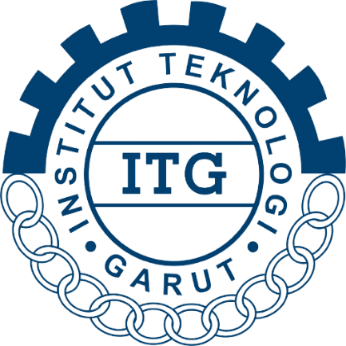
**LAPORAN PRATIKUM GRAFIK KOMPUTER**

Diajukan untuk memenuhi Tugas mata kuliah Pratikum Grafik Komputer

**PLANET**

Dosen Pengampu : Sri Rahayu, M.Kom

Instruktur Pratikum : Arul Budi Kalimat, S.Kom



Disusun oleh

Kelompok : 4

Jujun Munawar

2306148

M Jafar Sopian

2306160

Pitri Fatonah

2306150

**PROGRAM STUDI TEKNIK INFORMATIKA JURUSAN ILMU KOMPUTER**

**INSTITUT TEKNOLOGI GARUT 2025**

# KATA PENGANTAR

Puji dan syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga kami dapat menyelesaikan Laporan Praktikum Grafik Komputer ini. Laporan ini dibuat sebagai salah satu tugas dari mata kuliah Jaringan Komputer, dengan tujuan untuk memberikan pemahaman yang lebih baik tentang Materi Open GL ini.

Kami mengucapkan terima kasih kepada dosen pengampu Sri Rahayu, M.Kom, instruktur praktikum Arul Budi Kalimat, S.Kom, para asisten pembimbing, Dzulkifli P.S dan rekan lainnya yang telah meluangkan waktu serta memberikan bimbingan hingga laporan ini terselesaikan, serta semua pihak yang telah mendukung 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, 13 Januari 2025

Penulis

# DAFTAR ISI

[KATA PENGANTAR i](#_Toc187778527)

[DAFTAR ISI ii](#_Toc187778528)

[DAFTAR GAMBAR iii](#_Toc187778529)

[BAB I PENDAHULUAN 1](#_Toc187778530)

[1.1 Latar Belakang 1](#_Toc187778531)

[1.2 Rumusan Masalah 1](#_Toc187778532)

[1.3 Tujuan 1](#_Toc187778533)

[BAB II TINJAUAN PUSTAKA 2](#_Toc187778534)

[2.1 OpenGL 2](#_Toc187778535)

[2.2 Konfigurasi OpenGL pada Dev C++ 2](#_Toc187778536)

[2.3 Cara Kerja OpenGL 5](#_Toc187778537)

[2.4 PLANET Di OpenGL 5](#_Toc187778538)

[BAB III HASIL 6](#_Toc187778539)

[3.1 Source Code 6](#_Toc187778540)

[3.2 Output 12](#_Toc187778541)

[3.3 Penjelasan 13](#_Toc187778542)

[BAB IV 14](#_Toc187778543)

[4.1. Kesimpulan 14](#_Toc187778544)

[DAFTAR PUSTAKA 15](#_Toc187778545)

# DAFTAR GAMBAR

[Gambar 1 Open GL 2](#_Toc187778492)

[Gambar 2 Tampilan awal 2](#_Toc187778493)

[Gambar 3 Lembaran baru 3](#_Toc187778494)

[Gambar 4 Project 3](#_Toc187778495)

[Gambar 5 Pemberian nama file 4](#_Toc187778496)

[Gambar 6 Konfigurasi 4](#_Toc187778497)

[Gambar 7 Parameter 5](#_Toc187778498)

[Gambar 8 Tampilan awal 12](#_Toc187778499)

[Gambar 9 Dari atas 13](#_Toc187778500)

[Gambar 10 Dari samping 13](#_Toc187778501)

# BAB I PENDAHULUAN

## Latar Belakang

Grafika Komputer adalah suatu bidang yang mempelajari bagaimana menghasilkan suatu gambar menggunakan komputer, sehingga Grafika Komputer membahas teknik-teknik menggambar.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 displayhardware (Graphics System) yaitu OpenGL.[1]

