

Modul Praktikum 6

Pemodelan 3D dengan Jaring Poligon

A. Tujuan

1. Mahasiswa mampu menjelaskan tentang konsep pemodelan 3D dengan Jaring Poligon;
2. Mahasiswa mampu menerapkan Pemodelan 3D dengan Jaring Poligon dengan OpenGL.

B. Dasar Teori

Jaring Poligon

Ilustrasi yang terbaik untuk menggambarkan jaring poligon adalah menggunakan kawat yang dihubungkan membentuk jaring kawat untuk membuat objek tertentu.

Pemodelan dengan Jaring Poligon

Ada beberapa cara mendefinisikan jaring. Untuk kubus dapat diwakili dengan membuat satu daftar yang berisi daftar masing-masing poligon, lokasi titik dan normal untuk masing-masing titik. Dengan perwakilan ini kubus terdiri dari 24 titik, 24 normal dan 6 permukaan. Daftar ini dapat menciptakan data berlebih-lebihan karena ada beberapa titik dan normal yang didaftarkan lebih dari satu kali. Untuk itu, lebih baik digunakan metoda lain yaitu membuat 3 daftar. Ketiga daftar tersebut adalah:

- 1) Daftar untuk titik
- 2) Daftar untuk permukaan
- 3) Daftar normal

Dengan daftar ini, sebuah kubus terdiri dari (1) 8 buah titik, (2) 6 buah permukaan, (3) 6 buah normal.

1) Daftar titik

Titik	X	Y	Z
0	-1.0	-1.0	1.0
1	1.0	-1.0	1.0
2	1.0	1.0	1.0
3	-1.0	1.0	1.0
4	-1.0	-1.0	-1.0
5	-1.0	1.0	-1.0
6	1.0	1.0	-1.0
7	1.0	-1.0	-1.0

2) Daftar normal

Titik	Nx	Ny	Nz
0	0.0	0.0	1.0
1	0.0	0.0	-1.0
2	0.0	1.0	0.0
3	0.0	-1.0	0.0
4	1.0	0.0	0.0
5	-1.0	0.0	0.0

3) Daftar muka

Muka	Titik	normal
0(depan)	0,1,2,3	0,0,0,0
1(belakang)	4,5,6,7	1,1,1,1
2(atas)	5,3,2,6	2,2,2,2
3(bawah)	4,7,1,0	3,3,3,3
4(kanan)	7,6,2,1	4,4,4,4
5(kiri)	4,0,3,5	5,5,5,5

Selain itu diperlukan lintasan muka yaitu untuk menentukan:

Bagian dalam poligon;

Bagian luar poligon;

Letak arah normal

Lintasan muka mengikuti arah jarum jam sebagaimana terlihat dari luar objek.

Dalam poligon akan selalu ada pada sebelah kiri Anda.

Perhitungan Normal:

Mengapa grafika komputer membutuhkan normal?

Normal mengatakan pada anda letak bagian permukaan sebelah luar;

Normal digunakan untuk menghitung seberapa banyak sinar cahaya mengenai permukaan bagian luar;

Normal menentukan seberapa licinnya permukaan dirender.

C. Praktikum

Buatlah program dengan OpenGL untuk menampilkan objek 3D yaitu kubus dengan metoda pembuatan 3 daftar yaitu (1) daftar untuk titik (2) daftar untuk permukaan (3) daftar normal.

Contoh *source codenya*:

