

# Modul 5

## Tiga Dimensi

### I. Tugas Pendahuluan

1. Apa beda 2 dimensi dan 3 dimensi?
2. Jelaskan apa itu proyeksi!

### II. Pengantar

Alam fisik dalam persepsi manusia adalah sebuah ruang yang berformat 3 dimensi. Benda-benda yang ada di dalamnya umum direpresentasikan menggunakan format 3 dimensi: panjang, lebar, dan tinggi.

Dalam matematika, 3 dimensi ini biasa dinyatakan dalam sistem koordinat kartesian. Koordinat kartesian 3 dimensi memiliki 3 bidang yang saling tegak lurus satu dengan yang lainnya. Tiap bidang memiliki sumbu yang koordinat yang biasa disebut sumbu x, y, dan z.

### 3 dimensi di OpenGL

OpenGL menggunakan matrix sebagai komponen dasar untuk menghasilkan tampilan pada layar. Semua matrix ini didefinisikan untuk dapat memproses operasi-operasi dalam 3 dimensi.

Jika pada pelajaran-pelajaran sebelumnya obyek dibuat dalam 2 dimensi, sebenarnya obyek-obyek tersebut adalah obyek 3 dimensi. Hanya saja dimensi ketiga diabaikan. Termasuk didalam konsep ini adalah transformasi. Transformasi selalu dilakukan dalam format 3 dimensi.

### Proyeksi

Proyeksi diperlukan untuk menggambarkan obyek 3 dimensi kedalam format 2 dimensi, contohnya: arsitek perlu menuangkan idenya tentang sebuah obyek bangunan 3 dimensi diatas kertas (2 dimensi). Contoh lain adalah penggambaran dunia OpenGL yang 3 dimensi ke layar monitor yang 2 dimensi. Perubahan format dari 3 dimensi menjadi 2 dimensi ini memerlukan proses/aturan khusus. Proses/aturan ini disebut proyeksi grafis.

Ada 2 jenis proyeksi:

1. Perspektif

Cara mata manusia dan kamera menangkap gambar obyek sekelilingnya. Obyek yang jauh terlihat kecil, obyek yang dekat terlihat besar. 2 garis sejajar akan terlihat menyatu di kejauhan.

## 2. Parallel

Garis proyeksi selalu sejajar baik di obyek 3 dimensi maupun di penggambaran 2 dimensinya. Jenis proyeksi ini digunakan oleh orang-orang teknik (Arsitek, teknik mesin, teknik sipil) dalam menggambar pekerjaannya.

### Proyeksi di OpenGL

Dalam OpenGL, diperlukan beberapa perubahan setting agar sebuah aplikasi dapat menampilkan obyek 3 dimensi. Jika pada aplikasi-aplikasi sebelumnya perintah proyeksi yang digunakan adalah `gluOrtho2D`(kiri, kanan, bawah, atas) untuk menampilkan obyek 2 dimensi, perintah yang sama harus diganti agar aplikasi dapat menampilkan obyek 3 dimensi:

1. Untuk proyeksi parallel, gunakan `glOrtho`(kiri, kanan, bawah, atas, dekat, jauh);
2. Untuk proyeksi perspektif, gunakan `gluPerspective`(sudut buka, aspek rasio, dekat, jauh);

Program 5.1 menghasilkan tampilan kotak 3 dimensi.

```
void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    glColor3f(1.0, 1.0, 1.0);
    glRotated(35,1,1,1);
    glutWireCube(2);
    glFlush();
}

void init(void)
{
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-5.0, 5.0, -5.0, 5.0, 5.0, -5.0);
    glMatrixMode(GL_MODELVIEW);
    glShadeModel (GL_FLAT);
    glEnable(GL_DEPTH_TEST);
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Kubus");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
```

**Program 5.1 Kubus 3 dimensi**

Perhatikan perbedaan program diatas dengan program-program sebelumnya:

1. Menggunakan `glOrtho`, bukan `gluOrtho2D`
2. Perintah `glEnable(GL_DEPTH_TEST)` diperlukan disini
3. Modifikasi: `glutInitDisplayMode(GLUT_RGB | GLUT_DEPTH);`

