

LAPORAN PRATIUM GRAFIK KOMPUTER

Diajukan untuk memenuhi Tugas mata kuliah Pratikum Grafik Komputer

FOOD_3D

Dosen Pengampu : Sri Rahayu, M.Kom

Instruktur Pratikum : Arul Budi Kalimat, S.Kom



Disusun oleh

Kelompok : 7

Anyelir Kuntum Sari

2306128

Kamilaeni

2306098

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 Objek 3D.

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, 12 Januari 2025

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	2
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.....	6
2.4 FOOD _3D Di OpenGL.....	6
BAB III HASIL.....	7
3.1 Source Code.....	7
3.2 Output	17
3.3 Penjelasan.....	18
BAB IV	20
4.1. Kesimpulan.....	20
DAFTAR PUSTAKA	21

DAFTAR GAMBAR

Gambar 1 buka aplikasi dev c++	3
Gambar 2 membuat project pada dev c++.....	4
Gambar 3 memilih Console.....	4
Gambar 4 memberi nama file	5
Gambar 5 memasukan linker parameter.....	5

BAB I

PENDAHULUAN

1.1 Latar Belakang

Grafika komputer (Inggris: Computer graphics) adalah bagian dari ilmu komputer yang berkaitan dengan pembuatan dan manipulasi gambar (visual) secara digital. Bentuk sederhana dari grafika komputer adalah grafika komputer 2D yang kemudian berkembang menjadi grafika komputer 3D, pemrosesan citra (image processing), dan pengenalan pola (pattern recognition). Grafika komputer sering dikenal juga dengan istilah visualisasi data. Grafik meliputi gambar dan pencitraan lain yang dihasilkan oleh komputer berbentuk garis, lengkungan, kurva dan sebagainya. Komputer dapat menghasilkan pencitraan dalam sejumlah piksel, dan printer dot matrix akan mencetak citra/gambar tersebut dalam sejumlah titik. [1]

Perlu dikembangkan sebuah software yang mampu merepresentasikan visual dalam format tiga dimensi (3D) secara real time dengan memasukan data atau parameter – parameter yang diperoleh dari suatu alat eksternal. Tanpa perlu proses rendering kesebuah file, melainkan langsung bisa di displaykan pada layer monitor. Menggunakan 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 platfrom API (antar muka pemrograman aplikasi) untuk menulis aplikasi yang menghasilkan objek 2D dan 3D grafis. Bahasa pemrograman yang digunakan pada umumnya adalah Bahasa C++. [2]

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

Berikut adalah rumusan masalah tentang laporan praktikum :

1. Apa yang dimaksud dengan OpenGL ?
2. Bagaimana cara mengkonfigurasi OpenGL pada Dev C++?
3. Bagaimana cara kerja dari OpenGL ?
4. Bagaimana membuat Food_3D dalam OpenGL?

1.3 Tujuan

1. Berikut adalah tujuan dari laporan praktikum :
2. Mengetahui apa itu OpenGL
3. Mengetahui cara mengkonfigurasi OpenGL pada Dev C++
4. Mengetahui cara kerja dari OpenGL
5. Mengetahui cara pembuatan Food_3D dalam OpenGL

BAB II

TINJAUAN PUSTAKA

2.1 OpenGL

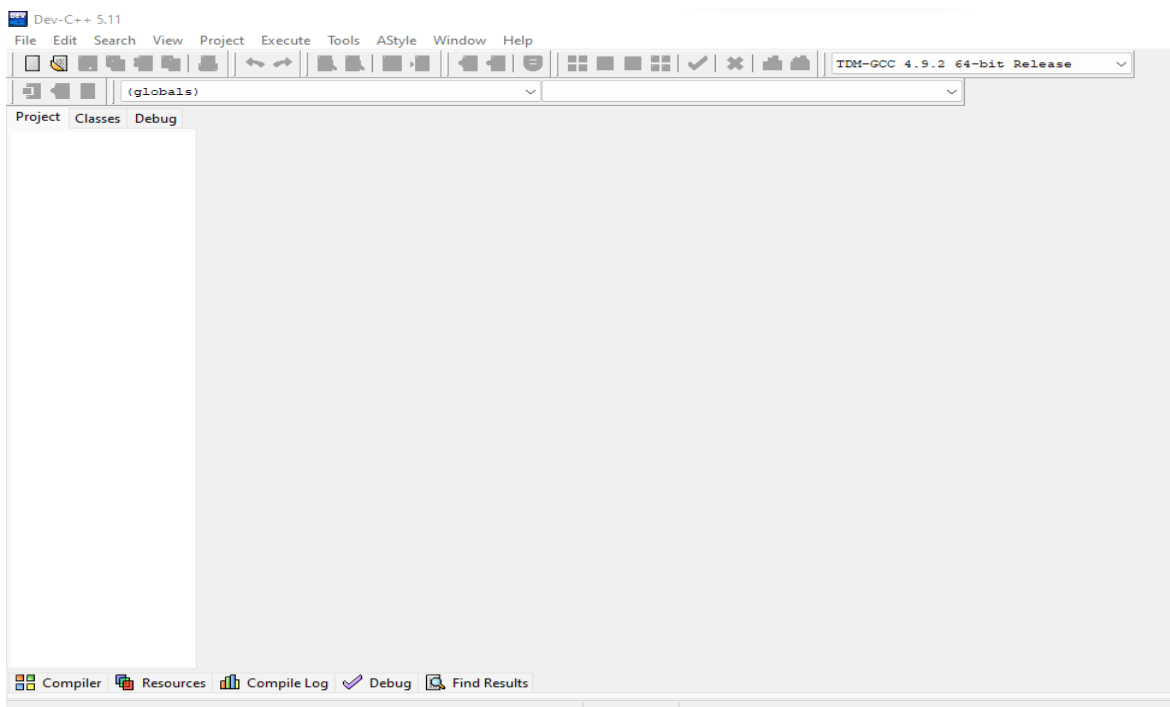
OpenGL merupakan kepanjangan dari open graphic library. OpenGL diproduksi oleh Silicon Graphics, Inc (SGI) dan pada awalnya ditujukan hanya untuk sistem komputer mereka, tetapi dalam perkembangannya, OpenGL diterima menjadi salah satu bakuan (standard) dalam grafika komputer dan saat ini telah diimplementasikan dalam berbagai sistem komputer. OpenGL merupakan pustaka program (program library) yang menyediakan sejumlah perintah yang berhubungan dengan grafika. Awal pemrograman grafis adalah mode teks yang diubah menjadi mode grafis dengan memanfaatkan fungsi dari VGA card.