Program-1

```
//-----  
#include <GL/glut.h>  
GLfloat light_diffuse[] = {1.0, 0.0, 0.0, 1.0};  
//GLfloat light_diffuse[] = {0.9f, 0.9f, 0.9f, 1.0f};  
GLfloat light_position[] = {1.0, 1.0, 1.0, 0.0};  
//GLfloat light_position[] = {2.0f, 2.0f, 2.0f, 0.0f};  
void gambar3D(void)  
{  
glBegin(GL_QUADS);  
// muka depan  
glNormal3f( 0.0f, 0.0f, 1.0f); //Normal menuju Kita  
glVertex3f(-1.0f, -1.0f, 1.0f); //Titik 1 (depan)  
glVertex3f( 1.0f, -1.0f, 1.0f); //Titik 2 (depan)  
glVertex3f( 1.0f, 1.0f, 1.0f); //Titik 3 (depan)  
glVertex3f(-1.0f, 1.0f, 1.0f); //Titik 4 (depan)  
// muka belakang  
glNormal3f( 0.0f, 0.0f,-1.0f); //Normal menjauh Kita  
glVertex3f(-1.0f, -1.0f,-1.0f); //Titik 1 (belakang)  
// muka atas  
glNormal3f( 0.0f, 1.0f, 1.0f); //Normal berarah atas
```

```

glVertex3f(-1.0f, 1.0f, -1.0f); //Titik 1 (atas)
glVertex3f(-1.0f, 1.0f, 1.0f); //Titik 2 (atas)
glVertex3f( 1.0f, 1.0f, 1.0f); //Titik 3 (atas)
glVertex3f( 1.0f, 1.0f, -1.0f); //Titik 4 (atas)
//muka bawah
glNormal3f( 0.0f, -1.0f, 0.0f); //Normal berarah bawah
glVertex3f(-1.0f, -1.0f, -1.0f); //Titik 1 (bawah)
glVertex3f( 1.0f, -1.0f, -1.0f); //Titik 2 (bawah)
glVertex3f( 1.0f, -1.0f, 1.0f); //Titik 3 (bawah)
glVertex3f(-1.0f, -1.0f, 1.0f); //Titik 4 (bawah)
//muka kanan
glNormal3f( 1.0f, 0.0f, 0.0f); //Normal berarah kekanan
glVertex3f( 1.0f, -1.0f, -1.0f); //Titik 1 (kanan)
glVertex3f( 1.0f, 1.0f, -1.0f); //Titik 2 (kanan)
glVertex3f( 1.0f, 1.0f, 1.0f); //Titik 3 (kanan)
glVertex3f( 1.0f, -1.0f, 1.0f); //Titik 4 (kanan)
//muka kiri
glNormal3f(-1.0f, 0.0f, 0.0f); //Normal berarah ke kiri
glVertex3f(-1.0f, -1.0f, -1.0f); //Titik 1 (kiri)
glVertex3f(-1.0f, -1.0f, 1.0f); //Titik 2 (kiri)
glVertex3f(-1.0f, 1.0f, 1.0f); //Titik 3 (kiri)
glVertex3f(-1.0f, 1.0f, -1.0f); //Titik 4 (kiri)
glEnd();
}
void display(void)
{
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
gambar3D();
glutSwapBuffers();
}
void inisialisasi(void)
{
//pencahayaannya
glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
glEnable(GL_LIGHT0);
glEnable(GL_LIGHTING);
//Buffer
glEnable(GL_DEPTH_TEST);
//setup kubus
glMatrixMode(GL_PROJECTION);
gluPerspective(40.0, 1.0, 1.0, 10.0);
glMatrixMode(GL_MODELVIEW);
gluLookAt(0.0, 0.0, 5.0, // melihat pada (0,0,5)
0.0, 0.0, 0.0, // center pada (0,0,0)
0.0, 1.0, 0.0); // arah Y
/* Mengatur posisi sudut */
//glTranslatef(0.0, 0.0, -1.0);
glRotatef(30, 0.0, 1.0, 1.0);
glRotatef(15, 1.0, 1.0, 0.0);

```

```

}
int main(int argc, char **argv)
{
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
glutCreateWindow("Grafika Komputer 3 Dimensi");
glutDisplayFunc(display);
inisialisasi();
glutMainLoop();
return 0;
}
//-----

```

Tampilannya sebagai berikut:



b. Polihendra

Polihendra adalah sebuah jaring yang mempunyai ruang terhingga dan tertutup.