4. Modifikasi: `glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);`
5. Pada beberapa kondisi, diperlukan juga modifikasi:
  - a. `glShadeModel(GL_FLAT);`

Program diatas menggunakan perintah `glutWireCube(ukuran)` untuk membuat sebuah kubus 3 dimensi. Perintah ini adalah perintah bawaan dari glut. Perintah bawaan lain untuk membuat obyek 3 dimensi adalah:

1. `glutWireTeapot(GLdouble size);`
2. `glutWireSphere(GLdouble radius, GLint slices, GLint stacks);`
3. `glutWireCone(GLdouble base, GLdouble height, GLint slices, GLint stacks);`
4. `glutWireTetrahedron(void);`
5. `glutWireOctahedron(void);`

Selain bentuk wireframe diatas, glut menyediakan juga obyek-obyek bawaan 3 dimensi dalam format solid dimana obyek ditampilkan dalam bentok penuh/solid. Format perintahnya sama seperti obyek glut wire diatas hanya tinggal mengganti kata Wire dengan kata Solid:

1. `glutWireCube(GLdouble size) → glutSolidCube(GLdouble size);`
2. `glutWireTeapot(GLdouble size) → glutSolidTeapot(GLdouble size);`
3. `glutWireSphere(GLdouble radius, GLint slices, GLint stacks) → glutSolidSphere(GLdouble radius, GLint slices, GLint stacks)`
4. `glutWireCone(GLdouble base, GLdouble height, GLint slices, GLint stacks) → glutSolidCone(GLdouble base, GLdouble height, GLint slices, GLint stacks)`
5. `glutWireTetrahedron(void) → glutSolidTetrahedron(void)`
6. `glutWireOctahedron(void) → glutSolidOctahedron(void)`

Jika dicoba, obyek-obyek solid ini tampak tidak jelas bentuknya di layar. Yang tampak hanyalah blok berwarna putih. **Ini wajar**. Di modul 10, pencahayaan yang benar akan menampilkan bentuk obyek-obyek ini dengan baik.

Ada kalanya obyek yang ingin dibuat/ditampilkan tidak tersedia dalam library bawaan glut. Untuk kasus seperti ini, programmer perlu membuat sendiri obyek tersebut. Di OpenGL, untuk membuat obyek 3 dimensi, salah satu caranya adalah dengan membuat tiap sisi obyek dari polygon 2 dimensi. Program 5.2 membuat kubus 3 dimensi dimana tiap sisinya dibuat dari persegi 2 dimensi.

```
void display()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glRotated(-35, 1, 1, 1);

    //depan
    glColor3f(0.0, 0.0, 1.0);
```

```

    glBegin(GL_POLYGON);
    glVertex3f(-1.0, -1.0, 1.0);
    glVertex3f(1.0, -1.0, 1.0);
    glVertex3f(1.0, 1.0, 1.0);
    glVertex3f(-1.0, 1.0, 1.0);
    glEnd();

    //belakang
    glColor3f(0.0,1.0,0.0);
    glBegin(GL_POLYGON);
    glVertex3f(1.0, -1.0, -1.0);
    glVertex3f(-1.0, -1.0, -1.0);
    glVertex3f(-1.0, 1.0, -1.0);
    glVertex3f(1.0, 1.0, -1.0);
    glEnd();

    //kiri
    glColor3f(1.0,0.0,0.0);
    glBegin(GL_POLYGON);
    glVertex3f(-1.0, -1.0, -1.0);
    glVertex3f(-1.0, -1.0, 1.0);
    glVertex3f(-1.0, 1.0, 1.0);
    glVertex3f(-1.0, 1.0, -1.0);
    glEnd();

    //kanan
    glColor3f(0.0,1.0,1.0);
    glBegin(GL_POLYGON);
    glVertex3f(1.0, -1.0, -1.0);
    glVertex3f(1.0, 1.0, -1.0);
    glVertex3f(1.0, 1.0, 1.0);
    glVertex3f(1.0, -1.0, 1.0);
    glEnd();

    //bawah
    glColor3f(1.0,0.0,1.0);
    glBegin(GL_POLYGON);
    glVertex3f(1.0, -1.0, 1.0);
    glVertex3f(-1.0, -1.0, 1.0);
    glVertex3f(-1.0, -1.0, -1.0);
    glVertex3f(1.0, -1.0, -1.0);
    glEnd();

    //atas
    glColor3f(1.0,1.0,0.0);
    glBegin(GL_POLYGON);
    glVertex3f(-1.0, 1.0, 1.0);
    glVertex3f(1.0, 1.0, 1.0);
    glVertex3f(1.0, 1.0, -1.0);
    glVertex3f(-1.0, 1.0, -1.0);
    glEnd();

    glFlush();
}

void myinit()
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-3.0,3.0,-3.0,3.0,3.0,-3.0);
    glMatrixMode(GL_MODELVIEW);
    glClearColor(0.0,0.0,0.0,1.0);
    glColor3f(0.0,0.0,0.0);
    glShadeModel(GL_FLAT);
    glEnable(GL_DEPTH_TEST);
}

int main(int argc, char* argv[])
{

