

**TECHNICAL REPORT**  
**GRAFIKA KOMPUTER**  
**MODUL 5**



**Disusun Oleh :**

TGL. PRAKTIKUM	: Jum'at, 4 Juni 2021
NAMA	: Achmad Farid Alfa Waid
NRP	: 190411100073
KELOMPOK	: 2
DOSEN	: Ach. Khozaimi, S.Kom., M.Kom

TELAH DISETUJUI TANGGAL :

.....  
ASISTEN PRAKTIKUM

Rizky Alfriansyah  
(180411100068)

**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS TRUNOJOYO MADURA**

# **BAB I**

## **TUJUAN DAN DASAR TEORI**

### **A. TUJUAN**

Mahasiswa dapat mengetahui dan memahami cara melakukan proyeksi object 3 dimensi pada openGL.

### **B. DASAR TEORI**

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 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**

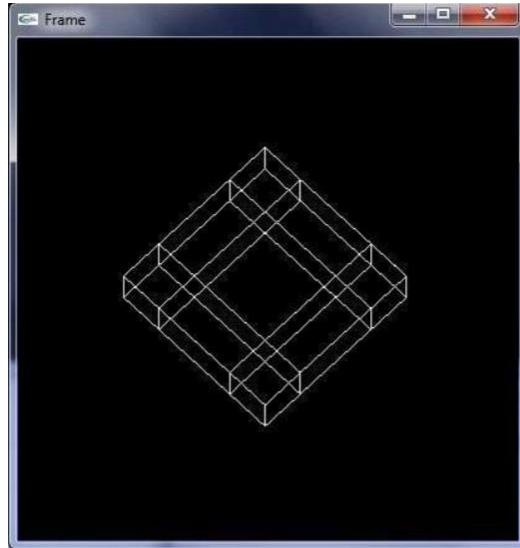
Proyeksi diperlukan untuk menggambarkan obyek 3 dimensi kedalam format 2 dimensi, contohnya: arsitek perlu menuangkan idenya tentang sebuah obyek bangunan 3 dimensi diatas kertas (2 dimensi). Contoh lain adalah penggambaran dunia OpenGL yang 3 dimensi ke layar monitor yang 2 dimensi. Perubahan format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

