

LAPORAN PRATIUM GRAFIK KOMPUTER

Diajukan untuk memenuhi Tugas mata kuliah Pratikum Grafik Komputer

TUGAS BESAR – KIPAS ANGIN DAN MEJA

Dosen Pengampu : Sri Rahayu, M.Kom

Instruktur Pratikum : Arul Budi Kalimat, S.Kom



Disusun oleh

Kelompok :

Ai Nur Azizah

2306077

Desti Yashfy Silmia

2306076

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN ILMU KOMPUTER

INSTITUT TEKNOLOGI GARUT

2024

KATA PENGANTAR

Puji dan syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya sehingga kami dapat menyelesaikan Laporan Praktikum Jaringan Komputer ini. Laporan ini dibuat sebagai salah satu tugas dari mata kuliah Jaringan Komputer, dengan tujuan untuk memberikan pemahaman yang lebih baik tentang Tugas Besar – Kipas Angin dan Meja.

Kami mengucapkan terima kasih kepada dosen pengampu Sri Rahayu M.Kom, instruktur praktikum Arul Budi Kalimat, S.Kom, serta semua pihak yang telah memberikan dukungan dalam penyusunan laporan ini.

Kami menyadari bahwa laporan ini masih memiliki kekurangan, untuk itu kami mengharapkan kritik dan saran yang membangun demi perbaikan di masa yang akan datang.

Garut, 12 Januari 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR	iii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	1
1.3 Tujuan.....	1
BAB II TINJAUAN PUSTAKA	2
2.1 OpenGL	2
2.2 Konfigurasi OpenGL pada Dev C++ atau VSCode.....	2
2.3 Cara Kerja OpenGL	3
2.4 (BERIJUDUL DENGAN TEMA YANG KALIAN BUAT) Di OpenGL	3
BAB III HASIL.....	4
3.1 Source Code.....	4
3.2 Output.....	14
3.3 Penjelasan	14
BAB IV	16
4.1. Kesimpulan.....	16
DAFTAR PUSTAKA	17

DAFTAR GAMBAR

Gambar 1 menambahkan parameter	3
Gambar 2 output kipas.....	14

BAB I

PENDAHULUAN

1.1 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 display hardware (Graphics System) yaitu OpenGL. [1]

1.2 Rumusan Masalah

1. Apa yang dimaksud dengan OpenGL ?
2. Bagaimana cara mengkonfigurasi OpenGL pada Dev C++ atau VSCode?
3. Bagaimana cara kerja dari OpenGL ?
4. Bagaimana membuat Kipas Angin dan Meja dalam OpenGL

1.3 Tujuan

1. Mengetahui apa itu OpenGL
2. Mengetahui cara mengkonfigurasi OpenGL pada Dev C++ atau VSCode.
3. Mengetahui cara kerja dari OpenGL
4. Mengetahui cara pembuatan Kipas Angin dan Meja dalam OpenGL

BAB II

TINJAUAN PUSTAKA

2.1 OpenGL

OpenGL merupakan grafik standar dengan kemajuan alat yang bagus dimana tetap menjaga model program sederhana karena OpenGL hanya sebagai rendering saja. Implementasi OpenGL dapat mengefisienkan akomodasi di beberapa level grafik hardware dari dasar framebuffer sampai ke tingkat yang lebih tinggi pada sistem grafik. [2]

Open Graphics Library (OpenGL) merupakan suatu perangkat lunak yang digunakan untuk mengembangkan aplikasi grafis 2D dan 3D. Sejak diperkenalkannya pada tahun 1992 oleh Silicon Graphics Inc (SGI), OpenGL dipergunakan secara luas pada industri yang bergerak dibidang grafis 2D dan 3D dan diterapkan dalam CAD, Virtual Reality, Visualisasi Ilmiah, Visualisasi Informasi serta Simulasi Penerbangan. Perangkat lunak ini mengusung format lintas-bahasa dan lintas-platform API. Selain itu juga memiliki lebih dari 250 interface (antarmuka) yang dapat dimanfaatkan untuk membangun bentuk tiga dimensi kompleks dari bentuk sederhananya. Sebagai sebuah perangkat lunak API, OpenGL juga dirancang untuk efisiensi perangkat keras pada banyak platform yang berbeda. OpenGL Utility Library (GLU) menyediakan banyak fitur pemodelan, seperti permukaan quadric dan NURBS Curves [1]. [3]

