

Nama : Khaisan Ramzi Muzhaffar

Nim : 241011400049

Kelas : 02TPLP001

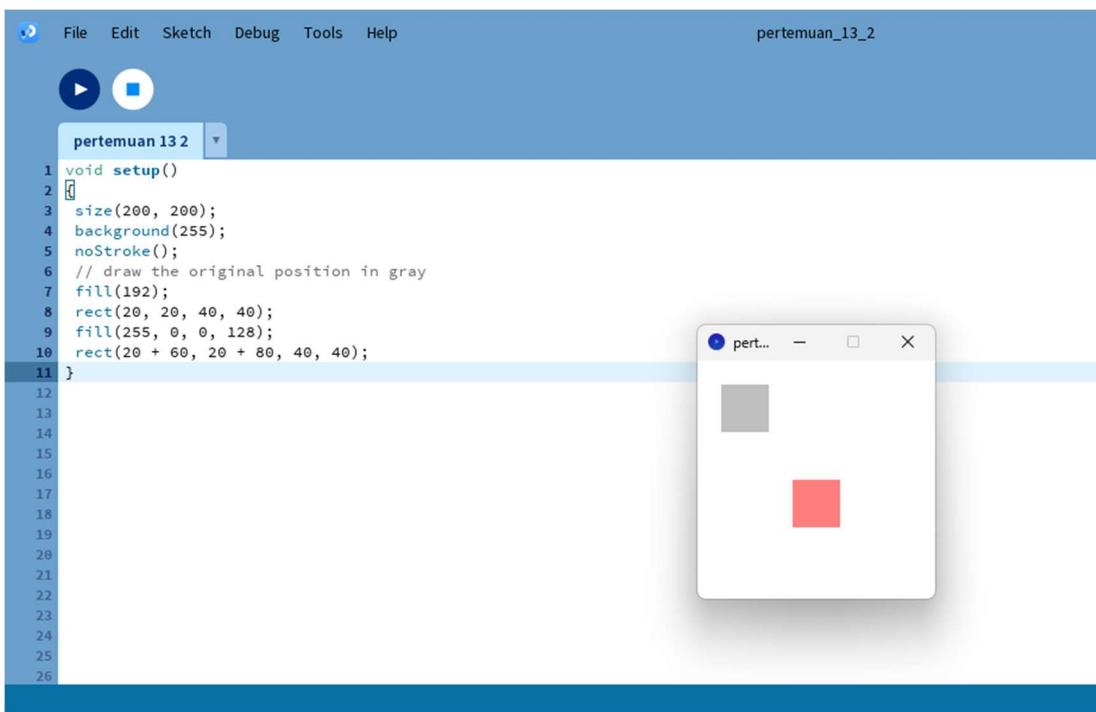
Matkul : Komputer Grafik

Dosen : Dede Supiyan, S.kom., M.Kom.