Dalam OpenGL, kita bebas mengekspresikan apa saja yang ingin kita buat. Sebagai contoh, dalam tugas besar ini, kami membuat planet. Tidak ada keraguan selama kita memahami dasar-dasar dengan baik[2].Praktikum grafik komputer sangat penting, terutama sebagai pengantar ke bidang-bidang lain dalam desain, misalnya. Hal-hal mendasar yang dipelajari saat ini mungkin akan menjadi lebih mudah di masa depan berkat perkembangan teknologi. Namun, perlu ada acuan yang jelas agar dasar-dasar teorinya tetap terjaga dan tidak terabaikan.[3]

## Rumusan Masalah

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

## Tujuan

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

# BAB II TINJAUAN PUSTAKA

## OpenGL

A blue sign with white text

Description automatically generatedOpenGL, atau Open Graphics Library, adalah antarmuka pemrograman aplikasi (API) yang banyak digunakan yang dirancang untuk merender grafik 2D dan 3D. Ini berfungsi sebagai alat penting dalam berbagai bidang, khususnya dalam pengembangan game, desain berbantuan komputer (CAD), dan visualisasi ilmiah.

Gambar 1 Open GL

## Konfigurasi OpenGL pada Dev C++

Langkah-langakah Konfigurasi Open GL :

1. A screenshot of a computer

   Description automatically generatedBuka terlebih dahulu aplikasi Dev c++.

Gambar 2 Tampilan awal

1. A screenshot of a computer

   Description automatically generatedKlik file -> pilih New -> Project.

Gambar 3 Lembaran baru

1. A screenshot of a computer

   Description automatically generatedSetelah itu pilih Console Application lalu ok.

Gambar 4 Project

1. Ubah nama file sesuai yang di ingin kan contoh: “test”.

A screenshot of a computer

Description automatically generated

Gambar 5 Pemberian nama file

1. A computer screen shot of a black screen

   Description automatically generatedSetelah tampilan seperti ini pilih pada tanda panah yang di tunjuk.

Gambar 6 Konfigurasi

1. A screenshot of a computer

   Description automatically generatedDhPilih parameter dan ketik parameter seperti di bawah agar open Gl dapat berjalan di Dev c++.

Gambar 7 Parameter

## Cara Kerja OpenGL

Sebelum menggunakan OpenGL, langkah pertama adalah inisialisasi, termasuk pengaturan viewport dengan glViewport dan perspektif dengan gluPerspective. Objek digambar menggunakan primitif dasar seperti titik, garis, dan poligon dengan glBegin dan glEnd. Transformasi dilakukan menggunakan fungsi seperti glTranslatef, glRotatef, dan glScalef untuk memindahkan, memutar, atau mengubah ukuran objek. Setelah itu, tahap rasterisasi mengubah primitif menjadi piksel pada framebuffer, dan operasi fragment memperbarui framebuffer dengan mencampur warna serta menerapkan efek visual seperti z-buffering. Akhirnya, gambar ditampilkan di layar menggunakan perintah seperti glFlush.

## PLANET Di OpenGL

Planet adalah salah satu benda besar yang berputar mengelilingi Matahari dalam tata surya. Dalam OpenGL, kita bebas mengekspresikan ide dan menciptakan berbagai objek, seperti planet. Kami dapat membuat sebuah planet dengan menambahkan kombinasi elemen visual dan logika agar simulasi tata surya berjalan dengan lancar dan terlihatiirealistis.

# BAB III HASIL

## Source Code

Source code Dev c++ Planet:

|  |
| --- |
| #include<GL/glut.h>  #include <math.h>  #include <cstring>  const int numStars = 100;  float rotationAngle = 0.0;  float rotation =0.0;  float rotationCube = 0.0;  float planetOrbit = 0.0;  float moonOrbit = 0.0;  float rotationX = 0.0f;  float rotationY = 0.0f;  float scale = 1.0f;  float cameraX = 0.0, cameraY = 0.0, cameraZ = 20.0;  float cameraAngleX = 0.0, cameraAngleY = 0.0;  float star[numStars][3];  float lightPos[] = {0.0f, 10.0f, 0.0f, 1.0f}; // posisi cahaya x, y, z, w  float lightPos1[] = {5.0f, 10.0f, 0.0f, 1.0f}; // Posisi cahaya kedua (x, y, z, w)  bool isMoving = true;  bool hidden = false;  void Sphere();  void Muter();  void Tetap(int w, int h);  void drawcartecius();  void Anak();  void hiddenCarte();  void Bintang();  void drawBintang();  void drawText(float x , float y , float z, const char\* text, int fontSize);  void keyboard(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, 800);  // glutInitWindowPosition(100, 100);  glutCreateWindow("Kelompok 4");  glEnable(GL\_DEPTH\_TEST);  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  gluPerspective(45.0, 1.0,1.0,100.0);  //Display  glutDisplayFunc(Sphere);  glutIdleFunc(Muter);  glutReshapeFunc(Tetap);  glutKeyboardFunc(keyboard);  glutFullScreen();  glutMainLoop();  return 0;  }  void Sphere()  {  glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);    glMatrixMode(GL\_MODELVIEW);  glLoadIdentity();  gluLookAt(cameraX, cameraY, cameraZ, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);    glEnable(GL\_LIGHTING);  glEnable(GL\_LIGHT0); // cahaya 1  glEnable(GL\_COLOR\_MATERIAL);  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);  // cahaya 2  glEnable(GL\_LIGHT1);  GLfloat ambientLigh1[] = {0.2f, 0.2f, 0.2f, 1.0f};  GLfloat diffuseLight1[] = {0.8f, 0.8f, 0.8f, 1.0f};  glLightfv(GL\_LIGHT1, GL\_AMBIENT, ambientLight);  glLightfv(GL\_LIGHT1, GL\_DIFFUSE, diffuseLight1);  glLightfv(GL\_LIGHT1, GL\_POSITION, lightPos1);    drawBintang();  hiddenCarte();    glPushMatrix();  glColor3ub(236,131,5); // Warna planet utama (orange)  glRotatef(rotation += 0.01,0,1,0);  glRotated(90,1.0,0.0,0.0);  glutSolidSphere(6.0,20,20);    glPushMatrix();  glTranslatef(0.0,0.0,-7.0);  drawText(0.0,0.0,0.0,"Matahari",20);  glPopMatrix();    glPopMatrix();  Anak();  Bintang();  glutSwapBuffers();  glutPostRedisplay ();  }  void Anak() // Oleh pipit  {  // Planet pertama  glPushMatrix();  glColor3ub(169, 169, 169); // abu-abu  glRotatef(planetOrbit, 0.0, 1.0, 0.0);  glTranslatef(8.0, 0.0, 0.0);  glRotatef(rotation \* 2, 0.0, 1.0, 0.0);  glutSolidSphere(0.5, 16, 16);    glPushMatrix();  glTranslatef(0.0,1.0,0.0);  drawText(0.0,0.0,0.0,"Merkurius",12);  glPopMatrix();    glPopMatrix();  // Planet kedua  glPushMatrix();  glColor3ub(233, 223, 196); // Warna krem  glRotatef(planetOrbit + 120, 0.0, 1.0, 0.0); // Rotasi dengan offset 120 derajat  glTranslatef(8.0, 0.0, 0.0); // Orbit lebih jauh  glRotatef(rotation \* 45, 0.0, 1.0, 0.0);  glutSolidSphere(0.9, 20, 20);    glPushMatrix();  glTranslatef(0.0,1.2,0.0);  drawText(0.0,0.0,0.0,"venus",12);  glPopMatrix();    glPopMatrix();  // Planet ketiga  glPushMatrix();  glColor3ub(0, 0, 255); // Warna biru  glRotatef(planetOrbit + 90, 0.0, 1.0, 0.0); // Rotasi dengan offset 240 derajat  glTranslatef(10.0, 0.0, 0.0); // Orbit lebih jauh  glRotatef(rotation \* 12.0, 0.0, 1.0, 0.0);  glutSolidSphere(1.0, 20, 20);    glPushMatrix();  glTranslatef(0.0,1.5,0.0);  drawText(0.0,0.0,0.0,"Bumi",12);  glPopMatrix();    glPushMatrix();  glColor3ub(200,200,200);  glRotatef(moonOrbit,0.0,0.1,0.0);  glTranslatef(3.0,0.0,1.5);  drawText(0.0,0.0,0.0,"Bulan",12);  glutSolidSphere(0.3,16,16);  glPopMatrix();    glPopMatrix();    // Planet keempat  