

# **LAPORAN PRATIUM GRAFIK KOMPUTER**

Diajukan untuk memenuhi Tugas mata kuliah Pratikum Grafik Komputer

## **PEMBUATAN ALAT – ALAT OLAHRAGA MENGUNAKAN OPENGL PADA C++**

Dosen Pengampu : Sri Rahayu, M.Kom

Instruktur Pratikum : Arul Budi Kalimat, S.Kom



Disusun oleh

Kelompok :

Abizar Algifari  
2306117

M. Nadhif Rahmat F.  
2306104

Destian M. Luthfi  
2306111

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN ILMU KOMPUTER**

**INSTITUT TEKNOLOGI GARUT**

**2024**

## **KATA PENGANTAR**

Puji dan syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya sehingga kami dapat menyelesaikan Laporan Praktikum Jaringan Komputer ini. Laporan ini dibuat sebagai salah satu tugas dari mata kuliah Jaringan Komputer, dengan tujuan untuk memberikan pemahaman yang lebih baik tentang OpenGL.

Kami mengucapkan terima kasih kepada dosen pengampu Sri Rahayu, M.Kom, instruktur praktikum Arul Budi Kalimat, S.Kom, serta semua pihak yang telah memberikan dukungan dalam penyusunan laporan ini.

Kami menyadari bahwa laporan ini masih memiliki kekurangan, untuk itu kami mengharapkan kritik dan saran yang membangun demi perbaikan di masa yang akan datang.

Garut, 13 Desember 2024

Penulis

## DAFTAR ISI

KATA PENGANTAR .....	i
DAFTAR ISI .....	ii
DAFTAR GAMBAR.....	iii
BAB I PENDAHULUAN.....	1
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah .....	1
1.3    Tujuan .....	2
BAB II TINJAUAN PUSTAKA.....	3
2.1    OpenGL.....	3
2.2    Konfigurasi OpenGL pada Dev C++ atau VSCode .....	3
2.3    Cara Kerja OpenGL.....	8
2.4    Alat – Alat Olahraga Di OpenGL .....	8
BAB III HASIL .....	9
3.1    Source Code .....	9
3.2    Output.....	17
3.3    Penjelasan .....	17
BAB IV .....	19
4.1.    Kesimpulan .....	19
DAFTAR PUSTAKA .....	19

## DAFTAR GAMBAR

Gambar 1: File Freeglut.....	3
Gambar 2: Isi File Freeglut.....	4
Gambar 3: Tampilan Dev C++ .....	4
Gambar 4: Membuat file baru.....	5
Gambar 5: New Project .....	5
Gambar 6: Beri nama file .....	6
Gambar 7: Tampilan pada Project .....	6
Gambar 8: Parameter OpenGL .....	7
Gambar 9: Alur Pemrosesan OpenGL .....	8

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan zaman yang disertai oleh perkembangan teknologi komputer grafika dan pencitraan tiga dimensi saat ini telah mengalami kemajuan yang sangat pesat dengan tingkat kualitas dan pencapaian yang cukup signifikan dalam dekade terakhir. Grafika komputer merupakan teknik dalam ilmu komputer dan matematika untuk merepresentasikan dan memanipulasikan gambar menggunakan komputer. Salah satu aplikasi yang komplit dari grafika komputer adalah untuk visualisasi data dalam bentuk grafika dua dimensi (2D) atau tiga dimensi (3D) dilengkapi dengan animasi[1] .

Grafika komputer sendiri adalah suatu bidang ilmu komputer dan matematika untuk mempresentasikan dan memanipulasi data gambar menggunakan komputer. Grafika komputer berkaitan dengan pembuatan dan manipulasi gambar (visual) secara digital. Bentuk sederhana dari grafika komputer adalah grafika komputer 2D yang selanjutnya dikembangkan menjadi grafika komputer 3D, pemrosesan citra (image processing), dan pengenalan pola (pattern recognition)[2] .

OpenGL (Open Graphic Library) merupakan library yang terdiri dari berbagai macam fungsi dan digunakan untuk menggambar beberapa objek 2D dan 3D. Library – library ini mendefinisikan sebuah Bahasa, cross platform API (antar muka pemrograman aplikasi) untuk menulis aplikasi yang menghasilkan objek 2D dan 3D grafis. Bahasa pemrograman yang digunakan pada umumnya adalah Bahasa C++[1].

C++ merupakan Bahasa pemrograman yang merupakan perkembangan dari Bahasa C. Bahasa itu diturunkan dari bahasa sebelumnya yaitu bahasa B. bahasa tersebut dirancang sebagai bahasa pemrograman yang dijalankan pada sistem Unix, Untuk mendukung fitur-fitur pada C++, dibangun efisiensi dan sistem support untuk pemrograman tingkat rendah (low level coding)[3].

### **1.2 Rumusan Masalah**

Di sini, Anda menyatakan masalah atau pertanyaan yang ingin dijawab melalui praktikum. Misalkan :

1. Apa yang dimaksud dengan OpenGL ?
2. Bagaimana cara mengkonfigurasi OpenGL pada Dev C++ atau VSCode?
3. Bagaimana cara kerja dari OpenGL?
4. Bagaimana membuat Alat – Alat Olahraga dalam OpenGL?

### **1.3 Tujuan**

Nyatakan tujuan dari laporan praktikum. Misalkan :

1. Mengetahui apa itu OpenGL
2. Mengetahui cara mengkonfigurasi OpenGL pada Dev C++ atau VSCode.
3. Mengetahui cara kerja dari OpenGL
4. Mengetahui cara pembuatan Alat – Alat Olahraga dalam OpenGL

## BAB II TINJAUAN PUSTAKA

### 2.1 OpenGL

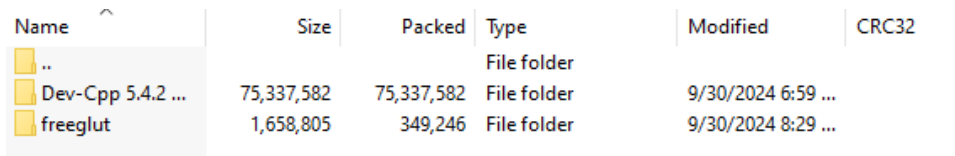
OpenGL merupakan spesifikasi standar yang mendefinisikan sebuah lintas bahasa, yang dapat menghasilkan grafis computer dua dimensi maupun tiga dimensi. OpenGL dikembangkan oleh Silicon Graphics Inc (SGI) pada tahun 1992 dan secara luas digunakan dalam CAD, realitas maya, visualisasi ilmiah, visualisasi informasi, dan simulasi penerbangan[3].

