

**PRAKTIKUM GRAFIK KOMPUTER**  
**TUGAS PERTEMUAN 5 – Proyeksi 3D**

“Disusun Untuk Memenuhi Tugas Mata Kuliah Praktikum Grafik Komputer”

Dosen pengampu : Sri Rahayu, S.T. M.Kom



Disusun Oleh :

Yoga Agustiansyah

2206050

Teknik Informatika B

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**INSTITUT TEKNOLOGI GARUT**

**2023**

## **PENDAHULUAN**

Proyeksi 3Dimensi adalah teknik yang digunakan untuk menggambarkan objek 3D ke dalam bidang 2D, sehingga dapat ditampilkan di layar komputer. Dalam dunia grafik komputer, terdapat dua jenis proyeksi utama: proyeksi ortografis dan proyeksi perspektif. Proyeksi ortografis memproyeksikan objek dengan garis sejajar, sementara proyeksi perspektif mencoba menirukan cara mata manusia melihat dunia nyata. Pemahaman tentang proyeksi ini merupakan langkah awal dalam pengembangan aplikasi 3D yang mencakup permainan, animasi, desain arsitektur, dan sebagainya. Dalam praktikum ini, kita akan menjelajahi konsep ini dan mengimplementasikannya menggunakan perangkat lunak grafik komputer seperti OpenGL atau DirectX.

## PEMBAHASAN

### A. Soal 1

Untuk program 1 modifikasi parameter `glRotated`, lalu amati perubahan tampilannya.

#### 1) Source Code Asal

```
#include <GL/glut.h>
#include <math.h>

void display (void) {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glColor3f(0.0, 0.850, 1);
    glRotated(35, 1, 1, 1);
    glutWireCube(2);
    glFlush();
}

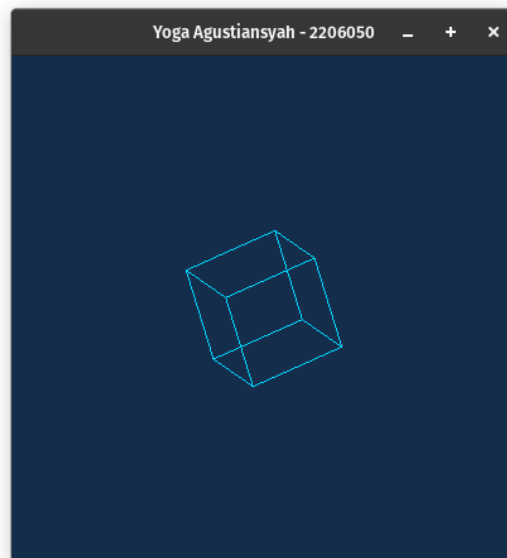
void init (void) {
    glClearColor(0.078, 0.178, 0.294, 1.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-5, 5, -5, 5, -5, 5);
    glMatrixMode(GL_MODELVIEW);
    glShadeModel(GL_FLAT);
    glEnable(GL_DEPTH_TEST);
}

void mouseClicked(int button, int state, int x, int y) {
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
        // Memanggil fuction display saat klik kiri mouse
        display();
        // Meminta glut untuk merender ulang tampilan
        glutPostRedisplay();
    }
}

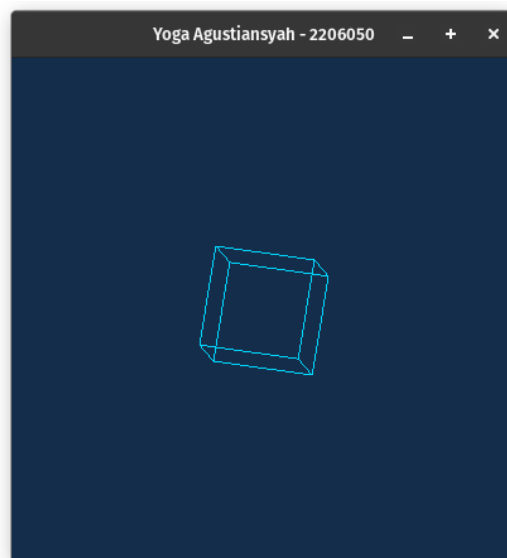
int main(int argc, char*argv[]){
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Yoga Agustiansyah - 2206050");
    glutDisplayFunc(display);
    init();
    glutMouseFunc(mouseClicked);
    glutMainLoop();
    return 0;
}
```

