

LAPORAN PRATIUM GRAFIK KOMPUTER

Diajukan untuk memenuhi Tugas mata kuliah Pratikum Grafik Komputer

OBJEK ALAT MAKAN DALAM BENTUK 3D

Dosen Pengampu : Sri Rahayu, M.Kom

Instruktur Pratikum : Arul Budi Kalimat, S.Kom



Disusun oleh

Kelompok : 5

M Anwar Sanusi

2306016

M Jamil Alfadilah

2306022

Khaeri Agustiana

2306001

PROGRAM STUDI TEKNIK INFORMATIKA

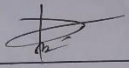
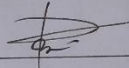
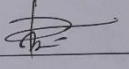
JURUSAN ILMU KOMPUTER

INSTITUT TEKNOLOGI GARUT

2024

ASISTENSI TUGAS PRAKTIKUM GRAFIK KOMPUTER

Nama Pembimbing : Garas Kirani
Kelompok & Kelas : Kelompok 5 - A
Anggota Kelompok :
- M. Anwar Sarusi
- Khaeri Agustiana
- M. Jamil Atrodi/ah
Tema : Objek 3D piring, sendok, Garpu

No	Tanggal	Uraian Asistensi	Tanda Tangan
1	08-01-23	Membahas konsep dan tema	
2	14-01-23	Progres Tb 1	
3	15-01-23	Progres Tb 2 Final	
4			
5			

Instruktur Praktikum



Arul Budi Kalimat, S.Kom

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 objek piring, sendok dan garpu

Kami mengucapkan terima kasih kepada dosen pengampu Sri Rahayu, M.Kom, instruktur praktikum Arul Bufi Kalimat S.Kom Dan Pembimbing kami Garnis, 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, 15 Januari 2024

Kelompok 5

DAFTAR ISI

KATA PENGANTAR	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR	v
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	1
1.3 Tujuan.....	1
BAB II TINJAUAN PUSTAKA.....	3
2.1 OpenGL.....	3
2.2 Konfigurasi OpenGL pada Dev C++	3
2.3 Cara Kerja OpenGL	8
2.4 Objek Piring, Sendok Dan Garpu.....	9
BAB III HASIL.....	10
3.1 Source Code.....	10
3.2 Output	17
3.3 Penjelasan.....	18
BAB IV	20
4.1. Kesimpulan.....	20
DAFTAR PUSTAKA	21

DAFTAR GAMBAR

Gambar 1 buka folder freeglut	3
Gambar 2 buka folder include/gl.....	3
Gambar 3 simoan pada C: programfile	3
Gambar 4 Masuk ke folder lib.....	4
Gambar 5 simpan pada C	4
Gambar 6 Masuk folder bin	4
Gambar 7 Simpan pada system 32	4
Gambar 8 Membuka Aplikasi Dev C++	5
Gambar 9 Pilih File Untuk Konfigurasi	5
Gambar 10 Memilih Console Aplication Untuk Konfigurasi	6
Gambar 11 Untuk Menyimpan File	6
Gambar 12 Mengisi Parameter.....	7
Gambar 13 Tampilan Setelah Konfigurasi Selesai	7

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kehidupan manusia dipenuhi dengan berbagai interaksi satu dengan yang lain, sehingga setiap individu atau kelompok manusia akan melakukan suatu respon atau reaksi yang sesuai dengan aksi yang diberikan. Atas dasar hal tersebut, maka interaksi manusia dengan yang lainnya akan berjalan secara efektif jika informasi yang diberikan oleh manusia satu ke manusia yang lain dapat dimengerti. Sehingga respon yang dihasilkan akan bersifat linier dengan aksi yang diberikan. Hal tersebut dapat didefinisikan sebagai komunikasi. Komunikasi merupakan cara bertukar informasi dalam konteks interaksi yang dilakukan seseorang atau kelompok dengan individu atau kelompok orang lain dengan syarat, informasi yang diberikan dapat dipahami dan dimengerti satu sama lain. Dalam komunikasi terdapat ketentuan dan standar yang digunakan, misalnya saja kesamaan Bahasa yang digunakan, struktur kalimat atau kata serta ucapan yang terdengar jelas dan dapat dipahami serta kemampuan memahami setiap ucapan dan kata yang dapat diterjemahkan secara tepat jika menggunakan bahasa yang berbeda. [1]

Salah satu lembaga yang memanfaatkan teknologi komputer adalah lembaga pendidikan. Program studi S1 Pendidikan Teknik Informatika adalah salah satu program studi yang bergerak di bidang tekno-logi informasi dan komunikasi, khususnya bidang teknik informatika. Mahasiswa pada program studi ini harus menguasai dua bidang yaitu pendidikan dan teknik informatika. Didalam bidang teknik informatika, ada sejumlah matakuliah yang wajib di kuasai oleh peserta didik program studi ini, salah satunya adalah matakuliah Praktikum Grafika Komputer. Pada matakuliah tersebut, peserta didik dilatih agar memiliki kemampuan dalam membuat dan mengembangkan program-program grafika komputer.[2]

Menurut Suyoto (2003), Grafika komputer(Computer Graphic) dapat diartikan sebagai seperangkat alat yang terdiri dari hardware dan software untuk membuat gambar, grafik atau citra realistik untuk seni, game komputer, foto dan film animasi. Sistem grafika komputer dapat dijalankan dengan komputer pribadi (Personal Computer) atau workstation. Grafika komputer semakin lama semakin pesat perkembangannya sehingga definisi dari grafika komputer dapat diartikan sebagai suatu studi tentang bagaimana menggambar (membuat grafik) dengan menggunakan komputer dan manipulasinya (merubah sedikit/transformatasi/ animasi).[3]

