

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : GK1
Kelas : 3IA19
Praktikum : 2
Tanggal : 10-27-2023
Materi : Membuat Objek 2D
NPM : 50421859
Nama : Muhamad Ariel D.P
Ketua Asisten :
Nama Asisten : Tommy Wijaya Rachman
Paraf Asisten :
Jumlah Lembar : 4



**LABORATORIUM INFORMATIKA
UNIVERSITAS GUNADARMA
2023**

LISTING

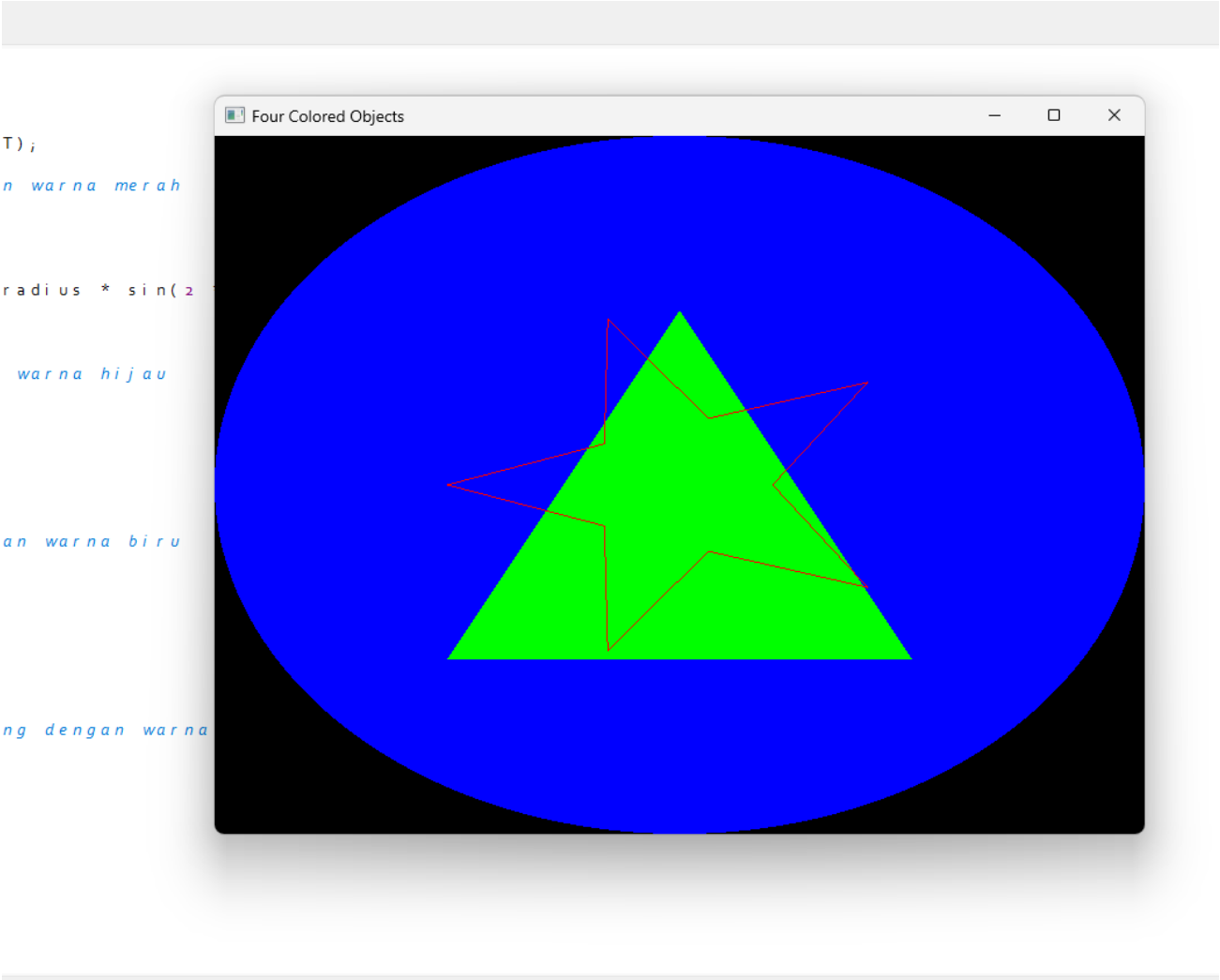
1. Buat Objek Baru Pada Program?