glPushMatrix();  glColor3ub(255, 69, 0); // Warna planet keempat (Merah)  glRotatef(planetOrbit + 135, 0.0, 1.0, 0.0);  glTranslatef(14.0, 0.0, 0.0);  glRotatef(rotation \* 1.2, 0.0, 1.0, 0.0);    glutSolidSphere(0.7, 20, 20);    glPushMatrix();  glTranslatef(0.0,1.2,0.0);  drawText(0.0,0.0,0.0,"Mars",12);  glPopMatrix();  glPopMatrix();  // Planet kelima  glPushMatrix();  glColor3ub(255, 165, 0); // Warna planet kelima (oranye)  glRotatef(planetOrbit + 180, 0.0, 1.0, 0.0);  glTranslatef(18.0, 0.0, 0.0);  glRotatef(rotation \* 0.5, 0.0, 1.0, 0.0);  glutSolidSphere(2.0, 25, 25);  glPushMatrix();  glTranslatef(0.0,2.2,0.0);  drawText(0.0,0.0,0.0,"Jupiter",12);  glPopMatrix();  glPopMatrix();  // Planet keenam  glPushMatrix();  glColor3ub(210, 180, 140); // Warna planet keenam (coklat)  glRotatef(planetOrbit + 255, 0.0, 1.0, 0.0);  glTranslatef(22.0, 0.0, 0.0);  glRotatef(rotation \* 0.3, 0.0, 1.0, 0.0);  glutSolidSphere(1.7, 25, 25);    glPushMatrix();//Cicin saturnus  glColor3ub(169,169,169);  glTranslatef(0.0, 0.1, 0.0);  glutSolidTorus(0.1,2.0,30,30);  glPopMatrix();    glTranslatef(0.0,0.85,0.0);  drawText(0.0,0.0,0.0,"Saturnus",12);  glPopMatrix();  glPopMatrix();  // Planet ketujuh  glPushMatrix();  glColor3ub(135, 206, 250); // Warna planet ketujuh (ungu)  glRotatef(planetOrbit + 270, 0.0, 1.0, 0.0);  glTranslatef(26.0, 0.0, 0.0);  glRotatef(rotation \* 0.2, 0.0, 1.0, 0.0);  glutSolidSphere(1.5, 20, 20);  glPushMatrix();  glTranslatef(0.0,2.0,0.0);  drawText(0.0,0.0,0.0,"Uranus",12);  glPopMatrix();  glPopMatrix();  }  void Bintang()  {  for (int i = 0; i < numStars; ++i)  { // 100 bintang  star[i][0] = (rand() % 201 - 100) / 10.0f; // Posisi X acak  star[i][1] = (rand() % 201 - 100) / 10.0f; // Posisi Y acak  star[i][2] = (rand() % 201 - 100) / 10.0f; // Posisi Z acak  }  glEnd();  }  void drawBintang()  {  glMatrixMode(GL\_MODELVIEW);  glPushMatrix();  glLoadIdentity();  glPointSize(2.0); // Ukuran bintang kecil  glBegin(GL\_POINTS);  glColor3ub(255, 255, 255); // Warna bintang (putih)  for (int i = 0; i < numStars; ++i)  { // 100 bintang  glVertex3f(star[i][0], star[i][1], star[i][2]); // Posisi bintang  }  glEnd();  glPopMatrix();  }  void Muter()  {  rotation += 0.05;  if (rotation > 360.0){  rotation -= 360.0;  }  planetOrbit += 0.02;  if (planetOrbit > 360.0) {  planetOrbit-= 360.0;  }  moonOrbit+=0.1;  if (moonOrbit > 360.0){  moonOrbit -= 360.0;  }    glutPostRedisplay();  }  void Tetap(int w, int h)  {    if (h == 0) h = 1;    glViewport(0, 0, w, h);  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  gluPerspective(45.0, (float)w / (float)h, 1.0, 100.0);  }  void drawcartecius()  {  glLineWidth(1.0);  glColor3f(1.0,0.0,0.0);  glBegin(GL\_LINES);  glVertex3f(-50.0,0.0,0.0);  glVertex3f(50.0,0.0,0.0);  glVertex3f(0.0,-50.0,0.0);  glVertex3f(0.0,50.0,0.0);  glVertex3f(0.0,0.0,-50.0);  glVertex3f(0.0,0.0,50.0);  glEnd();  }  void keyboard(unsigned char key, int x, int y)  {  float step = 0.5;  switch (key) {  case 'w':  cameraZ -= step;  break;  case 's':  cameraZ += step;  break;  case 'a':  cameraX -= step;  break;  case 'd':  cameraX += step;  break;  case 'q':  cameraY += step;  break;  case 'e':  cameraY -= step;  break;  case 'c':  hidden = !hidden;  break;  case 27: // ESC untuk keluar  exit(0);  break;  }  glutPostRedisplay();    }    void hiddenCarte()  {  if (hidden) {  drawcartecius();  }  }  void drawText(float x , float y , float z, const char\* text, int fontSize)  {  glColor3ub(0,255,255);  glPushMatrix();  glTranslatef(x, y, z);  glScalef(fontSize/8.0f, fontSize / 13.0f, 1.0f);  glRasterPos2f(0,0);  for (const char\* c = text; \*c != '\0';++c){  glutBitmapCharacter(GLUT\_BITMAP\_8\_BY\_13, \*c);  }  glPopMatrix();  glEnd();  } |

