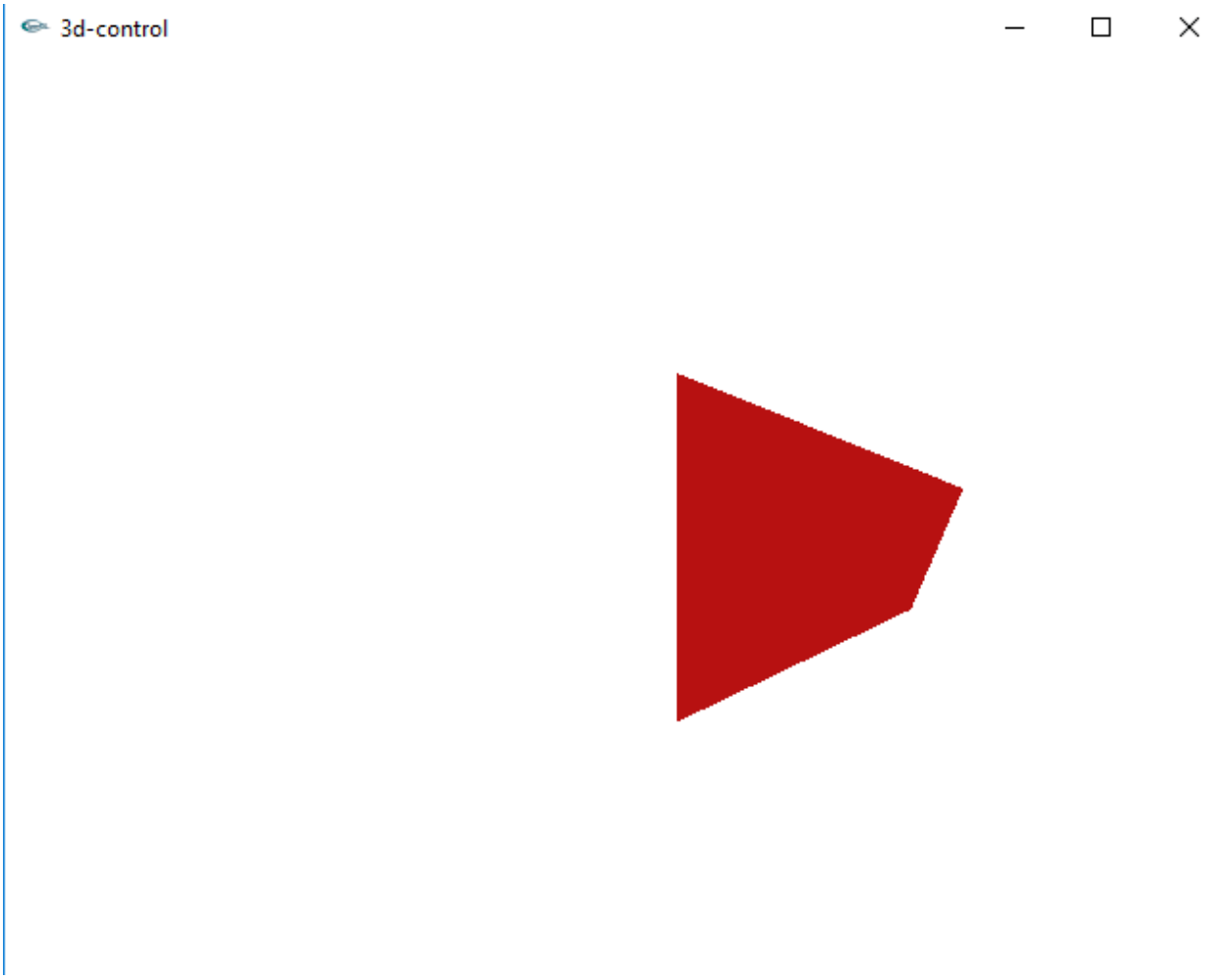
TUGAS PG 3




```
#include <windows.h>
#include "GL/glut.h"
#include "stdio.h"

float view_rotx = 20.0f, view_roty = 30.0f;
int oldMouseX, oldMouseY;
float R_Z=0.0f, R_X=0.0f, R_Y=0.0f;
float T_Z=-16.0f, T_X=0.0f, T_Y=0.0f;

void initGL(){
glShadeModel(GL_FLAT);

float ambient[] = {1.0f,1.0f,1.0f,1.0f};
float diffuse[] = {1.0f,1.0f,1.0f,1.0f};
float specular[] = {0.2f,1.0f,0.2f,1.0f};
float position[] = {20.0f,30.0f,20.0f,0.0f};

glLightfv(GL_LIGHT0, GL_AMBIENT, ambient);
glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuse);
glLightfv(GL_LIGHT0, GL_POSITION, position);
glMaterialfv(GL_FRONT, GL_SPECULAR, specular);

```

```
float mambient[] ={0.1745f, 0.01175f, 0.01175f, 0.55f};
float mdiffuse[] ={0.61424f, 0.04136f, 0.04136f, 0.55f };
float mspecular[] ={0.727811f, 0.626959f, 0.626959f, 0.55f };
float mshine =76.8f;

glMaterialfv(GL_FRONT,GL_AMBIENT,mambient);
glMaterialfv(GL_FRONT,GL_DIFFUSE,mdiffuse);

glMaterialfv(GL_FRONT,GL_SPECULAR,mspecular);
glMaterialf (GL_FRONT,GL_SHININESS,mshine);

glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
glEnable(GL_DEPTH_TEST);
glEnable(GL_NORMALIZE);
}

void Kubus(){
    glBegin(GL_POLYGON);/* f1: Surface bagian depan */
    glColor3f(1,0,1);
    glVertex3f(0.0f,0.0f,0.0f);
    glVertex3f(0.0f,0.0f,1.0f);
    glVertex3f(1.0f,0.0f,1.0f);
    glVertex3f(1.0f,0.0f,0.0f);
    glEnd();
    glColor3f(1,1,0);
    glBegin(GL_POLYGON);/* f2: Surface bagian bawah (ABFE) */
    glVertex3f(0.0f,0.0f,0.0f);
    glVertex3f(1.0f,0.0f,0.0f);
    glVertex3f(1.0f,1.0f,0.0f);
    glVertex3f(0.0f,1.0f,0.0f);
    glEnd();
    glBegin(GL_POLYGON);/* f3:Surface bagian belakang (CDHG)*/
    glVertex3f(1.0f,1.0f,0.0f);
    glVertex3f(1.0f,1.0f,1.0f);
    glVertex3f(0.0f,1.0f,1.0f);
