

LAPORAN PRATIUM GRAFIK KOMPUTER

Diajukan untuk memenuhi Tugas mata kuliah Pratikum Grafik Komputer

PEMBUATAN PROGRAM BAHASA C MENGGUNAKAN OPENGL (PROGRAM RESTORAN MAKANAN)

Dosen Pengampu : Sri Rahayu, M.Kom

Instruktur Pratikum : Arul Budi Kalimat, S.Kom



Disusun oleh

Kelompok : 10

Bagas Sujiwo
2306018

FirmanNurHakim
2306107

Romy Zaenul A
2306019

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN ILMU KOMPUTER

INSTITUT TEKNOLOGI GARUT

2025

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 [topik praktikum].

Kami mengucapkan terima kasih kepada dosen pengampu Sri Rahayu S.Kom 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, 10 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	1
1.3 Tujuan.....	1
BAB II TINJAUAN PUSTAKA	2
2.1 OpenGL.....	2
2.2 Konfigurasi OpenGL pada Dev C++	2
2.1 Cara Kerja OpenGL	6
2.2 Program Restoran Makanan dengan Fokus Tema Makanan Di OpenGL.....	7
BAB III HASIL.....	8
3.1 Source Code.....	8
3.2 Output	15
3.3 Penjelasan.....	15
BAB IV	17
4.1. Kesimpulan.....	17
DAFTAR PUSTAKA	18

DAFTAR GAMBAR

Gambar 1 File Mentahan OpenGL.....	2
Gambar 2 Proses Copy Mentahan	3
Gambar 3 Tempat Menyimpan Mentahan OpenGL.....	3
Gambar 4 Pembuatan Project	4
Gambar 5 Konfigurasi Dev C++	4
Gambar 6 Konfigurasi selesai	5
Gambar 7 Inisialisasi	6
Gambar 8 Menggambar Objek	7
Gambar 9 Output Program Restoran Makanan	15

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia teknologi modern, grafik komputer telah menjadi salah satu bidang yang sangat penting dan luas aplikasinya. Dari permainan video hingga aplikasi simulasi kompleks, grafik komputer memainkan peran krusial dalam menghasilkan visualisasi yang realistis dan interaktif. OpenGL (Open Graphics Library) adalah salah satu teknologi yang paling banyak digunakan dalam pengembangan aplikasi grafik komputer. OpenGL adalah suatu API (Application Programming Interface) yang cross-platform dan dirancang untuk rendering grafik 2D dan 3D dengan efisien. Dengan kemampuan yang dimilikinya, OpenGL telah menjadi standar industri dalam pengembangan aplikasi yang membutuhkan rendering grafik tinggi, seperti dalam industri game, desain CAD, dan visualisasi ilmiah.[1]

Praktikum yang dilakukan dalam laporan ini bertujuan untuk memberikan pemahaman yang lebih dalam tentang konsep-konsep dasar OpenGL serta penerapannya dalam pembuatan aplikasi grafik komputer. Melalui praktikum ini, mahasiswa diharapkan dapat memahami cara kerja OpenGL, mengimplementasikan fungsi-fungsi OpenGL dalam program, dan menghasilkan output grafik yang diinginkan. Praktikum ini juga bertujuan untuk meningkatkan keterampilan praktik dalam pengembangan aplikasi grafik, yang merupakan keterampilan yang sangat dibutuhkan dalam industri teknologi saat ini.

Studi tentang OpenGL tidak hanya penting dari sudut pandang aplikatif, tetapi juga dari sudut pandang teoritis. Pemahaman tentang bagaimana OpenGL bekerja dapat memberikan wawasan tentang bagaimana sistem operasi dan perangkat keras bekerja sama dalam menghasilkan grafik yang kompleks. Selain itu, pemahaman ini juga dapat membantu dalam mengoptimalkan performa aplikasi grafik, yang merupakan faktor krusial dalam pengembangan aplikasi modern.[2]

Dalam konteks pendidikan, praktikum OpenGL sangat penting karena memberikan kesempatan kepada mahasiswa untuk mengaplikasikan teori yang mereka pelajari dalam setting praktik. Melalui praktikum ini, mahasiswa dapat memahami konsep-konsep seperti rendering, transformasi, dan pemrograman grafik dengan lebih baik. Selain itu, praktikum ini juga membantu dalam mengembangkan keterampilan problem-solving dan pemrograman yang diperlukan dalam pengembangan aplikasi grafik komputer.

1.2 Rumusan Masalah

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

1.3 Tujuan

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 Program Restoran Makanan dalam OpenGL

BAB II

TINJAUAN PUSTAKA

2.1 OpenGL

OpenGL (Open Graphics Library) adalah standar API (Application Programming Interface) yang digunakan untuk mengembangkan aplikasi grafis dua dimensi (2D) dan tiga dimensi (3D). Sebagai antarmuka pemrograman aplikasi, OpenGL memungkinkan pengembang untuk membuat grafik yang dapat dijalankan di berbagai platform sistem operasi, menjadikannya cross-platform. Awalnya, OpenGL dikembangkan oleh Silicon Graphics Inc. (SGI) sebagai penerus dari IRIS GL, dengan tujuan menyediakan API grafis yang lebih portabel dan standar. Evolusi ini memungkinkan OpenGL untuk mendukung berbagai jenis perangkat keras dan sistem operasi, sehingga menjadi pilihan utama dalam pengembangan aplikasi grafis. [1]

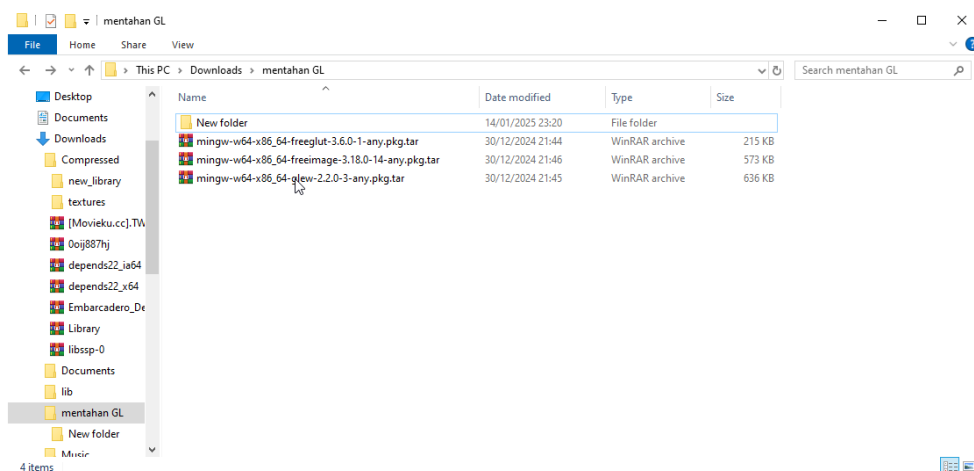
Dalam implementasinya, OpenGL menyediakan berbagai fungsi untuk menggambar objek grafis, termasuk titik, garis, dan poligon dalam ruang tiga dimensi. Selain itu, OpenGL mendukung fitur-fitur seperti pencahayaan, pewarnaan, tekstur, dan efek khusus lainnya yang memungkinkan pembuatan grafik yang realistis dan interaktif. Untuk mempermudah pengembangan aplikasi grafis, OpenGL sering digunakan bersama dengan toolkit seperti GLUT (OpenGL Utility Toolkit). GLUT menyediakan fungsi-fungsi tambahan untuk manajemen jendela, input pengguna, dan rendering, sehingga memudahkan pengembang dalam membuat aplikasi grafis yang kompleks. [3]