```

```

    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(400,400);
    glutInitWindowPosition(100,100);
    glutCreateWindow("Kotak");
    glutDisplayFunc(display);
    myinit();
    glutMainLoop();

    return 0;
}

```

**Program 5.2 Kubus 3 dimensi yang dibangun dari 6 polygon**

Modul ini memiliki kemungkinan untuk membuat beberapa komputer tidak dapat menampilkan bentuk obyek sebagaimana mestinya. Jika memang program sudah benar sementara tampilan tidak juga sesuai, bergabunglah dengan teman atau cari pinjaman komputer.

### III. Percobaan

1. Untuk program 5.1, modifikasi parameter `glRotated`, lalu amati perubahan tampilannya. Kombinasikan dengan memodifikasi parameter pada `glutSolidCube`; amati perubahannya juga.
2. Untuk program 5.1, ganti `glutWireCube` dengan obyek-obyek 3 dimensi bawaan yang lain: baik sesama wire maupun solid. Modifikasi parameter `glRotated`, lalu amati perubahan tampilannya.
3. Kombinasikan dengan memodifikasi parameter pada masing-masing obyek. Amati perubahannya juga.
4. Untuk program 5.2, modifikasi parameter `glRotated`, lalu amati perubahan tampilannya. Kombinasikan dengan perintah transformasi yang lain; amati perubahannya juga.

### IV. Tugas

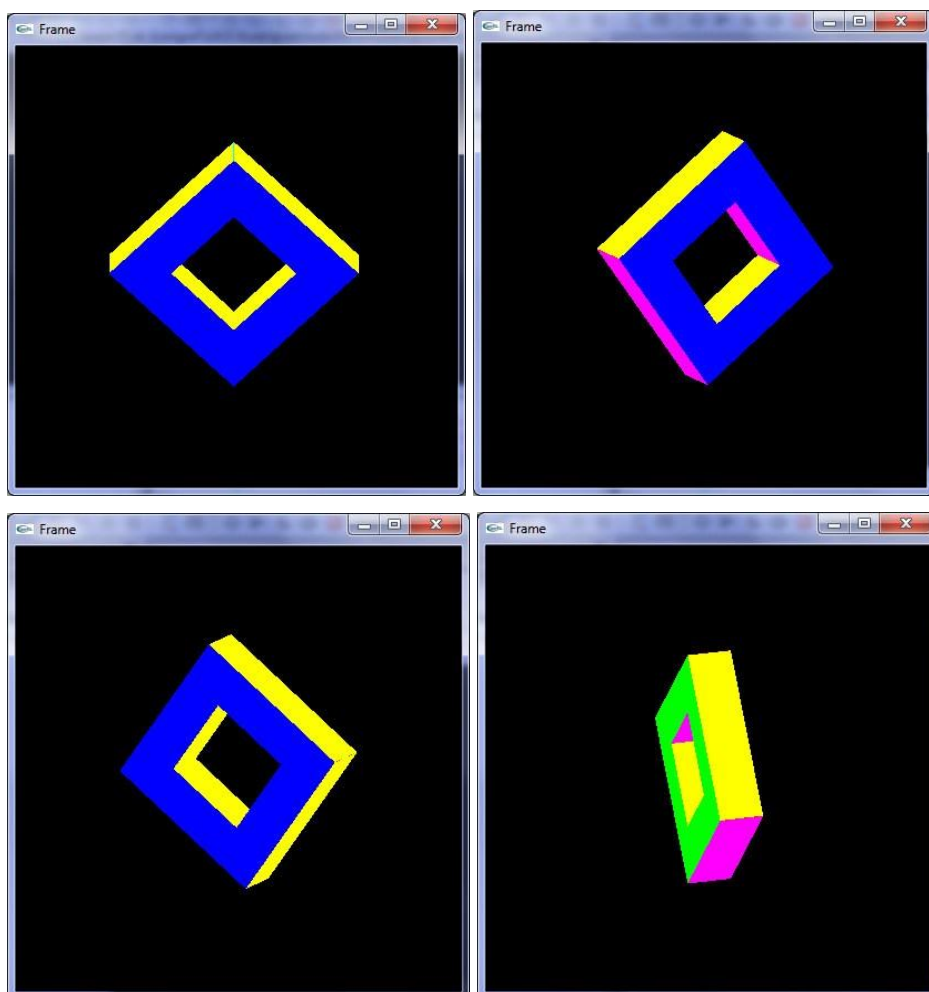
Selesaikan soal-soal berikut:

1. Buat frame yang terdiri dari 4 buah balok sebagai berikut. Gunakan `glutWireCube`.

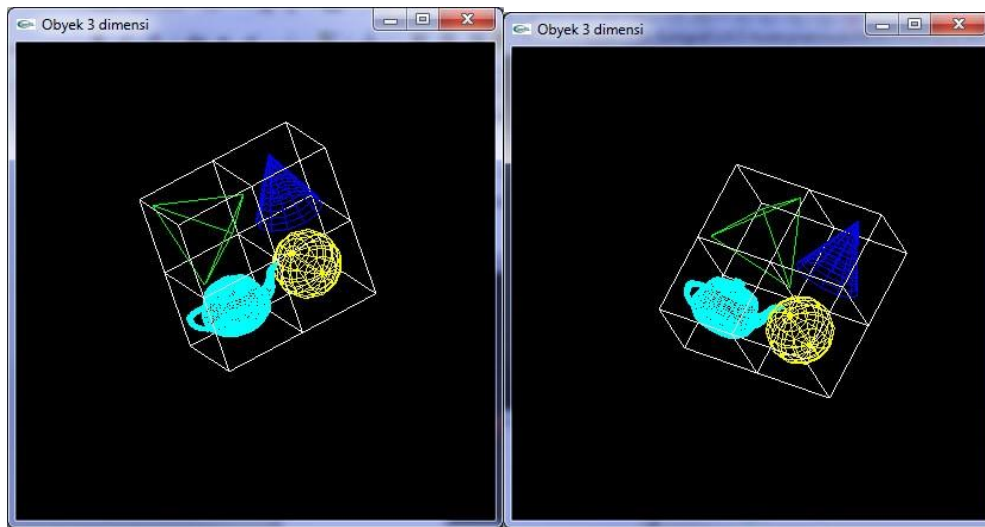


Gambar tidak harus persis, asal cukup mirip

2. Buat frame yang terdiri dari 4 buah balok seperti soal pertama secara manual menggunakan kumpulan polygon.



3. Buat 4 obyek 3 dimensi yang dibahas di modul ini dalam satu tampilan. Keempat obyek tersebut diletakkan dalam 4 kotak terusun sebagai berikut:



## V. Referensi

1. Edward Angel, “Interactive Computer Graphics Sixth Edition”, Pearson, 2012, ch 3, p 115 – 194
2. F. S. Hill, Jr., Stephen M. Kelley, “Computer Graphics Using OpenGL Third Edition”, Prentice Hall, 2007, ch 5, p 190 – 263