.cpp
1  #include<windows.h>
2  #include<gl/glut.h>
3  #include <math.h>
4
5  void display(void)
6  {
7      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
8      glColor3f(1.0, 1.0, 1.0);
9      glRotated(35,1,1,1);
10     glutWireCube(2);
11     glFlush();
12 }
13
14 void init(void)
15 {
16     glClearColor (0.0, 0.0, 0.0, 0.0);
17     glMatrixMode(GL_PROJECTION);
18     glLoadIdentity();
19     glOrtho(-5.0, 5.0, -5.0, 5.0, -5.0, 5.0);
20     glMatrixMode(GL_MODELVIEW);
21     glShadeModel (GL_FLAT);
22     glEnable(GL_DEPTH_TEST);
23 }
24
25 int main(int argc, char** argv)
26 {
27     glutInit(&argc, argv);
28     glutInitDisplayMode(GLUT_RGB | GLUT_DEPTH);
29     glutInitWindowSize(400, 400);
30     glutInitWindowPosition(100, 100);
31     glutCreateWindow("Kubus");
32     init();
33     glutDisplayFunc(display);
34     glutMainLoop();
35     return 0;
36 }

```

Hasilnya :



Perhatikan perbedaan program diatas dengan program-program sebelumnya:

1. Menggunakan glOrtho, bukan gluOrtho2D;
2. Perintah glShadeModel diperlukan disini;
3. Perintah glEnable(GL_DEPTH_TEST) diperlukan disini;

4. Pada beberapa kondisi, diperlukan juga modifikasi:
 - a) `glutInitDisplayMode(GLUT_RGB | GLUT_DEPTH);;`
 - b) `glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);`

Program diatas menggunakan perintah `glutWireCube(ukuran)` untuk membuat sebuah kubus 3 dimensi. Perintah ini adalah perintah bawaan dari glut. Perintah bawaan lain untuk membuat obyek 3 dimensi adalah:

1. `glutWireTeapot(GLdouble size);`
2. `glutWireSphere(GLdouble radius, GLint slices, GLint stacks);`
3. `glutWireCone(GLdouble base, GLdouble height, GLint slices, GLint stacks);`
4. `glutWireTetrahedron(void);`
5. `glutWireOctahedron(void);`

Selain bentuk *wireframe* diatas, glut menyediakan juga obyek-obyek bawaan 3 dimensi dalam format solid dimana obyek ditampilkan dalam bentok penuh/solid. Format perintahnya sama seperti obyek glut wire diatas hanya tinggal mengganti kata Wire dengan kata Solid:

1. `glutWireCube(GLdouble size) => glutSolidCube(GLdouble size);;`
2. `glutWireTeapot(GLdouble size) => glutSolidTeapot(GLdouble size);`
3. `glutWireSphere(GLdouble radius, GLint slices, GLint stacks) => glutSolidSphere(GLdouble radius, GLint slices, GLint stacks);`
4. `glutWireCone(GLdouble base, GLdouble height, GLint slices, GLint stacks) => glutSolidCone(GLdouble base, GLdouble height, GLint slices, GLint stacks);`
5. `glutWireTetrahedron(void) => glutSolidTetrahedron(void);`
6. `glutWireOctahedron(void) => glutSolidOctahedron(void).`

Jika dicoba, obyek-obyek ini tampak tidak jelas bentuknya di layar. Yang tampak hanyalah blok berwarna putih. Ini wajar. Ada kalanya obyek yang ingin dibuat/ditampilkan tidak tersedia dalam library bawaan glut. Untuk kasus seperti ini, programmer perlu membuat sendiri obyek tersebut. Di OpenGL, untuk membuat obyek 3 dimensi, salah satu caranya adalah dengan membuat tiap sisi obyek dari polygon 2 dimensi. Berikut contoh program kubus 3 dimensi dimana tiap sisinya dibuat dari persegi 2 dimensi.