Opengl ( Open Graphics Library) merupakan salah satu perangkat tambahan yang digunakan dalam membangun desain sebuah atau lebih bangun datar (2 Dimensi), bangun ruang (3Dimensi) dengan dukungan software bahasa pemrograman tertentu. Untuk menjalankan fungsi opengl secara luas, opengl harus diintegrasikan dengan Bahasa pemrograman berbasis C atau C++. Opengl termasuk salah satu jenis dari Application Programming Interface (API), hal ini disebabkan opengl tidak dapat bekerja sendiri tanpa dukungan software atau aplikasi Bahasa pemrograman[4].

Singkatnya, Open Graphics Library, OpenGL menghilangkan kebutuhan untuk pemrogram untuk menulis ulang bagian grafis dari sistem operasi setiap kali sebuah bisnis akan diupgrade ke versi baru dari sistem. Fungsi dasar dari OpenGL adalah untuk mengeluarkan koleksi perintah khusus atau executable ke sistem operasi. Perintah mengawali untuk menggambar. Mode merupakan konstanta yang menyatakan bagaimana opengl harus menghubungkan titik/vertex yang akan digambarkan[1].

### 2.2 Konfigurasi OpenGL pada Dev C++ atau VSCode

1. Sebelum masuk kedalam Dev C++ pastikan file freglut untuk OpenGL sudah terpasang dibagian folder C++ pada komputer atau laptop yang digunakan.
2. Jika belum terpasang, silahkan download terlebih dahulu file freglut untuk OpenGL
3. Setelah di download, ekstrak file freglut



Name	Size	Packed	Type	Modified	CRC32
..			File folder		
Dev-Cpp 5.4.2 ...	75,337,582	75,337,582	File folder	9/30/2024 6:59 ...	
freeglut	1,658,805	349,246	File folder	9/30/2024 8:29 ...	

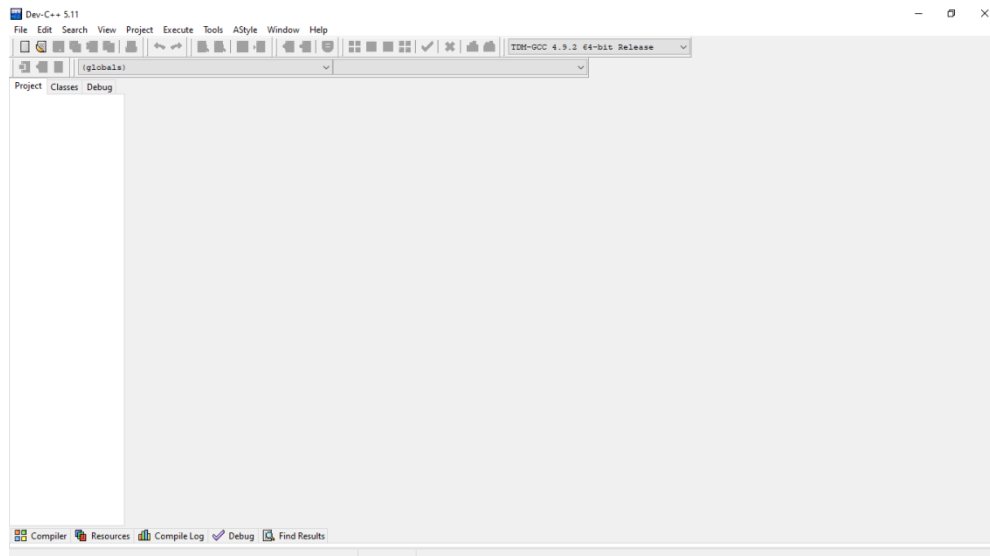
**Gambar 1: File Freeglut**

- Setelah di ekstrak akan muncul tiga file yaitu bin, include, dan lib

Name	Size	Packed	Type	Modified	CRC32
..			File folder		
bin	482,816	146,004	File folder	9/30/2024 8:29 ...	
include	39,795	10,533	File folder	9/30/2024 8:29 ...	
lib	1,129,448	189,756	File folder	9/30/2024 8:29 ...	
Copying.txt	1,466	792	Text Document	10/11/2009 8:3...	E58A075E
Readme.txt	5,280	2,161	Text Document	3/15/2015 8:45 ...	98053627