Jawab:

LISTING

```
Project3 - [Project3.dev] - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
[Icons] (globals)
Project Classes
Project3
  La2.cpp
La2.cpp
1 #include <GL/glut.h>
2 #include <cmath>
3
4 void display() {
5     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
6
7     // Membuat objek pertama berbentuk bintang dengan warna merah
8     glColor3f(1.0, 0.0, 0.0);
9     glBegin(GL_LINE_LOOP);
10    for(int i = 0; i < 10; ++i) {
11        float radius = (i % 2 == 0) ? 0.2 : 0.5;
12        glVertex2f(radius * cos(2 * M_PI * i / 10), radius * sin(2 * M_PI * i / 10));
13    }
14    glEnd();
15
16    // Membuat objek kedua berbentuk segitiga dengan warna hijau
17    glColor3f(0.0, 1.0, 0.0);
18    glBegin(GL_TRIANGLES);
19    glVertex2f(-0.5, -0.5);
20    glVertex2f(0.5, -0.5);
21    glVertex2f(0.0, 0.5);
22    glEnd();
23
24    // Membuat objek ketiga berbentuk lingkaran dengan warna biru
25    glColor3f(0.0, 0.0, 1.0);
26    glBegin(GL_POLYGON);
27    for(int i = 0; i < 360; i++) {
28        float rad = i * M_PI / 180;
29        glVertex2f(cos(rad), sin(rad));
30    }
31    glEnd();
32
33    // Membuat objek keempat berbentuk persegi panjang dengan warna kuning
34    glColor3f(1.0, 1.0, 0.0);
35    glBegin(GL_QUADS);
36    glVertex2f(-0.5, -0.25);
37    glVertex2f(0.5, -0.25);
38    glVertex2f(0.5, 0.25);
39    glVertex2f(-0.5, 0.25);
40    glEnd();
41
42    glutSwapBuffers();
43 }
44
45 int main(int argc, char** argv) {
46     glutInit(&argc, argv);
47     glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH);
48     glutInitWindowSize(800, 600);
49     glutCreateWindow("Four Colored Objects");
50
51     glEnable(GL_DEPTH_TEST);
52
53     glutDisplayFunc(display);
54
55     glutMainLoop();
56
57     return 0;
58 }
59
```

OUTPUT



2. Berikan Warna Yang Berbeda Pada Setiap Program?

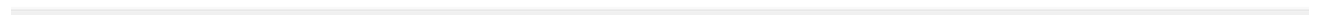
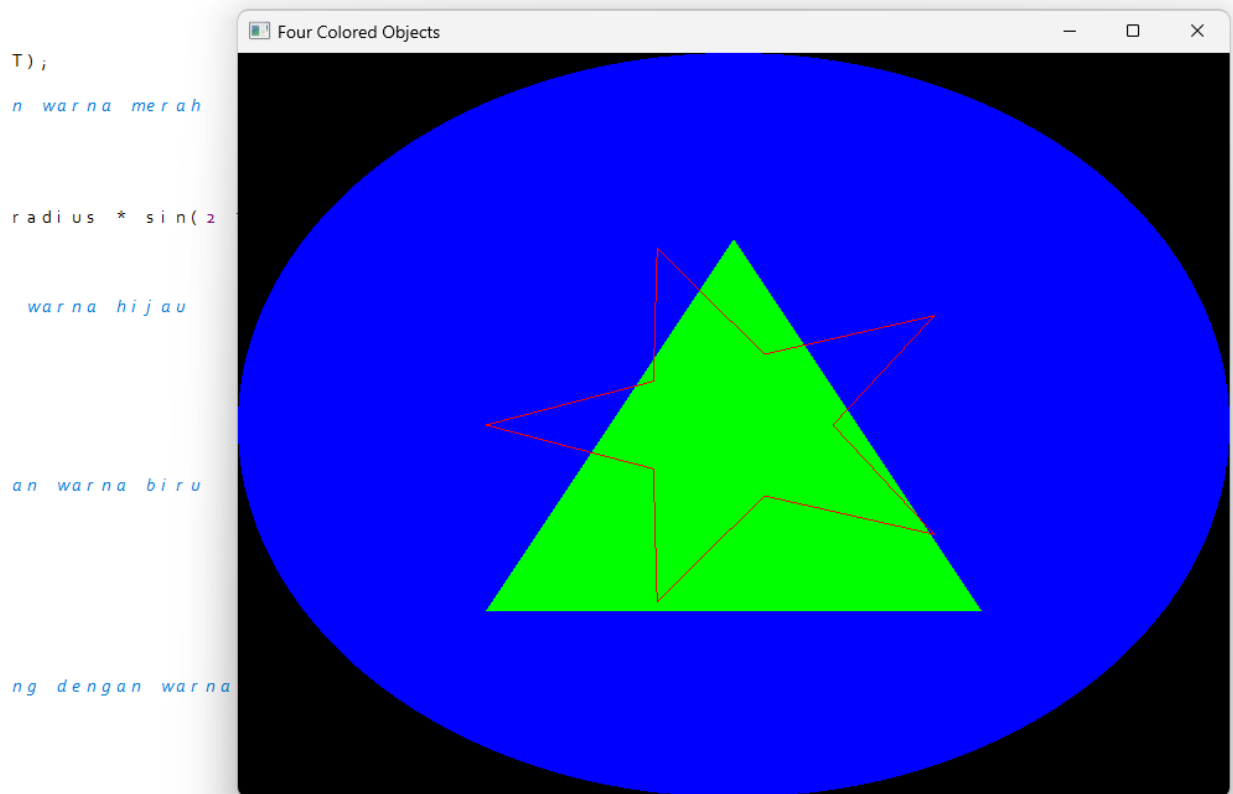
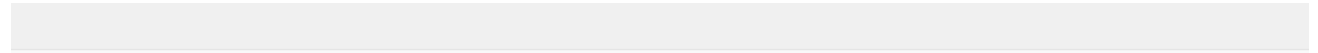
Jawab:

Warna Hitam Persegi Panjang

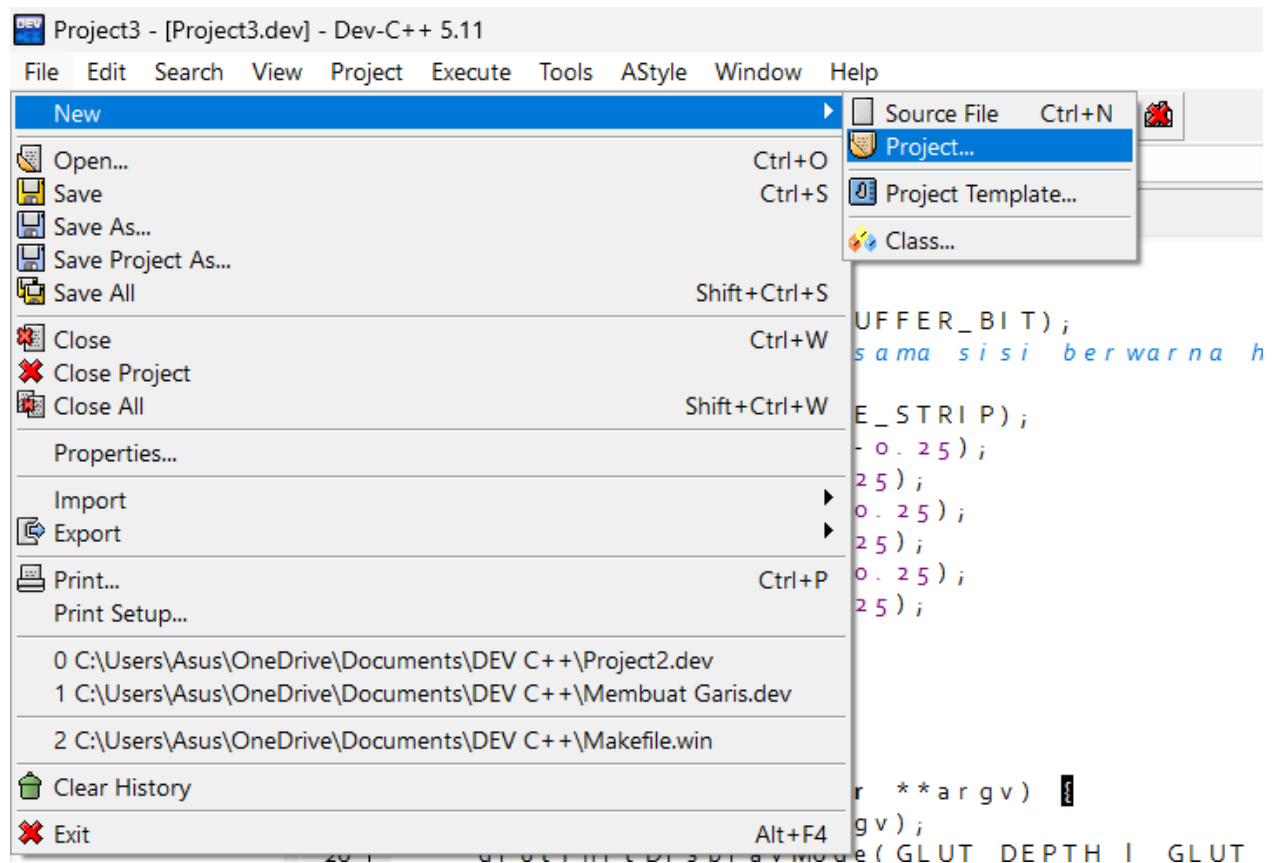
Warna Biru Oval/Lingkaran

