

Modul Praktikum 2

Primitive Drawing

A. Tujuan

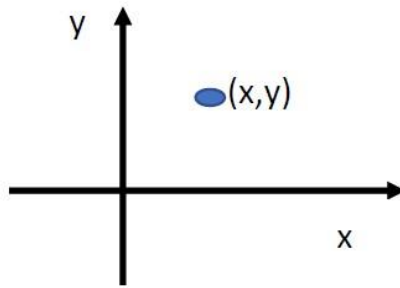
1. Mahasiswa mampu menjelaskan konsep *primitive drawing*.
2. Mahasiswa mampu membuat dan memanfaatkan output primitif.

B. Dasar Teori

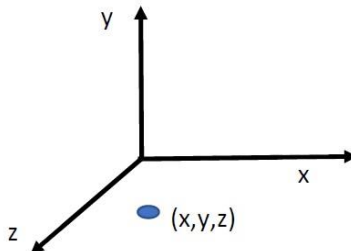
Primitive Drawing merupakan suatu metode menggambar objek pada suatu layar / monitor dengan menggunakan langkah-langkah geometri yang bersifat sederhana. Geometri merupakan suatu cabang ilmu matematika yang mengandung tentang konsep skema bangun ruang yang berfokus pada pengukuran, bentuk posisi dan beberapa parameter bangun ruang lainnya. Primitive drawing membangun sebuah bangun datar atau bangun ruang menggunakan bagian-bagian objek pada objek gambarnya, misalnya titik, garis, garis dari suatu objek. Sehingga pada tahap tertentu, seluruh bagian tersebut akan digabungkan menjadi sebuah deskripsi bentuk objek bangun datar (dua dimensi) atau bangun ruang (tiga dimensi).

Konsep Dasar Titik Koordinat

Opengl dapat berperan sebagai wadah atau tempat untuk menampilkan hasil komputasi dari instruksi atau perintah yang dituliskan dalam software Bahasa pemrograman C++. Misalnya membuat suatu bangun datar, maka secara konsep bangun datar yang telah direncanakan pasti memiliki beberapa bagian yang harus dipersiapkan, contohnya bangun datar memiliki titik-titik koordinat, fungsi dari titik koordinat tersebut sebagai acuan letak bangun datar yang akan dibangun, kemudian ada juga garis yang menghubungkan antar titik koordinat agar bentuk dari bangun datar tersebut dapat secara tegas dapat dibentuk. Hal ini juga didukung beberapa fitur tambahan lain, untuk mendeskripsikan bangun datar yang dibangun seperti, warna garis, warna bidang bangun datar dan lain sebagainya. Pada konsep dasar penempatan sumbu, sudah sangat populer dikenal. Terlebih lagi di masa sekolah hingga tingkat Pendidikan tinggi. Sumbu koordinat yang dimaksud adalah sumbu (x dan y). Ilustrasi bentuk titik koordinat dari sumbu (x,y) dapat dilihat pada Gambar berikut.

