**TUGAS BESAR**

**PROGRAM ANIMASI 3D BUS**

**DENGAN OPENGL DEVC++**

Diajukan Untuk Memenuhi

Salah Satu Tugas Mata Kuliah Komputer Grafika

Dosen : Hendri Karisma, S. Kom.



Disusun Oleh : 10108422 – Sandi Satriawan

10108418 – Sugiri

10108388 – Reyza Gamaresa

10108420 – Aditya Rinaldy

Kelas : IF-8/S1/VIII

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

**UNIVERSITAS KOMPUTER INDONESIA**

**2012**

**SOURCE CODE**

#include <iostream>

#include "stdlib.h"

#include "gl/glut.h"

#include "math.h"

#include "ambilgambar.h"

#include "terrain.h"

using namespace std;

int w=600, h=600, z=10;

int x1=0, y2=0, sudut=0, z1=0, a=0, b=0, c=0, d=0;

float skalaX=1, skalaY=1, skalaZ=1;

int cx, cy;

const GLfloat light\_ambient[] = { 0.3f, 0.3f, 0.3f, 1.0f };

const GLfloat light\_diffuse[] = { 0.7f, 0.7f, 0.7f, 1.0f };

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

const GLfloat light\_position[] = { 1.0f, 1.0f, 1.0f, 1.0f };

const GLfloat light\_ambient2[] = { 0.3f, 0.3f, 0.3f, 0.0f };

const GLfloat light\_diffuse2[] = { 0.3f, 0.3f, 0.3f, 0.0f };

const GLfloat mat\_ambient[] = { 0.8f, 0.8f, 0.8f, 1.0f };

const GLfloat mat\_diffuse[] = { 0.8f, 0.8f, 0.8f, 1.0f };

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

const GLfloat high\_shininess[] = { 100.0f };

class Terrain {

private:

int w; //Width

int l; //Length

float\*\* hs; //Heights

Vec3f\*\* normals;

bool computedNormals; //Whether normals is up-to-date

public:

Terrain(int w2, int l2) {

w = w2;

l = l2;

hs = new float\*[l];

for(int i = 0; i < l; i++) {

hs[i] = new float[w];

}

normals = new Vec3f\*[l];

for(int i = 0; i < l; i++) {

normals[i] = new Vec3f[w];

}

computedNormals = false;

}

~Terrain() {

for(int i = 0; i < l; i++) {

delete[] hs[i];

}

delete[] hs;

for(int i = 0; i < l; i++) {

delete[] normals[i];

}

delete[] normals;

}

int width() {

return w;

}

int length() {

return l;

}

//Sets the height at (x, z) to y

void setHeight(int x, int z, float y) {

hs[z][x] = y;

computedNormals = false;

}

//Returns the height at (x, z)

float getHeight(int x, int z) {

return hs[z][x];

}

//Computes the normals, if they haven't been computed yet

void computeNormals() {

if (computedNormals) {

return;

}

//Compute the rough version of the normals

Vec3f\*\* normals2 = new Vec3f\*[l];

for(int i = 0; i < l; i++) {

normals2[i] = new Vec3f[w];

}

for(int z = 0; z < l; z++) {

for(int x = 0; x < w; x++) {

Vec3f sum(0.0f, 0.0f, 0.0f);

Vec3f out;

if (z > 0) {

out = Vec3f(0.0f, hs[z - 1][x] - hs[z][x], -1.0f);

}

Vec3f in;

if (z < l - 1) {

in = Vec3f(0.0f, hs[z + 1][x] - hs[z][x], 1.0f);

}

Vec3f left;

if (x > 0) {

left = Vec3f(-1.0f, hs[z][x - 1] - hs[z][x], 0.0f);

}

Vec3f right;

if (x < w - 1) {

right = Vec3f(1.0f, hs[z][x + 1] - hs[z][x], 0.0f);

}

if (x > 0 && z > 0) {

sum += out.cross(left).normalize();

}

if (x > 0 && z < l - 1) {

sum += left.cross(in).normalize();

}

if (x < w - 1 && z < l - 1) {

sum += in.cross(right).normalize();

}

if (x < w - 1 && z > 0) {

sum += right.cross(out).normalize();

}

normals2[z][x] = sum;