## TUGAS PERTEMUAN 13

### 1) Translation

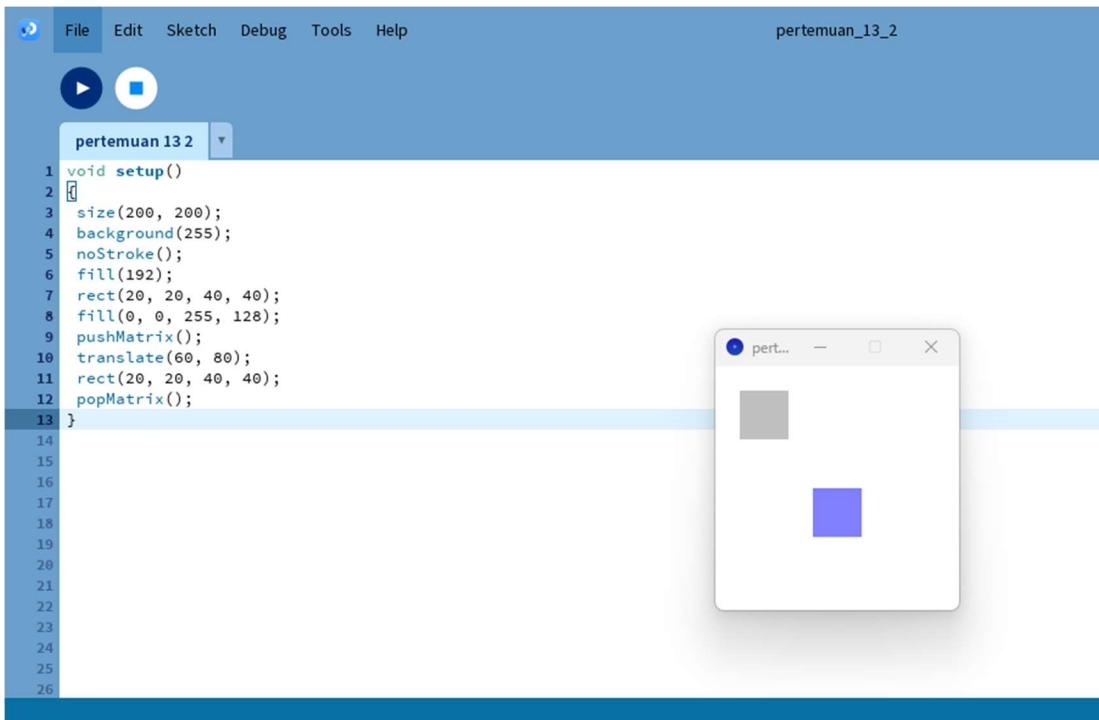
Fungsi translation() digunakan untuk menggeser titik awal atau titik acuan (0,0) dari sistem koordinat ke lokasi baru yang akan ditentukan. Dengan kata lain, ia memungkinkan Anda untuk memindahkan "asal" dari mana semua objek lain digambar atau dihitung, tanpa harus mengubah koordinat setiap objek secara individual. Ini sangat berguna dalam grafik komputer dan desain untuk menyederhanakan penempatan dan transformasi elemen.



```
File Edit Sketch Debug Tools Help pertemuan_13_2 pertemuan_13_2
1 void setup()
2 {
3     size(200, 200);
4     background(255);
5     noStroke();
6     // draw the original position in gray
7     fill(192);
8     rect(20, 20, 40, 40);
9     fill(255, 0, 0, 128);
10    rect(20 + 60, 20 + 80, 40, 40);
11 }
```

## Contoh 1 Program Translation

Pada gambar tersebut menampilkan sebuah output di jendela terpisah di sebelah kanan. Jadi, pada kode ini menggambarkan dua persegi pada layar, satu abu-abu di layar bagian atas dan satu merah muda di layar bagian bawah.



```
File Edit Sketch Debug Tools Help pertemuan_13_2
pertemuan 13_2
1 void setup()
2 {
3     size(200, 200);
4     background(255);
5     noStroke();
6     fill(192);
7     rect(20, 20, 40, 40);
8     fill(0, 0, 255, 128);
9     pushMatrix();
10    translate(60, 80);
11    rect(20, 20, 40, 40);
12    popMatrix();
13 }
```