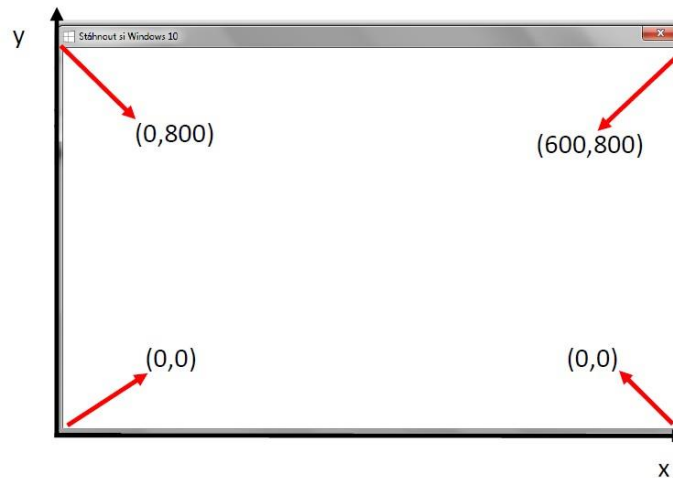


Sumbu x dan y dapat digunakan untuk menggambar objek titik, dan bangun datar yang sering juga dikenal dengan bangun datar atau dua dimensi (2D). Dengan adanya sumbu x dan y, user dapat menentukan tempat menggambar objek pada lokasi yang diinginkannya. Namun dalam visualisasi program executable, hasil dari proses compile and Run sumbu (x dan y) tidak dapat dilihat bentuknya. Namun user dapat menginisialisasi lokasi objek titik yang akan digambar dengan menggunakan instruksi yang ditulis pada text editor. Hal yang sama, juga dapat dilihat dengan menggunakan sumbu (x,y,z) , dimana sumbu ini merepresentasikan tampilan yang memiliki orientasi bidang 3 dimensi. Deskripsi titik koordinat dengan menggunakan sumbu (x,y,z) dapat dilihat pada Gambar berikut.



Dalam lembar kerja, user akan disajikan ruang atau dimensi lokasi menggambar yang biasa disebut juga dengan drawing window. Hal ini dapat membantu user untuk mengetahui seberapa besar ruang drawing window serta dapat memposisikan lokasi gambar secara tepat sesuai dengan ukuran drawing window yang telah di-setting sebelumnya. Jika dimensi drawing windows di-setting sebesar 600,800, maka user akan dapat memahami bawa nilai sumbu x maksimal adalah sebesar 600, dan nilai sumbu y maksimal adalah sebesar 800. Dengan besar dimensi tersebut, maka user dapat menggunakan titik koordinat (x,y) untuk menggambar objek titik dimasa saja. Dimana uraian titik koordinat terkecil dari sumbu (x,y) adalah $(0,0)$ hingga yang terbesar adalah $(600,800)$. Dengan titik koordinat sebanyak itu, maka user bebas menggambar titik sesuai

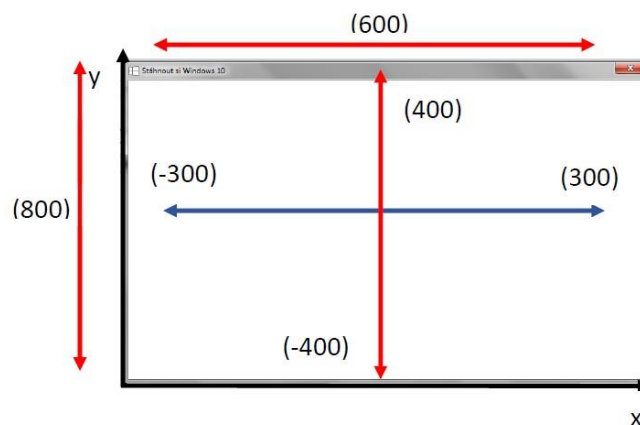
dengan cakupan dimensi drawing windows yang telah diatur. Deskripsi terkait ukuran drawing windows dapat dilihat pada Gambar berikut.



Untuk melakukan setting ukuran drawing windows, user dapat menggunakan perintah atau instruksi sebagai berikut :

1. `glutInitWindowSize(600,800);`
2. `glutInitWindowPosition(100,100);`

Kedua perintah ini cukup sering digunakan untuk mengkonfigurasi serta menentukan besarnya dimensi drawing windows. Selain itu, user juga dapat menggunakan perintah `glOrtho()`, instruksi ini memberikan user opsi untuk meletakkan objek yang akan digambar. Hasil penerapan instruksi `glOrtho()` dapat dilihat pada Gambar berikut.



