

LAPORAN PRATIUM GRAFIK KOMPUTER

Diajukan untuk memenuhi Tugas mata kuliah Pratikum Grafik Komputer

TUGAS BESAR

Dosen Pengampu : Sri Rahayu, M.Kom

Instruktur Pratikum : Arul Budi Kalimat, S.Kom



Disusun oleh

Kelompok :

Asyifa Azsma H.

2306071

Muhammad Rafi A.F

2306086

Zulfa Ghinan K.

2306092

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 Pemandangan Alam.

Kami mengucapkan terima kasih kepada dosen pengampu Sri Rahayu, S.T. M.Kom, instruktur praktikum Arul Budi Kalimat S.Kom Beserta jajaran asisten dosen yang tidak bisa di sebutkan satu persatu, 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, 7 Januari 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.....	1
BAB II TINJAUAN PUSTAKA	2
2.1 OpenGL	2
2.2 Konfigurasi OpenGL pada Dev C++ atau VSCode	2
2.3 Cara Kerja OpenGL	3
2.4 Representasi Visual Pemandangan 3D Dalam OpenGL.....	3
BAB III HASIL.....	4
3.1 Source Code.....	4
3.2 Output	9
3.3 Penjelasan	10
BAB IV PENUTUP	12
4.1. Kesimpulan.....	12
DAFTAR PUSTAKA	13

DAFTAR GAMBAR

Gambar 2. 1 Buka Dev C++	2
Gambar 2. 2 Setting Parameter	3
Gambar 2. 3 Ouput Matahari	9
Gambar 2. 4 Output Bulan.....	9

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi grafis memainkan peran penting dalam berbagai bidang seperti industri gim, simulasi ilmiah, arsitektur, dan hiburan. Salah satu alat utama yang digunakan dalam pengembangan grafis komputer adalah OpenGL. OpenGL adalah sebuah API (Application Programming Interface) lintas platform yang memberikan sejumlah besar fungsi untuk memanipulasi grafis dan gambar. Sebagai spesifikasi, OpenGL mendefinisikan hasil dari setiap fungsi dan cara kerjanya, sementara implementasi spesifik dilakukan oleh produsen perangkat keras grafis [1].

Tugas besar ini bertujuan untuk membuat representasi visual pemandangan 3D menggunakan OpenGL dengan bahasa pemrograman C. Elemen-elemen yang divisualisasikan meliputi gunung, matahari, bulan, langit, serta jalan, dengan fitur interaktif seperti transisi siang-malam, pengubahan posisi objek, dan visualisasi sumbu koordinat. Teknik pencahayaan, transformasi objek, dan pewarnaan diterapkan untuk menciptakan efek visual yang realistis dan menarik.

Topik ini penting karena penguasaan OpenGL memungkinkan pengembang untuk menciptakan aplikasi grafis yang efisien dan fleksibel. Teknologi ini memberikan kontrol penuh atas pengolahan grafis, termasuk pipeline grafis modern dan teknik pencahayaan. Visualisasi 3D memiliki peran signifikan dalam mengembangkan solusi kreatif dan inovatif di berbagai industri, sehingga penguasaan teknologi ini menjadi keterampilan penting di masa depan [1].

1.2 Rumusan Masalah

Dari latar belakang diatas dapat dibuat rumusan masalah seperti berikut :

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 representasi visual pemandangan 3D dalam OpenGL

1.3 Tujuan

Dari rumusan masalah diatas dapat dibuat tujuan sebagai berikut :

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 representasi visual pemandangan 3D dalam OpenGL