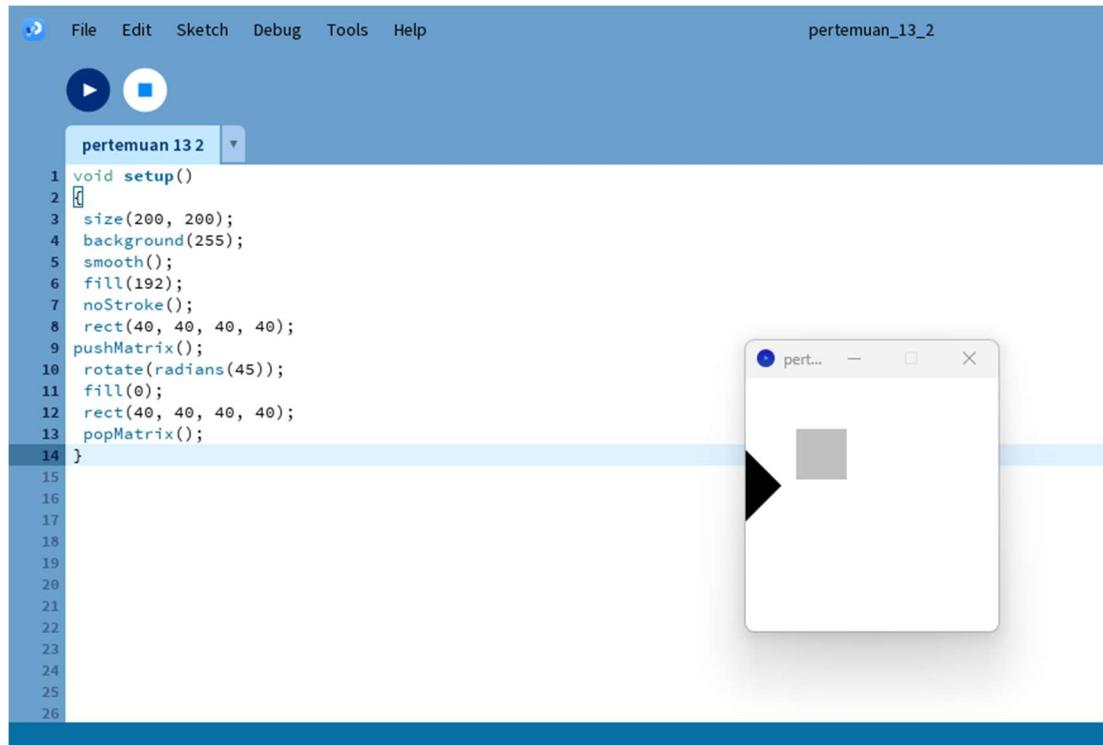
## Contoh 2 Progam Translation

Pada gambar kedua ini menampilkan sebuah output yang sama namun berbeda pada kode dan warna, ini menggambarkan dua persegi pada layar, satu abu-abu di bagian atas dan satu biru muda di bagian bawah, kode ini menggunakan fungsi translate() untuk menggeser sistem koordinat dan menggambar persegi biru di lokasi yang tergeser, hingga kemudian mengembalikan sistem koordinat ke keadaan semula.

## 2) Rotasi

Fungsi `rotate()` adalah perintah di Processing yang digunakan untuk memutar objek di sekitar titik pusatnya. Bayangkan ini seperti kamu memutar sebuah piringan hitam di porosnya.

Secara *default*, `rotate()` akan memutar semua yang kamu gambar setelahnya mengelilingi titik (0,0) pada kanvas Processing, yaitu sudut kiri atas layar. Ini berarti, jika kamu punya kotak di posisi (20,20) dan kamu memutar kanvasnya, kotak itu akan berputar mengelilingi (0,0) dan posisi visualnya akan berubah.



```
File Edit Sketch Debug Tools Help
pertemuan_13_2
pertemuan 13 2
1 void setup()
2 {
3     size(200, 200);
4     background(255);
5     smooth();
6     fill(192);
7     noStroke();
8     rect(40, 40, 40, 40);
9     pushMatrix();
10    rotate(radians(45));
11    fill(0);
12    rect(40, 40, 40, 40);
13    popMatrix();
14 }
15
16
17
18
19
20
21
22
23
24
25
26
```

Contoh Program Rotate

Pada gambar ini terdapat dua output satu persegi abu-abu dan satu segitiga hitam, namun itu bukan sebuah segitiga melainkan itu adalah persegi yang diputar dengan titik (0,0) sebagai pusat putarannya, dan diputar 45 derajat.

### 3) Proses Rotasi Grid

Rotasi grid, artinya kita memutar seluruh sistem koordinat di sekeliling sebuah titik. Jadi, daripada repot-repot menghitung ulang posisi setiap bagian objek satu per satu untuk memutarnya, kita cukup memutar "dunia" tempat objek itu berada. Otomatis, objek akan ikut berputar juga. Ini seperti memutar meja, bukan memutar setiap barang yang ada di atasnya.