## BAB II

### PEMBAHASAN

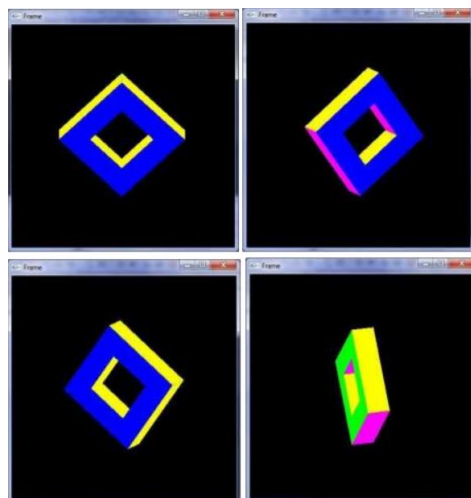
#### A. SOAL

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

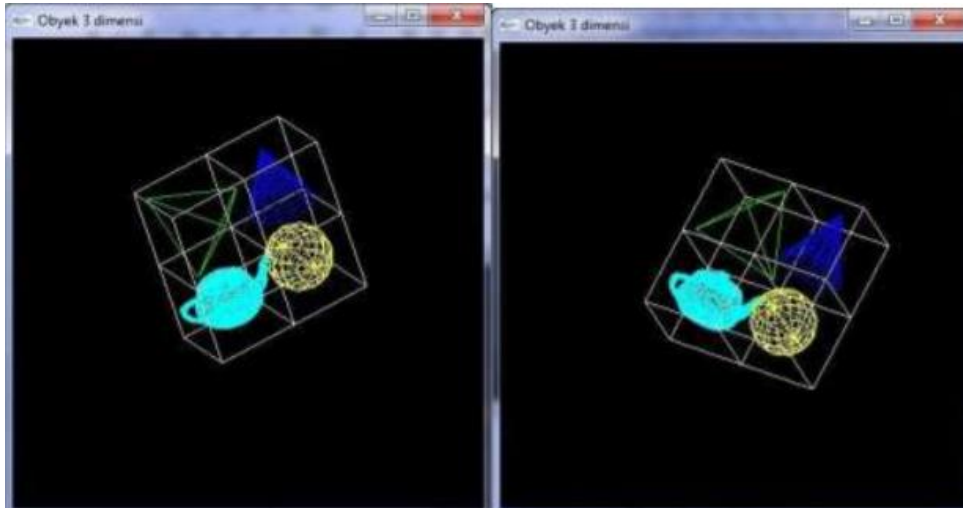


Gambar tidak harus persis, asal cukup mirip

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:



## B. JAWABAN

### 1. Code Program

#### 1) Nomor 1

```
#ifdef _APPLE_
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
#include <stdlib.h>

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    //balok 1
    glPushMatrix();
    glColor3f(1.0, 1.0, 1.0);
    glRotated(10,2,2,0);
    glutWireCube(5);
    glPopMatrix();
    //balok 2
    glPushMatrix();
    glColor3f(1.0, 1.0, 1.0);
    glTranslated(1,1,0);
    glRotated(10,2,2,0);
```

```

        glutWireCube(5);
        glPopMatrix();
        //balok 3
        glPushMatrix();
        glColor3f(1.0, 1.0, 1.0);
        glTranslated(0,1,0);
        glRotated(10,2,2,0);
        glutWireCube(5);
        glPopMatrix();
        //balok 4
        glPushMatrix();
        glColor3f(1.0, 1.0, 1.0);
        glTranslated(1,0,0);
        glRotated(10,2,2,0);
        glutWireCube(5);
        glPopMatrix();

        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("Nomor 1");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```

```
}
```

## 2) Nomor 2

```
#ifdef APPLE
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
#include <stdlib.h>

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(-0.5, -1.0, 0.5);
    glVertex3f(-1.0, -1.0, 0.5);
    glVertex3f(-1.0, 1.0, 0.5);
    glVertex3f(-0.5, 1.0, 0.5);
    glEnd();
    glColor3f(0.0, 0.0, 1.0);
    glBegin(GL_POLYGON);
    glVertex3f(1, -1.0, 0.5);
    glVertex3f(0.5, -1.0, 0.5);
    glVertex3f(0.5, 1.0, 0.5);
    glVertex3f(1, 1.0, 0.5);
    glEnd();
    glColor3f(0.0, 0.0, 1.0);
    glBegin(GL_POLYGON);
    glVertex3f(1, 1.0, 0.5);
    glVertex3f(1, 0.5, 0.5);
    glVertex3f(-1, 0.5, 0.5);
    glVertex3f(-1, 1.0, 0.5);
    glEnd();
    glColor3f(0.0, 0.0, 1.0);
    glBegin(GL_POLYGON);
    glVertex3f(-1, -1.0, 0.5);
    glVertex3f(-1, -0.5, 0.5);
```

```

glVertex3f(1, -0.5, 0.5);
glVertex3f(1, -1.0, 0.5);
glEnd();

//belakang
glColor3f(0.0,1.0,0.0);
glBegin(GL_POLYGON);
glVertex3f(-0.5, -1.0, -0.5);
glVertex3f(-1.0, -1.0, -0.5);
glVertex3f(-1.0, 1.0, -0.5);
glVertex3f(-0.5, 1.0, -0.5);
glEnd();
glColor3f(0.0,1.0,0.0);
glBegin(GL_POLYGON);
glVertex3f(1, -1.0, -0.5);
glVertex3f(0.5, -1.0, -0.5);
glVertex3f(0.5, 1.0, -0.5);
glVertex3f(1, 1.0, -0.5);
glEnd();
glColor3f(0.0,1.0,0.0);
glBegin(GL_POLYGON);
glVertex3f(1, 1.0, -0.5);
glVertex3f(1, 0.5, -0.5);
glVertex3f(-1, 0.5, -0.5);
glVertex3f(-1, 1.0, -0.5);
glEnd();
glColor3f(0.0,1.0,0.0);
glBegin(GL_POLYGON);
glVertex3f(-1, -1.0, -0.5);
glVertex3f(-1, -0.5, -0.5);
glVertex3f(1, -0.5, -0.5);
glVertex3f(1, -1.0, -0.5);
glEnd();

