

Pertemuan 4

Koordinat

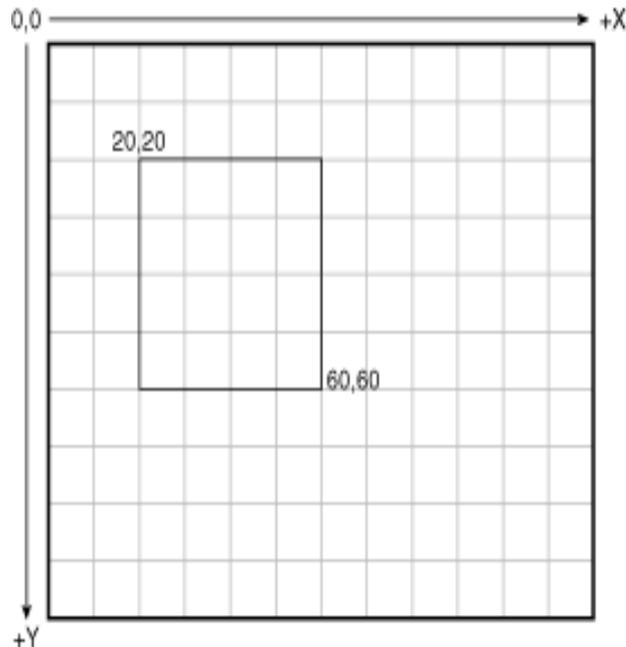
- memahami Koordinat gambar bidang Komputer grafik

Sistem Koordinat Grafik

- ❑ Untuk membangun sebuah objek pada layar, Anda harus memanggil salah satu dari metode yang tersedia didalam kelas Graphics.
- ❑ Pada semua metode drawing memiliki argumen yang mewakili titik-akhir, dan titik-awal dari sebuah objek yang berisi nilai-nilai dalam sistem koordinat yang membangun grafik, sebagai **contoh** : suatu garis dimulai dari titik-awal **(20,20)** dan titik-akhir **(40,40)**.
- ❑ Sistem **koordinat** pada java selalu dimulai dari **titik (0,0)** yang diposisikan pada sudut kiri atas. Nilai x **positif** berada di **kanan** dan nilai **y positif** berada di bawah.

Sistem Koordinat Grafik

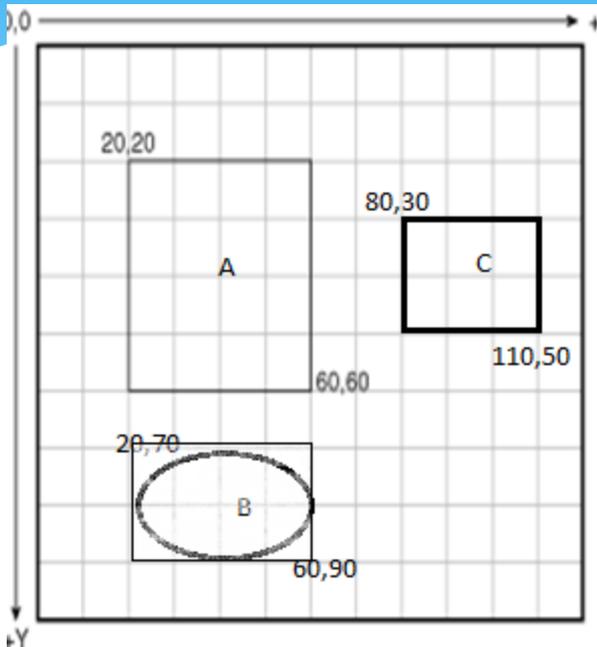
Sistem **koordinat** pada java selalu dimulai dari **titik (0,0)** yang diposisikan pada sudut kiri atas. Nilai x **positif** berada di **kanan** dan nilai y **positif** berada di bawah.



Catatan :

Sistem koordinat pada java berbeda dengan bahasa pemrograman yang lainnya, dimana dimulai dari sudut bawah. Apabila Anda belum terbiasa bekerja dengan posisi yang terbalik ini, maka disarankan agar Anda mulai memperbanyak latihan dengan sistem koordinat pada java.