Secara keseluruhan, OpenGL merupakan alat yang sangat penting dalam bidang grafika komputer, menyediakan fondasi yang kuat untuk pengembangan aplikasi grafis yang dapat dijalankan di berbagai platform dan mendukung berbagai fitur grafis canggih.

2.2 Konfigurasi OpenGL pada Dev C++

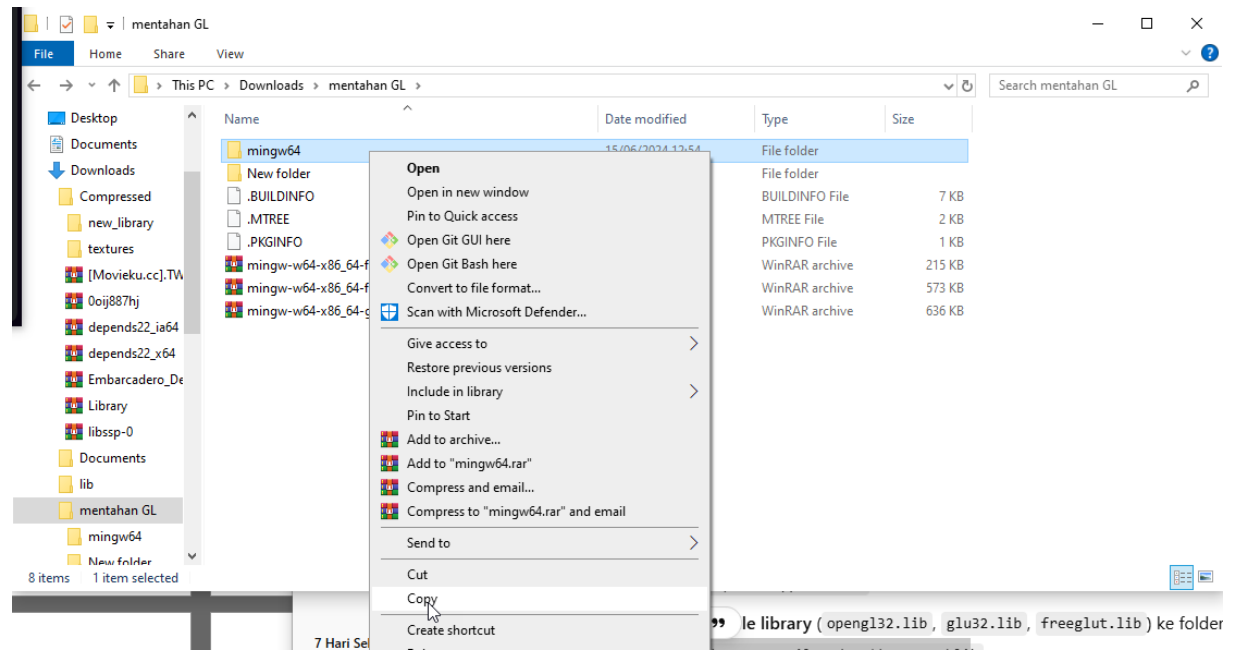
Uraikan dan Jelaskan langkah-langkah untuk konfigurasi OpenGL pada Dev C++. Setiap langkah harus menyertakan Screenshoot. Semua gambar ukurannya harus sama dan posisi gambar berada ditengah (center) dan berikan keterangan dari setiap gambar. Contoh :