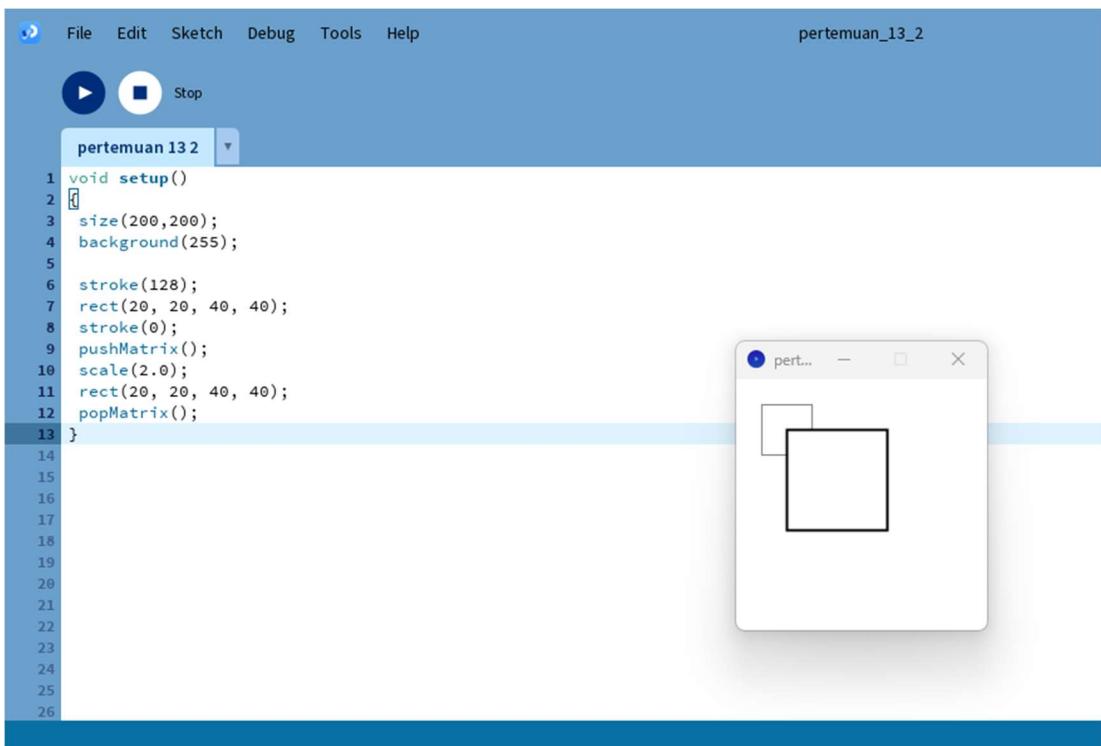
```
File Edit Sketch Debug Tools Help pertemuan_13_2 pertemuan 13 2 void setup() { size(200, 200); background(255); smooth(); fill(192); noStroke(); rect(40, 40, 40, 40); pushMatrix(); translate(40, 40); rotate(radians(45)); fill(0); rect(0, 0, 40, 40); popMatrix(); } 
```

Contoh Proses Rotasi Grid

Lihatlah gambar ini, dua persegi berdampingan di atas kanvas putih. Yang abu-abu terang diam tak bergerak, sementara yang hitam sudah berputar 45 derajat. Rahasianya? Bukan si kotak hitam yang diputar, tapi "dunianya" (sistem koordinat) yang berputar lebih dulu sebelum dia muncul.

#### 4) Scalling

Scalling itu tentang mengubah ukuran suatu objek atau gambar, baik memperbesar maupun memperkecil, tapi tanpa mengubah bentuk aslinya. Dalam dunia grafis komputer, ini dilakukan dengan mengalikan koordinat objek pakai faktor skala tertentu. Misalnya, kalau kamu pakai scale(2), objeknya akan jadi dua kali lebih besar. Sedangkan scale(0.5) bakal bikin objek itu setengah lebih kecil dari ukuran semula.



