

PRAKTIKUM GRAFIKA KOMPUTER

Pertemuan ke-2 (Primitive Drawing)



Yoga Agustiansyah
2206050

Jurusan Ilmu Komputer
Program Studi Teknik Informatika
Institut Teknologi Garut
Jl. Mayor Syamsu No. 1 Jayaraga Garut 44151 Indonesia

I. PENDAHULUAN

Pada praktikum ini, akan dibahas mengenai Primitive Drawing, sebuah konsep dalam pemrograman grafis yang memungkinkan kita untuk membuat gambar sederhana menggunakan garis dan bentuk dasar lainnya. Primitive Drawing mengacu pada penggunaan elemen-elemen grafis dasar, seperti titik, garis, lingkaran, atau persegi, untuk membentuk objek yang lebih kompleks. Dalam konteks ini, "primitive" merujuk pada bentuk-bentuk dasar ini.

Primitive Drawing merupakan fondasi penting dalam pengembangan aplikasi grafis komputer. Dengan memahami konsep ini, kita dapat membangun objek-objek grafis yang lebih kompleks dan menghasilkan visualisasi yang menarik.

Pada praktikum ini, fokus utama akan berada pada penggunaan garis sebagai salah satu bentuk dasar dalam Primitive Drawing. Elemen garis akan digunakan untuk membuat gambar dengan tiga garis horizontal yang bertumpuk secara vertikal, dengan masing-masing garis memiliki warna yang berbeda.

II. PEMBAHASAN

A. Source Code

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

void garis () {
    glClear (GL_COLOR_BUFFER_BIT);
    glLineWidth (10);

    glBegin(GL_LINES);

    glColor3f(0.0, 0.51, 0.85);
    glVertex2f(100,100);
    glColor3f(1, 0, 0.701);
    glVertex2f(-100,100);

    glColor3f(1, 0, 0.701);
    glVertex2f(100,50);
    glVertex2f(-100,50);

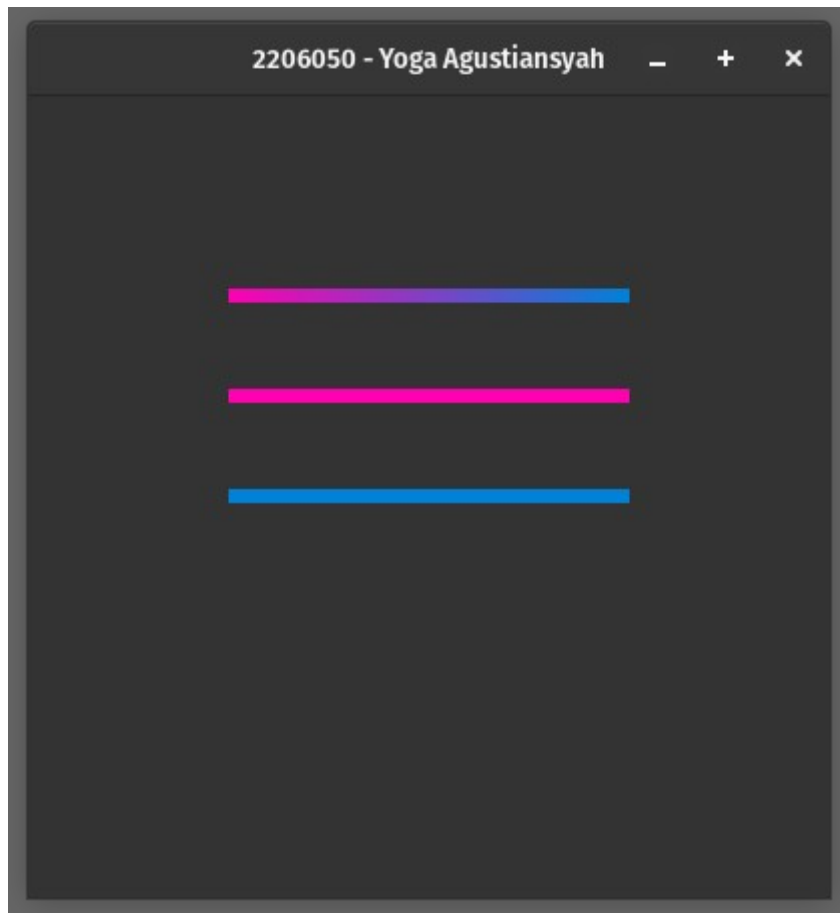
    glColor3f(0.0, 0.51, 0.85);
    glVertex2f(100,0);
    glVertex2f(-100,0);