```

#include<windows.h>
#include<gl/glut.h>
#include <math.h>
void display()
{
    glClear(GL_COLOR_BUFFER_BIT| GL_DEPTH_BUFFER_BIT);
    glRotated(-35, 1,1,1);
    //depan
    glColor3f(0.0,0.0,1.0);
    glBegin(GL_POLYGON);
    glVertex3f(-1.0, -1.0, 1.0);
    glVertex3f(1.0, -1.0, 1.0);
    glVertex3f(1.0, 1.0, 1.0);
    glVertex3f(-1.0, 1.0, 1.0);
    glEnd();
    //belakang
    glColor3f(0.0,1.0,0.0);
    glBegin(GL_POLYGON);
    glVertex3f(1.0, -1.0, -1.0);
    glVertex3f(-1.0, -1.0, -1.0);
    glVertex3f(-1.0, 1.0, -1.0);
    glVertex3f(1.0, 1.0, -1.0);
    glEnd();
    //kiri
    glColor3f(1.0,0.0,0.0);
    glBegin(GL_POLYGON);
    glVertex3f(-1.0, -1.0, -1.0);
    glVertex3f(-1.0, -1.0, 1.0);
    glVertex3f(-1.0, 1.0, 1.0);
    glVertex3f(-1.0, 1.0, -1.0);
    glEnd();
    //kanan
    glColor3f(0.0,1.0,1.0);
    glBegin(GL_POLYGON);
    glVertex3f(1.0, -1.0, -1.0);
    glVertex3f(1.0, 1.0, -1.0);
    glVertex3f(1.0, 1.0, 1.0);

```

```

    glVertex3f(1.0, -1.0, 1.0);
    glEnd();
    //bawah
    glColor3f(1.0,0.0,1.0);
    glBegin(GL_POLYGON);
    glVertex3f(1.0, -1.0, 1.0);
    glVertex3f(-1.0, -1.0, 1.0);
    glVertex3f(-1.0, -1.0, -1.0);
    glVertex3f(1.0, -1.0, -1.0);
    glEnd();
    //atas
    glColor3f(1.0,1.0,0.0);
    glBegin(GL_POLYGON);
    glVertex3f(-1.0, 1.0, 1.0);
    glVertex3f(1.0, 1.0, 1.0);
    glVertex3f(1.0, 1.0, -1.0);
    glVertex3f(-1.0, 1.0, -1.0);
    glEnd();
    glFlush();
}

void myinit()
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-3.0,3.0,-3.0,3.0,-3.0,3.0);
    glMatrixMode(GL_MODELVIEW);
    glClearColor(0.0,0.0,0.0,1.0);
    glColor3f(0.0,0.0,0.0);
    glShadeModel(GL_FLAT);
    glEnable(GL_DEPTH_TEST);
}

int main(int argc, char* argv[])
{
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(400,400);
    glutInitWindowPosition(100,100);

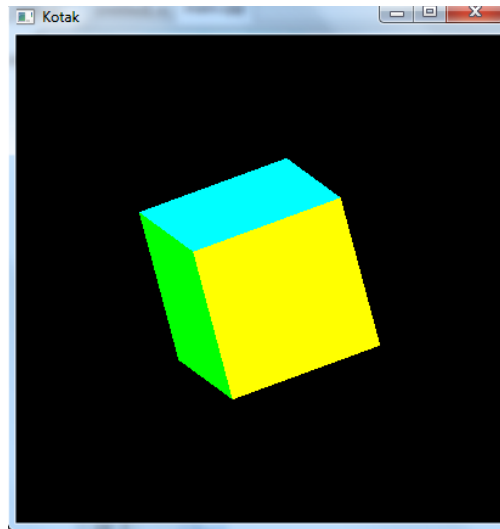
```

```

    glutCreateWindow("Kotak");
    glutDisplayFunc(display);
    myinit();
    glutMainLoop();
    return 0;
}

```

Hasilnya :



TUGAS MINGGUAN

1. Untuk program 1 modifikasi parameter `glRotated`, lalu amati perubahan tampilannya.
2. Untuk program 1 ganti nilai `glutWireCube`, lalu amati perubahan tampilannya.
3. Untuk program 2, modifikasi parameter `glRotated`, lalu amati perubahan tampilannya.
4. Rubahlah nama windownya dengan (NPM & Nama Masing-masing)
5. Buat laporannya dalam format pdf, lengkap dengan identitas dan cover.
6. Waktu pengerjaan 1 Minggu.