

LAPORAN PRATIUM GRAFIK KOMPUTER

Diajukan untuk memenuhi Tugas mata kuliah Pratikum Grafik Komputer

TUGAS BESAR PRATIUM GRAFIK KOMPUTER MEMBUAT SEBUAH BOARD GAMES

Dosen Pengampu : Sri Rahayu, M.Kom

Instruktur Pratikum : Arul Budi Kalimat, S.Kom



Disusun oleh

Kelompok 4:

Shaila Najwa A
2306125

Raina Abdina
2306125

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 Grafika Komputer ini. Laporan ini dibuat sebagai salah satu tugas dari mata kuliah Grafika 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, 11 Januari 2024

Kelompok 4

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 Membuat Boards Game pada OpenGL	6
BAB III HASIL.....	7
3.1 Source Code.....	7
3.2 Output	13
3.3 Penjelasan.....	15
BAB IV	19
4.1. Kesimpulan.....	19
DAFTAR PUSTAKA	20

DAFTAR GAMBAR

Gambar 1.Membuka Aplikasi Dev-C++	3
Gambar 2.Membuat Project pada Dev C++	4
Gambar 3.Mengatur Project pada Dev C++	4
Gambar 4.Memasukan Parameters untuk Project pada Dev C++	4
Gambar 5.Menyusun Kode untuk Membuat Project Pada Dev C++	5
Gambar 6.Tampilan Awal program yaitu ada Meja dan Dadu dan ada 3 Menu	14
Gambar 7.Menghentikan Rotasi Pada Dadu Melalui GUI	14
Gambar 8.Menampilkan Garis Kartesius Melalui GUI.....	14
Gambar 9.Dadu Bergerak ke Kanan.....	15
Gambar 10.Dadu Bergerak ke Kiri.....	15
Gambar 11.Memperkecil Dadu	15
Gambar 12.Memperbesar Dadu	15

BAB I

PENDAHULUAN

1.1 Latar Belakang

Praktikum ini bertujuan untuk memahami grafika komputer, yaitu bidang yang mempelajari cara menghasilkan gambar melalui bantuan komputer. Dalam grafika komputer, berbagai teknik menggambar menjadi fokus utama pembelajaran. Teknologi ini telah melahirkan beragam perangkat lunak desain grafis yang kini berkembang dengan sangat canggih.[1]

Dalam praktikum ini, bahasa C++ digunakan untuk mendukung pemahaman tentang grafika komputer, yaitu bidang yang berfokus pada cara menghasilkan dan memanipulasi gambar dengan bantuan komputer. Pemanfaatan C++ dalam grafika komputer memberikan dasar yang kuat dalam mempelajari teknik-teknik menggambar dan penerapannya dalam pengembangan perangkat lunak desain grafis yang kini berkembang pesat.[2]

Grafika komputer menghasilkan software-software desain grafis yang saat ini sudah sangat canggih sehingga menghasilkan software dengan GUI (Graphics User Interface) yang memudahkan dan menyenangkan. Grafika Komputer terdapat perantara aplikasi dan display hardware (Graphics System) yaitu OpenGL. [3]

Grafika komputer didefinisikan sebagai proses penggabungan dan kombinasi dari elemen-elemen seperti titik, garis, poligon, dan bentuk lainnya untuk menghasilkan objek dalam bentuk dua dimensi (2D) atau tiga dimensi (3D). Grafika komputer 2D melibatkan pembuatan objek gambar yang berbasis perspektif dua titik, seperti teks, bangun 2D seperti segitiga, persegi, lingkaran, dan bentuk serupa lainnya. Sementara itu, grafika komputer 3D menggunakan tiga titik perspektif melalui pendekatan matematis untuk menampilkan atau memvisualisasikan objek secara keseluruhan, sehingga menghasilkan gambar yang tampak lebih realistis. Dalam pengaplikasiannya, transformasi seperti translasi, penskalaan, dan rotasi digunakan untuk memodifikasi atau memanipulasi objek tersebut. Transformasi ini tidak hanya mempermudah pengaturan komposisi dan perspektif, tetapi juga memungkinkan pergerakan dinamis pada objek 3D dan perubahan sudut pandang sesuai kebutuhan pengguna, sehingga menciptakan visualisasi yang lebih realistis dan interaktif.[4]

Antarmuka pengguna grafis (Graphical User Interface-GUI) merupakan tampilan antarmuka computer yang memungkinkan pengguna berinteraksi dengan perangkat lunak atau sistem komputer menggunakan elemen grafis seperti ikon, tombol, jendela, dan menu. GUI bertujuan untuk membuat pengalaman pengguna lebih intuitif dan mudah digunakan, daripada berinteraksi dengan perintah teks atau baris perintah (Xie et al., 2022). [5]

1.2 Rumusan Masalah

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 Board Games 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 Board Games dalam OpenGL

