

HANDS-ON LABS 4 PEMROGRAMAN BERORIENTASI OBJEK + LAB



Oleh:

Martogi Jekson C. Siagian 2105541054

PROGRAM STUDI TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS UDAYANA

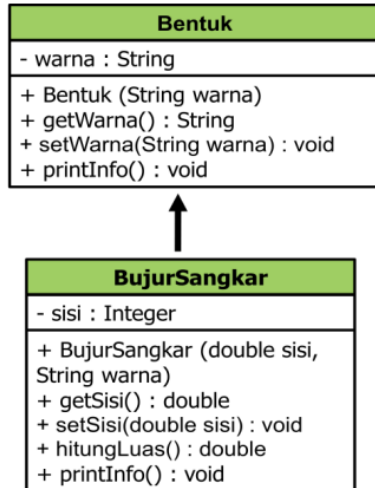
2023

Hands-On Labs 4

Tugas

Buatlah program dalam Java untuk beberapa soal berikut dengan menggunakan prinsip pewarisan.

Soal 1:



Untuk kelas Bentuk :

- **getWarna** adalah method yang akan mengembalikan nilai variabel warna
- **setWarna** adalah method untuk mengubah nilai variabel warna
- **printInfo** adalah method yang akan menuliskan "Bentuk berwarna [warna]"

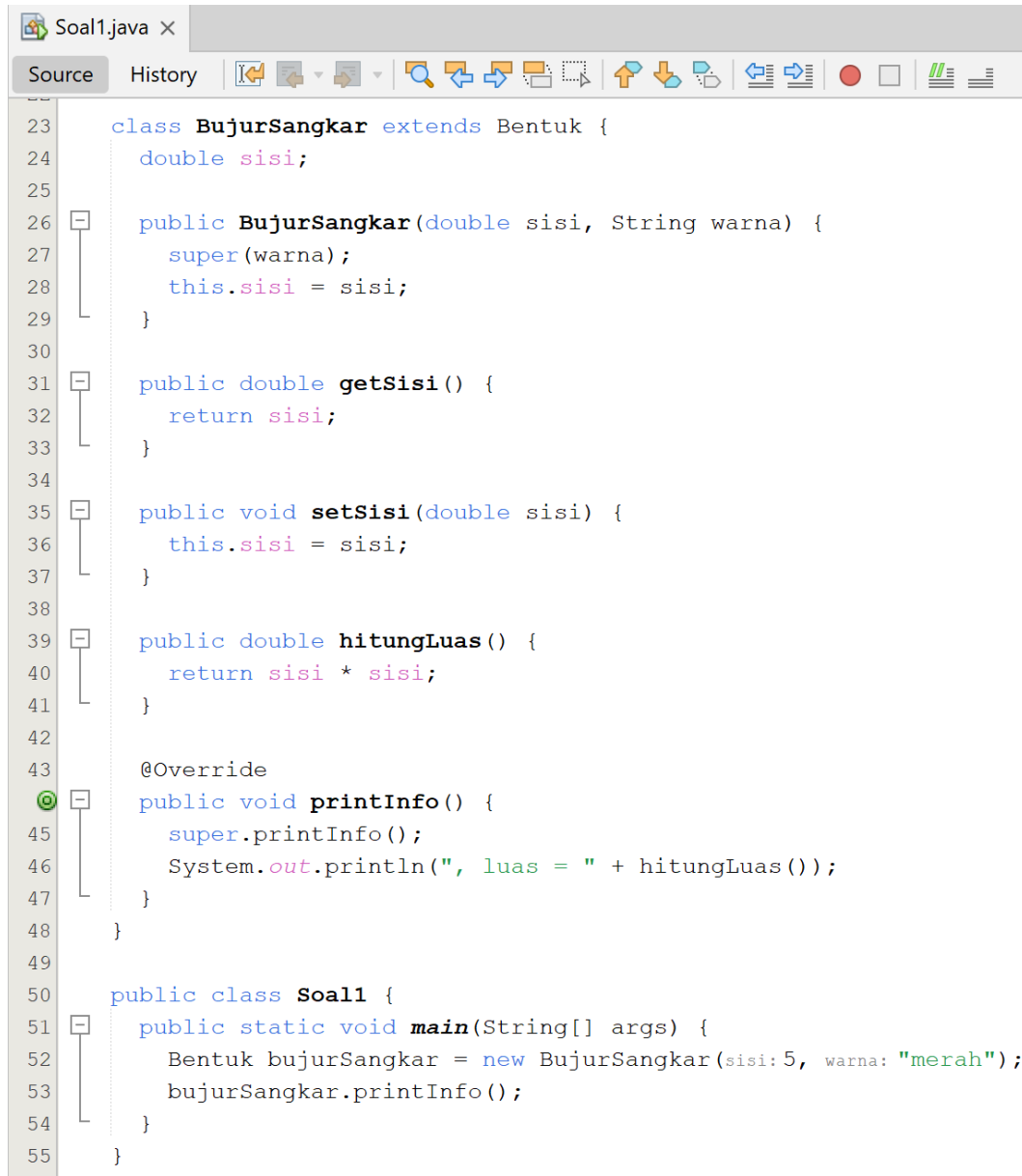
Untuk kelas BujurSangkar :