2.2 Konfigurasi OpenGL pada Dev C++ atau VSCode

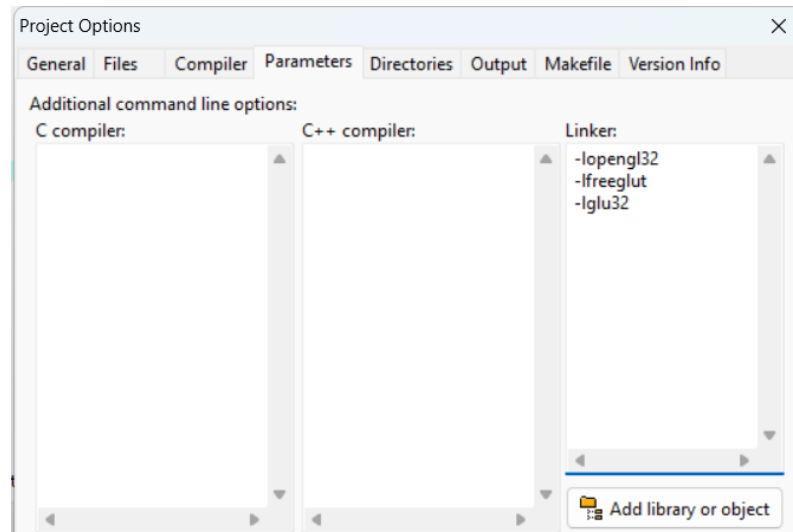
Berikut cara konfigurasi OpenGL pada Dev C++ Buka aplikasi Dev C++

1. Buka Dev C++
2. Buat Project Baru pada OpenGL, File >> New >> Project.
3. Pilih Console Application lalu klik OK
4. Klik Project >> Project Options >> Parameter
5. Tambahkan 3 parameter basic ini di bagian linker,
 - lopengl32
 - lfreeglut
 - lglu32
6. lalu klik OK.

2.3 Cara Kerja OpenGL

Berikut adalah cara kerja OpenGL di Dev-C :

1. Buka aplikasi Dev-C
2. Buat project baru (karena OpenGL dapat bekerja hanya didalam project tidak bisa di file biasa) dengan cara klik file >> pilih new >> kemudian tambahkan project
3. Setelah itu tekan ctrl+H dan pilih ikon parameter tambahkan parameter seperti dibawah. Setelah itu klik oke



Gambar 1 menambahkan parameter

Catatan : untuk beberapa kondisi parameter dapat ditambah misalnya ketika ingin menambahkan fungsi tekstur ada parameter khusus yang harus ditambahkan. Parameter diatas merupakan parameter dasar untuk menjalankan OpenGL

4. Selanjutnya tambahkan kode yang ingin dibuat.

2.4 Kipas Angin dan Meja Di OpenGL

Tema "Kipas Angin dan Meja" dipilih sebagai fokus tugas besar ini karena memberikan peluang untuk mengimplementasikan pembuatan objek 3D yang dilengkapi dengan beberapa fitur interaktif. Salah satu fitur utama adalah penggunaan fungsi keyboard untuk mengatur transformasi, skala, dan rotasi objek, sehingga pengguna dapat berinteraksi langsung dengan elemen grafis yang dibuat. Selain itu, kode juga mengintegrasikan pencahayaan untuk memberikan efek warna pada kipas angin, sementara elemen meja dan kakinya menggunakan warna yang diatur secara langsung tanpa bergantung pada pencahayaan. Dalam pembuatannya, digunakan dua teknik utama, yaitu pembuatan objek secara langsung dengan metode Solid dan Wire, serta pembuatan manual menggunakan glVertex, di mana objek dibentuk dari titik-titik koordinat. Untuk menambahkan konteks lingkungan, solid kubus dimanfaatkan untuk membentuk ruangan sebagai latar. Proyek ini tidak hanya mengintegrasikan elemen grafis dasar, tetapi juga menekankan pada penggunaan berbagai teknik OpenGL untuk menciptakan visualisasi yang interaktif dan mendalam

