

Laporan Akhir Praktikum

Mata Praktikum : Grafika Komputer 1

Kelas : 3IA01

Praktikum Ke- : 1

Tanggal : 20 Oktober 2020

Materi : Syntax dasar OpenGL

NPM : 53418015

Nama : Haikal Ardikatama

Ketua Asisten : Diokta Redho Lastin

Paraf Asisten :

Nama Asisten :

Jumlah Lembar : 9

Laboratorium Teknik Informatika
Universitas Gunadarma
2020

Listing Program

```
#include <iostream>
#include <gl/glut.h>
/*
Author : Haikal Ardikatama
Date   : 20/10/2020
Title  : Final Report 1
*/

void draw(){
    static int thick = 10;
    glColor3f(1.0f, 0.5f, 0.0f); //Orange
    glPointSize(10);

    //Define vertex sesuai posisi yang diinginkan
    glBegin(GL_POINTS);
    glVertex2i(0, 0);
    glVertex2i(0, 50);
    glVertex2i(0, -50);
    glVertex2i(50, 0);
    glVertex2i(-50, 0);
    glEnd();

    //Menggambar persegi yang paling besar
    glBegin(GL_LINES);
    glVertex2i(-200, -200);
    glVertex2i(-200, 200);
    glEnd();

    glBegin(GL_LINES);
    glVertex2i(200, -200);
    glVertex2i(200, 200);
    glEnd();

    glBegin(GL_LINES);
    glVertex2i(-200, 200);
    glVertex2i(200, 200);
    glEnd();
```

```

glBegin(GL_LINES);
glVertex2i(-200, -200);
glVertex2i(200, -200);
glEnd();

//Menggambar garis-garis horizontal di dalam persegi
glBegin(GL_LINES);
glVertex2i(-200, 25);
glVertex2i(200, 25);
glEnd();

glBegin(GL_LINES);
glVertex2i(-200, -25);
glVertex2i(200, -25);
glEnd();

//Menggambar garis-garis vertical di dalam persegi
glBegin(GL_LINES);
glVertex2i(-25, -200);
glVertex2i(-25, 200);
glEnd();

glBegin(GL_LINES);
glVertex2i(25, -200);
glVertex2i(25, 200);
glEnd();
}

void display(void) {
    glClear(GL_COLOR_BUFFER_BIT);
    draw();
    glutSwapBuffers();
}

```

```

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB);
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Final Report 1");
    glClearColor(1.0, 1.0, 1.0, 0.0);
    gluOrtho2D(-320, 320, -240, 240);
    glutIdleFunc(display);
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```

Logika Program

Pada pertemuan kali ini membahas mengenai penggunaan library OpenGL di dalam C++. Tugas laporan akhir yang diberikan adalah membuat 5 titik dan menggunakan garis untuk menggambar bentuk yang telah diberikan oleh PJ.

```

#include <iostream>
#include <gl/glut.h>
/*
Author : Haikal Ardikatama
Date   : 20/10/2020
Title  : Final Report 1
*/

```

Penjelasan :

Potongan kodingan di atas adalah syntax untuk penggunaan library dan terdapat komentar yang menjelaskan tentang nama penulis, tanggal, dan judul dari program yang akan dibuat.

```
void draw(){
    static int thick = 10;
    glColor3f(1.0f, 0.5f, 0.0f); //Orange
    glPointSize(10);

    //Define vertex sesuai posisi yang diinginkan
    glBegin(GL_POINTS);
    glVertex2i(0, 0);
    glVertex2i(0, 50);
    glVertex2i(0, -50);
    glVertex2i(50, 0);
    glVertex2i(-50, 0);
    glEnd();

    //Menggambar persegi yang paling besar
    glBegin(GL_LINES);
    glVertex2i(-200, -200);
    glVertex2i(-200, 200);
    glEnd();

    glBegin(GL_LINES);
    glVertex2i(200, -200);
    glVertex2i(200, 200);
    glEnd();

    glBegin(GL_LINES);
    glVertex2i(-200, 200);
    glVertex2i(200, 200);
    glEnd();

    glBegin(GL_LINES);
    glVertex2i(-200, -200);
    glVertex2i(200, -200);
    glEnd();