1. Siapkan mentahan OpenGL agar bisa terkonfigurasi di Dev C++



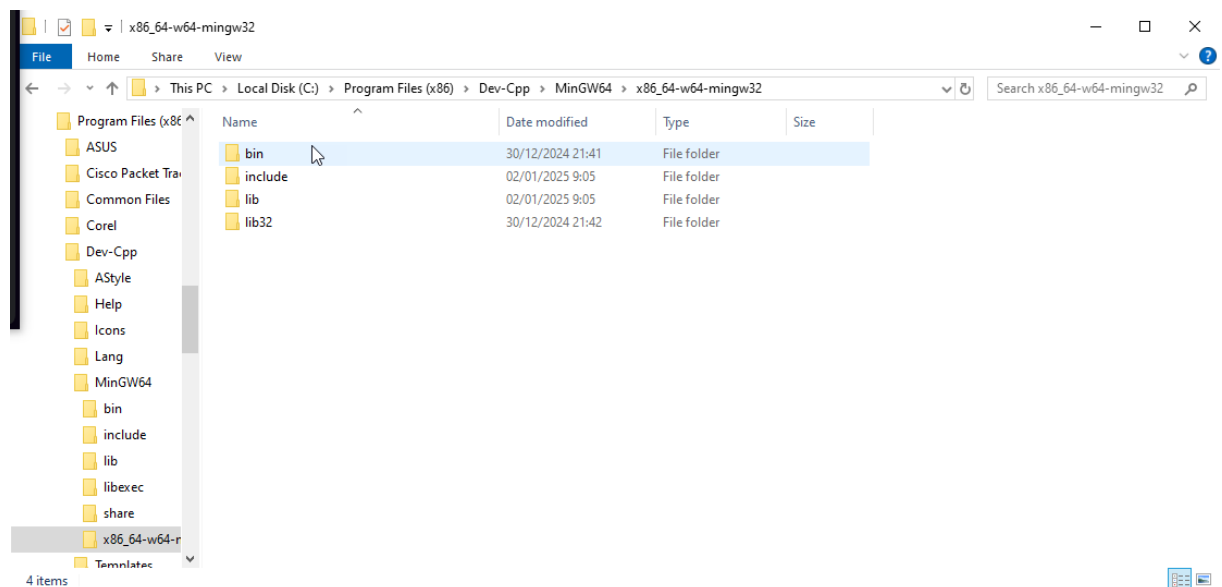
Gambar 1 File Mentahan OpenGL

2. Ekstrak semua file mentahan dan satukan
3. Copy semua file mingw64 yang terdapat freeglut,glew,freeimage



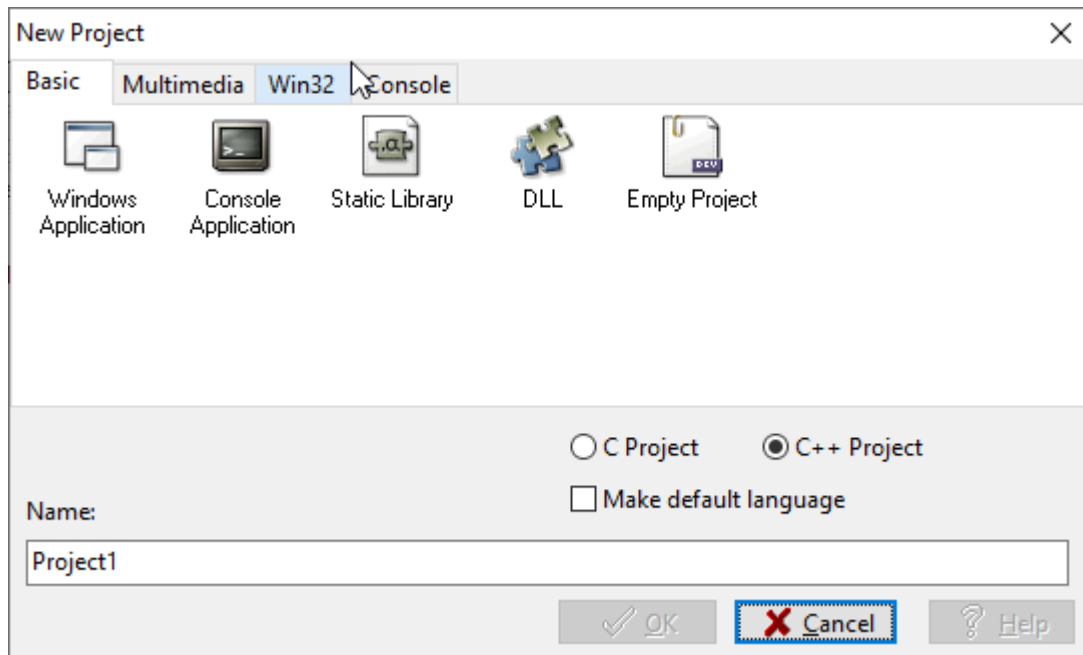
Gambar 2 Proses Copy Mentahan

4. Copy kan semua file dalam folder mingw64 ke > C:\Program Files (x86)\Dev-Cpp\MinGW64\x86_64-w64-mingw32



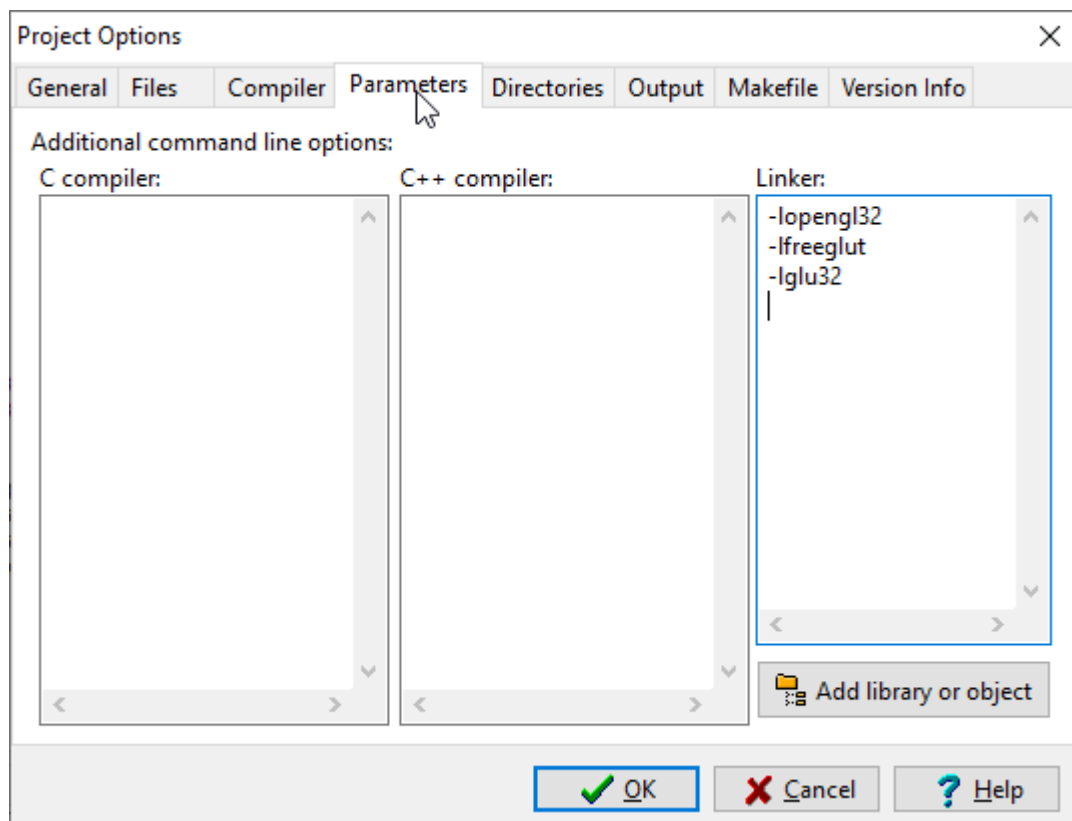
Gambar 3 Tempat Menyimpan Mentahan OpenGL