BAB II

TINJAUAN PUSTAKA

2.1 OpenGL

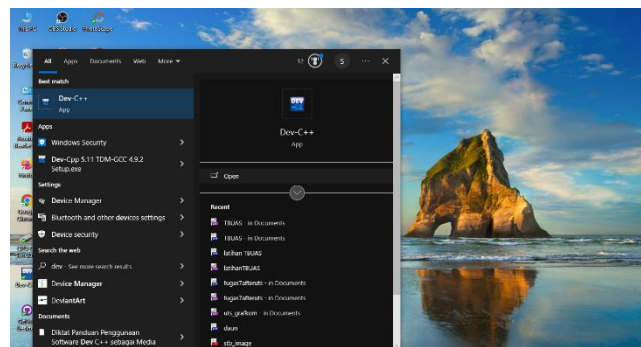
OpenGL adalah library grafis standar yang digunakan dalam pemrograman grafis. Sebagai library open-source, OpenGL mendukung berbagai platform dan bahasa pemrograman, termasuk Windows, Linux, dan Unix, sehingga menjadikannya pilihan populer di dunia grafis, selain DirectX.[1]

OpenGL (Open Graphics Library) adalah salah satu API grafis yang digunakan untuk membangun desain 2D dan 3D. OpenGL bekerja dengan integrasi bahasa pemrograman seperti C atau C++, karena tidak dapat berfungsi secara mandiri. Sebagai library open-source, OpenGL mendukung berbagai platform dan bahasa pemrograman, menjadikannya pilihan populer di dunia grafis.[2]

2.2 Konfigurasi OpenGL pada Dev C++ atau VSCode

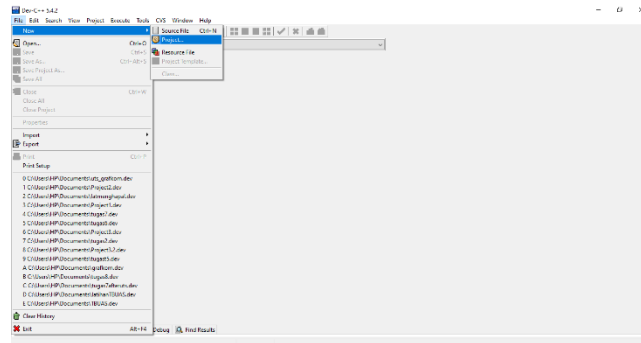
Berikut langkah-langkah untuk konfigurasi OpenGL pada Dev C++:

1. Pastikan aplikasi seperti Dev-C++, atau Visual Studio sudah terpasang dengan baik sesuai dengan sistem operasi yang digunakan. Jika aplikasi tersebut belum terinstal, unduh terlebih dahulu dari situs resmi masing-masing. Setelah terinstal, cari ikon aplikasi pada desktop atau melalui menu pencarian, kemudian klik dua kali untuk membuka aplikasi tersebut, Seperti pada gambar dibawah ini:



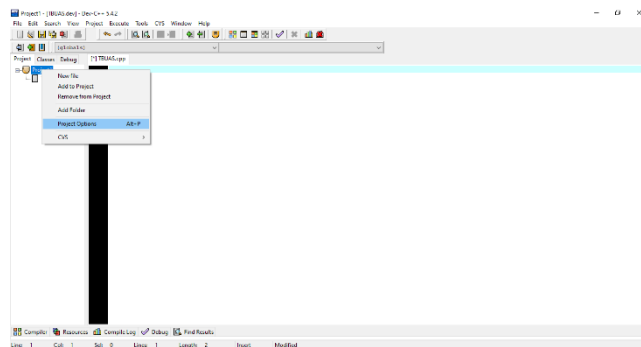
Gambar 1. Membuka Aplikasi Dev-C++

2. Setelah membuka aplikasi C++, pilih menu 'File', kemudian pilih opsi 'New' dan pilih 'Project', Seperti pada gambar dibawah ini:



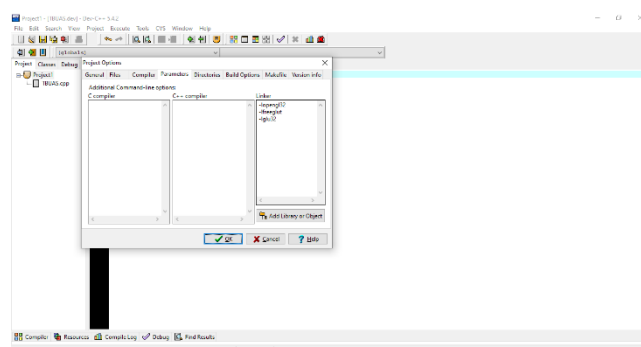
Gambar 2.Membuat Project pada Dev C++

3. Apabila telah membuat sebuah project, masukkan nama project yang akan dibuat, kemudian klik tombol 'OK'. Setelah itu, pilih opsi 'Project', dan lanjutkan dengan mengklik 'Project Option', Seperti pada gambar dibawah ini:



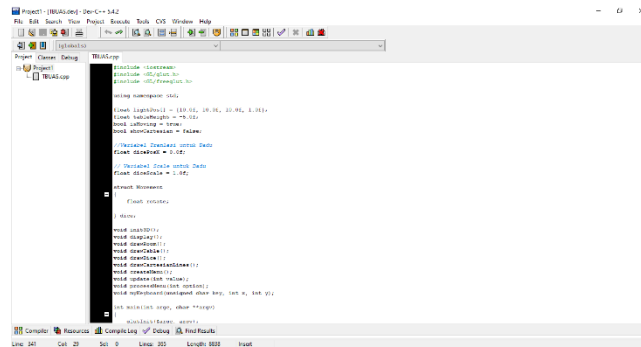
Gambar 3.Mengatur Project pada Dev C++

4. Setelah memasuki opsi 'Project Options', pilihlah opsi 'Parameters', kemudian masukkan linker yang sesuai untuk membuat proyek OpenGL, Seperti pada gambar dibawah ini:



Gambar 4.Memasukan Parameters untuk Project pada Dev C++

5. Selanjutnya, langkah yang perlu dilakukan adalah menyusun kode untuk project yang akan dibuat, Seperti pada gambar dibawah ini:



Gambar 5.Menyusun Kode untuk Membuat Project Pada Dev C++

Untuk menampilkan output dari program yang telah dibuat, Anda dapat mengklik opsi Compile and Run. Sebelum itu, pastikan untuk menyimpan file dengan memberikan nama file yang sesuai. Setelah proses kompilasi selesai, output akan ditampilkan jika kode yang ditulis tidak mengandung kesalahan atau error.

2.3 Cara Kerja OpenGL

OpenGL bekerja sebagai API grafis yang menghubungkan perangkat lunak aplikasi dengan perangkat keras grafis melalui pipeline rendering. Proses dimulai dengan inisialisasi lingkungan kerja, seperti mengatur window dan konteks OpenGL, dilanjutkan dengan definisi geometri menggunakan primitif seperti titik, garis, dan poligon dalam koordinat 3D. Selanjutnya, transformasi seperti translasi, rotasi, dan penskalaan diterapkan untuk mengatur posisi dan perspektif objek. Shader digunakan untuk mengontrol cahaya, warna, dan tekstur sebelum data objek diproses melalui pipeline rendering untuk ditampilkan di layar melalui buffer tampilan. OpenGL juga mendukung interaksi real-time, memungkinkan respons terhadap input pengguna untuk menciptakan pengalaman visual yang dinamis dan interaktif.

2.4 Membuat Boards Game pada OpenGL

Pembuatan Board Games menggunakan OpenGL dengan menggunakan bahasa pemrograman C++ dan pustaka GLUT untuk pengelolaan grafik. Pembuatan dimulai dengan inisialisasi OpenGL, yang mencakup pengaturan jendela grafik, pencahayaan, dan kamera. Komponen utama dari Board Game terdiri atas meja dan dadu. Meja dibentuk menggunakan fungsi `glutSolidCube` untuk bagian permukaan dan kaki, sementara dadu dibuat menggunakan fungsi yang sama dengan tambahan titik-titik dadu menggunakan `glutSolidSphere`. Interaksi pengguna meliputi translasi dadu ke kanan dan kiri menggunakan tombol `s` dan `r`, perubahan ukuran dadu dengan tombol `n` untuk memperkecil dan `a` untuk memperbesar, serta kontrol melalui menu GUI untuk memulai atau menghentikan rotasi dadu dan menampilkan atau menyembunyikan garis kartesius. Ketika aplikasi dijalankan, meja dan dadu yang berputar otomatis akan ditampilkan, dengan pengguna dapat mengontrolnya secara interaktif sesuai fitur yang disediakan.

