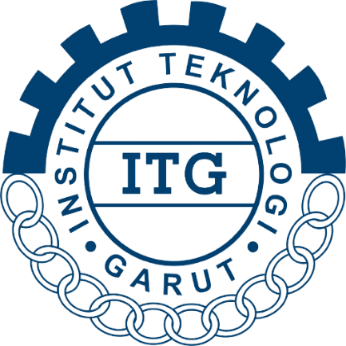
**LAPORAN PRATIKUM GRAFIK KOMPUTER**

Diajukan untuk memenuhi Tugas mata kuliah Pratikum Grafik Komputer

**IMPELEMENTASI PROGRAM GEDUNG 3D KEDALAM OPENGL**

Dosen Pengampu : Sri Rahayu, M.Kom

Instruktur Pratikum : Arul Budi Kalimat, S.Kom



Disusun oleh

Kelompok : 5

Yogi Sugiman

2306089

Rijal Daniel

2306090

Daffa Muhammad Hasya

2306088

**PROGRAM STUDI TEKNIK INFORMATIKA JURUSAN ILMU KOMPUTER**

**INSTITUT TEKNOLOGI GARUT 2024**

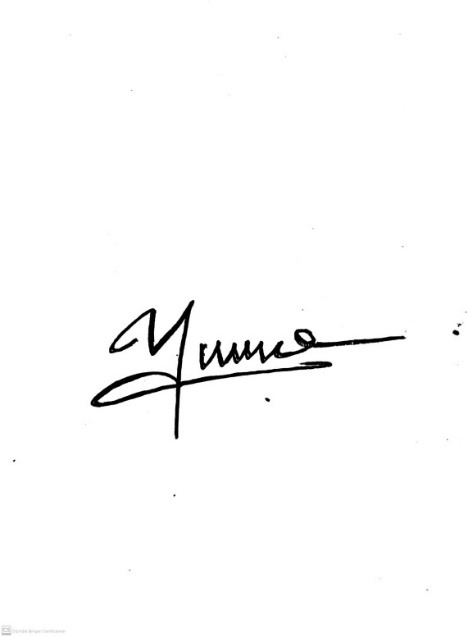
# KATA PENGANTAR

Puji dan syukur kehadirat 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 Membuat Gedung 3D diOpenGL.

Kami mengucapkan terima kasih kepada dosen pengampu Sri Rahayu, S.T. 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, 14 Desember 2024



Yogi Sugiman

# DAFTAR ISI

[KATA PENGANTAR i](#_Toc186293558)

[DAFTAR ISI ii](#_Toc186293559)

[DAFTAR GAMBAR iii](#_Toc186293560)

[BAB I PENDAHULUAN 1](#_Toc186293561)

[1.1 Latar Belakang 1](#_Toc186293562)

[1.2 Rumusan Masalah 1](#_Toc186293563)

[1.3 Tujuan 1](#_Toc186293564)

[BAB II TINJAUAN PUSTAKA 2](#_Toc186293565)

[2.1 OpenGL 2](#_Toc186293566)

[2.2 Konfigurasi OpenGL pada Dev C++ atau VSCode 2](#_Toc186293567)

[2.3 Cara Kerja OpenGL 3](#_Toc186293568)

[2.4 Membuat Gedung 3d di Dalam OpenGL 3](#_Toc186293569)

[BAB III HASIL 4](#_Toc186293570)

[3.1 Source Code 4](#_Toc186293571)

[3.2 Output 4](#_Toc186293572)

[3.3 Penjelasan 4](#_Toc186293573)

[BAB IV 5](#_Toc186293574)

[4.1. Kesimpulan 5](#_Toc186293575)

[DAFTAR PUSTAKA 6](#_Toc186293576)

# DAFTAR GAMBAR

[**Gambar 1 File Manager** 4](#_Toc188264641)

[**Gambar 2 VSCode** 5](#_Toc188264642)

[**Gambar 3 Output Program** 26](#_Toc188264643)

# BAB I PENDAHULUAN

## Latar Belakang

Dewasa ini teknologi semakin maju dan berkembang semakin pesat, bahkan hampir semua bidang pekerjaan masyarakat tidak dapat lepas dari keberadaan teknologi. Salah satu teknologi yang saat ini banyak dimanfaatkan masyarakat untuk menunjang pekerjaannya adalah grafika komputer. Bidang yang paling banyak menggunakan grafika komputer adalah bidang yang langsung berinteraksi dengan manusia dan yang bersifat visual. [1]

Grafika komputer adalah suatu proses pembuatan, penyimpanan, manipulasi model dan citra. Grafika komputer diartikan sebagai penggabungan dan kombinasi dari titik, garis, poligon serta bentuk lainnya yang akan menghasilkan objek berbentuk dua dimensi(2D) atau tiga dimensi(3D). Grafik komputer 2D merupakan pembuatan objek gambar yang masih berbasis perspektif 2 titik. Contohnya seperti gambar teks, bangun 2D seperti segitiga, persegi, lingkaran, dan sebagainya. Sedangkan grafik komputer 3D adalah suatu grafis yang menggunakan 3 titik perspektif dengan cara matematis untuk menampilkan atau melihat suatu objek, dimana gambar tersebut dapat dilihat secara menyeluruh dan hasilnya akan lebih nyata. Perpindahan objek pada grafika komputer dikenal dengan istilah transformasi yang terdiri dari translasi, penskalaan (scaling), perputaran (rotasi), dan sebagainya. [2]

Tugas Besar ini bertujuan untuk membuat objek 3D dengan tema “ Gedung 3D “ dengan cara matermatis atau melihat suatu objek, dimana gambar tersebut secara menyeluruh dan hasilya akan keliahatan nyata. Objek dari tema tersebut meliputi 2 gedung, lantai gedung, kursi taman, jalan serta halaman dari gedung. Dalam pemodelan objek gedung 3D ini, berbagai objek dimodelkan dan dimodifikasi dalam kondisi tertentu. Pembentukan objek ini dapat melakukan berbagai fungsi biasa yang disebut transformasi. Tujuan ari transformasi ini yaitu mengubah atau menyesuaikan komposisi semua objek, melihat objek dari sudut pandang yang berbeda, memindahkan semua objek dari tempat ke tempat lainnya yag berbeda, bergerak ke berbagai sisi dengan bentuk 3D sehingga objek tersebut akan lebih dinamis. Pada laporan Tugas Besar ini terddapat suatu Lightning atau pencahayaan, yang mana fungsi ddari pencahayaan ini yaitu untuk mengubah cahaya gelap ke terang begitu juga sebaliknya.

## Rumusan Masalah

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

* + 1. Apa yang dimkasud dengan OpenGL ?
    2. Bagaimana cara mengkonfigurasi OpenGL pada Dev C++ atau VSCode?
    3. Bagaimana cara kerja dari OpenGL ?
    4. Bagaimana membuat ***Gedung 3D*** dalam OpenGL

## Tujuan

Nyatakan tujuan dari laporan praktikum. Misalkan :

* + 1. Mengethaui apa itu OpenGL
    2. Mengetahui cara mengkonfigurasi OpenGL pada Dev C++ atau VSCode.
    3. Mengetahui cara kerja dari OpenGL
    4. Mengetahui cara pembuatan ***Gedung 3D*** dalam OpenGL