**Output**

**Saat program dijalankan**



**Saat dirotasi satukali**



## 2) Source Code Setelah dimodifikasi

```
#include <GL/glut.h>
#include <math.h>

void display (void) {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glColor3f(0.0, 0.850, 1);
    glRotated(50, 1, 1, 1);
    glutWireCube(2);
    glFlush();
}

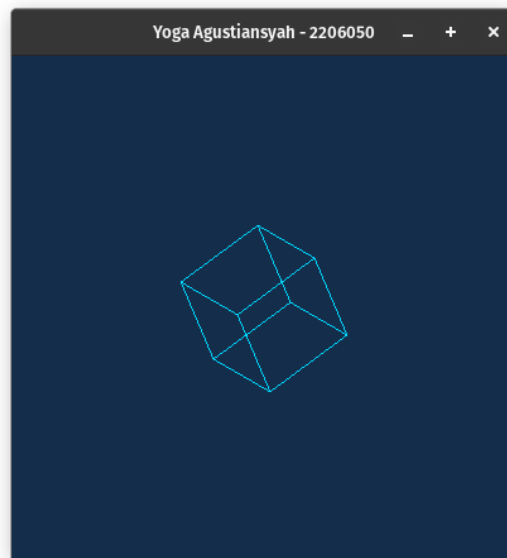
void init (void) {
    glClearColor(0.078, 0.178, 0.294, 1.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-5, 5, -5, 5, -5, 5);
    glMatrixMode(GL_MODELVIEW);
    glShadeModel(GL_FLAT);
    glEnable(GL_DEPTH_TEST);
}

void mouseClicked(int button, int state, int x, int y) {
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
        // Memanggil fuction display saat klik kiri mouse
        display();
        // Meminta glut untuk merender ulang tampilan
        glutPostRedisplay();
    }
}

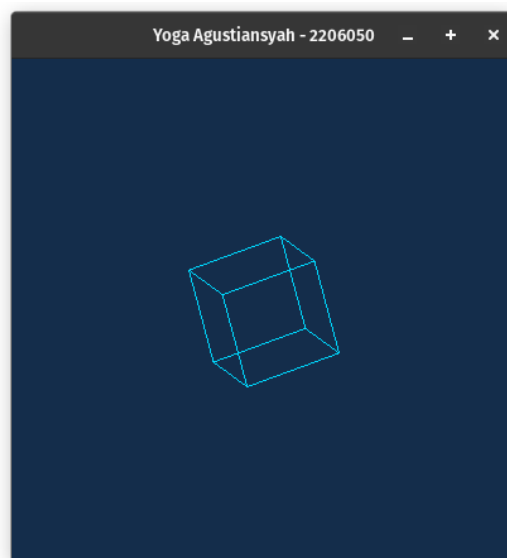
int main(int argc, char*argv[]){
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Yoga Agustiansyah - 2206050");
    glutDisplayFunc(display);
    init();
    glutMouseFunc(mouseClick);
    glutMainLoop();
    return 0;
}
```

## 2) Output

Saat program dijalankan



Saat dirotasi satukali



## 3) Penjelasan

Kedua program tersebut sama-sama menggambar kubus dan memiliki latar belakang yang sama, menggunakan proyeksi ortografik dan shading flat. Yang membedakan keduanya hanyalah besaran rotasi yang diterapkan pada kubus. Program pertama merotasi kubus sebesar 35 derajat, sementara program kedua merotasi kubus sebesar 50 derajat.