BAB III HASIL

3.1 Source Code

Berikut ini adalah source code yang telah kami buat :

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

void cylinder(float rbase, float rtop, float height); //ai
void blok(float tebal, int ratiol, int ratiop); // ai
void bilah (float r_inner, float r_outer, float tebal, int batang);
// ai
void kipas(); // ai
void hiddencarte(); //ai

int screen_width=600; // lebar screen ai
int screen_height=400; // tinggi screen ai
int button_up=0, button_down=0; // ai
int Turn=0; //ai

float rotation_y=0, rotation_y_plus=-15, direction; // merubah
rotasi ai
float Rhead=0, Rheadplus=0; // ai
float press=0, pressplus, pressplus1=0, pressplus2=0, pressplus3=0,
pressplus4=0, pressplus5=0; //ai
float scaleCube =70.0; //desti

bool Toleh = true, Tolehpress=true; //ai
bool RightTurn=true; //ai
bool speed1=true, speed2=false, speed3=false, speed4=false,
speed5=false; //ai
bool hidden = false; //ai

// Variabel untuk skala
float scaleFactor = 1.0; // desti

// Variabel untuk translasi
float translateX = 0.0f; // desti
float translateY = 0.0f; // desti
float translateZ = 0.0f; // desti

// Variabel untuk rotasi
float rotateX = 0.0f; //ai
float rotateY = 0.0f; //ai

// setting pencahayaan kipas angin
GLfloat ambient_light[]={0.0,0.0,0.15,1.0}; //ai
GLfloat source_light[]={0.9,0.5,0.7,1.0}; //ai
GLfloat light_pos[]={0.0,0.0,0.0,1.0}; //ai

void cartesius() { //ai
    glBegin(GL_LINES);

    glColor3f(1.0, 0.0, 0.0); // Warna merah
```



```

    glVertex3f(-100.0, 0.0, 0.0);
    glVertex3f(100.0, 0.0, 0.0);

    glColor3f(0.0, 1.0, 0.0); // Warna hijau
    glVertex3f(0.0, -100.0, 0.0);
    glVertex3f(0.0, 100.0, 0.0);

    glColor3f(0.0, 0.0, 1.0); // Warna biru
    glVertex3f(0.0, 0.0, 100.0);
    glVertex3f(0.0, 0.0, -100.0);

    glEnd();
}

void hiddencarte(){ //ai
    if (hidden){
        cartesius();
    }
}

void Cube() { //desti
    glPushMatrix();
    glDisable(GL_LIGHTING);
    glScalef(scaleCube, scaleCube, scaleCube); //skala
    glBegin(GL_QUADS);

    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);

    // Sisi belakang
    glColor3f(0.0, 1.0, 0.0); // Hijau
    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);

    // Sisi kiri
    glColor3f(0.0, 0.0, 1.0); // Biru
    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);

    // Sisi kanan
    glColor3f(1.0, 1.0, 0.0); // Kuning
    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);

    // Sisi atas
    glColor3f(1.0, 0.5, 0.5); // Merah muda
    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);

// Sisi bawah
glColor3f(0.5, 0.5, 0.5); // Abu-abu
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);

glEnd(); // Mengakhiri menggambar kubus

glPopMatrix();
glEnable(GL_LIGHTING);
}

void meja(){ //desti
    glPushMatrix();
    glTranslatef(translateX, translateY, translateZ);
//transformasi
    glRotatef(rotateX, 1.0f, 0.0f, 0.0f);
    glRotatef(rotateY, 0.0f, 1.0f, 0.0f);
    glScalef(scaleFactor, scaleFactor, scaleFactor); // skala
    glScalef(1.0, 1.0, 1.0); // Skala normal

