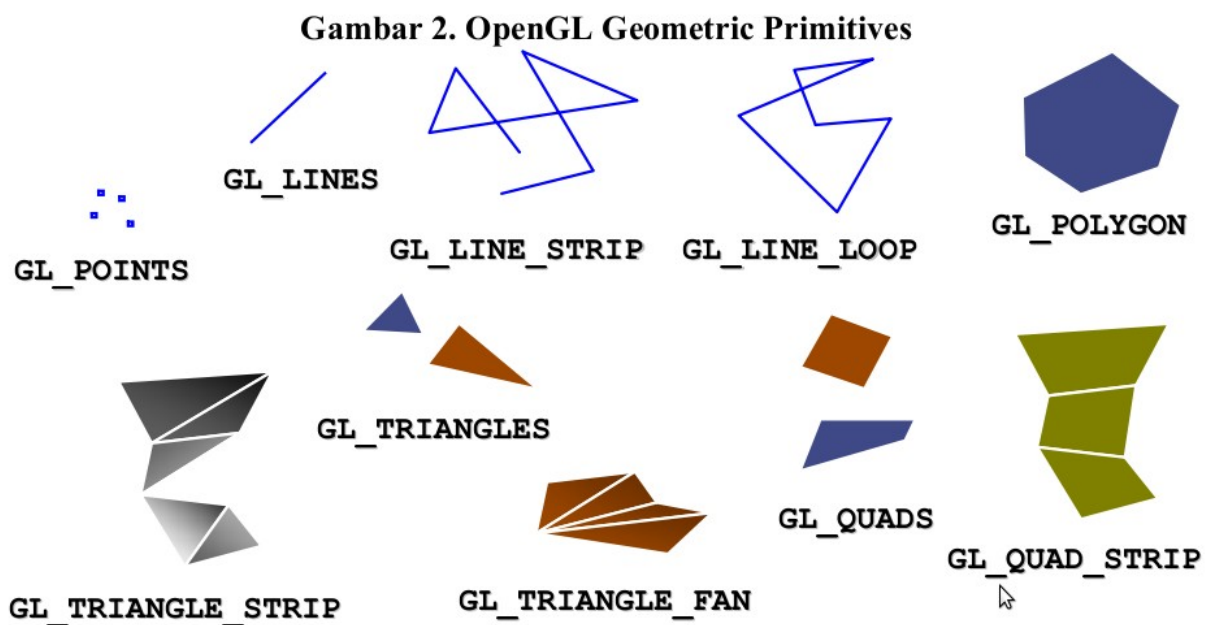


# Tutorial 02

## OpenGL Primitives

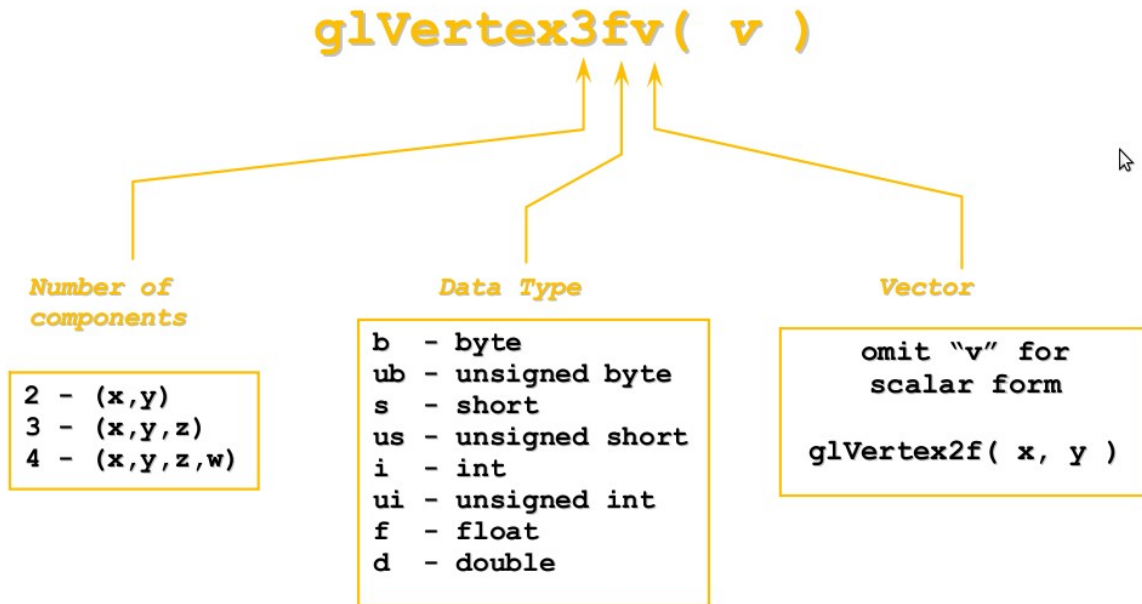
Pada tutorial 01 telah diberikan contoh program untuk merepresentasikan model obyek segiempat 2D. OpenGL memiliki beberapa komponen dasar untuk merepresentasikan suatu obyek. Komponen dasar tersebut, disebut sebagai OpenGL Geometric primitives. Gambar 2 menggambarkan semua OpenGL Geometric primitives yang tersedia.



*Gambar 2.1 OpenGL Primitives*

Setiap obyek harus dimodelkan sebagai kombinasi dari komponen-komponen dasar tersebut. Sebagai contoh, obyek segiempat pada tutorial 01 tersebut dimodelkan dengan menggunakan komponen dasar `GL_POLYGON`. Obyek tersebut dapat pula dimodelkan dengan komponen dasar `GL_TRIANGLES` atau pun `GL_QUAD`.