BAB II TINJAUAN PUSTAKA

2.1 OpenGL

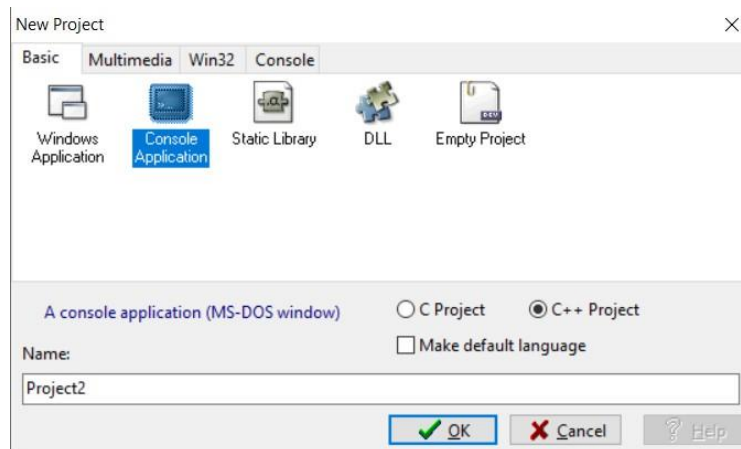
OpenGL (Open Graphics Library) adalah sebuah spesifikasi grafika yang low-level yang menyediakan fungsi untuk mempermudah pekerjaan pemrograman grafika. Fungsi utama OpenGL adalah mengeksekusi kumpulan perintah khusus untuk menghasilkan grafik 2D dan 3D. OpenGL bersifat cross-platform, artinya dapat dijalankan di berbagai sistem operasi, seperti Windows, Linux, dan MacOS. Bahasa pemrograman yang umum digunakan untuk mengimplementasikan OpenGL adalah C++ [2].

Dengan OpenGL, proses visualisasi grafik dapat dilakukan secara real-time tanpa memerlukan rendering ke dalam file terlebih dahulu. Hal ini menjadikannya pilihan utama untuk aplikasi yang memerlukan kecepatan dan fleksibilitas tinggi, seperti simulasi objek 3D dan visualisasi data berbasis grafika komputer [2].

2.2 Konfigurasi OpenGL pada Dev C++ atau VSCode

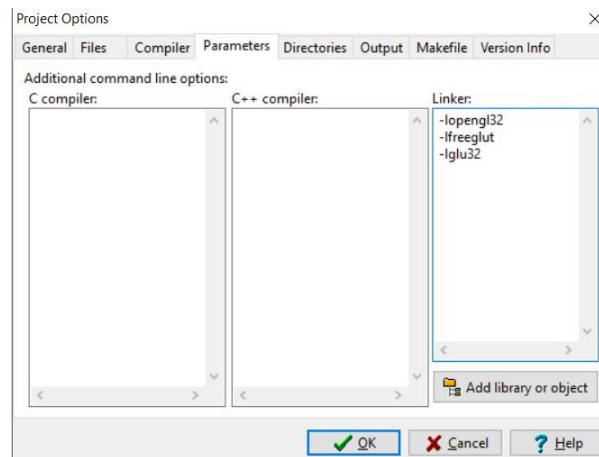
Uraikan dan Jelaskan langkah-langkah untuk konfigurasi OpenGL pada Dev C++. Setiap langkah harus menyertakan Screenshoot. Semua gambar ukurannya harus sama dan posisi gambar berada ditengah (center) dan berikan keterangan dari setiap gambar. Contoh :

1. Buka Aplikasi Dev C++ , Lalu New Project dan berikan nama project



Gambar 2. 1 Buka Dev C++

2. Lalu langkah selanjutnya, kita setting parameternya , Tambahkan : (-lopengl32 -lfreetype -lglu32)



Gambar 2. 2 Setting Parameter

3. Lalu langkah selanjutnya, kita mulai mengerjakan project yang akan kita buat hingga selesai.

2.3 Cara Kerja OpenGL

OpenGL lebih mengarah pada prosedural daripada API (Application Programming Interface) grafis deskriptif. Programmer menentukan langkah-langkah untuk menciptakan efek visual yang diinginkan, termasuk pemanggilan perintah-perintah OpenGL untuk menggambar grafik primitif seperti titik, garis, dan poligon dalam tiga dimensi. OpenGL juga mendukung fitur tambahan seperti pencahayaan, pewarnaan, tekstur, transparansi, dan efek khusus lainnya. Proses ini memanfaatkan fungsi-fungsi seperti pengaturan ukuran layar (viewport), koordinat 3D, dan rendering objek secara real-time [3].

