

## 1. Source code 3D dengan Open GL

```
//Program ini untuk mensimulasikan transformasi 3 Dimensi.
//Tekan x untuk putar dengan sumbu putar x
//Tekan y untuk putar dengan sumbu putar y
//Tekan z untuk putar dengan sumbu putar z
//Tekan w untuk menambah nilai y
//Tekan s untuk mengurangi nilai y
//Tekan a untuk menambah nilai x
//Tekan d untuk mengurangi nilai x
//Tekan q untuk menambah nilai z
//Tekan e untuk mengurangi nilai z

#include <stdlib.h>
#include <GL/glut.h>
static float ypoz = 0, zpoz = 0, xpoz = 0, a = 0, b = 0, c = -10;
void coba(int x1,int y1,int x2,int y2,int x3,int y3,int x4,int y4,int
z1,float z2,float z3){
    glColor3f(1,0,0); //sisi depan
    glVertex3d(x1,y1,z2);
    glVertex3d(x2,y2,z2);
    glVertex3d(x3,y3,z2);
    glVertex3d(x4,y4,z2);
    glColor3f(1,0,0); //right
    glVertex3d(x2,y2,z2);
    glVertex3d(x2,y2,z1);
    glVertex3d(x3,y3,z1);
    glVertex3d(x3,y3,z2);
    glColor3f(1,0,0); //back
    glVertex3d(x1,y1,z1);
    glVertex3d(x2,y2,z1);
    glVertex3d(x3,y3,z1);
    glVertex3d(x4,y4,z1);
    glColor3f(1,0,0); //left
    glVertex3d(x1,y1,z2);
    glVertex3d(x1,y1,z1);
    glVertex3d(x4,y4,z1);
    glVertex3d(x4,y4,z2);
    glColor3f(1,0,1); //bottom
    glVertex3d(x1,y1,z2);
    glVertex3d(x2,y2,z2);
    glVertex3d(x2,y2,z1);
    glVertex3d(x1,y1,z1);
    glColor3f(1,1,0); //top
    glVertex3d(x3,y3,z2);
    glVertex3d(x4,y4,z2);
    glVertex3d(x4,y4,z1);
    glVertex3d(x3,y3,z1);
}
```

```

void cobain(int x1,int y1,int x2,int y2,int x3,int y3,int x4,int
y4,float z1,float z2,float z3){ //atap
    glColor3f(1,1,1);//kanan
    glVertex3d(x3,y3,z1);
    glVertex3d(x3,y3,z2);
    glVertex3d(x3,y1,z3);
    glColor3f(1,1,1);//kiri
    glVertex3d(x4,y4,z2);
    glVertex3d(x4,y4,z1);
    glVertex3d(x4,y1,z3);
}
void cobaini(int x1,int y1,int x2,int y2,int x3,int y3,int x4,int
y4,float z1,float z2,float z3){//atap
    glColor3f(1,0,1);//belakang
    glVertex3d(x4,y4,z1);
    glVertex3d(x3,y3,z1);
    glVertex3d(x3,y1,z3);
    glVertex3d(x4,y1,z3);
    glColor3f(1,0,1);//depan
    glVertex3d(x3,y3,z2);
    glVertex3d(x4,y4,z2);
    glVertex3d(x4,y1,z3);
    glVertex3d(x3,y1,z3);
}
void init(void)
{
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glOrtho(-12,12,-6,6,-35,35);
    glEnable(GL_DEPTH_TEST);
    glShadeModel (GL_SMOOTH);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    gluPerspective(80.0,2.2, 1.0, 20.0);
    glMatrixMode (GL_MODELVIEW);
}

void display(void)
{
    glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity (); //digunakan untuk me-nonaktifkan/me-reset
    transformasi sebelumnya
    glTranslatef(a,b,c); //digunakan untuk melakukan perpindahan
    posisi gelombang
    glRotatef(xpoz,1,0,0);
    glRotatef(ypoz,0,1,0);
    glRotatef(zpoz,0,0,1);

    glBegin(GL_QUADS);
        //Rumah
        coba(-8,-2,8,-2,8,2,-8,2,-5,3,0);//rumah
        coba(2,-2,8,-2,8,2,2,2,-5,4,0);//kamar
        coba(-8,-2,-7,-2,-7,2,-8,2,-5,4,0);//dindingpagar kiri