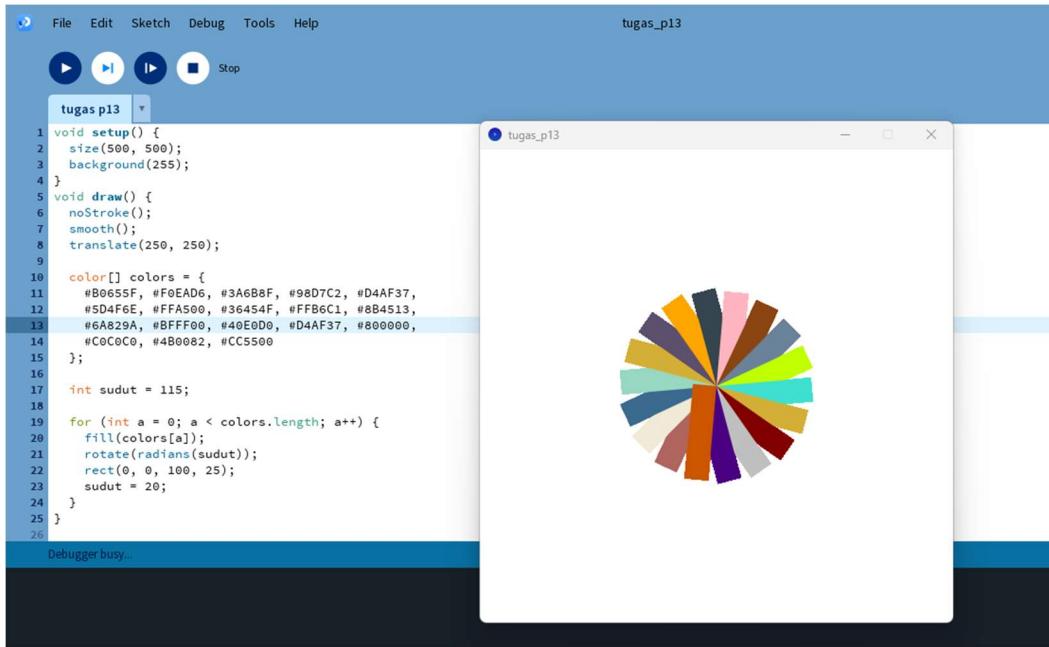
```
File Edit Sketch Debug Tools Help pertemuan_13_2 pertemuan_13_2 Stop pertemuan 13 2 void setup() { size(200,200); background(255); stroke(128); rect(20, 20, 40, 40); stroke(0); pushMatrix(); scale(2.0); rect(20, 20, 40, 40); popMatrix(); } 14 15 16 17 18 19 20 21 22 23 24 25 26
```

Contoh Program Scalling

Bisa dilihat di gambar ini, ada hasil dari kode Processing yang sederhana. Kode tersebut membuat dua persegi bergaris di atas kanvas putih berukuran 200x200 piksel.

