**LAPORAN TUGAS BESAR KOMPUTER GRAFIKA**

**PENGIMPLEMENTASIAN KOMPUTER GRAFIKA DALAM REPRESENTASI KAMAR**

**Oleh :**

**YULIUS ANGGA E.P 10108429**

**IMAM ASHADI 10108436**

**GHEA RATIMANJARI 10108449**

**R. IDHAM AZIZ M. 10108465**



**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

**UNIVERSITAS KOMPUTER INDONESIA**

**2012BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Seiring pesatnya perkembangan pada ilmu komputer grafik. Hal ini dimungkinkannya untuk menemukan perangkat keras yang lebih canggih, teknik-teknik ataupun algoritma untuk menciptakan gambar yang lebih berkualitas pada layar komputer sehingga mampu memvisualisasikan objek-objek yang lebih realistis. Aplikasi grafika komputer adalah program komputer yang rancang untuk menghasilkan grafis atau repsentasi visual dari data kedalam bentuk gambar. Adapun ruang lingkup aplikasi grafika komputer mulai dari objek dua dimensi sampai objek tiga dimensi.

Dalam sebuah grafika computer 3D adalah representasi dari data geometric 3 dimensi sebagai hasil dari pemrosesan dan pemberian efek cahaya terhadap grafika computer 2D. hasil ini kadang kala tampilkan secara waktu nyata untuk keperluan simulasi. Secara umum prinsip yang dipakai adalah mirip dengan grafika computer 2D, dalam hal penggunan algoritma grafika computer 3D sering disebut sebagai model 3D. namun, model 3D ini lebih menekankan pada representasi matematis untuk objek 3 dimensi. Data matematis ini belum bias dikatakan sebagai gambar grafis hingga saat ditampilkan secara visual pada layar computer atau printer. Proses penampilan suatu model matematis kebentuk citra 2 biasanya dikenal dengan proses 3D rendering.

Objek tiga dimensi merupakan salah satu komponen yang memegang peranan penting sebagai bentuk informasi visual, karena dibentuk oleh sekumpulan titik yang mempunyai nilai kooordinat tertentu yang dihubungkan sehingga menjadi sebuah objek tiga dimensi.

Oleh karena perkembangan grafika komputer yang dapat menghasilkan gambar 3 dimensi dengan batuan *shadow* dan *lightning* maka pada tugas besar ini penulis mengkaji mengenai “***Implementasi grafika computer untuk memvisualisasi kamar”****.*

* 1. **Perumusan Masalah**

Bagaimana mengimplementasikan komputer grafika dalam representasi kamar menggunakan opengl.

* 1. **Tujuan Perancangan**

Adapun tujuan yang ingin dicapai dalam perancangan desain interior ini adalah sebagai berikut :

1. Menambah wawasan mengenai perancangan pembangunan kamar menggunakan OpenGL
   1. **Manfaat Perancangan**

Dengan melihat kembali latar belakang diatas manfaat penelitian ini adalah mempresentasikan kamar dengan menggunakan opengl.

* 1. **Batasan Masalah**

Dalam pembangunan kamar ini kami memiliki ruang lingkup yang akan kami implementasikan yakni:

1. Tempat tidur
2. Kursi
3. Meja belajar
4. Jendela
5. Lukisan
6. Lemari

# BAB 2

# PERANCANGAN

# Komponen Pembangun Objek

Dalam membangun kamar ini dibutuhkan beberapa objek sebagai komponen-komponen pembentuknya untuk membangun design kamar yang diinginkan. Komponen yang digunakan untuk membuat objek kamar adalah:

1. solid cube
2. gl\_quads

# Bahasa Pemograman

Bahasa pemograman yang digunakan untuk membangun kamar ini adalah dev C++ dan openGL.