Hingga saat ini kita belum menerangkan secara detail masing-masing fungsi OpenGL. Secara umum perintah-perintah dalam OpenGL memenuhi aturan sebagai berikut:



*Gambar 2.2 OpenGL Command*

Dalam OpenGL, menggambar geometric primitives selalu dilakukan di antara fungsi

```

glBegin(PRIMITIVES)
// Fungsi Menggambar Primitives di sini
glEnd()
  
```

Setiap OpenGL geometric primitive disesifikasi oleh urutan vertex-vertex-nya dalam bentuk urutan koordinat homogenous. Koordinat homogenous adalah koordinat dalam bentuk ( x, y, z, w ). Setiap primitive memiliki standar tentang bagaimana vertex-vertex diorganisasikan. Program 02 dan Program 03 memberikan contoh bagaimana memodelkan primitive segitiga dan segidelapan.

```

/*
Modul GPU
Program 02
segitiga
*/

#include <GL/glut.h>

void renderScene();

int main(int argc, char* argv[])
{
    glutInit(&argc, argv);
    glutCreateWindow("Program 02");
  
```

```

        glutDisplayFunc(renderScene);
        glutMainLoop();

        return 0;
    }

void renderScene()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_TRIANGLES);
        glVertex2f(-0.5f, -0.5f);
        glVertex2f(0.5f, -0.5f);
        glVertex2f(0.0f, 0.25f);
    glEnd();
    glFlush();
}

/*
Modul GPU
Program 03
Polygon
*/

#include <GL/glut.h>

void renderScene();

int main(int argc, char* argv[])
{
    glutInit(&argc, argv);
    glutCreateWindow("Program 03");
    glutDisplayFunc(renderScene);
    glutMainLoop();

    return 0;
}

void renderScene()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_POLYGON);
        glColor3f(0.0f, 1.0f, 1.0f);
        glVertex2f(-0.5f, -0.5f);
        glVertex2f(-0.75f, 0.0f);
        glVertex2f(-0.5f, 0.5f);
        glVertex2f(0.0f, 0.75f);
        glVertex2f(0.5f, 0.5f);
    glEnd();
}

```

```

        glVertex2f(0.75f, 0.0f);
        glVertex2f(0.5f, -0.5f);
        glVertex2f(0.0f, -0.75f);
    glEnd();
    glFlush();
}

```

Perhatikan urutan dari vertex untuk setiap jenis OpenGL Geometric Primitive.

**Tugas: Buat Program untuk menggambar jenis OpenGL Geometric Primitive yang lain.**

**Bonus:** glColor3f() adalah fungsi untuk menentukan warna yang berlaku hingga fungsi berikutnya. Program 04 berikut adalah program yang sama dengan Program 03 hanya di sini setiap vertex diberi warna yang berbeda.

```

/*
Modul GLUT
Program 04
Polygon aneka warna
*/

#include <GL/glut.h>

void renderScene();

int main(int argc, char* argv[])
{
    glutInit(&argc, argv);
    glutCreateWindow("Program 04");
    glutDisplayFunc(renderScene);
    glutMainLoop();

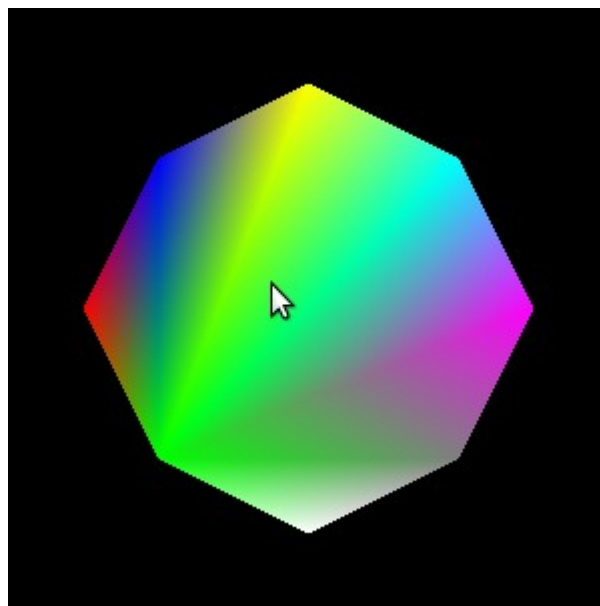
    return 0;
}

void renderScene()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_POLYGON);
        glColor3f(0.0f, 1.0f, 0.0f);
        glVertex2f(-0.5f, -0.5f);
        glColor3f(1.0f, 0.0f, 0.0f);
        glVertex2f(-0.75f, 0.0f);
        glColor3f(0.0f, 0.0f, 1.0f);
        glVertex2f(-0.5f, 0.5f);
        glColor3f(1.0f, 1.0f, 0.0f);
        glVertex2f(0.0f, 0.75f);
        glColor3f(0.0f, 1.0f, 1.0f);
        glVertex2f(0.5f, 0.5f);
    glEnd();
}

```

```
    glColor3f(1.0f, 0.0f, 1.0f);  
    glVertex2f(0.75f, 0.0f);  
    glColor3f(0.5f, 0.5f, 0.5f);  
    glVertex2f(0.5f, -0.5f);  
    glColor3f(1.0f, 1.0f, 1.0f);  
    glVertex2f(0.0f, -0.75f);  
glEnd();  
glFlush();  
}
```

Hasil screenshot bisa dilihat pada gambar berikut ini:



*Gambar 2.3 Screenshot Polygon Warna*