Pengaturan Koordinat Grafik



Penjelasan :

- ✓ Gambar A adalah bentuk sama sisi di mulai pada titik(20,20) dengan lebar=40 dan tinggi =40 maka titik akhirnya (60,60).
- ✓ Gambar B adalah bentuk Lingkaran di mulai pada titik(20,70) dengan lebar=40 dan tinggi =20 maka titik akhirnya (60,90).
- ✓ Gambar C adalah bentuk Persegi Panjang di mulai pada titik(80,30) dengan lebar=30 dan tinggi =20 maka titik akhirnya (110,50).

Catatan :

Menggambar banyak bentuk dalam satu bidang gambar (satu class) harus menentukan titik koodinat yang tepat agar tidak terjadi tumpukan gambar, untuk memudahkan menentukan koodinat maka merancang output gunakan milimeterblok sebelum implementasi kebahasa programan

WARNA

Warna dalam java, menggunakan standar 24-bit

komponen warna dapat memiliki nilai antara 0 dan 255. Nilai 0,0,0 adalah warna hitam, 255,255,255 adalah warna putih

Model warna abstrak Java dipetakan ke model warna dimana platform Java berjalan, biasanya hanya 256 warna atau kurang yang dapat dipilih

Jika warna yang diminta dalam objek warna tidak tersedia, warna akan dipetakan kedalam warna lain, tergantung bagaimana browser melihat warna tersebut dan platform apa yang digunakan

Pemanfaatan Objek Warna

Untuk menggambar objek dengan warna tertentu, kita harus membuat *instance* dari kelas Color untuk mewakili warna itu

objek warna baru

```
Color c = new Color  
(140, 140, 140);
```

*bilangan float dari 0,0
sampai 1, misalnya :*

```
Color c = new Color  
(0.55, 0.55, 0.55);
```

Tabel 3. Warna-warna standar

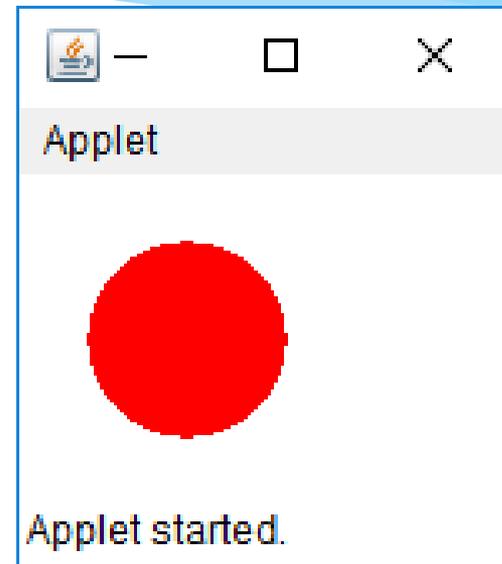
Color Name	RGB Value
Color.white	255, 255, 255
Color.black	0, 0, 0
Color.lightGray	192, 192, 192
Color.gray	128, 128, 128
Color.darkGray	64, 64, 64
Color.red	255, 0, 0
Color.green	0, 255, 0
Color.blue	0, 0, 255
Color.yellow	255, 255, 0
Color.magenta	255, 0, 255
Color.cyan	0, 255, 255
Color.pink	255, 175, 175
Color.orange	255, 200, 0

Color standar warna

Contoh :

```
import java.awt.Graphics;
import java.awt.Color;
public class MyOval extends
java.applet.Applet {
    public void paint(Graphics g) {
        g.setColor(Color.red);
        g.fillOval(20,20,60,60);
    }
}
```

Pengaturan
warna dengan
menggunakan
standar warna



Objek Warna

Jika warna yang ingin digambarkan tidak ada dalam objek Color standar, kita dapat membuat objek warna dengan kombinasi merah, hijau, biru (RGB), asalkan kita memiliki nilai RGB dari warna yang kita inginkan. Kita hanya perlu membuat objek warna baru misalnya

```
Color c = new Color (140,140,140);
```

Baris perintah di bawah ini akan menghasilkan warna abu-abu tua.