BAB III

HASIL

3.1 Source Code

Source code Boards Game :

```
#include <iostream>
#include <GL/glut.h>
#include <GL/freeglut.h>

using namespace std;

float lightPos[] = {10.0f, 10.0f, 10.0f, 1.0f};
float tableHeight = -5.0f;
bool isMoving = true;
bool showCartesian = false;

//Variabel Tranlasi untuk Dadu
float dicePosX = 0.0f;

// Variabel Scale untuk Dadu
float diceScale = 1.0f;

struct Movement
{
    float rotate;
} dice;

void init3D();
void display();
void drawRoom();
void drawTable();
void drawDice();
void drawCartesianLines();
void createMenu();
void update(int value);
void processMenu(int option);
void myKeyboard(unsigned char key, int x, int y);

int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(800, 600);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("TUGAS BESAR PRAKTIKUM GRAFKOM KELOMPOK4");

    init3D();
    glutDisplayFunc(display);
    glutKeyboardFunc(myKeyboard);
    createMenu();
    glutTimerFunc(1000 / 80, update, 0);
    glutMainLoop();
    return 0;
}
```

```

}

void init3D()
{
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_COLOR_MATERIAL);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);

    GLfloat ambientLight[] = {0.3f, 0.3f, 0.3f, 1.0f};
    GLfloat diffuseLight[] = {1.0f, 1.0f, 1.0f, 1.0f};
    glLightfv(GL_LIGHT0, GL_AMBIENT, ambientLight);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuseLight);
    glLightfv(GL_LIGHT0, GL_POSITION, lightPos);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(60.0, 4.0 / 3.0, 1.0, 100.0);
    gluLookAt(40.0, 30.0, 60.0, 0.0, 0.0, 0.0, 1.0, 0.0);
    glMatrixMode(GL_MODELVIEW);

    glClearColor(0.9, 0.9, 0.9, 1.0);
}

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

    glLightfv(GL_LIGHT0, GL_POSITION, lightPos);

    drawRoom(); // Menggambar Background Ruangan
    drawTable(); // Menggambar Meja

    if (showCartesian)
    {
        drawCartesianLines(); // Garis kartesius
    }

    drawDice(); // Menggambar Dadu

    glutSwapBuffers();
}

void drawRoom()
{
    // Lantai pada ruangan
    glPushMatrix();
    glColor3f(0.5f, 0.5f, 0.5f);
    glBegin(GL_QUADS);
    glVertex3f(-50.0f, tableHeight - 10, -50.0f);
    glVertex3f(50.0f, tableHeight - 10, -50.0f);
    glVertex3f(50.0f, tableHeight - 10, 50.0f);
    glVertex3f(-50.0f, tableHeight - 10, 50.0f);
    glEnd();
    glPopMatrix();
}

```

```

void drawTable()
{
    glPushMatrix();
    glColor3ub(139, 69, 19);
    glTranslated(0.0, tableHeight, 0.0);

    // Bagian Permukaan (Atas) Meja
    glPushMatrix();
    glScalef(30.0, 1.0, 20.0); // Ukuran Meja
    glutSolidCube(1.0);
    glPopMatrix();

    // Bagian Kaki (Bawah) Meja
    float legOffsetX = 14.5f;
    float legOffsetZ = 9.5f;

    glPushMatrix();
    glTranslated(legOffsetX, -5.0, legOffsetZ);
    glScalef(1.0, 10.0, 1.0);
    glutSolidCube(1.0);
    glPopMatrix();

    glPushMatrix();
    glTranslated(-legOffsetX, -5.0, legOffsetZ);
    glScalef(1.0, 10.0, 1.0);
    glutSolidCube(1.0);
    glPopMatrix();

    glPushMatrix();
    glTranslated(legOffsetX, -5.0, -legOffsetZ);
    glScalef(1.0, 10.0, 1.0);
    glutSolidCube(1.0);
    glPopMatrix();

    glPushMatrix();
    glTranslated(-legOffsetX, -5.0, -legOffsetZ);
    glScalef(1.0, 10.0, 1.0);
    glutSolidCube(1.0);
    glPopMatrix();

    glPopMatrix();
}

void drawDice()
{
    glPushMatrix();
    glTranslated(dicePosX, tableHeight + 3.0, 0.0); //untuk
Tranlasi pada Dadu
    glRotated(dice.rotate, 1.0, 1.0, 0.0);
    glScalef(diceScale, diceScale, diceScale);
    glColor3ub(255, 255, 255);
    glutSolidCube(4.0);

    glColor3ub(0, 0, 0);

    // Titik Satu (Bagian Depan Dadu)

```

```

glPushMatrix();
glTranslated(0.0, 0.0, 2.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

// Titik Dua (Bagian Belakang Dadu)
glPushMatrix();
glTranslated(-1.0, -1.0, -2.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

glPushMatrix();
glTranslated(1.0, 1.0, -2.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

// Titik Tiga (Bagian Kanan Dadu)
glPushMatrix();
glTranslated(2.0, -1.0, -1.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

glPushMatrix();
glTranslated(2.0, 0.0, 0.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

glPushMatrix();
glTranslated(2.0, 1.0, 1.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

// Titik Empat (Bagian Kiri Dadu)
glPushMatrix();
glTranslated(-2.0, 1.0, 1.0);
glutSolidSphere(0.3, 40, 40);
glPopMatrix();

glPushMatrix();
glTranslated(-2.0, -1.0, 1.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

glPushMatrix();
glTranslated(-2.0, 1.0, -1.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

glPushMatrix();
glTranslated(-2.0, -1.0, -1.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

// Titik Lima (Bagian Atas Dadu)
glPushMatrix();
glTranslated(-1.0, 2.0, 1.0);
glutSolidSphere(0.3, 10, 10);