5. Setelah selesai masuk ke Dev C++ dan pilih project lalu Console Application



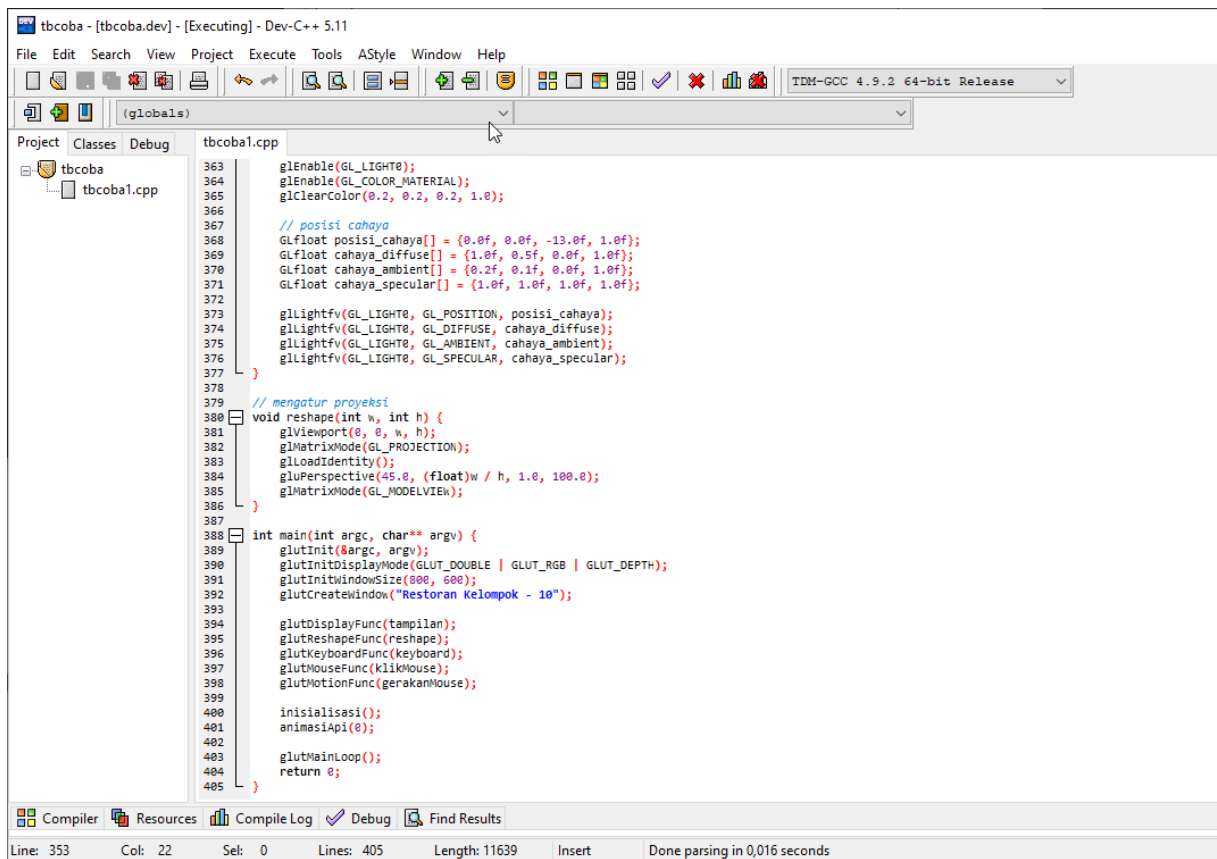
Gambar 4 Pembuatan Project

6. Masuk ke Project Option > parameter > isikan dengan (opengl32.lib, glu32.lib, freeglut.lib)



Gambar 5 Konfigurasi Dev C++

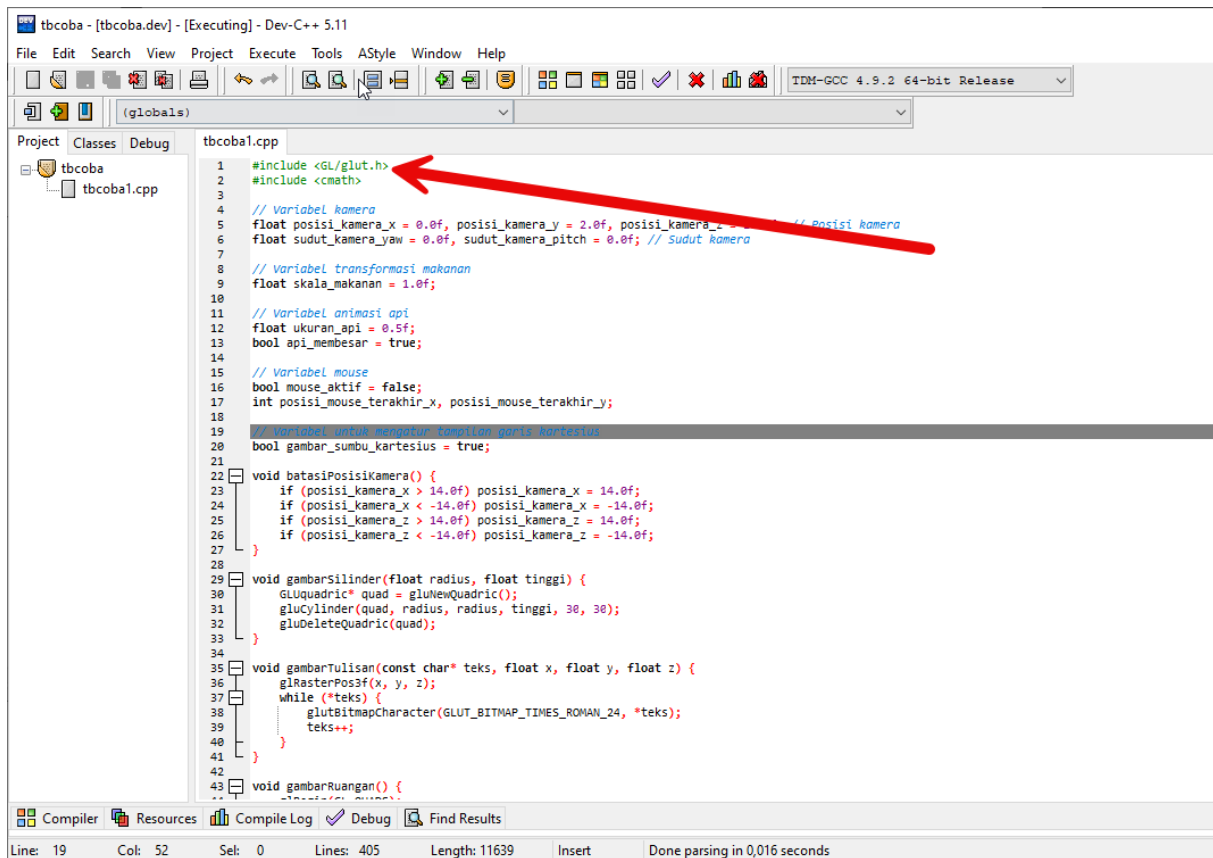
7. Dan Konfigurasi pun selesai siap untuk di jalankan



Gambar 6 Konfigurasi selesai

2.3 Cara Kerja OpenGL

cara kerja opengl yang pertama memasukan `#include<GL/glut.h>`, untuk mengidentifikasi opengl.



