**Tugas Besar Grafika Komputer**

**Pembuatan Objek Dengan OpenGL**

Diajukan untuk Memenuhi Tugas Grafika Komputer

Fakultas Teknik dan Ilmu Komputer

Universitas Komputer Indonesia

Kelompok 7 – IF-10

**Reza Pratama Gunawan 10109437**

**Saksono Braja 10109415**

**Nurul Aini Kastella 10109436**

**Moch Irvan Gustiawan 10109425**



**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

**UNIVERSTAS KOMPUTER INDONESIA**

**2013**

# Daftar Isi

[Daftar Isi i](#_Toc297346811)

[BAB I PENDAHULUAN 1](#_Toc297346812)

[1. Pendahuluan 1](#_Toc297346812)

[2. Rumusan Masalah 2](#_Toc297346812)

[3. Batasan Masalah 2](#_Toc297346812)

[BAB II ISI 3](#_Toc297346813)

[2.1 Tema Program 3](#_Toc297346814)

[2.2 Deskripsi Objek Program 3](#_Toc297346815)

[2.3 Bahasa Pemrograman yang digunakan 4](#_Toc297346816)

[2.4 Progres 4](#_Toc297346817)

[2.5 Perencanaan Kerja 4](#_Toc297346818)

[2.6 Pembagian Kerja 4](#_Toc297346818)

[2.7 Tampilan Program 5](#_Toc297346819)

[2.8 Interaksi objek dengan mouse dan keyboard 7](#_Toc297346820)

[BAB III KESIMPULAN DAN SARAN 8](#_Toc297346821)

[3.1 Kesimpulan 8](#_Toc297346822)

[3.2 Saran 8](#_Toc297346823)

[Lampiran 9](#_Toc297346824)

[Daftar Pustaka 31](#_Toc297346825)

# 

# PENDAHULUAN

# Perkembangan Dunia desain dari waktu ke waktu terus mengalami peningkatan dan perkembangan, terutama dalam desain 3D. Hal tersebut didukung dengan munculnya berbagai macam kreatifitas baru dalam pembuatan desain 3D. Seperti dalam bidang game, arsitektur, film, animasi dan masih banyak lagi pengaruh desain dan grafik dalam kehidupan sehari-hari.

# Berdasarkan tingkat kesulitan yang dihadapi dan efektifitas yang diperlukan banyak sekali tools yang dapat digunakan, salah satunya adalah OpenGL. OpenGL adalah graphic library yang sebagian bersifat open source dan dipakai pada banyak platform (windows, linux) serta dapat digunakan pada berbagai jenis compiler.

# Berdasarkan hal tersebut kami mencoba untuk mendesain sebuah ruangan kamar yang berisikan objek-objek dalam bentuk 3D menggunakan OpenGL. Adapun alasan kami membuat desain sebuah ruang kamar adalah karena ruang kamar merupakan salah satu tempat bagi manusia untuk berpikir, berinspirasi, berkreatifitas, belajar, beristirahat dan melakukan kegiatan penting lainnya. Adapun objek-objek yang ada dalam desain kami adalah objek-objek yang sebagaimana umunya ada didalam kamar, seperti, Sofa, tempat tidur, meja, kursi, lampu, pintu dan jendela. Dengan kata lain kami memberi judul tugas ini “PEMODELAN RUANGAN 3D”.

**RUMUSAN MASALAH**

Adapun rumusan masalah dalam “Pemodelan Ruangan 3D” adalah sebagai berikut :

1. Bagaimana mendesain ruang 3D?
2. Bagaimana mengunakan OpenGL?
3. Bagaimana membuat objek 3D dengan menggunakan vertex pada OpenGL?
4. Bagaimana mengimplementasikan ilmu - ilmu yang kami dapat dari mata kuliah Grafika Komputer?

# BATASAN MASALAH

Agar pembahasan masalah tidak menyimpang dari dari pokok bahasan, maka batasan masalah dari “Pemodelan Ruangan 3D” adalah sebagai berikut :

1. Objek - objek pada ruangan adalah meja, tempat tidur, bantal, kursi, lampu, pintu, jendela sofa,meja makan dan lemari.
2. Objek dalam bentuk 3D.
3. Aplikasi berjalan di desktop.

**LIST OF OBJECK**

Adapun rincian sederhana dari objek – objek yang akan didesain pada “Pemodelan Ruangan 3D” adalah sebagai berikut:

1. Meja berbentuk persegi dengan empat kaki meja terletak ditengah ruangan.
2. Kursi berupa sofa yang berada di tengah ruangan
3. Tempat tidur berbentuk persegi panjang dua lapis dengan bantal. Berada pada pojok ruangan.
4. Lampu dengan tinggi setengah dari tinggi ruangan berbentuk tiang dengan ujung atasnya berupa kerucut yang menghadap kebawah.
5. Pintu berbentuk persegi panjang yang dengan sebuah lingkaran yang berfungsi sebagai gagang pintu.
6. Jendela berbentuk persegi dengan bingkai list jendela.
7. Bantal berbentuk persegi.
8. Tangga berbentuk persegi panjang yang berbaris.
9. Meja makan berbentuk lingkaran dengan empat kaki.
10. Pohon berbentuk kerucut .
11. Kolam berbentuk persegi panjang.
12. Pagar berbentuk persegi panjang.

# ISI

## 2.1 Tema Program

Tema dari program pembuatan objek menggunakan openGL ini adalah pembuatan objek yang terdiri dari beberapa objek dan objek nantinya membentuk sebuah ruangan 3D.

## 2.2 Deskripsi Objek Program

Objek utama pada program ini memodelkan sebuah ruangan 3D, yang memiliki konsep diantaranya terdapat kubus terdiri dinding dan lantai,ruangan ini nantinya akan dilengkapi objek seperti :

Meja : dibuat dari glutsolidcube yang sehingga menjadi

sebuah meja dengan kaki meja

Tempat Tidur : dibuat dari glutsolidcube menjadi objek tempat tidur

dengan penambahan kaki tempat tidur.

Lemari : dibuat dari glutsolidcube menjadi sebuah objek

lemari.

Meja Makan : dibuat dari glutSolidSphere menjadi objek meja makan

dengan penambahan kaki meja menggunakan

glutsolidcube.

Sofa : dibuat dari glutsolidcube menjadi sebuah objek menjadi

sebuah sofa.

Tangga : dibuat dari glutsolidcube menjadi sebuah objek tangga

Lampu : dibuat dari glutsolidsphere menjadi sebuah objek lampu

Kolam : dibuat menggunakan texture dengan tepian kolam terbuat

dari glutsolidcube.

Pohon : dibuat dari glutsolidcone dan digabungkan dengan glutsolidsphere

Sehinngga menjadi objek pohon.

Pagar : dibuat dari

Bantal : dibuat dari glutsolidcube menjadi sebuah objek bantal.

Interaksi yang akan digunakan untuk berinteraksi dengan aplikasi menggunakan tombol-tombol arah (kiri,kanan,atas,bawah) yang terdapat pada keyboard dan beberapa tombol karakter lainnya. Fungsi dari interaksi tersebut diantaranya sebagai navigasi untuk melihat ruangan 3D dari berbagai sudut pandang.

**2.3 Bahasa Pemrograman yang digunakan**

Bahasa pemrograman yang digunakan adalah C OpenGL dengan menggunakan software developer eclipse serta media penyimpanan github.

## 2.4 Progress

Tahapan pembuatan program :

1. Merancang dan menetapkan konsep program yang akan dibuat.
2. Pengumpulan bahan dan referensi dari berbagai media.
3. Analisis dan prediksi pembuatan design.
4. Pembuatan program.
5. Pembuatan laporan.