Warna Hijau Segitiga

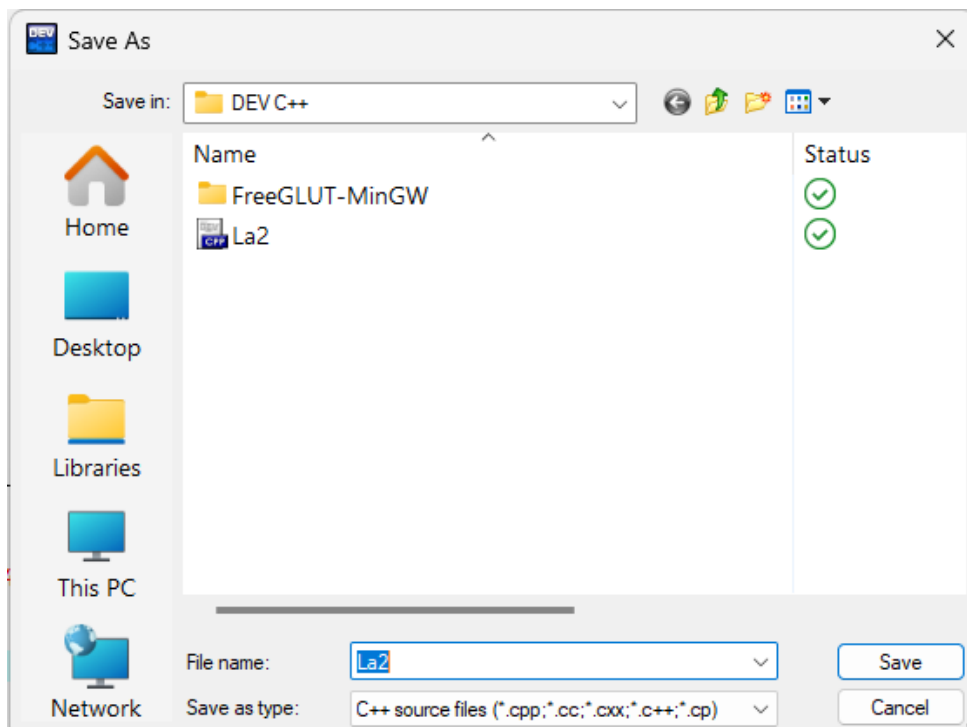
Warna Garis merah Bintang



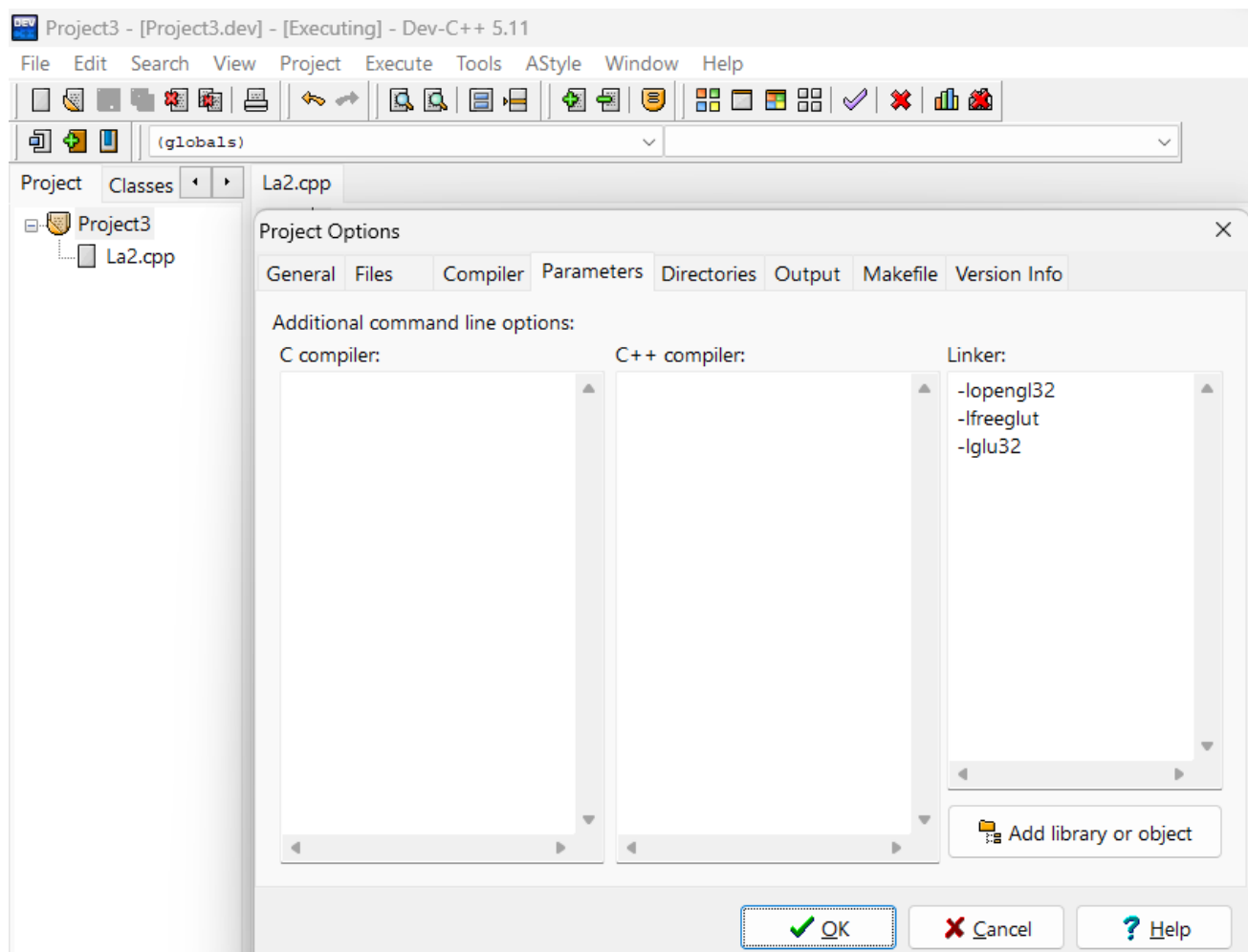