Fungsi polihendra dalam OpenGL

Nama	Fungsi OpenGL (kawat)	Fungsi OpenGL (pejal)	Jumlah bidang
Tetrahedron	glutWireTetrahedron()	glutSolidTetrahedron()	4
Octahedron	glutWireOctahedron()	glutSolidOctahedron()	8
Dodecahedron	glutWireDodecahedron()	glutSolidDodecahedron()	10
Icosahedron	glutWireIcosahedron()	glutSolidIcosahedron()	20

Contoh-2:

Buatlah program dengan OpenGL untuk menampilkan objek 3D Polihendra, yaitu Tetrahedron, Octahedron, Dodecahedron, Icosahedron dengan tampilan sketsa kawat saja.

Program-2:

```
//-----  
  
#include <GL/glut.h>  
  
GLfloat light_diffuse[] = {1.0, 0.0, 0.0, 1.0};  
GLfloat light_position[] = {1.0, 1.0, 1.0, 0.0};  
void gambar3D(void)  
{  
    glClear(GL_COLOR_BUFFER_BIT);  
    glColor3f(0.0, 0.0, 1.0);  
    glPushMatrix();  
    glTranslatef(0.0, 1.0, -8.0);  
    glutWireTetrahedron();  
    glPopMatrix();  
    glPushMatrix();  
    glTranslatef(2.0, 1.0, -8.0);  
    glutWireOctahedron();  
    glPopMatrix();  
    glPushMatrix();
```

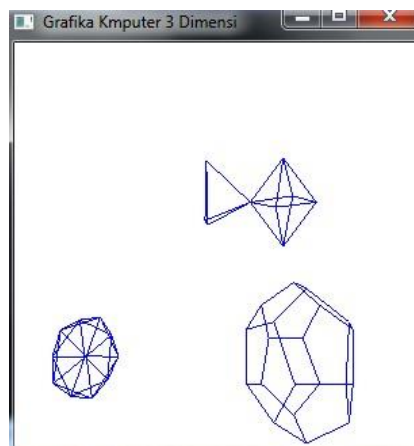


```

glTranslatef(2.5f, -2.5f, -8.0);
glutWireDodecahedron();
glPopMatrix();
glPushMatrix();
glTranslatef(-4, -2.5f, -8.0);
glutWireIcosahedron();
glPopMatrix();
glFlush();
}
void display(void)
{
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
gambar3D();
glutSwapBuffers();
}
void inisialisasi(void)
{
int w=800, h=600;
glShadeModel(GL_FLAT);
glClearColor(1.0, 1.0, 1.0, 0.0);
glClearAccum(0.0, 0.0, 0.0, 0.0);
glViewport(0, 0, w, h);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluPerspective(60.0, (GLfloat) w/(GLfloat) h, 1.0, 20.0);
glMatrixMode(GL_MODELVIEW);
}
int main(int argc, char **argv)
{
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
glutCreateWindow("Grafika Komputer 3 Dimensi");
glutDisplayFunc(display);
inisialisasi();
glutMainLoop();
return 0;
}
//

```

Tampilannya sebagai berikut:



Contoh-3

Buatlah program dengan OpenGL untuk menampilkan objek 3D Polihendra,yaitu Tetrahedron,Octahedron,Dodecahedon,Icosahedron pejal.

