

## Modul 2 Primitive Objects

### A. KOMPETENSI DASAR

- Memahami jenis-jenis primitive objects.
- Memahami dan dapat membuat objek primitif.
- Memahami penerapan Objek primitif menjadi bangun 2 dimensi dasar.
- Memahami dan dapat membuat fungsi untuk bangun 2 dimensi dasar.

### B. ALOKASI WAKTU

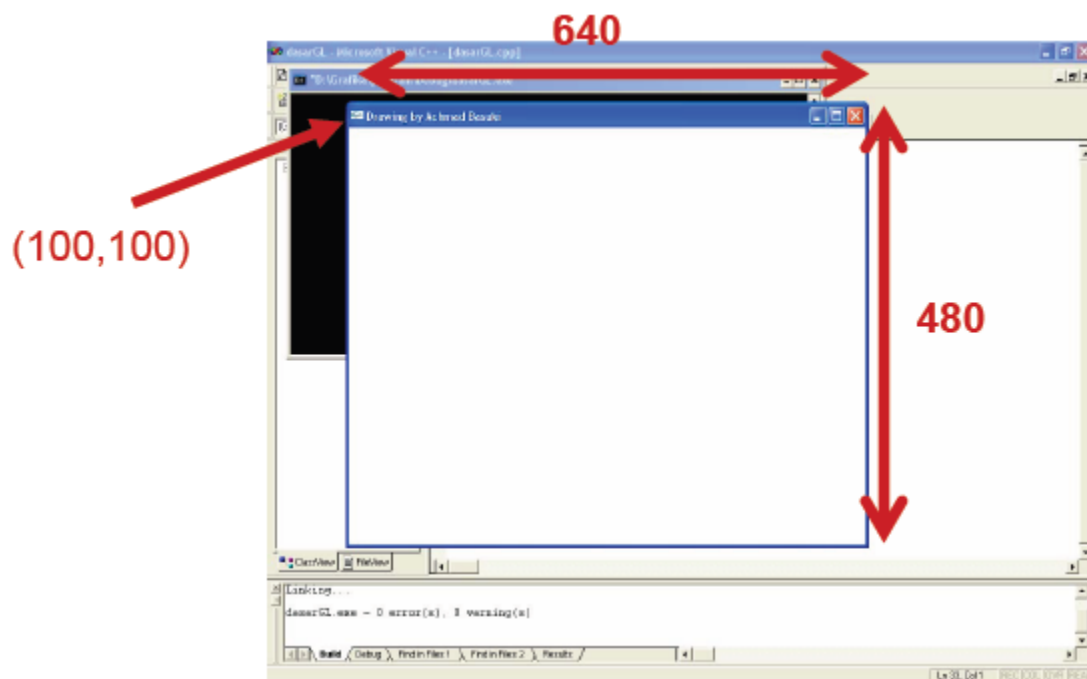
2 JS (2x50 menit)

### C. PETUNJUK

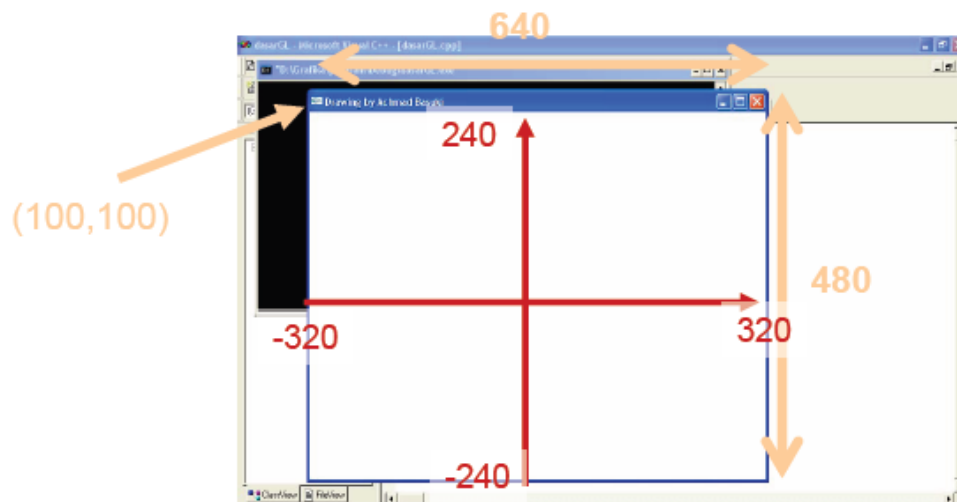
- Awali setiap aktivitas dengan do'a, semoga berkah dan mendapat kemudahan.
- Pahami Tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar.
- Kerjakan tugas-tugas dengan baik, sabar, dan jujur.
- Tanyakan kepada asisten/dosen apabila ada hal-hal yang kurang jelas.