Dengan demikian, program kedua (yang merotasi objek sebesar 50 derajat) akan dirotasi dengan sudut rotasi yang lebih besar daripada program pertama (yang merotasi objek sebesar 35 derajat). Hal ini berarti bahwa program kedua akan mulai dengan objek yang lebih mendekati sudut maksimum rotasi, sedangkan program pertama akan mulai dengan sudut yang lebih rendah.

## B. Soal 2

Untuk program 1 ganti glutWireCube dengan Object yang lain dan rubah ukuranya , lalu amati perubahan tampilannya.

### 1) Source Code Asal

```
#include <GL/glut.h>
#include <math.h>

void display (void) {
    glClearColor(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glColor3f(0.0, 0.850, 1);
    glRotated(35, 1, 1, 1);
    glutWireCube(2);
    glFlush();
}

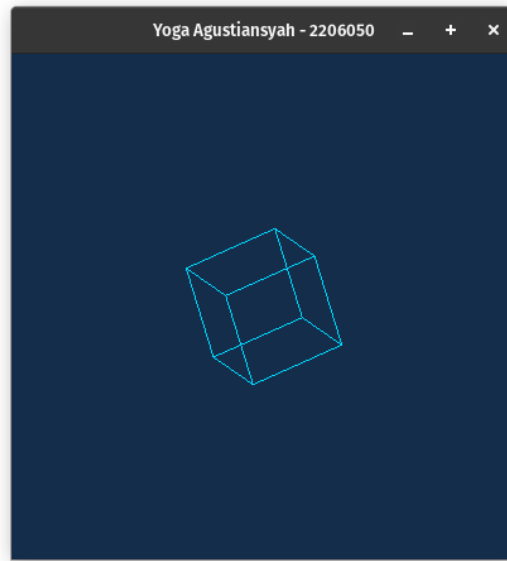
void init (void) {
    glClearColor(0.078, 0.178, 0.294, 1.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-5, 5, -5, 5, -5, 5);
    glMatrixMode(GL_MODELVIEW);
    glShadeModel(GL_FLAT);
    glEnable(GL_DEPTH_TEST);
}

void mouseClick(int button, int state, int x, int y) {
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
        // Memanggil fuction display saat klik kiri mouse
        display();
        // Meminta glut untuk merender ulang tampilan
        glutPostRedisplay();
    }
}

int main(int argc, char*argv[]){
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Yoga Agustiansyah - 2206050");
    glutDisplayFunc(display);
    init();
    glutMouseFunc(mouseClick);
    glutMainLoop();
    return 0;
}
```



## Output



## 2) Source Code Setelah dimodifikasi

```
#include <GL/glut.h>
#include <math.h>

void display (void) {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glColor3f(0.0, 0.850, 1);
    glRotated(35, 1, 1, 1);
    glutWireTorus(1, 2, 25, 25);
    glFlush();
}

void init (void) {
    glClearColor(0.078, 0.178, 0.294, 1.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-5, 5, -5, 5, -5, 5);
    glMatrixMode(GL_MODELVIEW);
    glShadeModel(GL_FLAT);
    glEnable(GL_DEPTH_TEST);
}

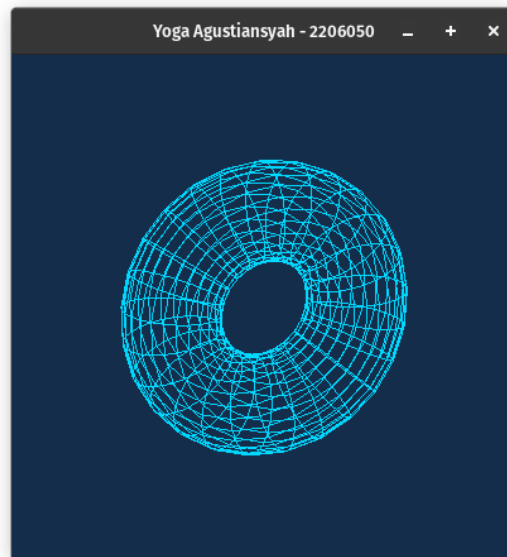
void mouseClicked(int button, int state, int x, int y) {
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
        // Memanggil fuction display saat klik kiri mouse
        display();
        // Meminta glut untuk merender ulang tampilan
        glutPostRedisplay();
    }
}
```

```

int main(int argc, char*argv[]){
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Yoga Agustiansyah - 2206050");
    glutDisplayFunc(display);
    init();
    glutMouseFunc(mouseClick);
    glutMainLoop();
    return 0;
}