**Gambar 2: Isi File Freeglut**

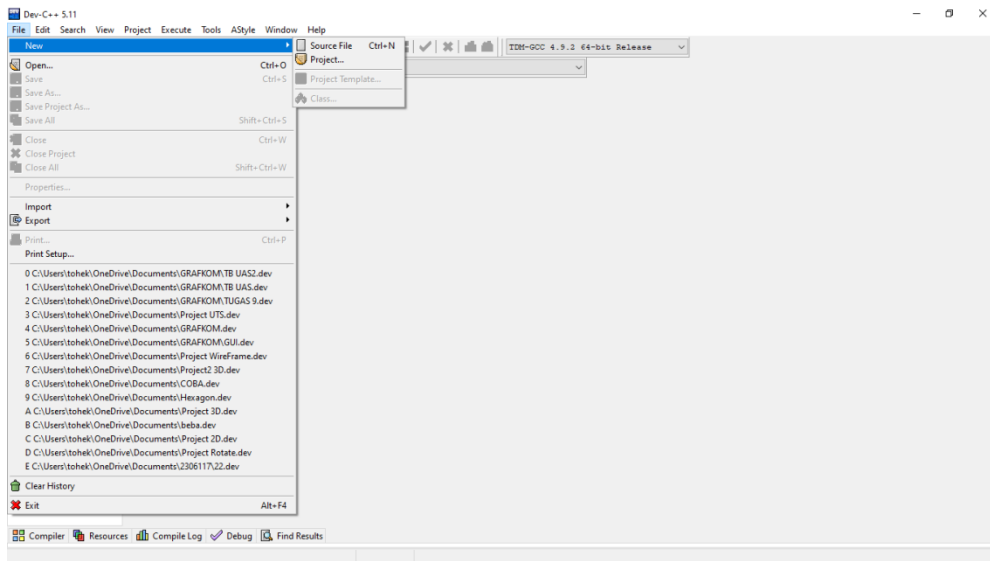
- Copy file dari folder bin lalu pastekan di C:\Windows\System32
- Copy file dari folder include lalu pastekan di C:\Program Files (x86)\Dev-Cpp\MinGW64\x86\_64-mingw32\include\GL
- Copy file dari folder lib lalu pastekan di C:\Program Files (x86)\Dev-Cpp\MinGW64\x86\_64-mingw32\lib
- Jika sudah semua, langsung saja buka aplikasi Dev C++ nya
- Buka Aplikasi Dev C++ 5.11