    // alas meja
    glPushMatrix();
    glColor3d(0.88, 0.72, 0.53); //coklat muda
    glTranslated(0.0, 0.0, 0.0);
    glScaled(1.5, 0.2, 1.6);
    glutSolidCube(10);
    glPopMatrix();

    // kaki meja
    glColor3d(0.88, 0.72, 0.53);
    glPushMatrix();
    glTranslatef(-7.5, -7.0, -8.0); // kaki kiri depan
    glScalef(0.2, 1.5, 0.2);
    glutSolidCube(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(7.5, -7.0, -8.0); // kaki kanan depan
    glScalef(0.2, 1.5, 0.2);
    glutSolidCube(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(-7.5, -7.0, 8.0); // kaki kiri belakang
    glScalef(0.2, 1.5, 0.2);
    glutSolidCube(10);
    glPopMatrix();

    glPushMatrix();
    glTranslatef(7.5, -7.0, 8.0); // kaki kanan belakang

```

```

        glScalef(0.2, 1.5, 0.2);
        glutSolidCube(10);
        glPopMatrix();

        glPopMatrix();
    }

void init(void) //ai
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    gluPerspective(45.0f, (GLfloat)screen_width/(GLfloat)screen_height,1
    .0f,1000.0f);
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_LIGHTING);
    glLightModelfv(GL_LIGHT_MODEL_AMBIENT, ambient_light);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, source_light);
    glLightfv(GL_LIGHT0, GL_POSITION, light_pos);
    glEnable(GL_LIGHT0);
    glEnable(GL_COLOR_MATERIAL);
    glColorMaterial(GL_FRONT, GL_AMBIENT_AND_DIFFUSE);
}

// fungsi proposional maximaze
void resize(int width, int height)
{
    screen_width=width;
    screen_height=height;

    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    glViewport(0,0,screen_width,screen_height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    gluPerspective(55.0f, (GLfloat)screen_width/(GLfloat)screen_height,1
    .0f,1000.0f);

    glutPostRedisplay();
}

void display(void) //ai
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    glTranslatef(0.0, 0.0, -70.0);
    glPushMatrix();
    glTranslated(0.0, -3.0, 0.0);
    hiddencarte();
    meja();
    glPopMatrix();

    glPushMatrix();

```

```

    Cube();

    // Terapkan translasi berdasarkan posisi yang diperbarui
    glTranslatef(translateX, translateY, -10.0);
    glScalef(scaleFactor, scaleFactor, scaleFactor);

    glRotatef(270, 1.0, 0.0, 0.0);

    kipas();
    glPopMatrix();

    glutSwapBuffers();
}

void kipas(){ //ai
    glPushMatrix();
    cylinder(2.5, 1.5, 16); // silinder baling bawah 2
    cylinder(2.5, 2.5, 6); // silinder baling bawah 1
    glTranslatef(0.0, 0.0, 14.0);
    glRotatef(90, 0.0, 1.0, 0.0);

    glPushMatrix();
    glRotatef(270, 0.0, 1.0, 0.0);
    glTranslatef(0.0, 0.0, 1.0);
    cylinder(0.5, 1, 4); // silinder baling atas leher
    glPopMatrix();

    glutSolidTorus(1.5, 2, 6, 16);
    glTranslatef(0.0, 0.0, -2.0);
    cylinder(1, 1, 4.3); // silinder penghubung baling atas dan
    baling bawah
    glTranslatef(0.0, 0.0, 2.0);
    glRotatef(270, 0.0, 1.0, 0.0);

    glPushMatrix();
    glTranslatef(0, 0, 10); //bilah kipas
    glRotatef(100, 1.0, 0.0, 0.0); //bilah kipas