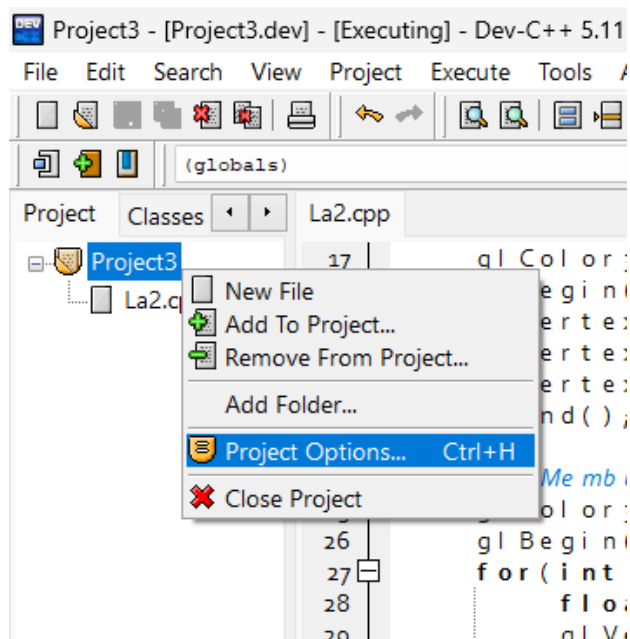
3. Jelaskan Langkah2 nya Dengan Singkat?