Program 3:

```
#include <GL/glut.h>
GLfloat light_diffuse[] = {1.0, 0.0, 0.0, 1.0};
GLfloat light_position[] = {1.0, 1.0, 1.0, 0.0};
void gambar3D(void)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    GLfloat intensitasCahaya[]={0.9f, 0.9f, 0.9f, 1.0f};
    GLfloat posisiCahaya[]={2.0f, 2.0f, 2.0f, 0.0f};
    glLightfv(GL_LIGHT0, GL_POSITION, posisiCahaya);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, intensitasCahaya);
    GLfloat bahan_ambient[]={0.0f, 0.5f, 0.6f, 1.0f};
    GLfloat bahan_diffuse[]={0.6f, 0.6f, 0.6f, 1.0f};
    GLfloat bahan_specular[]={1.0f, 1.0f, 1.0f, 1.0f};
    GLfloat bahan_shininess[]={90.0f};
    glMaterialfv(GL_FRONT, GL_AMBIENT, bahan_ambient);
    glMaterialfv(GL_FRONT, GL_DIFFUSE, bahan_diffuse);
    glMaterialfv(GL_FRONT, GL_SPECULAR, bahan_specular);
    glMaterialfv(GL_FRONT, GL_SHININESS, bahan_shininess);
    glPushMatrix();
    glTranslatef(0.0, 1.0, -8.0);
    glutSolidTetrahedron();
    glPopMatrix();
    glPushMatrix();
    glTranslatef(2.0, 1.0, -8.0);
    glutSolidOctahedron();
    glPopMatrix();
    glPushMatrix();
    glTranslatef(2.5f, -2.5f, -8.0);
    glutSolidDodecahedron();
    glPopMatrix();
    glPushMatrix();
    glTranslatef(-4, -2.5f, -8.0);
    glutSolidIcosahedron();
}
```

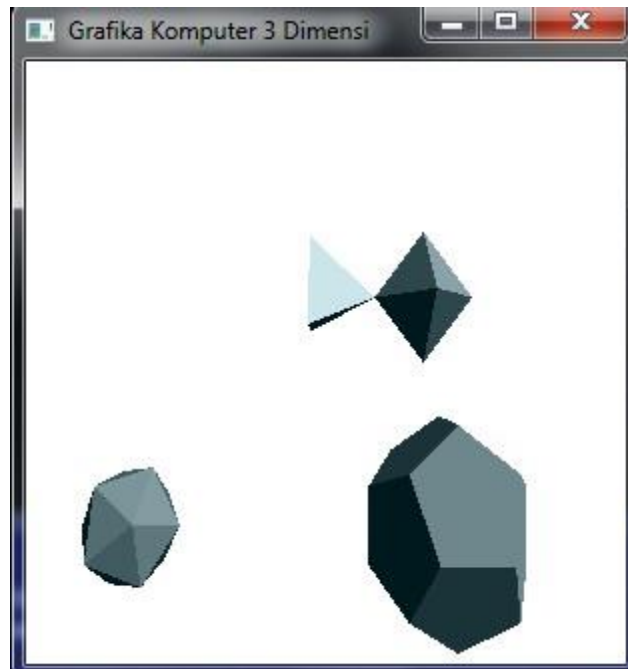


```

glPopMatrix();
glFlush();
}
void display(void)
{
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
gambar3D();
glutSwapBuffers();
}
void inisialisasi(void)
{
int w=800, h=600;
glShadeModel(GL_FLAT);
glClearColor(1.0,1.0,1.0,0.0);
glClearAccum(0.0,0.0,0.0,0.0);
glViewport(0,0,w,h);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluPerspective(60.0,(GLfloat) w/(GLfloat) h,1.0,20.0);
glMatrixMode(GL_MODELVIEW);
glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
glEnable(GL_SMOOTH);
glEnable(GL_DEPTH_TEST);
glEnable(GL_NORMALIZE);
}
int main(int argc, char **argv)
{
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
glutCreateWindow("Grafika Komputer 3 Dimensi");
glutDisplayFunc(display);
inisialisasi();
glutMainLoop();
return 0;
}
//-----\

```

Tampilannya:



TUGAS MINGGUAN

1. Buatlah program dengan OpenGL untuk menampilkan objek 3D dengan metoda 3 daftar yaitu (1) titik (2) normal (3) permukaan:
 - a) Prisma;
 - b) Antiprisma;
 - c) Tetrahedron;
 - d) Octahedron;
 - e) Dodecahedron;
 - f) Icosahedron
2. Modifikasi program-3 dengan mengubah warna masing-masing polihendra dengan warna yang berbeda;
3. Rubahlah nama windownya dengan (NPM & Nama Masing-masing);
4. Buat laporannya dalam format pdf, lengkap dengan identitas dan cover;
5. Waktu pengerjaan 1 Minggu.