### D. DASAR TEORI

1. Sistem Koordinat OpenGL



2. Fungsi `gluOrtho2D(left, right, bottom, top) → gluOrtho2D(-320, 320, -240, 240)`



3. Fungsi dasar menggambar titik adalah `glVertex?()`

a. `glVertex2i(x, y)`

Menggambar titik pada koordinat x dan y, dengan nilai satuan berupa **integer**.

Contoh `glVertex2i(10, 10);`

b. `glVertex2f(x, y)`

Menggambar titik pada koordinat x dan y, dengan nilai satuan berupa float.

Contoh `glVertex2f(10.0, 10.0);`

c. `glVertex2d(x, y)`

Menggambar titik pada koordinat x dan y, dengan nilai satuan berupa double.

Contoh `glVertex2d(10.0, 10.0);`

4. Objek primitif

- a. **Titik** : `GL_POINTS`
- b. **Garis** : `GL_LINES`
- c. **Poliline** : `GL_LINE_STRIP`
- d. **Poligon (no fill)** : `GL_LINE_LOOP`
- e. **Triangle** : `GL_TRIANGLES`

*Triangle* atau segitiga adalah tiga buah titik yang terhubung menjadi suatu segitiga dengan blok di tengahnya.

- f. **Quads** : `GL_QUADS`

Quad atau segiempat adalah empat buah titik yang terhubung menjadi suatu segiempat dengan blok di tengahnya.

- g. **Polygon** : `GL_POLYGON`

Polygon merupakan suatu fungsi yang mirip dengan polyline, tetapi menghasilkan kurva tertutup dengan blok warna (*fill*).

## E. AKTIFITAS KELAS PRAKTIKUM

### 1. PRIMITIVE OBJEK

- 1) Berikut adalah script dasar pada fungsi `Main ()` untuk kegiatan praktikum primitive objek

```
glutInitWindowPosition(100,100);
glutInitWindowSize(640,480);
    dan
gluOrtho2D(-320.,320.,-240.,240.);
```

- 2) Buatlah project baru pada Visual Studio dengan nama **prak1-Points**. Dan tambahkan fungsi callback untuk `glutDisplayFunc(drawDot)`; pada fungsi main. Tampilkan hasilnya berupa screenshot. Jelaskan Proses Rendering.

```
void drawDot() {
    glBegin(GL_POINTS);
    glVertex2i(50,0);
    glVertex2i(50,50);
    glVertex2i(0,0);
    glEnd();
    glFlush();
}
```

- 3) Buatlah project baru pada Visual Studio dengan nama **prak1-PointSize**. Dan tambahkan fungsi `glPointSize(int size)`; untuk mengubah ukuran objek Points (**perhatian : ganti int size dengan nilai int yang diinginkan, sehingga fungsi menjadi `glPointSize(5)`**);). Pindah-pindahkan `glPointSize` pada posisi a, b, dan c. Sajikan Screen shot hasil a, b, c dan simpulkan.

```
[a]
void drawDot() {
    [b]
    glBegin(GL_POINTS);
        glVertex2i(50, 0);
        [c]
        glVertex2i(50, 50);
        glVertex2i(0, 0);
    glEnd();
}
```

#### Petunjuk Pengerjaan

Untuk Kegiatan Praktikum Langkah 4 dan selanjutnya. Isikan nilai `x0, y0, x1, y1, ... dst`, dengan nilai koordinat ( dalam format bilangan desimal) sesuai dengan kreatifitas masing-masing.