```

## 2) Output



## 3) Penjelasan

Program dimodifikasi untuk menggambar sebuah torus (cincin 3 dimensi) menggunakan sintaks `glutWireTorus(1, 2, 25, 25)`. Fungsi ini digunakan untuk menggambar torus dalam mode "kawat" atau garis-garis yang menggambarkan kerangka torus tanpa diisi dengan warna atau tekstur.

**GlutWireTorus** memiliki beberapa parameter yang perlu diisi, yaitu:

- **innerRadius**: Parameter ini adalah jari-jari torus bagian dalam. Ini menentukan radius dari "pemotongan" torus di bagian tengahnya. Dalam program ini, torus akan memiliki jari-jari luar (outer radius) sebesar 1 unit.

- **outerRadius:** Parameter ini adalah jari-jari torus bagian luar. Ini menentukan radius dari torus secara keseluruhan. Dalam program ini, torus akan memiliki jari-jari luar (outer radius) sebesar 2 unit.
- **sides:** Parameter ini adalah jumlah "sisi" atau "slice" dalam torus. Ini mengontrol jumlah lingkaran horizontal di sekitar torus. Dalam program ini, torus akan memiliki 25 slice horizontal. Semakin besar nilai ini, semakin banyak lingkaran horizontal yang akan terbentuk, sehingga torus akan terlihat lebih halus.
- **rings:** Parameter ini adalah jumlah "ring" atau "stack" dalam torus. Ini mengontrol jumlah lingkaran vertikal di sekitar torus. Dalam program ini, torus akan memiliki 25 stack vertikal. Semakin besar nilai ini, semakin banyak lingkaran vertikal yang akan terbentuk, sehingga torus akan terlihat lebih halus secara vertikal.

### C. Soal 3

Untuk program 2, modifikasi parameter `glRotated`, lalu amati perubahan tampilannya.