1.2 Rumusan Masalah

Rumusan masalah dalam praktikum tersebut yaitu:

1. Apa yang dimaksud dengan OpenGL ?
2. Bagaimana cara mengkonfigurasi OpenGL pada Dev C++
3. Bagaimana cara kerja dari OpenGL ?
4. Bagaimana membuat objek piring, sendok dan garpu dalam OpenGL

1.3 Tujuan

Adapun tujuan dalam praktikum tersebut:

1. Mengetahui apa itu OpenGL
2. Mengetahui cara mengkonfigurasi OpenGL pada Dev C++
3. Mengetahui cara kerja dari OpenGL

4. Mengetahui cara pembuatan objek piring, sendok dan garpu dalam OpenGL

BAB II

TINJAUAN PUSTAKA

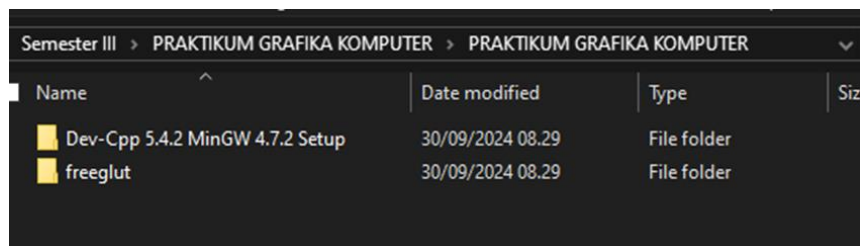
2.1 OpenGL

OpenGL adalah suatu spesifikasi grafikasi yang low level yang menyediakan fungsi untuk mempermudah pekerjaan atau untuk keperluan –keperluan pemrograman grafika (Graphics Programming / GP) Termasuk Grafika primitif (titik, garis, dan lingkaran). Singkatnya, Open Graphics Library, OpenGL menghilangkan kebutuhan untuk pemrogram untuk menulis ulang bagian grafis dari sistem operasi setiap kali sebuah bisnis akan diupgrade ke versi baru dari sistem. Fungsi dasar dari OpenGL adalah untuk mengeluarkan koleksi perintah khusus atau executable ke sistem operasi. Perintah mengawali untuk menggambar. Mode merupakan konstanta yang menyatakan bagaimana opengl harus menghubungkan titik/vertex yang akan digambarkan.[3]

2.2 Konfigurasi OpenGL pada Dev C++

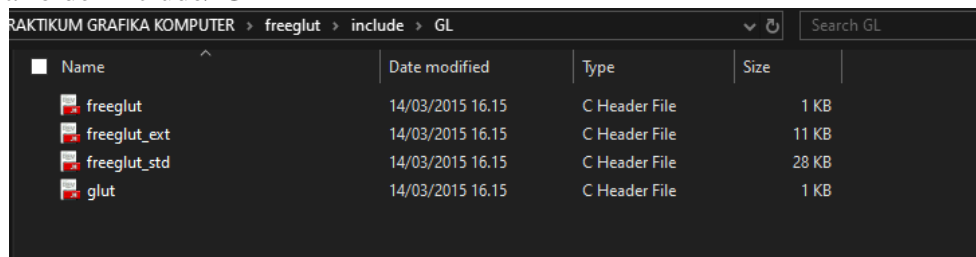
Untuk mengkonfigurasi opengl pada Dev C++ ada beberapa hal yang harus dilakukan, berikut langkah langkah untuk mengkonfigurasi open gl pada Dev C++

1. Buka folder freeglut yang sudah di install



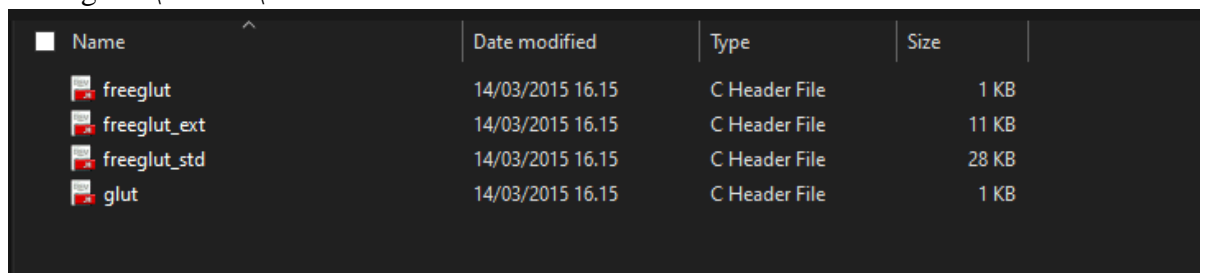
Gambar 1 buka folder freeglut

2. Buka folder include/ GL





Gambar 2 buka folder include/gl

3. Lalu simpan pada C:\Programfile (x86)\dev-cpp\minGW64\x86_64-w64-mingw32\include\GL














Gambar 3 simoan pada C: programfile

4. Kemudian masuk ke folder lib, lalu buka folder x64, dan copy keseluruhan file didalamnya

	libfreeglut.a	14/03/2015 16.26	A File	103 KB
	libfreeglut_static.a	14/03/2015 16.27	A File	482 KB