# Tampilan Objek Kamar

## Berikut adalah keseluruhan tampilan kamar

1. Tampilan kamar tampak depan



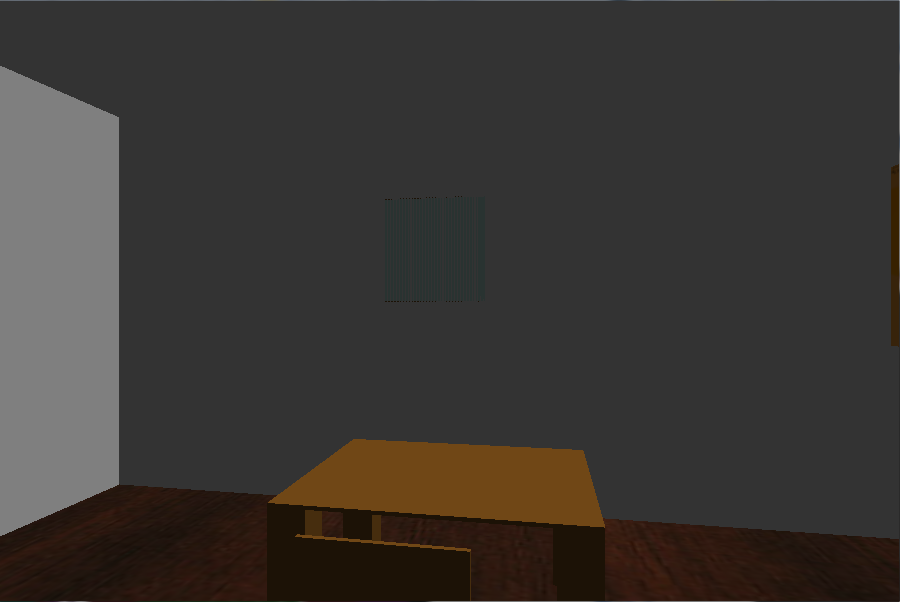
**Gambar 2.1** Tampak depan kamar

1. Tampilan belakang kamar



**Gambar 2.2** Tampak belakang kamar

1. Tampilan samping kamar



**Gambar 2.3** Tampak samping kamar

# Interaksi

Untuk mengatur posisi kamera tampilan kamar dapat menggunakan keyboard dengan interaksi sebagai berikut :

1. Tekan atas untuk maju
2. Tekan bawah untuk mundur
3. Tekan x untuk memutar objek kebawah
4. Tekan z untuk memutar objek ke sampan (arah z)
5. Tekan y untuk memutar objek kesamping (arah y)

# BAB 3 KESIMPULAN DAN SARAN

## Kesimpulan

Merancang sebuah kamar menggunakan opengl membutuh beberapa objek seperti gl\_quads dan solid cube yang dibentuk sedemikian rupa menggunakan transalasi ataupun rotasi sehingga membentuk sebuah kamar

Selain itu, untuk mengatur posisi kamera user dapat menggunakan control di keyboard.

## Saran

Pada tahap merancang kamar ini masih banyak kekurangan sehingga diperlukan saran untuk memperbaiki kekurangan tersebut.

* + - 1. Objek kurang lengkap sehingga kurang mempresentasikan kamar dalam kehidupan nyata
      2. Penggunaan lighting yang masih kurang kedepannya diharapkan ada sebuah interaksi user untuk mengatur lighting

DAFTAR PUSTAKA

1. http://nehe.gamedev.net (diakses pada tanggal 17 Juni 2012 pukul 20.30)
2. http://opengl.org (diakses pada tanggal 17 juni pukul 20.00)

LAMPIRAN

/\*

Tugas besar Grafkom

kelompok 5 : Yulius angga E.P (10108429)

Imam ashadi (10108436)

Ghea ratimanjari (10108449)

R. Idham Aziz Marlin (10108465)

- tekan bawah untuk mundur

- tekan atas untuk maju

- tekan x untuk memutar objek kebawah (arah x)

- tekan z untuk memutar objek kesamping (arah z)

- tekan y untuk memutar objek kesamping (arah y)

\*/

#include <cstdlib>

#include <iostream>

#include <GL/gl.h>

#include <GL/glu.h>

#include <GL/glut.h>

#include <math.h>

#define PI 3.1415926535

using namespace std;

int w=900, h=600, z=0;

int x2=0, y2=0, sudut=0, z2=0;

int i,j;

float sudut1 = 0;

int n;

static float materialA[] = {0.8,1.0,0.97}; // coklat tua

static float materialB[] = {0.55,0.35,0.11}; // biru muda

static float materialC[] = {1.0,1.0,1.0}; // putih

static float materialD[] = {0.016,0.216,0.518}; //biru tua

static float materialE[] = {0.86,0.62,0.12}; // coklat muda

GLUquadric \*texture;