**2.5** Perencanaan Kerja

Kinerja dalam proses pembuatan program ini harus dilakukan semaksimal mungkin agar mendapatkan hasil yang baik.

Berikut langkah-langkah yang dilalui dalam perencanaan kerja:

1. Tahapan perancangan dan penetapan konsep :

* Mencari referensi.
* Menentukan konsep yang sesuai dilihat dari kelengkapan referensi yang ada.

1. Tahapan perancangan dan kebutuhan

* Menganalisis akan kebutuhan objek yang dibutuhkan.
* Menyiapkan dan membuat objek-objek satu persatu
* Merancang struktur letak desain objek.
* Membuat beberapa interkasi menarik.

1. Pembuatan Laporan.

## 2.6 Pembagian Kerja

Pembagian pekerjaan dalam pengerjaan tugas ini dibagi sedemikian mungkin agar mendapatkan hasil yang memuaskan.

Berikut pembagian kerja :

Reza Pratama Gunawan 🡪 Penentuan konsep, pembuatan aplikasi,

pengerjaan design, pembuatan laporan

Saksono Braja 🡪 Penentuan konsep Pembuatan aplikasi

pengerjaan design, pembuatan laporan

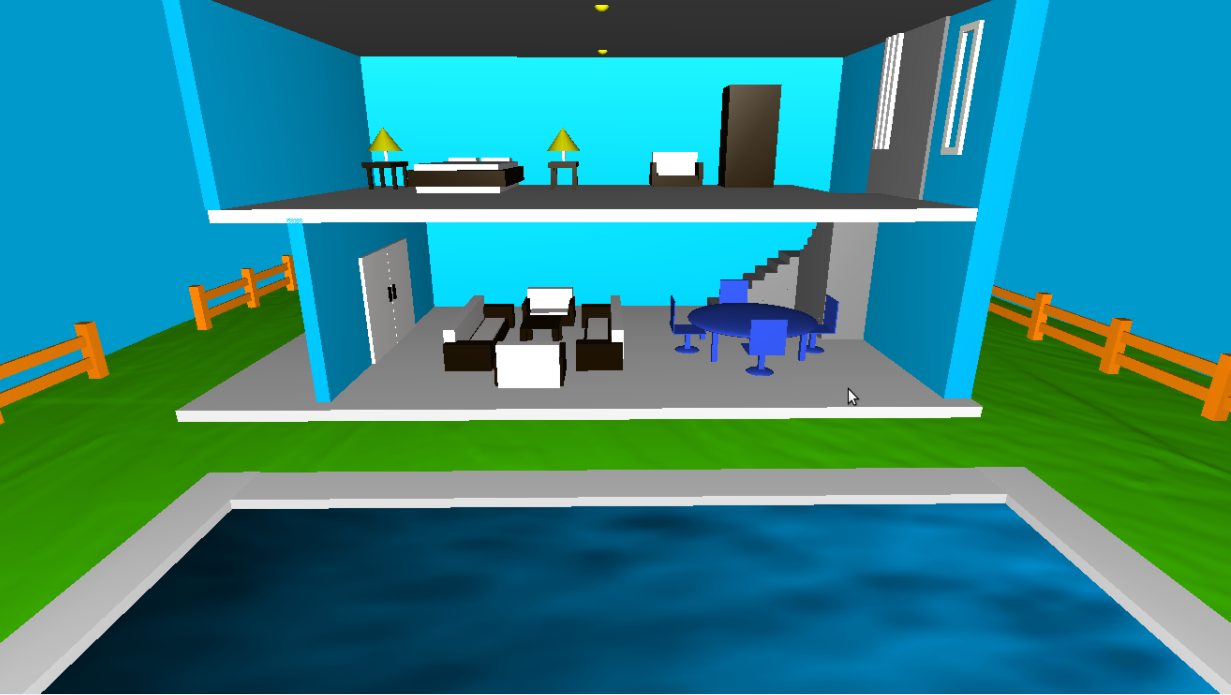
Nurul Aini Kastella 🡪 Pembuatan laporan

Moch Irvan Gustiawan 🡪 Pembuatan laporan

**2.7** Tampilan Program



**Gambar 1** tampak samping kiri atas



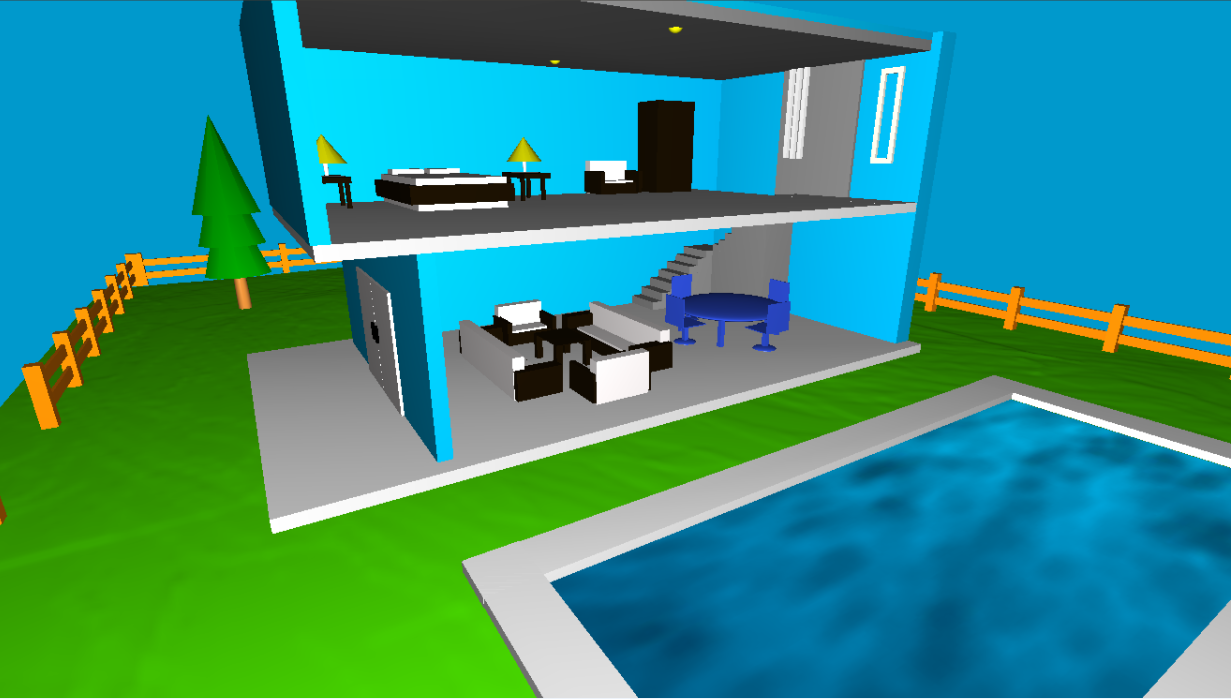
**Gambar 2** Tampilan tampak samping kanan