Dalam opengl terdapat syntax syntax yang bisa digunakan seperti :

- Inisialisasi OpenGL
glutInit() menginisialisasi pustaka GLUT.
glutCreateWindow() membuat jendela untuk aplikasi OpenGL.
glutMainLoop() memulai loop utama yang terus berjalan hingga aplikasi ditutup.

```
int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(800, 600);
    glutCreateWindow("Restoran Kelompok - 10");

    glutDisplayFunc(tampilan);
    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
    glutMouseFunc(klikMouse);
    glutMotionFunc(gerakanMouse);

    inisialisasi();
    animasiApi(0);

    glutMainLoop();
    return 0;
}
```

Gambar 7 Inisialisasi

- Menggambar Objek
`glClear(GL_COLOR_BUFFER_BIT)` membersihkan buffer warna, memastikan area gambar bersih sebelum menggambar objek baru.
`glBegin(GL_TRIANGLES)` memulai menggambar objek segitiga.
`glVertex2f(x, y)` mendefinisikan titik-titik segitiga dalam koordinat 2D.
`glEnd()` mengakhiri definisi objek yang sedang digambar.
`glFlush()` memastikan bahwa semua perintah rendering diproses dan ditampilkan.

```
void display() {
    glClear(GL_COLOR_BUFFER_BIT); // Menghapus buffer warna
    glBegin(GL_TRIANGLES);       // Memulai menggambar segitiga
        glVertex2f(0.0f, 1.0f);  // Titik pertama segitiga
        glVertex2f(-1.0f, -1.0f); // Titik kedua segitiga
        glVertex2f(1.0f, -1.0f); // Titik ketiga segitiga
    glEnd();                     // Mengakhiri menggambar objek
    glFlush();                   // Memastikan semua perintah dieksekusi
}
```

Gambar 8 Menggambar Objek

2.4 Program Restoran Makanan dengan Fokus Tema Makanan Di OpenGL

Pembuatan program restoran makanan menggunakan OpenGL melibatkan beberapa tahapan yang dimulai dengan perancangan tampilan antarmuka grafis. Pada tahap ini, kita akan menggambar objek-objek 2D dan 3D yang mewakili elemen-elemen restoran, seperti meja, kursi, menu makanan, dan dekorasi. Untuk menggambar objek-objek tersebut, OpenGL menyediakan berbagai fungsi grafis yang memungkinkan pembuatan bentuk dasar seperti persegi panjang, lingkaran, dan poligon, yang kemudian dapat dimodifikasi lebih lanjut dengan pengaturan warna, pencahayaan, dan tekstur. Salah satu aspek penting dalam tahap ini adalah penggunaan transformasi geometris, seperti translasi, rotasi, dan skala, untuk memanipulasi posisi dan orientasi objek dalam ruang 3D.

Selain itu, interaktivitas menjadi elemen kunci dalam pembuatan program restoran makanan ini. Pengguna diharapkan dapat memilih makanan dari menu atau berinteraksi dengan berbagai elemen dalam restoran, seperti memilih meja atau mengubah tampilan ruang. OpenGL memungkinkan penerapan input pengguna melalui perangkat seperti mouse atau keyboard, yang kemudian diterjemahkan menjadi aksi dalam program, misalnya memilih makanan atau menavigasi antar area restoran. Implementasi tekstur pada objek 3D, seperti tampilan makanan dan interior restoran, akan memberikan nuansa realistis pada tampilan visual program. Dengan menggabungkan elemen-elemen grafis dan interaktivitas ini, program restoran makanan dalam OpenGL dapat memberikan pengalaman pengguna yang menarik dan menyenangkan.

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 <cmath>

// Variabel kamera
float posisi_kamera_x = 0.0f, posisi_kamera_y = 2.0f,
posisi_kamera_z = 10.0f; // Posisi kamera
float sudut_kamera_yaw = 0.0f, sudut_kamera_pitch = 0.0f; // Sudut
kamera

// Variabel transformasi makanan
float skala_makanan = 1.0f;

// Variabel animasi api
float ukuran_api = 0.5f;
bool api_membesar = true;

// Variabel mouse
bool mouse_aktif = false;
int posisi_mouse_terakhir_x, posisi_mouse_terakhir_y;

// Variabel untuk mengatur tampilan garis kartesius
bool gambar_sumbu_kartesius = true;

void batasiPosisiKamera() {
    if (posisi_kamera_x > 14.0f) posisi_kamera_x = 14.0f;
    if (posisi_kamera_x < -14.0f) posisi_kamera_x = -14.0f;
    if (posisi_kamera_z > 14.0f) posisi_kamera_z = 14.0f;
    if (posisi_kamera_z < -14.0f) posisi_kamera_z = -14.0f;
}

void gambarSilinder(float radius, float tinggi) {
    GLUQuadric* quad = gluNewQuadric();
    gluCylinder(quad, radius, radius, tinggi, 30, 30);
    gluDeleteQuadric(quad);
}

void gambarTulisan(const char* teks, float x, float y, float z) {
    glRasterPos3f(x, y, z);
    while (*teks) {
        glutBitmapCharacter(GLUT_BITMAP_TIMES_ROMAN_24, *teks);
        teks++;
    }
}

void gambarRuangan() {
    glBegin(GL_QUADS);

    // Lantai
    glColor3f(1.0, 1.0, 1.0);
```

```

glVertex3f(-15.0, -2.0, -15.0);
glVertex3f(15.0, -2.0, -15.0);
glVertex3f(15.0, -2.0, 15.0);
glVertex3f(-15.0, -2.0, 15.0);

// Atap
glColor3f(0.3, 0.2, 0.3);
glVertex3f(-15.0, 10.0, -15.0);
glVertex3f(15.0, 10.0, -15.0);
glVertex3f(15.0, 10.0, 15.0);
glVertex3f(-15.0, 10.0, 15.0);

// belakang
glColor3f(0.6, 0.9, 0.6);
glVertex3f(-15.0, -2.0, -15.0);
glVertex3f(15.0, -2.0, -15.0);
glVertex3f(15.0, 10.0, -15.0);
glVertex3f(-15.0, 10.0, -15.0);

// kiri
glColor3f(0.6, 0.9, 0.6);
glVertex3f(-15.0, -2.0, -15.0);
glVertex3f(-15.0, -2.0, 15.0);
glVertex3f(-15.0, 10.0, 15.0);
glVertex3f(-15.0, 10.0, -15.0);

// kanan
glColor3f(0.6, 0.9, 0.6);
glVertex3f(15.0, -2.0, -15.0);
glVertex3f(15.0, -2.0, 15.0);
glVertex3f(15.0, 10.0, 15.0);
glVertex3f(15.0, 10.0, -15.0);

glEnd();

glColor3f(1.0, 1.0, 1.0);
gambarTulisan("<<<< RESTORAN KELOMPOK 10 >>>>", 6.0f, 5.0f,
15.0f);
}

void gambarMeja(float x, float y, float z, float panjang, float
lebar, float tinggi, float rotasi) {
    glPushMatrix();
    glTranslatef(x, y, z);
    glRotatef(rotasi, 0.0f, 1.0f, 0.0f);
    glScalef(panjang, tinggi, lebar);

    glColor3f(0.5, 0.35, 0.05); // Warna kayu
    glutSolidCube(1.5);

    glPopMatrix();
}

void gambarApi(float x, float y, float z) {
    glPushMatrix();
    glTranslatef(x, y, z);

```

