**LAPORAN**

**Project Pembuatan 3D Objek di Laboratorium**

**Workshop Komputer Grafik**

**Dosen Pengampu :** Rizky Yuniar Hakkun, S.Kom, MT

****

Oleh : David Dinhaz & Railla Adha Fajri Ardhani  
NRP : 5221600017 & 5221600025

**Program Studi Sarjana Terapan Teknologi Game  
Departemen Teknologi Multimedia Kreatif  
Politeknik Elektronika Negeri Surabaya**

**2022**

|  |  |
| --- | --- |
| Nama | Railla Adha Fajri Ardhani |
| NRP | 5221600025 |
| Tanggal |  |
| Praktek |  |

|  |  |
| --- | --- |
| 1 | Buat model obyek 3D benda - benda yang ada di Lab GT302 :   * Menerapkan perspektif 3D (gunakan proyeksi 3D) * Menerapkan transformasi * Obyek bisa lebih dari satu * Susunan obyek sesuai dengan real yang ada di Lab * Setiap Kelompok terdiri dari 2 orang (kelompok trakhir boleh 3) dan obyek 3D setiap kelompok tidak boleh sama. * Upload project di github/gitlab/sejenisny dan berikan link repository di classroom * Waktu pengerjaan 1 minggu (pengumpulan 19 Oktober 2022) * Jangan lupa menyertakan nama anggota di kelompoknya |
|  | Source Code :  #define \_USE\_MATH\_DEFINES  #include <windows.h>  #include <cstdlib>  #include <cmath>  #include <GL/glew.h>  #include <GL/freeglut.h>  /\* Global variables \*/  char title[] = "PROJECT 1 - ERASER";  GLfloat anglePyramid = 0.0f;  GLfloat angleCube = 0.0f;  int refreshMills = 15;  void initGL()  {      glClearColor(0.0f, 0.5f, 1.0f, 1.0f);      glClearDepth(1.0f);      glEnable(GL\_DEPTH\_TEST);      glDepthFunc(GL\_LEQUAL);      glShadeModel(GL\_SMOOTH);      glHint(GL\_PERSPECTIVE\_CORRECTION\_HINT, GL\_NICEST);  }  void display()  {      glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);      glMatrixMode(GL\_MODELVIEW);      glLoadIdentity();      glTranslatef(1.5f, 0.0f, -7.0f);      glRotatef(angleCube, 1.0f, 1.0f, 1.0f);      glBegin(GL\_QUADS);      //KAIN      glColor3f(0.0f, 0.0f, 0.1f);     // Hitam      glVertex3f( 1.5f, 0.5f, -0.1f);      glVertex3f(-1.5f, 0.5f, -0.1f);      glVertex3f(-1.5f, 0.5f,  0.1f);      glVertex3f( 1.5f, 0.5f,  0.1f);      // Bottom face (y = -1.0f)      glColor3f(0.0f, 0.0f, 0.1f);      glVertex3f( 1.5f, -0.5f,  0.1f);      glVertex3f(-1.5f, -0.5f,  0.1f);      glVertex3f(-1.5f, -0.5f, -0.1f);      glVertex3f( 1.5f, -0.5f, -0.1f);      // Front face  (z = 1.0f)      glColor3f(0.0f, 0.0f, 0.1f);      glVertex3f( 1.5f,  0.5f, 0.1f);      glVertex3f(-1.5f,  0.5f, 0.1f);      glVertex3f(-1.5f, -0.5f, 0.1f);      glVertex3f( 1.5f, -0.5f, 0.1f);      // Back face (z = -1.0f)      glColor3f(0.0f, 0.0f, 0.1f);      glVertex3f( 1.5f, -0.5f, -0.1f);      glVertex3f(-1.5f, -0.5f, -0.1f);      glVertex3f(-1.5f,  0.5f, -0.1f);      glVertex3f( 1.5f,  0.5f, -0.1f);      // Left face (x = -1.0f)      glColor3f(0.0f, 0.0f, 0.1f);      glVertex3f(-1.5f,  0.5f,  0.1f);      glVertex3f(-1.5f,  0.5f, -0.1f);      glVertex3f(-1.5f, -0.5f, -0.1f);      glVertex3f(-1.5f, -0.5f,  0.1f);      // Right face (x = 1.0f)      glColor3f(0.0f, 0.0f, 0.1f);      glVertex3f(1.5f,  0.5f, -0.1f);      glVertex3f(1.5f,  0.5f,  0.1f);      glVertex3f(1.5f, -0.5f,  0.1f);      glVertex3f(1.5f, -0.5f, -0.1f);      // BADAN      //Top face      glColor3f(0.91f, 0.76f, 0.65f);     // Cream      