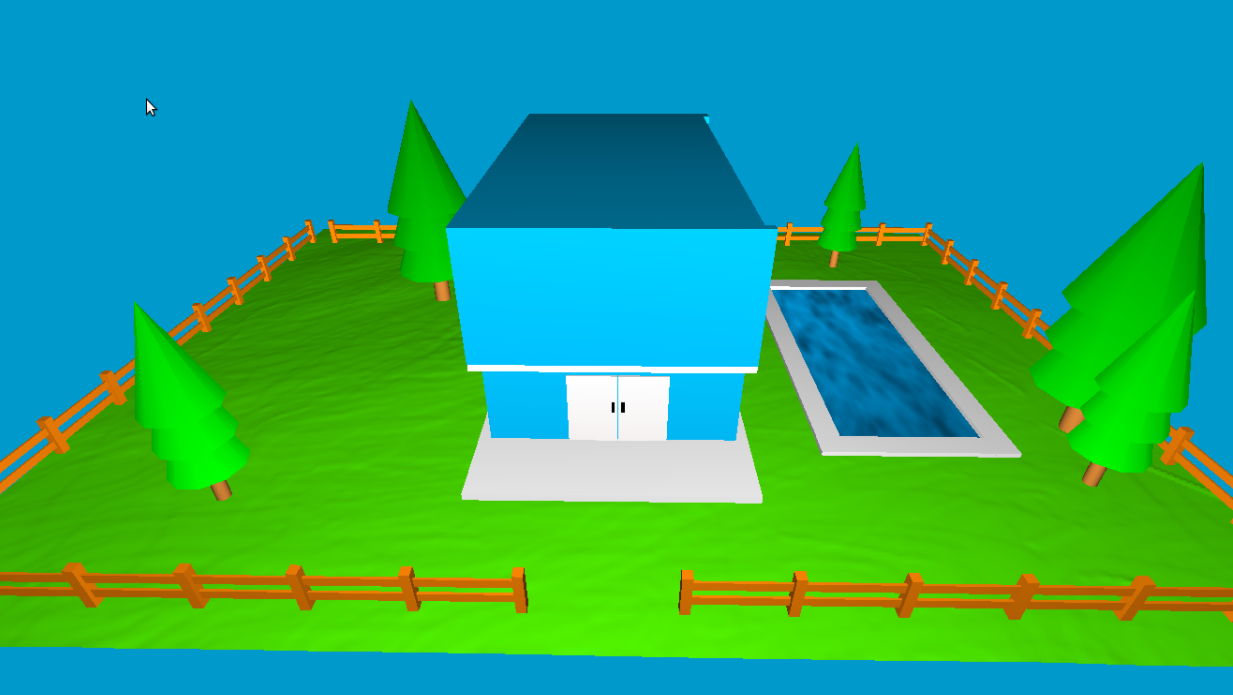
**Gambar 3** Tampilan tampak dari kiri bawah



**Gambar 4** Tampilan tampak dari samping kiri atas



**Gambar 5** Tampilan tampak dari samping kiri



**Gambar 6** Tampilan tampak dari depan

**2.8 Interaksi Objek Dengan Mouse dan Keyboard**

Objek ini dapat berinteraksi dengan menggunakan mouse dan keyboard. Pada interaksi objek menggunakan mouse, kamera dapat dinavigasikan dengan gerakan mouse yang di geser untuk melihat visualisasi objek secara memutar dari berbagai sudut pandang. Berikut merupakan table interaksi yang dapat digunakan pada tombol keyboard:

|  |  |
| --- | --- |
| **Key Board** | |
| Tombol | Fungsi |
| W | Menggerakkan visualisasi kamera ke depan |
| A | Menggerakkan visualisasi kamera ke kiri |
| S | Menggerakkan visualisasi kamera ke belakang |
| D | Menggerakkan visualisasi kamera ke kanan |
| Panah Atas | Menggerakkan visualisasi kamera ke depan |
| Panah Bawah | Menggerakkan visualisasi kamera ke kiri |
| Panah Kanan | Menggerakkan visualisasi kamera ke belakang |
| Panah Kiri | Menggerakkan visualisasi kamera ke kanan |

Tabel 1. Tabel Interaksi

# BAB III

# KESIMPULAN DAN SARAN

## 3.1 Kesimpulan

## Dari pembahasan dapat diambil kesimpulan sebagai berikut :

## Untuk memahami secara mendalam transformasi didalam grafik komputer sangatlah penting menguasai ilmu matematik.

## Dunia desain dari waktu ke waktu terus mengalami peningkatan dan perkembangan, terutama dalam desain 3D. Hal tersebut didukung dengan munculnya berbagai macam kreatifitas baru dalam pembuatan desain 3D. Seperti dalam bidang game, arsitektur, film, animasi dan masih banyak lagi pengaruh desain dan grafik dalam kehidupan sehari-hari.

## 

## 3.2 Saran

Setelah melihat uraian di atas, penyusun mempunyai saran.

1. Harus lebih mencintai dunia gambar/grafik karena akan lebih mudah memahami permasalahan yang ada dalam pembuatan grafik computer.
2. Bila ingin membuat suatu gambar/grafik, pilihlah program aplikasi yang lebih bagus menurut anda, lebih kompleks, dan yang khusus menangani pembuatan gambar/grafik serta yang mengikuti perkembangan jaman .

