```
#include <GL/gl.h>
#include <GL/glu.h>
#include <GL/glut.h>


void init(void)
{
     GLfloat mat_specular[] = { 1.0, 1.0, 1.0, 1.0 };
     GLfloat mat_shininess[] = { 50.0 };
     GLfloat light_position[] = { 0.0,1.0, 1.0, 1.0 };


     glClearColor (0.0, 0.0, 0.0, 0.0);
     glShadeModel (GL_SMOOTH);
//penggunaan shading

     glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
     glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
     glLightfv(GL_LIGHT0, GL_POSITION, light_position);



     glEnable(GL_LIGHTING);
     glEnable(GL_LIGHT0);

     glEnable(GL_DEPTH_TEST);




}
//prosedur memanggil objek bola
void bola(void)
{
     glutSolidSphere (0.8, 120, 20);
}
//prosedur memanggil objek kubus
void kubus(void)
{
     glutSolidCube(0.6);
}
//prosedur memanggil objek teapot
void Teapot(void)
{
     glutSolidTeapot(0.4);
}

//membuat torus
void torus(void)
{

     glutWireTorus(0.15, 0.4, 50, 100);
```

```
}

//static GLdouble spin;
void display(void)
{
    const double t= glutGet(GLUT_ELAPSED_TIME) /1000.0;
    const double a = t*90.0, b = t/2, c = t/10;

    //GLfloat light_position[] = { 0.0,0.0, 1.0, 1.0 };

    glutSwapBuffers();
     glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

     //bola 1 ----------
     glLoadIdentity();

     //glPushMatrix();
     //gluLookAt (0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
     //glRotated(spin, 1.0, 0.0, 0.0);
     //glLightfv(GL_LIGHT0, GL_POSITION, light_position);
     //glPopMatrix();

     glEnable(GL_COLOR_MATERIAL);
     glColorMaterial(GL_FRONT,GL_SPECULAR);
     glPushMatrix();
     glTranslatef(0.5,0.5,-1.0);
     glRotatef(a,0,1,0);
     glColor3f(1.0,0.4,0.0);
     bola();
     glPopMatrix();

     //kubus
     glPushMatrix();
     glTranslatef(-0.7,-0.6,-1.0);
     glRotatef(a,1,1,1);
     kubus();
     glPopMatrix();

     //teapot
     glPushMatrix();
     glTranslatef(0.5,-0.6,-1.0);
     glRotatef(a,0.0,1.0,0.0);
     Teapot();
     glPopMatrix();

     //torus
     glPushMatrix();
     glTranslatef(-0.8,0.5,-1.0);
     glRotatef(a,1.0,1.0,0.0);
     torus();
     glPopMatrix();
     glDisable(GL_COLOR_MATERIAL);
```

```
     glFlush ();
     glutPostRedisplay();
}

void reshape (int w, int h)
{
     glViewport (0, 0, (GLsizei) w, (GLsizei) h);
     glMatrixMode (GL_PROJECTION);
     glLoadIdentity();
     if (w <= h)
        glOrtho (-1.5, 1.5, -1.5*(GLfloat)h/(GLfloat)w,
        1.5*(GLfloat)h/(GLfloat)w, -10.0, 10.0);
     else
        glOrtho (-1.5*(GLfloat)w/(GLfloat)h,
        1.5*(GLfloat)w/(GLfloat)h, -1.5, 1.5, -10.0, 10.0);

     glMatrixMode(GL_MODELVIEW);
     glLoadIdentity();
     gluLookAt (0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize (700, 700);
    glutInitWindowPosition (100, 70);
    glutCreateWindow ("Program Pencahayaan");
    init ();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMainLoop();
    return 0;
}
```