# BAB II TINJAUAN PUSTAKA

## OpenGL

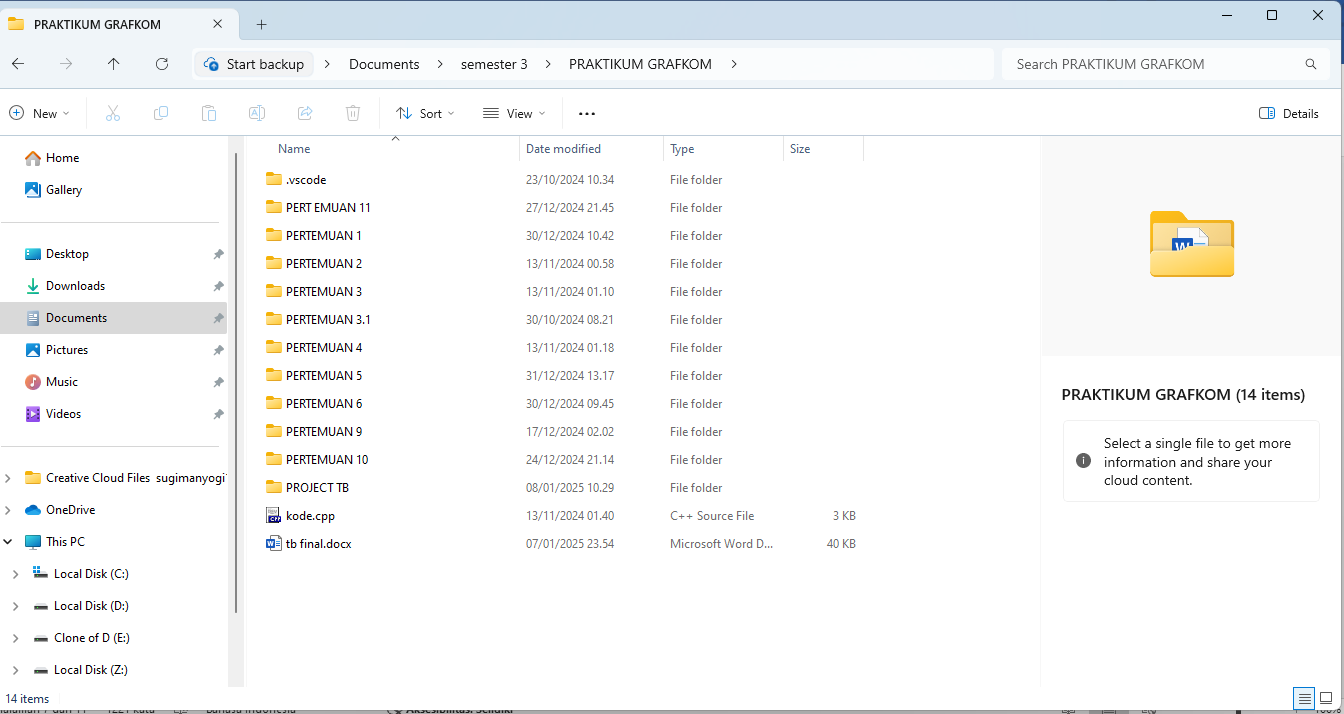
Pada awal mulanya OpenGL merupakan bahasa pemrograman yang didesain untuk digunakan pada bahasa pemrograman C dan juga bahasa pemrograman C++. namun seiring dengan perkembangan waktu dan kebutuhan yang semakin besar akhirnya OpenGL berubah menjadi sebuah library yang multi bahasa pemrograman hingga saat ini mampu berjalan pada bahasa pemrograman diluar C dan C++ seperti bahasa pemrograman java, visual basic, delphi serta bahasa pemrograman fortran. Selain itu OpenGL sendiri dapat dikategorikan sebuah antarmuka pemrograman aplikasi (Application Programming Interface), dimana konsep seperti ini membuat OpenGL tidak bergantung pada bahasa pemrograman tertentu sehingga dapat pula digunakan untuk cross-platform seperti berjalan pada sistem operasi windows, linux dan sistem operasi lainnya (Baek & Kim, 2019).

OpenGL (Open Graphics Library) merupakan sebuah standar API yang dapat digunakan untuk membuat sebuah aplikasi yang berbasiskan grafis, baik dalam bentuk gambar dua dimensi (2D) maupun tiga dimensi (3D). OpenGL sendiri memiliki sifat sebagai cross-platform atau dapat dijalankan dalam berbagai platform sistem operasi dan berbagai macam bahasa pemrograman sehingga banyak peneliti dan pengembang memanfaatkan library OpenGL ini. Selain itu keuntungan dari pendekatan OpenGL sendiri adalah memungkinkan fleksibilitas yang besar dalam proses menghasilkan sebuah gambar. Aplikasi ini memiliki konsep open-source atau sumber terbuka untuk kebutuhan trade-off rendering kecepatan dan kualitas gambar dengan mengubah langkah-langkah dimana foto tersebut diperoleh (Loseille & Feuillet, 2021).[3]

## Konfigurasi OpenGL pada Dev C++ atau VSCode

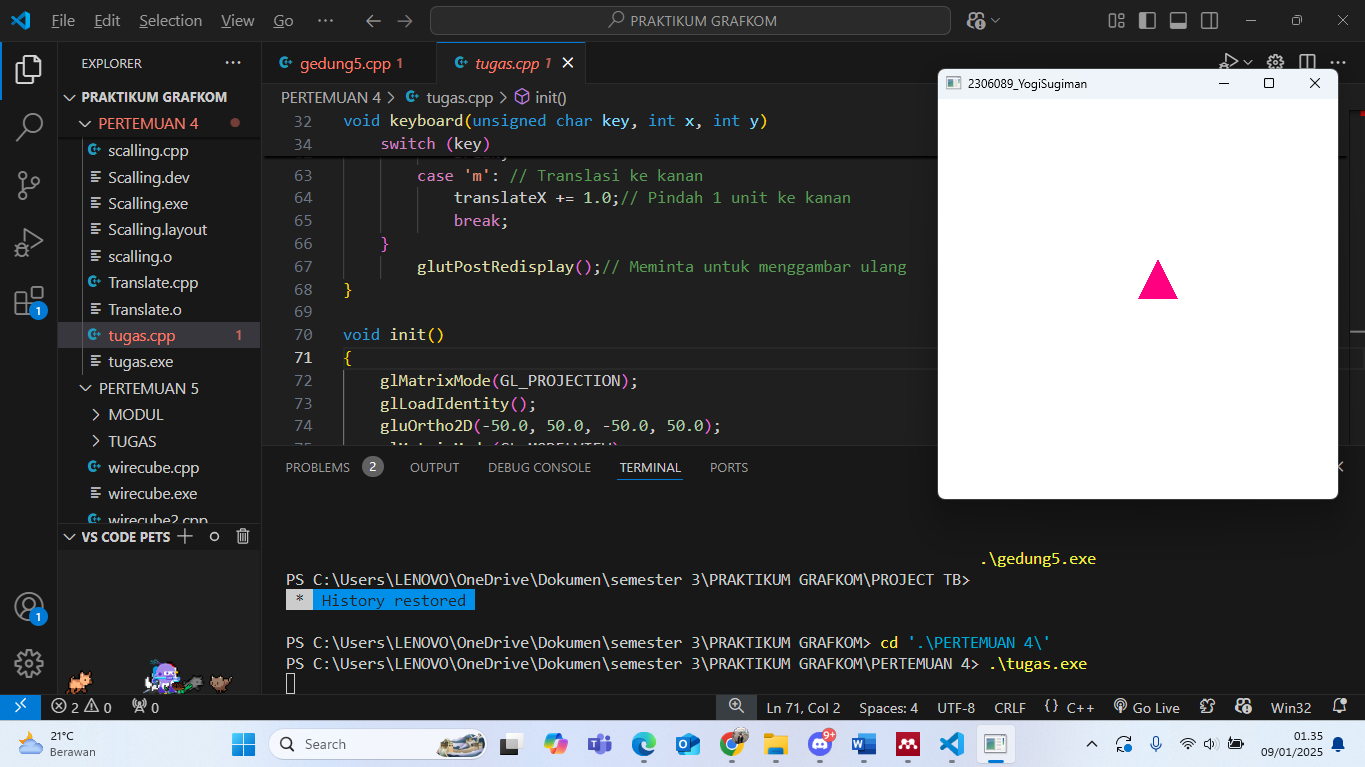
Pada laporan ini kita akan menginplementasikan dan mengonfirasi OpenGL pada aplikasi VSCode untuk setiap langkah-langkahnya sebagai berikut.

1. Buka aplikasi VSCode nya
2. Download file parameternya, apabila sudah ada kita langsung konfigurasikan saja
3. Buka pada file manager dan simpan sesuai anda ingin membuat project dan masukan paramternya atau satu file kan asma file projectnya seperti pada gambar dibawah ini.



**Gambar 1 File Manager**

* + 1. Apabila sudah disatu file kan, maka kita lanjut untuk pengeksekusian ke programnya. Pertama pertama kita buka aplikasi VSCode nya dan pilihlah program yang ingin kita eksukusi. Apabila sudah ada projectnya yaitu dengan mengklik ***contrrol + shift + b*** yang mana dari fungsi tersebut yaitu untuk melanjutkan project ke proses ***Build Run.*** Apabila sudah di build dan sudah tidak ada yang error, maka kita masuk ke pengekeksekusian pada programnya dengan membuka ***File*** menggunakan ***cd [nama file].exe*** maka langsung enter seperti contoh pada gambar dibawah ini.