Pada awal pemrograman grafis, warna yang bisa dimanfaatkan sebanyak 8 warna sampai dengan 16 warna. Dengan pemrograman visual, maka penggunaan warna menjadi lebih banyak (0 sampai dengan 255 atau 16 bit warna). Dengan menggunakan rutin tersebut, para pemrogram komputer grafik dapat membuat grafis yang lebih realistis dari sekedar gambar 2D dan 3D biasa. Portabilitas OpenGL sebagai kumpulan library memungkinkan untuk diakses melalui bahasa pemrograman (dengan menyertakan library OpenGL di direktori library bahasa tersebut) dan dijalankan pada platform komputer apapun. [4]

2.2 Konfigurasi OpenGL pada Dev C++ atau VSCode

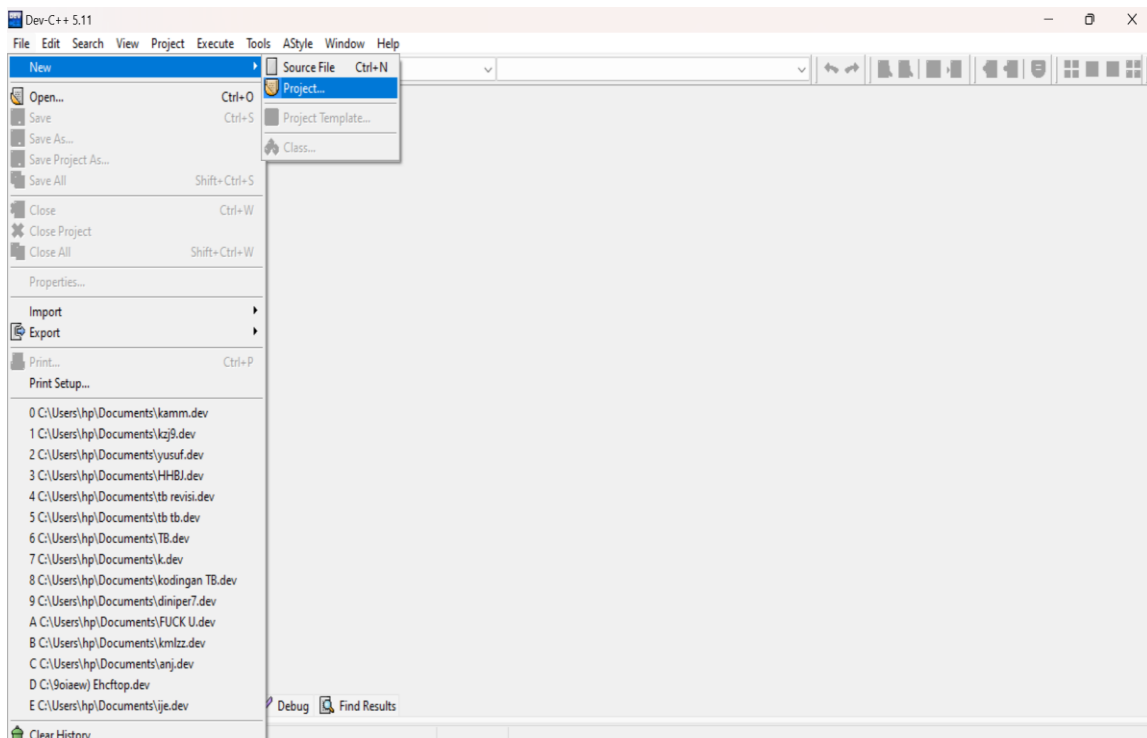
Uraikan dan Jelaskan langkah-langkah untuk konfigurasi OpenGL pada Dev C++