**Gambar 3: Tampilan Dev C++**

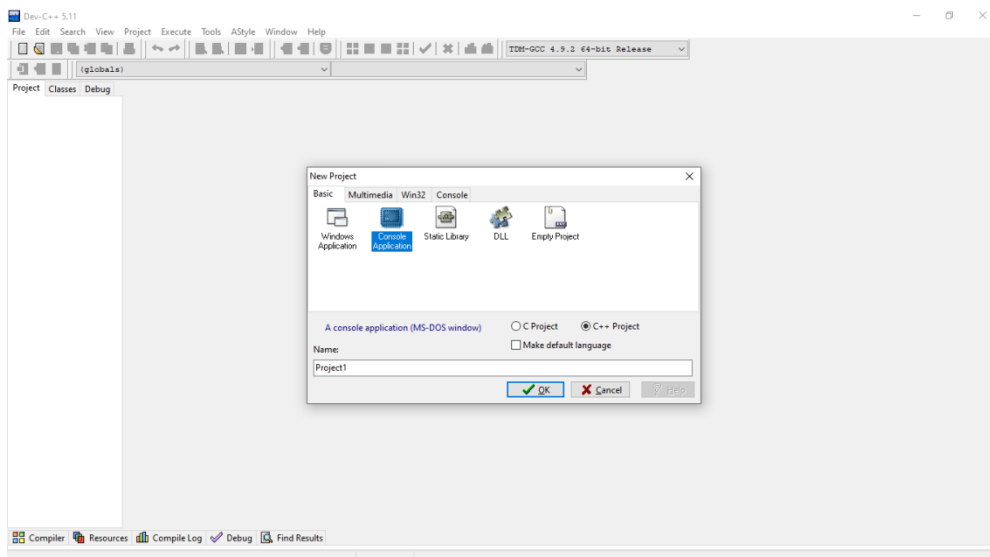


10. Klik menu file pada bagian atas lalu pilih new lalu project



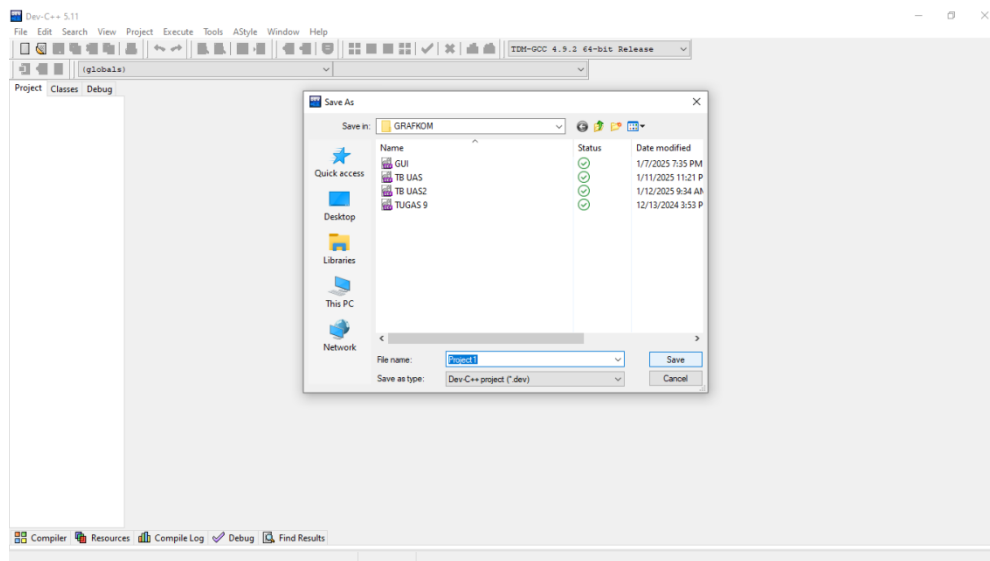
**Gambar 4: Membuat file baru**

11. Setelah pilih project, kita pilih Console Application untuk New Project nya, lalu ok



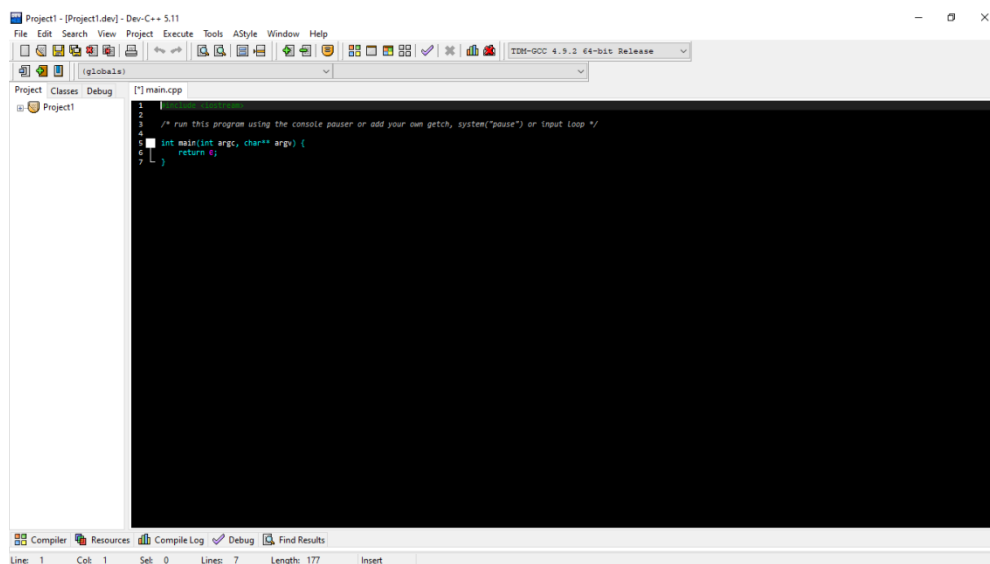
**Gambar 5: New Project**

12. Setelah itu, beri nama dulu file yang akan dibuat



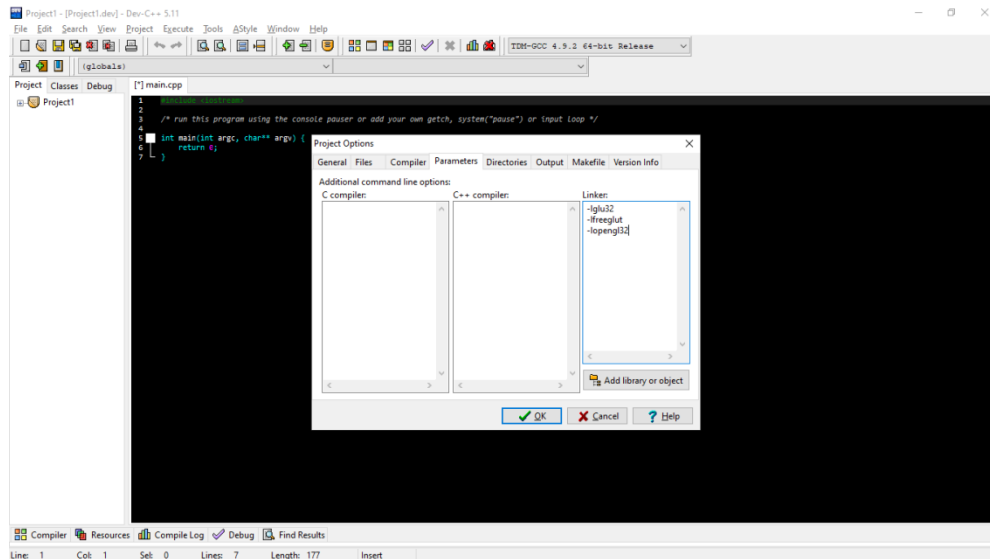
**Gambar 6: Beri nama file**

13. Akan muncul tampilan awal dari hasil New Project



**Gambar 7: Tampilan pada Project**

14. Sebelum meng-compile atau memasukkan kode yang dibuat, masuk ke menu parameter lalu masukkan parameter seperti pada gambar yang berfungsi untuk menautkan pustaka (libraries) OpenGL saat melakukan kompilasi program C++

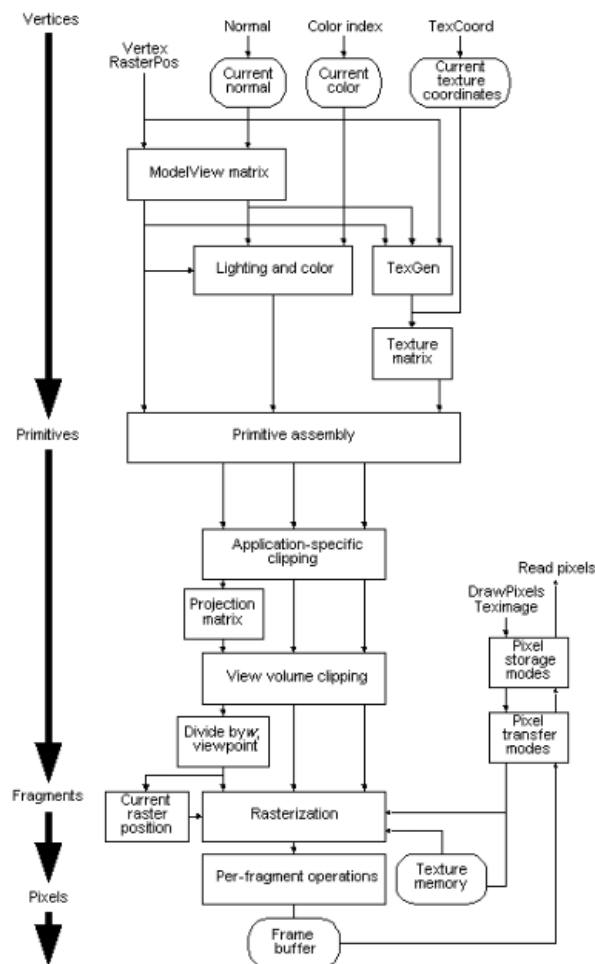


**Gambar 8: Parameter OpenGL**

### 2.3 Cara Kerja OpenGL

Banyak fungsi OpenGL digunakan khusus untuk menggambar objek seperti titik, garis, poligon, dan bitmap. Beberapa fungsi mengontrol cara beberapa gambar ini terjadi (seperti yang memungkinkan antialias atau teksur). Fungsi lain secara khusus berkaitan dengan manipulasi framebuffer. Topik di bagian ini menjelaskan bagaimana semua fungsi OpenGL bekerja sama untuk membuat alur pemrosesan OpenGL. Bagian ini juga melihat lebih dekat tahapan di mana data benar-benar diproses, dan mengikat tahapan ini ke fungsi OpenGL[5].