**Gambar 2 VSCode**

## Cara Kerja OpenGL

1. **Pengantar OpenGl**

OpenGL adalah standar industri untuk pemrograman grafik yang memungkinkan pengembang membuat aplikasi grafis yang kompleks dengan kemampuan untuk menggambar objek, mengatur pencahayaan, tekstur, dan banyak aspek grafik komputer lainnya.

1. **Arsitektur OpenGL**

Dengan arsitektur berbasis state machine, OpenGL dapat menyimpan status dari berbagai parameter grafis. Berikut ini adalah beberapa elemen utama arsitektur OpenGL:

1. Context: Sebelum Anda dapat menggunakan OpenGL, Anda harus membuat konteks OpenGL. Konteks ini menyimpan semua informasi tentang status OpenGL, seperti pengaturan pencahayaan, tekstur, dan buffer.
2. Pipeline Grafis: Pipeline grafis yang terdiri dari beberapa tahap digunakan oleh OpenGL untuk memproses data yang dikirimkan oleh aplikasi ke layar.
3. Vertex Processing: Mengolah data verteks (titik-titik yang membentuk objek) dan menerapkan transformasi (seperti translasi, rotasi, dan skala).
4. Primitive Assembly: Menggabungkan verteks menjadi primitif (seperti garis, segitiga, dll.).
5. Rasterization: Mengubah primitif menjadi fragmen (pixel) yang akan ditampilkan di layar.
6. Fragment Processing: Mengolah fragmen untuk menentukan warna akhir dan atribut lainnya.
7. Output Merger: Menggabungkan fragmen yang dihasilkan dengan buffer warna untuk menghasilkan gambar akhir.
8. **Langkah-langkah Dasar dalam Menggunakan OpenGL**

Berikut adalah langkah-langkah dasar yang biasanya diikuti saat menggunakan OpenGL:

1. Inisialisasi:

Buat jendela dan konteks OpenGL menggunakan library seperti GLUT, GLFW, atau SDL Atur parameter awal seperti warna latar belakang, mode tampilan, dan pengaturan pencahayaan.

1. Menggambar Objek:

Definisikan objek 3D dengan menggunakan verteks. Anda dapat menggunakan fungsi seperti glBegin() dan glEnd() untuk menggambar primitif.

Terapkan transformasi (translasi, rotasi, skala) pada objek menggunakan fungsi seperti glTranslatef(), glRotatef(), dan glScalef().

1. Pengaturan Pencahayaan dan Tekstur:

Atur sumber cahaya dan parameter pencahayaan menggunakan fungsi seperti glLightfv().

Terapkan tekstur pada objek menggunakan fungsi seperti glBindTexture() dan glTexImage2D().

1. Menggambar Frame:

Bersihkan buffer warna dan kedalaman menggunakan glClear().

Gambar objek dan panggil fungsi untuk menggambar ulang.

Tampilkan hasil di layar dengan glutSwapBuffers() atau fungsi serupa.

1. Interaksi Pengguna:

Tangani input dari keyboard dan mouse untuk memungkinkan interaksi pengguna dengan objek di dalam scene.

1. Loop Utama:

Masukkan aplikasi ke dalam loop utama yang terus-menerus memperbarui tampilan dan menangani input pengguna.

## Membuat Gedung 3D didalam OpenGL

## Konsep dari proyek ini adalah menciptakan sebuah simulasi 3D yang menggambarkan lingkungan perkotaan dengan elemen-elemen seperti gedung, jalan, dan taman. Proyek ini menggunakan OpenGL dan GLUT untuk menggambar objek-objek 3D dan memberikan interaksi kepada pengguna melalui mouse dan keyboard. Berikut adalah beberapa aspek kunci dari konsep proyek ini:

## Lingkungan Perkotaan 3D:

## Proyek ini bertujuan untuk merepresentasikan sebuah lingkungan perkotaan yang terdiri dari beberapa gedung, jalan, dan elemen taman seperti kursi dan meja. Ini memberikan gambaran visual yang menarik dan realistis tentang bagaimana sebuah kota dapat terlihat.

## Interaksi Pengguna:

## Pengguna dapat berinteraksi dengan lingkungan 3D menggunakan mouse dan keyboard. Dengan mouse, pengguna dapat memutar tampilan untuk melihat objek dari berbagai sudut. Dengan keyboard, pengguna dapat mengubah pengaturan pencahayaan, mengubah ukuran dan posisi objek, serta menampilkan atau menyembunyikan sumbu koordinat.

## Pencahayaan

## Pencahayaan adalah elemen penting dalam rendering 3D yang memberikan kedalaman dan realisme pada objek. Proyek ini mengimplementasikan pencahayaan ambient dan diffuse, yang memungkinkan objek terlihat lebih hidup dan memberikan efek bayangan yang lebih baik.

## Transformasi Objek:

## Proyek ini menggunakan transformasi geometris seperti translasi, rotasi, dan skala untuk memanipulasi objek dalam ruang 3D. Ini memungkinkan pengguna untuk melihat objek dari berbagai perspektif dan mengubah ukuran objek sesuai kebutuhan.

## Visualisasi Koordinat:

## Dengan menampilkan sumbu koordinat, proyek ini membantu pengguna memahami posisi objek dalam ruang 3D. Ini juga memberikan referensi visual yang berguna saat berinteraksi dengan objek.

## Penggunaan Poligon untuk Menggambar:

## Objek-objek dalam proyek ini digambar menggunakan poligon, yang merupakan cara umum untuk merepresentasikan bentuk 3D dalam grafika komputer. Setiap objek memiliki warna yang ditentukan, memberikan visualisasi yang jelas.

## Pengembangan dan Pembelajaran:

## Proyek ini juga berfungsi sebagai alat pembelajaran bagi pengembang yang ingin memahami dasar-dasar pemrograman grafis 3D, penggunaan OpenGL, dan konsep-konsep dasar dalam pengembangan game atau simulasi.