#### 1) Source Code Asal

```
#include <GL/glut.h>
#include <math.h>
void display () {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glColor3f(0.0, 0.850, 1);

    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, 3, -3, 3, -3, 3);
    glMatrixMode(GL_MODELVIEW);
    glClearColor(0, 0, 0, 1);
    glColor3f(0, 0, 0);
    glShadeModel(GL_FLAT);
    glEnable(GL_DEPTH_TEST);
}

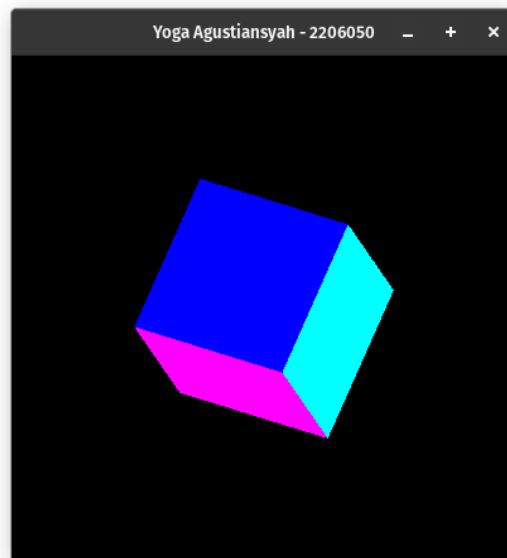
void mouseClicked(int button, int state, int x, int y) {
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
        // Memanggil fuction display saat klik kiri mouse
        display();
        // Meminta glut untuk merender ulang tampilan
        glutPostRedisplay();
    }
}

int main(int argc, char*argv[]){
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB |
GLUT_DEPTH);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Yoga Agustiansyah - 2206050");
    glutDisplayFunc(display);
    myInit();
    glutMouseFunc(mouseClick);
    glutMainLoop();
    return 0;
}

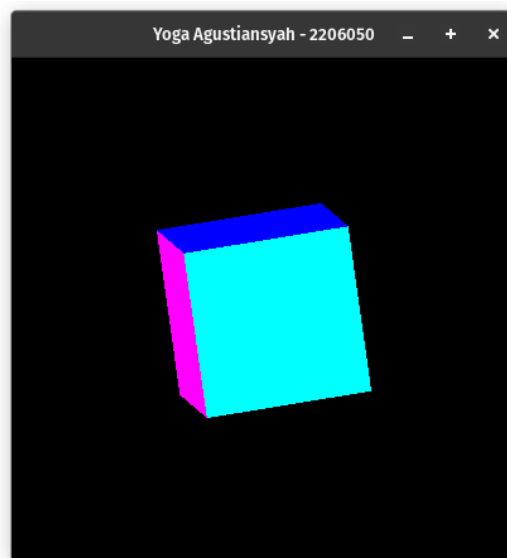
```

Output

Saat program dijalankan



Saat dirotasi satukali



## 2) Source Code Setelah dimodifikasi

```
#include <GL/glut.h>
#include <math.h>

void display () {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glColor3f(0.0, 0.850, 1);

    glRotated(-100, 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();

    glColor3f(0.0,1.0,0.0);
    // belakang
    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, 3, -3, 3, -3, 3);
        glMatrixMode(GL_MODELVIEW);
        glClearColor(0, 0, 0, 1);
        glColor3f(0, 0, 0);
        glShadeModel(GL_FLAT);
        glEnable(GL_DEPTH_TEST);
    }

    void mouseClicked(int button, int state, int x, int y) {
        if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
            // Memanggil fuction display saat klik kiri mouse
            display();
            // Meminta glut untuk merender ulang tampilan
            glutPostRedisplay();
        }
    }

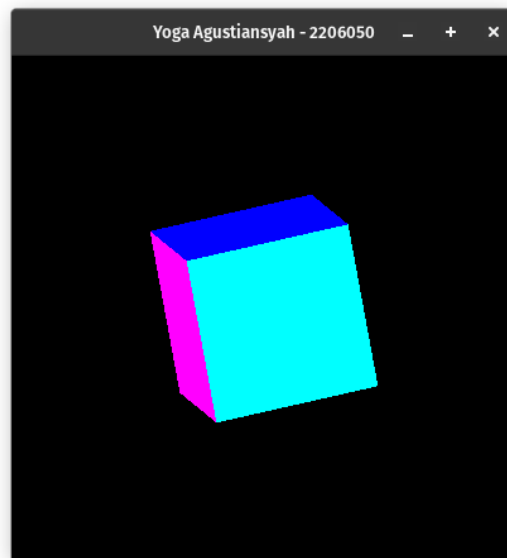
    int main(int argc, char*argv[]){
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB |
        GLUT_DEPTH);
        glutInitWindowSize(400, 400);
        glutInitWindowPosition(100, 100);
        glutCreateWindow("Yoga Agustiansyah - 2206050");
        glutDisplayFunc(display);
        myInit();
        glutMouseFunc(mouseClick);
        glutMainLoop();
        return 0;
    }