2.4 Representasi Visual Pemandangan 3D Dalam OpenGL

Tema "Representasi Visual Pemandangan 3D dalam OpenGL" bertujuan untuk menggambarkan pemandangan alam yang dinamis dengan fokus pada transisi antara siang dan malam. Proyek ini memanfaatkan teknik manipulasi objek dan pencahayaan yang dapat diatur melalui fungsi-fungsi dalam OpenGL, seperti `glVertex3f` untuk menentukan koordinat objek dalam ruang tiga dimensi dan `glColor3f` untuk memberikan warna pada objek. Transformasi objek dilakukan dengan operasi rotasi, translasi, dan skala guna menciptakan interaksi dinamis antara pengguna dan elemen visual, seperti gunung, awan, matahari, dan bulan [4].

Teknik pencahayaan memegang peran penting dalam menciptakan suasana yang realistis. OpenGL mendukung berbagai teknik pencahayaan, termasuk Basic Lighting untuk sumber cahaya dasar dan Lighting Maps untuk memberikan tekstur cahaya yang lebih kompleks. Sistem koordinat kartesius digunakan untuk memahami dan mengontrol penempatan serta pergerakan objek dalam ruang 3D [1].

Dengan pendekatan ini, proyek tidak hanya memberikan pemahaman tentang dasar-dasar grafika komputer, termasuk pewarnaan, transformasi, dan efek pencahayaan, tetapi juga menyediakan pengalaman interaktif yang realistis dalam visualisasi pemandangan 3D [1].

BAB III

HASIL

3.1 Source Code

Untuk Source code yang sudah kelompok kami buat, adalah sebagai berikut:

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

// Variabel global
float angle = 0.0;
float sunPositionY = 6.0;
float moonPositionY = 6.0;
float scaleSun = 1.0f;
float scaleMoon = 1.0f;
GLfloat lightPosition[] = {0.0, -5.0, 0.0, 1.0};
GLfloat lightAmbient[] = {0.2, 0.2, 0.2, 1.0};
GLfloat lightDiffuse[] = {1.0, 1.0, 1.0, 1.0};
GLfloat lightSpecular[] = {1.0, 1.0, 1.0, 1.0};
float mountainScale = 1;
bool isDay = true; // Mode siang/malam
bool showCartecius = false; // tampilkan garis kartesius

//syifa
void gunung(float offsetX) {
    glPushMatrix();
    glTranslatef(offsetX, 0.0, 0.0);
    glScalef(mountainScale, mountainScale, mountainScale);

    // Sisi depan gunung
    glBegin(GL_TRIANGLES);
    glColor3f(0.7, 0.2, 0.2); // Merah bata di bagian bawah
    glVertex3f(-3, 0.0, -4.0);
    glColor3f(0.9, 0.5, 0.3); // Oranye di tengah
    glVertex3f(0.0, 5.0, 0.0);
    glColor3f(1.0, 0.8, 0.6); // Krem di puncak
    glVertex3f(3, 0.0, -4.0);
    glEnd();

    // Sisi belakang gunung
    glBegin(GL_TRIANGLES);
    glColor3f(0.7, 0.2, 0.2);
    glVertex3f(0.0, 5.0, 0.0);
    glColor3f(0.9, 0.5, 0.3);
    glVertex3f(3, 0.0, 4.0);
    glVertex3f(-3, 0.0, 4.0);
    glEnd();

    // Sisi kanan gunung
    glBegin(GL_TRIANGLES);
    glColor3f(0.7, 0.2, 0.2);
    glVertex3f(0.0, 5.0, 0.0);
    glColor3f(0.9, 0.5, 0.3);
    glVertex3f(3, 0.0, 4.0);
    glVertex3f(3, 0.0, -4.0);
    glEnd();
}
```



```

    // Sisi kiri gunung
    glBegin(GL_TRIANGLES);
    glColor3f(0.7, 0.2, 0.2);
    glVertex3f(0.0, 5.0, 0.0);
    glColor3f(0.9, 0.5, 0.3);
    glVertex3f(-3, 0.0, -4.0);
    glVertex3f(-3, 0.0, 4.0);
    glEnd();