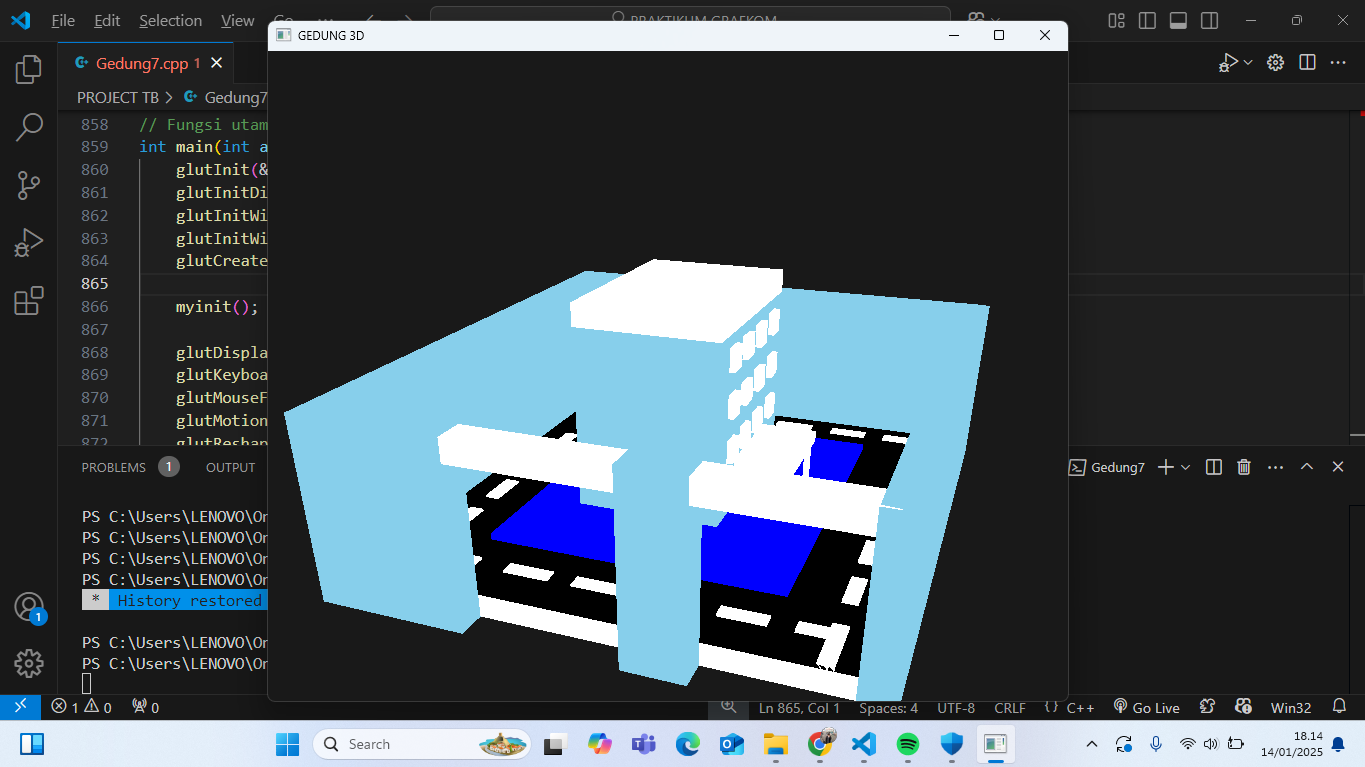
# BAB III HASIL

## Source Code

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

|  |
| --- |
| #include <GL/freeglut.h>  #include <math.h>  // Variabel untuk rotasi menggunakan mouse  float xrot = 0; // Rotasi di sumbu X  float yrot = 0; // Rotasi di sumbu Y  float xdiff = 0; // Selisih posisi mouse di sumbu X  float ydiff = 0; // Selisih posisi mouse di sumbu Y  bool mouseDown = true; // Status mouse (apakah ditekan atau tidak)  // Variabel untuk pengaturan pencahayaan  GLfloat light\_position[] = { 0.0, 20.0, 70.0, 1.0 };  // 0.0  -> Koordinat x dari sumber cahaya (posisi horizontal)  // 20.0 -> Koordinat y dari sumber cahaya (posisi vertikal)  // 70.0 -> Koordinat z dari sumber cahaya (kedalaman)  // 1.0  -> Menunjukkan bahwa ini adalah cahaya posisi (bukan cahaya arah)  GLfloat ambient\_light[] = { 0.2, 0.2, 0.2, 1.0 };  // Parameter:  // 0.2 -> Komponen merah dari cahaya ambient (intensitas cahaya merah)  // 0.2 -> Komponen hijau dari cahaya ambient (intensitas cahaya hijau)  // 0.2 -> Komponen biru dari cahaya ambient (intensitas cahaya biru)  // 1.0 -> Komponen alpha (opasitas), 1.0 berarti sepenuhnya terlihat  GLfloat diffuse\_light[] = { 1.0, 1.0, 1.0, 1.0 };  // Parameter:  // 1.0 -> Komponen merah dari cahaya diffuse (intensitas cahaya merah)  // 1.0 -> Komponen hijau dari cahaya diffuse (intensitas cahaya hijau)  // 1.0 -> Komponen biru dari cahaya diffuse (intensitas cahaya biru)  // 1.0 -> Komponen alpha (opasitas), 1.0 berarti sepenuhnya terlihat  bool lightingEnabled = true; // Status pencahayaan  // Variabel untuk transformasi objek  float xPOS = 1.0; // Skala sumbu X  float yPOS = 1.0; // Skala sumbu Y  float angle = 0.0; // Sudut rotasi  float translateX = 0.0; // Posisi translasi sumbu X  float translateY = 0.0; // Posisi translasi sumbu Y  bool hidden = false; // Status untuk menampilkan atau menyembunyikan sumbu koordinat  // Deklarasi fungsi untuk menggambar sumbu koordinat  void hiddenCarte();  // Fungsi untuk mengatur ukuran jendela dan proyeksi  void ukur(int lebar, int tinggi) {      if (tinggi == 0) tinggi = 1; // Menghindari pembagian dengan nol      glMatrixMode(GL\_PROJECTION); // Mengatur mode matriks proyeksi      glLoadIdentity(); // Mengatur matriks proyeksi ke identitas      gluPerspective(45, lebar / tinggi, 5, 450); // Mengatur perspektif      glTranslatef(0, 0, -340); // Menggeser pandangan ke belakang      glMatrixMode(GL\_MODELVIEW); // Kembali ke mode matriks model  }  // Fungsi untuk menggambar sumbu koordinat  void drawCartecius() {      glColor3f(1.0, 0.0, 0.0); // Mengatur warna garis menjadi merah      glBegin(GL\_LINES); // Memulai menggambar garis      // Garis sumbu X      glVertex3f(-150.0, 0.0, 0.0);      glVertex3f(150.0, 0.0, 0.0);      // Garis sumbu Y      glVertex3f(0.0, -150.0, 0.0);      glVertex3f(0.0, 150.0, 0.0);      // Garis sumbu Z      glVertex3f(0.0, 0.0, -150.0);      glVertex3f(0.0, 0.0, 150.0);      glEnd(); // Mengakhiri menggambar garis  }  // Fungsi untuk menggambar sumbu koordinat jika status hidden adalah true  void hiddenCarte() {      if (hidden) {          drawCartecius(); // Memanggil fungsi untuk menggambar sumbu koordinat      }  }  // Fungsi untuk inisialisasi pengaturan OpenGL  void myinit(void) {      glClearColor(0.0, 0.0, 0.0, 0.0); // Mengatur warna latar belakang      glMatrixMode(GL\_PROJECTION); // Mengatur mode matriks proyeksi      glEnable(GL\_DEPTH\_TEST); // Mengaktifkan pengujian kedalaman      glMatrixMode(GL\_MODELVIEW); // Kembali ke mode matriks model      glPointSize(10.0); // Mengatur ukuran titik      glLineWidth(7.0f); // Mengatur lebar garis  }  // Fungsi idle untuk memperbarui tampilan saat tidak ada interaksi  void idle() {      if (mouseDown) { // Jika mouse ditekan          xrot += 0.0; // Menambah rotasi di sumbu X          yrot += 0.0; // Menambah rotasi di sumbu Y      }      glutPostRedisplay(); // Meminta untuk menggambar ulang  }  // Fungsi untuk menangani input dari keyboard  void kunci(unsigned char key, int x, int y)  {      switch (key) {      case 'q': // Jika tombol 'q' ditekan          exit(0); // Keluar dari program          break;      case 'l': // Jika tombol 'l' ditekan          lightingEnabled = !lightingEnabled; // Mengubah status pencahayaan          if (lightingEnabled) {              glEnable(GL\_LIGHTING); // Mengaktifkan pencahayaan          } else {              glDisable(GL\_LIGHTING); // Menonaktifkan pencahayaan          }          glutPostRedisplay(); // Meminta untuk menggambar ulang          break;      case 'a': // Jika tombol 'a' ditekan          ambient\_light[0] += 0.1f; // Meningkatkan intensitas cahaya ambient          ambient\_light[1] += 0.1f;          ambient\_light[2] += 0.1f;          if (ambient\_light[0] > 1.0f) ambient\_light[0] = 1.0f; // Membatasi nilai maksimum          if (ambient\_light[1] > 1.0f) ambient\_light[1] = 1.0f;          if (ambient\_light[2] > 1.0f) ambient\_light[2] = 1.0f;          glLightfv(GL\_LIGHT0, GL\_AMBIENT, ambient\_light); // Mengatur cahaya ambient          glutPostRedisplay(); // Meminta untuk menggambar ulang          break;      case 'd': // Jika tombol 'd' ditekan          ambient\_light[0] -= 0.1f; // Mengurangi intensitas cahaya ambient          ambient\_light[1] -= 0.1f;          ambient\_light[2] -= 0.1f;          if (ambient\_light[0] < 0.0f) ambient\_light[0] = 0.0f; // Membatasi nilai minimum          if (ambient\_light[1] < 0.0f) ambient\_light[1] = 0.0f;          if (ambient\_light[2] < 0.0f) ambient\_light[2] = 0.0f;          glLightfv(GL\_LIGHT0, GL\_AMBIENT, ambient\_light); // Mengatur cahaya ambient          glutPostRedisplay(); // Meminta untuk menggambar ulang          break;      case 's': // Jika tombol 's' ditekan          diffuse\_light[0] += 0.1f; // Meningkatkan intensitas cahaya difus          diffuse\_light[1] += 0.1f;          diffuse\_light[2] += 0.1f;          if (diffuse\_light[0] > 1.0f) diffuse\_light[0] = 1.0f; // Membatasi nilai maksimum          if (diffuse\_light[1] > 1.0f) diffuse\_light[1] = 1.0f;          if (diffuse\_light[2] > 1.0f) diffuse\_light[2] = 1.0f;          glLightfv(GL\_LIGHT0, GL\_DIFFUSE, diffuse\_light); // Mengatur cahaya difus          glutPostRedisplay(); // Meminta untuk menggambar ulang          break;      case 'f': // Jika tombol 'f' ditekan          diffuse\_light[0] -= 0.1f; // Mengurangi intensitas cahaya difus          diffuse\_light[1] -= 0.1f;          diffuse\_light[2] -= 0.1f;          if (diffuse\_light[0] < 0.0f) diffuse\_light[0] = 0.0f; // Membatasi nilai minimum          if (diffuse\_light[1] < 0.0f) diffuse\_light[1] = 0.0f;          if (diffuse\_light[2] < 0.0f) diffuse\_light[2] = 0.0f;          glLightfv(GL\_LIGHT0, GL\_DIFFUSE, diffuse\_light); // Mengatur cahaya difus          glutPostRedisplay(); // Meminta untuk menggambar ulang          break;      case 'r': // Jika tombol 'r' ditekan          xPOS += 0.1; // Memperbesar skala sumbu X          break;      case 't': // Jika tombol 't' ditekan          xPOS -= 0.1; // Memperkecil skala sumbu X          break;      case 'y': // Jika tombol 'y' ditekan          yPOS += 0.1; // Memperbesar skala sumbu Y ```cpp          break;      case 'u': // Jika tombol 'u' ditekan          yPOS -= 0.1; // Memperkecil skala sumbu Y          break;      case 'i': // Jika tombol 'i' ditekan          angle += 5.0; // Rotasi 5 derajat ke kanan          break;      case 'o': // Jika tombol 'o' ditekan          angle -= 5.0; // Rotasi 5 derajat ke kiri          break;      case 'p': // Jika tombol 'p' ditekan          translateY += 1.0; // Translasi 1 unit ke atas          break;      case 'k': // Jika tombol 'k' ditekan          translateY -= 1.0; // Translasi 1 unit ke bawah          break;      case 'n': // Jika tombol 'n' ditekan          translateX -= 1.0; // Translasi 1 unit ke kiri          break;      case 'm': // Jika tombol 'm' ditekan          translateX += 1.0; // Translasi 1 unit ke kanan          break;      case 'c': // Jika tombol 'c' ditekan          hidden = !hidden; // Mengubah status untuk menampilkan atau menyembunyikan sumbu koordinat          break;      }      glutPostRedisplay(); // Meminta untuk menggambar ulang  }  // Fungsi untuk menangani interaksi mouse  void mouse(int button, int state, int x, int y) {      if (button == GLUT\_LEFT\_BUTTON && state == GLUT\_DOWN) { // Jika tombol kiri mouse ditekan          mouseDown = true; // Menandai bahwa mouse sedang ditekan          xdiff = x - yrot; // Menghitung selisih posisi mouse di sumbu X          ydiff = y - xrot; // Menghitung selisih posisi mouse di sumbu Y      } else {          mouseDown = false; // Menandai bahwa mouse tidak ditekan      }  }  // Fungsi untuk menangani gerakan mouse  void mouseMotion(int x, int y) {      if (mouseDown) { // Jika mouse sedang ditekan          yrot = x - xdiff; // Mengatur rotasi di sumbu Y berdasarkan gerakan mouse          xrot = y - ydiff; // Mengatur rotasi di sumbu X berdasarkan gerakan mouse          glutPostRedisplay(); // Meminta untuk menggambar ulang      }  }  // Fungsi untuk inisialisasi pengaturan OpenGL  void inisialisasi() {      glEnable(GL\_DEPTH\_TEST); // Mengaktifkan pengujian kedalaman      glEnable(GL\_COLOR\_MATERIAL); // Mengaktifkan material warna      glClearColor(0.1, 0.1, 0.1, 1.0); // Mengatur warna latar belakang      glEnable(GL\_LIGHTING); // Mengaktifkan pencahayaan      glEnable(GL\_LIGHT0); // Mengaktifkan sumber cahaya 0      glLightfv(GL\_LIGHT0, GL\_POSITION, light\_position); // Mengatur posisi cahaya      glLightfv(GL\_LIGHT0, GL\_AMBIENT, ambient\_light); // Mengatur cahaya ambient      glLightfv(GL\_LIGHT0, GL\_DIFFUSE, diffuse\_light); // Mengatur cahaya difus  }  void DrawCube(float x, float y, float z){      glPushMatrix();      glScalef(x,y,z);      glutSolidCube(1.0);      glPopMatrix();  }  void drawBuilding() {   //==========================MENGGAMBAR GEDUNG==============================      glColor3f(0.53f, 0.81f, 0.92f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(0.0, -10.0, -5.0); // Posisi dasar gedung      glScalef(70.0, 80.0, 50.