GLuint loadTexture;

void kursi(void);

void meja(void);

void gambarRuang(void);

int LoadBitmap(char \*filename)

{

FILE \* file;

char temp;

long i;

// own version of BITMAPINFOHEADER from windows.h for Linux compile

struct {

int biWidth;

int biHeight;

short int biPlanes;

unsigned short int biBitCount;

unsigned char \*data;

} infoheader;

GLuint num\_texture;

if( (file = fopen(filename, "rb"))==NULL) return (-1); // Open the file for reading

fseek(file, 18, SEEK\_CUR); /\* start reading width & height \*/

fread(&infoheader.biWidth, sizeof(int), 1, file);

fread(&infoheader.biHeight, sizeof(int), 1, file);

fread(&infoheader.biPlanes, sizeof(short int), 1, file);

if (infoheader.biPlanes != 1) {

printf("Planes from %s is not 1: %u\n", filename, infoheader.biPlanes);

return 0;

}

// read the bpp

fread(&infoheader.biBitCount, sizeof(unsigned short int), 1, file);

if (infoheader.biBitCount != 24) {

printf("Bpp from %s is not 24: %d\n", filename, infoheader.biBitCount);

return 0;

}

fseek(file, 24, SEEK\_CUR);

// read the data

if(infoheader.biWidth<0){

infoheader.biWidth = -infoheader.biWidth;

}

if(infoheader.biHeight<0){

infoheader.biHeight = -infoheader.biHeight;

}

infoheader.data = (unsigned char \*) malloc(infoheader.biWidth \* infoheader.biHeight \* 3);

if (infoheader.data == NULL) {

printf("Error allocating memory for color-corrected image data\n");

return 0;

}

if ((i = fread(infoheader.data, infoheader.biWidth \* infoheader.biHeight \* 3, 1, file)) != 1) {

printf("Error reading image data from %s.\n", filename);

return 0;

}

for (i=0; i<(infoheader.biWidth \* infoheader.biHeight \* 3); i+=3) { // reverse all of the colors. (bgr -> rgb)

temp = infoheader.data[i];

infoheader.data[i] = infoheader.data[i+2];

infoheader.data[i+2] = temp;

}

fclose(file); // Closes the file stream

glGenTextures(1, &num\_texture);

glBindTexture(GL\_TEXTURE\_2D, num\_texture); // Bind the ID texture specified by the 2nd parameter

// The next commands sets the texture parameters

glTexParameterf(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_S, GL\_REPEAT); // If the u,v coordinates overflow the range 0,1 the image is repeated

glTexParameterf(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_T, GL\_REPEAT);

glTexParameterf(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR); // The magnification function ("linear" produces better results)

glTexParameterf(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR\_MIPMAP\_NEAREST); //The minifying function

glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_MODULATE);

// Finally we define the 2d texture

glTexImage2D(GL\_TEXTURE\_2D, 0, 3, infoheader.biWidth, infoheader.biHeight, 0, GL\_RGB, GL\_UNSIGNED\_BYTE, infoheader.data);

// And create 2d mipmaps for the minifying function

gluBuild2DMipmaps(GL\_TEXTURE\_2D, 3, infoheader.biWidth, infoheader.biHeight, GL\_RGB, GL\_UNSIGNED\_BYTE, infoheader.data);

free(infoheader.data); // Free the memory we used to load the texture

return (num\_texture); // Returns the current texture OpenGL ID

}

void Init(void)

{

glClearColor(0,0,0,1);

glEnable(GL\_DEPTH\_TEST);

glDepthFunc(GL\_LEQUAL);

glHint(GL\_PERSPECTIVE\_CORRECTION\_HINT, GL\_NICEST);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(45.0,(GLdouble) w/(GLdouble) h, 1.0,300.0);

glMatrixMode(GL\_MODELVIEW);

glEnable(GL\_LIGHTING); // mengaktifkan fungsi pencahayaan

glEnable(GL\_LIGHT0); // mengaktifkan sumber cahaya

glShadeModel(GL\_SMOOTH); //mengaktifkan shade model

glHint(GL\_PERSPECTIVE\_CORRECTION\_HINT, GL\_NICEST);

glEnable(GL\_NORMALIZE);

glDisable(GL\_CULL\_FACE);

texture = gluNewQuadric();

gluQuadricTexture(texture, GL\_TRUE);