C. Praktikum

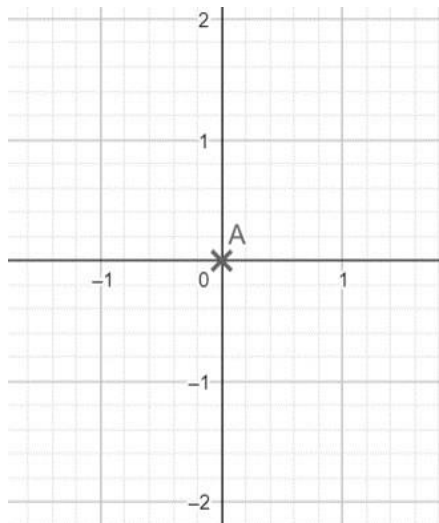
Membuat Sebuah Objek Titik

Seorang user ingin menggambar sebuah objek titik pada sumbu koordinat x dan y, dengan menggunakan Dev C++ dan OpenGL dengan kriteria :

- a. Titik koordinat (0,0)
- b. Warna Merah
- c. Ukuran drawing windows (600, 800)
- d. Daerah matriks (disesuaikan)
- e. Besar objek titik 55,5

Maka penyelesaiannya dapat dilakukan dengan langkah berikut :

1. Menetapkan titik koordinat lokasi penggambaran titik pada sumbu (x,y) yaitu (0,0), seperti yang ditunjukkan pada Gambar berikut.



2. Preproses drawing objek telah diperoleh, kemudian melakukan beberapa konfigurasi yaitu dengan menggunakan instruksi :
 - a. `#include <GL/glut.h>`
Instruksi ini digunakan untuk memanggil library OpenGL, dengan begitu hasil dari kompilasi instruksi yang berupa kode objek dapat ditampilkan oleh OpenGL melalui program yang bersifat executable (.exe)
 - b. `main(int argc, char** argv)`

argc dan argv merupakan instruksi yang akan di-forward ke main() dalam bahasa pemrograman C atau C++. argc merupakan kepanjangan dari argument for count yang didefinisikan sebagai salah satu parameter dalam tipe int (integer) serta memiliki fungsi untuk menunjukkan banyaknya parameter yang digunakan pada proses eksekusi suatu program. Sementara argv merupakan singkatan dari argument for vector yang berfungsi untuk menyimpan seluruh parameter yang digunakan selama proses compile and run berlangsung.

c. void titik (); void merupakan fungsi dengan tipe data kosong, berbeda dengan integer dan float yang memiliki nilai. Oleh karena void tidak memiliki nilai, maka operasi yang dieksekusi tidak akan menyertakan perintah Return. Penulisan parameter pada fungsi void bersifat opsional.

d. glutInit(&argc,argv); memiliki fungsi dalam melakukan proses argument command line yang disertakan. Fungsi ini merupakan satu dari sekian fungsi yang masuk ke dalam kategori, yang harus dipanggil pertama kali, atau masuk dalam fungsi yang harus dibuat diawal program.

e. glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB); merupakan fungsi yang dapat digunakan untuk memberikan efek RGB pada grafik, titik atau pun garis yang ingin dibuat oleh user. Serta memungkinkan user untuk memilih menggunakan buffer windows secara single atau double. Perbedaan utamanya adalah jika buffer windows secara single kode objek yang dihasilkan proses linking akan langsung ditampilkan pada program yang bersifat executable (.exe), sehingga user dapat melihat secara langsung objek dalam bentuk gambar. Sementara double, user dapat membayangkan menggunakan dua buffer windows, dimana hanya salah satunya saja yang dapat terlihat sementara yang lainnya tidak terlihat.

f. glutInitWindowSize(int x, int y); dideskripsikan sebagai perintah untuk melakukan setting ukuran drawing windows, sebesar (x,y).

g. glutInitWindowPosition(); digunakan untuk menentukan posisi sebuah windows

h. glutCreateWindow("....."); digunakan untuk membuat suatu windows

i. glClearColor(); berfungsi untuk membersihkan atau me-reset nilai pixel.