```

```

        coba(-8, -2, -7, -2, -7, 0, -8, 0, -5, 5.5, 0);
        coba(7, -2, 8, -2, 8, 0, 7, 0, -5, 5.5, 0);
        cobaini(-3, 5, 0, 0, 8, 2, -8, 2, -5.5, 4.5, 0);

glColor3f(1, 1, 1); //plafon depan
glVertex3d(8, 1.8, -3);
glVertex3d(-8, 1.8, -3);
glVertex3d(-8, 1.8, 4.5);
glVertex3d(8, 1.8, 4.5); //
glVertex3d(8, 1.8, 4.5);
glVertex3d(-8, 1.8, 4.5);
glVertex3d(-8, 2, 4.5);
glVertex3d(8, 2, 4.5);

glColor3f(1, 1, 1); //plafon belakang
glVertex3d(8, 1.8, -5.5);
glVertex3d(-8, 1.8, -5.5);
glVertex3d(-8, 1.8, 4.5);
glVertex3d(8, 1.8, 4.5);
glVertex3d(8, 1.8, -5.5);
glVertex3d(-8, 1.8, -5.5);
glVertex3d(-8, 2, -5.5);
glVertex3d(8, 2, -5.5);

glVertex3d(4, -0.5, 4.1); //cendela kamar
glVertex3d(6, -0.5, 4.1);
glVertex3d(6, 1, 4.1);
glVertex3d(4, 1, 4.1);
glVertex3d(2.7, -0.5, 4.1); //cendela kamar2
glVertex3d(3.8, -0.5, 4.1);
glVertex3d(3.8, 1, 4.1);
glVertex3d(2.7, 1, 4.1);

glVertex3d(-2, -2, 3.05); //pintu depan
glVertex3d(2, -2, 3.05);
glVertex3d(2, 1.3, 3.05);
glVertex3d(-2, 1.3, 3.05);
glVertex3d(-2, -2, -5.05); //pintu belakang
glVertex3d(2, -2, -5.05);
glVertex3d(2, 1, -5.05);
glVertex3d(-2, 1, -5.05);

glVertex3d(-4.5, -1, 3.05); //cendela rmh1 depan
glVertex3d(-2.5, -1, 3.05);
glVertex3d(-2.5, 1.3, 3.05);
glVertex3d(-4.5, 1.3, 3.05);
glVertex3d(-6.5, -1, -5.05); //cendela rmh1 belakang
glVertex3d(-2.5, -1, -5.05);
glVertex3d(-2.5, 1, -5.05);
glVertex3d(-6.5, 1, -5.05);

glVertex3d(-7, -1, 3.05); //cendela rmh2 depan

```

```

    glVertex3d(-4.8, -1, 3.05);
    glVertex3d(-4.8, 1.3, 3.05);
    glVertex3d(-7, 1.3, 3.05);
    glVertex3d(6.5, -1, -5.05); //cendela rmh2 belakang
    glVertex3d(2.5, -1, -5.05);
    glVertex3d(2.5, 1, -5.05);
    glVertex3d(6.5, 1, -5.05);
    glEnd();

    glBegin(GL_TRIANGLES);
    cobain(-3, 5, 0, 0, 8, 2, -8, 2, -5.5, 4.5, 0);
    glEnd();

    glBegin(GL_LINES);
    glColor3f(0, 0, 0);
    glVertex3d(8, 1.8, 4.5);
    glVertex3d(-8, 1.8, 4.5);
    glVertex3d(-8, 1.8, -5.5);
    glVertex3d(8, 1.8, -5.5);
    glColor3f(1, 0, 0);
    glVertex3d(-4.5, -1, -5.05);
    glVertex3d(-4.5, 1, -5.05);
    glVertex3d(4.5, -1, -5.05);
    glVertex3d(4.5, 1, -5.05);
    glEnd();
    glutSwapBuffers(); //Fungsi berikutnya adalah glutSwapBuffers(),
    yaitu digunakan untuk menukar bagian belakan buffer menjadi buffer
    layar (screen buffer). Dalam modus double-buffered, perintah
    menggambar pada layar, tetapi menggambar pada buffer belakang layar.
    Pada saat ingin menampilkan gambar, gambar yang berada di buffer
    belakang layar dipindahkan ke buffer layar, sehingga menghasilkan
    animasi yang sempurna.
}

void keyboard(unsigned char key, int x, int y)
{
    switch (key) {
        // These will allow you to force some extra rotation by
        pressing the 'y' and 'z' key.
        case 'x':
            xpoz=xpoz+5;
            if (xpoz>360) xpoz=0;
            glutPostRedisplay();
            break;
        case 'y':
            ypoz=ypoz+5;
            if (ypoz>360) ypoz=0;
            glutPostRedisplay();
            break;
        case 'z':
            zpoz = zpoz+1;
            if (zpoz>360) zpoz=0;
    }
}