A. Buka aplikasi Dev C++



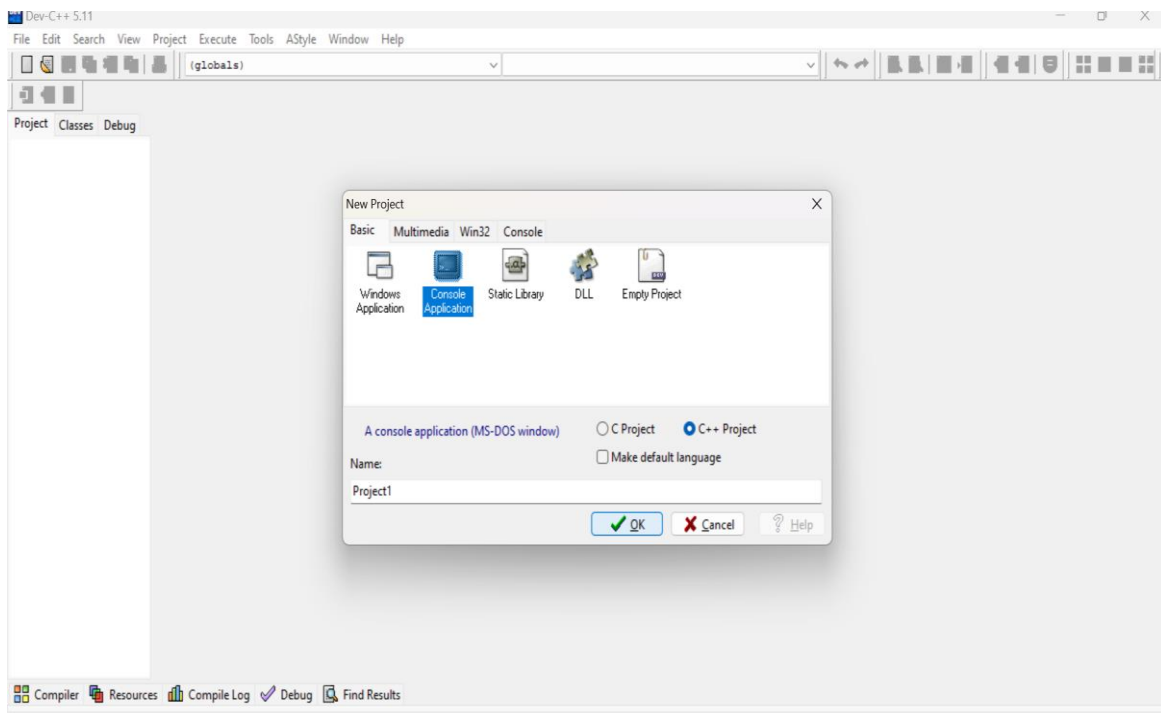
Gambar 1 buka aplikasi dev c++

B. Buat Project Baru pada OpenGL



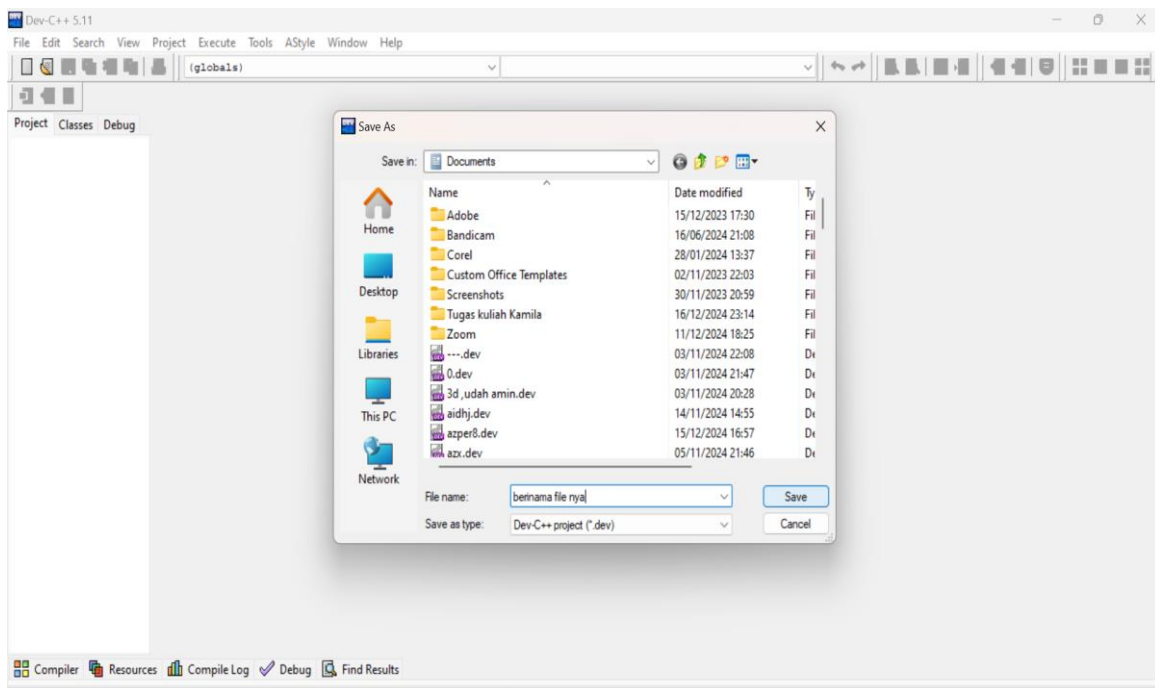
Gambar 2 membuat project pada dev c++

C. Pilih Console Application kemudian klik Ok



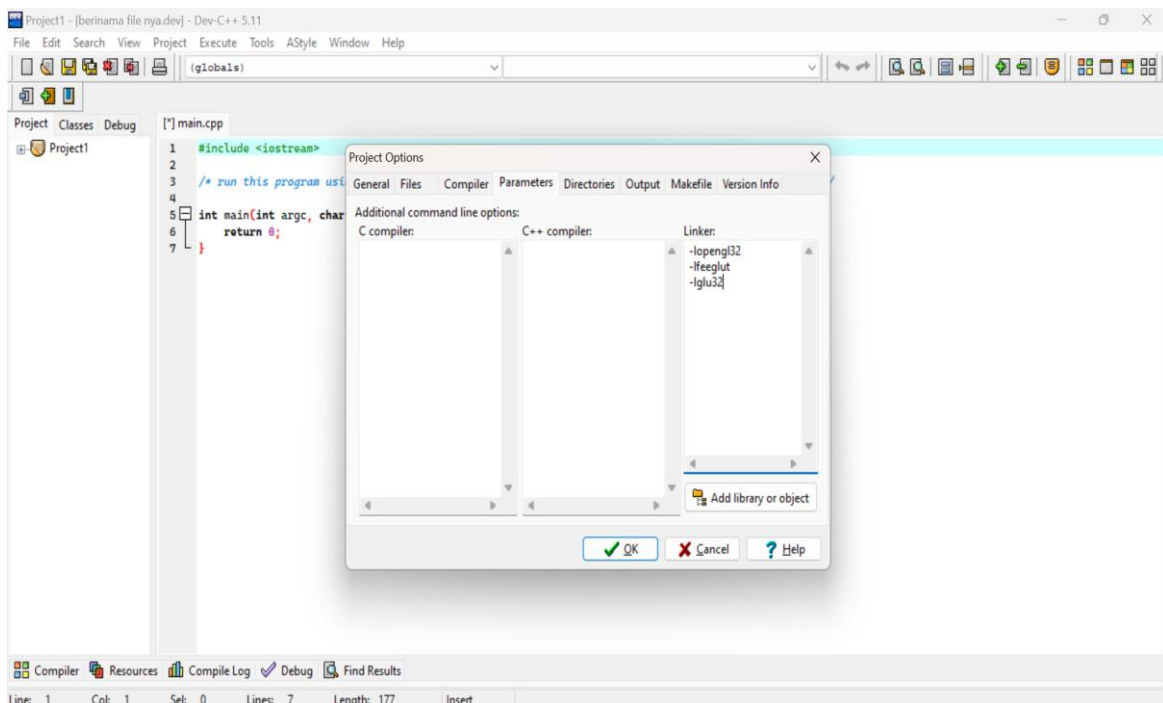
Gambar 3 memilih Console

D. Setelah itu Beri Nama File nya kemudian klik Save



Gambar 4 memberi nama file

E. Langkah selanjutnya masuk ke dalam Project Options kemudian ke Parameters dan masukan Linker nya



Gambar 5 memasukan linker parameter

2.3 Cara Kerja OpenGL

OpenGL lebih mengarah pada prosedural daripada sebuah deskriptif API grafis. Untuk mendeskripsikan scene dan bagaimana penampilannya, .Langkah tersebut termasuk memanggil banyak perintah OpenGL, perintah tersebut digunakan untuk menggambarkan grafis primitif seperti titik, garis dan poligon dalam tiga dimensi. Sebagai tambahan, OpenGL mendukung lighting, shading ,texture mapping, blending, transparency, dan banyak kemampuan efek khusus lainnya. OpenGL mempunyai banyak fungsi dan penggunaan perintah yang sangat luas, penggunaan OpenGL membutuhkan library tambahan yang harus di letakkan pada direktori sistem dari windows (OS).[5]