Gambar 4 Masuk ke folder lib

5. Lalu simpan pada C:\Programfile (x86)\dev-cpp\minGW64\x86_64-w64-mingw32\lib

	Idscripts	01/11/2024 14.22	File folder	
	binmode.o	08/12/2014 00.48	O File	2 KB
	CRT_fp8.o	08/12/2014 00.48	O File	3 KB
	CRT_fp10.o	08/12/2014 00.48	O File	2 KB
	CRT_glob.o	08/12/2014 00.48	O File	2 KB
	CRT_noglob.o	08/12/2014 00.48	O File	2 KB
	crt1.o	08/12/2014 00.48	O File	28 KB
	crt1u.o	08/12/2014 00.48	O File	28 KB
	crt2.o	08/12/2014 00.48	O File	28 KB
	crt2u.o	08/12/2014 00.48	O File	28 KB
	crtbegin.o	08/12/2014 00.48	O File	1 KB







Gambar 5 simpan pada C

6. Lalu masukan folder bin, lalu buka folder x64, lalu copy seluruh file freeglut

	x64	01/11/2024 14.05	File folder	
	freeglut.dll	14/03/2015 16.24	Application exten...	222 KB

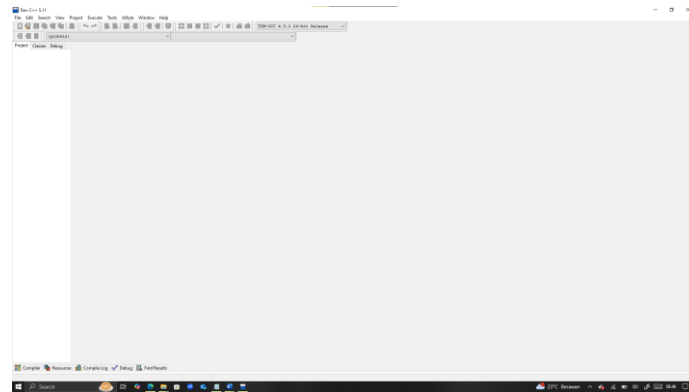
Gambar 6 Masuk folder bin

7. Lalu simpan pada system 32

	addr2line	08/12/2014 00.33	Application	867 KB
	ar	08/12/2014 00.33	Application	890 KB
	as	08/12/2014 00.33	Application	1.468 KB
	c++	08/12/2014 05.34	Application	793 KB
	c++filt	08/12/2014 00.33	Application	866 KB
	cpp	08/12/2014 05.34	Application	791 KB

