

Nama : Augustav Fahrul Alzaiya  
NIM : 2200018048  
Prak : Grafkom 12:15

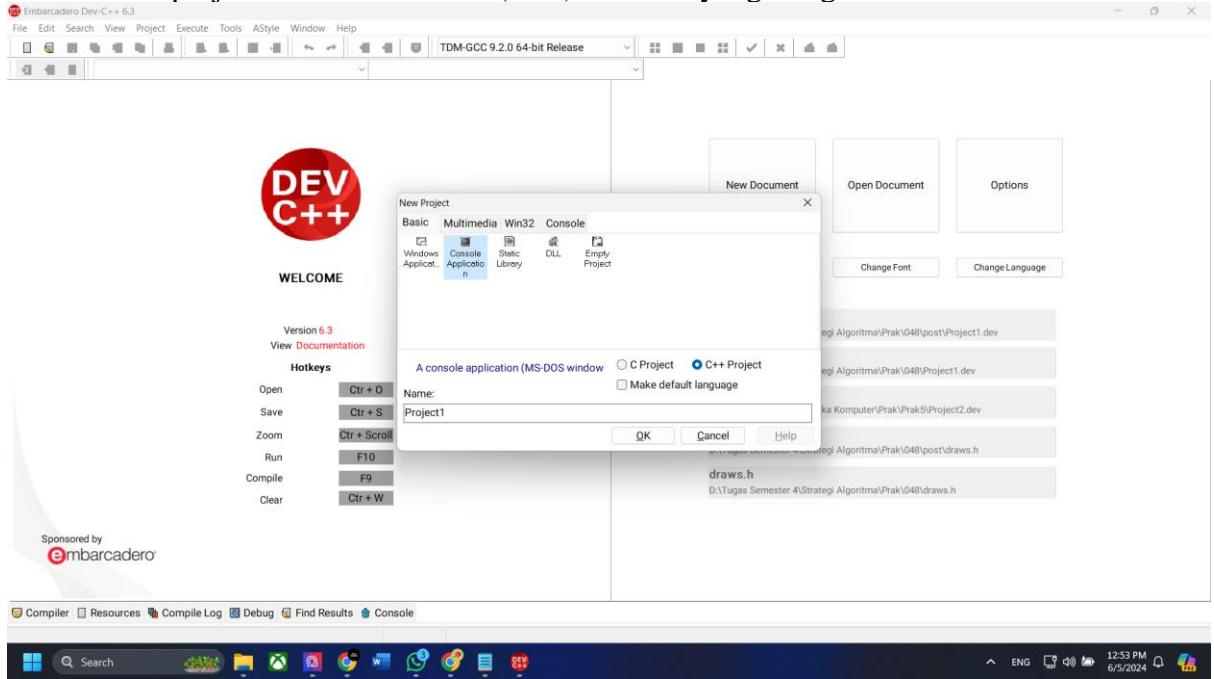
- Pretest 8

Nama: Augustav Fahrul A. NIM : 2200018048	Asisten: Paraf Asisten:	Tanggal: 5-6-2024 Nilai:
--	----------------------------	-----------------------------