2.4 FOOD_3D Di OpenGL

Membuat Food_3D dengan 4 objek yaitu (Donat, Lolipop, Apel, Jeruk)

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 <stdbool.h>

//Kamilaeni
float angle = 0.0f; // Sudut rotasi
float rotationSpeed = 4.0f; // Kecepatan rotasi
float cameraPosX = 5.0f; // Posisi X kamera
float cameraPosY = 5.0f; // Posisi Y kamera
float cameraPosZ = 7.0f; // Posisi Z kamera
float scaleFactor = 1.0f; // Faktor skala objek, defaultnya 1.0
(ukuran normal)
bool showCartecius = true; // Status sumbu kartesius

// Fungsi untuk menggambar donat
void drawDonut() {
    glPushMatrix();
    GLfloat creamColor[] = {1.0f, 0.75f, 0.8f}; // Warna strawberry
    susu
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, creamColor); //
    Set warna material
    glRotatef(180.0f, 1.0f, 0.0f, 0.0f); // Perputaran
    glutSolidTorus(0.2, 0.4, 32, 32);
    glPopMatrix();
}

//Anyelir
// Fungsi untuk menggambar lolipop
void drawLollipop() {
    glPushMatrix();
```

```

// Kepala lolipop
GLfloat headColor[] = {1.0f, 0.0f, 0.0f}; // Warna merah
kepala lolipop
glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, headColor);
glTranslatef(0.0f, 0.5f, 0.0f);
glutSolidSphere(0.4, 32, 32);