}

}

//Smooth out the normals

const float FALLOUT\_RATIO = 0.5f;

for(int z = 0; z < l; z++) {

for(int x = 0; x < w; x++) {

Vec3f sum = normals2[z][x];

if (x > 0) {

sum += normals2[z][x - 1] \* FALLOUT\_RATIO;

}

if (x < w - 1) {

sum += normals2[z][x + 1] \* FALLOUT\_RATIO;

}

if (z > 0) {

sum += normals2[z - 1][x] \* FALLOUT\_RATIO;

}

if (z < l - 1) {

sum += normals2[z + 1][x] \* FALLOUT\_RATIO;

}

if (sum.magnitude() == 0) {

sum = Vec3f(0.0f, 1.0f, 0.0f);

}

normals[z][x] = sum;

}

}

for(int i = 0; i < l; i++) {

delete[] normals2[i];

}

delete[] normals2;

computedNormals = true;

}

//Returns the normal at (x, z)

Vec3f getNormal(int x, int z) {

if (!computedNormals) {

computeNormals();

}

return normals[z][x];

}

};

GLuint loadTexture(Image\* image) {

GLuint textureId;

glGenTextures(1, &textureId); //Make room for our texture

glBindTexture(GL\_TEXTURE\_2D, textureId); //Tell OpenGL which texture to edit

//Map the image to the texture

glTexImage2D(GL\_TEXTURE\_2D, //Always GL\_TEXTURE\_2D

0, //0 for now

GL\_RGB, //Format OpenGL uses for image

image->width, image->height, //Width and height

0, //The border of the image

GL\_RGB, //GL\_RGB, because pixels are stored in RGB format

GL\_UNSIGNED\_BYTE, //GL\_UNSIGNED\_BYTE, because pixels are stored

//as unsigned numbers

image->pixels); //The actual pixel data

return textureId; //Returns the id of the texture

}

GLuint \_textureId;

//Loads a terrain from a heightmap. The heights of the terrain range from

//-height / 2 to height / 2.

Terrain\* loadTerrain(const char\* filename, float height) {

Image\* image = loadBMP(filename);

Terrain\* t = new Terrain(image->width, image->height);

for(int y = 0; y < image->height; y++) {

for(int x = 0; x < image->width; x++) {

unsigned char color =

(unsigned char)image->pixels[3 \* (y \* image->width + x)];

float h = height \* ((color / 255.0f) - 0.5f);

t->setHeight(x, y, h);

}

}

delete image;

t->computeNormals();

return t;

}

float \_angle = 60.0f;

Terrain\* \_terrain;

void cleanup() {

delete \_terrain;

}

void myKeyboard(unsigned char key, int x, int y){

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

else if (key == 'd') z-=5;

else if (key == 'w') {

x1=1;

y2=0;

z1=0;

sudut+=10;

}

else if (key == 's') {

y2=1;

x1=0;

z1=0;

sudut+=-10;

}

else if (key == 'q') {

y2=0;

x1=0;

z1=1;

sudut+=-10;

}

else if (key == 'z') {

a=-1;

b=0;

c=0;

d=1;

}

else if (key == 'c') {

a=1;

b=0;

c=0;

d=-1;

}

}

void init(){

glShadeModel(GL\_SMOOTH);

glShadeModel(GL\_SMOOTH);

glShadeModel(GL\_SMOOTH);

glEnable(GL\_NORMALIZE);

glEnable(GL\_COLOR\_MATERIAL);

glClearColor(0.0f,0.0f,0.0f,0.0f);

glClearDepth(1.0f);

glEnable(GL\_DEPTH\_TEST);

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

Image\* image = loadBMP("1.bmp");

\_textureId = loadTexture(image);

delete image;

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

return;

}

void renderScene(void){

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

glClearColor(0.0, 1.8, 5.0, 0.0);

glLoadIdentity();

glTranslatef(0,0,z-50);

glRotatef(sudut+30,x1,y2,z1);

glScalef(skalaX, skalaY, skalaZ);

glPushMatrix();

float scale = 70.0f / max(\_terrain->width() - 1, \_terrain->length() - 1);

glScalef(scale, scale, scale);

glTranslatef(-(float)(\_terrain->width() - 1) / 2,

0.0f,

-(float)(\_terrain->length() - 1) / 2);