Diagram berikut merinci alur pemrosesan OpenGL. Untuk sebagian besar alur, terdapat tiga panah vertikal di antara tahap utama. Panah ini mewakili simpul dan dua jenis data utama yang dapat dikaitkan dengan simpul: nilai warna dan koordinat tekstur. Perhatikan juga bahwa simpul dirakit menjadi primitif, lalu menjadi fragmen, dan akhirnya menjadi piksel dalam framebuffer[5].



**Gambar 9: Alur Pemrosesan OpenGL**

### 2.4 Alat – Alat Olahraga Di OpenGL

## BAB III

### HASIL

#### 3.1 Source Code

Untuk Source code kalian masukan source code yang telah kalian buat lalu masukan ke dalam kolom dibawah ini :

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

float Xpos = 0.0, Ypos = 0.0, xpos = 0.0, ypos = 0.0;
float aspect;
float angle = 0.0;
float scaleFactor = 1.0f; // Faktor zoom
float scaleStep = 0.1f;
int windows[10];
int size = 0;
bool hidden = false;
bool rotationEnabled = false;
GLfloat light_position[4] = { 1.0f, 1.0f, 1.0f, 0.0f };

void addWindows(const char *title, void (displayCallback)(void));
void firstWindows();
void secondWindows();
void thirdWindows();
void hiddenCarte();
void inisialisasi();
void satu();

// Abizar
// Fungsi untuk menggambar silinder
void drawCylinder(GLfloat radius, GLfloat height) {
    GLUQuadric* quadric = gluNewQuadric();
    gluQuadricNormals(quadric, GLU_SMOOTH);

    // Menggambar silinder
    gluCylinder(quadric, radius, radius, height, 32, 32);

    gluDeleteQuadric(quadric);
}

// ABIZAR
void drawCartesius(){
    glLineWidth(1.0);
    glColor3f(1.0, 0.0, 0.0);
    glBegin(GL_LINES);

    glVertex3f(-50.0, 0.0, 0.0);
    glVertex3f(50.0, 0.0, 0.0);

    glVertex3f(0.0, -50.0, 0.0);
    glVertex3f(0.0, 50.0, 0.0);

    glVertex3f(0.0, 0.0, -50.0);
```

```

        glVertex3f(0.0, 0.0, 50.0);
        glEnd();
    }

//ABIZAR
void Dumbell() {

    glColor3f(0.3, 0.3, 0.3);
    glPushMatrix();
    glRotatef(90.0f, 0.0f, 2.0f, 0.0f);
    glTranslatef(0.0f, 0.0f, -3.0f); // Pindah ke posisi pegangan
    drawCylinder(0.3f, 6.0f); // Radius dan tinggi pegangan
    glPopMatrix();

    // Gambar disk di kiri
    glColor3f(0.2, 0.2, 0.2); // Warna hitam
    glPushMatrix();
    glRotatef(90.0f, 0.0f, 2.0f, 0.0f);
    glTranslatef(0.0f, 0.0f, -1.2f); // Pindah ke kanan
    glutSolidTorus(0.5, 1.0, 30, 10);
    glPopMatrix();

    glColor3f(0.2, 0.2, 0.2); // Warna hitam
    glPushMatrix();
    glRotatef(90.0f, 0.0f, 2.0f, 0.0f);
    glTranslatef(0.0f, 0.0f, -1.8f); // Pindah ke kanan
    glutSolidTorus(0.5, 0.8, 30, 10); // Gambar disk
    glPopMatrix();

    glColor3f(0.2, 0.2, 0.2); // Warna hitam
    glPushMatrix();
    glRotatef(90.0f, 0.0f, 2.0f, 0.0f);
    glTranslatef(0.0f, 0.0f, -2.1f); // Pindah ke kiri
    glutSolidTorus(0.6, 0.5, 30, 10); // Gambar disk
    glPopMatrix();

    // Gambar disk di kanan
    glColor3f(0.2, 0.2, 0.2); // Warna hitam
    glPushMatrix();
    glRotatef(90.0f, 0.0f, 2.0f, 0.0f);
    glTranslatef(0.0f, 0.0f, 1.2f); // Pindah ke kanan
    glutSolidTorus(0.5, 1.0, 30, 10);
    glPopMatrix();

    glColor3f(0.2, 0.2, 0.2); // Warna hitam
    glPushMatrix();
    glRotatef(90.0f, 0.0f, 2.0f, 0.0f);
    glTranslatef(0.0f, 0.0f, 1.8f); // Pindah ke kanan
    glutSolidTorus(0.5, 0.8, 30, 10); // Gambar disk
    glPopMatrix();

    glColor3f(0.2, 0.2, 0.2); // Warna hitam
    glPushMatrix();
    glRotatef(90.0f, 0.0f, 2.0f, 0.0f);
    glTranslatef(0.0f, 0.0f, 2.1f); // Pindah ke kiri
    glutSolidTorus(0.6, 0.5, 30, 10); // Gambar disk
    glPopMatrix();
}

```

```

}

//ABIZAR
void drawMatras() {

    glColor3f(0.8f, 0.8f, 0.8f);
    glPushMatrix();
    glScalef(18.0f, -0.5f, 18.0f);
    glTranslatef(0.0f, 2.0f, 0.0f);
    glutSolidCube(1.0f);
    glPopMatrix();

    glColor3f(0.1f, 0.1f, 0.1f);
    glPushMatrix();
    glScalef(4.0f, -0.1f, 10.0f);
    glTranslatef(0.0f, 2.0f, 0.0f);
    glutSolidCube(1.0f);
    glPopMatrix();

    glEnd();
}

//ABIZAR
void thirdWindows() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);

    glPushMatrix();
    glScalef(0.5f, 0.5f, 0.5f);
    glRotatef(45.0f, 0.0f, 2.0f, 0.0f);
    glTranslatef(-7.0f, 0.0f, -3.0f);
    Dumbell();
    glPopMatrix();

    glPushMatrix();
    glScalef(0.5f, 0.5f, 0.5f);
    glRotatef(45.0f, 0.0f, 2.0f, 0.0f);
    glTranslatef(-7.0f, 0.0f, -10.0f);
    Dumbell();
    glPopMatrix();