# Lampiran

Source Code Program

|  |
| --- |
| /\*  \* main.cpp  \*  \* Created on: Jun 10, 2013  \* Author: Reza  \*/  #include <stdio.h>  #include <stdlib.h>  #include <string.h>  #include <math.h>  #ifdef \_\_APPLE\_\_  #include <OpenGL/OpenGL.h>  #include <GLUT/glut.h>  #else  #include <GL/glut.h>  #include <GL/glu.h>  #include <GL/gl.h>  #include "imageloader.h"  #include "vec3f.h"  #endif  static GLfloat spin = 0.0;  float angle = 0;  using namespace std;  float lastx, lasty;  GLint stencilBits;  static int camx = 100;  static int camy = 50;  static int camz = 50;  float rot = 0;  class Rumput {  private:  int w; //lebar  int l; //panjang  float\*\* t; //tinggi  Vec3f\*\* acx;  bool komposisi;  public:  Rumput(int w2, int l2) {  w = w2;  l = l2;  t = new float\*[l];  for (int i = 0; i < l; i++) {  t[i] = new float[w];}  acx = new Vec3f\*[l];  for (int i = 0; i < l; i++) {  acx[i] = new Vec3f[w];}  komposisi = false;}  ~Rumput() {  for (int i = 0; i < l; i++) {  delete[] t[i];}  delete[] t;  for (int i = 0; i < l; i++) {  delete[] acx[i];}  delete[] acx;}  int width() {  return w;}  int length() {  return l;}  //Mengatur tinggi (x, z) terhadap y  void setHeight(int x, int z, float y) {  t[z][x] = y;  komposisi = false;}  //Mengembalikan nilai tinggi (x, z)  float getHeight(int x, int z) {  return t[z][x];}  //Menghitung var acx, jika belum dihitung  void computeNormals() {  if (komposisi) {  return;}  //Menghitung var acx  Vec3f\*\* acx2 = new Vec3f\*[l];  for (int i = 0; i < l; i++) {  acx2[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, t[z - 1][x] - t[z][x], -1.0f);}  Vec3f in;  if (z < l - 1) {  in = Vec3f(0.0f, t[z + 1][x] - t[z][x], 1.0f);}  Vec3f left;  if (x > 0) {  left = Vec3f(-1.0f, t[z][x - 1] - t[z][x], 0.0f);}  Vec3f right;  if (x < w - 1) {  right = Vec3f(1.0f, t[z][x + 1] - t[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();}  acx2[z][x] = sum;}}  // memperhalus var acx  const float FALLOUT\_RATIO = 0.5f;  for (int z = 0; z < l; z++) {  for (int x = 0; x < w; x++) {  Vec3f sum = acx2[z][x];  if (x > 0) {  sum += acx2[z][x - 1] \* FALLOUT\_RATIO;}  if (x < w - 1) {  sum += acx2[z][x + 1] \* FALLOUT\_RATIO;}  if (z > 0) {  sum += acx2[z - 1][x] \* FALLOUT\_RATIO;}  if (z < l - 1) {  sum += acx2[z + 1][x] \* FALLOUT\_RATIO;}  if (sum.magnitude() == 0) {  sum = Vec3f(0.0f, 1.0f, 0.0f);}  acx[z][x] = sum;}}  for (int i = 0; i < l; i++) {  delete[] acx2[i];  }  delete[] acx2;  komposisi = true;  }  //Mengembalikan nilai (x, z)  Vec3f getNormal(int x, int z) {  if (!komposisi) {  computeNormals();  }  return acx[z][x];  }  };  //Tutup Class  void initRendering() { // Inisialisasi  //glenable : Fungsi mengaktifkan atau menonaktifkan kemampuan OpenGL.  glEnable(GL\_DEPTH\_TEST);//Melakukan perbandingan kedalaman dan memperbarui kedalaman  glEnable(GL\_COLOR\_MATERIAL);//Fungsi menyebabkan warna bahan untuk melacak warna saat ini  glEnable(GL\_LIGHTING);//  glEnable(GL\_LIGHT0);//Fungsi mengembalikan sumber cahaya nilai parameter.  glEnable(GL\_NORMALIZE);//  glShadeModel(GL\_SMOOTH);//Fungsi memilih shading datar atau halus.  }  // Loads Rumput berdasarkan tinggi. Tinggi rumput dihitung dari jarak  // -tinggi / 2 kepada tinggi / 2.  // Loads terain di procedure inisialisasi  Rumput\* loadRumput(const char\* filename, float height) {  Image\* image = loadBMP(filename);  Rumput\* t = new Rumput(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;  // Buat tipe data terain  Rumput\* \_Rumput;  Rumput\* \_RumputTanah;  Rumput\* \_RumputAir;  const GLfloat light\_ambient[] = { 0.3f, 0.3f, 0.3f, 1.0f }; //mengatur  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 };  void load\_BMP\_texture(char \*filename) {  FILE \*file;  short int bpp;  short int planes;  long size;  unsigned int texture;  long imwidth;  long imheight;  char \*imdata;  file = fopen(filename, "rb");  fseek(file, 18, SEEK\_CUR);  fread(&imwidth, 4, 1, file);  fread(&imheight, 4, 1, file);  size = imwidth \* imheight \* 3;  fread(&bpp, 2, 1, file);  fread(&planes, 2, 1, file);  fseek(file, 24, SEEK\_CUR);  imdata = (char \*)malloc(size);  fread(imdata, size, 1, file);  char temp;  for(long i = 0; i < size; i+=3){  temp = imdata[i];  imdata[i] = imdata[i+2];  imdata[i+2] = temp;  }  fclose(file);  glGenTextures(1, &texture); // then we need to tell OpenGL that we are generating a texture  glBindTexture(GL\_TEXTURE\_2D, texture); // now we bind the texture that we are working with  glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_S, GL\_REPEAT);  glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_T, GL\_REPEAT);  glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);  glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);  glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_MODULATE);  glTexImage2D(GL\_TEXTURE\_2D, 0, GL\_RGB, imwidth, imheight, 0, GL\_RGB, GL\_UNSIGNED\_BYTE, imdata);  free(imdata); // free the texture  }  void cleanup() {  delete \_Rumput;  delete \_RumputTanah;}  void displayRumput(Rumput \*Rumput, GLfloat r, GLfloat g, GLfloat b) {  float scale = 350.0f / max(Rumput->width() - 1, Rumput->length() - 1);  glScalef(scale, scale, scale);  glTranslatef(-(float) (Rumput->width() - 1) / 2, 0.0f,  -(float) (Rumput->length() - 1) / 2);  glColor3f(r, g, b);  for (int z = 0; z < Rumput->length() - 1; z++) {  //Makes OpenGL draw a triangle at every three consecutive vertices  glBegin(GL\_TRIANGLE\_STRIP);//Menggambar segitiga  for (int x = 0; x < Rumput->width(); x++) {  Vec3f normal = Rumput->getNormal(x, z);  glNormal3f(normal[0], normal[1], normal[2]);  glVertex3f(x, Rumput->getHeight(x, z), z);  normal = Rumput->getNormal(x, z + 1);  glNormal3f(normal[0], normal[1], normal[2]);  glVertex3f(x, Rumput->getHeight(x, z + 1), z + 1);  }  glEnd();//fungsi membatasi simpul dari primitif atau kelompok seperti primitif  }  }  void tempatTidur(){  glPushMatrix();//Membuat baris kode diantaranya menjadi tidak berlaku untuk bagian luar.  glTranslatef(-15.0,-8.5,-12.0);//digunakan untuk merubah titik tengah sumbu koordinat  glScalef(6.0,1.0,9.0);//Skalasi merupakan bentuk transformasi yang dapat mengubah ukuran (besar-kecil) suatu objek.  glColor3f(0.0980, 0.0608, 0.0077);//Mengatur warna berdasarkan warna desimal  glutSolidCube(1.5);  glPopMatrix();//Membuat baris kode diantaranya menjadi tidak berlaku untuk bagian luar.  