//kiri kuning
glColor3f(1.0,1.0,0.0);
glBegin(GL_POLYGON);
glVertex3f(-1.0, -1.0, -0.5);
glVertex3f(-1.0, -1.0, 0.5);
glVertex3f(-1.0, 1.0, 0.5);

```

```
glVertex3f(-1.0, 1.0, -0.5);
glEnd();
glColor3f(1.0,1.0,0.0);
glBegin(GL_POLYGON);
glVertex3f(-0.5, -0.5, -0.5);
glVertex3f(-0.5, -0.5, 0.5);
glVertex3f(-0.5, 0.5, 0.5);
glVertex3f(-0.5, 0.5, -0.5);
glEnd();
```

```
//kanan ungu
glColor3f(1.0,0.0,1.0);
glBegin(GL_POLYGON);
glVertex3f(1.0, -1.0, -0.5);
glVertex3f(1.0, 1.0, -0.5);
glVertex3f(1.0, 1.0, 0.5);
glVertex3f(1.0, -1.0, 0.5);
glEnd();
glColor3f(1.0,0.0,1.0);
glBegin(GL_POLYGON);
glVertex3f(1.0, -1.0, 0.5);
glVertex3f(-1.0, -1.0, 0.5);
glVertex3f(-1.0, -1.0, -0.5);
glVertex3f(1.0, -1.0, -0.5);
glEnd();
```

```
//bawah dalam
glColor3f(1.0,0.0,1.0);
glBegin(GL_POLYGON);
glVertex3f(0.5, -0.5, -0.5);
glVertex3f(0.5, 0.5, -0.5);
glVertex3f(0.5, 0.5, 0.5);
glVertex3f(0.5, -0.5, 0.5);
glEnd();
glColor3f(1.0,0.0,1.0);
glBegin(GL_POLYGON);
glVertex3f(0.5, -0.5, 0.5);
glVertex3f(-0.5, -0.5, 0.5);
glVertex3f(-0.5, -0.5, -0.5);
```



```

    glVertex3f(0.5, -0.5, -0.5);
    glEnd();

    //atas dalam
    glColor3f(1.0,1.0,0.0);
    glBegin(GL_POLYGON);
    glVertex3f(-1.0, 1.0, 0.5);
    glVertex3f(1.0, 1.0, 0.5);
    glVertex3f(1.0, 1.0, -0.5);
    glVertex3f(-1.0, 1.0, -0.5);
    glEnd();
    glColor3f(1.0,1.0,0.0);
    glBegin(GL_POLYGON);
    glVertex3f(-0.5, 0.5, 0.5);
    glVertex3f(0.5, 0.5, 0.5);
    glVertex3f(0.5, 0.5, -0.5);
    glVertex3f(-0.5, 0.5, -0.5);
    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("Nomer 2");

```

```

        glutDisplayFunc(display);
        myinit();
        glutMainLoop();
        return 0;
    }

```

### 3) Nomor 3

```

#ifdef APPLE
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif
#include <stdlib.h>
void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    // teapot
    glPushMatrix();
    glColor3f(1.0, 1.0, 1.0);
    glTranslated(-1,-1,0);
    glRotated(10,2,2,0);
    glutWireCube(2);
    glColor3f(0.0, 1.0, 1.0);
    glutWireTeapot(0.5);
    glPopMatrix();

    //sphere
    glPushMatrix();
    glColor3f(1.0, 1.0, 1.0);
    glTranslated(1,-1,0);
    glRotated(10,2,2,0);
    glutWireCube(2);
    glColor3f(1.0, 1.0, 0.0);
    glutWireSphere(0.8,10,10);
    glPopMatrix();