glVertex3f( 1.45f, 0.45f, 0.11f);      glVertex3f(-1.45f, 0.45f, 0.11f);      glVertex3f(-1.45f, 0.45f, 0.55f);      glVertex3f( 1.45f, 0.45f, 0.55f);      // Bottom face (y = -1.0f)      glColor3f(0.91f, 0.76f, 0.65f);     // Cream      glVertex3f( 1.45f, -0.45f, 0.55f);      glVertex3f(-1.45f, -0.45f, 0.55f);      glVertex3f(-1.45f, -0.45f, 0.11f);      glVertex3f( 1.45f, -0.45f, 0.11f);      // Front face  (z = 1.0f)      glColor3f(0.91f, 0.76f, 0.65f);     // Cream      glVertex3f( 1.45f,  0.45f, 0.55f);      glVertex3f(-1.45f,  0.45f, 0.55f);      glVertex3f(-1.45f, -0.45f, 0.55f);      glVertex3f( 1.45f, -0.45f, 0.55f);      // Left face (x = -1.0f)      glColor3f(0.91f, 0.76f, 0.65f);     // Cream      glVertex3f(-1.45f,  0.45f, -0.0f);      glVertex3f(-1.45f,  0.45f, 0.55f);      glVertex3f(-1.45f, -0.45f, 0.55f);      glVertex3f(-1.45f, -0.45f, -0.0f);      // right face      glColor3f(0.91f, 0.76f, 0.65f);     // Cream      glVertex3f(1.45f,  0.45f, 0.55f);      glVertex3f(1.45f,  0.45f, -0.0f);      glVertex3f(1.45f, -0.45f, -0.0f);      glVertex3f(1.45f, -0.45f, 0.55f);          // PEGANGAN      glColor3f(1.0f, 1.0f, 1.0f);     // Putih      glVertex3f( 1.45f, 0.45f, 0.55f);      glVertex3f(-1.45f, 0.45f, 0.55f);      glVertex3f(-1.55f, 0.5f, 0.65f);      glVertex3f( 1.55f, 0.5f, 0.65f);        // Bottom face (y = -1.0f)      glColor3f(1.0f, 1.0f, 1.0f);     // Putih      glVertex3f( 1.55f, -0.5f, 0.65f);      glVertex3f(-1.55f, -0.5f, 0.65f);      glVertex3f(-1.45f, -0.45f, 0.55f);      glVertex3f( 1.45f, -0.45f, 0.55f);      // Front face  (z = 1.0f)      glColor3f(1.0f, 1.0f, 1.0f);     // Putih      glVertex3f( 1.55f,  0.5f, 0.65f);      glVertex3f(-1.55f,  0.5f, 0.65f);      glVertex3f(-1.55f, -0.5f, 0.65f);      glVertex3f( 1.55f, -0.5f, 0.65f);      // Left face (x = -1.0f)      glColor3f(1.0f, 1.0f, 1.0f);     // Putih      glVertex3f(-1.55f, 0.5f, 0.65f);      glVertex3f(-1.45f, 0.45f, 0.55f);      glVertex3f(-1.45f, -0.45f, 0.55f);      glVertex3f(-1.55f, -0.5f, 0.65f);      // right face      glColor3f(1.0f, 1.0f, 1.0f);     // Putih      glVertex3f(1.55f,  0.5f, 0.65f);      glVertex3f(1.45f,  0.45f, 0.55f);      glVertex3f(1.45f, -0.45f, 0.55f);      glVertex3f(1.55f, -0.5f, 0.65f);      glEnd();      glutSwapBuffers();      angleCube -= 0.15f;  }  void timer(int value)  {      glutPostRedisplay();      glutTimerFunc(refreshMills, timer, 0);  }  void reshape(GLsizei width, GLsizei height)  {      if (height == 0) height = 1;      GLfloat aspect = (GLfloat)width / (GLfloat)height;      glViewport(0, 0, width, height);      glMatrixMode(GL\_PROJECTION);      glLoadIdentity();      gluPerspective(45.0f, aspect, 0.1f, 100.0f);  }  /\* Main function: GLUT runs as a console application starting at main() \*/  int main(int argc, char\*\* argv)  {      glutInit(&argc, argv);      glutInitDisplayMode(GLUT\_DOUBLE);      glutInitWindowSize(640, 480);      glutInitWindowPosition(25, 50);      glutCreateWindow(title);      glutDisplayFunc(display);      glutReshapeFunc(reshape);      initGL();      glutTimerFunc(0, timer, 0);      glutMainLoop();      return 0;  } |
|  | Output Program : |
|  | Penjelasan: |
|  |  |