```

```

glPopMatrix();

glPushMatrix();
glTranslated(1.0, 2.0, -1.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

glPushMatrix();
glTranslated(0.0, 2.0, 0.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

glPushMatrix();
glTranslated(-1.0, 2.0, -1.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

glPushMatrix();
glTranslated(1.0, 2.0, 1.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

// Titik Enam (Bagian Bawah Dadu)
glPushMatrix();
glTranslated(-1.0, -2.0, 1.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

glPushMatrix();
glTranslated(1.0, -2.0, -1.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

glPushMatrix();
glTranslated(0.0, -2.0, 1.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

glPushMatrix();
glTranslated(-1.0, -2.0, -1.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

glPushMatrix();
glTranslated(1.0, -2.0, 1.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

glPushMatrix();
glTranslated(0.0, -2.0, -1.0);
glutSolidSphere(0.3, 10, 10);
glPopMatrix();

glPopMatrix();
}

void drawCartesianLines()

```

```

{
    glPushMatrix();
    glLineWidth(2.0);
    glColor3f(0.8f, 0.6f, 1.0f);
    glBegin(GL_LINES);
    glVertex3f(-50.0f, tableHeight, 0.0f);
    glVertex3f(50.0f, tableHeight, 0.0f);
    glEnd();

    glColor3f(0.8f, 0.6f, 1.0f);
    glBegin(GL_LINES);
    glVertex3f(0.0f, tableHeight - 50.0f, 0.0f);
    glVertex3f(0.0f, tableHeight + 50.0f, 0.0f);
    glEnd();

    glColor3f(0.8f, 0.6f, 1.0f);
    glBegin(GL_LINES);
    glVertex3f(0.0f, tableHeight, -50.0f);
    glVertex3f(0.0f, tableHeight, 50.0f);
    glEnd();
    glPopMatrix();
}

void createMenu()
{
    glutCreateMenu(processMenu);
    glutAddMenuEntry("Start/Stop Rotation", 1);
    glutAddMenuEntry("Tampilkan/Sembunyikan Garis Kartesius", 2);
    glutAddMenuEntry("Exit", 3);
    glutAttachMenu(GLUT_RIGHT_BUTTON);
}

void processMenu(int option)
{
    switch (option)
    {
        case 1:
            isMoving = !isMoving;
            break;
        case 2:
            showCartesian = !showCartesian;
            break;
        case 3:
            exit(0);
    }
    glutPostRedisplay();
}

void update(int value)
{
    if (isMoving)
    {
        dice.rotate += 1.0f;
    }
    glutPostRedisplay();
    glutTimerFunc(1000 / 80, update, 0);
}

```



```

void myKeyboard(unsigned char key, int x, int y)
{
    //Batas minumum dan maksimum posisi dadu pada translasi
    const float Min_X = -12.0;
    const float Max_X = 13.0;
    const float Min_Scale = 0.5;
    const float Max_Scale = 2.0;

    switch (key)
    {
        case 's': // Tombol 's' untuk bergerak ke Kanan
            if (!isMoving && dicePosX + 0.1 <= Max_X) // Bergerak saat
Rotasi Berhenti
                dicePosX += 0.1;
            break;

        case 'r': // Tombol 'r' untuk bergerak ke Kiri
            if (!isMoving && dicePosX - 0.1 >= Min_X) // Bergerak saat
Rotasi Berhenti
                dicePosX -= 0.1;
            break;

        case 'n': // Tombol 'n' untuk Memperkecil Dadu
            if (!isMoving && diceScale - 0.1 >= Min_Scale) // Bergerak
saat Rotasi Berhenti
                diceScale -= 0.1;
            break;

        case 'a': // Tombol 'a' untuk Memperbesar Dadu
            if (!isMoving && diceScale + 0.1 <= Max_Scale) // Bergerak
saat Rotasi Berhenti
                diceScale += 0.1;
            break;

        default:
            break;
    }

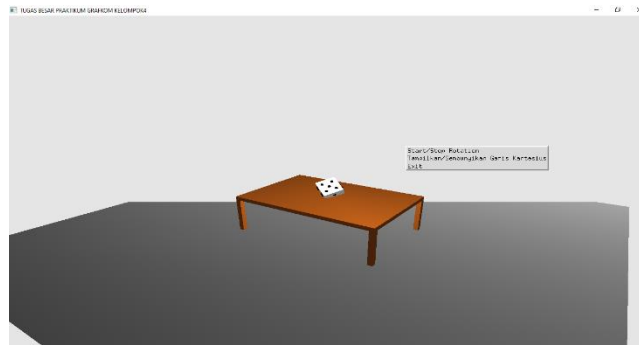
    glutPostRedisplay();
}