glMatrixMode(); berfungsi untuk menginisialisasi matriks

- k. `glOrtho()`; berfungsi sebagai penentu letak secara spesifik objek akan digambar, misalnya `glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0)`; perintah ini akan membuat objek yang digambar berada pada interval sumbu x dengan titik koordinat $[-1,1]$, sumbu y dengan titik koordinat $[-1,1]$ dan sumbu z dengan titik koordinat $[-1,1]$.
 - l. `glutDisplayFunc()`; merupakan perintah yang digunakan untuk menampilkan hasil komputasi suatu fungsi, namun penulisan nama fungsi harus sesuai dengan fungsi yang ditulis sebelumnya.
 - m. `glutMainLoop()`; merupakan perintah untuk mengulang atau melakukan proses looping
3. Selanjutnya melakukan penggambaran objek titik dengan menggunakan perintah
- a. `glClear(GL_COLOR_BUFFER_BIT)`; digunakan untuk menghapus warna pada layar
 - b. `glPointSize()`; digunakan untuk menentukan besar objek yang akan digambar, semakin besar nilainya akan semakin besar objeknya.
 - c. `glBegin()`; digunakan untuk menuliskan perintah menggambar objek titik
 - d. `glColor3f()`; digunakan untuk memberikan warna yang baru pada objek, setelah sebelumnya warna yang ada telah dihapus.
 - e. `glVertex2f()`; dideskripsikan sebagai instruksi untuk menuliskan titik koordinat objek akan digambar pada drawing windows.
4. Mengakhiri program dengan statement tambahan,
- a. `glEnd()`; mengakhiri perintah `glBegin()`
 - b. `glFinish()`; adalah sebagai instruksi untuk

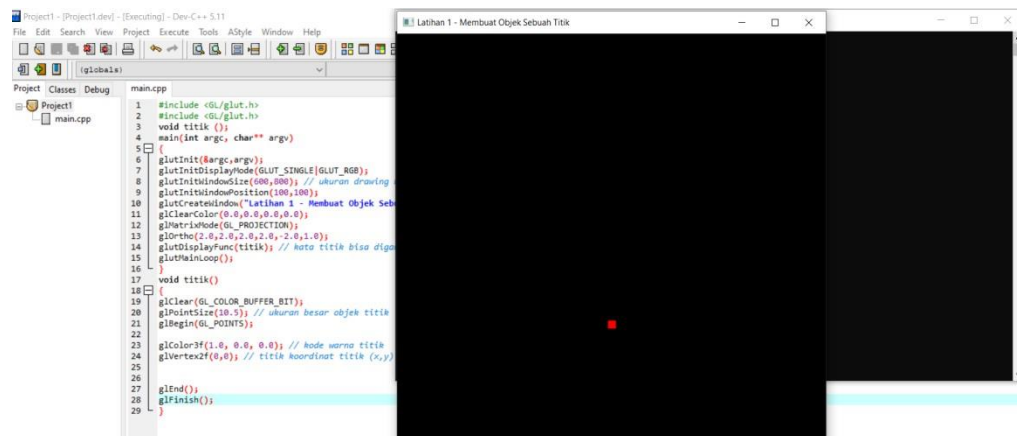
Sehingga dari uraian diatas, maka seluruh list program dapat dituliskan dan digabungkan sebagai berikut :

```

1 //List Program Menggambar Objek 1
2 #include <GL/glut.h>
3 #include <GL/glut.h>
4 void titik ();
5 main(int argc, char** argv)
6 {
7     glutInit(&argc,argv);
8     glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
9     glutInitWindowSize(600,800); // ukuran drawing windows
10    glutInitWindowPosition(100,100);
11    glutCreateWindow("Latihan 1 - Membuat Objek Sebuah Titik"); // judul drawing windows
12    glClearColor(0.0,0.0,0.0,0.0);
13    glMatrixMode(GL_PROJECTION);
14    glOrtho(2.0,2.0,2.0,2.0,-2.0,1.0);
15    glutDisplayFunc(titik); // kata titik bisa diganti, namun harus sesuai dengan kata fungsi yang dibawahnya
16    glutMainLoop();
17 }
18 void titik()
19 {
20     glClear(GL_COLOR_BUFFER_BIT);
21     glPointSize(10.5); // ukuran besar objek titik
22     glBegin(GL_POINTS);
23     glColor3f(1.0, 0.0, 0.0); // kode warna titik
24     glVertex2f(0,0); // titik koordinat titik (x,y)
25     glEnd();
26     glFinish();
27 }

```

Tampilan dan hasil proses *compile and run* dengan program diatas dapat dilihat Gambar. dan Gambar berikut.