    glEnd ();

    glFlush();
    glFinish ();
}

int main(int argc, char**argv) {
    glutInit (&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition(100,100);
    glutCreateWindow ("2206050 - Yoga Agustiansyah");
    glClearColor(0.2, 0.2, 0.2, 1.0);
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(-200, 200, -200, 200);
    glutDisplayFunc (garis);
    glutMainLoop();
    return 0;
}
```

B. Output



Dari kode yang sudah dibuat, kita mendapatkan output berupa 3 buah garis yang memanjang secara horizontal dan bersusun secara vertikal. Ketiga garis tersebut memiliki 3 warna berbeda, garis paling atas memiliki warna gradien magenta dan cyan, garis dengan posisi di tengah berwarna magenta, dan garis yang paling bawah memiliki warna cyan. Ketiga garis tersebut ditampilkan dengan sebuah window bernama “2206050 – Yoga Agustiansyah ”dengan background abu-abu gelap.

C. Penjelasan Program

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

penyertaan direktif preprosesor yang mengimpor header file dari library GLUT (OpenGL Utility Toolkit). Library ini digunakan untuk membuat tampilan grafis dalam program.

```
void garis () {  
    glClear (GL_COLOR_BUFFER_BIT);  
    glLineWidth (10);
```

definisi dari fungsi **garis()**. Fungsi ini akan digunakan untuk menggambar garis-garis pada tampilan grafis. **glClear(GL_COLOR_BUFFER_BIT)** menghapus tampilan grafis sebelum menggambar yang baru, dan **glLineWidth(10)** mengatur ketebalan garis menjadi 10 unit.

```
glBegin(GL_LINES);
```

glBegin(GL_LINES) menandakan awal dari definisi garis-garis. Dalam blok ini, akan didefinisikan garis-garis yang akan digambar.

```
glColor3f(0.0, 0.51, 0.85);  
glVertex2f(100,100);  
glColor3f(1, 0, 0.701);  
glVertex2f(-100,100);
```

- **glColor3f(r, g, b)** digunakan untuk mengatur warna yang akan digunakan dalam gambar. r, g, dan b adalah komponen warna merah, hijau, dan biru dalam format float antara 0.0 hingga 1.0. Pada program ini, kode warna (0.0, 0.51, 0.85) akan menghasilkan warna cyan, dan kode warna (1, 0, 0.701) akan menghasilkan warna magenta, karena ditempatkan diantara 2 titik awal dan akhir garis, perubahan kode ini akan menghasilkan warna gradien.
- **glVertex2f(x, y)** digunakan untuk mendefinisikan titik awal dan akhir dari garis. Dalam kasus ini, didefinisikan dua titik untuk membuat garis horizontal dengan warna yang berbeda. Titik awalnya adalah (100,100) dan titik akhir garis adalah (-100,100), ini menandakan garis akan datarik sepanjang 200 titik, dari kanan (x bernilai positif) ke kiri (x bernilai negatif).

```
glColor3f(1, 0, 0.701);  
glVertex2f(100,50);  
glVertex2f(-100,50);
```

- **glColor3f(r, g, b)** digunakan untuk mengatur warna `yglClearColor(r, g, b, a)` mengatur warna latar belakang jendela dengan komponen merah, hijau, biru, dan alpha (ketelitian warna transparansi). `ang` akan digunakan untuk garis dengan posisi di tengah. Kode warna (1, 0, 0.701) akan menghasilkan warna magenta.
- **glVertex2f(x, y)** digunakan untuk mendefinisikan titik awal dan akhir dari garis kedua. Dalam kasus ini, didefinisikan dua titik untuk membuat garis horizontal dengan satu warna. Titik awal adalah (100,50) dan titik akhir garis adalah (-100,50), ini menandakan garis akan datarik secara horizontal sepanjang 200 titik, dari kanan (x bernilai positif) ke kiri (x bernilai negatif), dengan posisi 50 titik lebih rendah dari garis sebelumnya, bisa dilihat dengan nilai y yang lebih kecil, dari garis sebelumnya y = 100, menjadi y = 50.