- **getSisi** adalah method yang akan mengembalikan nilai variabel sisi
- **setSisi** adalah method untuk mengubah nilai variabel sisi
- **hitungLuas** adalah method yang akan mengembalikan hasil perhitungan luas bujursangkar
- **printInfo** adalah method yang akan menuliskan "Bujursangkar berwarna [warna], luas = [luas]"

Jawab :

Berikut Kode yang saya buat

```
Soal1.java X
Source History
1 package soal1;
2
3
4 class Bentuk {
5     String warna;
6
7     public Bentuk(String warna) {
8         this.warna = warna;
9     }
10
11     public String getWarna() {
12         return warna;
13     }
14
15     public void setWarna(String warna) {
16         this.warna = warna;
17     }
18
19     public void printInfo() {
20         System.out.println("Bentuk berwarna " + warna);
21     }
22 }
```



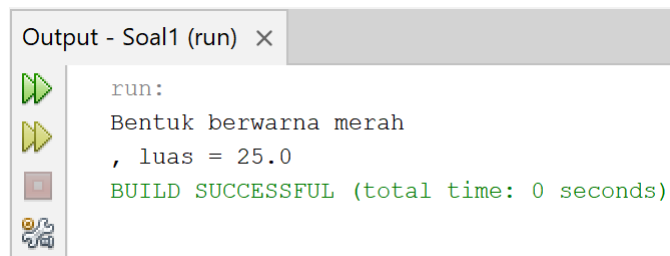
```
23 class BujurSangkar extends Bentuk {
24     double sisi;
25
26     public BujurSangkar(double sisi, String warna) {
27         super(warna);
28         this.sisi = sisi;
29     }
30
31     public double getSisi() {
32         return sisi;
33     }
34
35     public void setSisi(double sisi) {
36         this.sisi = sisi;
37     }
38
39     public double hitungLuas() {
40         return sisi * sisi;
41     }
42
43     @Override
44     public void printInfo() {
45         super.printInfo();
46         System.out.println(", luas = " + hitungLuas());
47     }
48 }
49
50 public class Soal1 {
51     public static void main(String[] args) {
52         Bentuk bujurSangkar = new BujurSangkar(sisi:5, warna: "merah");
53         bujurSangkar.printInfo();
54     }
55 }
```

Berikut Penjelasan mengenai kode yang saya buat: Kode yang saya buat merupakan contoh kode dalam bahasa pemrograman Java yang mendefinisikan beberapa kelas terkait bentuk geometri. Kode ini mencakup dua kelas: Bentuk sebagai kelas induk dan BujurSangkar sebagai kelas anak, serta memiliki sebuah kelas Soal1 yang berisi metode main.

- **Bentuk** adalah kelas dasar yang memiliki dua atribut: warna (String) yang mewakili warna bentuk. Kelas ini memiliki konstruktor Bentuk(String warna) yang digunakan untuk menginisialisasi atribut warna, serta metode-metode getWarna(), setWarna(String warna), dan printInfo(). Metode printInfo() mencetak informasi tentang bentuk, termasuk warnanya.

- **BujurSangkar** adalah subkelas dari **Bentuk** yang mewarisi atribut warna dari kelas **Bentuk** dan menambahkan atribut tambahan: sisi (double) yang mewakili panjang sisi dari bujur sangkar. Kelas ini memiliki konstruktor **BujurSangkar(double sisi, String warna)** yang memanggil konstruktor kelas induk **Bentuk** dengan menggunakan kata kunci **super(warna)** untuk menginisialisasi atribut warna dan menginisialisasi atribut sisi. Selain itu, kelas ini memiliki metode **getSisi()**, **setSisi(double sisi)**, dan **hitungLuas()** yang menghitung luas bujur sangkar berdasarkan panjang sisi. Terdapat juga penggantian (override) metode **printInfo()** yang mencetak informasi tentang bujur sangkar, termasuk warna dan luasnya.
- **Kelas Soal1** adalah kelas utama yang berisi metode **main(String[] args)**. Di dalam metode **main**, Anda membuat objek **bujurSangkar** dengan tipe data **Bentuk**, tetapi sebenarnya objek tersebut adalah instance dari kelas **BujurSangkar** dengan panjang sisi 5 dan warna "merah". Kemudian, Anda memanggil metode **printInfo()** pada objek **bujurSangkar**, yang akan mencetak informasi tentang bujur sangkar ini, termasuk warna dan luasnya.