    // BARBEL BESAR
    glPushMatrix();
    glScalef(0.7f, 0.7f, 0.7f);
    glRotatef(180.0f, 0.0f, 2.0f, 0.0f);
    glTranslatef(-1.5f, Ypos, -9.0f);

    glColor3f(0.3, 0.3, 0.3);
    glPushMatrix();
    glRotatef(90.0f, 0.0f, 2.0f, 0.0f);
    glTranslatef(0.0f, 0.0f, -3.2f); // Pindah ke posisi pegangan
    drawCylinder(0.3f, 10.0f); // Radius dan tinggi pegangan
    glPopMatrix();
}

```

```

// Gambar disk di kiri
glColor3f(0.2, 0.2, 0.2); // Warna hitam
glPushMatrix();
glRotatef(90.0f, 0.0f, 2.0f, 0.0f);
glTranslatef(0.0f, 0.0f, -1.4f); // Pindah ke kanan
glutSolidTorus(0.6, 1.0, 30, 10);
glPopMatrix();

glColor3f(0.2, 0.2, 0.2); // Warna hitam
glPushMatrix();
glRotatef(90.0f, 0.0f, 2.0f, 0.0f);
glTranslatef(0.0f, 0.0f, -2.0f); // Pindah ke kanan
glutSolidTorus(0.6, 1.0, 30, 10); // Gambar disk
glPopMatrix();

glColor3f(0.2, 0.2, 0.2); // Warna hitam
glPushMatrix();
glRotatef(90.0f, 0.0f, 2.0f, 0.0f);
glTranslatef(0.0f, 0.0f, -2.5f); // Pindah ke kiri
glutSolidTorus(0.3, 0.4, 30, 10); // Gambar disk
glPopMatrix();

// Gambar disk di kanan
glColor3f(0.2, 0.2, 0.2); // Warna hitam
glPushMatrix();
glRotatef(90.0f, 0.0f, 2.0f, 0.0f);
glTranslatef(0.0f, 0.0f, 4.8f); // Pindah ke kanan
glutSolidTorus(0.6, 1.0, 30, 10);
glPopMatrix();

glColor3f(0.2, 0.2, 0.2); // Warna hitam
glPushMatrix();
glRotatef(90.0f, 0.0f, 2.0f, 0.0f);
glTranslatef(0.0f, 0.0f, 5.6f); // Pindah ke kanan
glutSolidTorus(0.6, 1.0, 30, 10);
glPopMatrix();

glColor3f(0.2, 0.2, 0.2); // Warna hitam
glPushMatrix();
glRotatef(90.0f, 0.0f, 2.0f, 0.0f);
glTranslatef(0.0f, 0.0f, 6.0f); // Pindah ke kiri
glutSolidTorus(0.3, 0.4, 30, 10); // Gambar disk
glPopMatrix();
glPopMatrix();

glRotatef(Xpos, 0.0, 1.0, 0.0);
drawMatras();

    inisialisasi();
    hiddenCarte();
    glutSwapBuffers();
    glutPostRedisplay();
}

// NADHIEF

```



```

void secondWindows() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); //
    satu();

    hiddenCarte();

    glPushMatrix(); // menyimpan transformasi saat ini
    glColor3ub(128, 128, 128); // warna abu-abu
    glTranslatef(xpos, 2.8f, 0.0f); // Menggeser sistem koordinat
    if (rotationEnabled) {
        glRotatef(angle, 0, 1, 1);
    }
    glutSolidSphere(5, 20, 20);
    glPopMatrix();

    glPushMatrix();
    glColor3ub(34, 139, 34); // warna hijau
    glScalef(18.0f, -0.5f, 18.0f); // mengubah skala (ukuran)
    glTranslatef(0.0f, 2.0f, 0.0f);
    if (rotationEnabled) {
        glRotatef(angle, 0, 1, 0);
    }
    glutSolidCube(1.5f);
    glPopMatrix();

    glutSwapBuffers();
    glutPostRedisplay();

    angle += 0.1;
}

//DESTIAN
void drawHighJumpPole() {
    GLUQuadric* quadric = gluNewQuadric(); // Membuat objek
    kuadrik

    // Tiang lompat tinggi (batang tongkat)
    glColor3f(0.0f, 0.0f, 1.0f); // Warna biru
    glPushMatrix();
    glTranslatef(3.0f, 1.0f, -4.0f); // Posisi dasar tiang
    glScalef(0.5f, 0.5f, 3.0f); // Membuat bentuk tiang (tinggi
    dan bulat)
    gluCylinder(quadric, 0.5f, 0.5f, 6.0f, 32, 32); // Membuat
    silinder (tiang lompat tinggi)
    glPopMatrix();

    gluDeleteQuadric(quadric); // Menghapus objek kuadrik
}

//DESTIAN
void firstWindows() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    hiddenCarte();
    glPushMatrix();

```

```

        glEnable(GL_LIGHTING);
        glEnable(GL_LIGHT0);
        inisialisasi();
        glScalef(scaleFactor, scaleFactor, 1.0f);
        glTranslatef(0.0f, 1.0f, ypos); // Geser tiang lompat tinggi
        drawHighJumpPole(); // Menggambar tongkat lompat tinggi

        glPopMatrix();
        glutSwapBuffers();
        glutPostRedisplay();
    }

void reshape(int w, int h){
    if (h == 0)
        h = 1;
    aspect = (float)w / (float)h;

    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(70.0, aspect, 1.0, 100.0);
    gluLookAt(10.0, 10.0, 17.0, 0.0, 0.0, 0.0, 0.0, 1.0,
0.0);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