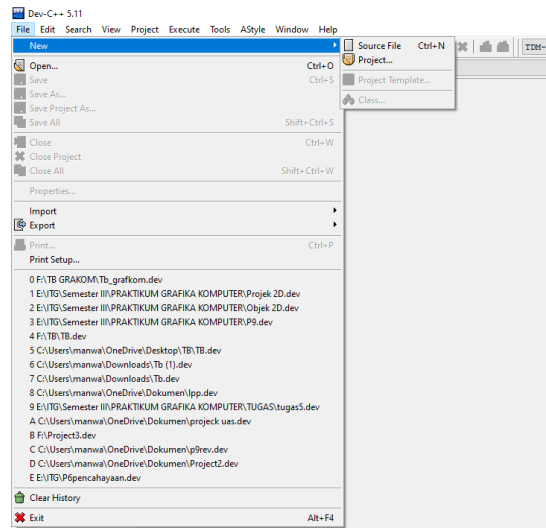
Gambar 7 Simpan pada system 32

8. Buka aplikasi Dev C++



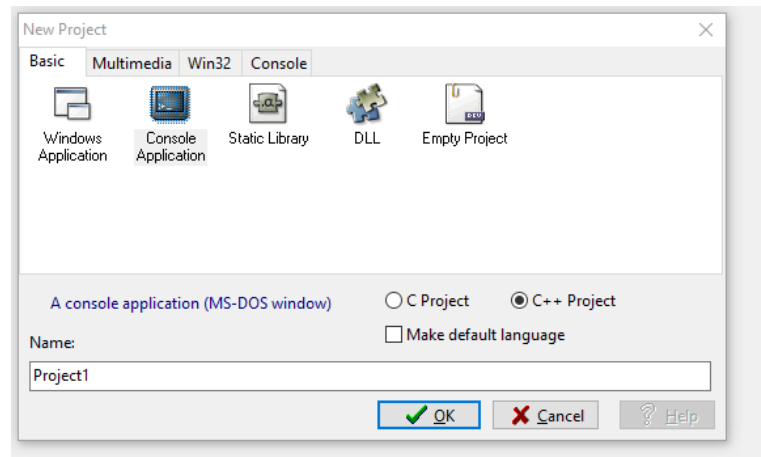
Gambar 8 Membuka Aplikasi Dev C++

9. Langkah selanjutnya kita ke menu file, lalu klik new dan pilih project



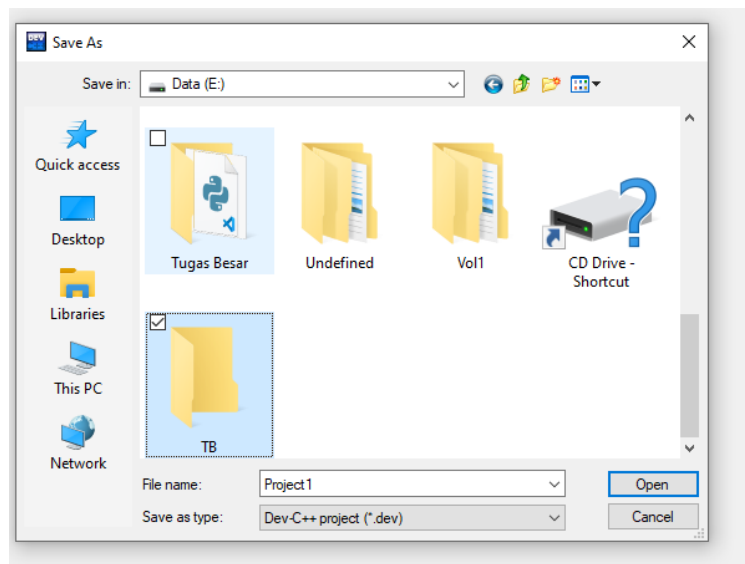
Gambar 9 Pilih File Untuk Konfigurasi

10. Selanjutnya pilih console Application



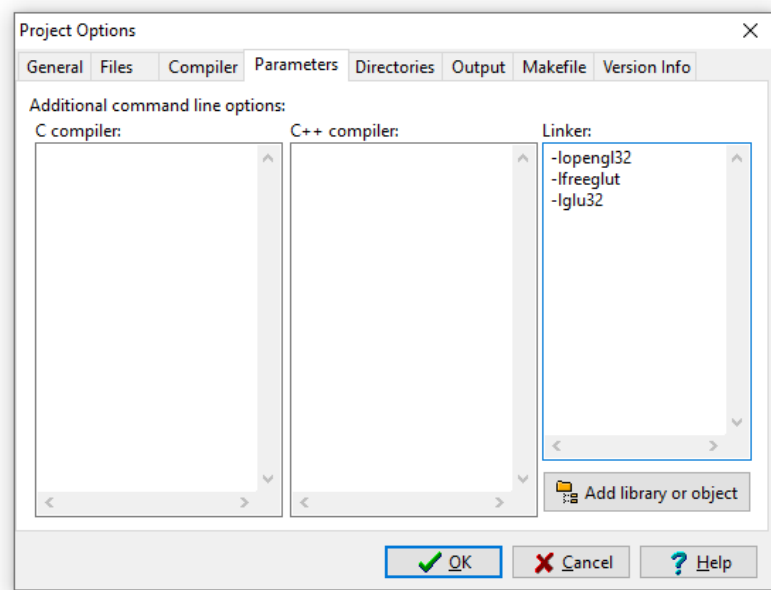
Gambar 10 Memilih Console Application Untuk Konfigurasi

11. Selanjutnya pilih folder untuk menyimpan file tersebut



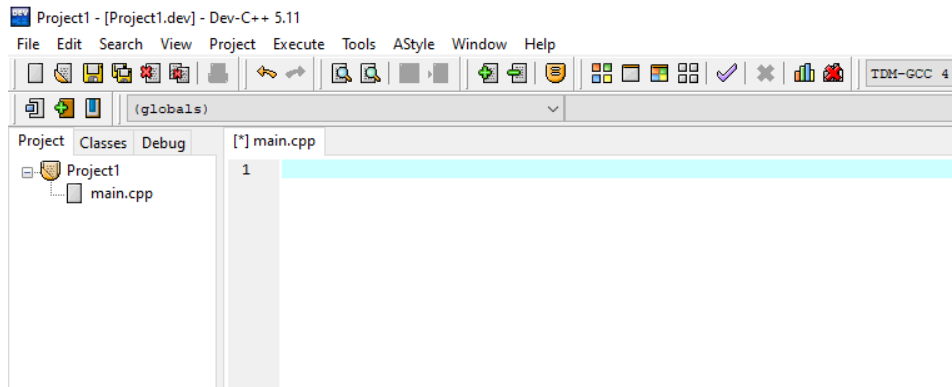
Gambar 11 Untuk Menyimpan File

12. Selanjutnya pada keyboard klik ctrl+h untuk mengatur dan masuk ke parameter lalu isi parameter tersebut seperti di bawah ini



Gambar 12 Mengisi Parameter

13. Dan konfigurasi selesai



Gambar 13 Tampilan Setelah Konfigurasi Selesai

2.3 Cara Kerja OpenGL

OpenGL adalah API (Application Programming Interface) yang digunakan untuk menghasilkan grafik 2D dan 3D. Di bawah ini adalah penjelasan tentang cara kerja OpenGL sesuai dengan langkah-langkah konfigurasi pada Dev C++ yang telah dijelaskan sebelumnya.

1. Instalasi dan Konfigurasi Library OpenGL di Dev C++:
 - a. OpenGL membutuhkan beberapa file dan folder yang berisi header (untuk mendeklarasikan fungsi dan tipe data), library (untuk pengikatan fungsi saat kompilasi), dan file eksekusi (untuk penggunaan runtime).
 - b. Langkah pertama yang dilakukan adalah mengonfigurasi lingkungan pengembangan seperti Dev C++ untuk dapat menggunakan OpenGL dengan menyalin file yang diperlukan ke dalam direktori Dev C++ yang sesuai.
 - Folder include/GL berisi header yang mendeklarasikan fungsi dan struktur data yang digunakan dalam OpenGL.
 - Folder lib/x64 berisi file library yang diperlukan untuk menghubungkan (link) fungsi OpenGL saat proses kompilasi.
 - Folder bin/x64 berisi file DLL yang diperlukan untuk menjalankan program OpenGL pada sistem operasi Windows.
2. Pengaturan di Dev C++:
 - a. Setelah menyalin file-file tersebut, langkah selanjutnya adalah mengatur Dev C++ untuk mengenali dan menggunakan file OpenGL.
 - b. Anda membuka Dev C++ dan membuat proyek baru dengan memilih Console Application.
 - c. Pada tahap ini, Anda mengatur parameter dalam pengaturan proyek untuk memberi tahu Dev C++ tentang lokasi file header dan library OpenGL. Hal ini memungkinkan Dev C++ untuk mengenali dan menghubungkan kode OpenGL dengan benar saat kompilasi.
3. Fungsi dan Proses OpenGL:

Setelah OpenGL dikonfigurasi dengan benar, Anda bisa mulai menulis kode yang menggunakan API OpenGL.