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0.2f, 0.9f, 0.f);

for(int z = 0; z < \_terrain->length() - 1; z++) {

//Makes OpenGL draw a triangle at every three consecutive vertices

glBegin(GL\_TRIANGLE\_STRIP);

for(int x = 0; x < \_terrain->width(); x++) {

Vec3f normal = \_terrain->getNormal(x, z);

glNormal3f(normal[0], normal[1], normal[2]);

glVertex3f(x, \_terrain->getHeight(x, z), z);

normal = \_terrain->getNormal(x, z + 1);

glNormal3f(normal[0], normal[1], normal[2]);

glVertex3f(x, \_terrain->getHeight(x, z + 1), z + 1);

}

glEnd();

} glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

// bis

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(1,1,1);

glRotatef(1,0,0,0);

glTranslatef(0,4,80);

glScalef(4,4.5,5);

glutSolidCube(3);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(1,0,0);

glRotatef(1,0,0,0);

glTranslatef(3,0,0);

glutSolidCube(3);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(1,0,0);

glRotatef(1,0,0,0);

glTranslatef(3,0,0);

glutSolidCube(3);

glDisable(GL\_COLOR\_MATERIAL);

//jdep

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(0.9,0.6,0.73);

glutSolidCube(1.3);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(0.04,-0.02,-1.4);

glutSolidCube(1.3);

glDisable(GL\_COLOR\_MATERIAL);

//bandepnkiri

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(-0.5,-1.8,-0.7);

glutSolidTorus(0.3,0.4,20,40);

glDisable(GL\_COLOR\_MATERIAL);

//banbelakangkiri

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(-5,0.001,-0.1);

glutSolidTorus(0.3,0.4,20,40);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(-1.5,0.001,-0.01);

glutSolidTorus(0.3,0.4,20,40);

glDisable(GL\_COLOR\_MATERIAL);

//banbelakangkanan

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(-0.1,0.001,2.9);

glutSolidTorus(0.3,0.4,20,40);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(1.5,0.001,0.1);

glutSolidTorus(0.3,0.4,20,40);

glDisable(GL\_COLOR\_MATERIAL);

//bandepankanan

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(5,0.001,0.1);

glutSolidTorus(0.3,0.4,20,40);

glDisable(GL\_COLOR\_MATERIAL);

//jensampingknan

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(-6.2,2,-0.6);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(1.2,0.01,0.01);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(1.2,0.01,0.01);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(1.2,0.01,0.01);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(1.2,0.01,0.01);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

//pintuknn

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(1.2,0.01,0.01);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(0,-0.8,0.001);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

//pintukiri

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(0,0,-2.1);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(0,0.8,0);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

//jendelsmpingkiri

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(-1.2,0,0);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(-1.2,0,0);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(-1.2,0,0);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(-1.2,0,0);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0,0,1);

glRotatef(1,0,0,0);

glTranslatef(-1.2,0,0);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

//BANGUNAN /////////////////////

//tembok bangunan kanan

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(10,1,-9);

glColor3f(2,2,2);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(2,2,2);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(2,2,2);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(2,2,2);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(2,2,2);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

//btingkat2

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,2,0);

glColor3f(2,2,2);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(2,2,2);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(2,2,2);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(2,2,2);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(2,2,2);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

//Btingkat3

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,2,0);

glColor3f(2,2,2);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(2,2,2);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(2,2,2);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

//Btingkat4

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,2,0);

glColor3f(2,2,2);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

//jendla

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,0,0.9);

glColor3f(0,0,0);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,-2,0);

glColor3f(0,0,0);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(0,0,0);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(4,0,0);

glColor3f(0,0,0);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,-2,0);

glColor3f(0,0,0);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(0,0,0);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-4,0,0);

glColor3f(0,0,0);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(0,0,0);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(0,0,0);

glutSolidCube(1);

glDisable(GL\_COLOR\_MATERIAL);

//banguna2t1

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(18,-2,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,0,-2);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

//tinngkt2

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,2,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,0,2);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,2,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,0,-2);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

//tingkat3

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,2,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,0,2);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,2,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,0,-2);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

//tingkat4

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,2,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,0,2);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,2,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,0,-2);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