- 4) Buatlah project baru pada Visual Studio dengan nama **prak1-Lines**. Dan tambahkan fungsi callback untuk `glutDisplayFunc(drawLine)`; . Tampilkan hasilnya berupa screenshot. Dan Jelaskan Proses Rendering untuk vertexnya.

```
void drawLine() {
    glBegin(GL_LINES);
    glVertex2d(x0, y0);
    glVertex2d(x1, y1);
    glVertex2d(x2, y2);
    glEnd();
}
```

- 5) Buatlah project baru pada Visual Studio dengan nama **prak1-LineWidth**. Dan tambahkan fungsi **glLineWidth(int size)**; untuk mengubah ukuran objek Points. Pindah-pindahkan **glLineWidth** pada a,b, dan c. Sajikan Screen shot hasil a,b,c dan simpulkan.

```
void drawLine() {
    [a]
    glBegin(GL_LINES);
    glVertex2d(x0, y0);

    glVertex2d(x1, y1);
    [b]
    glVertex2d(x2, y2);
    [c]
    glEnd();
}
```

- 6) Buatlah project baru pada Visual Studio dengan nama **prak1-LineStrip**. Buatlah agar hasil yang ditampilkan membentuk **bangun simetris terbuka**. Sajikan screenshotnya. Dan Jelaskan proses rendering vertexnya.

```
void drawPoliline() {
    glBegin(GL_LINE_STRIP);
    glVertex2d(x0, y0);
    glVertex2d(x1, y1);
    glVertex2d(x2, y2);
    glVertex2d(x3, y3);

    glEnd();
}
```

- 7) Buatlah project baru pada Visual Studio dengan nama **prak1-Polygon**. Buatlah agar hasil yang ditampilkan membentuk **bangun simetris**. Sajikan screenshotnya. Dan jelaskan proses rendering vertexnya.

```
glBegin(GL_LINE_LOOP);
    glVertex2d(x0, y0);
    glVertex2d(x1, y1);
    glVertex2d(x2, y2);
    glVertex2d(x3, y3);
    glVertex2d(x4, y4);
    glEnd();
```

- 8) Buatlah project baru pada Visual Studio dengan nama **prak1-Triangle**. Buatlah agar hasil yang ditampilkan membentuk **2 segitiga yang terpisah**. Sajikan screenshotnya. Dan jelaskan proses rendering vertexnya.

```
glBegin(GL_TRIANGLES);
    glVertex2d(x0, y0);
    glVertex2d(x1, y1);
    glVertex2d(x2, y2);
    glVertex2d(x3, y3);
    glVertex2d(x4, y4);
    glVertex2d(x5, y5);
glEnd();
```

- 9) Buatlah project baru pada Visual Studio dengan nama **prak1-TriangleStrip**. Buatlah agar hasil yang ditampilkan membentuk **Bangun Bebas**. Sajikan screenshotnya. Dan jelaskan proses rendering vertexnya.

```
glBegin(GL_TRIANGLE_STRIP);
    glVertex2d(x0, y0);
    glVertex2d(x1, y1);
    glVertex2d(x2, y2);
    glVertex2d(x3, y3);
    glVertex2d(x4, y4);
    glVertex2d(x5, y5);
glEnd();
```

- 10) Buatlah project baru pada Visual Studio dengan nama **prak1-TriangleFan**. Buatlah agar hasil yang ditampilkan membentuk **Bangun Bebas**. Sajikan screenshotnya. Dan jelaskan proses rendering vertexnya.

```
glBegin(GL_TRIANGLE_FAN);
    glVertex2d(x0, y0);
    glVertex2d(x1, y1);
    glVertex2d(x2, y2);
    glVertex2d(x3, y3);
    glVertex2d(x4, y4);
    glVertex2d(x5, y5);
glEnd();
```