    if (Toleh == true) {
        if (Turn >= 60) // toleh kanan
            RightTurn = false;
        if (Turn <= -60) //toleh kiri
            RightTurn = true;
        if (RightTurn == true) {
            Rheadplus++; //kecepatan kipas
            Turn++; //nilai toleh kipas
        } else {
            Rheadplus--;
            Turn--;
        }
    }
    Rhead = Rhead + Rheadplus;
    glRotatef(Rhead, 0.0, 1.0, 0.0); //rotasi toleh kipas
    Rheadplus = 0;

    glTranslatef(0, 0, -3);
    cylinder(4, 4, 6); // silinder belakang kipas

```

```

cylinder(1, 0.5, 15); // silinder tonjolan di depan kipas
glRotatef(270, 1.0, 0.0, 0.0);

    if (Tolehpres == true) // tombol baling baling
        cylinder(0.3, 0.5, 6);
    else
        cylinder(0.3, 0.5, 7);

glRotatef(90, 1.0, 0.0, 0.0); //bilah kipas
glPushMatrix();
glTranslatef(0.0, 0.0, 11);
glutWireTorus(5, 7, 10, 64);
glutSolidTorus(0.5, 12, 10, 64);
rotation_y += rotation_y_plus; //perputaran baling baling
if (rotation_y > 359) rotation_y = 0;
glRotatef(rotation_y, 0.0, 0.0, 1.0);

    bilah(3, 10, 3, 5); // bilah kipas (inner radius, outer radius,
ketebalan, jumlah bilah)
glPopMatrix();
glPopMatrix();
glPopMatrix();
glRotatef(90, 1.0, 0.0, 0.0);
glTranslatef(0.0, -1.0, -4);

    blok(2, 7, 10); // blok dasar tombol (teba, p, l)

// speed selection
glTranslatef(-6, 1, 14); //posisi awal tombol
glRotatef(270, 1.0, 0.0, 0.0);
glTranslatef(2.0, 0.0, 0.0);

    glPushMatrix();
glRotatef(pressplus1, 1.0, 0.0, 0.0);
blok(0.5, 2, 2); // blok tombol speed 1
glPopMatrix();

glTranslatef(2.0, 0.0, 0.0);

glPushMatrix();
glRotatef(pressplus2, 1.0, 0.0, 0.0);
blok(0.5, 2, 2); // blok tombol speed 2
glPopMatrix();

glTranslatef(2.0, 0.0, 0.0);

glPushMatrix();
glRotatef(pressplus3, 1.0, 0.0, 0.0);
blok(0.5, 2, 2); // blok tombol speed 3
glPopMatrix();
glTranslatef(2.0, 0.0, 0.0);

    glPushMatrix();
glRotatef(pressplus4, 1.0, 0.0, 0.0);
blok(0.5, 2, 2); // blok tombol speed 4
glPopMatrix();

```

```

    pressplus5 = 0;

}

// membuat silinder
void cylinder(float rbase, float rtop, float height)
{
    float i;
    glColor3d(1.0,1.0,1.0); //warna putih
    glPushMatrix();
    glTranslatef(0.0, 0.0, -rbase / 4);
    glutSolidCone(rbase, 0, 32, 4); //bagian bawah
    for (i = 0; i <= height; i += rbase / 8) //tengah cincin
    {
        glTranslatef(0.0, 0.0, rbase / 8);
        glutSolidTorus(rbase / 4, rbase - ((i * (rbase - rtop)) /
height), 16, 16);
    }
    glTranslatef(0.0, 0.0, rbase / 4);
    glutSolidCone(rtop, 0, 32, 4);
    glPopMatrix();
}