```

        glColor3f(1.0, 0.5, 0.0);
        glutSolidSphere(ukuran_api, 20, 20);

        glPopMatrix();
    }

void gambarKompore(float x, float y, float z) {
    glPushMatrix();
    glTranslatef(x, y, z);
    // Badan kompor
    glColor3f(0.2, 0.2, 0.2);
    glScalef(2.0, 0.5, 1.0);
    glutSolidCube(1.0);
    glPopMatrix();
    // Api
    gambarApi(x, y + 0.5, z);
}

void gambarMakananDiMeja(float posisi_meja_x, float posisi_meja_y,
float posisi_meja_z) {
    glPushMatrix();
    glTranslatef(posisi_meja_x, posisi_meja_y + 0.5,
posisi_meja_z);
    glScalef(skala_makanan, skala_makanan, skala_makanan);

    // Piring pertama
    glPushMatrix();
    glColor3f(1.0f, 1.0f, 1.0f);
    glTranslatef(0.0f, -0.3f, -6.0f);
    glScalef(1.5f, 0.1f, 1.5f);
    glutSolidSphere(0.8, 50, 50);
    glPopMatrix();
    // Apel
    glPushMatrix();
    glColor3f(1.0f, 0.0f, 0.0f);
    glTranslatef(0.0f, 0.2f, -6.0f);
    glutSolidSphere(0.4, 30, 30);
    glColor3f(0.0f, 1.0f, 0.0f);
    glTranslatef(0.0f, 0.4f, 0.0f);
    glutSolidCone(0.1, 0.2, 30, 30);
    glPopMatrix();

    // Piring kedua
    glPushMatrix();
    glColor3f(1.0f, 1.0f, 1.0f);
    glTranslatef(0.0f, -0.3f, -2.0f);
    glScalef(1.5f, 0.1f, 1.5f);
    glutSolidSphere(0.8, 50, 50);
    glPopMatrix();
    // Pizza
    glPushMatrix();
    glColor3f(1.0f, 0.8f, 0.6f);
    glTranslatef(0.0f, 0.2f, -2.0f);
    glScalef(1.0f, 0.1f, 1.0f);
    glutSolidSphere(0.6, 30, 30);
    glColor3f(1.0f, 0.0f, 0.0f);
    glTranslatef(0.0f, 0.1f, 0.0f);

```

```

glutSolidSphere(0.5, 30, 30);
glColor3f(1.0f, 1.0f, 0.0f);
glTranslatef(0.0f, 0.1f, 0.0f);
glutSolidSphere(0.4, 30, 30);
glPopMatrix();

// Piring ketiga
glPushMatrix();
glColor3f(1.0f, 1.0f, 1.0f);
glTranslatef(0.0f, -0.3f, 2.0f);
glScalef(1.5f, 0.1f, 1.5f);
glutSolidSphere(0.8, 50, 50);
glPopMatrix();
// kueh
glPushMatrix();
glColor3f(1.0f, 0.8f, 0.6f);
glTranslatef(0.0f, 0.2f, 2.0f);
glScalef(0.8f, 0.5f, 0.8f);
glutSolidCube(0.6);
glColor3f(1.0f, 0.0f, 0.0f);
glTranslatef(0.0f, 0.4f, 0.0f);
glutSolidSphere(0.1, 30, 30);
glPopMatrix();

// Piring keempat
glPushMatrix();
glColor3f(1.0f, 1.0f, 1.0f);
glTranslatef(0.0f, -0.3f, 6.0f);
glScalef(1.5f, 0.1f, 1.5f);
glutSolidSphere(0.8, 50, 50);
glPopMatrix();
// Burger
glPushMatrix();
glColor3f(0.8f, 0.5f, 0.2f);
glTranslatef(0.0f, 0.2f, 6.0f);
glutSolidSphere(0.4, 30, 30);
glColor3f(0.5f, 0.25f, 0.1f);
glTranslatef(0.0f, 0.2f, 0.0f);
glutSolidSphere(0.35, 30, 30);
glColor3f(0.0f, 1.0f, 0.0f);
glTranslatef(0.0f, 0.2f, 0.0f);
glutSolidSphere(0.4, 30, 30);
glColor3f(1.0f, 0.0f, 0.0f);
glTranslatef(0.0f, 0.2f, 0.0f);
glutSolidSphere(0.35, 30, 30);
glColor3f(0.8f, 0.5f, 0.2f);
glTranslatef(0.0f, 0.2f, 0.0f);
glutSolidSphere(0.4, 30, 30);
glPopMatrix();

// Piring kelima
glPushMatrix();
glColor3f(1.0f, 1.0f, 1.0f);
glTranslatef(0.0f, -0.3f, 10.0f);
glScalef(1.5f, 0.1f, 1.5f);
glutSolidSphere(0.8, 50, 50);
glPopMatrix();

```

```

// Minuman
glPushMatrix();
glColor3f(0.0f, 0.0f, 1.0f);
glTranslatef(0.0f, 0.5f, 10.0f);
glScalef(0.5f, 0.8f, 0.5f);
glutSolidCube(0.6);
glColor3f(1.0f, 1.0f, 0.0f);
glTranslatef(0.0f, 0.5f, 0.0f);
glRotatef(90, 1, 0, 0);
glutSolidTorus(0.05, 0.2, 30, 30);
glPopMatrix();

glPopMatrix();
}

void aturKamera() {
    float arah_kamera_x = sin(sudut_kamera_yaw);
    float arah_kamera_y = sin(sudut_kamera_pitch);
    float arah_kamera_z = -cos(sudut_kamera_yaw);

    gluLookAt(posisi_kamera_x, posisi_kamera_y, posisi_kamera_z,
               posisi_kamera_x + arah_kamera_x, posisi_kamera_y +
    arah_kamera_y, posisi_kamera_z + arah_kamera_z,
               0.0, 10.0, 0.0);
}

void gerakanMouse(int x, int y) {
    if (mouse_aktif) {
        float sensitivitas = 0.003f;

        sudut_kamera_yaw += (x - posisi_mouse_terakhir_x) *
    sensitivitas;
        sudut_kamera_pitch += (y - posisi_mouse_terakhir_y) *
    sensitivitas;

        if (sudut_kamera_pitch > 1.5f) sudut_kamera_pitch = 1.5f;
        if (sudut_kamera_pitch < -1.5f) sudut_kamera_pitch = -
    1.5f;

        posisi_mouse_terakhir_x = x;
        posisi_mouse_terakhir_y = y;

        glutPostRedisplay();
    }
}

void klikMouse(int tombol, int state, int x, int y) {
    if (tombol == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
        mouse_aktif = true;
        posisi_mouse_terakhir_x = x;
        posisi_mouse_terakhir_y = y;
    } else if (tombol == GLUT_LEFT_BUTTON && state == GLUT_UP) {
        mouse_aktif = false;
    }
}