    //cone
    glPushMatrix();
    glColor3f(1.0, 1.0, 1.0);
    glTranslated(1,1,0);
    glRotated(10,2,2,0);

```

```

        glutWireCube(2);
        glRotated(60,1,3,1);
        glColor3f(0.0, 0.0, 1.0);
        glutWireCone(0.6,1,10,5);
        glPopMatrix();

        //tetrahedon
        glPushMatrix();
        glColor3f(1.0, 1.0, 1.0);
        glTranslated(-1,1,0);
        glRotated(10,2,2,0);
        glutWireCube(2);
        glRotated(60,2,2,0);
        glColor3f(0.0, 1.0, 0.0);
        glutWireTetrahedron();
        glPopMatrix();
        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("Nomer 3");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```

## 2. Penjelasan Code Program

### 1) Nomor 1

```
//balok 1
    glPushMatrix();
    glColor3f(1.0, 1.0, 1.0);
    glRotated(10,2,2,0);
    glutWireCube(5);
    glPopMatrix();
//balok 2
    glPushMatrix();
    glColor3f(1.0, 1.0, 1.0);
    glTranslated(1,1,0);
    glRotated(10,2,2,0);
    glutWireCube(5);
    glPopMatrix();
//balok 3
    glPushMatrix();
    glColor3f(1.0, 1.0, 1.0);
    glTranslated(0,1,0);
    glRotated(10,2,2,0);
    glutWireCube(5);
    glPopMatrix();
//balok 4
    glPushMatrix();
    glColor3f(1.0, 1.0, 1.0);
    glTranslated(1,0,0);
    glRotated(10,2,2,0);
    glutWireCube(5);
    glPopMatrix();
```

Memanggil fungsi glutWireCube untuk membuat sebuah balok, pemanggilan tersebut dilakukan sebanyak 4 kali dan masing – masing diletakkan di dalam glPushMatrix dan glPopMatrix agar tidak ada perubahan titik koordinat awal. Kemudian saat pemanggilan glutWireCube didalamnya dikasih angka untuk mrmberi ukuran pada balok yang akan ditampilkan.

2) Nomor 2

//depan

```
glColor3f(0.0,0.0,1.0);
glBegin(GL_POLYGON);
glVertex3f(-0.5, -1.0, 0.5);
glVertex3f(-1.0, -1.0, 0.5);
glVertex3f(-1.0, 1.0, 0.5);
glVertex3f(-0.5, 1.0, 0.5);
glEnd();
glColor3f(0.0,0.0,1.0);
glBegin(GL_POLYGON);
glVertex3f(1, -1.0, 0.5);
glVertex3f(0.5, -1.0, 0.5);
glVertex3f(0.5, 1.0, 0.5);
glVertex3f(1, 1.0, 0.5);
glEnd();
glColor3f(0.0,0.0,1.0);
glBegin(GL_POLYGON);
glVertex3f(1, 1.0, 0.5);
glVertex3f(1, 0.5, 0.5);
glVertex3f(-1, 0.5, 0.5);
glVertex3f(-1, 1.0, 0.5);
glEnd();
glColor3f(0.0,0.0,1.0);
```

```
glBegin(GL_POLYGON);  
glVertex3f(-1, -1.0, 0.5);  
glVertex3f(-1, -0.5, 0.5);  
glVertex3f(1, -0.5, 0.5);  
glVertex3f(1, -1.0, 0.5);  
glEnd();
```

Membuat tampilan bagian depan object utama yang dibuat dengan warna biru.