## Output

1. A yellow circle with white dots

   Description automatically generatedGambar awal

Gambar 8 Tampilan awal

1. A yellow and blue spheres in space

   Description automatically generated with medium confidenceGambar atas

Gambar 9 Dari atas

1. Gambar samping

A group of planets in space

Description automatically generated

Gambar 10 Dari samping

## Penjelasan

1. Fungsi Utama

* Membuat jendela grafik dengan judul "Kelompok 4".
* Mengatur mode tampilan 3D menggunakan gluPerspective.
* Menginisialisasi fungsi-fungsi callback seperti display(Sphere), idle (`Muter`), reshape (Tetap), dan keyboard (keyboard).

1. Objek Simulasi

* Matahari: Objek utama dengan warna oranye.
* Planet-planet: Total 7 planet dengan warna, ukuran, dan orbit berbeda. Setiap planet bergerak mengelilingi Matahari.
* Bulan: Mengorbit di sekitar planet ketiga (Bumi).

1. Efek Visual

* Pencahayaan: Menggunakan dua sumber cahaya (dengan konfigurasi GL\_LIGHT0 dan GL\_LIGHT1) untuk memberikan efek realistis pada objek.
* Bintang: Dibuat dengan titik-titik putih acak yang disebar di sekitar ruang simulasi untuk menyerupai latar langit.

1. Navigasi kamera

* Posisi kamera dapat diubah dengan tombol W, A, S, D, Q, E untuk mendekat, menjauh, atau bergerak di sumbu X, Y, dan Z.

1. Fungsi Interaktif

* Tombol “C”: Menampilkan atau menyembunyikan sistem koordinat kartesius (garis sumbu X, Y, Z).
* Tombol “ESC” : Keluar dari aplikasi.

1. Pergerakan objek

* Semua planet, bulan, dan Matahari memiliki rotasi yang terus-menerus, diatur dalam fungsi Muter.
* Orbit planet dan bulan diperbarui secara real-time untuk menampilkan simulasi pergerakan yang dinamis.

1. Teks Informasi

* Nama setiap objek (Matahari, planet, dan bulan) ditampilkan di dekat masing-masing objek menggunakan fungsi drawText.

# 

# BAB IV

## Kesimpulan

OpenGL adalah library dasar yang memberikan fondasi kuat untuk pengembangan grafis 3D. Praktikum ini memperkenalkan konsep dasar, namun tidak mencakup semua fitur OpenGL. Untuk memperdalam pemahaman, eksplorasi mandiri sangat dianjurkan agar memperoleh pengetahuan yang lebih komprehensif dan sesuai dengan kebutuhan pribadi.

# DAFTAR PUSTAKA

[1] Ahmad Taufik Al Afkari Siahaan, *Panduan Pengunaan Dec c++ dan Open GL*. Sumatera Utara, 2024.

[2] “Pemrograman Grafis 3D dengan OpenGL dan GLUT | Vicknite.” Accessed: Dec. 16, 2024. [Online]. Available: https://vicknite.wordpress.com/2010/05/25/basic-of-opengl-3d-project-tutorial/

[3] “Pengantar Pemrograman OpenGL GLUT C++ di Linux | Ade Malsasa Akbar.” Accessed: Dec. 16, 2024. [Online]. Available: https://malsasa.wordpress.com/2014/04/15/pengantar-pemrograman-opengl-glut-c-di-linux/