//Nadhief
void satu(){
    GLfloat light_ambient[4] = { 0.2f, 0.2f, 0.2f, 1.0f };
    GLfloat light_diffuse[4] = { 1.0f, 1.0f, 1.0f, 1.0f };
    GLfloat light_specular[4] = { 1.0f, 1.0f, 1.0f, 1.0f };
    GLfloat light_position[4] = { 4.5f, 10.0f, 5.5f, 1.0f };
    glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient); //cahaya
    sekitar
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse); //cahaya
    menyebar
    glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular); //cahaya
    pantulan
    glLightfv(GL_LIGHT0, GL_POSITION, light_position); //posisi
    cahaya
    glEnable(GL_LIGHT0); // Mengaktifkan sumber cahaya GL_LIGHT0
    glEnable(GL_LIGHTING); // Mengaktifkan sistem pencahayaan
    global
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_COLOR_MATERIAL);
    glClearColor(0.1, 0.1, 0.1, 1.0);
}

//DESTIAN
void inisialisasi(){
    GLfloat light_ambient[4] = { 0.2f, 0.2f, 0.2f, 1.0f };
    GLfloat light_diffuse[4] = { 1.0f, 1.0f, 1.0f, 1.0f };
    GLfloat light_specular[4] = { 1.0f, 1.0f, 1.0f, 1.0f };
    GLfloat light_position[4] = { 1.0f, 1.0f, 1.0f, 0.0f };

```

```

        glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
        glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
        glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
        glLightfv(GL_LIGHT0, GL_POSITION, light_position);
        glEnable(GL_LIGHT0);
        glEnable(GL_DEPTH_TEST);
        glEnable(GL_COLOR_MATERIAL);
        glClearColor(0.1, 0.1, 0.1, 1.0);
    }

//ABIZAR
void addWindows(const char *title, void (displayCallback)(void))
{
    windows[size++] = glutCreateWindow(title);
    glutDisplayFunc(displayCallback);
}

//ABIZAR
void hiddenCarte()
{
    if(hidden)
    {
        drawCartesius();
    }
}

//ABIZAR
void myKeyboard(unsigned char key , int x, int y){
    switch (key){
        case 'q':
            for (int i = 0; i < size; i++)
            {
                glutDestroyWindow(windows[i]);
            }
            break;
        case 'h':
            hidden = !hidden;
            break;
        case 'a':
            Xpos += 0.01;
            break;
        case 'd':
            Xpos -= 0.01;
            break;
        case 'w':
            Ypos += 1.0;
            break;
        case 's':
            Ypos -= 1.0;
            break;
    }
}

//Nadhief
void mykeyboard(unsigned char key , int x,int y){
    switch (key){
        case 'q':

```

```

        for (int i = 0; i < size; i++)
        {
            glutDestroyWindow(windows[i]);
        }
        break;
    case 'j':
        xpos -= 1.0;
        break;
    case 'l':
        xpos += 1.0;
        break;
    case 'k': // Hentikan atau lanjutkan rotasi
        rotationEnabled = !rotationEnabled;
        break;
    case 'i':
        hidden = !hidden;
        break;
    }
}

//DESTIAN
void keyboard(unsigned char key , int x,int y){
    switch (key){
        case 'q':
            for (int i = 0; i < size; i++)
            {
                glutDestroyWindow(windows[i]);
            }
            break;
        case '=': // Zoom in
            scaleFactor += scaleStep;
            break;
        case '-': // Zoom out
            scaleFactor -= scaleStep;
            if (scaleFactor < 0.1f) scaleFactor = 0.1f; // Hindari
            nilai negatif atau terlalu kecil
            break;
        case 'o':
            ypos -= 1.0;
            break;
        case 'p':
            ypos += 1.0;
            break;
    }
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    //ABIZAR
    glutInitWindowSize(500,400);
    glutInitWindowPosition(440, 150);
    addWindows("Abizar Algifari", thirdWindows);
    glutReshapeFunc(reshape);
    glutKeyboardFunc(myKeyboard);
    //Nadhief
    glutInitWindowSize(400,400);
    glutInitWindowPosition(30, 150);
    addWindows(" Muhammad Nadhief R F", secondWindows);
}

```

```

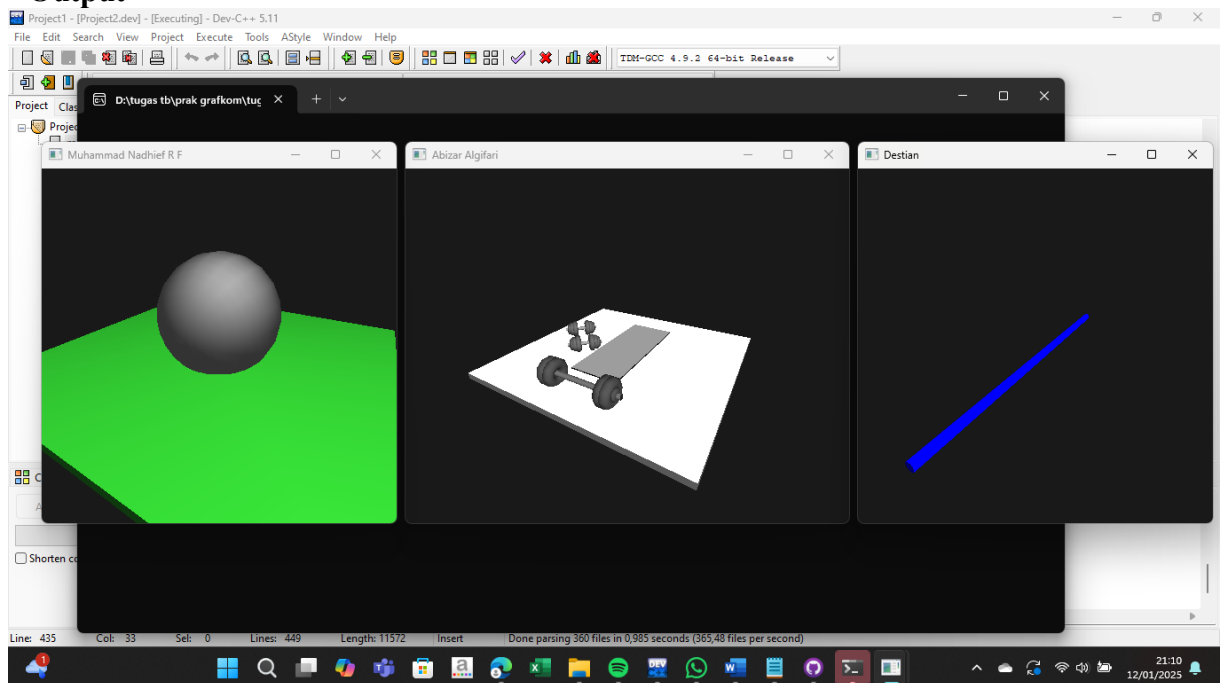
    glutReshapeFunc(reshape);
    glutKeyboardFunc(mykeyboard);
    //DESTIAN
    glutInitWindowSize(400,400);
    glutInitWindowPosition(950, 150);
    addWindows("Destian", firstWindows);
    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);