```

3.2 Output

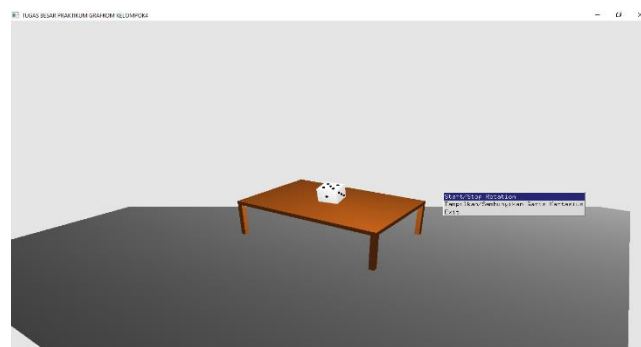
Gambar screenshot hasil source code Bords Game:

1. Setelah program dijalankan, tampilan awal akan menampilkan sebuah meja dan dadu yang sedang berotasi. Terdapat tiga opsi menu yang tersedia, yaitu: pertama, Start/Stop Rotation untuk memulai atau menghentikan rotasi dadu; kedua, Tampilkan/Sembunyikan Garis Kartesius untuk menampilkan atau menyembunyikan garis kartesius; dan ketiga, Exit untuk keluar dari program:



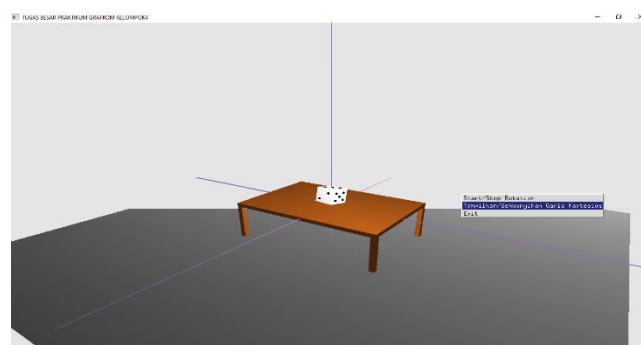
Gambar 6. Tampilan Awal program yaitu ada Meja dan Dadu dan ada 3 Menu

2. Pada tampilan awal, dadu akan berputar secara otomatis. Untuk menghentikan rotasi tersebut, tersedia opsi 'Stop Rotation' yang dapat diakses melalui antarmuka grafis (GUI).



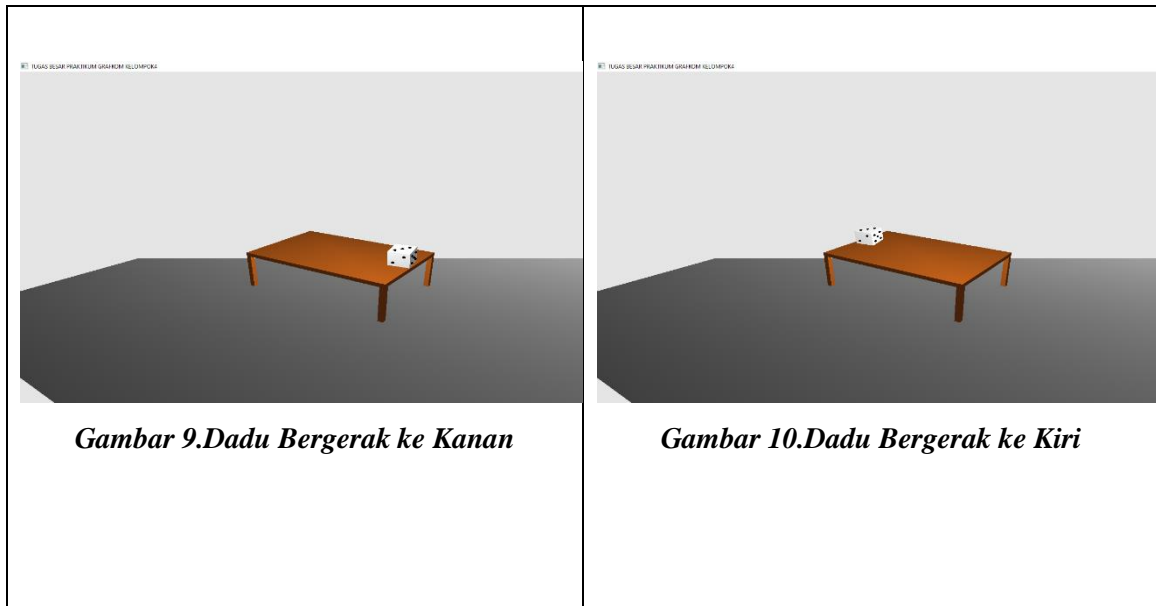
Gambar 7. Menghentikan Rotasi Pada Dadu Melalui GUI

3. Untuk menampilkan atau menyembunyikan garis Kartesius, dengan memilih opsi yang tersedia melalui antarmuka grafis (GUI).