//membuat bilah kipas
void bilah(float r_inner, float r_outer, float tebal, int baling)
//ai
{
    float i;
    glPushMatrix();
    glTranslatef(0.0, 0.0, -tebal / 4);
    cylinder(r_inner, r_inner, tebal);
    glTranslatef(0.0, 0.0, tebal / 2); //memisahkan bagian atas dan
bawah
    glRotatef(90, 0.0, 1.0, 0.0);
    for (i = 0; i < baling; i++) //baling
    {
        glTranslatef(0.0, 0.0, r_inner);
        glRotatef(315, 0.0, 0.0, 1.0);
        blok(0.5, r_inner * 4.5, (r_outer - r_inner + (r_inner /
4)) * 2);
        glRotatef(45, 0.0, 0.0, 1.0);
        glTranslatef(0.0, 0.0, -r_inner);
        glRotatef(360 / baling, 1.0, 0.0, 0.0);
    }
    glPopMatrix();
}

// method untuk membuat blok kipas angin
void blok(float tebal, int ratiol, int ratiop) //ai
{
    float i, j;
    glPushMatrix();
    for (i = 0; i < ratiop; i++)
    {
        glTranslatef(-(ratiol + 1) * tebal / 2, 0.0, 0.0);
        for (j = 0; j < ratiol; j++)

```

```

        {
            glTranslatef(tebal, 0.0, 0.0);
            glutSolidCube(tebal);
        }
        glTranslatef(-(ratio1 - 1) * tebal / 2, 0.0, tebal);
    }
    glPopMatrix();
}

// fungsi keyboard
void keyboard_s(unsigned char key, int x,      int y)
{
    if (rotation_y_plus != 0) // ai, untuk memastikan baling bisa
on off
        direction = (rotation_y_plus / abs(rotation_y_plus));
    else
        direction = -1;

    switch (key)
    {
    case '1': // speed pertama
        if (!speed1)
        {
            rotation_y_plus = 15 * direction;
            speed1 = true;
            pressplus1 = 180;
            speed2 = false;
            pressplus2 = 0;
            speed3 = false;
            pressplus3 = 0;
            speed4 = false;
            pressplus4 = 0;
            if (Tolehpress == true)
                Toleh = true;
        }
        else
        {
            speed1 = false;
            pressplus1 = 0;
            rotation_y_plus = 0;
            Toleh = false;
        }
        break;

    case '2': // speed kedua
        if (!speed2)
        {
            rotation_y_plus = 30 * direction;
            speed1 = false;
            pressplus1 = 0;
            speed2 = true;
            pressplus2 = 180;
            speed3 = false;
            pressplus3 = 0;
            speed4 = false;
            pressplus4 = 0;
            if (Tolehpress == true)

```

```

        Toleh = true;
    }
    else
    {
        speed2 = false;
        pressplus2 = 0;
        rotation_y_plus = 0;
        Toleh = false;
    }
    break;

case '3': // speed ketiga
    if (!speed3)
    {
        rotation_y_plus = 45 * direction;
        speed1 = false;
        pressplus1 = 0;
        speed2 = false;
        pressplus2 = 0;
        speed3 = true;
        pressplus3 = 180;
        speed4 = false;
        pressplus4 = 0;
        if (Tolehpres == true)
            Toleh = true;
    }
    else
    {
        speed3 = false;
        pressplus3 = 0;
        rotation_y_plus = 0;
        Toleh = false;
    }
    break;

case '4': // speed keempat
    if (!speed4)
    {
        rotation_y_plus = 60 * direction;
        speed1 = false;
        pressplus1 = 0;
        speed2 = false;
        pressplus2 = 0;
        speed3 = false;
        pressplus3 = 0;
        speed4 = true;
        pressplus4 = 180;
        if (Tolehpres == true)
            Toleh = true;
    }
    else
    {
        speed4 = false;
        pressplus4 = 0;
        rotation_y_plus = 0;
        Toleh = false;
    }
}

```