void gambarSumbu() {

```



```

glBegin(GL_LINES);
// Sumbu X
glColor3f(1.0, 0.0, 0.0);
glVertex3f(-15.0, 0.0, 0.0);
glVertex3f(15.0, 0.0, 0.0);
// Sumbu Y
glColor3f(0.0, 1.0, 0.0);
glVertex3f(0.0, -2.0, 0.0);
glVertex3f(0.0, 10.0, 0.0);
// Sumbu Z
glColor3f(0.0, 0.0, 1.0);
glVertex3f(0.0, 0.0, -15.0);
glVertex3f(0.0, 0.0, 15.0);
glEnd();
}

void keyboard(unsigned char key, int x, int y) {
    float kecepatan_gerak = 0.5f;
    float arah_x = sin(sudut_kamera_yaw);
    float arah_z = -cos(sudut_kamera_yaw);
    switch (key) {
        case 'w':
            posisi_kamera_x += arah_x * kecepatan_gerak;
            posisi_kamera_z += arah_z * kecepatan_gerak;
            break;
        case 's':
            posisi_kamera_x -= arah_x * kecepatan_gerak;
            posisi_kamera_z -= arah_z * kecepatan_gerak;
            break;
        case '+':
            skala_makanan += 0.1f;
            break;
        case '-':
            if (skala_makanan > 0.2f) skala_makanan -= 0.1f;
            break;
        case 'c':
            gambar_sumbu_kartesian = !gambar_sumbu_kartesian;
            break;
        case 27: // ESC
            exit(0);
    }

    batasiPosisiKamera();
    glutPostRedisplay();
}

void animasiApi(int value) {
    if (api_membesar) {
        ukuran_api += 0.05f;
        if (ukuran_api >= 0.8f) api_membesar = false;
    } else {
        ukuran_api -= 0.05f;
        if (ukuran_api <= 0.5f) api_membesar = true;
    }
    glutPostRedisplay();
    glutTimerFunc(100, animasiApi, 0);
}

```

```

void gambarKarpet(float x, float z, float panjang, float lebar) {
    glPushMatrix();
    glTranslatef(x, -2.0f, z);

    glColor3f(0.6, 0.0, 0.0);
    glRotatef(90.0f, 0.0f, 1.0f, 0.0f);
    glScalef(panjang, 0.05f, lebar);
    glutSolidCube(1.0);
    glPopMatrix();
}
// rendering
void tampilan() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    aturKamera();
    gambarRuangan();
    if (gambar_sumbu_kartesian) {
        gambarSumbu();
    }
    // Meja kanan
    gambarMeja(10.0f, -1.5f, -5.0f, 15.0f, 2.5f, 0.5f, 90.0f);
    gambarMakananDiMeja(10.0f, -1.0f, -5.0f);
    gambarKarpet(10.0f, -5.0f, 22.0f, 10.5f);
    // Meja kiri
    gambarMeja(-10.0f, -1.5f, -5.0f, 15.0f, 2.5f, 0.5f, 90.0f);
    gambarMakananDiMeja(-10.0f, -1.0f, -5.0f);
    gambarKarpet(-10.0f, -5.0f, 22.0f, 10.5f);
    // Meja belakang
    gambarMeja(0.0f, -1.5f, -13.0f, 3.0f, 3.0f, 0.5f, 0.0f);
    gambarKompas(0.0f, -1.0f, -13.0f);

    glutSwapBuffers();
}

void inisialisasi() {
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glEnable(GL_COLOR_MATERIAL);
    glClearColor(0.2, 0.2, 0.2, 1.0);

    // posisi cahaya
    GLfloat posisi_cahaya[] = {0.0f, 0.0f, -13.0f, 1.0f};
    GLfloat cahaya_diffuse[] = {1.0f, 0.5f, 0.0f, 1.0f};
    GLfloat cahaya_ambient[] = {0.2f, 0.1f, 0.0f, 1.0f};
    GLfloat cahaya_specular[] = {1.0f, 1.0f, 1.0f, 1.0f};

    glLightfv(GL_LIGHT0, GL_POSITION, posisi_cahaya);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, cahaya_diffuse);
    glLightfv(GL_LIGHT0, GL_AMBIENT, cahaya_ambient);
    glLightfv(GL_LIGHT0, GL_SPECULAR, cahaya_specular);
}

// mengatur proyeksi
void reshape(int w, int h) {
    glViewport(0, 0, w, h);
}

```

```

glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluPerspective(45.0, (float)w / h, 1.0, 100.0);
glMatrixMode(GL_MODELVIEW);
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(800, 600);
    glutCreateWindow("Restoran Kelompok - 10");

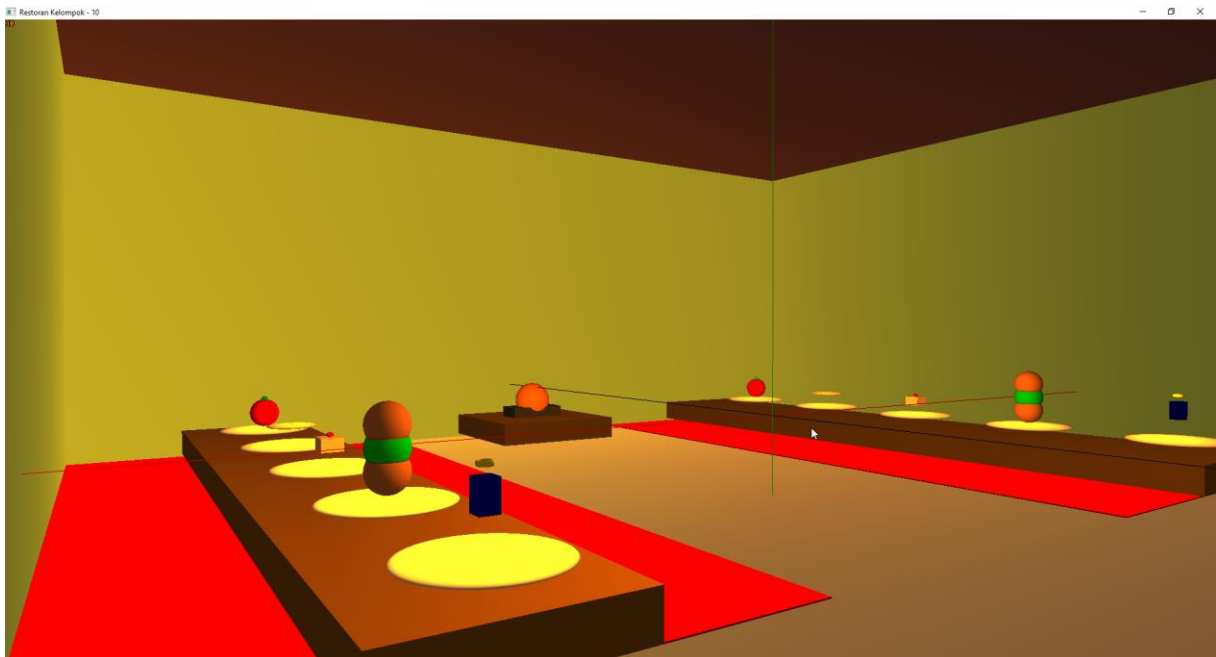
    glutDisplayFunc(tampilan);
    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
    glutMouseFunc(klikMouse);
    glutMotionFunc(gerakanMouse);