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //==========================================================================  //==========================MENGGAMBAR ATAP GEDUNG==============================      glColor3f(1.0f, 1.0f, 1.0f);// Mengatur warna atap      glPushMatrix();      glTranslatef(0.0, 35.0, -5.0); // Posisi atap      glScalef(70.0, 10.0, 50.0); // Skala atap      glutSolidCube(1.0); // Menggambar atap      glPopMatrix();  //==========================MENGGAMBAR JENDELA==============================      //menggambar jendela atas kiri      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-23.0, 20.0, 2.0); // Posisi dasar gedung      glScalef(7.0, 10.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //menggambar jendela tengah kiri      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-23.0, 0.0, 2.0); // Posisi dasar gedung      glScalef(7.0, 10.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //menggambar jendela bawah kiri      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-23.0, -20.0, 2.0); // Posisi dasar gedung      glScalef(7.0, 10.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //========================================================      //menggambar jendela tengah kiri 1 atas      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-9.0, 20.0, 2.0); // Posisi dasar gedung      glScalef(7.0, 10.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //menggambar jendela tengah kiri 1 tengah      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-9.0, 0.0, 2.0); // Posisi dasar gedung      glScalef(7.0, 10.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //menggambar jendela tengah kiri 1 Tengah      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-9.0, -20.0, 2.0); // Posisi dasar gedung      glScalef(7.0, 10.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //========================================================  //menggambar jendela tengah bawah kanan      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(6.0, -20.0, 2.0); // Posisi dasar gedung      glScalef(7.0, 10.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //menggambar jendela tengah kanan      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(6.0, 0.0, 2.0); // Posisi dasar gedung      glScalef(7.0, 10.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //menggambar jendela tengah atas kanan      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(6.0, 20.0, 2.0); // Posisi dasar gedung      glScalef(7.0, 10.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //========================================================  //menggambar jendela atas kanan 2      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(22.0, 20.0, 2.0); // Posisi dasar gedung      glScalef(7.0, 10.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //menggambar jendela tengah kanan 2      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(22.0, 0.0, 2.0); // Posisi dasar gedung      glScalef(7.0, 10.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //menggambar jendela tengah kanan 2      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(22.0, -20.0, 2.0); // Posisi dasar gedung      glScalef(7.0, 10.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //=========================================================================  //========================PINTU GEDUNG====================================      //menggambar pintu gedung      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(0.0, -42.0, 1.0); // Posisi dasar gedung      glScalef(25.0, 15.0, 39.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //=========================================================================  //========================PEMISAH PINTU====================================      //pemisah pintu tengah      glColor3f(0.0f, 0.0f, 0.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(0.0, -42.0, 1.0); // Posisi dasar gedung      glScalef(2.0, 15.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //pemisah kiri      glColor3f(0.0f, 0.0f, 0.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-13.0, -42.0, 1.0); // Posisi dasar gedung      glScalef(2.0, 15.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //pemisah pintu kanan      glColor3f(0.0f, 0.0f, 0.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(13.0, -42.0, 1.0); // Posisi dasar gedung      glScalef(2.0, 15.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //pemisah pintu atas      glColor3f(0.0f, 0.0f, 0.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(0.0, -35.0, 1.0); // Posisi dasar gedung      glScalef(25.0, 2.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //=========================================================================  //========================GAGANG PINTU=======================================      //gagang pintu kanan      glColor3f(0.0f, 0.0f, 0.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(3.0, -44.0, 1.0); // Posisi dasar gedung      glScalef(2.0, 2.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //gagang pintu kiri      glColor3f(0.0f, 0.0f, 0.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-3.0, -44.0, 1.0); // Posisi dasar gedung      glScalef(2.0, 2.0, 40.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //=========================================================================  //========================HALAMAN==========================================      //menggambar halaman      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-5.0, -60.0, 2.0); // Posisi dasar gedung      glScalef(180.0, 10.0, 160.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //=========================================================================  //========================LANTAI GEDUNG====================================      //menggambar lantai gedung      glColor3f(0.0f, 0.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-5.0, -52.0, 2.0); // Posisi dasar gedung      glScalef(120.0, 3.0, 100.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //=========================================================================  //======================MEMBUAT JALAN =====================================      //jalan 1      glColor3f(0.0f, 0.0F, 0.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-5.0, -55.0, 66.0); // Posisi dasar gedung      glScalef(180.0, 3.0, 31.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //jalan 2      glColor3f(0.0f, 0.0F, 0.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(65.0, -55.0, 1.0); // Posisi dasar gedung      glScalef(30.0, 3.0, 160.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //jalan 3      glColor3f(0.0f, 0.0F, 0.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-5.0, -55.0, -63.0); // Posisi dasar gedung      glScalef(170.0, 3.0, 31.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //jalan 4      glColor3f(0.0f, 0.0F, 0.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-80.0, -55.0, 1.0); // Posisi dasar gedung      glScalef(30.0, 3.0, 160.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //===========================================================  //======================GARIS JALAN=================================      //garis jalan 1      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-85.0, -54.0, 70.0); // Posisi dasar gedung      glScalef(20.0, 3.0, 5.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 2      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-55.0, -54.0, 70.0); // Posisi dasar gedung      glScalef(15.0, 3.0, 5.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 3      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-30.0, -54.0, 70.0); // Posisi dasar gedung      glScalef(15.0, 3.0, 5.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 4      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-5.0, -54.0, 70.0); // Posisi dasar gedung      glScalef(15.0, 3.0, 5.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 5      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(20.0, -54.0, 70.0); // Posisi dasar gedung      glScalef(15.0, 3.0, 5.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 6      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(45.0, -54.0, 70.0); // Posisi dasar gedung      glScalef(15.0, 3.0, 5.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 7      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(65.0, -54.0, 70.0); // Posisi dasar gedung      glScalef(10.0, 3.0, 5.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 8      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(70.0, -54.0, 65.0); // Posisi dasar gedung      glScalef(5.0, 3.0, 15.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 9      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(70.0, -54.0, 43.0); // Posisi dasar gedung      glScalef(5.0, 3.0, 14.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 10      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(70.0, -54.0, 20.0); // Posisi dasar gedung      glScalef(5.0, 3.0, 15.