// Batang lolipop
GLfloat stickColor[] = {0.8f, 0.8f, 0.8f}; // Warna abu-abu
batang
glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, stickColor); //
Set warna material
glTranslatef(0.0f, -1.0f, 0.0f);
glScalef(0.1f, 2.0f, 0.1f); // Skala batang
glutSolidCube(1.0); //kubus

glPopMatrix();
}

// Fungsi untuk menggambar daun
void drawDaun() {
    glPushMatrix();

    // Warna daun hijau
    GLfloat leafColor[] = {0.0f, 0.8f, 0.0f};
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, leafColor);

    // Bentuk daun (menggunakan disk)
    glRotatef(90.0f, 1.0f, 0.0f, 0.0f); // Posisikan daun
horizontal
    gluDisk(gluNewQuadric(), 0.0, 0.15, 32, 1); //Gambar daun sbg
disk/cakram/bulat

    glPopMatrix();
}

//Kamilaeni
// Fungsi untuk menggambar apel

```

```

void drawApple() {
    glPushMatrix();

    // Badan apel
    GLfloat appleColor[] = {1.0f, 0.0f, 0.0f}; // Warna merah
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, appleColor);
    glutSolidSphere(0.4, 32, 32);

    // Tangkai apel
    GLfloat stemColor[] = {0.4f, 0.2f, 0.0f}; // Warna coklat
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, stemColor);
    glPushMatrix();
    glTranslatef(0.0f, 0.4f, 0.0f);
    glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);
    gluCylinder(gluNewQuadric(), 0.04, 0.04, 0.3, 16, 16);
    glPopMatrix();

    // Daun apel
    glPushMatrix();
    glTranslatef(0.1f, 0.5f, 0.0f); // Posisikan daun di samping
    tangkai
    glRotatef(45.0f, 0.0f, 0.0f, 1.0f); // Rotasi untuk posisi
    daun
    drawDaun();
    glPopMatrix();

    glPopMatrix();
}

//Anyelir
// Fungsi untuk menggambar jeruk
void drawOrange() {
    glPushMatrix();

    // Badan jeruk
    GLfloat orangeColor[] = {1.0f, 0.5f, 0.0f}; // Warna oranye
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, orangeColor);
    glutSolidSphere(0.4, 32, 32);

```

```

// Tangkai jeruk
GLfloat stemColor[] = {0.4f, 0.2f, 0.0f}; // Warna coklat
glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, stemColor);
glPushMatrix();
glTranslatef(0.0f, 0.4f, 0.0f);
glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);
gluCylinder(gluNewQuadric(), 0.04, 0.04, 0.3, 16, 16);
glPopMatrix();

// Daun jeruk
glPushMatrix();
glTranslatef(0.1f, 0.5f, 0.0f); // Posisikan daun di samping
tangkai
glRotatef(45.0f, 0.0f, 0.0f, 1.0f); // Rotasi untuk posisi
daun
drawDaun();
glPopMatrix();

glPopMatrix();
}

//Kamilaeni
// Fungsi untuk menggambar sumbu kartesius
void drawCartecius() {
    if (!showCartecius) return; // Jika showCartecius false,
hilang

    glDisable(GL_LIGHTING); // Nonaktifkan pencahayaan untuk garis
glBegin(GL_LINES);

    // Sumbu X (merah)
glColor3f(1.0f, 0.0f, 0.0f);
glVertex3f(-10.0f, 0.0f, 0.0f);
glVertex3f(10.0f, 0.0f, 0.0f);

    // Sumbu Y (hijau)
glColor3f(0.0f, 1.0f, 0.0f);

```

```

glVertex3f(0.0f, -10.0f, 0.0f);
glVertex3f(0.0f, 10.0f, 0.0f);

// Sumbu Z (biru)
glColor3f(0.0f, 0.0f, 1.0f);
glVertex3f(0.0f, 0.0f, -10.0f);
glVertex3f(0.0f, 0.0f, 10.0f);

glEnd();
glEnable(GL_LIGHTING); // Aktifkan kembali pencahayaan
}

//Anyelir
// Fungsi untuk menggambar latar belakang dengan dinding papan
kayu dan lantai kotak-kotak
void drawBackground() {
    glDisable(GL_LIGHTING); // Nonaktifkan pencahayaan untuk latar
    belakang

    // Gambar lantai dengan pola kotak-kotak
    for (float z = -10.0f; z <= 10.0f; z += 1.0f) {
        for (float x = -10.0f; x <= 10.0f; x += 1.0f) {
            if ((int)(x + z) % 2 == 0) { // Untuk menggambar
            elemen/objek kotak dengan pola berulang, dengan posisi berbeda x+z
                glColor3f(0.7f, 0.4f, 0.2f); // Warna coklat
            terang
            } else {
                glColor3f(0.6f, 0.3f, 0.1f); // Warna coklat gelap
            }
            glPushMatrix();
            glTranslatef(x, -5.0f, z); // Posisi kotak di lantai
            glScalef(1.0f, 0.1f, 1.0f); // Skala kotak
            glutSolidCube(1.0f);
            glPopMatrix();
        }
    }

    // Warna papan kayu coklat

```

```

    GLfloat woodColor[] = {0.55f, 0.27f, 0.07f}; // Warna coklat
    kayu

    // Gambar dinding (papan kayu vertikal)
    for (float i = -10.0f; i <= 10.0f; i += 1.0f) {
        // Dinding belakang
        glPushMatrix();
        glTranslatef(i, 0.0f, -10.0f); // Posisi papan belakang
        menyentuh lantai
        glScalef(1.0f, 10.0f, 0.1f);    // Skala papan
        glColor3fv(woodColor);
        glutSolidCube(1.0f);
        glPopMatrix();