//belakang

```
glColor3f(0.0,1.0,0.0);  
glBegin(GL_POLYGON);  
glVertex3f(-0.5, -1.0, -0.5);  
glVertex3f(-1.0, -1.0, -0.5);  
glVertex3f(-1.0, 1.0, -0.5);  
glVertex3f(-0.5, 1.0, -0.5);  
glEnd();  
glColor3f(0.0,1.0,0.0);  
glBegin(GL_POLYGON);  
glVertex3f(1, -1.0, -0.5);  
glVertex3f(0.5, -1.0, -0.5);  
glVertex3f(0.5, 1.0, -0.5);  
glVertex3f(1, 1.0, -0.5);  
glEnd();  
glColor3f(0.0,1.0,0.0);  
glBegin(GL_POLYGON);  
glVertex3f(1, 1.0, -0.5);  
glVertex3f(1, 0.5, -0.5);  
glVertex3f(-1, 0.5, -0.5);  
glVertex3f(-1, 1.0, -0.5);  
glEnd();
```

```
glColor3f(0.0,1.0,0.0);  
glBegin(GL_POLYGON);  
glVertex3f(-1, -1.0, -0.5);  
glVertex3f(-1, -0.5, -0.5);  
glVertex3f(1, -0.5, -0.5);  
glVertex3f(1, -1.0, -0.5);  
glEnd();
```

Kemudian membuat bagian belakang object dengan warna hijau.

//kiri kuning

```
glColor3f(1.0,1.0,0.0);  
glBegin(GL_POLYGON);  
glVertex3f(-1.0, -1.0, -0.5);  
glVertex3f(-1.0, -1.0, 0.5);  
glVertex3f(-1.0, 1.0, 0.5);  
glVertex3f(-1.0, 1.0, -0.5);  
glEnd();  
glColor3f(1.0,1.0,0.0);  
glBegin(GL_POLYGON);  
glVertex3f(-0.5, -0.5, -0.5);  
glVertex3f(-0.5, -0.5, 0.5);  
glVertex3f(-0.5, 0.5, 0.5);  
glVertex3f(-0.5, 0.5, -0.5);  
glEnd();
```

Membuat object pada bagian kiri dan atas dengan warna kuning.

//kanan ungu

```
glColor3f(1.0,0.0,1.0);  
glBegin(GL_POLYGON);  
glVertex3f(1.0, -1.0, -0.5);  
glVertex3f(1.0, 1.0, -0.5);  
glVertex3f(1.0, 1.0, 0.5);
```

```
glVertex3f(1.0, -1.0, 0.5);  
glEnd();  
glColor3f(1.0,0.0,1.0);  
glBegin(GL_POLYGON);  
glVertex3f(1.0, -1.0, 0.5);  
glVertex3f(-1.0, -1.0, 0.5);  
glVertex3f(-1.0, -1.0, -0.5);  
glVertex3f(1.0, -1.0, -0.5);  
glEnd();
```

Membuat object pada bagian kanan dan bawah dengan warna ungu.

//bawah dalam

```
glColor3f(1.0,0.0,1.0);  
glBegin(GL_POLYGON);  
glVertex3f(0.5, -0.5, -0.5);  
glVertex3f(0.5, 0.5, -0.5);  
glVertex3f(0.5, 0.5, 0.5);  
glVertex3f(0.5, -0.5, 0.5);  
glEnd();  
glColor3f(1.0,0.0,1.0);  
glBegin(GL_POLYGON);  
glVertex3f(0.5, -0.5, 0.5);  
glVertex3f(-0.5, -0.5, 0.5);  
glVertex3f(-0.5, -0.5, -0.5);  
glVertex3f(0.5, -0.5, -0.5);  
glEnd();
```

Membuat object pada bagian bawah dalam dengan warna ungu.

//atas dalam

```
glColor3f(1.0,1.0,0.0);  
glBegin(GL_POLYGON);  
glVertex3f(-1.0, 1.0, 0.5);
```



```

glVertex3f(1.0, 1.0, 0.5);
glVertex3f(1.0, 1.0, -0.5);
glVertex3f(-1.0, 1.0, -0.5);
glEnd();
glColor3f(1.0,1.0,0.0);
glBegin(GL_POLYGON);
glVertex3f(-0.5, 0.5, 0.5);
glVertex3f(0.5, 0.5, 0.5);
glVertex3f(0.5, 0.5, -0.5);
glVertex3f(-0.5, 0.5, -0.5);
glEnd();

```

Membuat object pada bagian atas dalam dengan warna kuning.

