Ovaldy | 161111062

```
#include "windows.h"
#include "GL/glut.h"
#include "stdio.h"
#include "math.h"
float view_rotx = 0.0f, view_roty = 180.0f;
float R_Z=0.0f, R_X=0.0f, R_Y=0.0f;
float T_Z=-4.0f, T_X=0.0f, T_Y=-0.0f;

int oldMouseX, oldMouseY;

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 timer(int value){
glutPostRedisplay();
glutTimerFunc(15, timer, 0);
}

void mouseControl(int button, int state, int x, int y){
oldMouseX = x;
oldMouseY = y;
}

void mouseMotion(int x, int y){
int getX = x;
int getY = y;
float thetaY = 360.0f*(getX - oldMouseX)/640;
float thetaX = 360.0f*(getY - oldMouseY)/480;
oldMouseX = getX;
oldMouseY = getY;
view_rotx += thetaX;
view_roty += thetaY;
```

```
}
void reshape(GLsizei width, GLsizei height){
if (height == 0) height = 1;
GLfloat aspect = (GLfloat)width / (GLfloat)height;
glViewport(30, 6, width, height);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluPerspective(45.0f, aspect, 1.0f, 20.0f);
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
}

void tube(){
float BODY_LENGTH=2.0f;
float BODY_RADIUS=1.0f;
int SLICES=30;
int STACKS=30;
GLUquadric *q = gluNewQuadric();
gluCylinder(q, BODY_RADIUS, BODY_RADIUS, BODY_LENGTH, SLICES, STACKS);
gluDisk(q, 0.0f, BODY_RADIUS, SLICES, STACKS); //lingkaran untuk tutup atas
glTranslatef(0.0f, 0.0f, BODY_LENGTH);
gluDisk(q, 0.0f, BODY_RADIUS, SLICES, STACKS); //lingkaran untuk tutup bawah
}

void drawHemisphere(int scaley, int scalex, GLfloat r) {

int i, j;
GLfloat v[scalex*scaley][3];

for (i=0; i<scalex; ++i) {
for (j=0; j<scaley; ++j) {
v[i*scaley+j][0]=r*cos(j*2*M_PI/scaley)*cos(i*M_PI/(2*scalex));
v[i*scaley+j][1]=r*sin(i*M_PI/(2*scalex));
v[i*scaley+j][2]=r*sin(j*2*M_PI/scaley)*cos(i*M_PI/(2*scalex));
}
}

glBegin(GL_QUADS);
for (i=0; i<scalex-1; ++i) {
for (j=0; j<scaley; ++j) {
glVertex3fv(v[i*scaley+j]);
        glVertex3fv(v[i*scaley+(j+1)%scaley]);
        glVertex3fv(v[(i+1)*scaley+(j+1)%scaley]);
        glVertex3fv(v[(i+1)*scaley+j]);
        }
        }
        glEnd();
        }


        void display_pen(){
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
        glLoadIdentity();
        glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
        glTranslatef(0,0.0f, -10.0f);
        glRotatef(view_rotx, 1.0f, 0.0f, 0.0f);
        glRotatef(view_roty, 0.0f, 1.0f, 0.0f);
        glRotatef(45, 0.0f, 0.0f, 1.0f);
        glRotatef(90,0.0f,1.0f,0.0f);
        tube();
        glRotatef(90,1.0f,0.0f,0.0f);
```

```
120.        drawHemisphere(30, 30, 1.0f);
121.        glFlush();
122.        glutSwapBuffers();
123.        }
124.
125.        void display_hemisphere(){
126.        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
127.        glLoadIdentity();
128.        glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
129.        glTranslatef(0,0.0f, -10.0f);
130.        glRotatef(view_rotx, 1.0f, 0.0f, 0.0f);
131.        glRotatef(view_roty, 0.0f, 1.0f, 0.0f);
132.        drawHemisphere(30,30,1.0f);
133.        glFlush();
134.        glutSwapBuffers();
135.        }
136.
137.        int main(int argc, char **argv){
138.        glutInit(&argc, argv);
139.
140.        glutInitDisplayMode(GLUT_DOUBLE | GLUT_DEPTH);
141.        glutInitWindowSize(480, 360);
142.        glutInitWindowPosition(50, 50);
143.        glutCreateWindow("hemisphere");
144.        glutDisplayFunc(display_hemisphere);
145.        glutReshapeFunc(reshape);
146.        initGL();
147.        glutMouseFunc(mouseControl);
148.        glutMotionFunc(mouseMotion);
149.        glutTimerFunc(0, timer, 0);
150.
151.        glutInitDisplayMode(GLUT_DOUBLE | GLUT_DEPTH);
152.        glutInitWindowSize(480, 360);
153.        glutInitWindowPosition(50, 50);
154.        glutCreateWindow("pen");
155.        glutDisplayFunc(display_pen);
156.        glutReshapeFunc(reshape);
157.        initGL();
158.        glutMouseFunc(mouseControl);
159.        glutMotionFunc(mouseMotion);
160.        glutTimerFunc(0, timer, 0);
161.        glutMainLoop();
162.        return 0;
163.        }
```