}

void resize(int w1,int h1)

{

if (h1==0)

{

h1=1;

}

// mereset viewport yang sekarang

glViewport(0,0,w1,h1);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

// menghitung aspek rasio window

gluPerspective(60.0,(float) w1/(float) h1, 0.1,100.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

void lukisan()

{

glPushMatrix();

glEnable ( GL\_TEXTURE\_2D );

glBindTexture ( GL\_TEXTURE\_2D, loadTexture);

loadTexture = LoadBitmap("seohyun.bmp");

glTranslatef(0.0, 0.0, 18);

glScalef(5.0, 3.0, 0.4);

glTranslatef(0.0, 0.5, 1.0);

glRotatef(180,0,1,0);

glBegin(GL\_POLYGON);

glTexCoord2f(0.0, 0.0);

glVertex3f( -1.0,-1.0, 0.0);

glTexCoord2f(1.0, 0.0);

glVertex3f( 1.0,-1.0, 0.0);

glTexCoord2f(1.0, 1.0);

glVertex3f( 1.0, 1.0, 0.0);

glTexCoord2f(0.0, 1.0);

glVertex3f( -1.0, 1.0, 0.0);

glTexCoord2f(0.0, 0.0);

glVertex3f( -1.0,-1.0, 0.0);

glEnd();

//glutSolidCube(1.0);

glDisable ( GL\_TEXTURE\_2D );

glPopMatrix();

}

void kursi(void)

{

glClearColor(0,0,0,1);

glLoadIdentity();

glTranslatef(0,0,z);

glRotatef(sudut,x2,y2,z2);

glPushMatrix();

glTranslatef(-5.0,-7.0,0);

glScalef(3.0,0.1,3.0);

glutSolidCube(1.0);

glPopMatrix();

glPushMatrix();

glTranslatef(-6.5,-5.5,0);

glRotatef(90,0.0,0.0,1.0);

glScalef(3.0,0.1,3.0);

glutSolidCube(1.0);

glPopMatrix();

glPushMatrix();

glTranslatef(-5.0,-8.0,0);

glScalef(0.5,2.0,0.5);

glutSolidCube(1.0);

glPopMatrix();

}

void meja(void)

{

glPushMatrix();

glTranslatef(0.0,-5.0,0.0);

glScalef(6.0,0.1,8.0);

glutSolidCube(1.0);

glPopMatrix();

glPushMatrix();

glTranslatef(-2.5,-7.0,3.5);

glScalef(1.0,4.0,1.0);

glutSolidCube(1.0);

glPopMatrix();

glPushMatrix();

glTranslatef(2.5,-7.0,3.5);

glScalef(1.0,4.0,1.0);

glutSolidCube(1.0);

glPopMatrix();

glPushMatrix();

glTranslatef(-2.5,-7.0,-3.5);

glScalef(1.0,4.0,1.0);

glutSolidCube(1.0);

glPopMatrix();

glPushMatrix();

glTranslatef(2.5,-7.0,-3.5);

glScalef(1.0,4.0,1.0);

glutSolidCube(1.0);

glPopMatrix();

}

void jendela(float x,float y)

{

glMaterialfv(GL\_FRONT,GL\_DIFFUSE,materialA);

glMaterialfv(GL\_FRONT,GL\_AMBIENT,materialA);

glBegin(GL\_QUADS); // jendela

glVertex3f(10.0,0.0,0);

glVertex3f(10.0,0.0,-x);

glVertex3f(10.0,y,-x);

glVertex3f(10.0,y,0);

glEnd();

glMaterialfv(GL\_FRONT,GL\_DIFFUSE,materialB);

glMaterialfv(GL\_FRONT,GL\_AMBIENT,materialB);

glBegin(GL\_QUADS); // kusen jendela

glVertex3f(10.0,-0.05,0.05); // bawah

glVertex3f(10.0,-0.05,-x-0.05);