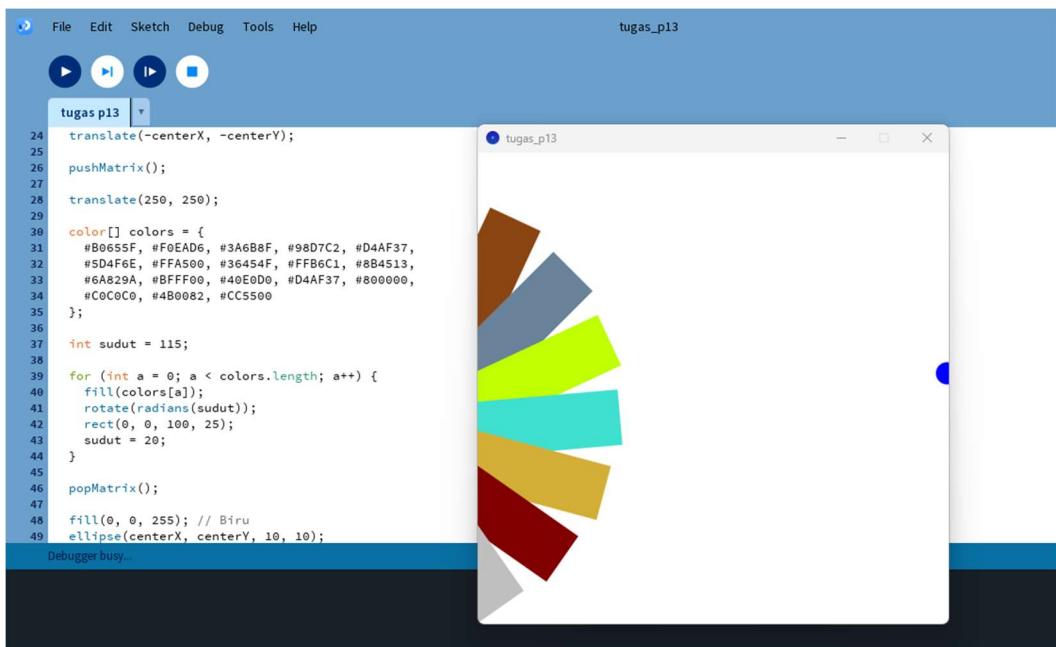
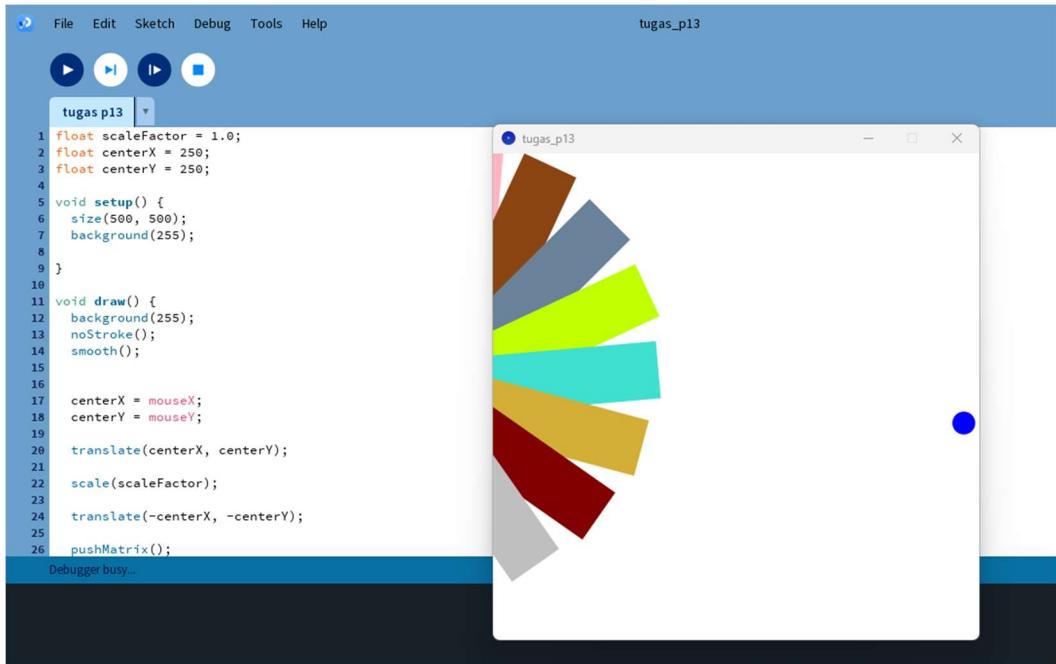
## TUGAS

- 1) Membuat Objek Lingkaran Dengan Berbagai Macam Variasi Warna.?



Program Processing ini dirancang untuk menciptakan semacam desain radial yang menyerupai roda atau kipas, terdiri dari 18 segmen berwarna-warni. Setiap segmen dibangun dari persegi panjang yang diorientasikan dengan rotasi presisi dari titik pusat (150,150). Dengan variasi warna pada setiap bilah, terbentuklah komposisi melingkar yang estetik. Secara fundamental, kode ini mendemonstrasikan aplikasi praktis dari transformasi rotasi dan teknik pewarnaan dalam lingkungan pemrograman grafis.

**2) Buatlah Program Untuk Melakukan Proses Scalling Dari Sembarang Titik Pusat Skala.?**





Output dari program ini adalah roda berbilah 18 yang kaya warna, namun yang menarik adalah bagaimana ia melakukan penskalaan (pembesaran). Proses ini tidak berpusat pada objek itu sendiri, melainkan pada sebuah titik referensi eksternal, katakanlah (200, 200). Untuk mencapai efek ini, program akan memanipulasi sistem koordinat: memindahkannya ke titik skala, menerapkan pembesaran, lalu mengembalikannya. Hasilnya, Anda akan melihat keseluruhan objek mengembang dari titik tersebut, mempertahankan pusat visualnya. Inilah metode ampuh untuk mengubah ukuran objek secara proporsional dari lokasi mana pun.