        // Dinding depan
        glPushMatrix();
        glTranslatef(i, 0.0f, 10.0f); // Posisi papan depan
        menyentuh lantai
        glScalef(1.0f, 10.0f, 0.1f);    // Skala papan
        glColor3fv(woodColor);
        glutSolidCube(1.0f);
        glPopMatrix();
    }

    for (float i = -10.0f; i <= 10.0f; i += 1.0f) {
        // Dinding kiri
        glPushMatrix();
        glTranslatef(-10.0f, 0.0f, i); // Posisi papan kiri
        menyentuh lantai
        glScalef(0.1f, 10.0f, 1.0f);    // Skala papan
        glColor3fv(woodColor);
        glutSolidCube(1.0f);
        glPopMatrix();

        // Dinding kanan
        glPushMatrix();
        glTranslatef(10.0f, 0.0f, i); // Posisi papan kanan
        menyentuh lantai

```



```

        glScalef(0.1f, 10.0f, 1.0f);    // Skala papan
        glColor3fv(woodColor);
        glutSolidCube(1.0f);
        glPopMatrix();
    }

    // Langit-langit (seperti papan)
    for (float i = -10.0f; i <= 10.0f; i += 1.0f) {
        glPushMatrix();
        glTranslatef(i, 10.0f, 0.0f); // Posisi papan langit-
langit di atas objek
        glScalef(1.0f, 0.1f, 20.0f);    // Skala papan
        glColor3fv(woodColor);
        glutSolidCube(1.0f);
        glPopMatrix();
    }
    glEnable(GL_LIGHTING); // Aktifkan kembali pencahayaan
}

//Kamilaeni
void display() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();

    // Atur kamera
    gluLookAt(cameraPosX, cameraPosY, cameraPosZ, 0.0, 0.0, 0.0,
0.0, 1.0, 0.0);

    // Gambar latar belakang
    drawBackground();

    // Gambar objek
    glPushMatrix();
    drawCartecius();

    glPushMatrix();
    glTranslatef(-3.0f, 0.0f, 0.0f);

```

```

        glRotatef(angle, 0.0f, 1.0f, 0.0f);
        glScalef(scaleFactor, scaleFactor, scaleFactor); // Terapkan
skala
        drawDonut();
        glPopMatrix();

        glPushMatrix();
        glTranslatef(-1.0f, 0.0f, 0.0f);
        glRotatef(angle, 0.0f, 1.0f, 0.0f);
        glScalef(scaleFactor, scaleFactor, scaleFactor); // Terapkan
skala
        drawLolipop();
        glPopMatrix();

        glPushMatrix();
        glTranslatef(1.0f, 0.0f, 0.0f);
        glRotatef(angle, 0.0f, 1.0f, 0.0f);
        glScalef(scaleFactor, scaleFactor, scaleFactor); // Terapkan
skala
        drawApple();
        glPopMatrix();

        glPushMatrix();
        glTranslatef(3.0f, 0.0f, 0.0f);
        glRotatef(angle, 0.0f, 1.0f, 0.0f);
        glScalef(scaleFactor, scaleFactor, scaleFactor); // Terapkan
skala
        drawOrange();
        glPopMatrix();

        glPopMatrix();
        glutSwapBuffers();
}

//Anyelir
// Fungsi untuk memperbarui animasi saat berputar
void animasi(int value) {
    angle += rotationSpeed; // Tambah sudut rotasi

```

```

    if (angle > 360.0f) {
        angle -= 360.0f;
    }

    glutPostRedisplay(); // Meminta redisplay
    glutTimerFunc(15, animasi, 0); // Memanggil fungsi ini lagi
    setelah 16 ms
}

//Kamilaeni
// Fungsi untuk menangani input keyboard
void keyboard(unsigned char key, int x, int y) {
    switch (key) {
        case 'w': // Maju
            cameraPosZ -= 0.1f;
            break;
        case 's': // Mundur
            cameraPosZ += 0.1f;
            break;
        case 'a': // Kiri
            cameraPosX -= 0.1f;
            break;
        case 'd': // Kanan
            cameraPosX += 0.1f;
            break;
        case 'q': // Naik
            cameraPosY += 0.1f;
            break;
        case 'e': // Turun
            cameraPosY -= 0.1f;
            break;
        case 'c': // Toggle visibilitas sumbu kartesius
            showCartecius = !showCartecius;
            break;
        case 'z': // Perbesar objek
            scaleFactor += 0.1f;
            break;
        case 'x': // Perkecil objek
            if (scaleFactor > 0.1f) { // Jangan biarkan skala

```

```

menjadi terlalu kecil
        scaleFactor -= 0.1f;
    }
    break;
case 27: // Tombol ESC untuk keluar
    exit(0);
    break;
default:
    break;
}
glutPostRedisplay(); // Minta redisplay setelah perubahan
}

//Anyelir
// Inisialisasi
void init() {
    glEnable(GL_DEPTH_TEST); //kedalaman cahaya
    glEnable(GL_LIGHTING); //megaktifkan pencahayaan
    glEnable(GL_LIGHT0);

    glClearColor(0.1f, 0.1f, 0.1f, 1.0f);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(45.0, 1.0, 1.0, 100.0); // 45 derajat
    glMatrixMode(GL_MODELVIEW);