 - Inisialisasi: Sebelum menggunakan OpenGL, Anda perlu melakukan inisialisasi, seperti mengatur mode tampilan, ukuran jendela, dan sebagainya.
 - Rendering: Setelah inisialisasi, Anda dapat menggunakan berbagai fungsi OpenGL untuk menggambar objek, seperti titik, garis, segi empat, dll.
 - Loop Render: Biasanya, program OpenGL akan memiliki loop utama di mana semua objek digambar dan diperbarui dalam tampilan secara berulang.
 - Penutupan: Ketika program selesai, Anda harus membersihkan sumber daya yang digunakan oleh OpenGL, seperti buffer dan konteks grafik.
4. Pengaturan Parameter (Ctrl+H):
 - a. Mengatur parameter proyek di Dev C++ memastikan bahwa Dev C++ tahu di mana menemukan file header (include) dan file library (lib) OpenGL.
 - b. Dengan mengatur ini, Dev C++ dapat menghubungkan program yang Anda buat dengan OpenGL, memungkinkan Anda untuk menggunakan fungsi OpenGL dalam kode Anda.

5. Program OpenGL:

- a. Setelah semua konfigurasi selesai, Anda dapat mulai menulis kode OpenGL yang akan dijalankan dalam aplikasi yang telah dikonfigurasi.
- b. Biasanya, program OpenGL berisi fungsi-fungsi untuk menggambar objek grafis dan melakukan perhitungan matematis terkait grafik 3D, seperti proyeksi, pencahayaan, dan transformasi objek.

2.4 Objek Piring, Sendok Dan Garpu

Program Open Gl 3D Untuk membuat objek piring, sendok, dan garpu dalam OpenGL, kita bisa menggunakan kombinasi primitif dan transformasi. Piring dapat digambarkan dengan bentuk torus atau disk, dengan menggunakan fungsi `glutSolidTorus()` untuk menggambar piring berbentuk torus yang lebih besar dan diletakkan di bawah objek lain. Sendok terdiri dari dua bagian utama, yaitu pegangan yang digambarkan dengan bentuk silinder, dan bagian sendok yang biasanya berbentuk bola atau kerucut (cone) untuk ujung sendok yang membulat. Garpu juga terdiri dari pegangan yang digambarkan dengan silinder, serta gigi-gigi garpu yang biasanya dibuat dengan beberapa silinder kecil yang lebih tipis, disusun untuk membentuk gigi-gigi garpu tersebut. Dengan menggunakan bentuk dasar ini, kita dapat membentuk objek yang lebih kompleks melalui transformasi seperti translasi, rotasi, dan skala.

BAB III

HASIL

3.1 Source Code

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

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

// Variabel
float tableRotationAngle = 0.0f;
float scaleFactor = 1.0f;
float translationX = 0.0f, translationY = -1.0f, translationZ = 0.0f;
float rotationAngle = 0.0f;
float showCartecius = false;
float plateBaseColor[3] = {1.0f, 1.0f, 0.0f};
float plateEdgeColor[3] = {1.0f, 0.6f, 0.0f};
float spoonColor[3] = {0.5f, 0.5f, 0.5f};
float forkColor[3] = {0.5f, 0.5f, 0.5f};
float tableColor[3] = {0.5f, 0.3f, 0.1f};
float cameraRotationX = 0.0f;
float cameraRotationY = 0.0f;

// Fungsi menerapkan transformasi
void applyTransformations(float tx, float ty, float tz, float scale, float angle) {
    glTranslatef(tx, ty, tz);
    glRotatef(angle, 0.0f, 1.0f, 0.0f);
    glScalef(scale, scale, scale);
}

//
=====
=====M
JAMIL=====
=====
void drawTable() {
    glPushMatrix();

    glColor3fv(tableColor);

    // Gambar alas meja
    glPushMatrix();
    glTranslatef(0.0f, 0.5f, 0.0f);
    glScalef(4.0f, 0.2f, 4.0f);
    glutSolidCube(1.0f);
    glPopMatrix();

    //Kaki Meja
    for (float x = -1.8f; x <= 1.8f; x += 3.6f) {
        for (float z = -1.8f; z <= 1.8f; z += 3.6f) {
            glPushMatrix();
            glTranslatef(x, -1.0f, z);
```

```

        glScalef(0.1f, 1.5f, 0.1f);
        glutSolidCube(2.0);
        glPopMatrix();
    }
}

glPopMatrix();
}

// Menggambarr sendok oleh M Jamil
void drawSpoon() {
    glColor3fv(spoonColor);

    // Gagang sendok
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.0f);
    glScalef(0.1, 0.1, 1.5);
    glutSolidCube(0.5);
    glPopMatrix();

    // Kepala sendok
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.4f);
    glScalef(0.2, 0.2, 0.5);
    glutSolidSphere(.5, 25, 25);
    glPopMatrix();
}

// garfu
=====
= KHAERI
AGUSTIANA=====
=====
void drawFork() {
    glColor3fv(forkColor);

    // Gagang garpu
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.0f);
    glScalef(0.1, 0.1, 1.5);
    glutSolidCube(0.5);
    glPopMatrix();