```

```

        glutPostRedisplay();
        break;
    case 'w':
        b = b + 1;
        glutPostRedisplay();
        break;
    case 's':
        b = b - 1;
        glutPostRedisplay();
        break;
    case 'a':
        a = a + 1;
        glutPostRedisplay();
        break;
    case 'd':
        a = a - 1;
        glutPostRedisplay();
        break;
    case 'q':
        c = c + 1;
        glutPostRedisplay();
        break;
    case 'e':
        c = c - 1;
        glutPostRedisplay();
        break;
    }
}
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_RGB | GLUT_DEPTH | GLUT_DOUBLE);
    glutInitWindowSize (500, 500);
    glutInitWindowPosition (100, 100);
    glutCreateWindow("Rumah|09_109,09_123,09_128");
    init ();
    glutDisplayFunc(display);
    glutKeyboardFunc(keyboard);
    glutMainLoop();
    return 0;
}

```

## 2. Penjelasan Fungsi- Fungsi Source Code

- `#include <stdlib.h>`  
`#include <GL/glut.h>`

Fungsi ini digunakan untuk menggunakan fungsi GLUT yang ada dalam Windows

- ***static float***

berfungsi untuk mendeklarasikan posisi awal X, Y, Z

- ***glColor3f (1,0,0)***

berfungsi untuk mendefinisikan warna dari objek yang dibuat dengan warna (1,0,0) yaitu warna merah

- ***glVertex3d(x1,y1,z2)***

Berfungsi untuk menyatakan sebuah titik koordinat x, y dan z.

- ***glClearColor (0.0, 0.0, 0.0, 0.0);***

Berfungsi Mendefinisikan warna dari windows yang dibuat dengan warna (0,0,0) yaitu warna hitam

- ***glOrtho(-12,12,-6,6,-35,35);***

Mendefinisikan besarnya sistem koordinat dengan range sumbu x adalah [-12,12], range untuk sumbu y adalah [-6,6], range untuk sumbu z adalah [-35,35]

- ***glEnable(GL\_DEPTH\_TEST);***

Berfungsi untuk mengaktifkan atau menampilkan GL\_DEPTH\_TEST

- ***glShadeModel (GL\_SMOOTH);***

Berfungsi untuk memberi warna bayangan dengan satu atau banyak warna

- ***glMatrixMode (GL\_PROJECTION);***

Berfungsi untuk memberi nilai masukan pada matrix dengan model projection.

- ***glLoadIdentity ();***

Fungsi untuk memanggil matriks identitas ke matriks saat ini, agar nilai-nilai awal kembali ke pusat koordinat (0,0,0)

- ***gluPerspective(80.0,2.2, 1.0, 20.0);***

Berfungsi untuk mendeklarasikan jarak ke depan dan ke belakang dari bidang pandang (clipping plane)

- ***glTranslatef(a,b,c);***

Fungsi ini menggeser seluruh pemandangan ke kiri-kanan, atas-bawah, maju-mundur tergantung nilai x, y, dan z

- ***glRotatef(xpoz,1,0,0);***

Fungsi untuk memutar seluruh obyek dengan sudut angle (0–359) dan vektor yang dinyatakan dengan nilai x, y, dan z

- ***glBegin(GL\_QUADS);***

Berfungsi untuk membentuk segi empat

- ***glBegin(GL\_TRIANGLES);***

Berfungsi untuk membentuk segitiga, dan tiap segitiga saling lepas

- ***glBegin(GL\_LINES);***

Berfungsi untuk membentuk garis

- ***glutSwapBuffers();***

digunakan untuk menukar bagian belakan buffer menjadi buffer layar (screen buffer). Dalam modus double-buffered, perintah menggambar pada layar, tetapi menggambar pada buffer belakang layar. Pada saat ingin menampilkan gambar, gambar yang berada di buffer belakang layar dipindahkan ke buffer layar, sehingga menghasilkan animasi yang sempurna.

- ***glutInitDisplayMode (GLUT\_RGB | GLUT\_DEPTH | GLUT\_DOUBLE);***

untuk inisialisasi modus display. Fungsi tersebut meminta window yang bersifat double buffered, dengan warna RGB

- ***glutInitWindowSize (500, 500);***

***glutInitWindowPosition (100, 100);***

Membuat windows dengan ukuran(500,500) dengan titik kiri atas jendela diletakkan pada posisi (100,100) di layar computer

- ***glutKeyboardFunc(keyboard);***

Berfungsi untuk mengaktifkan fungsi pada keyboard

### 3. Penjelasan Fungsi Void

Pada source code di atas, terdapat 5 void yang dicetak tebal. Tiap-tiap void memiliki fungsi masing-masing.