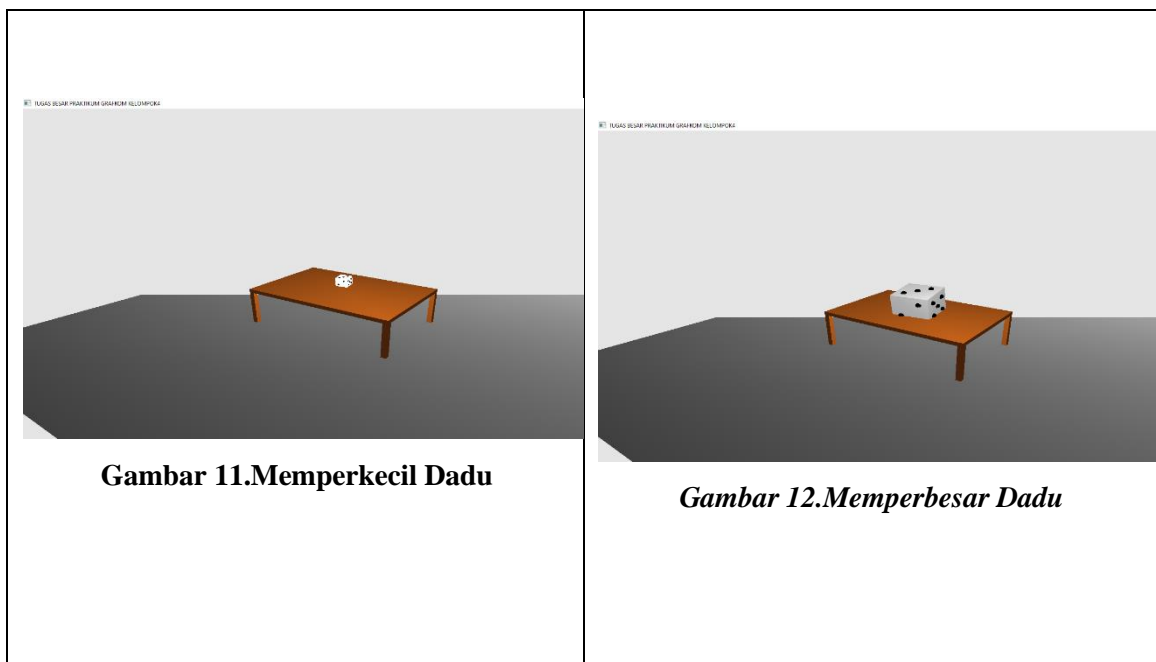


Gambar 8. Menampilkan Garis Kartesius Melalui GUI

4. Untuk melakukan translasi pada objek dadu, dapat menggunakan fungsi keyboard. Tekan tombol 'S' untuk menggerakkan objek ke arah kanan, dan tekan tombol 'R' untuk menggerakkan objek ke arah kiri.



5. Untuk melakukan perubahan skala pada objek dadu, dapat digunakan fungsi keyboard dengan menekan tombol 'n' untuk memperkecil ukuran dadu, dan tombol 'a' untuk memperbesar ukuran dadu.



3.3 Penjelasan

Kode ini adalah program OpenGL menggunakan GLUT (OpenGL Utility Toolkit) untuk membuat simulasi grafis 3D. Program ini menggambar sebuah ruangan, meja, dan sebuah dadu. Ada kontrol interaktif untuk mengatur rotasi dadu, translasi dadu pada sumbu X, serta mengubah ukuran dadu.

✓ Struktur Program

-Inisialisasi (init3D):

- Mengaktifkan fitur OpenGL seperti tes kedalaman, pencahayaan, dan warna.
- Menentukan parameter pencahayaan dan sudut pandang kamera.

-Fungsi (display):

- Membersihkan buffer.
- Mengatur posisi pencahayaan.
- Memanggil fungsi menggambar ruangan, meja, dan dadu.
- Menampilkan buffer ke layar menggunakan glutSwapBuffers().

-Objek Grafis

- Ruangan:

Lantai (permukaan horizontal) digambar menggunakan GL_QUADS.

- Meja:

Dibangun dari balok persegi panjang untuk bagian atas meja dan kaki meja.

- Dadu:

Digambar sebagai kubus dengan ukuran 4 unit.

Ditambahkan titik-titik (seperti pada dadu biasa) menggunakan glutSolidSphere pada berbagai sisi kubus.

-Transformasi Pada Objek

- Rotasi Dadu:

Dikontrol menggunakan menu konteks (klik kanan, opsi "Start/Stop Rotation").

Jika isMoving aktif, rotasi dadu bertambah pada setiap panggilan fungsi timer.

