**LAPORAN AKHIR PRAKTIKUM**

Mata Praktikum : Grafik Komputer

Kelas : 3IA24

Praktikum ke- : 3

Tanggal : 25 Oktober 2023

Materi : Transformasi pada OpenGL

NPM : 50421704

Nama : Juan Samuel Christopher

Ketua Asisten : Ayu Fara Paramitha

Nama Asisten :

Paraf Asisten :

Jumlah Lembar : 5

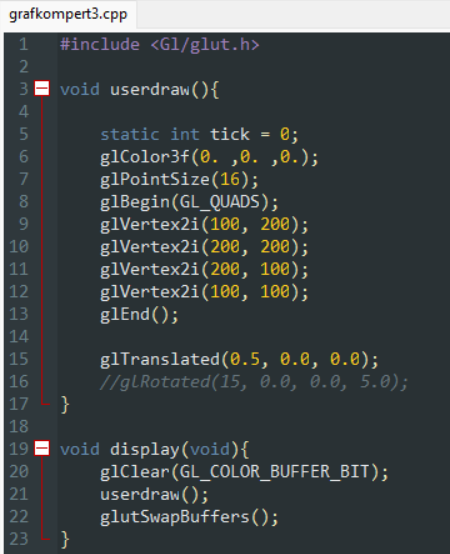


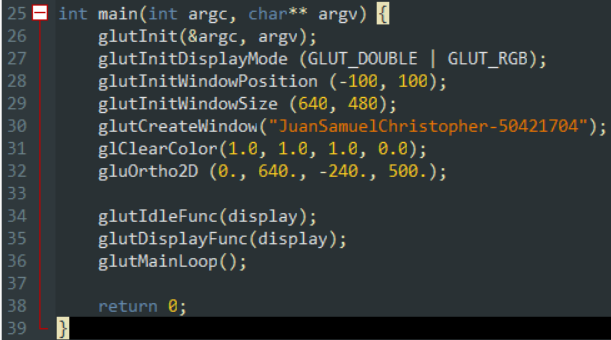
**LABORATORIUM TEKNIK INFORMATIKA**

**UNIVERSITAS GUNADARMA**

**2023**

**LISTING**

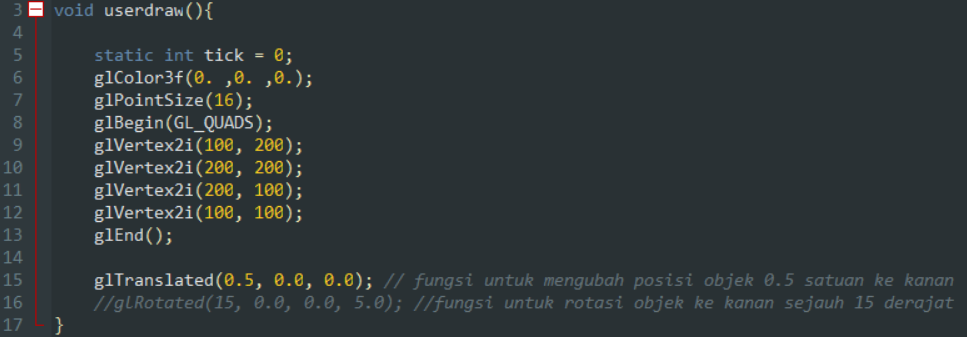
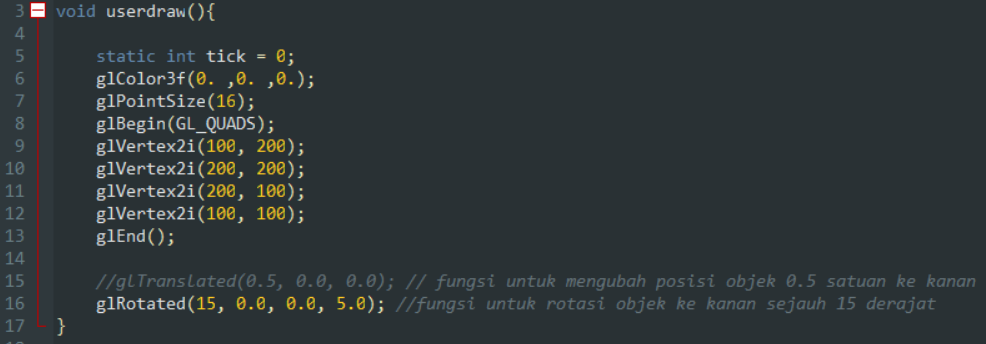




**LOGIKA**

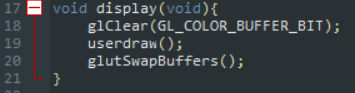


Pada pertemua ke 2 ini kita akan belajar cara menggambar Gambar Primitice pada OpenGL. Kode program ini diawali dengan memasukkan library GLUT (OpenGL Utility Toolkit ) dengan #include <Gl/glut.h> yang akan digunakan untuk membuat console application menggunakan OpenGL

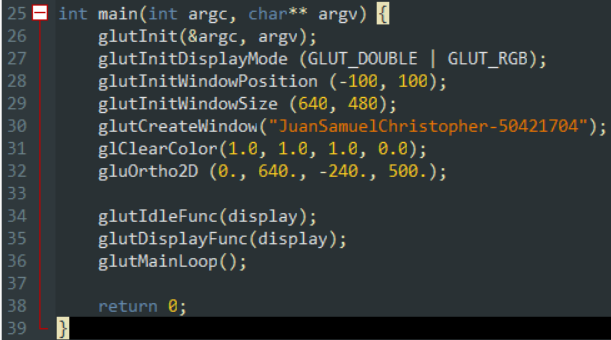


Pada blok program diatas terdapat fungsi **void** yang bernama **Userdraw** yang akan menggambar objek persegi. Lalu ada **static int tick = 0**, adalah deklarasi variable bernama **tick** yang digunakan untuk menghitung waktu, lalu **glColor3f** adalah fungsi yang mengatur warna gambar dalam mode RGB dimana disini R=0, G=0,B=0, maka warna diatur ke warna hitam. Kemudian **glPointSize(16);** adalah fungsi yang mengatur ukuran titik yang akan digunakan pada gambar disini diset ukuran 16.