    // Gigi garpu
    for (float i = -0.15f; i <= 0.15f; i += 0.15f) {
        glPushMatrix();
        glTranslatef(i, 0.0f, 0.55f);
        glScalef(0.05, 0.05, 0.6);
        glutSolidCube(0.5);
        glPopMatrix();
    }

    // Bagian bawah garpu
    glPushMatrix();

```



```

        glTranslatef(0.0f, 0.0f, 0.4f);
        glRotatef(-90.0f, 0.0f, 1.0f, 0.0f);
        glScalef(0.02, 0.02, 0.2);
        glutSolidCube(1.5);
        glPopMatrix();
    }

    // Fungsi untuk mengatur pencahayaan
    void initLighting() {
        glEnable(GL_LIGHTING);
        glEnable(GL_LIGHT0);

        GLfloat ambientLight[] = {1.0f, 1.0f, 1.0f, 1.0f};
        glLightfv(GL_LIGHT0, GL_AMBIENT, ambientLight);

        GLfloat diffuseLight[] = {1.0f, 1.0f, 1.0f, 1.0f};
        glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuseLight);

        GLfloat lightPosition[] = {0.0f, 5.0f, 5.0f, 1.0f};
        glLightfv(GL_LIGHT0, GL_POSITION, lightPosition);
    }

    //Piring
    =====
    ===== M ANWAR
    SANUSI=====
    =====
    void drawConcavePlate() {
        GLUQuadric* quad = gluNewQuadric();
        gluQuadricNormals(quad, GLU_SMOOTH);

        glPushMatrix();
        glColor3fv(plateBaseColor);
        gluDisk(quad, 0.0, 0.45, 50, 50); //untuk membuat dasar piring
        glPopMatrix();

        glPushMatrix();
        glColor3fv(plateEdgeColor);
        glTranslatef(0.0f, 0.0f, -0.1f);
        glutSolidTorus(0.05, 0.5, 50, 50); //untuk membuat sisi piring
        melengkung
        glPopMatrix();

        glPushMatrix();
        glColor3fv(plateBaseColor);
        glTranslatef(0.0f, 0.0f, -0.1f);
        gluCylinder(quad, 0.5, 0.45, 0.1, 50, 50); // membuat silinder
        piring
        glPopMatrix();
    }

```

```

gluDeleteQuadric(quad);

}

// Garis cartecius
=====M
ANWAR
SANUSI=====
==
void drawCartecius(float tx, float ty, float tz, float
rotationAngle) {
    if (showCartecius) {

        glPushMatrix();
        glTranslatef(tx, -1.2f, 1);
        glRotatef(rotationAngle, 0.0f, 1.0f, 0.0f);

        // Sumbu X
        glPushMatrix();
        glColor3f(1.0f, 0.0f, 0.0f);
        glBegin(GL_LINES);
        glVertex3f(-3.0f, 0.0f, 0.0f);
        glVertex3f(3.0f, 0.0f, 0.0f);
        glEnd();
        glPopMatrix();

        // Sumbu Y
        glPushMatrix();
        glColor3f(0.0f, 1.0f, 0.0f);
        glBegin(GL_LINES);
        glVertex3f(0.0f, -3.0f, 0.0f);
        glVertex3f(0.0f, 3.0f, 0.0f);
        glEnd();
        glPopMatrix();

        // Sumbu Z
        glPushMatrix();
        glColor3f(0.0f, 0.0f, 1.0f);
        glBegin(GL_LINES);
        glVertex3f(0.0f, 0.0f, -3.0f);
        glVertex3f(0.0f, 0.0f, 3.0f);
        glEnd();
        glPopMatrix();

        glPopMatrix();
    }
}

// menampilkan tampilan
void display() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();

    gluLookAt(0.0, 0.0, 10.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

```

```

    glRotatef(cameraRotationX, 1.0f, 0.0f, 0.0f);
    glRotatef(cameraRotationY, 0.0f, 1.0f, 0.0f);

    // Terapkan translasi dan rotasi untuk seluruh objek secara
    bersamaan
    glPushMatrix();
    applyTransformations(translationX, translationY, translationZ,
    scaleFactor, rotationAngle);
    // Aktifkan penggunaan warna material agar warna objek tetap
    terjaga
    glEnable(GL_COLOR_MATERIAL);

    drawTable();

    // Gambar piring di atas meja
    glPushMatrix();
    glTranslatef(0.0f, 0.7f, 0.0f); // Translasi piring
    glRotatef(90, 90.0f, 1.0f, 0.0f);
    drawConcavePlate();
    glPopMatrix();

    // Gambar sendok di kanan piring
    glPushMatrix();
    glTranslatef(1.5f, 0.7f, 0.0f); // Translasi sendok
    drawSpoon();
    glPopMatrix();

    // Gambar garpu di kiri piring
    glPushMatrix();
    glTranslatef(-1.0f, 0.7f, 0.0f); // Translasi garpu
    drawFork();
    glPopMatrix();

    glPopMatrix();
    glPopMatrix();
    drawCartecius(0.0f, 0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glutSwapBuffers();
}

// Fungsi untuk menangani perubahan ukuran jendela
void reshape(int w, int h) {
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(45.0, (double)w / (double)h, 1.0, 100.0);
    glMatrixMode(GL_MODELVIEW);
}

// Fungsi keyboard untuk translasi objek, rotasi objek, dan
menampilkan garis Cartecius
void keyboard(unsigned char key, int x, int y) {
    float step = 0.1f; // Ukuran translasi per tombol tekan
    float rotationStep = 1.0f; // Ukuran langkah rotasi

```