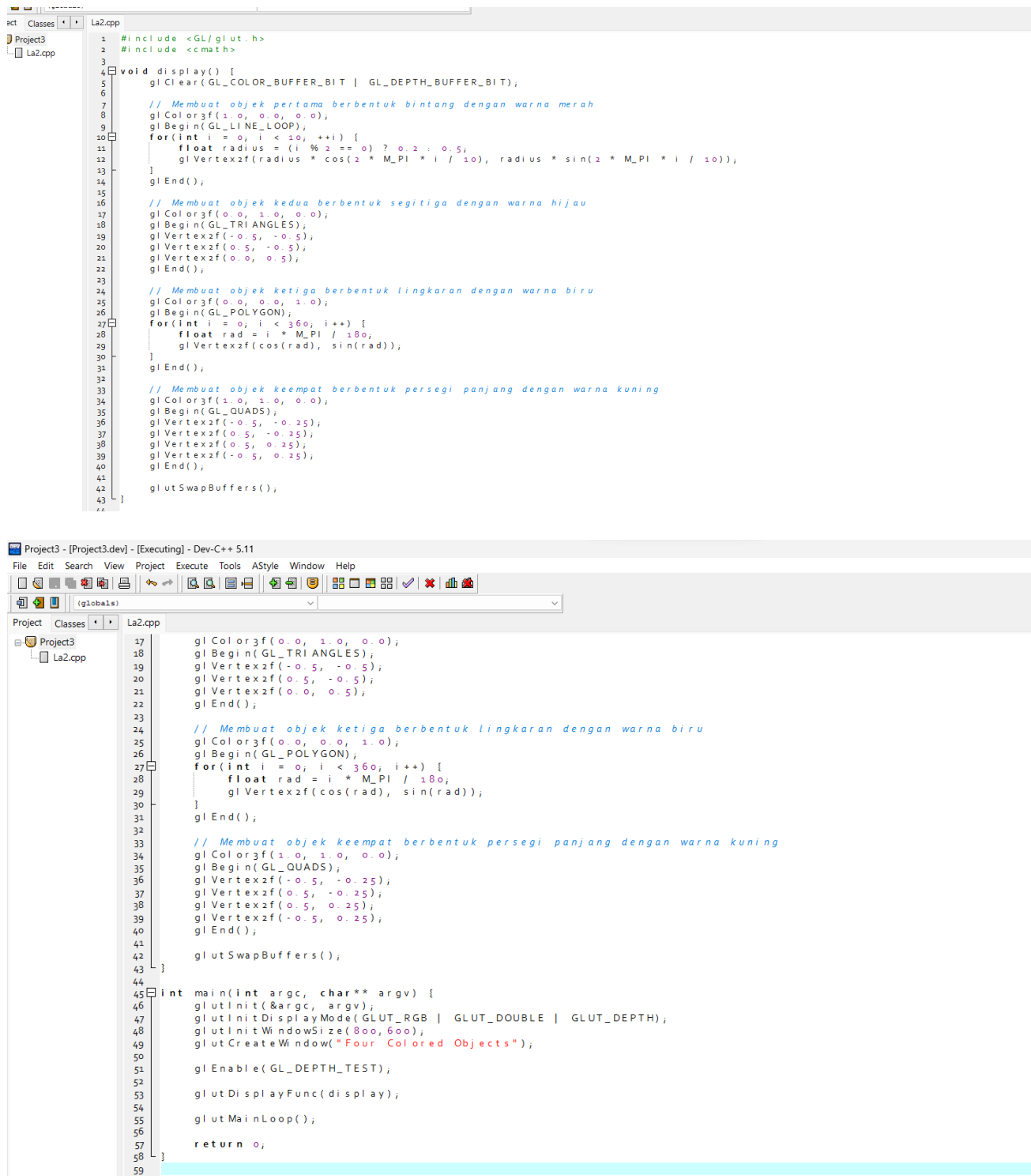
Pertama-tama kita buat baru project dengan pilih menu new selanjutnya pilih menu project



Selanjutny akita save kodingannya supaya bisa di running nantinya



Selanjutnya sebelum menuliskan program masukan Linker terlebih dahulu yang berfungsi untuk menjalankan program Open GL



Program ini adalah program OpenGL yang digunakan untuk menggambar tiga objek dengan bentuk dan warna yang berbeda:

1. ****Bintang berwarna merah****: Objek pertama adalah bintang yang digambar dengan menggunakan `'GL_LINE_LOOP'`. Titik-titik pada bintang dihasilkan dengan mengubah radius antara 0.2 dan 0.5, yang menciptakan efek bintang.