    // Dasar gunung
    glBegin(GL_QUADS);
    glColor3f(0.5, 0.1, 0.1); // Merah gelap untuk dasar
    glVertex3f(-3.0, 0.0, -4.0);
    glVertex3f(3.0, 0.0, -4.0);
    glVertex3f(3.0, 0.0, 4.0);
    glVertex3f(-3.0, 0.0, 4.0);
    glEnd();

    glPopMatrix();
}

//zulfa
void jalan() {
    glBegin(GL_QUADS);
    glColor3f(0.0, 0.0, 0.0);
    glVertex3f(1.5, 0.1, 4.0);
    glVertex3f(-1.5, 0.1, 4.0);
    glVertex3f(-2.0, 0.1, 10.0);
    glVertex3f(2.0, 0.1, 10.0);
    glEnd();
}

//rafi
void matahari() {
    glPushMatrix();
    glTranslatef(0.0, sunPositionY, -5.0);
    glScalef(scaleSun, scaleSun, scaleSun);
    glColor3f(1.0, 1.0, 0.0);
    glutSolidSphere(1.0, 20, 20);
    glPopMatrix();
}

//rafi
void bulan() {
    glPushMatrix();
    glTranslatef(0.0, moonPositionY, -5.0);
    glScalef(scaleMoon, scaleMoon, scaleMoon);
    glColor3f(1.0, 1.0, 1.0);
    glutSolidSphere(1.0, 20, 20);
    glPopMatrix();
}

//syifa
void drawBackground() {
    // Langit bagian atas
    glBegin(GL_QUADS);
    if (isDay) {
        glColor3f(0.5, 0.7, 1.0); // Siang: biru muda
    } else {

```

```

        glColor3f(0.0, 0.0, 0.2); // Malam: biru gelap
    }
    glVertex3f(-10.0, 10.0, -10.0);
    glVertex3f(10.0, 10.0, -10.0);
    glVertex3f(10.0, 0.0, -10.0);
    glVertex3f(-10.0, 0.0, -10.0);
    glEnd();

    // Langit bagian bawah (hijau tetap ada)
    glBegin(GL_QUADS);
    glColor3f(0.0, 0.6, 0.0); // Warna hijau untuk tanah
    glVertex3f(10.0, 0.0, 10.0);
    glVertex3f(-10.0, 0.0, 10.0);
    glVertex3f(-10.0, 0.0, -10.0);
    glVertex3f(10.0, 0.0, -10.0);
    glEnd();
}
//zulfa
void drawCartecius()
{
    glLineWidth(2.0); // Lebar garis lebih tebal agar terlihat
    jelas

    // Sumbu X
    glColor3f(1.0, 0.0, 0.0);
    glBegin(GL_LINES);
    glVertex3f(-50.0, 0.0, 0.0);
    glVertex3f(50.0, 0.0, 0.0);
    glEnd();

    // Sumbu Y
    glColor3f(0.0, 1.0, 0.0);
    glBegin(GL_LINES);
    glVertex3f(0.0, -50.0, 0.0);
    glVertex3f(0.0, 50.0, 0.0);
    glEnd();

    // Sumbu Z
    glColor3f(0.0, 0.0, 1.0);
    glBegin(GL_LINES);
    glVertex3f(0.0, 0.0, -50.0);
    glVertex3f(0.0, 0.0, 50.0);
    glEnd();
}
//syifa
void display() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    gluLookAt(
        10.0 * sin(angle), 5.0, 10.0 * cos(angle),
        0.0, 0.0, 0.0,
        0.0, 1.0, 0.0
    );

    glLightfv(GL_LIGHT0, GL_POSITION, lightPosition);
    glLightfv(GL_LIGHT0, GL_AMBIENT, lightAmbient);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, lightDiffuse);
}

```

```

    glLightfv(GL_LIGHT0, GL_SPECULAR, lightSpecular);

    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);