### 3) Nomor 3

```

// teapot
glPushMatrix();
glColor3f(1.0, 1.0, 1.0);
glTranslated(-1,-1,0);
glRotated(10,2,2,0);
glutWireCube(2);
glColor3f(0.0, 1.0, 1.0);
glutWireTeapot(0.5);
glPopMatrix();

```

Membuat object teapot dengan memanggil fungsi glutWireTeapot yang diletakkan didalam fungsi glutWireCube dan fung ini diletakkan di dalam glPushMatrix dan glPopMatrix, kemudian untuk pewarnaan glutWireCube menggunakan warna putih, sedangkan glutWireTeapot menggunakan warna biru muda. Untuk letak atau posisi koordinan ditentukan dengan menggunakan glTranslated kemudian dirotate dengan glRotated.

```
//sphere
    glPushMatrix();
    glColor3f(1.0, 1.0, 1.0);
    glTranslated(1,-1,0);
    glRotated(10,2,2,0);
    glutWireCube(2);
    glColor3f(1.0, 1.0, 0.0);
    glutWireSphere(0.8,10,10);
    glPopMatrix();
```

Membuat object sphere dengan memanggil fungsi glutWireSphere yang diletakkan didalam fungsi glutWireCube dan fungsi ini diletakkan di dalam glPushMatrix dan glPopMatrix, kemudian untuk pewarnaan glutWireCube menggunakan warna putih, sedangkan glutWireSphere menggunakan warna kuning. Untuk letak atau posisi koordinan ditentukan dengan menggunakan glTranslated kemudian dirotate dengan glRotated.

```
//cone
    glPushMatrix();
    glColor3f(1.0, 1.0, 1.0);
    glTranslated(1,1,0);
    glRotated(10,2,2,0);
    glutWireCube(2);
    glRotated(60,1,3,1);
    glColor3f(0.0, 0.0, 1.0);
    glutWireCone(0.6,1,10,5);
    glPopMatrix();
```

Membuat object cone dengan memanggil fungsi glutWireCone yang diletakkan didalam fungsi glutWireCube dan fungsi ini diletakkan di dalam glPushMatrix dan glPopMatrix, kemudian untuk pewarnaan glutWireCube menggunakan warna putih, sedangkan glutWireCone

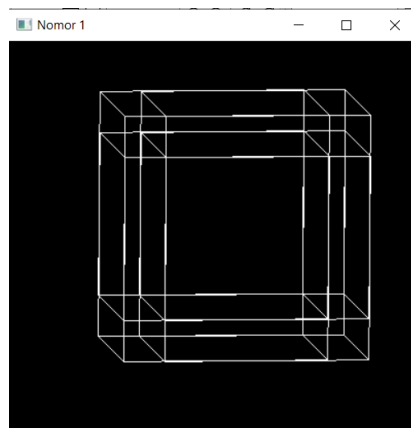
menggunakan warna biru. Untuk letak atau posisi koordinan ditentukan dengan menggunakan `glTranslated` kemudian dirotate dengan `glRotated`.

```
//tetrahedon
glPushMatrix();
glColor3f(1.0, 1.0, 1.0);
glTranslated(-1,1,0);
glRotated(10,2,2,0);
glutWireCube(2);
glRotated(60,2,2,0);
glColor3f(0.0, 1.0, 0.0);
glutWireTetrahedron();
glPopMatrix();
```