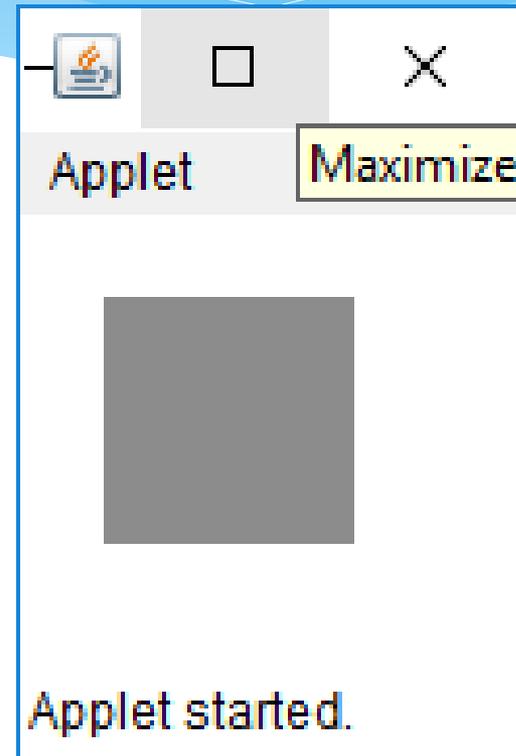
Objek Class Warna

Contoh 2:

```
import java.awt.Graphics;
import java.awt.Color;
public class MyOval extends
java.applet.Applet {
    public void paint(Graphics g) {
        Color c = new Color (140,140,140);

        g.setColor(c);
        g.fillRect(20,20,60,60);
    }
}
```

Pengaturan
warna dengan
menggunakan
warna RGB



Pemeriksaan Dan Pengaturan Warna

Metode `setColor()` (metode dari objek `Graphics`) dapat digunakan untuk keperluan itu.

```
g.setColor(Color.green);
```

warna latar belakang dan latar depan applet dengan menggunakan metode `setBackground()` dan `setForeground()`. Kedua metode ini di definisikan dalam kelas **`java.awt.Component`**,

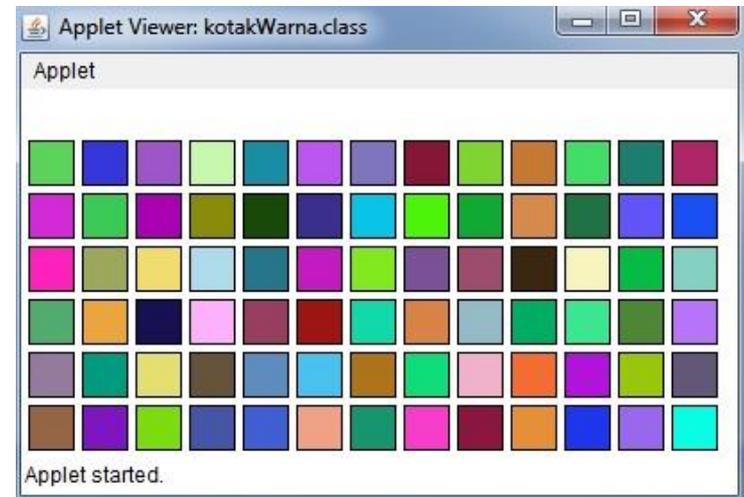
```
setBackground(Color.white)  
setForeground(Color.black);
```

- (1) `getColor()` → didefinisikan dalam objek `Graphics`,
- (2) `getForeground()` → didefinisikan dalam applet,
- (3) `getBackground()` → didefinisikan dalam applet

Contoh Membuat warna Persegi

```
* import java.awt.Graphics;
* import java.awt.Color;
* public class kotakWarna extends java.applet.Applet
* {
*     public void paint(Graphics g)
*     {
*         int rval, gval, bval;
*         for (int j = 30; j < (size().height -25); j += 30)
*         for (int i = 5; i < (size().width -25); i += 30)
*         {
*             rval = (int)Math.floor(Math.random() * 256);
*             gval = (int)Math.floor(Math.random() * 256);
*             bval = (int)Math.floor(Math.random() * 256);
*             g.setColor(new Color(rval,gval,bval));
*             g.fillRect(i, j, 25, 25);
*             g.setColor(Color.black);
*             g.drawRect(i-1, j-1, 25, 25);
*         }
*     }
* }
```

Catatan : dua buah loop for menjadi jantung dari program diatas, yang pertama menggambar baris, dan kedua menggambar kotak-kotak dalam tiap baris. Saat kotak digambar, warna acak dihitung dahulu kemudian kotaknya digambar. Garis rangka hitam digunakan pada tiap-tiap kotak.



Latihan

Soal :

1. Buatlah persegi dengan kodinat (40,40) dan sisi =50 Yang warnakan dengan `Color.green`.
2. Buatlah oval dengan kodinat (40,40) dan sisi =30 Yang warnakan dengan `rgb(62, 254, 255)`.