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glPushMatrix();  glTranslatef(-15.0,-8.5,-12.0);  glColor3ub (255, 250, 250);  glScalef(6.0,2.0,10.0);  glutSolidCube(1.3);  glPopMatrix();  // bantal kiri  glPushMatrix();  glRotated(-90, 0.0, 1.0, 0.0);  glScaled(0.4, 0.1, 0.6);  glTranslatef(-43, -70, 28);  glColor3ub (255, 250, 250);  glutSolidCube(5.0);  glPopMatrix();  // bantal kanan  glPushMatrix();  glRotated(-90, 0.0, 1.0, 0.0);//Rotasi merupakan bentuk transformasi yang digunakan untuk memutar posisisuatu benda  glScaled(0.4, 0.1, 0.6);//fungsi kalikan matriks saat ini dengan matriks skala umum  glTranslatef(-43, -70, 22);  glColor3ub (255, 250, 250);  glutSolidCube(5.0);  glPopMatrix();//  }  void sofaPanjang(){  // bawah sofa  glRotated(-90, 0.0, 1.0, 0.0);  glPushMatrix();  glScaled(1.5, 0.5, 5);  glTranslatef(1, -17, 0);  glColor3f(0.0980, 0.0608, 0.0077);  glutSolidCube(5.0);  glPopMatrix();  // senderan sofa  glRotated(-90, 0.0, 1.0, 0.0);  glPushMatrix();  glScaled(5, 1.5, 0.5);  glTranslatef(0, -4, -11);  glColor3ub(255, 250, 250);  glutSolidCube(5.0);  glPopMatrix();  // sisi kanan sofa  glRotated(-90, 0.0, 1.0, 0.0);  glPushMatrix();  glScaled(2, 1, 0.5);  glTranslatef(-0.8, -7, 25);  glColor3f(0.0980, 0.0608, 0.0077);  glutSolidCube(5.0);  glPopMatrix();  //sisi kiri sofa  glPushMatrix();  glScaled(2, 1, 0.5);  glTranslatef(-0.8, -7, -25);  glColor3f(0.0980, 0.0608, 0.0077);  glutSolidCube(5.0);  glPopMatrix();  // bantal sofa  glPushMatrix();  glScaled(1, 0.1, 4);  glTranslatef(-1, -70, 0);  glColor3ub(255, 250, 250);  glutSolidCube(5.0);  glPopMatrix();  }  void pintuUtama(){  glPushMatrix();  glScaled(3, 5, 0.8);  glTranslatef(3, 2.5, 76);  glColor3ub (255, 250, 250);  glutSolidCube(5.0);  glPushMatrix();  glTranslatef(-5.055, 0, 0);  glColor3ub (255, 250, 250);  glutSolidCube(5.0);  glPopMatrix();  glPopMatrix();  glPushMatrix();  glScaled(0.2, 0.7, 1);  glTranslatef(0, 20, 60.88);  glColor3ub(000, 000, 000);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glScaled(0.2, 0.7, 1);  glTranslatef(15, 20, 60.88);  glColor3ub(000, 000, 000);  glutSolidCube(5.0);  glPopMatrix();  }  void tangga(){  glRotated(-90, 0.0, 1.0, 0.0);  glPushMatrix();  glScaled(0.7, 0.5, 2.5);  glTranslatef(-30, 5, 12);  glColor3ub(138, 138, 138);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glScaled(0.7, 1.0, 2.5);  glTranslatef(-35, 3.8, 12);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glScaled(0.7, 1.5, 2.5);  glTranslatef(-40, 3.5, 12);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glScaled(0.7, 2, 2.5);  glTranslatef(-45, 3.2, 12);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glScaled(0.7, 2.5, 2.5);  glTranslatef(-50, 3, 12);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glScaled(0.7, 3, 2.5);  glTranslatef(-55, 2.8, 12);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glScaled(2, 3.5, 2.5);  glTranslatef(-22.5, 2.7, 12);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glRotated(-90, 0.0, 1.0, 0.0);  glScaled(0.7, 4, 2.5);  glTranslatef(31, 2.6, 19.5);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glRotated(-90, 0.0, 1.0, 0.0);  glScaled(0.7, 4.5, 2.5);  glTranslatef(27, 2.4, 19.5);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glRotated(-90, 0.0, 1.0, 0.0);  glScaled(0.7, 5, 2.5);  glTranslatef(23, 2.4, 19.5);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glRotated(-90, 0.0, 1.0, 0.0);  glScaled(0.7, 5.5, 2.5);  glTranslatef(19, 2.4, 19.5);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glRotated(-90, 0.0, 1.0, 0.0);  glScaled(0.7, 6, 2.5);  glTranslatef(15, 2.4, 19.5);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glRotated(-90, 0.0, 1.0, 0.0);  glScaled(0.7, 6.5, 2.5);  glTranslatef(11, 2.4, 19.5);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glRotated(-90, 0.0, 1.0, 0.0);  glScaled(0.7, 7, 2.5);  glTranslatef(7, 2.4, 19.5);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glRotated(-90, 0.0, 1.0, 0.0);  glScaled(0.7, 7.5, 2.5);  glTranslatef(3, 2.4, 19.5);  glutSolidCube(5.0);  glPopMatrix();  }  void lampuMeja() {  glPushMatrix();  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3d(0.0, 1.0, 0.0);  glRotated(-90.0, 1.0, 0.0, 0.0);  glColor3ub(255, 255, 000);  glutSolidCone(15.0, 20.0, 15, -10);  glPopMatrix();  glPushMatrix();  glRotatef(90, 1, 0, 0);  glTranslatef(0.0, 0, -2);  glScaled(1.4, 2, 30);  glColor3ub(255, 250, 250);  glutSolidTorus(0.4, 0.4, 60, 80);  glPopMatrix();  glPushMatrix();  glColor3ub(46, 79, 219);  glTranslatef(0, -9.1, -0.05);  glScaled(2, 0.1, 2);  glColor3ub(255, 250, 250);  glutSolidSphere(6, 25, 25);  glPopMatrix();  }  void meja(void){  glPushMatrix();  glTranslatef(0.0,-5.0,0.0);  glColor3f(0.0980, 0.0608, 0.0077);  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 kursi(void) {  // Batang Tiang Kanan  glPushMatrix();  glScaled(0.06, 0.2, 0.06);  glTranslatef(43, 0, 380.5);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(1, 1, 1);  glutSolidCube(5.0);  glPopMatrix();  // Batang Tiang Kiri  glPushMatrix();  glScaled(0.06, 0.2, 0.06);  glTranslatef(3, 0, 380.5);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(1, 1, 1);  glutSolidCube(5.0);  glPopMatrix();  // Batang depan knan  glPushMatrix();  glScaled(0.06, 0.2, 0.06);  glTranslatef(43, 0, 390.5);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(1, 1, 1);  glutSolidCube(5.0);  glPopMatrix();  // Batang Depan Kiri  glPushMatrix();  glScaled(0.06, 0.2, 0.06);  glTranslatef(3, 0, 390.5);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(1, 1, 1);  glutSolidCube(5.0);  glPopMatrix();  // atas kursi  glPushMatrix();  glScaled(0.6, 0.05, 0.3);  glTranslatef(2.4, 8, 77);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(1.0000, 0.5252, 0.0157);  glutSolidCube(5.0);  glPopMatrix();  }  void bangunan(void) {  //lantai 1  glPushMatrix();  glScaled(1.115, 0.03, 2);  glTranslatef(0.0, 0, 0.38);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(255, 255, 255);  glutSolidCube(5.0);  glPopMatrix();  // lantai 2  glPushMatrix();  glScaled(1.015, 0.03, 1.64);  glTranslatef(0.0, 80, 0.57);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(255, 255, 255);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glScaled(0.5, 0.03, 0.15);  glTranslatef(2.56, 80, -23.6);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(255, 255, 255);  glutSolidCube(5.0);  glPopMatrix();  //lantai 3  glPushMatrix();  glScaled(0.95, 0.03, 1.8);  glTranslatef(0.0,160, 0.3);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(138, 138, 138);  glutSolidCube(5.0);  glPopMatrix();  //lapisan lantai 3  glPushMatrix();  glScaled(0.95, 0.02, 1.8);  glTranslatef(0.0,246, 0.3);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(000, 191, 255);  glutSolidCube(5.0);  glPopMatrix();  //Dinding