glVertex3f(10.0,-0.01,-x-0.05);

glVertex3f(10.0,-0.01,0.05);

glVertex3f(10.0,y+0.01,0.05); // atas

glVertex3f(10.0,y+0.01,-x-0.05);

glVertex3f(10.0,y+0.05,-x-0.05);

glVertex3f(10.0,y+0.05,0.05);

glVertex3f(10.0,-0.01,0+0.05); // kiri

glVertex3f(10,-0.01,0.01);

glVertex3f(10,y+0.01,0.01);

glVertex3f(10,y+0.01,+0.05);

glVertex3f(10.0,-0.01,-x-0.05); // kanan

glVertex3f(10,-0.01,-x-0.01);

glVertex3f(10,y+0.01,-x-0.01);

glVertex3f(10,y+0.01,-x-0.05);

glEnd();

}

void kaki(float h, float tebal)

{

glPushMatrix();

glBegin(GL\_QUADS);

glScalef(1.0,4.0,1.0);

glNormal3f(0,0,1);

glVertex3f(0,0,0);

glVertex3f(tebal,0,0);

glVertex3f(tebal,h,0);

glVertex3f(0,h,0);

glNormal3f(1,0,0);

glVertex3f(tebal,0,0);

glVertex3f(tebal,0,-tebal);

glVertex3f(tebal,h,-tebal);

glVertex3f(tebal,h,0);

glNormal3f(0,0,-1);

glVertex3f(tebal,0,-tebal);

glVertex3f(0,0,-tebal);

glVertex3f(0,h,-tebal);

glVertex3f(tebal,h,-tebal);

glNormal3f(-1,0,0);

glVertex3f(0,0,-tebal);

glVertex3f(0,0,0);

glVertex3f(0,h,0);

glVertex3f(0,h,-tebal);

glEnd();

glPopMatrix();

}

void kotak1(float x, float y, float z)

{

glPushMatrix();

glBegin(GL\_QUADS);

glNormal3f(1,0,0);

glVertex3f(x,0,0);

glVertex3f(x,0,-z);

glVertex3f(x,y,-z);

glVertex3f(x,y,0);

glNormal3f(0,1,0);

glVertex3f(x,y,0);

glVertex3f(x,y,-z);

glVertex3f(10,y,-z);

glVertex3f(0,y,0);

glNormal3f(-1,0,0);

glVertex3f(0,y,0);

glVertex3f(0,y,-z);

glVertex3f(0,0,-z);

glVertex3f(0,0,0);

glNormal3f(0,-1,0);

glVertex3f(0,0,0);

glVertex3f(0,0,-z);

glVertex3f(x,0,-z);

glVertex3f(x,0,0);

glEnd();

glPopMatrix();

}

void kotak2(float x, float y, float z)

{

glPushMatrix();

glBegin(GL\_QUADS);

glNormal3f(1,0,0);

glVertex3f(x,0,0);

glVertex3f(x,0,-z);

glVertex3f(x,y,-z);

glVertex3f(x,y,0);

glNormal3f(0,1,0);

glVertex3f(x,y,0);

glVertex3f(x,y,-z);

glVertex3f(0,y,-z);

glVertex3f(0,y,0);

glNormal3f(-1,0,0);

glVertex3f(0,y,0);

glVertex3f(0,y,-z);

glVertex3f(0,0,-z);

glVertex3f(0,0,0);

glNormal3f(0,-1,0);

glVertex3f(0,0,0);

glVertex3f(0,0,-z);

glVertex3f(x,0,-z);

glVertex3f(x,0,0);

glNormal3f(0,0,1);

glVertex3f(0,0,0);

glVertex3f(x,0,0);

glVertex3f(x,y,0);

glVertex3f(0,y,0);

glNormal3f(0,0,-1);

glVertex3f(0,0,-z);

glVertex3f(x,0,-z);

glVertex3f(x,y,-z);

glVertex3f(0,y,-z);

glEnd();

glPopMatrix();

}

void kotak3(float x, float y, float z)