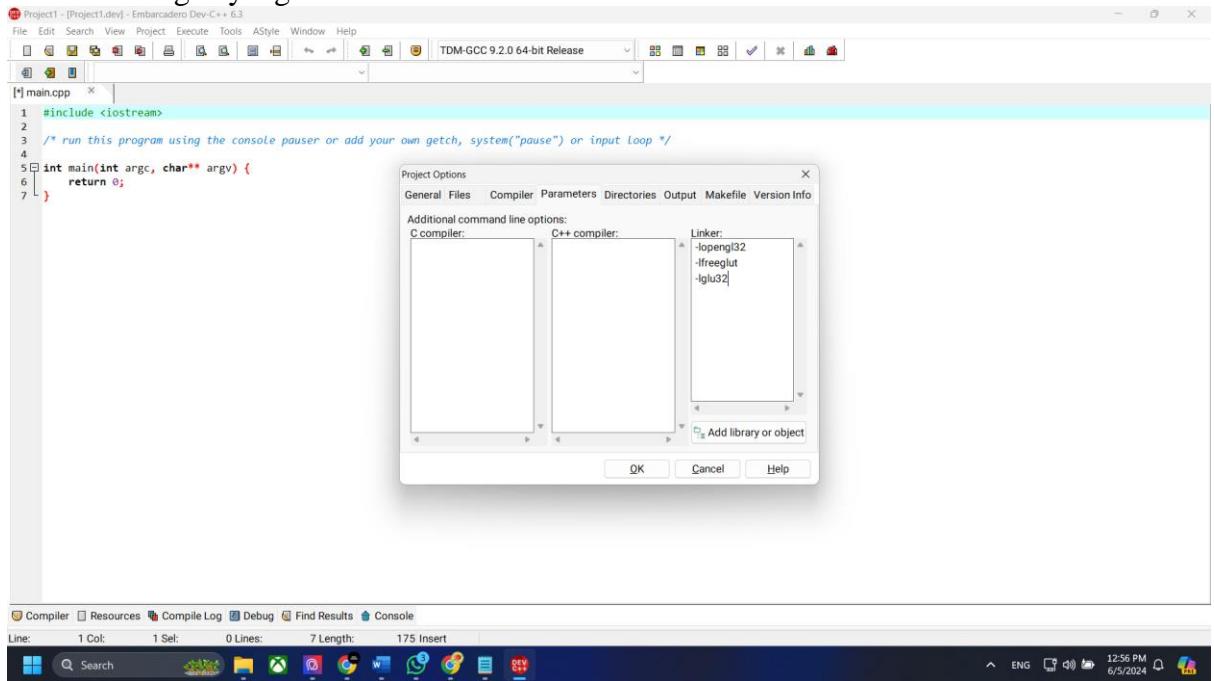
1. Boundary Representation adalah salah satu metode dalam representasi objek 3D yang menggambarkan bentuk dan struktur objek melalui batas-batas atau permukaannya.
2. a. Poligon Mesh → Permukaan objek dibentuk oleh sekumpulan poligon, biasanya segitiga atau segi empat yang saling berhubungan
- b. NURBS (Non-Uniform Rational B-Splines)  
→ menggunakan kurva dan permukaan yang dapat diatur dengan kontrol point, knot Vector, dan weight
- c. Subdivision Surfaces → Teknik yang mulai dengan sebuah mesh kasar dan kemudian secara rekursif membagi setiap poligon untuk menghasilkan permukaan yang lebih halus.
3. Kurva NURBS → generatif hasil dari B-Spline yang menambahkan weight ke setiap kontrol point, memungkinkan kontrol yang lebih besar, dan fleksibilitas dalam menggambarkan bentuk kompleks.
  - NURBS → menggunakan weight, lebih fleksibel, lebih kompleks
  - B-Splines → tidak menggunakan weight, kurang fleksibel, dan lebih sederhana.

- Langkah Praktikum 8

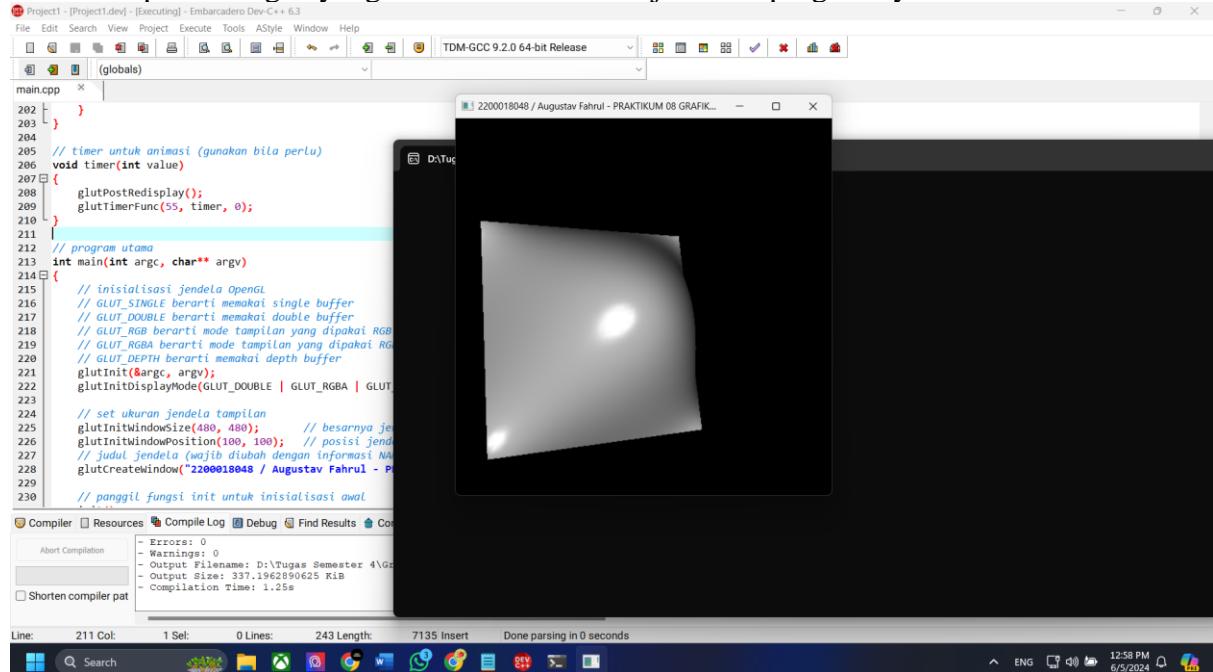
1. Buat projek baru dan letakkan (save) di lokasi yang diinginkan