    GLfloat lightPos[] = {3.0f, 3.0f, 3.0f, 1.0f};
    GLfloat lightAmbient[] = {0.2f, 0.2f, 0.2f, 1.0f};
    GLfloat lightDiffuse[] = {0.8f, 0.8f, 0.8f, 1.0f};
    GLfloat lightSpecular[] = {1.0f, 1.0f, 1.0f, 1.0f};

    glLightfv(GL_LIGHT0, GL_POSITION, lightPos);
    glLightfv(GL_LIGHT0, GL_AMBIENT, lightAmbient);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, lightDiffuse);
    glLightfv(GL_LIGHT0, GL_SPECULAR, lightSpecular);

    // Properti pencahayaan
    GLfloat matAmbient[] = {0.8f, 0.8f, 0.8f, 1.0f};

```

```

GLfloat matDiffuse[] = {1.0f, 1.0f, 1.0f, 1.0f};
GLfloat matSpecular[] = {1.0f, 1.0f, 1.0f, 1.0f};
GLfloat matShininess[] = {50.0f}; //tingkat kilauan

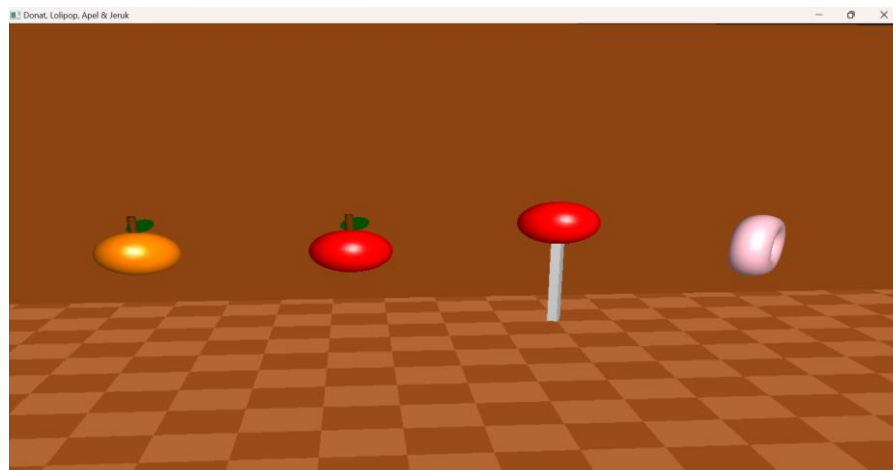
glMaterialfv(GL_FRONT, GL_AMBIENT, matAmbient);
glMaterialfv(GL_FRONT, GL_DIFFUSE, matDiffuse);
glMaterialfv(GL_FRONT, GL_SPECULAR, matSpecular);
glMaterialfv(GL_FRONT, GL_SHININESS, matShininess);
}

//Kamilaeni
// Fungsi utama
int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(800, 600);
    glutCreateWindow("Food_3D");