    // Gambar garis kartesius jika aktif
    if (showCartecius) {
        drawCartecius();
    }

    drawBackground();
    gunung(-1.5);
    gunung(1.5);

    if (isDay) {
        matahari();
    } else {
        bulan();
    }

    jalan();

    glDisable(GL_LIGHT0);
    glDisable(GL_LIGHTING);

    glutSwapBuffers();
}
//syifa
void reshape(int width, int height) {
    glViewport(0, 0, (GLsizei)width, (GLsizei)height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(80.0, (GLfloat)width/(GLfloat)height, 1.0,
100.0);
    glMatrixMode(GL_MODELVIEW);
}
//syifa
void init() {
    glClearColor(0.5, 0.7, 1.0, 0.0);
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_LIGHTING);
    glEnable(GL_COLOR_MATERIAL);
}
//rafi
void keyboard(unsigned char key, int x, int y) {
    switch (key) {
        case 'a':
            angle += 0.1;
            break;
        case 'd':
            angle -= 0.1;
            break;
        case '1': // Siang
            isDay = true;
            glClearColor(0.5, 0.7, 1.0, 0.0);
            break;
        case '2': // Malam

```

```

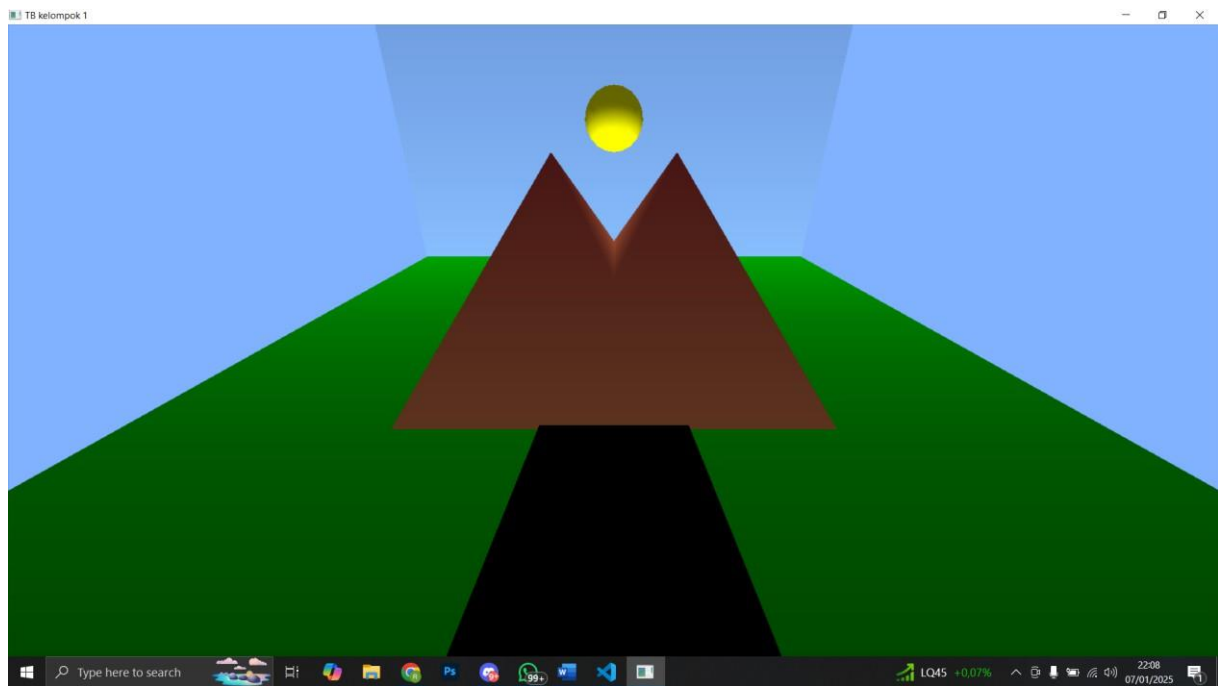
        isDay = false;
        glClearColor(0.0, 0.0, 0.2, 0.0);
        break;
    case '=':
        scaleSun += 0.1f;
        break;
    case '-':
        scaleSun -= 0.1f;
        break;
    case ']':
        scaleMoon += 0.1f;
        break;
    case '[':
        scaleMoon -= 0.1f;
        break;
    case 'c': // garis kartesius
        showCartecius = !showCartecius;
        break;
    }
    glutPostRedisplay();
}