{

glPushMatrix();

glBegin(GL\_QUADS);

glNormal3f(1,0,0);

glVertex3f(x,0,0);

glVertex3f(x,0,-z);

glVertex3f(x,y,-z);

glVertex3f(x,y,0);

glNormal3f(-1,0,0);

glVertex3f(0,y,0);

glVertex3f(0,y,-z);

glVertex3f(0,0,-z);

glVertex3f(0,0,0);

glNormal3f(0,-1,0);

glVertex3f(0,0,0);

glVertex3f(0,0,-z);

glVertex3f(x,0,-z);

glVertex3f(x,0,0);

glEnd();

glPopMatrix();

}

void pintu()

{

glPushMatrix();

glTranslatef(4.5,-5.5,-17.0);

glScalef(6.0,5.0,2.5);

glTexCoord2f(0.0, 0.0);glutSolidCube(1.5);

glPopMatrix();

glPushMatrix();

glEnable ( GL\_TEXTURE\_2D );

glBindTexture ( GL\_TEXTURE\_2D, loadTexture);

loadTexture = LoadBitmap("rak.bmp");

glBegin(GL\_QUADS);

//tembok depan

glColor3f(1.0,0.0,0.0);

glNormal3f(5.0f, 5.0f, 5.0f);

glTexCoord2f(0.0, 0.0);glVertex3f(0.0f, -1.4f,-15.0f); // atas kiri

glTexCoord2f(1.0, 0.0);glVertex3f(0.0f,-9.5f,-15.0f); // bawah kiri

glTexCoord2f(1.0, 1.0);glVertex3f( 9.0f,-9.5f,-15.0f); // bawah kanan

glTexCoord2f(0.0, 1.0);glVertex3f( 9.0f, -1.4f,-15.0f); // atas kanan

glEnd();

glDisable ( GL\_TEXTURE\_2D );

glPopMatrix();

}

void tempatTidur()

{

glPushMatrix();

glTranslatef(-15.0,-8.5,-12.0);

glScalef(6.0,1.0,9.0);

glutSolidCube(1.5);

glPopMatrix();

glMaterialfv(GL\_FRONT,GL\_DIFFUSE,materialD);

glMaterialfv(GL\_FRONT,GL\_AMBIENT,materialD);

glPushMatrix();

glTranslatef(-15.0,-8.5,-12.0);

glScalef(6.0,2.0,10.0);

glutSolidCube(1.3);

glPopMatrix();

}

void gambarRuang(void)

{

glClearColor(0,0,0,1);

glLoadIdentity();

glTranslatef(0,0,z);

glRotatef(sudut,x2,y2,z2);

glMaterialfv(GL\_FRONT,GL\_DIFFUSE,materialC);

glMaterialfv(GL\_FRONT,GL\_AMBIENT,materialC);

glPushMatrix();

glEnable ( GL\_TEXTURE\_2D );

glBindTexture ( GL\_TEXTURE\_2D, loadTexture);

loadTexture = LoadBitmap("wood1.bmp");

glBegin(GL\_QUADS);

//lantai

glNormal3f(0.0f, 1.0f, 0.0f);

glTexCoord2f(0.0, 0.0);glVertex3f(-20.0f,-10.0f,-20.0f); // belakang kiri

glTexCoord2f(1.0, 0.0);glVertex3f(-20.0f,-10.0f, 20.0f); // depan kiri

glTexCoord2f(1.0, 1.0);glVertex3f( 10.0f,-10.0f, 20.0f); // depan kanan

glTexCoord2f(0.0, 1.0);glVertex3f( 10.0f,-10.0f,-20.0f); // belakang kanan

glEnd();

glDisable ( GL\_TEXTURE\_2D );

glPopMatrix();

glBegin(GL\_QUADS);