Kiri Bawah  glPushMatrix();  glScaled(0.035, 0.5, 1.6);  glTranslatef(-70.0, 2.45, 0.0);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(000, 191, 255);  glutSolidCube(5.0);  glPopMatrix();  // //Dinding Kanan Bawah  // glPushMatrix();  // glScaled(0.035, 0.5, 1.6);  // glTranslatef(70.0, 2.45, 0.0);  // glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  // glColor3f(0.4613, 0.4627, 0.4174);  // glutSolidCube(5.0);  // glPopMatrix();  //  // //Dinding Kanan Atas  // glPushMatrix();  // glScaled(0.035, 0.5, 1.8);  // glTranslatef(70.0, 7.45, 0.3);  // glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  // glColor3ub(000, 191, 255);  // glutSolidCube(5.0);  // glPopMatrix();  //Dinding Kiri Atas  glPushMatrix();  glScaled(0.035, 0.5, 1.8);  glTranslatef(-70.0, 7.45, 0.3);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(000, 191, 255);  glutSolidCube(5.0);  glPopMatrix();  //Dinding Belakang bawah  glPushMatrix();  glScaled(1.015, 0.5, 0.07);  glTranslatef(0, 2.45,-58);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(000, 191, 255);  glutSolidCube(5.0);  glPopMatrix();  //Dinding Belakang atas  glPushMatrix();  glScaled(1.015, 0.5, 0.07);  glTranslatef(0, 7.45,-58);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(000, 191, 255);  glutSolidCube(5.0);  glPopMatrix();  //Dinding Depan bawah  glPushMatrix();  glScaled(1.015, 0.5, 0.035);  glTranslatef(0, 2.25,116);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(000, 191, 255);  glutSolidCube(5.0);  glPopMatrix();  //Dinding Depan atas  glPushMatrix();  glScaled(1.015, 0.5, 0.035);  glTranslatef(0, 7.45, 142);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(000, 191, 255);  glutSolidCube(5.0);  glPopMatrix();  // HIASAN DINDING  //background  glPushMatrix();  glScaled(0.35, 0.5, 0.035);  glTranslatef(1, 7.2,-112);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(138, 138, 138);  glutSolidCube(5.0);  glPopMatrix();  //strip 1  glPushMatrix();  glScaled(0.017,0.33, 0.035);  glTranslatef(-16.6, 12,-110);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub (255, 255 ,255);  glutSolidCube(5.0);  glPopMatrix();  //strip 2  glPushMatrix();  glScaled(0.017,0.33, 0.035);  glTranslatef(-6.6, 12,-110);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub (255, 255 ,255);  glutSolidCube(5.0);  glPopMatrix();  //strip 3  glPushMatrix();  glScaled(0.017,0.33, 0.035);  glTranslatef(3.6, 12,-110);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub (255, 255 ,255);  glutSolidCube(5.0);  glPopMatrix();  // JENDELA ATAS KANAN  // bingkai atas  glPushMatrix();  glScaled(0.08, 0.017, 0.035);  glTranslatef(22.5, 265,-112);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(255, 255, 240);  glutSolidCube(5.0);  glPopMatrix();  // bingkai bawah  glPushMatrix();  glScaled(0.08, 0.017, 0.035);  glTranslatef(22.5, 183,-112);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(255, 255, 240);  glutSolidCube(5.0);  glPopMatrix();  // bingkai kiri  glPushMatrix();  glScaled(0.017,0.28, 0.035);  glTranslatef(96.6, 13.5,-112);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(255, 255, 240);  glutSolidCube(5.0);  glPopMatrix();  // bingkai kanan  glPushMatrix();  glScaled(0.017,0.28, 0.035);  glTranslatef(115.1, 13.5,-112);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(255, 255, 240);  glutSolidCube(5.0);  glPopMatrix();  // MEJA KIRI  // tatakan meja  glPushMatrix();  glRotated(90, 1.0, 0.0, 0.0);  glScaled(0.98, 0.48, 0.08);  glTranslatef(-1.5, 9.0, -36);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(255, 222, 173);  glutSolidCube(1.0);  glPopMatrix();  // kaki meja kiri belakang  glPushMatrix();  glScaled(0.06, 0.5,0.06);  glTranslatef(-30, 5.2, 75.5);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(255, 222, 173);  glutSolidCube(1.0);  glPopMatrix();  // kaki meja kiri depan  glPushMatrix();  glScaled(0.06, 0.5,0.06);  glTranslatef(-20, 5.2, 75.5);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(255, 222, 173);  glutSolidCube(1.0);  glPopMatrix();  // kaki meja kanan depan  glPushMatrix();  glScaled(0.06, 0.5,0.06);  glTranslatef(-20, 5.2, 68.5);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(255, 222, 173);  glutSolidCube(1.0);  glPopMatrix();  // kaki meja kanan belakang  glPushMatrix();  glScaled(0.06, 0.5,0.06);  glTranslatef(-30, 5.2, 68.5);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(255, 222, 173);  glutSolidCube(1.0);  glPopMatrix();  // MEJA KIRI  // tatakan meja  glPushMatrix();  glRotated(90, 1.0, 0.0, 0.0);  glScaled(0.98, 0.48, 0.08);  glTranslatef(-1.5, 9.0, -36);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(0.0980, 0.0608, 0.0077);  glutSolidCube(1.0);  glPopMatrix();  // kaki meja kiri belakang  glPushMatrix();  glScaled(0.06, 0.5,0.06);  glTranslatef(-30, 5.2, 75.5);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glutSolidCube(1.0);  glPopMatrix();  // kaki meja kiri depan  glPushMatrix();  glScaled(0.06, 0.5,0.06);  glTranslatef(-20, 5.2, 75.5);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glutSolidCube(1.0);  glPopMatrix();  // kaki meja kanan depan  glPushMatrix();  glScaled(0.06, 0.5,0.06);  glTranslatef(-20, 5.2, 68.5);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glutSolidCube(1.0);  glPopMatrix();  // kaki meja kanan belakang  glPushMatrix();  glScaled(0.06, 0.5,0.06);  glTranslatef(-30, 5.2, 68.5);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glutSolidCube(1.0);  glPopMatrix();  // MEJA KIRI  // tatakan meja  glPushMatrix();  glRotated(90, 1.0, 0.0, 0.0);  glScaled(0.98, 0.48, 0.08);  glTranslatef(-1.50, 2.5, -36);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(0.0980, 0.0608, 0.0077);  glutSolidCube(1.0);  glPopMatrix();  // kaki meja kiri belakang  glPushMatrix();  glScaled(0.06, 0.5,0.06);  glTranslatef(-30, 5.2, 23);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glutSolidCube(1.0);  glPopMatrix();  // kaki meja kiri depan  glPushMatrix();  glScaled(0.06, 0.5,0.06);  glTranslatef(-20, 5.2, 23);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glutSolidCube(1.0);  glPopMatrix();  // kaki meja kanan depan  glPushMatrix();  glScaled(0.06, 0.5,0.06);  glTranslatef(-20, 5.2, 16.8);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glutSolidCube(1.0);  glPopMatrix();  // kaki meja kanan belakang  glPushMatrix();  glScaled(0.06, 0.5,0.06);  glTranslatef(-30, 5.2, 16.8);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glutSolidCube(1.0);  glPopMatrix();  // LEMARI  glPushMatrix();  glRotated(-90, 0.0, 1.0, 0.0);  glScaled(0.18, 0.35, 0.16);  glTranslatef(-12, 