- 11) Buatlah project baru pada Visual Studio dengan nama **prak1-Quads**. Buatlah agar hasil yang ditampilkan membentuk **2 Bangun Segi Empat Terpisah**. Sajikan screenshotnya. Dan jelaskan proses rendering vertexnya.

```
glBegin(GL_QUADS);
    glVertex2d(x0, y0);
    glVertex2d(x1, y1);
    glVertex2d(x2, y2);
    glVertex2d(x3, y3);
    glVertex2d(x4, y4);
    glVertex2d(x5, y5);
    glVertex2d(x6, y6);
    glVertex2d(x7, y7);
glEnd();
```

- 12) Buatlah project baru pada Visual Studio dengan nama **prak1-QuadStrip**. Buatlah agar hasil yang ditampilkan membentuk **Bangun Bebas**. Sajikan screenshotnya. Dan jelaskan proses rendering vertexnya.

```
glBegin(GL_QUAD_STRIP);
    glVertex2d(x0,y0);
    glVertex2d(x1,y1);
    glVertex2d(x2,y2);
    glVertex2d(x3,y3);
    glVertex2d(x4,y4);
    glVertex2d(x5,y5);
glEnd();
```

- 13) Buatlah project baru pada Visual Studio dengan nama **prak1-Poligon**. Buatlah agar hasil yang ditampilkan membentuk **Bangun Bebas**. Sajikan screenshotnya. Dan jelaskan proses rendering vertexnya.

```
glBegin(GL_POLYGON);
    glVertex2d(x0,y0);
    glVertex2d(x1,y1);
    glVertex2d(x2,y2);
    glVertex2d(x3,y3);
    glVertex2d(x4,y4);
    glVertex2d(x5,y5);
    glVertex2d(x6,y6);
    glVertex2d(x7,y7);
glEnd();
```

## F. TUGAS PRAKTIKUM

Contoh program membuat segi empat.

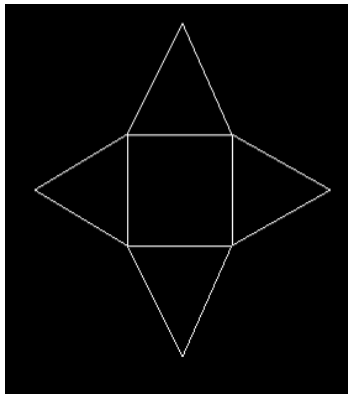
```
#include <stdlib.h>
#include <GL/glut.h>

void userdraw()
{
    static int tick=0;
    glBegin(GL_LINE_STRIP);
        glVertex3f(-90,90,0);
        glVertex3f(-90,0,0);
        glVertex3f(80,0,0);
        glVertex3f(80,90,0);
    glEnd();
}

void display(void) {
    glClear(GL_COLOR_BUFFER_BIT);
    userdraw();
    glutSwapBuffers();
}
```

```
void main (int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB);
    glutInitWindowPosition(300,150);
    glutInitWindowSize(640,480);
    glutCreateWindow("Pemrograman Grafis GLUT");
    gluOrtho2D(-640.,640,-240.,240.);
    glutIdleFunc(display);
    glutDisplayFunc(display);
    glutMainLoop();
}
```

- a. Troubleshooting program di atas sehingga bisa menjadi sebuah segi empat. Bagian mana yang harus diubah? Jelaskan alasannya.
- b. Kembangkan program di atas agar dapat menjadi jaring-jaring bangun limas seperti gambar di bawah ini. Sertakan source code dan gambarnya!



- c. Carilah gambar jaring-jaring prisma segitiga. Sertakan gambar aslinya, kemudian implementasikan ke dalam program. Sajikan hasil screenshot serta hasil programnya.