    glVertex3f(0.0f,1.0f,0.0f);
    glEnd();
    glBegin(GL_POLYGON);/* f4: Surface bagian atas (EFGH)*/
    glVertex3f(1.0f,1.0f,1.0f);
    glVertex3f(1.0f,0.0f,1.0f);
    glVertex3f(0.0f,0.0f,1.0f);
    glVertex3f(0.0f,1.0f,1.0f);
    glEnd();
    glBegin(GL_POLYGON);/* f5: Surface bagian kiri
    (ADEH)*/
    glVertex3f(0.0f,0.0f,0.0f);
    glVertex3f(0.0f,1.0f,0.0f);
    glVertex3f(0.0f,1.0f,1.0f);
    glVertex3f(0.0f,0.0f,1.0f);
    glEnd();
    glBegin(GL_POLYGON);/* f6: Surface bagian kanan (BCFG)*/
    glVertex3f(1.0f,0.0f,0.0f);
    glVertex3f(1.0f,0.0f,1.0f);
    glVertex3f(1.0f,1.0f,1.0f);
    glVertex3f(1.0f,1.0f,0.0f);
    glEnd();
}
void LimasPoli(){
    glColor3f(1,1,0);
    glBegin(GL_POLYGON);/* f1: Surface bagian depan */
```

```
    glColor3f(1,0,1);
    glVertex3f(0.0f,0.0f,0.0f); //B
    glVertex3f(1.0f,0.0f,0.0f); //C
    glVertex3f(0.5f,0.5f,1.0f); //A
    glEnd();
    glBegin(GL_POLYGON);
    glVertex3f(1.0f,0.0f,0.0f); //C
    glVertex3f(0.5f,1.0f,0.0f); //D
    glVertex3f(0.5f,0.5f,1.0f); //A
    glEnd();
    glBegin(GL_POLYGON);
    glVertex3f(0.5f,0.5f,1.0f); //A
    glVertex3f(0.5f,1.0f,0.0f); //D
    glVertex3f(0.0f,0.0f,0.0f); //B
    glEnd();
    glBegin(GL_POLYGON);/* f2: Surface bagian bawah (ABFE) */
            glVertex3f(0.0f,0.0f,0.0f); //B
            glVertex3f(1.0f,0.0f,0.0f); //C
            glVertex3f(0.5f,1.0f,0.0f); //D
            glEnd();
        }
        void LimasLine(){


            glBegin(GL_LINES);/* f2: Surface bagian bawah (ABFE) */
                glVertex3f(0.0f,0.0f,0.0f); //B
                glVertex3f(1.0f,0.0f,0.0f); //C

                glVertex3f(1.0f,0.0f,0.0f); //C
                glVertex3f(0.5f,1.0f,0.0f); //D

                glVertex3f(0.5f,1.0f,0.0f); //D