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 11      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(70.0, -54.0, -3.0); // Posisi dasar gedung      glScalef(5.0, 3.0, 15.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 12      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(70.0, -54.0, -26.0); // Posisi dasar gedung      glScalef(5.0, 3.0, 15.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 13      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(70.0, -54.0, -54.0); // Posisi dasar gedung      glScalef(5.0, 3.0, 20.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 14      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(66.0, -54.0, -64.0); // Posisi dasar gedung      glScalef(12.0, 3.0, 5.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 15      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(46.0, -54.0, -64.0); // Posisi dasar gedung      glScalef(15.0, 3.0, 5.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();        //garis jalan 16      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(20.0, -54.0, -64.0); // Posisi dasar gedung      glScalef(15.0, 3.0, 5.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 17      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-5.0, -54.0, -64.0); // Posisi dasar gedung      glScalef(15.0, 3.0, 5.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 18      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-30.0, -54.0, -64.0); // Posisi dasar gedung      glScalef(15.0, 3.0, 5.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 19      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-55.0, -54.0, -64.0); // Posisi dasar gedung      glScalef(15.0, 3.0, 5.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 20      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-78.0, -54.0, -64.0); // Posisi dasar gedung      glScalef(10.0, 3.0, 5.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 21      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-80.0, -54.0, 20.0); // Posisi dasar gedung      glScalef(5.0, 3.0, 15.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();       //garis jalan 22      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-80.0, -54.0, 65.0); // Posisi dasar gedung      glScalef(5.0, 3.0, 13.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 23      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-80.0, -54.0, 43.0); // Posisi dasar gedung      glScalef(5.0, 3.0, 15.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 24      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-80.0, -54.0, -3.0); // Posisi dasar gedung      glScalef(5.0, 3.0, 15.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 25      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-80.0, -54.0, -26.0); // Posisi dasar gedung      glScalef(5.0, 3.0, 15.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //garis jalan 26      glColor3f(1.0f, 1.0F, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-80.0, -54.0, -54.0); // Posisi dasar gedung      glScalef(5.0, 3.0, 20.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //============================================================  //======================Lobby=================================      //paragi ngiuhan      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(0.0, -30.0, 25.0); // Posisi dasar gedung      glScalef(35.0, 5.0, 30.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //tiang 1      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(15.0, -40.0, 38.0); // Posisi dasar gedung      glScalef(2.0, 25.0, 2.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //tiang 2      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-15.0, -40.0, 38.0); // Posisi dasar gedung      glScalef(2.0, 25.0, 2.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();  //=======================================================  //======================benteng=================================      //benteng 1      glColor3f(0.53f, 0.81f, 0.92f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(85.0, -25.0, 1.0); // Posisi dasar gedung      glScalef(10.0, 80.0, 170.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //benteng 2      glColor3f(0.53f, 0.81f, 0.92f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-3.0, -25.0, -80.0); // Posisi dasar gedung      glScalef(185.0, 80.0, 10.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //benteng 3      glColor3f(0.53f, 0.81f, 0.92f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-100.0, -25.0, -60.0); // Posisi dasar gedung      glScalef(10.0, 80.0, 50.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //benteng 4      glColor3f(0.53f, 0.81f, 0.92f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-8.0, -25.0, 85.0); // Posisi dasar gedung      glScalef(195.0, 80.0, 10.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();      //benteng 5      glColor3f(0.53f, 0.81f, 0.92f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-100.0, -25.0, 25.0); // Posisi dasar gedung      glScalef(10.0, 80.0, 20.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();        //benteng 6      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-100.0, 10.0, -10.0); // Posisi dasar gedung      glScalef(10.0, 10.0, 50.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();       //benteng 7      glColor3f(1.0f, 1.0f, 1.0f); // Mengatur warna gedung      glPushMatrix();      glTranslatef(-100.0, 10.0, 60.0); // Posisi dasar gedung      glScalef(10.0, 10.0, 50.0); // Skala gedung      glutSolidCube(1.0); // Menggambar gedung      glPopMatrix();    }  // Fungsi untuk menggambar tampilan  void tampilan(){      glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT); // Menghapus buffer warna dan kedalaman      glLoadIdentity(); // Mengatur matriks model ke identitas      // Translasi      glTranslatef(translateX, translateY, 0.0); // Menggeser objek      // Rotasi      glRotatef(angle, 0.0, 0.0, 1.0); // Rotasi di sekitar sumbu Z      // Skala      glScalef(xPOS, yPOS, 1.0); // Mengatur skala objek      gluLookAt(0, 0, 4, //posisi kamera                0, 0, 0, // titik sudut pandang                0, 1, 0); // Mengatur posisi kamera      glRotatef(xrot, 1, 0, 0); // Rotasi berdasarkan input mouse di sumbu X      glRotatef(yrot, 0, 1, 0); // Rotasi berdasarkan input mouse di sumbu Y      glPushMatrix(); // Menyimpan matriks saat ini      hiddenCarte(); // Menggambar sumbu koordinat jika diperlukan  //=====================================================  // Menggambar kursi 1 taman  glBegin(GL\_POLYGON); // Sandaran kursi  glColor3f(1.0f, 0.5f, 0.0f); // Mengatur warna coklat  glVertex3f(-40, -45, 15);  glVertex3f(-50, -45, 15);  glVertex3f(-50, -50, 15);  glVertex3f(-40, -50, 15);  glEnd();  glBegin(GL\_POLYGON); // Alas kursi/tempat duduk  glColor3f(1.0f, 0.5f, 0.0f); // Mengatur warna coklat  glVertex3f(-40, -47, 15);  glVertex3f(-50, -47, 15);  glVertex3f(-50, -47, 12);  glVertex3f(-40, -47, 12);  glEnd();  glBegin(GL\_POLYGON); // Kaki-kaki kursi  glColor3f(1.0f, 0.5f, 0.0f); // Mengatur warna coklat  glVertex3f(-40, -47, 12);  glVertex3f(-50, -47, 12);  glVertex3f(-50, -50, 12);  glVertex3f(-40, -50, 12);  glEnd();  // Menggambar kursi 2  glBegin(GL\_POLYGON); // Sandaran kursi  glColor3f(1.0f, 0.5f, 0.0f); // Mengatur warna coklat  glVertex3f(-40, -45, 0);  glVertex3f(-50, -45, 0);  glVertex3f(-50, -50, 0);  glVertex3f(-40, -50, 0);  glEnd();  glBegin(GL\_POLYGON); // Alas meja  glColor3f(1.0f, 0.5f, 0.0f); // Mengatur warna coklat  glVertex3f(-40, -47, 3);  glVertex3f(-50, -47, 3);  glVertex3f(-50, -50, 3);  glVertex3f(-40, -50, 3);  glEnd();  glBegin(GL\_POLYGON); // Kaki-kaki meja  glColor3f(1.0f, 0.5f, 0.0f); // Mengatur warna coklat  glVertex3f(-40, -47, 3);  glVertex3f(-50, -47, 3);  glVertex3f(-50, -47, 0);  glVertex3f(-40, -47, 0);  glEnd();  // Menggambar meja taman samping gedung  glBegin(GL\_POLYGON);  glColor3f(1.0f, 0.5f, 0.0f); // Mengatur warna coklat  glVertex3f(-40, -47, 10);  glVertex3f(-50, -47, 10);  glVertex3f(-50, -50, 10);  glVertex3f(-40, -50, 10);  glEnd();  glBegin(GL\_POLYGON); // Kiri meja  glColor3f(1.0f, 0.5f, 0.0f); // Mengatur warna coklat  glVertex3f(-40, -47, 5);  glVertex3f(-50, -47, 5);  glVertex3f(-50, -50, 5);  glVertex3f(-40, -50, 5);  glEnd();  glBegin(GL\_POLYGON); // Alas meja  glColor3f(1.0f, 0.5f, 0.0f); // Mengatur warna coklat  glVertex3f(-40, -47, 5);  glVertex3f(-50, -47, 5);  glVertex3f(-50, -47, 10);  glVertex3f(-40, -47, 10);  glEnd();  drawBuilding();  glPushMatrix(); // Menyimpan matriks saat ini  glPopMatrix(); // Mengembalikan matriks yang disimpan  glutSwapBuffers(); // Menukar buffer untuk menampilkan hasil  }  // Fungsi utama  int main(int argc, char \*\*argv){      glutInit(&argc, argv); // Inisialisasi GLUT      glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB); // Mengatur mode tampilan      glutInitWindowPosition(260, 20); // Mengatur posisi jendela      glutInitWindowSize(800, 650); // Mengatur ukuran jendela      glutCreateWindow("GEDUNG 3D"); // Membuat jendela dengan judul      myinit(); // Memanggil fungsi inisialisasi      glutDisplayFunc(tampilan); // Mengatur fungsi tampilan      glutKeyboardFunc(kunci); // Mengatur fungsi keyboard      glutMouseFunc(mouse); // Mengatur fungsi mouse      glutMotionFunc(mouseMotion); // Mengatur fungsi gerakan mouse      glutReshapeFunc(ukur); // Mengatur fungsi perubahan ukuran jendela      inisialisasi(); // Memanggil fungsi inisialisasi OpenGL      glutMainLoop(); // Memasuki loop utama GLUT      drawBuilding();  } |