```

switch (key) {
    case 'w': // maju
        translationZ -= step;
        break;
    case 's': //mundur
        translationZ += step;
        break;
    case 'a': //ke kiri
        translationX -= step;
        break;
    case 'd': // ke kanan
        translationX += step;
        break;
    case 'r': // rotsi berputar seajar jarum jam
        rotationAngle += rotationStep;
        initLighting();
        break;
    case 'f': // rotsi berputar berlawanan jarum jam
        rotationAngle -= rotationStep;
        initLighting();
        break;
    case '+': // zoom
        scaleFactor += 0.1f;
        initLighting();
        break;
    case '-': // zoom out
        scaleFactor -= 0.1f;
        if (scaleFactor < 0.1f) scaleFactor = 0.1f;
        initLighting();
        break;
    case 'c': // untuk menampilkan dan menghilangkan
cartecius
        showCartecius = !showCartecius;
        break;
    case 'z': // rotasi kamera ke atas
        cameraRotationX += 2.0f;
        break;
    case 'x': // rotasi kamera ke bawah
        cameraRotationX -= 2.0f;
        break;
    default:
        break;
}

glutPostRedisplay();
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(800, 600);
    glutCreateWindow("Tb_Grafkom Kel 5");
    glEnable(GL_DEPTH_TEST);

    glDisable(GL_LIGHTING);

    initLighting();

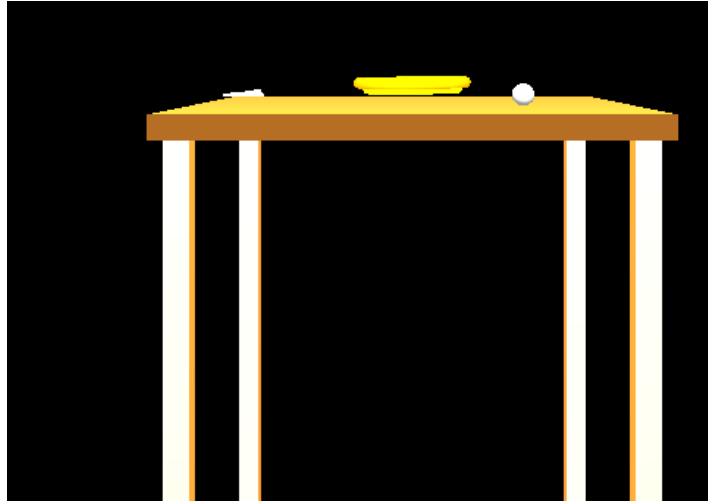
```

```
glutDisplayFunc(display);  
glutReshapeFunc(reshape);  
glutKeyboardFunc(keyboard);
```

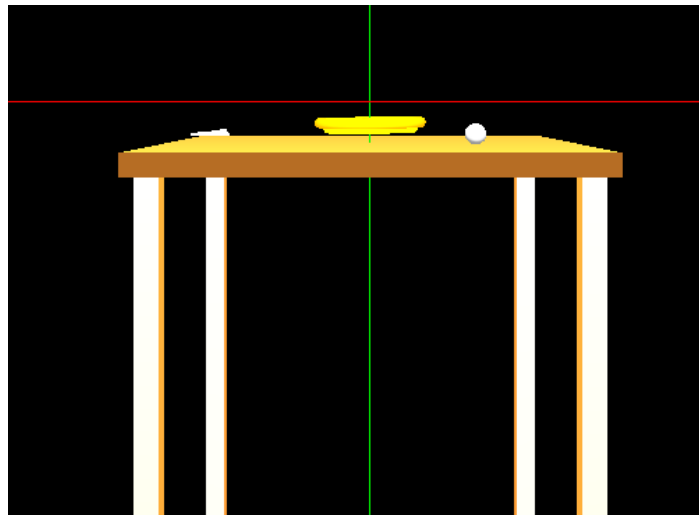
```
glutMainLoop();  
return 0;
```