1. **Void coba** menjelaskan mengenai bagaimana kotak kubus 3 dimensi dapat terbentuk melalui beberapa titik. Dimana titik-titik tersebut apabila digabung dan ditarik garis akan menghasilkan kubus 3 dimensi. Dalam void coba sendiri diberi parameter integer untuk tiap angka yang masuk. Di dalam void coba memiliki 5 bagian yang harus diselesaikan agar dapat membentuk kubus 3 dimensi, yaitu bagian depan, atas, kanan, kiri, dan belakang.
2. **Void cobain** dalam void ini terdapat 4 bagian yang terbentuk dari segitiga sama sisi. Niat dalam membuat bentuk limas segiempat adalah ingin membuat atap yang bagus bagi rumah kubus 3 dimensi.
3. **Void init(void)** berfungsi untuk membuat tempat menggambar
4. **Void init** mengatur pengaturan warna latar dan koordinat system.
5. **Void Display** berisi prosedur kerja program yang akan dijalankan.
6. **Void Keyboard** dalam source code diatas berisi penggunaan tombol keyboard yang digunakan untuk menggerakkan obyek.

#### 4. Proses Pengerjaan

- Membuat kubus, perumpamaan titik koordinatnya terdapat dalam sintak berikut :

```
void coba(int x1,int y1,int x2,int y2,int x3,int y3,int x4,int
y4,int z1,int z2,int z3){
    glColor3f(1,0,0); //sisi depan
    glVertex3d(x1,y1,z2);
    glVertex3d(x2,y2,z2);
    glVertex3d(x3,y3,z2);
    glVertex3d(x4,y4,z2);
    glColor3f(0,1,0); //right
    glVertex3d(x2,y2,z2);
    glVertex3d(x2,y2,z1);
    glVertex3d(x3,y3,z1);
    glVertex3d(x3,y3,z2);
    glColor3f(0,0,1); //back
    glVertex3d(x1,y1,z1);
    glVertex3d(x2,y2,z1);
    glVertex3d(x3,y3,z1);
    glVertex3d(x4,y4,z1);
    glColor3f(0,1,1); //left
    glVertex3d(x1,y1,z2);
    glVertex3d(x1,y1,z1);
    glVertex3d(x4,y4,z1);
```



```

    glVertex3d(x4,y4,z2);
    glColor3f(1,0,1);//bottom
    glVertex3d(x1,y1,z2);
    glVertex3d(x2,y2,z2);
    glVertex3d(x2,y2,z1);
    glVertex3d(x1,y1,z1);
    glColor3f(1,1,0);//top
    glVertex3d(x3,y3,z2);
    glVertex3d(x4,y4,z2);
    glVertex3d(x4,y4,z1);
    glVertex3d(x3,y3,z1);
}

```

Titik yang dipakai dalam sintak diatas.  
 glBegin(GL\_QUADS);

```

    coba(-8,-2,2,-2,2,2,-8,2,-2,2,0);

```

```

    glEnd();

```

- Pembuatan atap dalam bentuk limas segi empat dijelaskan dalam sintak berikut :

```

void cobain(int x1,int y1,int x2,int y2,int x3,int y3,int x4,int
y4,int z1,int z2,int z3){
    glColor3f(1,1,1)//belakang;
    glVertex3d(x4,y4,z1);
    glVertex3d(x3,y3,z1);
    glVertex3d(x1,y1,z3);
    glColor3f(0,1,1)//kanan;
    glVertex3d(x3,y3,z1);
    glVertex3d(x3,y3,z2);
    glVertex3d(x1,y1,z3);
    glColor3f(1,0,1)//depan;
    glVertex3d(x3,y3,z2);
    glVertex3d(x4,y4,z2);
    glVertex3d(x1,y1,z3);
    glColor3f(1,1,0)//kiri;
    glVertex3d(x4,y4,z2);
    glVertex3d(x4,y4,z1);
    glVertex3d(x1,y1,z3);
}

```

Titik yang dipakai dalam sintak diatas.

```

glBegin(GL_TRIANGLES);
cobain(-3,5,0,0,-2,2,-8,2,-2,2,1);
glEnd();

```