- Translasi Dadu:

Tombol 's' untuk memindahkan dadu ke kanan.

Tombol 'r' untuk memindahkan dadu ke kiri.

Translasi dadu hanya bekerja jika rotasi berhenti (isMoving == false).

- Skalasi Dadu:

Tombol 'n' untuk memperkecil dadu.

Tombol 'a' untuk memperbesar dadu.

Batas minimum dan maksimum ukuran dadu adalah 0.5 dan 2.0.

-Garis Kartesius:

- Opsi "Tampilkan/Sembunyikan Garis Kartesius" di menu untuk menampilkan atau menyembunyikan garis bantu.

-Menu dengan GUI

Menu konteks (createMenu) dengan tiga opsi:

- Menghentikan/menjalankan rotasi.
- Menampilkan/menyembunyikan garis kartesius.
- Keluar dari program.

-Timer:

- Fungsi update dipanggil setiap 1/80 detik untuk memperbarui rotasi dadu jika isMoving aktif.

✓ Output yang Dihasilkan pada Program:

Visualisasi:

- Sebuah ruangan dengan meja kayu berwarna coklat.
- Sebuah dadu putih dengan titik hitam di atas meja.
- Jika garis kartesius diaktifkan, tiga garis bantu (merah, hijau, biru) pada sumbu X, Y, dan Z muncul.

• Interaksi:

Dadu berotasi jika rotasi aktif.

- Dadu dapat digeser ke kiri/kanan menggunakan tombol 'r' dan 's'.
- Ukuran dadu dapat diubah dengan tombol 'n' (kecil) dan 'a' (besar).

Menu klik kanan menyediakan kontrol tambahan.

✓ Fungsi2 di OpenGL

- Fungsi display menentukan semua elemen grafis yang digambar pada setiap frame.
- glPushMatrix dan glPopMatrix menjaga transformasi (skala, translasi, rotasi) agar tidak memengaruhi objek lain.

✓ Fungsi Rotasi

Variabel dice.rotate terus bertambah setiap frame jika isMoving aktif, memberikan efek rotasi pada dadu.

✓ Translasi dan Skalasi:

- Translasi dadu dikontrol oleh dicePosX dan bersyarat pada status rotasi.
- Skalasi dadu dikontrol oleh diceScale dengan batasan minimum dan maksimum.

✓ Garis Kartesius:

Fungsi `drawCartesianLines` menggambar garis bantu hanya jika `showCartesian` aktif.

✓ Efek Visual:

Pencahayaan (`GL_LIGHTING`) dan warna material menghasilkan efek bayangan dan realisme pada objek.

✓ Penjelasan:

- Inisialisasi 3D dan Pencahayaan:

Memberikan ilusi dimensi ruang dan pencahayaan pada objek.

- Dadu:

Dibangun sebagai kubus dengan detail titik-titik pada setiap sisi.

- Transformasi:

Rotasi, translasi, dan skalasi memberikan kendali pengguna terhadap objek.

- Menu Konteks:

Mempermudah kontrol fungsi seperti rotasi dan garis kartesius.

- Timer:

Membuat animasi rotasi dadu dengan pembaruan periodik.

BAB IV

4.1. Kesimpulan

Pemahaman penggunaan OpenGL dalam pengembangan grafis 3D melalui pembuatan Board Games yang melibatkan objek meja dan dadu. Dalam Laporan ini, mempelajari cara konfigurasi OpenGL, penerapan transformasi seperti translasi, rotasi, dan skalasi, serta penggunaan pencahayaan dan warna untuk meningkatkan visualisasi. Interaksi dengan objek dilakukan melalui input keyboard dan menu GUI, memungkinkan kontrol terhadap rotasi dadu, perubahan skala, dan tampilan garis kartesius. Dan memberikan wawasan praktis tentang pipeline rendering OpenGL, konsep grafis 3D, dan pengembangan aplikasi interaktif.

DAFTAR PUSTAKA

- [1] A. Basuki and N. Ramadijanti, “PENGANTAR GRAFIKA KOMPUTER,” PENS-ITS.
- [2] “KATA PENGANTAR.”
- [3] 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>
- [4] * Herriyance and M. Dahria, “J-SISKO TECH Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD Penerapan Transformasi Translasi Dan Rotasi Untuk Visualisasi Objek 3d Pada Aplikasi Desktop,” *J. Teknol. Sist. Inf. dan Sist. Komput. TGD*, vol. 1, no. 2, pp. 96–105, 2018.
- [5] Y. Prihastomo and W. Winanti, “Tren Perkembangan Graphical User Interface Melalui Permohonan Desain Industri Di Indonesia,” *J. Commun. Educ.*, vol. 18, no. 1, pp. 93–100, 2024, doi: 10.58217/joce-ip.v18i1.390.