//tingkat5

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,2,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,0,2);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,2,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,0,-2);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

//jendela

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(1,-1,3);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(3.5,0,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,-3,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-3.5,0,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,-3,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(3.5,0,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,-3,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-3.5,0,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,-3,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(3.5,0,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

//jalan

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glColor3f(0.0f,0.0f,0.0f);

glTranslatef(-40,-6.5,20);

glBegin(GL\_QUADS);

glVertex3f(15.0f,0.0f,0.0f);

glVertex3f(75.0f,0.0f,0.0f);

glVertex3f(75.0f,0.0f,-15.0f);

glVertex3f(15.0f,0.0f,-15.0f);

glDisable(GL\_COLOR\_MATERIAL);

glEnd();

//garisjalam

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(1,0.3,-5);

glColor3f(1.0f,1.0f,1.0f);

glBegin(GL\_QUADS);

glVertex3f(15.0f,0.0f,0.0f);

glVertex3f(30.0f,0.0f,0.0f);

glVertex3f(30.0f,0.0f,-5.0f);

glVertex3f(15.0f,0.0f,-5.0f);

glDisable(GL\_COLOR\_MATERIAL);

glEnd();

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(20,0,0);

glColor3f(1.0f,1.0f,1.0f);

glBegin(GL\_QUADS);

glVertex3f(15.0f,0.0f,0.0f);

glVertex3f(30.0f,0.0f,0.0f);

glVertex3f(30.0f,0.0f,-5.0f);

glVertex3f(15.0f,0.0f,-5.0f);

glDisable(GL\_COLOR\_MATERIAL);

glEnd();

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(20,0,0);

glColor3f(1.0f,1.0f,1.0f);

glBegin(GL\_QUADS);

glVertex3f(15.0f,0.0f,0.0f);

glVertex3f(30.0f,0.0f,0.0f);

glVertex3f(30.0f,0.0f,-5.0f);

glVertex3f(15.0f,0.0f,-5.0f);

glDisable(GL\_COLOR\_MATERIAL);

glEnd();

//gedung3

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(8,2,-17);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,2,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

//tingkat3

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,2,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,2,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-2,0,0);

glColor3f(1,1,1);

glutSolidCube(2.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(1,0,1);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(3,0,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(3,0,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(3,0,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(3,0,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(3,0,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(0,-3,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-3,0,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-3,0,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-3,0,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-3,0,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glPushMatrix();

glEnable(GL\_COLOR\_MATERIAL);

glTranslatef(-3,0,0);

glColor3f(0,0,0);

glutSolidCube(1.5);

glDisable(GL\_COLOR\_MATERIAL);

glutSwapBuffers();

}

void resize(int w1, int h1){

glViewport(0,0,w1,h1);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(45.0,(float) w1/(float) h1, 1.0,300.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

void timer(int value){

glutPostRedisplay();

glutTimerFunc(50,timer,0);

}

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

glutInit(&argc, argv);

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

glutInitWindowPosition(140,60);

glutInitWindowSize(1024,700);

glutCreateWindow("TUGAS BESAR GRAFIK KOMPUTER - BIS");

glutDisplayFunc(renderScene);

\_terrain = loadTerrain("1.bmp",13);

glutReshapeFunc(resize);

glutKeyboardFunc(myKeyboard);