Lanjut lagi ke **glBegin(GL\_QUADS);** adalah fungsi yang menandakan permulaan gambar yang berarti saya akan menggambar objek bersegi empat yang ditandai oleh adanya **GL\_QUADS.** Kemudian pada line 9 – 12 ada fungsi **glVertex2f** adalah fungsi yang menggambar titik , yang berarti ada 4 titik dimana masing masing titik mempunyai koordinat x dan y yang diatur sedemikian rupa seperti blok program diatas yang akan menggambar persegi yang diinginkan. Dan pada line 15 & 16 ada 2 fungsi yang masing masing harus di running satu per satu , Digambar kiri ada **glRotated** yang berguna untuk merotasi gambar kea rah kanan 15 derajat, dan di gambar kanan ada **glTranslated** yang akan menggeser objek 0.5 satuan ke kanan. Lalu terakhir ada **glEnd();** yang menandakan akhir dari gambar yang berarti seluruh strip bersegi empat telah Digambar.



Selanjutnya ada **void display (void); {** adalah deklarasi fungsi bernama **display**  yang akan dipanggil untuk menampilkan objek/gambar pada layar. Lalu **glClear(GL\_COLOR\_BUFFER\_BIT)** adalah fungsi yang akan membersihkan layar dan menghapus gambar sebelumnnya. Dan **userdraw()** yang berfungsi untuk memanggil **void userdraw()** untuk mengammbar objek persegi yang kita atur tadi. Kemudian **glutSwapBuffers();** adalah fungsi yang bertanggungjawab untuk menukar dua buffer gambar, ini diperlukan karena program ini menggunakan mode double buffer.



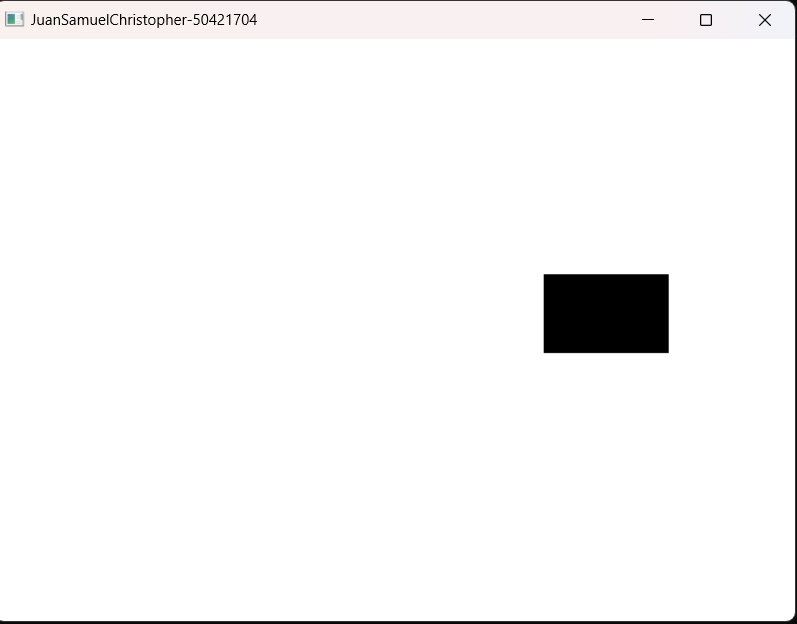
Pada blok program terakhir **int main(int argc, char\*\* argv) {** Ini adalah fungsi main(), yang merupakan titik masuk utama program. **glutInit(&argc, argv):** Fungsi ini menginisialisasi GLUT dengan argumen baris perintah yang diteruskan dari command line. **glutInitDisplayMode (GLUT\_DOUBLE | GLUT\_RGB);** Fungsi ini mengatur mode tampilan GLUT dengan double buffer dan mode warna RGB.

Lanjut ada **glutInitWindowPosition (100, 100);** Fungsi ini mengatur posisi jendela aplikasi. glutInitWindowSize (640, 480); Fungsi ini mengatur ukuran jendela aplikasi. **glutCreateWindow("JuanSamuelChristopher - 50421704");** Fungsi ini membuat jendela dengan judul yang diberikan. **glClearColor** adalah fungsi untuk warna background yang berarti disini (1.0, 1.0, 1.0, 0.0) adalah warna putih, dan **gluOrtho2D i**ni mengatur parameter proyeksi ortografik untuk jendela.

Terakhir ada **glutIdleFunc(display);** Fungsi ini menentukan fungsi display() sebagai fungsi yang akan dipanggil ketika aplikasi berada dalam mode idle. **glutDisplayFunc(display);** Fungsi ini menentukan fungsi display() sebagai fungsi yang akan digunakan untuk menampilkan objek pada layar. **glutMainLoop();** Fungsi ini memulai loop utama GLUT, yang akan menangani interaksi pengguna dan menampilkan objek pada layar. **Return 0;** Ini adalah nilai kembali dari fungsi main() yang menunjukkan bahwa program selesai dengan sukses.

**OUTPUT**

* **Translated**

****

* **Rotated**

****