```
}
```

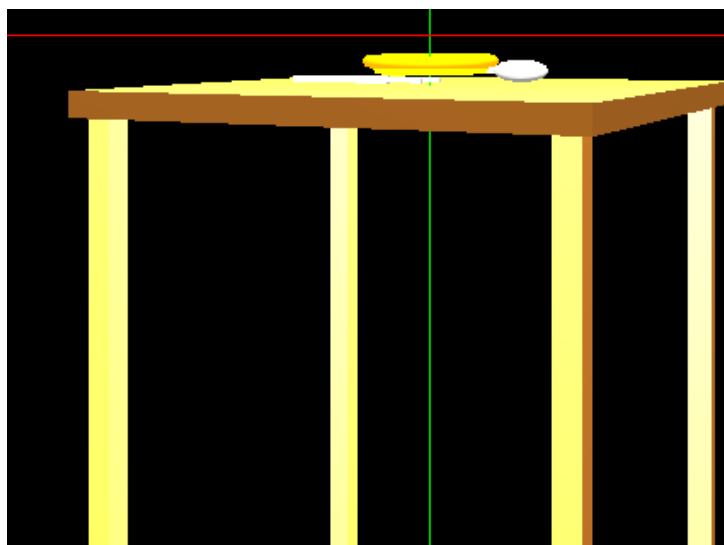
3.2 Output



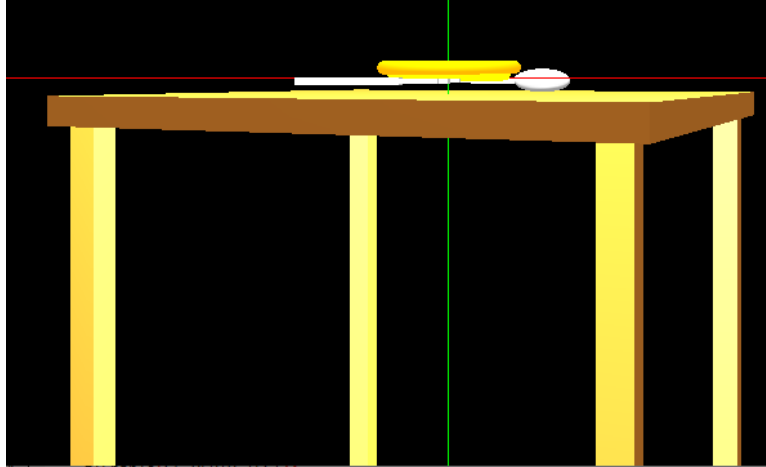
Gambar 14 Output 1



Gambar 15 Output 2



Gambar 16 Output 3



Gambar 17 Output 4



Gambar 18 Output 5

3.3 Penjelasan

a. Penjelasan Source code

Ada beberapa source code yaitu sebagai berikut:

- Objek dan Transformasi: Kode menggambar berbagai objek 3D seperti meja, piring, sendok, dan garpu menggunakan OpenGL, serta mengaplikasikan translasi, rotasi, dan skala pada objek-objek tersebut.
- Objek 3D:
 - drawTable(): Membuat meja dengan alas dan empat kaki.
 - drawConcavePlate(): Membuat piring berbentuk cekung.
 - drawSpoon(): Membuat sendok dengan gagang dan kepala sendok.
 - drawFork(): Membuat garpu dengan gagang dan tiga gigi.
- Pencahayaan: Fungsi initLighting() mengatur pencahayaan dasar untuk objek-objek 3D.
- Garis Cartesian: drawCartecius() menggambar sumbu X, Y, Z untuk membantu orientasi objek dalam ruang 3D. Garis ini dapat ditampilkan/ditutupi dengan tombol c.
- Kontrol Keyboard:

- w, a, s, d: Untuk translasi objek di sumbu X dan Z.
- r, f: Untuk rotasi objek.
- +, -: Untuk zoom in/out.
- c: Untuk menampilkan atau menyembunyikan garis Cartesian.
- z, x: Untuk rotasi kamera.
- Penyegaran Tampilan: `glutPostRedisplay()` dipanggil setelah perubahan input untuk memperbarui tampilan objek.
 Kode ini menggunakan OpenGL dan GLUT untuk menggambar dan mengontrol tampilan 3D dengan berbagai objek dan interaksi berbasis keyboard.
- b. Output yang dihasilkan
 Pada source Code di atas output yang di hasilkan adalah, bentuk objek piring, sendok. Dan garpu dengan mempunyai alas yaitu objek meja dengan mempunyai fungsi rotasi, translasi, zoom in dan zoom out,
- c. Penjelasan source code menghasilkan output seperti itu
 Source code tersebut menghasilkan output yang berupa tampilan 3D dari objek-objek seperti meja, piring, sendok, garpu, dan garis Cartesian, karena beberapa alasan utama yang terlibat dalam pemrograman OpenGL. Jadi dengan kode `drawtable` untuk membuat meja, `drawconcaveplate` untuk membuat piring, `dwarspon` untuk membuat sendok, dan `draw fork` untuk membuat garpu, sehingga terbentuklah objek objek tersebut. Selain itu ada juga translasi dengan kode `glTranslatef` untuk translasi, `glrotatef` untuk rotasi dan `glscalef` untuk skala. Serta ada pun pencahayaan pada dengan source code `glenable(GL_LIGHTING)`.

BAB IV

4.1. Kesimpulan

Tujuan dari praktikum Grafika Komputer umumnya tercapai, yaitu memberikan pemahaman tentang dasar-dasar grafika komputer, baik dalam aspek pemrograman maupun penerapan teknik visualisasi. Dalam praktikum ini, mahasiswa belajar mengenai konsep grafika komputer, algoritma yang digunakan untuk menggambar, serta bagaimana membangun aplikasi grafika baik 2D maupun 3D. Mahasiswa mempelajari berbagai teknik dasar grafika komputer seperti pemodelan, rendering, serta manipulasi gambar digital, Algoritma yang digunakan dalam grafika komputer, seperti algoritma penggambaran garis, pengisian warna, dan transformasi, Penggunaan pustaka grafika untuk menciptakan gambar 2D dan 3D yang realistis, Pengolahan gambar dan animasi dalam konteks grafika komputer. Dengan demikian, praktikum ini memberikan pengalaman yang mendalam tentang penerapan teknik grafika komputer di dunia nyata, meningkatkan keterampilan mahasiswa dalam bidang ini.

DAFTAR PUSTAKA

- [1] T. Pipit Mulyah, Dyah Aminatun, Sukma Septian Nasution, Tommy Hastomo, Setiana Sri Wahyuni Sitepu, “濟無No Title No Title No Title,” *J. GEEJ*, vol. 7, no. 2, 2020.
- [2] I. R. Taufiqy, H. W. Herwanto, and M. Wirawan, “Pengembangan Media Pembelajaran Virtual Laboratory Praktikum Grafika Komputer Pada Program Studi S1 Pendidikan Teknik Informatika,” *Semin. Nas. Teknol. Pendidik. UM*, 2015.
- [3] 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.