2. Kemudian masukkan linker. Linker ini berguna untuk membaca gambar gambar dari codingan yang telah dibuat



### 3. Copas codingan yang diberikan kemudian jalankan programnya



The screenshot shows the Embarcadero Dev-C++ IDE interface. The main window displays the code for `main.cpp`. The code initializes OpenGL, sets up a window, and includes a timer function for rendering. A preview window shows a dark, slightly reflective surface, likely a rendered sphere or similar object. The status bar at the bottom indicates the code has been parsed successfully.

```
202     }
203 }
204
205 // timer untuk animasi (gunakan bila perlu)
206 void timer(int value)
207 {
208     glutPostRedisplay();
209     glutTimerFunc(55, timer, 0);
210 }
211
212 // program utama
213 int main(int argc, char** argv)
214 {
215     // initialisasi jendela OpenGL
216     // GLUT_SINGLE berarti memakai single buffer
217     // GLUT_DOUBLE berarti memakai double buffer
218     // GLUT_RGB berarti mode tampilan yang dipakai RGB
219     // GLUT_RGBA berarti mode tampilan yang dipakai RGBA
220     // GLUT_DEPTH berarti memakai depth buffer
221     glutInit(&argc, argv);
222     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA | GLUT_DEPTH);
223
224     // set ukuran jendela tampilan
225     glutInitWindowSize(480, 480); // besarnya jendela
226     glutInitWindowPosition(100, 100); // posisi jendela
227     // judul jendela (wajib diubah dengan informasi NAME)
228     glutCreateWindow("2200018048 / Augustav Fahrul - PRAKTIKUM 08 GRAFIK");
229
230     // panggil fungsi init untuk initialisasi awal
231 }
```

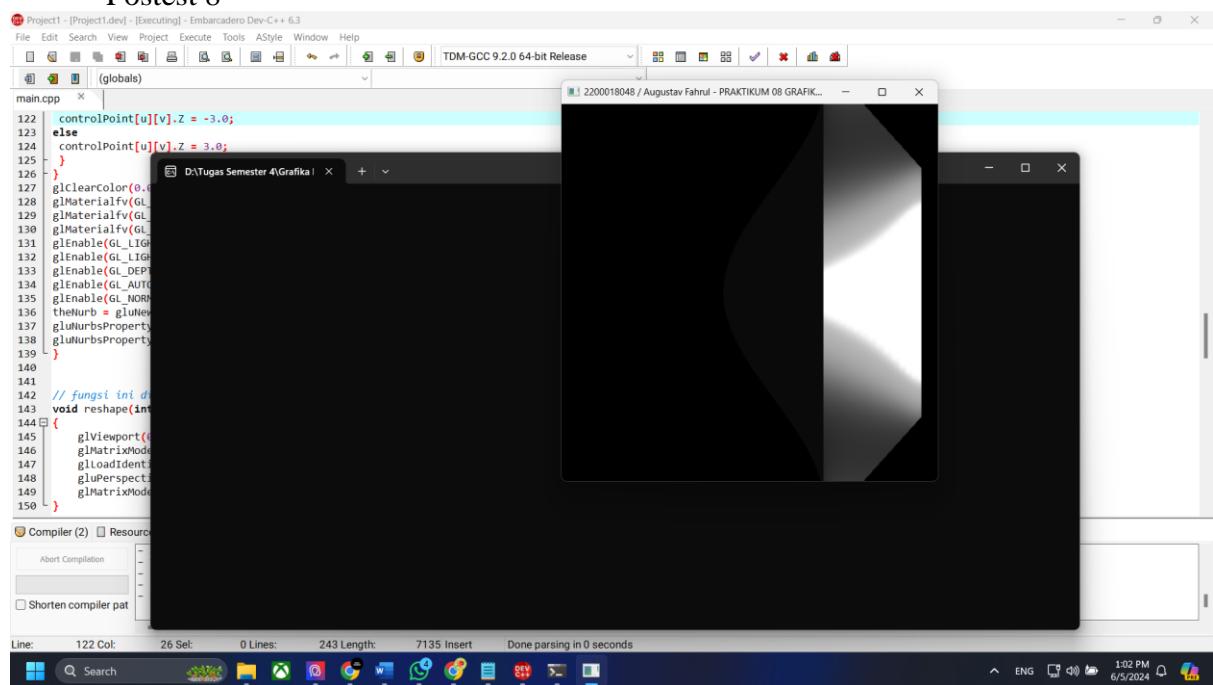
Compiler tab shows:

- Errors: 0
- Warnings: 0
- Output Filename: D:\Tugas Semester 4\Grafika\2200018048 / Augustav Fahrul - PRAKTIKUM 08 GRAFIK
- Output Size: 337.1962890625 KIB
- Compilation Time: 1.25s

Line: 211 Col: 1 Sel: 0 Lines: 243 Length: 7135 Insert Done parsing in 0 seconds

Pada codingan diatas, asisten memberikan codingan yang dimana membuat sebuah gambar yang mencekung ke depan. Poin paling penting pada cekungan ini ada pada bagian void init(void), karena pada bagian tersebut ada codingan yang mendeklarasikan gambarnya mencekung ke dalam atau keluar, rata atau tidaknya gambar tersebut.

### - Postest 8



The screenshot shows the Embarcadero Dev-C++ IDE interface. The main window displays the code for `main.cpp`, which includes OpenGL initialization, rendering, and reshape functions. A preview window shows a dark, reflective surface with a bright light source, creating a strong highlight and shadow effect. The status bar at the bottom indicates the code has been parsed successfully.

```
122     controlPoint[u][v].Z = -3.0;
123 else
124     controlPoint[u][v].Z = 3.0;
125 }
126
127 glClearColor(0.0,
128 glMaterialfv(GL_FRONT, GL_AMBIENT, black);
129 glMaterialfv(GL_FRONT, GL_DIFFUSE, white);
130 glMaterialfv(GL_FRONT, GL_SPECULAR, white);
131 glEnable(GL_LIGHTING);
132 glEnable(GL_COLOR_MATERIAL);
133 glEnable(GL_DEPTH_TEST);
134 glEnable(GL_AUTO_NORMAL);
135 glEnable(GL_NORM);
136 theNurb = gluNewNurbsRenderer();
137 gluNurbsProperty(theNurb, GLU_NURBS_RENDERER, GLU_NURBS_RENDERER);
138 gluNurbsProperty(theNurb, GLU_NURBS_RENDERER, GLU_NURBS_RENDERER);
139
140
141 // fungsi ini dibutuhkan
142 void reshape(int width, int height)
143 {
144     glViewport(0, 0, width, height);
145     glMatrixMode(GL_MODELVIEW);
146     glLoadIdentity();
147     gluPerspective(45.0, width / height, 0.1, 100.0);
148     glMatrixMode(GL_MODELVIEW);
149 }
```

Compiler tab shows:

- Abort Compilation
- Shorten compiler path

Line: 122 Col: 26 Sel: 0 Lines: 243 Length: 7135 Insert Done parsing in 0 seconds

Pada bagian postest, asisten meminta gambarnya agar menjadi rata, caranya ialah dengan mengganti sedikit angka yang ada di codingan berikut :

```
if ((u == 1 || u == 2) && (v == 1 || v == 2))
controlPoint[u][v].Z = 3.0;
else
controlPoint[u][v].Z = -3.0;
}
```

Menjadi

```
if ((u == 1 || u == 2) && (v == 1 || v == 2))
controlPoint[u][v].Z = -3.0;
else
controlPoint[u][v].Z = 3.0;
}
```

Jadi pada codingan diatas, diinisialisasikan untuk yang awal yaitu minusnya di point kedua(dalam else) maka gambar akan mencekung ke depan atau pada bagian yang terang.

Dan jika codingan diatas diganti minusnya ke bagian if dan else nya menjadi positif, maka gambar akan mencekung ke belakang seperti gambar yang sudah kita lihat.