    inisialisasi();
    animasiApi(0);

    glutMainLoop();
    return 0;
}

```

3.2 Output



Gambar 9 Output Program Restoran Makanan

3.3 Penjelasan

mengenai source code ini berfokus pada pembuatan simulasi 3D dari sebuah restoran dengan menggunakan OpenGL. Dalam program ini, terdapat beberapa elemen utama yang digambar dan dianimasikan, seperti meja, makanan, kompor dengan api, serta ruangan restoran.

1. Variabel Kamera dan Kontrol:

- Kamera diatur dengan variabel posisi (`posisi_kamera_x`, `posisi_kamera_y`, `posisi_kamera_z`) dan sudut pandang (`sudut_kamera_yaw`, `sudut_kamera_pitch`). Fungsi `aturKamera()` digunakan untuk mengatur posisi dan orientasi kamera berdasarkan input dari mouse dan keyboard.
 - Fungsi `gerakanMouse()` dan `klikMouse()` menangani interaksi mouse untuk mengubah sudut pandang kamera, memungkinkan pengguna untuk melihat restoran dari berbagai sudut.
2. Pembuatan Objek:
- Fungsi seperti `gambarSilinder()`, `gambarMeja()`, dan `gambarMakananDiMeja()` digunakan untuk menggambar berbagai objek di dalam restoran. Misalnya, meja digambar menggunakan objek kubus dengan warna kayu, sementara makanan seperti pizza, burger, dan minuman digambar menggunakan objek seperti bola dan kerucut dengan pewarnaan yang sesuai.
 - Api di kompor dianimasikan menggunakan fungsi `animasiApi()`, yang membuat api membesar dan mengecil secara berulang dengan cara mengubah ukuran objek bola yang menggambarkan api.
3. Animasi dan Interaktivitas:
- Fungsi `animasiApi()` mengatur animasi api di kompor dengan mengubah ukuran api secara berkala. Fungsi ini dipanggil secara periodik menggunakan `glutTimerFunc()`.
 - Pengguna dapat berinteraksi dengan objek dalam scene melalui input keyboard dan mouse. Misalnya, tombol 'w' dan 's' digunakan untuk menggerakkan kamera maju atau mundur, tombol '+' dan '-' digunakan untuk mengubah skala makanan, dan tombol 'c' untuk mengaktifkan atau menonaktifkan tampilan sumbu kartesius.
4. Pencahayaan dan Tampilan:
- Program ini juga mengaktifkan pencahayaan dengan menggunakan `glEnable(GL_LIGHTING)` dan mengatur properti cahaya seperti posisi dan warna dengan fungsi `glLightfv()`. Hal ini memberikan efek pencahayaan yang realistis pada objek di dalam ruangan.

Output yang Dihasilkan: Output dari program ini adalah sebuah simulasi 3D dari sebuah restoran dengan meja-meja yang dihiasi makanan dan minuman. Pengguna dapat melihat restoran dari berbagai sudut menggunakan mouse dan keyboard. Terdapat animasi api di kompor yang berubah ukuran, serta sumbu kartesius yang dapat diaktifkan atau dinonaktifkan sesuai dengan pilihan pengguna.

Alasan Mengapa Output Terjadi:

- Output ini dihasilkan karena penggunaan berbagai fungsi OpenGL untuk menggambar objek 3D dan memberikan interaktivitas. Fungsi-fungsi seperti `glutSolidCube()`, `glutSolidSphere()`, dan `glutSolidCone()` digunakan untuk menggambar objek-objek yang membentuk elemen-elemen restoran, sementara pencahayaan dan animasi api memberikan efek visual yang lebih dinamis. Interaksi dengan keyboard dan mouse memungkinkan pengguna untuk berinteraksi dengan scene, memberikan pengalaman yang lebih imersif.

BAB IV

4.1. Kesimpulan

Praktikum ini bertujuan untuk memberikan pemahaman yang lebih dalam tentang OpenGL dan penerapannya dalam pembuatan aplikasi grafik komputer, khususnya dalam konteks program restoran makanan.

Setelah melalui serangkaian kegiatan praktikum, kami dapat menyimpulkan bahwa tujuan praktikum ini telah tercapai. Kami berhasil memahami konsep dasar OpenGL, mulai dari pengaturan konfigurasi di Dev C++ atau VSCode, hingga cara kerja OpenGL dalam rendering grafik. Selain itu, kami juga berhasil mengimplementasikan fungsi-fungsi OpenGL dalam program yang kami buat.

Melalui praktikum ini, kami belajar banyak hal, termasuk bagaimana cara membuat objek 2D dan 3D, menerapkan transformasi, serta mengelola tampilan grafis dengan baik. Kami juga mendapatkan pengalaman berharga dalam pemrograman grafik, yang sangat penting untuk pengembangan aplikasi di industri teknologi saat ini.

Secara keseluruhan, praktikum ini tidak hanya meningkatkan keterampilan teknis kami, tetapi juga memperdalam pemahaman kami tentang interaksi antara perangkat keras dan perangkat lunak dalam menghasilkan grafik yang kompleks. Kami berharap pengalaman ini dapat menjadi dasar yang kuat untuk eksplorasi lebih lanjut dalam bidang grafik komputer di masa depan.

DAFTAR PUSTAKA

- [1] Gunawan, A., & Falani, A. Z. (2022). IMPLEMENTASI OPENGL (OPEN GRAPHICS LIBRARY) UNTUK PEMBUATAN BANGUN DATAR SEBAGAI MEDIA PEMBELAJARAN. In *Jurnal Pendidikan Teknologi Informasi (JUKANTI)* (Issue 5).
- [2] Hadi, S., Matematika, F., Ilmu, D., & Alam, P. (2014). *MODUL GRAFIKA KOMPUTER Disusun oleh: Program Studi Teknik Informatika*.
- [3] Suhardiman, D., Kaunang, S. T. G., Sengkey, R., & Rumagit, A. M. (n.d.). *Pembuatan Simulasi Pergerakan Objek 3D (Tiga Dimensi) Menggunakan OpenGL*.