//atap

glNormal3f(0.0f,-1.0f, 0.0f);

glVertex3f(-20.0f, 10.0f, 20.0f); // depan kiri

glVertex3f(-20.0f, 10.0f,-20.0f); // belakang kiri

glVertex3f( 10.0f, 10.0f,-20.0f); // belakang kanan

glVertex3f( 10.0f, 10.0f, 20.0f); // depan kanan

//tembok depan

glColor3f(1.0,0.0,0.0);

glNormal3f(0.0f, 0.0f, 1.0f);

glVertex3f(-20.0f, 10.0f,-20.0f); // atas kiri

glVertex3f(-20.0f,-10.0f,-20.0f); // bawah kiri

glVertex3f( 10.0f,-10.0f,-20.0f); // bawah kanan

glVertex3f( 10.0f, 10.0f,-20.0f); // atas kanan

//tembok belakang

glNormal3f(0.0f, 0.0f,-1.0f);

glVertex3f( 10.0f, 10.0f, 20.0f); // atas kanan

glVertex3f( 10.0f,-10.0f, 20.0f); // bawah kanan

glVertex3f(-20.0f,-10.0f, 20.0f); // bawah kiri

glVertex3f(-20.0f, 10.0f, 20.0f); // atas kiri

//tembok kiri

glNormal3f(1.0f, 0.0f, 0.0f);

glVertex3f(-20.0f, 10.0f, 20.0f); // atas depan

glVertex3f(-20.0f,-10.0f, 20.0f); // bawah depan

glVertex3f(-20.0f,-10.0f,-20.0f); // bawah belakang

glVertex3f(-20.0f, 10.0f,-20.0f); // atas belakang

//tembok kanan

glNormal3f(-1.0f, 0.0f, 0.0f);

glVertex3f( 10.0f, 10.0f,-20.0f); // atas belakang

glVertex3f( 10.0f,-10.0f,-20.0f); // bawah belakang

glVertex3f( 10.0f,-10.0f, 20.0f); // bawah depan

glVertex3f( 10.0f, 10.0f, 20.0f); // ayas depan

glEnd();

glutSwapBuffers();

}

void fungsiKeyboard(unsigned char key, int x, int y)

{

if (key =='s') z+=5;

else if (key == 'x')

{

x2=1;

y2=0;

z2=0;

sudut+=10;

}

else if (key == 'y')

{

y2=1;

x2=0;

z2=0;

sudut+=-10;

}

else if (key == 'z')

{

y2=0;

x2=0;

z2=1;

sudut+=-10;

}

}

void fungsiSpecialKeyboard(int key, int x, int y)

{

switch(key)

{

case GLUT\_KEY\_UP:

z+=5;

break;

case GLUT\_KEY\_DOWN:

z-=5;

break;

}

}

void setMaterial()

{

GLfloat mat\_ambient[] = {0.7f,0.7f,0.7f,1.0f};

GLfloat mat\_diffuse[] = {0.6f,0.6f,0.6f,1.0f};

GLfloat mat\_specular[] = {1.0f,1.0f,1.0f,1.0f};

GLfloat mat\_shininess[] = {50.0f};

glMaterialfv(GL\_FRONT,GL\_AMBIENT,mat\_ambient);

glMaterialfv(GL\_FRONT,GL\_DIFFUSE,mat\_diffuse);

glMaterialfv(GL\_FRONT,GL\_SPECULAR,mat\_specular);

glMaterialfv(GL\_FRONT,GL\_SHININESS,mat\_shininess);

}

void setLighting()

{

GLfloat lightIntensity[] = {0.7f,0.7f,0.7f,1.0f};

GLfloat light\_position[] = {2.0f,6.0f,3.0f,0.0f};

glLightfv(GL\_LIGHT0,GL\_DIFFUSE,lightIntensity);

glLightfv(GL\_LIGHT0,GL\_POSITION,light\_position);

}

void timer(int value)

{

glutPostRedisplay();

glutTimerFunc(50,timer,0);

}

void displayObjek()

{

setLighting();

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

jendela(5.0,5.0);

kursi();

meja();

lukisan();

pintu();

tempatTidur();

gambarRuang();

glFlush();

}

int main (int argc, char \*\*argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_DEPTH | GLUT\_RGBA);

glutInitWindowPosition(100,100);

glutInitWindowSize(w,h);

glutCreateWindow("Objek Kamar");

gluOrtho2D(-w/2,w/2,-h/2,h/2);

glutDisplayFunc(displayObjek);

glutReshapeFunc(resize);

glutKeyboardFunc(fungsiKeyboard);

glutSpecialFunc(fungsiSpecialKeyboard);

glutTimerFunc(1,timer,0);

Init();

glutMainLoop();

}