```

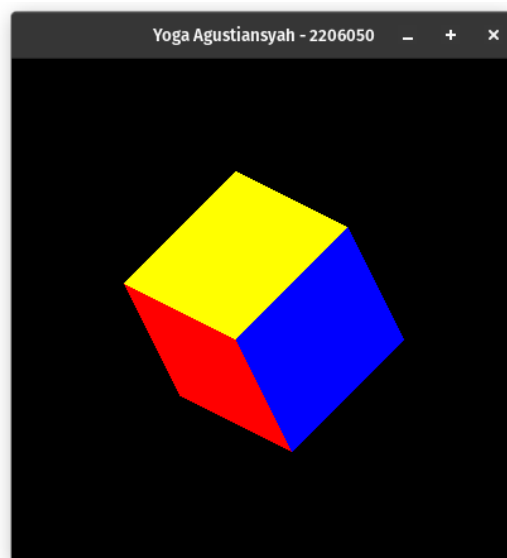


## 2) Output

Saat program dijalankan



Saat dirotasi satukali



## 3) Penjelasan

Kedua program tersebut sama-sama menggambar berbagai segi enam atau hexahedron dengan berbagai warna. Ini menciptakan tampilan dari berbagai sisi dari objek ini dengan berbagai warna, semua hexagon tersebut membentuk sebuah kubus. Yang membedakan keduanya hanyalah besaran rotasi yang diterapkan.

Program pertama merotasi kubus sebesar 35 derajat, sementara program kedua merotasi kubus sebesar 100 derajat.

Dengan demikian, program kedua (yang merotasi objek sebesar 100 derajat) akan dirotasi dengan sudut rotasi yang lebih besar daripada program pertama (yang merotasi objek sebesar 35 derajat). Hal ini berarti bahwa program kedua akan mulai dengan objek yang lebih mendekati sudut maksimum rotasi, sedangkan program pertama akan mulai dengan sudut yang lebih rendah.

## KESIMPULAN

Pada praktikum mata kuliah Grafik Komputer yang membahas proyeksi 3D, kita telah menjalankan beberapa program yang memungkinkan kita untuk memahami konsep dasar proyeksi 3D dan implementasinya menggunakan perangkat lunak grafik komputer. Berikut adalah beberapa kesimpulan yang dapat diambil dari praktikum ini:

1. **Pemahaman Proyeksi 3D:** Praktikum ini memberikan pemahaman dasar tentang proyeksi 3D, yang merupakan teknik penting dalam menggambarkan objek tiga dimensi ke dalam dunia dua dimensi.
2. **Penerapan Proyeksi Ortografis:** Program pertama mengilustrasikan proyeksi ortografis dengan rotasi objek berbentuk kubus dalam ruang tiga dimensi. Ini memungkinkan pemahaman tentang proyeksi dengan garis sejajar dan cara mengatur perspektif.
3. **Penerapan Proyeksi Perspektif:** Program kedua menggunakan proyeksi perspektif dengan menampilkan torus yang terlihat lebih dekat dengan perspektif yang lebih nyata. Ini membantu memahami cara mengimplementasikan proyeksi perspektif dalam grafik komputer.
4. **Penerapan Rotasi dan Tampilan 3D:** Program ketiga memperlihatkan cara membuat tampilan objek tiga dimensi yang lebih kompleks dengan pemahaman rotasi dan penerapan warna. Hal ini menunjukkan bagaimana objek 3D dapat diatur dan ditampilkan dalam berbagai sudut pandang.

Praktikum ini merupakan langkah awal yang penting dalam memahami dan menguasai konsep dasar grafik komputer, terutama dalam hal proyeksi 3D. Dengan pemahaman ini, kita siap untuk mengeksplorasi lebih lanjut dalam pengembangan aplikasi grafis 3D yang lebih kompleks dan bermanfaat. Kesimpulannya, praktikum ini memberikan dasar yang kuat untuk pengembangan lebih lanjut dalam dunia grafik komputer.