void specialKeys(int key, int x, int y) {
    switch (key) {
        case GLUT_KEY_UP:
            sunPositionY += 0.1;
            break;
        case GLUT_KEY_DOWN:
            sunPositionY -= 0.1;
            break;
        case GLUT_KEY_LEFT:
            moonPositionY += 0.1;
            break;
        case GLUT_KEY_RIGHT:
            moonPositionY -= 0.1;
            break;
    }
    glutPostRedisplay();
}

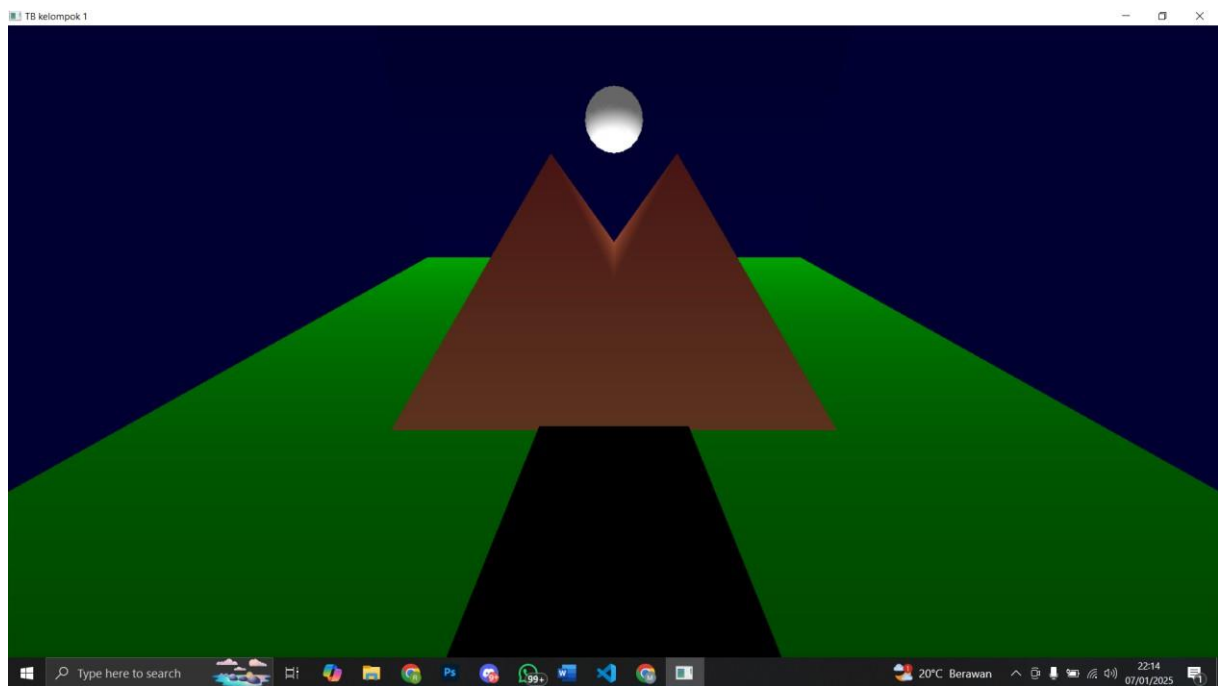
int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(800, 600);
    glutCreateWindow("TB kelompok 1");
    init();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutKeyboardFunc(keyboard);
    glutSpecialFunc(specialKeys);
    glutMainLoop();
    return 0;
}

```

3.2 Output



Gambar 2. 3 Ouput Matahari



Gambar 2. 4 Output Bulan

Link Output : <https://drive.google.com/file/d/1VsyEgZEWfj4geepGOks1mEtrfH-kuVez/view?usp=sharing>

3.3 Penjelasan

1. Header dan Variabel Global

Program ini menggunakan dua header utama, yaitu `#include <GL/glut.h>` untuk fungsi-fungsi OpenGL dan GLUT, serta `#include <math.h>` untuk operasi matematika seperti sinus dan kosinus. Variabel global yang digunakan meliputi `angle` untuk mengatur sudut rotasi kamera, `sunPositionY` dan `moonPositionY` untuk menentukan posisi vertikal matahari dan bulan, serta `scaleSun` dan `scaleMoon` untuk mengontrol ukuran skala kedua elemen tersebut. Selain itu, variabel pencahayaan seperti `lightPosition`, `lightAmbient`, `lightDiffuse`, dan `lightSpecular` digunakan untuk menentukan sifat cahaya. Program juga memiliki variabel boolean `isDay` untuk menentukan mode siang atau malam dan `showCartecius` untuk menampilkan atau menyembunyikan garis kartesius.

2. Fungsi Menggambar Gunung

Fungsi `gunung(float offsetX)` bertanggung jawab menggambar gunung berbentuk piramida. Gunung terdiri dari beberapa sisi segitiga (depan, belakang, kanan, kiri) yang dihias dengan gradasi warna mulai dari merah bata di bagian bawah, oranye di tengah, hingga krem di puncaknya. Dasar gunung berupa persegi panjang berwarna merah gelap. Fungsi ini menggunakan transformasi `glTranslatef` untuk mengatur posisi gunung berdasarkan offset yang diberikan.

3. Fungsi Menggambar Jalan

Fungsi `jalan()` menggambar jalan berbentuk persegi panjang yang berwarna hitam. Jalan ini dirancang menggunakan empat titik sudut yang didefinisikan dengan `glVertex3f`, menggambarkan jalur lurus di tengah pemandangan.

4. Fungsi Menggambar Matahari