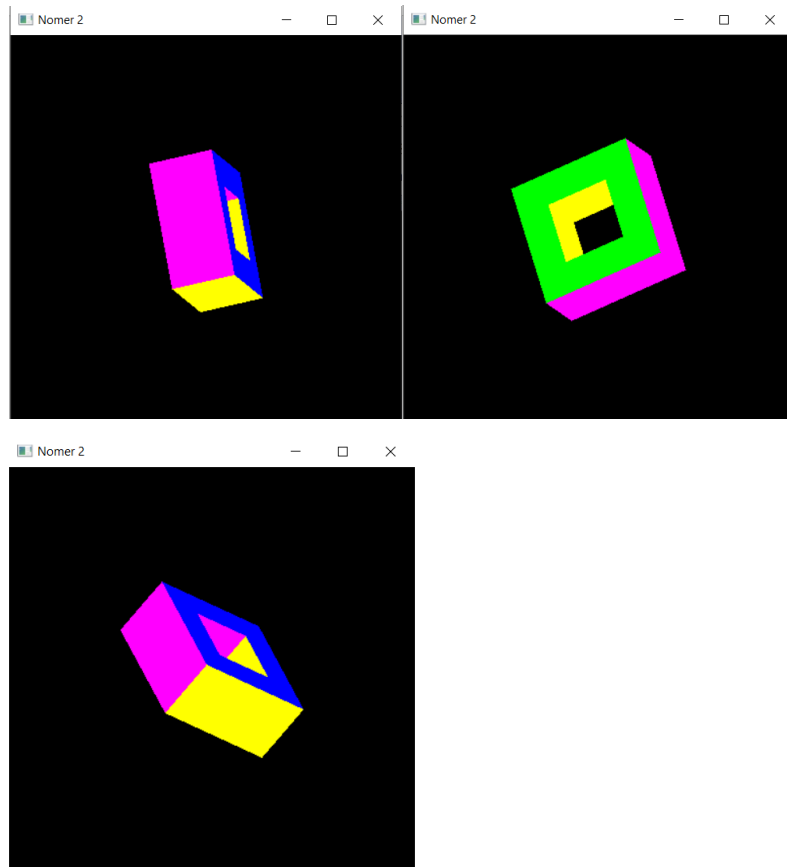
Membuat object tetrahedon dengan memanggil fungsi `glutWireTetrahedron` yang diletakkan didalam fungsi `glutWireCube` dan fungsi ini diletakkan di dalam `glPushMatrix` dan `glPopMatrix`, kemudian untuk pewarnaan `glutWireCube` menggunakan warna putih, sedangkan `glutWireTetrahedron` menggunakan warna hijau. Untuk letak atau posisi koordinan ditentukan dengan menggunakan `glTranslated` kemudian dirotate dengan `glRotated`.

### 3. Hasil Running Program

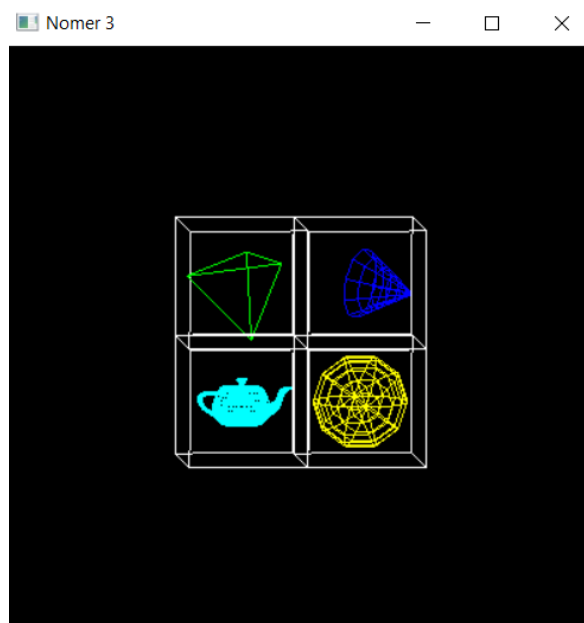
#### 1) Nomor 1



## 2) Nomor 2



## 3) Nomor 3



## **BAB III**

### **PENUTUP**

#### **A. Kesimpulan**

1. Mahasiswa mampu object 3 dimensi sederhana, baik secara manual ataupun menggunakan object dasar bawaan glut atau codeblocks.
2. Proyeksi diperlukan untuk menggambarkan obyek 3 dimensi ke dalam format 2 dimensi.
3. 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

#### **B. Saran**

Diperlukan banyak latihan dan mencoba megotak-atik sendiri agar mahasiswa lebih memahami dan mendalami fungsi – fungsi yang ada pada openGL.