                glVertex3f(0.0f,0.0f,0.0f); //B
            glEnd();

            glBegin(GL_LINES);/* f1: Surface bagian depan */
                glVertex3f(0.5f,0.5f,1.0f); //A
                glVertex3f(0.0f,0.0f,0.0f); //B
                glVertex3f(0.5f,0.5f,1.0f); //A
                glVertex3f(1.0f,0.0f,0.0f); //C
                glVertex3f(0.5f,0.5f,1.0f); //A
                glVertex3f(0.5f,1.0f,0.0f); //D
            glEnd();
        }
        void Prisma(){
            glBegin(GL_POLYGON);/* f1: Surface bagian depan */
            glVertex3f(0.0f,0.0f,0.0f);
            glVertex3f(0.0f,0.0f,1.0f);
            glVertex3f(1.0f,0.0f,1.0f);
            glVertex3f(1.0f,0.0f,0.0f);
            glEnd();
            glBegin(GL_POLYGON);/* f2: Surface bagian bawah (ABFE) */
                glVertex3f(0.0f,0.0f,0.0f);
                glVertex3f(1.0f,0.0f,0.0f);
                glVertex3f(0.5f,1.0f,0.0f);
            glEnd();
            glBegin(GL_POLYGON);/* f4: Surface bagian atas (EFGH)*/
                glVertex3f(0.5f,1.0f,1.0f);
                glVertex3f(1.0f,0.0f,1.0f);
                glVertex3f(0.0f,0.0f,1.0f);
            glEnd();
```

```
145.            glBegin(GL_POLYGON);/* f5: Surface bagian kiri
146.            (ADEH)*/
147.                glVertex3f(0.0f,0.0f,0.0f);
148.                glVertex3f(0.5f,1.0f,0.0f);
149.                glVertex3f(0.5f,1.0f,1.0f);
150.                glVertex3f(0.0f,0.0f,1.0f);
151.            glEnd();
152.            glBegin(GL_POLYGON);/* f6: Surface bagian kanan (BCFG)*/
153.                glVertex3f(1.0f,0.0f,0.0f);
154.                glVertex3f(1.0f,0.0f,1.0f);
155.                glVertex3f(0.5f,1.0f,1.0f);
156.                glVertex3f(0.5f,1.0f,0.0f);
157.            glEnd();
158.