Fungsi `matahari()` digunakan untuk menggambar matahari dengan bentuk bola menggunakan `glutSolidSphere`. Matahari berwarna kuning, dengan posisinya ditentukan oleh variabel `sunPositionY`. Transformasi posisi dan skala diterapkan melalui `glTranslatef` dan `glScalef`.

5. Fungsi Menggambar Bulan

Fungsi `bulan()` menggambar bulan dengan cara serupa seperti matahari, menggunakan `glutSolidSphere`. Bulan berwarna putih, dan posisinya diatur oleh variabel `moonPositionY`. Transformasi posisi dan skala juga diterapkan dengan `glTranslatef` dan `glScalef`.

6. Fungsi Menggambar Latar Belakang

Fungsi `drawBackground()` menggambar latar belakang yang terdiri dari dua bagian: langit dan tanah. Langit berwarna biru muda untuk siang hari dan biru gelap untuk malam, sementara tanah selalu berwarna hijau. Kedua elemen ini digambar menggunakan quad (persegi panjang) dengan warna yang diatur sesuai mode siang atau malam.

7. Fungsi Menggambar Garis Kartesius

Fungsi `drawCartecius()` menggambar garis kartesius berupa sumbu X, Y, dan Z menggunakan warna merah. Garis-garis ini digambar dengan fungsi `glBegin(GL_LINES)` dan hanya ditampilkan jika variabel `showCartecius` bernilai true.

8. Fungsi Rendering dan Transformasi

Fungsi `display()` bertanggung jawab untuk merender seluruh elemen pemandangan. Kamera diatur menggunakan `gluLookAt` untuk memberikan sudut pandang dinamis berdasarkan rotasi. Fungsi ini juga mengaktifkan pencahayaan dengan `glEnable(GL_LIGHTING)` dan memanggil berbagai fungsi menggambar seperti latar belakang, gunung, jalan, dan matahari atau bulan. Jika `showCartecius` diaktifkan, garis kartesius juga akan digambar.

9. Fungsi Penanganan Ukuran Jendela

Fungsi `reshape(int width, int height)` menangani perubahan ukuran jendela. Fungsi ini mengatur ulang viewport menggunakan `glViewport` dan menyesuaikan aspek rasio dengan `gluPerspective`, memastikan tampilan tetap proporsional.

10. Fungsi Interaksi Keyboard

Fungsi `keyboard(unsigned char key, int x, int y)` menangani input dari keyboard. Tombol 'a' dan 'd' mengatur rotasi kamera, tombol '1' dan '2' mengubah mode siang atau malam, sedangkan tombol '=' dan '-' memperbesar atau memperkecil matahari. Tombol ']' dan '[' digunakan untuk mengatur ukuran bulan, sementara tombol 'c' digunakan untuk menampilkan atau menyembunyikan garis kartesius.

11. Fungsi Interaksi Tombol Panah