## Output



**Gambar 3 Output Program**

## Penjelasan

1. **Library yang Diimpor**

#include <GL/freeglut.h>

#include <math.h>

#include <GL/freeglut.h>

Mengimpor library FreeGLUT, yang merupakan toolkit untuk membuat aplikasi OpenGL. Library ini menyediakan fungsi untuk mengelola jendela, input, dan rendering.

#include <math.h>:

Mengimpor library matematika standar C, yang menyediakan fungsi matematika seperti trigonometri, eksponensial, dan lainnya.

1. **Deklarasi Variabel Global**

float xrot = 0; // Rotasi di sumbu X

float yrot = 0; // Rotasi di sumbu Y

float xdiff = 0; // Selisih posisi mouse di sumbu X

float ydiff = 0; // Selisih posisi mouse di sumbu Y

bool mouseDown = false; // Status mouse (apakah ditekan atau tidak)

Variabel ini digunakan untuk menyimpan informasi tentang rotasi objek berdasarkan input mouse dan status mouse.

1. **Variabel Pencahayaan**

GLfloat light\_position[] = { 0.0, 20.0, 70.0, 1.0 };

GLfloat ambient\_light[] = { 0.2, 0.2, 0.2, 1.0 };

GLfloat diffuse\_light[] = { 1.0, 1.0, 1.0, 1.0 };

bool lightingEnabled = true; // Status pencahayaan

Variabel ini digunakan untuk mengatur posisi dan intensitas cahaya dalam scene. light\_position menentukan posisi sumber cahaya, sedangkan ambient\_light dan diffuse\_light mengatur intensitas cahaya ambient dan difus.

1. **Variabel Transformasi Objek**

float xPOS = 1.0; // Skala sumbu X

float yPOS = 1.0; // Skala sumbu Y

float angle = 0.0; // Sudut rotasi

float translateX = 0.0; // Posisi translasi sumbu X

float translateY = 0.0; // Posisi translasi sumbu Y

bool hidden = false; // Status untuk menampilkan atau menyembunyikan sumbu koordinat

Variabel ini digunakan untuk mengatur transformasi objek, termasuk skala, rotasi, dan translasi.

1. **Deklarasi Fungsi**

void hiddenCarte();

Deklarasi fungsi hiddenCarte() yang akan digunakan untuk menggambar sumbu koordinat.

1. **Fungsi ukur(int lebar, int tinggi)**

void ukur(int lebar, int tinggi) {

if (tinggi == 0) tinggi = 1; // Menghindari pembagian dengan nol

glMatrixMode(GL\_PROJECTION); // Mengatur mode matriks proyeksi

glLoadIdentity(); // Mengatur matriks proyeksi ke identitas

gluPerspective(45, lebar / tinggi, 5, 450); // Mengatur perspektif

glTranslatef(10, 0, -340); // Menggeser pandangan ke belakang

glMatrixMode(GL\_MODELVIEW); // Kembali ke mode matriks model

}

Fungsi ini mengatur ukuran jendela dan proyeksi perspektif. Jika tinggi jendela adalah 0, maka diatur menjadi 1 untuk menghindari pembagian dengan nol. Menggunakan gluPerspective() untuk mengatur sudut pandang.

1. **Fungsi drawCartecius()**

void drawCartecius() {

glColor3f(1.0, 0.0, 0.0); // Mengatur warna garis menjadi merah

glBegin(GL\_LINES); // Memulai menggambar garis // Garis sumbu X

glVertex3f(-100.0, 0.0, 0.0);

glVertex3f(100.0, 0.0, 0.0);

// Garis sumbu Y

glVertex3f(0.0, -100.0, 0.0);

glVertex3f(0.0, 100.0, 0.0);

// Garis sumbu Z

glVertex3f(0.0, 0.0, -100.0);

glVertex3f(0.0, 0.0, 100.0);

glEnd(); // Mengakhiri menggambar garis

}

Fungsi ini menggambar sumbu koordinat (X, Y, Z) dengan menggunakan garis. Warna garis diatur menjadi merah, dan setiap sumbu digambar menggunakan glVertex3f().

1. **Fungsi hiddenCarte()**

void hiddenCarte() {

if (hidden) {

drawCartecius(); // Memanggil fungsi untuk menggambar sumbu koordinat jika hidden bernilai true

}

}

Fungsi ini memanggil drawCartecius() untuk menggambar sumbu koordinat jika variabel hidden bernilai true.