159.        }
160.        void keyFunction(unsigned char key, int x, int y){
161.            switch(key){
162.                case 85: // Rotasi sumbu Z dengan tombol U
163.                R_Z = R_Z + -30.0f;
164.                break;
165.                case 75: // Rotasi sumbu Y dengan tombol K
166.                R_Y = R_Y + -30.0f;
167.                break;
168.                case 74: // Rotasi sumbu X dengan tombol J
169.                R_X = R_X + -30.0f;
170.                break;
171.                case 81: // Translasi sumbu Z dengan tombol Q
172.                T_Z = T_Z + -1.0f;
173.                break;
174.                case 69: // Translasi sumbu Z dengan tombol E
175.                T_Z = T_Z + 1.0f;
176.                break;
177.                case 83: // Translasi sumbu Y dengan tombol S
178.                T_Y = T_Y + -1.0f;
179.                break;
180.                case 87: // Translasi sumbu Y dengan tombol W
181.                T_Y = T_Y + 1.0f;
182.                break;
183.                case 65: // Translasi sumbu X dengan tombol A
184.                T_X = T_X + -1.0f;
185.                break;
186.                case 68: // Translasi sumbu X dengan tombol A
187.                T_X = T_X + 1.0f;
188.                break;
189.
190.            }
191.        }
192.        float sudut = 0;
193.        void display(){
194.            glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
195.            glLoadIdentity();
196.            //gluLookAt(4,4,4, // eye pos
197.            //0,0,0, // look at
198.            //0,0,1); // up
199.            glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
200.            //glTranslatef(0.5f, 0.5f, 0.5f);
201.            //glRotatef(90, 0,0,1);
202.            //glRotatef(view_rotx, 1.0f, 0.0f, 0.0f);
203.            //glRotatef(view_roty, 0.0f, 1.0f, 0.0f);
204.            //glTranslatef(-0.5f, -0.5f, -0.5f);
205.
```

```
206.            //WITH BUTTON
207.            glTranslatef(-T_X, T_Y, T_Z);
208.            glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);
209.            glRotatef(180.0f, 0.0f, 0.0f, 1.0f);
210.
211.            glRotatef(R_X, 1.0f, 0.0f, 0.0f);
212.            glRotatef(R_Y, 0.0f, 1.0f, 0.0f);
213.            glRotatef(R_Z, 0.0f, 0.0f, 1.0f);
214.
215.            Prisma();
216.            sudut++;
217.            glFlush();
218.            glutSwapBuffers();
219.        }
220.
221.        void timer(int value){
222.            glutPostRedisplay();
223.            glutTimerFunc(15, timer, 0);
224.        }
225.
226.        void reshape(GLsizei width, GLsizei height){
227.            if (height == 0) height = 1;
228.                GLfloat aspect = (GLfloat)width / (GLfloat)height;
229.                glViewport(30, 6, width, height);
230.                glMatrixMode(GL_PROJECTION);
231.                glLoadIdentity();
232.                gluPerspective(45.0f, aspect, 1.0f, 20.0f);
233.                glMatrixMode(GL_MODELVIEW);
234.                glLoadIdentity();
235.        }
236.
237.        void mouseControl(int button, int state, int x, int y){
238.            oldMouseX = x;
239.            oldMouseY = y;
240.        }
241.
242.        void mouseMotion(int x, int y){
243.            int getX = x;
244.            int getY = y;
245.            float thetaY = 360.0f*(getX - oldMouseX)/640;
246.            float thetaX = 360.0f*(getY - oldMouseY)/480;
247.            oldMouseX = getX;
248.            oldMouseY = getY;
249.            view_rotx += thetaX;
250.            view_roty += thetaY;
251.        }
252.
253.        int main(int argc, char **argv){
254.            glutInit(&argc, argv);
255.            glutInitDisplayMode(GLUT_DOUBLE | GLUT_DEPTH);
256.            glutInitWindowSize(640, 480);
257.            glutInitWindowPosition(50, 50);
258.            glutCreateWindow("3d-control");
259.            glutDisplayFunc(display);
260.            glutReshapeFunc(reshape);
261.            glutKeyboardFunc(keyFunction);
262.            initGL();
263.            glutMouseFunc(mouseControl);
264.            glutMotionFunc(mouseMotion);
265.            glutTimerFunc(0, timer, 0);
266.            glutMainLoop();
```

```
267.             return 0;
268.         }
```