```

        break;

    case '5': // menghentikan toleh kipas kiri dan kanan
        if (Tolehpress == false)
        {
            if (speed1 || speed2 || speed3 || speed4)
                Toleh = true;
            Tolehpress = true;
        }
        else
        {
            if (speed1 || speed2 || speed3 || speed4)
                Toleh = false;
            Tolehpress = false;
        }
        break;

    case 'i': // meningkatkan skala , desti
        scaleFactor += 0.1f; // meningkatkan skala
        break;

    case 'x': //agar tidak bernilai negatif
        if (scaleFactor > 0.1f)
            scaleFactor -= 0.1f;
        break;

    case 'w': // bergerak ke atas
        translateY += 0.5f;
        break;

    case 's': // bergerak ke bawah
        translateY -= 0.5f;
        break;

    case 'a': // bergerak ke kiri
        translateX -= 0.5f;
        break;

    case 'd': // bergerak ke kanan
        translateX += 0.5f;
        break;

    case 'q': // rotasi ke kiri
        rotateY -= 5.0f;
        break;

    case 'e': // rotasi ke kanan
        rotateY += 5.0f;
        break;

    case '+':
        scaleCube += 1.0f;
        break;

    case '-':
        if (scaleCube > 1.0f) // pastikan skala tidak negatif
            scaleCube -= 1.0f;
        break;

    case 'h': //ai

```

```

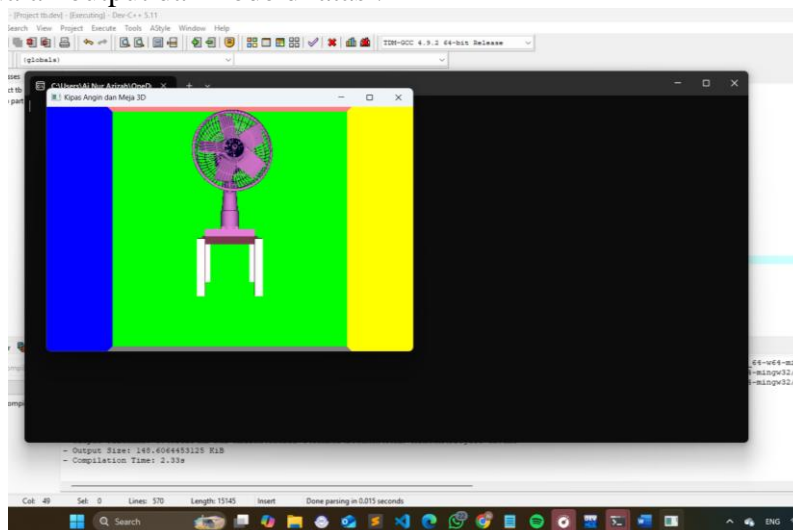
        hidden= !hidden;
        break;
    }
    glutPostRedisplay();
}