    glutMainLoop();
    return 0;
}

```

Ketentuan dari penulisan source code menggunakan :  
Font Courier New dengan ukuran font 11

### 3.2 Output



### 3.3 Penjelasan

Kode di atas adalah sebuah program OpenGL yang menggunakan GLUT untuk menggambar objek 3D dan melakukan interaksi dengan keyboard. Program ini terbagi dalam tiga bagian utama yang masing-masing bertanggung jawab untuk menggambar objek tertentu:

1. Abizar (thirdWindows):

- Menggambar beberapa objek, termasuk dumbbell (alat angkat besi), matras, dan kartesius 3D.
- Menggunakan pencahayaan untuk memperjelas objek yang digambar.
- Fungsi Dumbell digunakan untuk menggambar dumbbell dengan silinder sebagai pegangan dan torus sebagai disk.
- Fungsi drawMatras menggambar matras tempat objek diletakkan.

2. Muhammad Nadhief (secondWindows):

- Menggambar bola dan objek lainnya yang dapat diputar.
- Fungsi satu mengatur pencahayaan dan memungkinkan bola untuk berputar saat tombol tertentu

ditekan (misalnya, rotasi diaktifkan dengan tombol 'k').

3. Destian (firstWindows):

- Menggambar tiang lompat tinggi menggunakan objek `gluCylinder`.
- Mengatur posisi tiang dengan transformasi 3D, serta menggunakan pencahayaan untuk memperjelas objek.
- Setiap bagian menggunakan GLUT untuk menggambar objek dan memungkinkan interaksi menggunakan keyboard. Fungsi-fungsi kunci termasuk pengaturan cahaya (`glLightfv`), transformasi objek (`glTranslatef`, `glRotatef`), dan penggunaan berbagai objek dasar seperti torus dan silinder.

Pengguna dapat mengubah posisi objek, rotasi, dan zoom menggunakan input dari keyboard, serta menampilkan atau menyembunyikan koordinat kartesius dengan menekan tombol tertentu.

## **BAB IV**

### **4.1. Kesimpulan**

- **Pemahaman tentang OpenGL:** OpenGL adalah sebuah pustaka grafika yang digunakan untuk menggambar objek-objek 2D dan 3D pada aplikasi berbasis komputer. Dengan menggunakan OpenGL, kita dapat membuat visualisasi grafis yang kompleks dan interaktif, seperti yang dilakukan dalam pembuatan alat olahraga pada laporan ini.
- **Pengaturan dan Konfigurasi OpenGL:** Dalam praktik ini, kami telah berhasil mengonfigurasi OpenGL pada Dev C++ dan VSCode. Proses instalasi meliputi penambahan file-file pustaka Freeglut yang diperlukan untuk mendukung fungsionalitas OpenGL, yang kemudian diintegrasikan dengan kode C++ untuk dapat memanipulasi grafis 3D.
- **Pemahaman tentang Alur Kerja OpenGL:** Kami mempelajari alur pemrosesan dalam OpenGL yang melibatkan pembuatan dan pengolahan primitif grafis dari simpul hingga piksel pada framebuffer. Ini memberikan gambaran tentang bagaimana OpenGL bekerja untuk menghasilkan gambar yang ditampilkan pada layar.
- **Pembuatan Alat Olahraga Menggunakan OpenGL:** Dalam praktikum ini, kami berhasil membuat model beberapa alat olahraga sederhana, seperti dumbbell dan matras, menggunakan berbagai fungsi grafis di OpenGL. Proses ini mencakup penggunaan objek-objek dasar seperti silinder, torus, dan kubus yang digabungkan untuk membentuk representasi tiga dimensi dari alat olahraga.

Dengan demikian, praktikum ini memberikan pemahaman yang mendalam tentang pemrograman grafika 3D menggunakan OpenGL serta aplikasinya dalam pembuatan model objek yang berguna dalam berbagai bidang, termasuk simulasi dan visualisasi produk.

### DAFTAR PUSTAKA

- [1] M. Adnani and A. Zakki Falani, "Implementasi Open Gl Untuk Pembuatan Objek 3D," vol. 03.
- [2] C. Al Vareel, R. Francisco Hutabarat, and S. Harapan Gulo, "Pembuatan Animasi Integral Sebagai Implementasi Grafika Komputer Dengan Menggunakan Morphing dan OpenGL," *Volume*, vol. 5, pp. 269–279, Jul. 2022, [Online]. Available: <https://ojs.trigunadharma.ac.id/index.php/jsk/index>
- [3] M. B. Priyantono and A. A. Rachmawan, "IMPLEMENTASI SISTEM SIMULASI PENAMPILAN TATA SURYA BERBASIS 3D MENGGUNAKAN OPENGL," *Jurnal Teknologi Informasi*, vol. 4, no. 1, 2020.
- [4] Siahaan, Ahmad Taufik Al Afkari. "Diktat Panduan Penggunaan Aplikasi Dev C++ V. 5.11 Sebagai Media Praktikum Mata Kuliah Grafika Komputer Fakultas Sains Dan Teknologi UIN Sumatera Utara Medan." (2024).
- [5] Microsoft Learn, "Pengantar OpenGL," Microsoft. Accessed: Jan. 12, 2025. [Online]. Available: <https://learn.microsoft.com/id-id/windows/win32/opengl/opengl-processing-pipeline>