9.5, 13);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(0.0980, 0.0608, 0.0077);  glutSolidCube(5.0);  glPopMatrix();  // LAMPU ATAS KANAN  glPushMatrix();  glScaled(0.05, 0.05, 0.05);  glTranslatef(34.5, 95.4, 10);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE); //untuk memunculkan warna  glColor3ub(255, 255, 000);  glutSolidSphere(2.0,20,50);  glPopMatrix();  // LAMPU ATAS KIRI  glPushMatrix();  glScaled(0.05, 0.05, 0.05);  glTranslatef(-32.5, 95.4, 10);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(255, 255, 000);  glutSolidSphere(2.0,20,50);  glPopMatrix();  // LAMPU BAWAH KANAN  glPushMatrix();  glScaled(0.05, 0.05, 0.05);  glTranslatef(34.5, 47, 10);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE); //untuk memunculkan warna  glColor3ub(255, 255, 000);  glutSolidSphere(2.0,20,50);  glPopMatrix();  // LAMPU BAWAH KIRI  glPushMatrix();  glScaled(0.05, 0.05, 0.05);  glTranslatef(-32.5, 47, 10);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(255, 255, 000);  glutSolidSphere(2.0,20,50);  glPopMatrix();  }  void mejaMakan(void){  // lingkaran meja  glPushMatrix();  glColor3ub(46, 79, 219);  glTranslatef(0, 1.5, -1);  glScaled(1, 0.1, 1);  glutSolidSphere(6, 25, 25);  glPopMatrix();  // kaki meja kiri depan  glPushMatrix();  glScaled(0.1, 0.6, 0.1);  glTranslatef(33, 0, 30);  glColor3ub(46, 79, 219);  glutSolidCube(5.0);  glPopMatrix();  // kaki meja kiri belakang  glPushMatrix();  glScaled(0.1, 0.6, 0.1);  glTranslatef(-33, 0, 30);  glColor3ub(46, 79, 219);  glutSolidCube(5.0);  glPopMatrix();  // kaki meja kanan depan  glPushMatrix();  glScaled(0.1, 0.6, 0.1);  glTranslatef(33, 0, -50);  glColor3ub(46, 79, 219);  glutSolidCube(5.0);  glPopMatrix();  // kaki meja kanan belakang  glPushMatrix();  glScaled(0.1, 0.6, 0.1);  glTranslatef(-33, 0, -50);  glColor3ub(46, 79, 219);  glutSolidCube(5.0);  glPopMatrix();  }  void kursiMakan(void)  {  glPushMatrix();  glColor3ub(46, 79, 219);  glTranslatef(-5, -9.1, -0.05);  glScaled(0.2, 0.05, 0.2);  glutSolidSphere(6, 25, 25);  glPopMatrix();  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 pagar(){  glPushMatrix();  glScaled(0.2, 1 ,0.2);  glTranslatef(20, 0, 290);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(1.0000, 0.5252, 0.0157);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glScaled(0.2, 1 ,0.2);  glTranslatef(70, 0, 290);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(1.0000, 0.5252, 0.0157);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glScaled(0.2, 1 ,0.2);  glTranslatef(120, 0, 290);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(1.0000, 0.5252, 0.0157);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glScaled(0.2, 1 ,0.2);  glTranslatef(170, 0, 290);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(1.0000, 0.5252, 0.0157);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glScaled(0.2, 1 ,0.2);  glTranslatef(220, 0, 290);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(1.0000, 0.5252, 0.0157);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glScaled(0.2, 1 ,0.2);  glTranslatef(270, 0, 290);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(1.0000, 0.5252, 0.0157);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glScaled(0.2, 1 ,0.2);  glTranslatef(320, 0, 290);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(1.0000, 0.5252, 0.0157);  glutSolidCube(5.0);  glPopMatrix();  // papan memanjang  glPushMatrix();  glScaled(12, 0.2 ,0.05);  glTranslatef(2.8, 5, 1160);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(1.0000, 0.5252, 0.0157);  glutSolidCube(5.0);  glPopMatrix();  glPushMatrix();  glScaled(12, 0.2 ,0.05);  glTranslatef(2.8, -5, 1160);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3f(1.0000, 0.5252, 0.0157);  glutSolidCube(5.0);  glPopMatrix();  }  void pohonCemara(){  //batang  glColor3f(0.8, 0.5, 0.2);  glPushMatrix();  glScalef(0.2, 2, 0.2);  glutSolidSphere(1.0, 20, 16);  glPopMatrix();  //daun  glColor3f(0.0, 1.0, 0.0);  glPushMatrix();  glScalef(1, 1, 1.0);  glTranslatef(0, 1, 0);  glRotatef(270, 1, 0, 0);  glutSolidCone(1,3,10,1);  glPopMatrix();  glPushMatrix();  glScalef(1, 1, 1.0);  glTranslatef(0, 2, 0);  glRotatef(270, 1, 0, 0);  glutSolidCone(1,3,10,1);  glPopMatrix();  glPushMatrix();  glScalef(1, 1, 1.0);  glTranslatef(0, 3, 0);  glRotatef(270, 1, 0, 0);  glutSolidCone(1,3,10,1);  glPopMatrix();  }  void TepiKolam(){  //tepi kiri  glPushMatrix();  glScaled(0.1, 0.02, 1.65);  glTranslatef(45, 0, 0.2);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glColor3ub(255, 255, 255);  glutSolidCube(5.0);  glPopMatrix();  // tepi kanan  glPushMatrix();  glScaled(0.1, 0.02, 1.65);  glTranslatef(80, 0, 0.2);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glutSolidCube(5.0);  glPopMatrix();  // tepi depan  glPushMatrix();  glScaled(0.8, 0.02, 0.1);  glTranslatef(7.8, 0, 42);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glutSolidCube(5.0);  glPopMatrix();  // tepi belakang  glPushMatrix();  glScaled(0.8, 0.02, 0.1);  glTranslatef(7.8, 0, -35.5);  glColorMaterial(GL\_FRONT\_AND\_BACK, GL\_AMBIENT\_AND\_DIFFUSE);  glutSolidCube(5.0);  glPopMatrix();  }  unsigned int LoadTextureFromBmpFile(char \*filename);  void display(void) {  glClearStencil(0); //clear the stencil buffer  glClearDepth(1.0f);//Fungsi menentukan nilai yang jelas untuk kedalaman  glClearColor(0.0, 0.6, 0.8, 1);//mendefinisikan warna dari windows  glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT | GL\_STENCIL\_BUFFER\_BIT); //untuk membersihkan memori buffer warna atau memori buffer kedalaman dari keadaan sebelumnya.  glLoadIdentity();//untuk memanggil matriks terakhir yang disimpan.  gluLookAt(camx, camy, camz, 0.0, 0.0, 5.0, 0.0, 1.0, 0.0);//Fungsi mendefinisikan transformasi pandangan.  