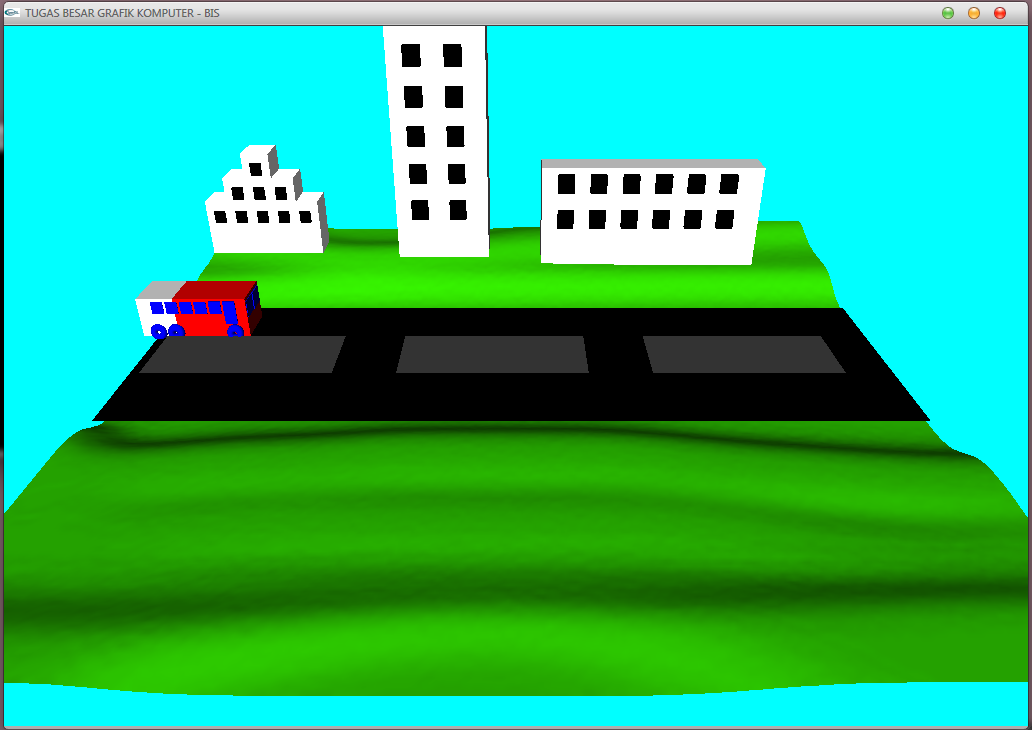
glutTimerFunc(1,timer,0);

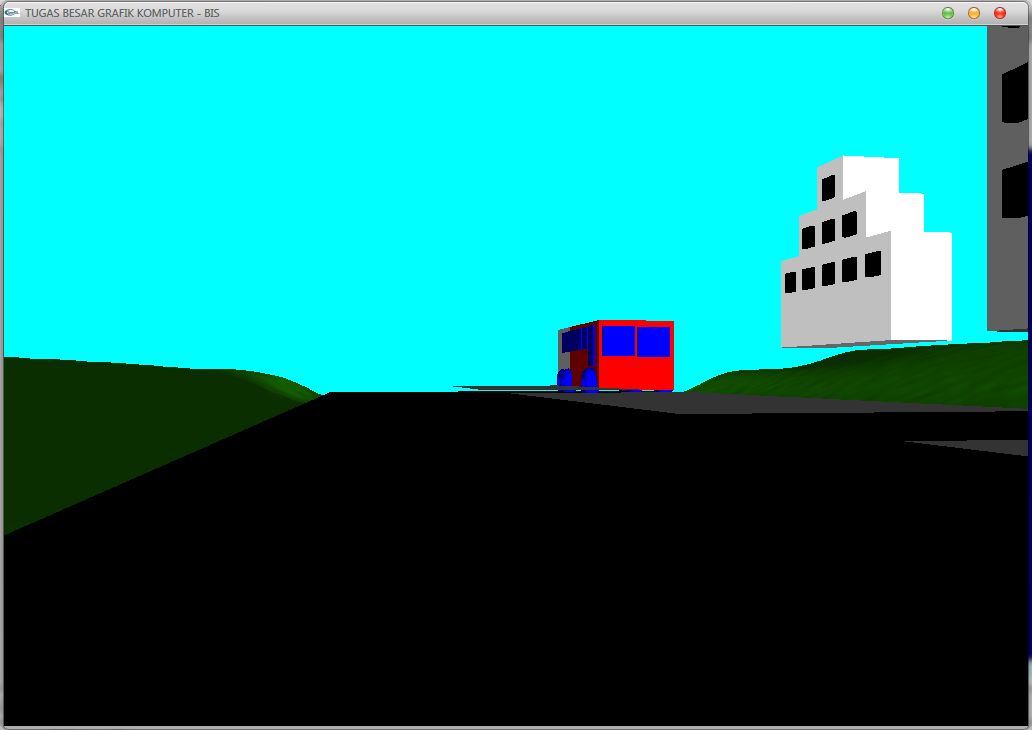
init();

glutMainLoop();

}

**PRINTSCREEN**

****Gambar1. Bis 3D

****Gambar2. Bis 3D