    init();
    glutDisplayFunc(display);
    glutKeyboardFunc(keyboard);
    glutTimerFunc(15, animasi, 0);
    glutMainLoop();
    return 0;
}

```

3.2 Output



3.3 Penjelasan

kode ini menggambarkan objek-objek 3D yang terdiri dari donat, lolipop, apel, dan jeruk, serta menambahkan latar belakang dengan dinding papan kayu dan lantai kotak-kotak. Kamera dapat digerakkan dengan keyboard, dan objek-objek ini akan berputar secara animasi. Sumbu kartesius dapat ditampilkan atau disembunyikan menggunakan tombol c.

A. Variable

1. Angle : Mengatur sudut rotasi objek.
2. rotationSpeed : Menentukan kecepatan rotasi objek.
3. cameraPosX, cameraPosY, cameraPosZ: Mengatur posisi kamera.
4. scaleFactor : Faktor skala objek (digunakan untuk memperbesar atau memperkecil objek).
5. showCartecius : Boolean untuk menampilkan atau menyembunyikan sumbu kartesius.

B. Fungsi Utama Objek

1. drawDonut :
 - Menggambar donat 3D menggunakan glutSolidTorus.
 - Warna donat disetel dengan glMaterialfv.
2. drawLollipop :
 - Kepala lolipop berbentuk bola menggunakan glutSolidSphere.
 - Batang lolipop berbentuk persegi panjang menggunakan glutSolidCube.
3. drawLeaf :
 - Menggambar daun berbentuk lingkaran tipis menggunakan gluDisk dengan warna hijau.
4. drawApple :
 - Menggambar apel (bola merah) dengan tangkai (silinder coklat) dan daun (lingkaran hijau).
5. drawOrange:
 - Mirip dengan apel, tetapi warna jeruk adalah oranye.
6. drawCartecius :
 - Menggambar sumbu kartesius (X, Y, Z) sebagai garis warna merah, hijau, dan biru.
7. drawBackground :
 - Menggambar latar belakang dengan:
8. Lantai kayu berpola kotak-kotak.

- Dinding dan langit-langit berbentuk papan kayu.

C. Fungsi Display

1. Display :

- Fungsi utama untuk menggambar seluruh objek dan elemen pada layar.
- Kamera diatur menggunakan gluLookAt untuk melihat ke pusat koordinat.
- Objek digambar dengan berbagai transformasi seperti translasi, rotasi, dan skala.

2. Animasi/perputaran :

- Memperbarui sudut rotasi objek setiap 16 ms (60 FPS).
- Memastikan rotasi tetap dalam kisaran 0° - 360° .

D. Input Keyboard

1. Fungsi keyboard menangani input keyboard untuk mengontrol animasi :

- **w, s** : Maju dan mundur kamera (Z).
- **a, d** : Geser kamera ke kiri atau kanan (X).
- **q, e** : Naik dan turun kamera (Y).
- **c** : Toggle visibilitas sumbu kartesius.
- **z, x** : Perbesar atau perkecil objek.
- **Esc** : Keluar dari aplikasi.

E. Inisialisasi OpenGL

1. Fungsi init/inisialisasi :

- Mengaktifkan pencahayaan (GL_LIGHTING, GL_LIGHT0) dan kedalaman (GL_DEPTH_TEST).
- Pencahayaan diatur dengan properti seperti posisi, intensitas, dan warna.
- Warna latar belakang ditetapkan ke abu-abu gelap.

F. Fungsi main

1. Memulai program dengan inisialisasi GLUT.

- Membuat jendela 3D ukuran 800x600
- Mengatur fungsi callback untuk display, keyboard, dan update.
- Menjalankan glutMainLoop untuk memulai loop aplikasi.

BAB IV

4.1. Kesimpulan

Laporan ini membahas pemanfaatan OpenGL (Open Graphics Library) dalam pembuatan simulasi objek makanan 3D, yaitu donat, lolipop, apel, dan jeruk. OpenGL dipilih karena merupakan pustaka grafis lintas platform yang mendukung fitur-fitur seperti pencahayaan, tekstur, dan animasi, sehingga memungkinkan penggambaran grafika 2D dan 3D secara real-time. Bahasa pemrograman yang digunakan adalah C++, yang dipilih karena efisiensinya serta kemampuannya dalam pengolahan tingkat rendah. Proses konfigurasi dilakukan menggunakan Dev C++ atau VSCode dengan menyusun langkah-langkah teknis untuk memastikan pustaka OpenGL dapat digunakan dengan benar, seperti menambahkan parameter linker dan membuat proyek baru.

Output yang dihasilkan mencakup berbagai elemen, seperti objek-objek makanan 3D, latar belakang berupa lantai berpola kotak-kotak dan dinding papan kayu, serta sumbu kartesius yang divisualisasikan. Posisi kamera juga di akses melalui input dari keyboard, sehingga bisa melihat dari perspektif objek. Codingan ini juga dilengkapi dengan berbagai fungsi utama, seperti fungsi display untuk menampilkan elemen-elemen grafis pada layar, fungsi inisialisasi untuk mengatur pencahayaan dan warna latar belakang, serta fungsi keyboard untuk memproses interaksi pengguna.

Keunggulan OpenGL sebagai pustaka grafis terlihat pada kemampuannya untuk menghasilkan grafika yang realistis dan interaktif, dengan portabilitas yang tinggi karena dapat diakses melalui berbagai bahasa pemrograman dan platform. Simulasi ini menunjukkan bagaimana OpenGL dapat dimanfaatkan secara efektif untuk merepresentasikan objek 3D secara real-time, serta memberikan pengalaman belajar yang mendalam mengenai implementasi dan pengaplikasian konsep grafika komputer. Kesimpulannya, laporan ini berhasil mendemonstrasikan bagaimana teknologi grafika komputer dapat digunakan untuk menciptakan visualisasi yang menarik dan interaktif, sekaligus memberikan wawasan praktis dalam pengembangan aplikasi berbasis OpenGL.

DAFTAR PUSTAKA

- [1] “[1] D. Suhardiman *et al.*, “Pembuatan Simulasi Pergerakan Objek 3D (Tiga Dimensi) Menggunakan OpenGL,” *J. Inform.*, vol. 2, no. 1, pp. 1–4, 2015.
- [2] Muhammad Adnani and Achmad Zakki Falani, “Implementasi Open Gl Untuk Pembuatan Objek 3d,” *J. Zetroem*, vol. 3, no. 1, pp. 1–6, 2021, doi: 10.36526/ztr.v3i1.1249.
- [3] M. B. Priyantono and A. A. Rachmawan, “Implementasi Sistem Simulasi Penampilan Tata Surya Berbasis 3D Menggunakan Opengl,” *J. Teknol. Inf.*, vol. 4, no. 1, pp. 91–95, 2020, doi: 10.36294/jurti.v4i1.1231.
- [4] J. Hendrik and T. Chandra, “Pemanfaatan Sintaks OpenGL dalam Menggambar Obyek Dua Dimensi,” *J. TIMES*, vol. 4, no. 1, pp. 22–25, 2015, doi: 10.51351/jtm.4.1.2015218.
- [5] J. Widadi and A. Dahlan Jl ProfDrSoepomo, “Media Pembelajaran Materi Pengenalan Opengl Pada Mata Kuliah Grafika Komputer,” *J. Sarj. Tek. Inform.*, vol. 6, no. 1, pp. 47–53, 2018, [Online]. Available: <http://journal.uad.ac.id/index.php/JSTIF>