glPushMatrix();  glTranslatef(0,0,0);  glScalef(15, 15, 15);  bangunan(); //Bangunan  glPopMatrix();  glPushMatrix();  glTranslatef(0,0,0);  pintuUtama();//2 Pintu utama  glPopMatrix();  glPushMatrix();  glTranslatef(0,0,-5);  tangga();// Tangga lantai 1 dan 2  glPopMatrix();  glPushMatrix();  glTranslatef(0, 0.5, -20);  kursi();// Kursi  glPopMatrix();  glPushMatrix();  glScalef(2.5, 2.5, 2.5);  glTranslatef(3, 2, -10);  mejaMakan();// Meja makan  glPopMatrix();  glPushMatrix();  glScalef(2.5, 2.5, 2.5);  glRotatef(90, 0, 1, 0);  glTranslatef(10, 9.5, 3);  kursiMakan();// Kursi meja makan  glPushMatrix();  glRotatef(-180, 0, 1, 0);  glTranslatef(-2, 0, 0);  kursiMakan();  glPopMatrix();  glPushMatrix();  glRotatef(-90, 0, 1, 0);  glTranslatef(-2, 0, -1);  kursiMakan();  glPopMatrix();  glPushMatrix();  glRotatef(90, 0, 1, 0);  glTranslatef(-1, 0, 0.6);  kursiMakan();  glPopMatrix();  glPopMatrix();  glPushMatrix();  glScalef(2.5, 2.5, 2.5);  glRotatef(90, 0, 1, 0);  glTranslatef(-1.6, 24.7, 5);  tempatTidur();// Tempat tidur  glPopMatrix();  glPushMatrix();  glRotatef(90, 0, 1, 0);  glTranslatef(-22.5, 15, 5);  glScalef(1.5, 1.5, 1.5);  meja();// Meja ruang tamu  glPopMatrix();  glPushMatrix();  glTranslatef(5, 13, 35);  glScalef(1.3, 1.2, 1);  sofaPanjang();// Sofa panjang ruang tamu  glPushMatrix();  glRotatef(90, 0, 1, 0);  glTranslatef(0, 0, 25);  sofaPanjang();  glPopMatrix();  glPopMatrix();  glPushMatrix();  glRotatef(-90, 0, 1, 0);  glTranslatef(22.5, 13, 15);  glScalef(0.5, 1.2, 1);  sofaPanjang();//Sofa panjang ruang tamu  glPushMatrix();  glRotatef(90, 0, 1, 0);  glTranslatef(0, 0, 40);  sofaPanjang();  glPopMatrix();  glPushMatrix();  glRotatef(-90, 0, 1, 0);  glTranslatef(-70, 30, 12);  sofaPanjang();  glPopMatrix();  glPopMatrix();  glPushMatrix();  //lampu kiri  glScalef(0.3, 0.3, 0.3);  glTranslatef(-75, 155, 216);  lampuMeja(); //Lampu ruang tidur  // lampu kanan  glPushMatrix();  glTranslatef(0, 0, -156);  lampuMeja();// Lampu ruang tidur  glPopMatrix();  glPopMatrix();  glPushMatrix();  glScalef(2.5, 2.5, 2.5);  glTranslatef(3, 2, -10);  pagar();// pagar  glPushMatrix();  glTranslatef(-75, 0, 0);  pagar();// pagar  glPopMatrix();  glPushMatrix(); // pagar sisi kiri  glRotatef(-90, 0, 1, 0);  glTranslatef(-7, 0, 13);  pagar();// pagar  glPushMatrix();  glTranslatef(-30, 0, 0);  pagar();// pagar  glPopMatrix();  glPopMatrix();  glPushMatrix(); // pagar sisi kanan  glRotatef(-90, 0, 1, 0);  glTranslatef(-7, 0, -122);  pagar();// pagar  glPushMatrix();  glTranslatef(-30, 0, 0);  pagar();// pagar  glPopMatrix();  glPopMatrix();  glPushMatrix(); // pagar sisi belakang  glTranslatef(0, 0, -91);  pagar();// pagar  glPushMatrix();  glTranslatef(-60, 0, 0);  pagar();// pagar  glPopMatrix();  glPushMatrix();  glTranslatef(-70, 0, 0);  pagar();// pagar  glPopMatrix();  glPopMatrix();  glPopMatrix();  // POHON CEMARA  glPushMatrix();  glColor3f(1, 0, 0);  glTranslatef(-110, 0, 85);  glScalef(10, 10, 10);  pohonCemara();  glPopMatrix();  glPushMatrix();  glColor3f(1, 0, 0);  glTranslatef(140, 0, 75);  glScalef(10, 10, 10);  pohonCemara();  glPopMatrix();  glPushMatrix();  glColor3f(1, 0, 0);  glTranslatef(110, 0, -80);  glScalef(10, 10, 10);  pohonCemara();  glPopMatrix();  glPushMatrix();  glColor3f(1, 0, 0);  glTranslatef(-80, 0, -40);  glScalef(17, 14, 17);  pohonCemara();  glPopMatrix();  glPushMatrix();  glColor3f(1, 0, 0);  glTranslatef(150, 0, 50);  glScalef(17, 14, 17);  pohonCemara();  glPopMatrix();  //Kolam  glEnable(GL\_TEXTURE\_2D);  glColor3ub(000,191,255);  load\_BMP\_texture("water.bmp");  glBegin(GL\_POLYGON);  glTexCoord2f(0,0);  glVertex3f(120, -1, -50);  glTexCoord2f(1,0);  glVertex3f(70, -1, -50);  glTexCoord2f(1,1);  glVertex3f(70, -1, 60);  glTexCoord2f(0,1);  glVertex3f(120, -1, 60);  glEnd();  glDisable(GL\_TEXTURE\_2D);  // Tepi kolam  glPushMatrix();  glTranslatef(0,0,0);  glScalef(15, 15, 15);  TepiKolam(); //Bangunan  glPopMatrix();  glPushMatrix();  displayRumput(\_Rumput, 0.3f, 0.9f, 0.0f);// Memanggil class rumpunt dan inisialisasi warna  glPopMatrix();  glutSwapBuffers();  glFlush();  rot++;  angle++;  }  void init(void) {  glEnable(GL\_DEPTH\_TEST);  glEnable(GL\_LIGHTING);  glEnable(GL\_LIGHT0);  glDepthFunc(GL\_LESS);  glEnable(GL\_NORMALIZE);  glEnable(GL\_COLOR\_MATERIAL);  glDepthFunc(GL\_LEQUAL);  glShadeModel(GL\_SMOOTH);  glHint(GL\_PERSPECTIVE\_CORRECTION\_HINT, GL\_NICEST);//Fungsi set up perspektif matriks proyeksi.  glEnable(GL\_CULL\_FACE);  \_Rumput = loadRumput("heightmap.bmp", -20);  \_RumputTanah = loadRumput("heightmapTanah.bmp", -20);  \_RumputAir = loadRumput("heightmapAir.bmp", -20);  }  static void navigasi(int key, int x, int y) {  switch (key) {  case GLUT\_KEY\_HOME:  camy++;  break;  case GLUT\_KEY\_END:  camy--;  break;  case GLUT\_KEY\_UP:  camz--;  break;  case GLUT\_KEY\_DOWN:  camz++;  break;  case GLUT\_KEY\_RIGHT:  camx++;  break;  case GLUT\_KEY\_LEFT:  camx--;  break;  case GLUT\_KEY\_F1: {  glLightfv(GL\_LIGHT0, GL\_AMBIENT, light\_ambient);//Yaitu menyebabkan warna sekitar/ pantulan yang telah banyak.  glLightfv(GL\_LIGHT0, GL\_DIFFUSE, light\_diffuse);//Yaitu cahaya yang diterima oleh material akan menyebar kesegala arah.  glMaterialfv(GL\_FRONT, GL\_AMBIENT, mat\_ambient);  glMaterialfv(GL\_FRONT, GL\_DIFFUSE, mat\_diffuse);  }  ;  break;  case GLUT\_KEY\_F2: {  glLightfv(GL\_LIGHT0, GL\_AMBIENT, light\_ambient2);  glLightfv(GL\_LIGHT0, GL\_DIFFUSE, light\_diffuse2);  glMaterialfv(GL\_FRONT, GL\_AMBIENT, mat\_ambient);  glMaterialfv(GL\_FRONT, GL\_DIFFUSE, mat\_diffuse);  }  ;  break;  default:  break;  }  }  void keyboard(unsigned char key, int x, int y) {  if (key == 'd') {  spin = spin - 1;  if (spin > 360.0)  spin = spin - 360.0;  }  if (key == 'a') {  spin = spin + 1;  if (spin > 360.0)  spin = spin - 360.0;  }  if (key == 'q') {  camz++;  }  if (key == 'e') {  camz--;  }  if (key == 's') {  camy--;  }  if (key == 'w') {  camy++;  }  }  void reshape(int w, int h) {//Fungsi set up perspektif matriks proyeksi.  glViewport(0, 0, (GLsizei) w, (GLsizei) h);  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  gluPerspective(60, (GLfloat) w / (GLfloat) h, 0.1, 1000.0);  glMatrixMode(GL\_MODELVIEW);  }  int main(int argc, char \*\*argv) {  glutInit(&argc, argv);  glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGBA | GLUT\_STENCIL | GLUT\_DEPTH); //add a stencil buffer to the window  glutInitWindowSize(800, 600);  glutInitWindowPosition(100, 100);  glutCreateWindow("PEMODELAN RUANG 3D KELOMPOK 7");  init();  glutDisplayFunc(display);  glutIdleFunc(display);  glutReshapeFunc(reshape);  glutSpecialFunc(navigasi);  glutKeyboardFunc(keyboard);  glLightfv(GL\_LIGHT0, GL\_SPECULAR, light\_specular);//mengatur cahaya kilau pada objek  glLightfv(GL\_LIGHT0, GL\_POSITION, light\_position);// mengkonfigurasi sumber cahaya  glMaterialfv(GL\_FRONT, GL\_SPECULAR, mat\_specular);///mengatur cahaya kilau pada objek  glMaterialfv(GL\_FRONT, GL\_SHININESS, high\_shininess);// mengatur definisi kesilauan  glColorMaterial(GL\_FRONT, GL\_DIFFUSE);  glutMainLoop();  return 0;  } |

# Daftar Pustaka

[1] Hearn Donald, Baker M. Pauline, Computer Graphics With OpenGL, Pearson

Prentice Hall, United States of America, 2004.

[2] Khronos Group (2008). *The Industry’s Foundation for High Performance*

*Graphics.* Available at <http://opengl.org/>

[3] http://en.wikipedia.org/wiki/Objek3D