    //Menggambar garis-garis horizontal di dalam persegi
    glBegin(GL_LINES);
    glVertex2i(-200, 25);
    glVertex2i(200, 25);
    glEnd();

    glBegin(GL_LINES);
    glVertex2i(-200, -25);
    glVertex2i(200, -25);
    glEnd();

    //Menggambar garis-garis vertical di dalam persegi
    glBegin(GL_LINES);
    glVertex2i(-25, -200);
    glVertex2i(-25, 200);
    glEnd();

    glBegin(GL_LINES);
    glVertex2i(25, -200);
    glVertex2i(25, 200);
    glEnd();
}
```

Penjelasan :

Potongan kodingan di atas adalah kodingan mengenai function draw yang bertipe void. Fungsi dari function draw adalah untuk menggambar bentuk yang telah diinginkan. Di dalam fungsi draw terdapat beberapa blok program yang ditandai dengan komentar-komentar yang telah diberikan.

Bagian awal dari fungsi ini adalah pendefinisian ketebalan dari titik dan pendeklarasian warna yang akan digunakan di program ini.

Bagian kedua adalah kodingan yang digunakan untuk mendefine vertex-vertex sesuai dengan posisi yang kita inginkan dalam kasus ini terdapat 5 vertex yang dapat dijabarkan menjadi vertex1(0,0) , vertex2(0,50) , vertex3(0, -50), vertex4(50, 0), dan vertex5(-50, 0).

Bagian ketiga adalah kodingan yang digunakan untuk menggambar persegi. Penggambaran persegi di program ini memanfaatkan fungsi dari library OpenGL yaitu glBegin(GL_LINES). Fungsi tersebut merupakan fungsi yang digunakan untuk menggambar garis. Fungsi ini memerlukan koordinat x1,y1 dan koordinat x2,y2 yang dapat didefinisikan menggunakan fungsi glVertex2i(x, y). Garis-garis yang diperlukan untuk menggambar persegi di bagian ketiga antara lain :

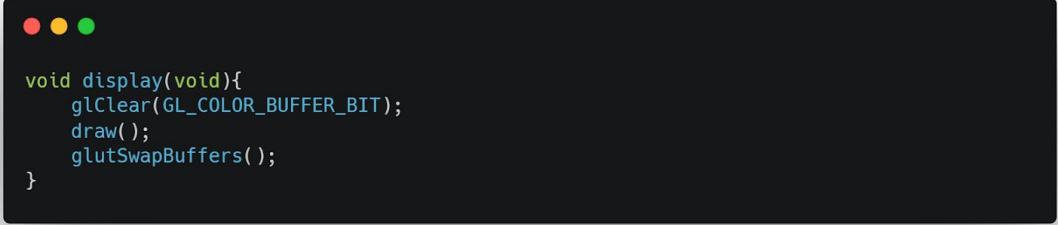
1. Garis 1 dari vertex (-200, -200) ke vertex(-200, 200).
2. Garis 2 dari vertex (200, -200) ke vertex(200, 200).
3. Garis 3 dari vertex (-200, 200) ke vertex(200, 200).
4. Garis 4 dari vertex (-200, -200) ke vertex(200, -200).

Bagian keempat adalah kodingan yang digunakan untuk menggambar 2 garis horizontal di dalam persegi. Penggambaran garis memanfaatkan fungsi yang sama seperti bagian ketiga. Garis-garis yang diperlukan untuk bagian keempat antara lain :

1. Garis 1 dari vertex (-200, 25) ke vertex(200, 25).
2. Garis 2 dari vertex (-200, -25) ke vertex(200, -25).

Bagian kelima adalah kodingan yang digunakan untuk menggambar 2 garis vertikal di dalam persegi. Penggambaran garis memanfaatkan fungsi yang sama seperti bagian ketiga. Garis-garis yang diperlukan untuk bagian keempat antara lain :

1. Garis 1 dari vertex (-25,-200) ke vertex(-25, 200).
2. Garis 2 dari vertex (25, -200) ke vertex(25, 200).