Fungsi `specialKeys(int key, int x, int y)` menangani input tombol panah. Tombol panah atas dan bawah menggerakkan matahari ke atas atau ke bawah, sedangkan tombol panah kiri dan kanan menggerakkan bulan ke atas atau ke bawah.

12. Fungsi Inisialisasi dan Fungsi Utama

Fungsi `init()` menginisialisasi pengaturan OpenGL, seperti mengatur warna latar belakang awal, mengaktifkan fitur `GL_DEPTH_TEST` untuk rendering 3D, serta mengaktifkan pencahayaan dengan `GL_LIGHTING`. Fungsi `main(int argc, char** argv)` adalah fungsi utama yang menginisialisasi GLUT, membuat jendela aplikasi, dan mengatur fungsi callback seperti `display`, `reshape`, `keyboard`, serta `specialKeys`. Program dijalankan dalam loop utama menggunakan `glutMainLoop`.

BAB IV PENUTUP

4.1. Kesimpulan

Kesimpulan dari praktikum ini adalah bahwa implementasi OpenGL dalam pengembangan grafika komputer memberikan pemahaman yang mendalam mengenai konsep dasar visualisasi 3D, termasuk manipulasi objek, pencahayaan, pewarnaan, dan transformasi koordinat. Proyek ini berhasil merepresentasikan pemandangan alam 3D yang dinamis dengan fitur transisi siang-malam, interaksi pengguna, dan visualisasi sumbu koordinat. Penguasaan teknologi OpenGL memberikan manfaat signifikan dalam pengembangan aplikasi grafis yang efisien dan fleksibel, serta menjadi dasar untuk eksplorasi lebih lanjut dalam bidang grafika komputer. Praktikum ini juga mempertegas pentingnya pemrograman grafis dalam berbagai aplikasi modern, seperti simulasi, gim, dan visualisasi data.

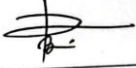
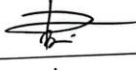
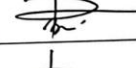
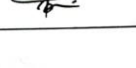
DAFTAR PUSTAKA

- [1] J. de Vries, *Learn OpenGL*, vol. 42, no. 4. 2020.
- [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] D. Suhardiman *et al.*, "Pembuatan Simulasi Pergerakan Objek 3D (Tiga Dimensi) Menggunakan OpenGL," *J. Inform.*, vol. 2, no. 1, pp. 1–4, 2015.
- [4] A. S. Riyadi, "Implementasi Open Gl32 Untuk Memanipulasi Gambar Segitiga Dan Segiempat," *J. KomTekInfo Fak. Ilmu Komput.*, vol. 1, no. 1, 2014.

Lampiran

ASISTENSI TUGAS PRAKTIKUM GRAFIK KOMPUTER

Nama Pembimbing : Garnis Kirani
Kelompok & Kelas : Kelompok 1 kelas C
Anggota Kelompok : - Aswita Azsma H.
- M. Rafi
- Zulfa Ghinan
Tema : ~~zulfah~~ Permandangan

No	Tanggal	Uraian Asistensi	Tanda Tangan
1	27/12 ²⁰²⁴	Sketcha Project	
2	30/12 ²⁰²⁴	Progres Coding	
3	4/01 ²⁰²⁵	Finishing	
4	8/01 ²⁰²⁵	Laporan	
5			

Instruktur Praktikum



Arul Budi Kalimat, S.Kom