1. **Fungsi myinit(void)**

void myinit(void) {

glClearColor(0.0, 0.0, 0.0, 0.0); // Mengatur warna latar belakang

glMatrixMode(GL\_PROJECTION); // Mengatur mode matriks proyeksi

glEnable(GL\_DEPTH\_TEST); // Mengaktifkan pengujian kedalaman

glMatrixMode(GL\_MODELVIEW); // Kembali ke mode matriks model

glPointSize(10.0); // Mengatur ukuran titik

glLineWidth(7.0f); // Mengatur lebar garis

}

Fungsi ini digunakan untuk inisialisasi pengaturan OpenGL, termasuk mengatur warna latar belakang, mengaktifkan pengujian kedalaman, dan mengatur ukuran titik serta lebar garis.

1. **Fungsi idle()**

void idle() {

if (mouseDown) {

xrot += ydiff; // Menambah rotasi di sumbu X

yrot += xdiff; // Menambah rotasi di sumbu Y

}

glutPostRedisplay(); // Meminta OpenGL untuk menggambar ulang tampilan

}

Fungsi ini dipanggil saat tidak ada interaksi, untuk memperbarui tampilan berdasarkan rotasi yang ditentukan oleh gerakan mouse**.**

1. **Fungsi kunci(unsigned char key, int x, int y)**

void kunci(unsigned char key, int x, int y) {

switch (key) {

case 27: // Jika tombol ESC ditekan

exit(0); // Keluar dari program

break;

case 'l': // Mengubah status pencahayaan

lightingEnabled = !lightingEnabled;

break;

// Tambahkan lebih banyak kasus untuk tombol lainnya

}

}

Fungsi ini menangani input dari keyboard. Jika tombol ESC ditekan, program akan keluar. Tombol lain dapat digunakan untuk mengubah status pencahayaan atau melakukan tindakan lainnya.

1. **Fungsi mouse(int button, int state, int x, int y)**

void mouse(int button, int state, int x, int y) {

if (button == GLUT\_LEFT\_BUTTON) {

if (state == GLUT\_DOWN) {

mouseDown = true; // Menandai bahwa mouse sedang ditekan

xdiff = x - xrot; // Menghitung selisih posisi mouse

ydiff = y - yrot;

} else {

mouseDown = false; // Menandai bahwa mouse tidak ditekan

}

}

}

Fungsi ini menangani interaksi mouse. Jika tombol kiri mouse ditekan, status mouseDown diatur menjadi true, dan selisih posisi mouse dihitung untuk rotasi.

1. **Fungsi mouseMotion(int x, int y)**

void mouseMotion(int x, int y) {

if (mouseDown) {

xrot = y - ydiff; // Mengatur rotasi objek berdasarkan gerakan mouse

yrot = x - xdiff;

}

}

Fungsi ini menangani gerakan mouse. Jika mouse sedang ditekan, rotasi objek diatur berdasarkan perubahan posisi mouse.

1. **Fungsi inisialisasi()**

void inisialisasi() {

glEnable(GL\_DEPTH\_TEST); // Mengaktifkan pengujian kedalaman

// Mengatur material warna dan pencahayaan glLightfv(GL\_LIGHT0, GL\_POSITION, light\_position); // Mengatur posisi cahaya

glLightfv(GL\_LIGHT0, GL\_AMBIENT, ambient\_light); // Mengatur cahaya ambient

glLightfv(GL\_LIGHT0, GL\_DIFFUSE, diffuse\_light); // Mengatur cahaya difus

glEnable(GL\_LIGHT0); // Mengaktifkan sumber cahaya

}

Fungsi ini digunakan untuk inisialisasi pengaturan pencahayaan dan pengujian kedalaman dalam OpenGL. Mengatur posisi dan intensitas cahaya serta mengaktifkan sumber cahaya.

1. **Fungsi DrawCube(float x, float y, float z)**

void DrawCube(float x, float y, float z) {

glTranslatef(x, y, z); // Mengatur posisi kubus

glBegin(GL\_QUADS); // Memulai menggambar kubus

// Menggambar sisi-sisi kubus dengan warna yang berbeda

glColor3f(1.0, 0.0, 0.0); // Merah

glVertex3f(-1.0, -1.0, -1.0);

glVertex3f(1.0, -1.0, -1.0);

glVertex3f(1.0, 1.0, -1.0);

glVertex3f(-1.0, 1.0, -1.0);

// Tambahkan sisi lainnya dengan warna dan koordinat yang sesuai

glEnd(); // Mengakhiri menggambar kubus

}

Fungsi ini menggambar kubus di posisi yang ditentukan. Menggunakan glTranslatef() untuk mengatur posisi kubus dan glBegin(GL\_QUADS) untuk mendefinisikan setiap sisi kubus.

1. **Fungsi display()**

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT); // Menghapus buffer warna dan kedalaman

glLoadIdentity(); // Mengatur matriks model ke identitas

glRotatef(xrot, 1.0, 0.0, 0.0); // Rotasi di sumbu X

glRotatef(yrot, 0.0, 1.0, 0.0); // Rotasi di sumbu Y

inisialisasi(); // Memanggil fungsi inisialisasi

DrawCube(0.0, 0.0, 0.0); // Menggambar kubus di pusat

hiddenCarte(); // Menggambar sumbu koordinat jika diperlukan

glutSwapBuffers(); // Menampilkan buffer yang telah digambar

}

Fungsi ini bertanggung jawab untuk menggambar semua objek di jendela. Menghapus buffer, mengatur rotasi objek, dan memanggil fungsi untuk menggambar kubus dan sumbu koordinat.

1. **Fungsi main(int argc, char\*\* argv)**

int main(int argc, char\*\* argv) {

glutInit(&argc, argv); // Menginisialisasi GLUT

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB | GLUT\_DEPTH); // Mengatur mode tampilan

glutInitWindowSize(800, 600); // Mengatur ukuran jendela

glutCreateWindow("OpenGL Cube"); // Membuat jendela

glutDisplayFunc(display); // Mengatur fungsi callback untuk menggambar

glutIdleFunc(idle); // Mengatur fungsi callback untuk idle

glutKeyboardFunc(kunci); // Mengatur fungsi callback untuk keyboard

glutMouseFunc(mouse); // Mengatur fungsi callback untuk mouse

glutMotionFunc(mouseMotion); // Mengatur fungsi callback untuk gerakan mouse

myinit(); // Memanggil fungsi inisialisasi

glutMainLoop(); // Memulai loop utama GLUT

return 0; // Mengembalikan nilai 0

}

Fungsi utama yang menjalankan program. Menginisialisasi GLUT, mengatur mode tampilan, ukuran jendela, dan fungsi callback untuk menggambar, idle, dan input. Memulai loop utama GLUT untuk menjalankan aplikasi.

# BAB IV Kesimpulan

Program di atas adalah aplikasi grafis 3D yang menggunakan OpenGL dan FreeGLUT untuk menggambar sebuah scene yang mencakup sebuah gedung, sumbu koordinat, dan elemen-elemen lainnya seperti jalan dan furnitur. Dengan memanfaatkan berbagai fungsi OpenGL, program ini memungkinkan pengguna untuk berinteraksi dengan scene melalui input mouse dan keyboard, termasuk rotasi objek, translasi, dan pengaturan pencahayaan. Pencahayaan diatur dengan menggunakan sumber cahaya ambient dan diffuse, yang memberikan efek visual yang lebih menarik pada objek yang digambar. Selain itu, program ini juga mengimplementasikan pengujian kedalaman untuk memastikan bahwa objek yang lebih dekat dengan kamera ditampilkan dengan benar di atas objek yang lebih jauh. Dengan struktur yang terorganisir dan penggunaan fungsi-fungsi yang modular, program ini memberikan dasar yang baik untuk pengembangan aplikasi grafis 3D yang lebih kompleks.

# DAFTAR PUSTAKA

[1] H. Herriyance, C. Al Vareel, E. Ewaldo, R. F. Hutabarat, and S. H. Gulo, “Pembuatan Animasi Integral Sebagai Implementasi Grafika Komputer Dengan Menggunakan Morphing dan OpenGL,” *J-SISKO TECH (Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD)*, vol. 5, no. 2, p. 269, 2022, doi: 10.53513/jsk.v5i2.5747.

[2] \* 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,” *Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD*, vol. 1, no. 2, pp. 96–105, 2018.

[3] A. Gunawan, A. Z. Falani, S. Literatur, M. Sistem, and G. Komputer, “Implementasi Opengl ( Open Graphics Library ) Untuk,” *Implementasi Opengl (Open Graphics Library) Untuk Pembuatan Bangun Datar Sebagai Media Pembelajaran*, vol. 5, no. April, pp. 78–86, 2022.