```
void display(void){  
    glClear(GL_COLOR_BUFFER_BIT);  
    draw();  
    glutSwapBuffers();  
}
```

Penjelasan :

Potongan kodingan di atas adalah kodingan fungsi display. Fungsi ini berguna untuk menampilkan hasil output di layar. Fungsi display terdiri dari 3 fungsi yaitu

1. `glClear(GL_COLOR_BUFFER_BIT)`
Fungsi ini berguna untuk menghapus buffer yang terdapat di layar. Dalam program ini fungsi `glClear` memiliki input `GL_COLOR_BUFFER_BIT` yang merupakan sebuah konstanta dari library OpenGL yang mengindikasikan buffer mengenai warna.
2. `draw()`
Fungsi ini berguna untuk menggambarkan bentuk yang diinginkan.
3. `glutSwapBuffers()`
Fungsi ini berguna untuk melakukan pertukaran buffer pada lapisan yang digunakan untuk jendela saat ini.

```

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB);
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Final Report 1");
    glClearColor(1.0, 1.0, 1.0, 0.0);
    gluOrtho2D(-320, 320, -240, 240);
    glutIdleFunc(display);
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```

Penjelasan :

Potongan kodingan di atas merupakan potongan kodingan dari fungsi main. Fungsi ini adalah fungsi yang dijalankan ketika kita menjalankan program. Fungsi main berisi beberapa fungsi antara lain adalah :

1. `glutInit(int argc, char**argv)`
`glutInit` akan menginisialisasi pustaka GLUT dan menegosiasikan sesi dengan sistem jendela. Selama proses ini, `glutInit` dapat menyebabkan penghentian program GLUT dengan pesan kesalahan kepada pengguna jika GLUT tidak dapat diinisialisasi dengan benar. Contoh situasi ini termasuk kegagalan untuk terhubung ke sistem jendela, kurangnya dukungan sistem jendela untuk OpenGL, dan opsi baris perintah yang tidak valid.
2. `glutInitDisplayMode()`
Fungsi ini digunakan untuk menentukan apakah akan menggunakan model pewarnaan RGBA atau indeks warna. Dengan fungsi ini juga bisa menentukan apakah akan menggunakan windows buffer single atau double.
3. `glutInitWindowSize(int width, int height)`
Fungsi ini digunakan untuk menspesifikasikan besar ukuran layar dalam pixel. dalam kasus ini ukuran layarnya adalah 640 x 480.
4. `glutInitWindowPosition(int x, int y)`
Fungsi ini menentukan letak window dengan patokannya adalah pada window kiri atas. Dalam kasus ini $x = 100$ dan $y = 100$

5. `glutCreateWindow(char *string);`
fungsi untuk membuat window dalam konteks openGL, dan akan menghasilkan identitas atau judul window tersebut. Dalam kasus ini judulnya adalah final report 1.
6. `glClearColor()`
Fungsi ini digunakan untuk memilih warna yang digunakan untuk membersihkan latar dalam mode RGBA. Dalam kasus ini warna yang digunakan adalah putih dengan opacity 0.
7. `gluOrtho2D`
Suatu fungsi untuk mengatur proyeksi hasil eksekusi dan mendefinisikan besarnya sistem koordinat dengan urutan kiri-kanan dan bawah-atas. fungsi ini bertujuan menentukan titik 0,0 di dalam layar tersebut.
8. `glutIdleFunc(display)`
Pernyataan ini digunakan untuk refresh setiap proses menggambar selesai dan biasanya digunakan untuk membuat animasi. Untuk gambar yang tidak bergerak, pernyataan ini tidak digunakan.
9. `glutDisplayFunc(display)`
Digunakan untuk memanggil fungsi display atau method untuk membuat objek primitive. Dalam kasus ini memanggil fungsi display yang sudah didefine sebelumnya.
10. `glutMainLoop()`
Fungsi ini berguna untuk menandakan pemrosesan glut terjadi.

Output

