

## Modul 3

# Objek 2 Dimensi

### A. KOMPETENSI DASAR.

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

### B. ALOKASI WAKTU

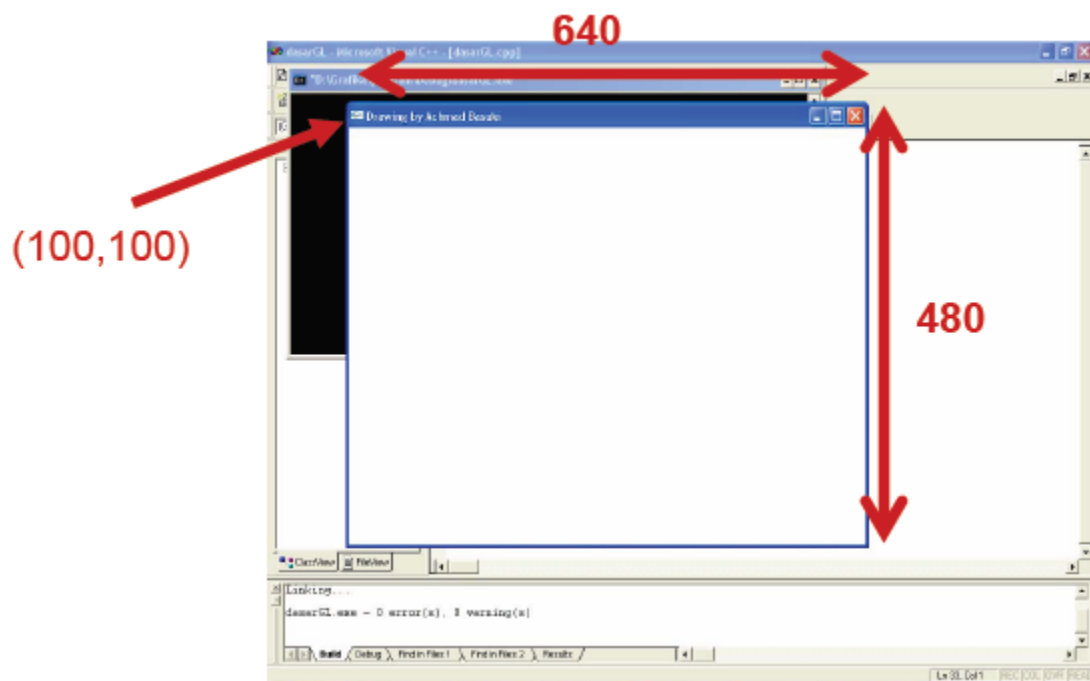
2 JS (2x50 menit)

### C. PETUNJUK

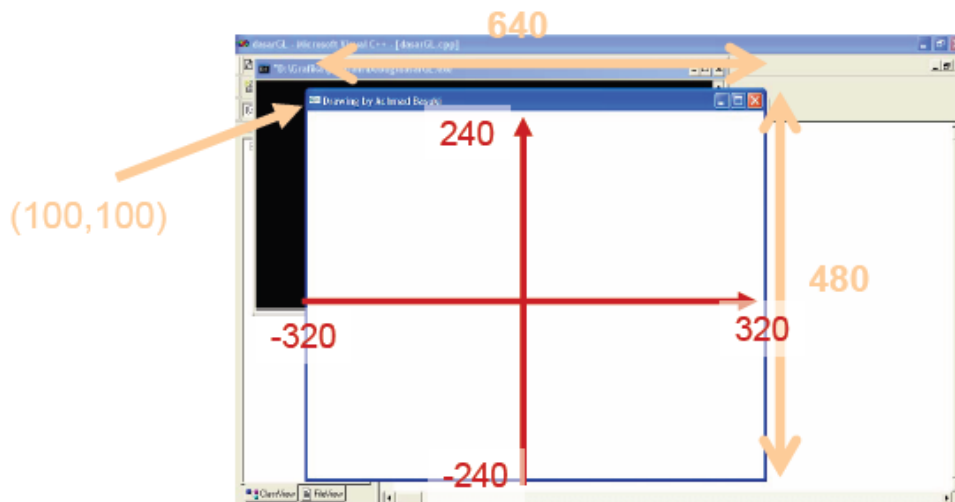
- Awali setiap aktivitas dengan do'a, semoga berkah dan mendapat kemudahan.
- Pahami Tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar.
- Kerjakan tugas-tugas dengan baik, sabar, dan jujur.
- Tanyakan kepada asisten/dosen apabila ada hal-hal yang kurang jelas.

### 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*).

#### h. Color

Untuk memberi warna pada objek, seperti titik atau garis, dapat dilakukan dengan menggunakan fungsi: `glColor3f(red, green, blue);`

Di mana red, green, blue berada pada 0 sampai dengan 1, yang menunjukkan skala pencerahan dari masing-masing skala.

```
glColor3f(0.,0.,0.); //black
glColor3f(0.,0.,1.); //blue
glColor3f(0.,1.,0.); //green
glColor3f(0.,1.,1.); //cyan
glColor3f(1.,0.,0.); //red
glColor3f(1.,0.,1.); //magenta
glColor3f(1.,1.,0.); //yellow
glColor3f(1.,1.,1.); //white
```