Membuat Beberapa Objek Titik

Seorang user ingin menggambar beberapa objek titik pada sumbu koordinat x dan y, dengan menggunakan Dev C++ dan OpenGL dengan kriteria :

- Titik koordinat titik pertama (0,0) ; titik kedua (0,0.15) ; titik ketiga (0,0.3) ; titik keempat (0,0.45) ; titik kelima (0,0.6)
- Warna titik pertama merah, titik kedua kuning, titik ketiga hijau, titik keempat biru, titik kelima ungu
- Ukuran drawing windows (600, 800)

d. Daerah matriks (disesuaikan)

e. Besar objek titik 10,5

f. Background hitam diganti dengan warna abu-abu

Penyelesaiannya dapat dengan membuat duplikasi pada program “Latihan-I” dengan menambahkan instruksi sesuai dengan keinginan user, yaitu dengan menambahkan empat titik lainnya. Modifikasi list program pertama dapat dilakukan dengan cara menduplikasi perintah pembentukan titik pertama untuk empat titik lainnya, serta melakukan penambahan instruksi untuk mengubah warna, dan titik koordinat masing-masing objek titik. Untung mengubah warna background drawing windows dapat dilakukan dengan mengubah kode warna pada instruksi “glClearColor()”. List program secara lengkap dapat dilihat dibawah ini.

```
main.cpp
1  #include <GL/glut.h>
2  #include <GL/glut.h>
3  void titik ();
4  main(int argc, char** argv)
5  {
6      glutInit(&argc,argv);
7      glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
8      glutInitWindowSize(600,800); // ukuran drawing windows
9      glutInitWindowPosition(100,100);
10     glutCreateWindow("Latihan 2 - Membuat Objek Banyak Titik, dengan Variasi Warna"); // judul drawing windows
11     glClearColor(0.6f, 0.6f, 0.6f, 1.0f); //Ganti warna drawing windows dari hitam menjadi abu-abu
12     glMatrixMode(GL_PROJECTION);
13     glOrtho(2.0,2.0,2.0,2.0,-2.0,1.0);
14     glutDisplayFunc(titik); // kata titik bisa diganti, namun harus sesuai dengan kata fungsi yang dibawahnya
15     glutMainLoop();
16 }
17 void titik()
18 {
19     glClear(GL_COLOR_BUFFER_BIT);
20     glPointSize(10.0); // ukuran besar objek titik
21     glBegin(GL_POINTS);
22     glColor3f(1.0, 0.0, 0.0); // kode warna titik pertama
23     glVertex2f(0,0); // titik koordinat titik (x,y)
24     glColor3f(1.0, 1.0, 0.0); // kode warna titik kedua
25     glVertex2f(0,0.15); // titik koordinat titik (x,y)
26     glColor3f(0.0, 1.0, 0.0); // kode warna titik ketiga
27     glVertex2f(0,0.30); // titik koordinat titik ketiga (x,y)
28     glColor3f(0.0, 0.0, 1.0); // kode warna titik keempat
29     glVertex2f(0,0.45); // titik koordinat titik keempat (x,y)
30     glColor3f(1.0, 0.0, 1.0); // kode warna titik kelima
31     glVertex2f(0,0.60); // titik koordinat titik kelima (x,y)
32     glEnd();
33     glFlush();
34 }
```

Hasil *compile and run* dari program diatas dapat dilihat pada Gambar berikut.