int main(int argc, char **argv) //desti
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
//desti
    glutInitWindowSize(screen_width, screen_height); //ai
    glutInitWindowPosition(100,100); //desti
    glutCreateWindow("Kipas Angin dan Meja 3D"); //desti
    glutDisplayFunc(display); //ai
    glutIdleFunc(display); //ai
    glutReshapeFunc(resize); //ai
    glutKeyboardFunc(keyboard_s); //desti
    init();//ai
    glutMainLoop(); //ai
    return 0;
}

```

3.2 Output

Berikut ini adalah output dari kode di atas :



Gambar 2 output kipas

3.3 Penjelasan

Pada kode di atas, terdapat dua objek utama yang digambar menggunakan OpenGL: sebuah kipas angin dan meja. Kipas angin terdiri dari beberapa elemen, termasuk silinder untuk batang kipas, torus untuk bilah kipas, serta objek-objek kecil lainnya untuk menyusun bagian-bagian kipas. Fungsi utama dalam pembuatan kipas angin adalah kipas(), yang membuat objek-objek tersebut dengan rotasi dan translasi yang ditentukan untuk menciptakan efek pergerakan kipas. Proses rotasi kipas dilakukan menggunakan variabel Rhead, yang mempengaruhi rotasi kipas kanan-kiri, serta variabel Angguk untuk pergerakan vertikal.

Pada objek meja, proses pembuatan dilakukan dengan menggunakan objek kubus yang disesuaikan skalanya dan posisinya. Fungsi `meja()` menggambar meja dengan elemen-elemen seperti alas meja dan empat kaki meja yang masing-masing digambar menggunakan objek kubus. Translasi dan rotasi dilakukan untuk memposisikan meja sesuai dengan skala dan orientasi yang diinginkan. Setiap bagian dari meja (alas dan kaki) digambar secara terpisah dengan transformasi untuk menyesuaikan posisi relatif antar bagian-bagian meja.

Kedua objek ini, kipas angin dan meja, diposisikan di dalam dunia 3D dengan menggunakan fungsi transformasi seperti `glTranslatef()`, `glRotatef()`, dan `glScalef()`, yang memungkinkan pengaturan posisi, rotasi, dan skala objek-objek tersebut. Program ini juga mengaktifkan pencahayaan OpenGL dengan mengatur berbagai sumber cahaya dan bahan pencahayaan untuk menghasilkan efek pencahayaan yang realistis pada objek-objek 3D yang digambar.

BAB IV

4.1. Kesimpulan




Laporan ini membahas pembuatan objek 3D (Kipas Angin dan Meja) menggunakan OpenGL. Diawal dijelaskan apa itu grafika komputer dan peran. Tujuan dari laporan ini adalah untuk memahami konsep OpenGL, cara mengonfigurasinya di Dev C++, juga proses pembuatan objek 3D terutama kipas angin dan meja. Selain itu pada laporan ini juga menjelaskan lebih dalam tentang OpenGL sebagai perangkat lunak standar untuk aplikasi grafis 2D dan 3D, serta langkah-langkah konfigurasi dan cara kerjanya. Tema "Kipas Angin dan Meja" dipilih untuk mengimplementasikan objek 3D interaktif dengan fitur seperti transformasi objek melalui fungsi keyboard. Kipas angin dibuat dengan berbagai elemen seperti silinder dan torus, sedangkan meja dibentuk dari objek kubus. Keduanya menggunakan teknik transformasi untuk mengatur posisi dan rotasi dalam ruang 3D. Pencahayaan juga diterapkan untuk memberikan efek yang lebih menarik pada objek. Secara keseluruhan, laporan ini menunjukkan bagaimana teori grafika komputer bisa diterapkan dalam praktik menggunakan OpenGL, serta kemampuan untuk menciptakan objek 3D yang menarik dan interaktif dengan berbagai teknik grafis yang mendalam.

DAFTAR PUSTAKA

- [1] J. Widadi and A. Dahlan Jl ProfDrSoepomo, “Media Pembelajaran Materi Pengenalan Opengl Pada Mata Kuliah Grafika Komputer,” *J. Sarj. Tek. Inform.*, vol. 6, no. 1, pp. 47–53, 2018, [Online]. Available: <http://journal.uad.ac.id/index.php/JSTIF>
- [2] “jbptunikompp-gdl-s1-2005-mohamadzai-1752-bab-i (1).”
- [3] R. Iskandar, R. Candra, and O. Font, “PEMBUATAN GARIS BESAR FONT (OUTLINE FONTS) MENGGUNAKAN PROGRAM OPENGL 32 Kata kunci,” pp. 25–28, 2013.

ASISTENSI TUGAS PRAKTIKUM GRAFIK KOMPUTER

Nama Pembimbing : Irpan
Kelompok & Kelas : Kelompok 3 Informatika C
Anggota Kelompok : - Ai Nur Azizah (2306077)
 - Desti Yashfy Silmia (2306076)
Tema : Kipas Angin + Meja

No	Tanggal	Uraian Asistensi	Tanda Tangan
1	28/12	progress pembuatan awal (prototype, tema, obsec)	
2	4/1	progress, target penyesuaian tampilan	
3	10/1	cartesian, nambah kubus	
4			
5			

Instruktur Praktikum



Arul Budi Kalimat, S.Kom