## E. AKTIFITAS KELAS PRAKTIKUM

### 1. OBJEK 2D

Untuk membangun sebuah bangun 2 Dimensi dengan efisien, maka diperlukan sebuah fungsi yang memiliki parameter tertentu untuk membangun bangun tersebut. Perhatikan kode program bangun 2D dengan menggunakan fungsi berikut ini:

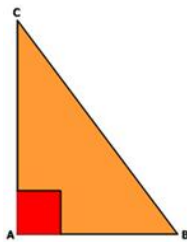
```
void segiEmpat(int posx, int posy, int w, int h){
    glBegin(GL_QUADS);
        glVertex2i(posx, posy); //1 -> 1
        glVertex2i(w+posx, posy); //4 -> 2
        glVertex2i(w+posx, posy-h); //2 -> 3
        glVertex2i(posx, posy-h); //3 -> 4
    glEnd();
}

void renderScene(void) {
    glClear(GL_COLOR_BUFFER_BIT);
    /* ... */
    glColor3f(1.,0.,0.);
    segiEmpat(-200,200,100,100);
    glColor3f(0.,1.,0.);
    segiEmpat(100,200,150,150);
    glColor3f(0.,0.,1.);
    segiEmpat(0,0,50,50);
    glFlush();
}
```

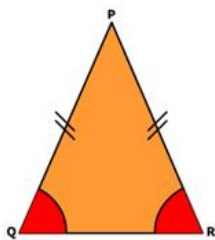
1. Buatlah project baru pada Visual Studio. Buatlah agar tampilan obyek 2 dimensi berbentuk **PERSEGI PANJANG** seperti di bawah ini, gunakan primitif dan struktur vertex **yang paling efisien**. Tulis program dan Tampilkan hasilnya berupa screenshot yang berupa tampilan 3 bangun dengan posisi, dimensi, dan warna yang berbeda.



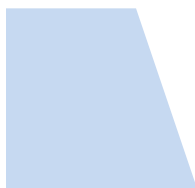
2. Buatlah project baru pada Visual Studio. Buatlah agar tampilan obyek 2 dimensi berbentuk **SEGITIGA SIKU-SIKU** seperti di bawah ini, gunakan primitif dan struktur vertex **yang paling efisien**. Tulis program dan Tampilkan hasilnya berupa screenshot yang berupa tampilan 3 bangun dengan posisi, dimensi, dan warna yang berbeda.



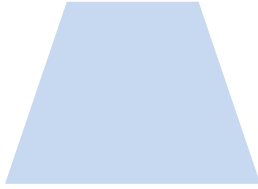
3. Buatlah project baru pada Visual Studio. Buatlah agar tampilan obyek 2 dimensi berbentuk **SEGITIGA SAMA KAKI** seperti di bawah ini, gunakan primitif dan struktur vertex **yang paling efisien**. Tulis program dan Tampilkan hasilnya berupa screenshot yang berupa tampilan 3 bangun dengan posisi, dimensi, dan warna yang berbeda.



4. Buatlah project baru pada Visual Studio. Buatlah agar tampilan obyek 2 dimensi berbentuk **TRAPESIUM SIKU** seperti di bawah ini gunakan primitif dan struktur vertex **yang paling efisien**. Tulis program dan Tampilkan hasilnya berupa screenshot yang berupa tampilan 3 bangun dengan posisi, dimensi, dan warna yang berbeda.



5. Buatlah project baru pada Visual Studio. Buatlah agar tampilan obyek 2 dimensi berbentuk **TRAPESIUM SAMA KAKI** seperti di bawah ini, gunakan primitif dan struktur vertex **yang paling efisien**. Tulis program dan Tampilkan hasilnya berupa screenshot yang berupa tampilan 3 bangun dengan posisi, dimensi, dan warna yang berbeda.

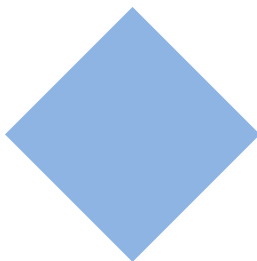


6. Buatlah project baru pada Visual Studio. Buatlah agar tampilan obyek 2 dimensi berbentuk **JAJAR GENJANG** seperti di bawah ini, gunakan primitif dan struktur vertex **yang paling efisien**. Tulis program dan Tampilkan hasilnya berupa screenshot yang berupa tampilan 3 bangun dengan posisi, dimensi, dan warna yang berbeda.

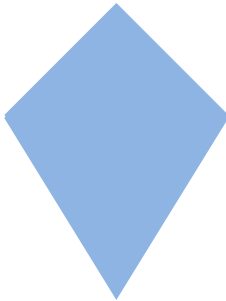


## F. TUGAS ASISTENSI

1. Buatlah sebuah fungsi untuk membuat tampilan objek 2 dimensi berbentuk belah ketupat seperti di bawah ini, gunakan primitif dan struktur vertex **yang paling efisien**. Tulis program dan Tampilkan hasilnya berupa screenshot yang berupa tampilan 3 bangun dengan posisi, dimensi, dan warna yang berbeda.



2. Buatlah sebuah fungsi untuk membuat tampilan objek 2 dimensi berbentuk layang-layang seperti di bawah ini, gunakan primitif dan struktur vertex **yang paling efisien**. Tulis program dan Tampilkan hasilnya berupa screenshot yang berupa tampilan 3 bangun dengan posisi, dimensi, dan warna yang berbeda.



3. Buatlah Fungsi untuk lingkaran dengan menentukan struktur vertex dan paramater yang digunakan dalam fungsinya.