2. ****Segitiga berwarna hijau****: Objek kedua adalah segitiga yang digambar dengan menggunakan `'GL_TRIANGLES'`. Titik-titik pada segitiga didefinisikan secara eksplisit.

Fungsi `'glColor3f'` digunakan untuk mengatur warna objek, dan `'glVertex2f'` digunakan untuk mendefinisikan titik-titik pada objek. Fungsi `'glBegin'` dan `'glEnd'` digunakan untuk menandai awal dan akhir dari setiap objek yang digambar.

Selanjutnya Program ini adalah program OpenGL yang digunakan untuk menggambar empat objek dengan bentuk dan warna yang berbeda:

1. ****Bintang berwarna merah****: Objek pertama adalah bintang yang digambar dengan menggunakan `'GL_LINE_LOOP'`. Titik-titik pada bintang dihasilkan dengan mengubah radius antara 0.2 dan 0.5, yang menciptakan efek bintang.

2. ****Segitiga berwarna hijau****: Objek kedua adalah segitiga yang digambar dengan menggunakan `'GL_TRIANGLES'`. Titik-titik pada segitiga didefinisikan secara eksplisit.

3. ****Lingkaran berwarna biru****: Objek ketiga adalah lingkaran yang digambar dengan menggunakan `'GL_POLYGON'`. Titik-titik pada lingkaran dihasilkan dengan mengubah sudut dari 0 hingga 360 derajat, yang menciptakan efek lingkaran.

4. ****Persegi panjang berwarna kuning****: Objek keempat adalah persegi panjang yang digambar dengan menggunakan `'GL_QUADS'`. Titik-titik pada persegi panjang didefinisikan secara eksplisit.

Fungsi `'glColor3f'` digunakan untuk mengatur warna objek, dan `'glVertex2f'` digunakan untuk mendefinisikan titik-titik pada objek. Fungsi `'glBegin'` dan `'glEnd'` digunakan untuk menandai awal dan akhir dari setiap objek yang digambar.

Selanjutnya Kode ini adalah bagian dari program OpenGL yang digunakan untuk menginisialisasi dan menjalankan window aplikasi OpenGL. Berikut adalah penjelasan detailnya:

- ``glutInit(&argc, argv);``: Fungsi ini digunakan untuk menginisialisasi GLUT (OpenGL Utility Toolkit). Parameter yang diterima adalah argumen baris perintah yang biasanya diteruskan dari fungsi main.
- ``glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH);``: Fungsi ini digunakan untuk mengatur mode tampilan awal. Dalam hal ini, mode tampilan diatur ke RGB (warna merah hijau biru), double buffering (untuk mencegah flickering), dan depth buffer (untuk penanganan kedalaman objek 3D).
- ``glutInitWindowSize(800,600);``: Fungsi ini digunakan untuk mengatur ukuran awal window aplikasi. Dalam hal ini, ukuran window diatur ke 800x600 piksel.
- ``glutCreateWindow("Four Colored Objects");``: Fungsi ini digunakan untuk membuat window aplikasi dengan judul "Four Colored Objects".
- ``glEnable(GL_DEPTH_TEST);``: Fungsi ini digunakan untuk mengaktifkan depth testing, yang penting untuk penanganan kedalaman objek 3D.
- ``glutDisplayFunc(display);``: Fungsi ini digunakan untuk menentukan fungsi callback yang akan dipanggil setiap kali window aplikasi perlu di-render ulang. Dalam hal ini, fungsi ``display`` akan dipanggil.
- ``glutMainLoop();``: Fungsi ini digunakan untuk memulai loop utama GLUT. Loop ini akan berjalan terus menerus sampai program dihentikan.
- ``return 0;``: Ini adalah kode keluar standar untuk fungsi main yang menandakan bahwa program telah berakhir dengan sukses.

Output