```
glColor3f(0.0, 0.51, 0.85);  
glVertex2f(100,0);  
glVertex2f(-100,0);
```

- **glColor3f(r, g, b)** digunakan untuk mengatur warna yang akan digunakan untuk garis dengan posisi di paling bawah. Kode warna (0.0, 0.51, 0.85) akan menghasilkan warna cyan.
- **glVertex2f(x, y)** digunakan untuk mendefinisikan titik awal dan akhir dari garis ketiga. Dalam kasus ini, didefinisikan dua titik untuk membuat garis horizontal dengan satu warna. Titik awal adalah (100,0) dan titik akhir garis adalah (-100,0), ini menandakan garis akan datarik secara horizontal sepanjang 200 titik, dari kanan (x bernilai positif) ke kiri (x bernilai negatif), dengan posisi 50 titik lebih rendah dari garis kedua, bisa dilihat dengan nilai y yang lebih kecil, dari garis sebelumnya y = 50, menjadi y = 0.

```
glEnd ();
```

glEnd() menandakan akhir dari definisi garis-garis. Semua garis yang didefinisikan dalam blok **glBegin()** dan **glEnd()** akan digambar pada tampilan grafis.

```
glFlush();  
glFinish ();
```

glFlush() digunakan untuk memastikan bahwa semua perintah gambar yang tertunda dieksekusi, dan **glFinish()** menunggu hingga semua perintah selesai dieksekusi sebelum program selesai.

```
int main(int argc, char**argv) {  
    glutInit (&argc, argv);  
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
```

Ini adalah fungsi **main()** dari program Anda. Ini adalah titik awal dari program. Fungsi ini melakukan inisialisasi GLUT, mengatur mode tampilan menjadi "single buffer" dan "RGB color mode".

```
    glutInitWindowSize(400, 400);  
    glutInitWindowPosition(100,100);  
    glutCreateWindow ("2206050 - Yoga Agustiansyah");
```

Membuat dan mengatur ukuran menjadi 400px X 400px dan posisi sesuai yang diinginkan, window tersebut diberi judul "2206050 - Yoga Agustiansyah".

```
    glClearColor(0.2, 0.2, 0.2, 1.0);
```

glClearColor(r, g, b, a) untuk mengatur warna latar belakang jendela dengan komponen merah, hijau, biru, dan alpha (ketelitian warna transparansi). Pada program ini, digunakan kode (0.2, 0.2, 0.2, 1.0) untuk mendapatkan warna abu-abu gelap.

```
    glMatrixMode(GL_PROJECTION);  
    gluOrtho2D(-200, 200, -200, 200);
```

Ini mengatur mode matriks menjadi GL_PROJECTION dan mengatur sistem koordinat menggunakan **gluOrtho2D()**. Dengan **gluOrtho2D()** kita bisa membatasi koordinat maksimal yang ditampilkan, pada program ini, Xmin = -200, Xmaks = 200, Ymin = -200 dan Ymaks = 200.

```
    glutDisplayFunc (garis);
```

Mengatur fungsi **garis()** sebagai fungsi yang akan dipanggil saat tampilan diperbarui.

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

glutMainLoop() adalah loop utama yang menjalankan program. Ini akan menjaga tampilan grafis tetap aktif hingga program ditutup.

return 0; adalah bagian dari fungsi main() dalam program. Ini digunakan untuk mengakhiri program dengan mengembalikan nilai 0 kepada sistem operasi. Nilai 0 ini adalah nilai yang umumnya digunakan sebagai penanda bahwa program telah berakhir dengan sukses tanpa adanya masalah.

III. Kesimpulan

Dari hasil praktikum ini, kita bisa mengetahui bagaimana proses pembuatan objek grafis dengan memanfaatkan konsep Primitive Drawing pada lingkungan OpenGL. Dengan bantuan beberapa function kita bisa membuat berbagai objek dengan berbagai variasi, mulai dari warna, bentuk, dan ukuran. Kemampuan untuk memanipulasi warna, bentuk, dan ukuran dengan baik merupakan aspek penting dalam menghasilkan visualisasi yang menarik.

Keberhasilan dalam praktikum dengan membuat tampilan yang menampilkan tiga garis dengan warna yang berbeda membuktikan potensi Primitive Drawing dalam menciptakan visualisasi yang menarik. Program yang telah dikembangkan juga dapat dijadikan dasar untuk pengembangan lebih lanjut dalam menggambar objek-objek grafis yang lebih kompleks.