Menggambar Garis Antar Titik Koordinat

Seorang user ingin menggambar dua buah objek titik yang terhubung satu dengan yang lainnya melalui sebuah garis pada sumbu koordinat x dan y, dengan menggunakan Dev C++ dan OpenGL dengan kriteria :

- Titik koordinat titik pertama (0,0) ; titik kedua (300,400) dengan warna garis hitam
- Warna background putih

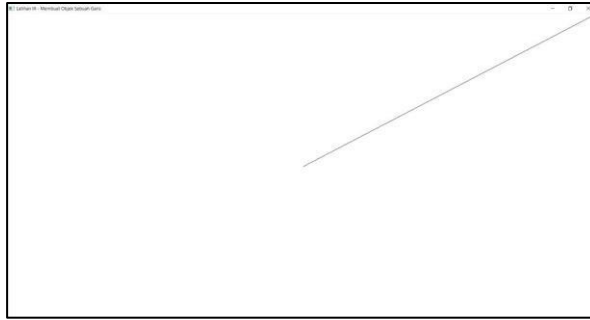
Penyelesaian dengan list program dibawah ini :

```

1 //List program membuat garis yang menghubungkan dua titik
2 #include <GL/glut.h>
3 void Garis () //kata garis dapat diganti sesuai keinginan
4 main(int argc, char** argv)
5 {
6     glutInit(&argc,argv);
7     glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
8     glutInitWindowSize(600,800); // ukuran drawing windows
9     glutInitWindowPosition(100,100);
10    glutCreateWindow("Latihan III - Membuat Objek Sebuah Garis"); // judul drawing windows
11    glClearColor(1.0,1.0,1.0,0.0); //kode warna background
12    glMatrixMode(GL_PROJECTION);
13    gluOrtho2D(-300.0,300.0,-400.0,400.0); //nilainya harus masuk dalam rentang nilai drawing windows
14    glutIdleFunc(Garis); // kata garis mengikuti perintah void yang paling atas
15    glutDisplayFunc(Garis); // kata garis mengikuti perintah void yang paling atas
16    glutMainLoop();
17 }
18 void Garis() // kata garis mengikuti perintah void yang paling atas
19 {
20     glClear(GL_COLOR_BUFFER_BIT);
21     glBegin(GL_LINES); //fungsi membuat garis
22     glColor3f(0.0, 0.0, 0.0); // kode warna titik pertama
23     glVertex2d(0,0); // titik koordinat titik (x,y) pertama
24     glColor3f(0.0, 0.0, 0.0); // kode warna titik kedua
25     glVertex2d(300,400); // titik koordinat titik (x,y) kedua
26     glEnd();
27     glFinish();
28 }
29

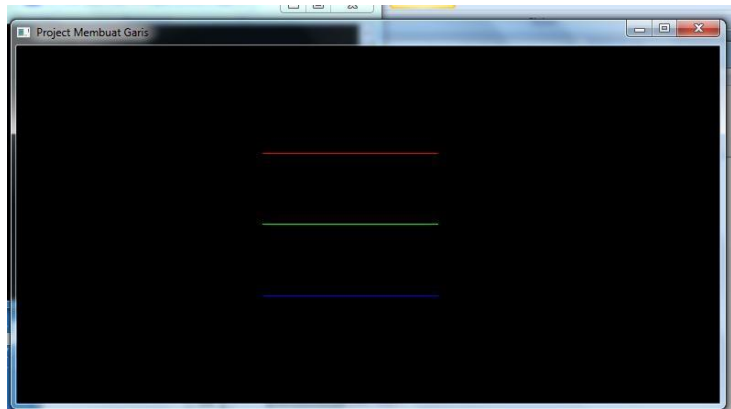
```

Hasil proses *compile and run* pada list program diatas, ditunjukkan pada Gambar berikut.



TUGAS MINGGUAN

1. Buatlah 3 garis sebagaimana gambar berikut dan rubah warnanya.



2. Rubahlah nama windownya dengan (NPM & Nama Masing-masing)
3. Buat laporannya dalam format pdf, lengkap dengan identitas dan cover.
4. Waktu pengerjaan 1 Minggu.