Berikut Hasil dari program Ketika dijalankan :



```
Output - Soal1 (run) ×
run:
Bentuk berwarna merah
, luas = 25.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

adi, program ini menciptakan sebuah objek bujur sangkar dengan panjang sisi 5 dan warna "merah", dan kemudian mencetak informasi mengenai bentuk tersebut. Hasil cetakan akan mencantumkan warna "merah" dan luas bujur sangkar yang dihitung.

Soal 2:

Lingkaran
- radius
+ constructor (double radius, String warna) + double getRadius() + void setRadius(double r) + double hitungLuas() + void printInfo()

Buatlah kelas Lingkaran sebagai turunan kelas Bentuk.

- **constructor** akan menginisialisasi radius dan warna
- **getRadius** adalah method yang akan mengembalikan nilai variabel radius
- **setRadius** adalah method untuk mengubah nilai variabel radius
- **hitungLuas** adalah method yang akan mengembalikan hasil perhitungan luas lingkaran (PHI jadikan sbg konstanta kelas)
- **printInfo** adalah method yang akan menuliskan "Lingkaran [warna], luas = [luas]"

Jawab:

Berikut Kode yang saya buat

```
Soal2.java x
Source History
class Lingkaran extends Bentuk {
    double radius;
    static final double PHI = 3.14;

    public Lingkaran(double radius, String warna) {
        super(warna);
        this.radius = radius;
    }

    public double getRadius() {
        return radius;
    }

    public void setRadius(double radius) {
        this.radius = radius;
    }

    public double hitungLuas() {
        return PHI * radius * radius;
    }

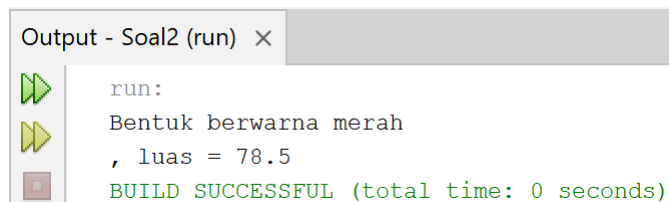
    @Override
    public void printInfo() {
        super.printInfo();
        System.out.println(", luas = " + hitungLuas());
    }
}

public class Soal2 {
    public static void main(String[] args) {
        Lingkaran lingkaran = new Lingkaran(radius: 5, warna: "merah");
        lingkaran.printInfo();
    }
}
```

Berikut Merupakan penjelasan dari kode yang saya buat

1. Lingkaran adalah kelas turunan (subclass) dari kelas Bentuk. Hal ini berarti Lingkaran mewarisi atribut dan method dari Bentuk dan juga dapat menambahkan atribut dan method tambahan yang unik untuk lingkaran:
 - Atribut tambahan radius digunakan untuk menyimpan jari-jari lingkaran.
 - Atribut PHI adalah konstanta yang digunakan untuk menghitung luas lingkaran.
 - Constructor Lingkaran(double radius, String warna) digunakan untuk menginisialisasi atribut radius dan mewarisi atribut warna dari superclass Bentuk.
 - Method getRadius() digunakan untuk mendapatkan nilai atribut radius.
 - Method setRadius(double radius) digunakan untuk mengatur nilai atribut radius.
 - Method hitungLuas() menghitung luas lingkaran menggunakan rumus πr^2 (phi r kuadrat).
 - Method printInfo() meng-override method printInfo() dari superclass Bentuk untuk mencetak informasi tambahan tentang lingkaran, yaitu luasnya.
2. Di dalam metode main, Anda membuat objek lingkaran dari kelas Lingkaran dengan jari-jari 5 dan warna "merah". Kemudian, Anda memanggil lingkaran.printInfo(), yang akan mencetak informasi tentang lingkaran, termasuk warna dan luasnya.

Berikut Hasil dari program Ketika dijalankan



```
Output - Soal2 (run) ×
run:
Bentuk berwarna merah
, luas = 78.5
BUILD SUCCESSFUL (total time: 0 seconds)
```

Soal 3:

Buatlah kelas Silinder sbg turunan kelas Lingkaran

Silinder
- tinggi
+ constructor (double tinggi, double radius, String warna) + double getTinggi() + void setTinggi(double t) + double hitungVolume() + void printInfo()

- **constructor** akan menginisialisasi variabel tinggi, radius, dan warna
- **getTinggi** adl method yg akan mengembalikan tinggi
- **setTinggi** adl method yg akan mengubah tinggi
- **hitungVolume** adl method yg akan mengembalikan hasil perhitungan volum silinder
- **printInfo** adl method yg akan menuliskan "Silinder warna [warna], volume = [volume]"

Jawab:

Berikut Kode yang saya buat

```
Soal3.java x
Source History
class Silinder extends Lingkaran {
    double tinggi;

    Silinder(double tinggi, double radius, String warna) {
        super(radius, warna);
        this.tinggi = tinggi;
    }

    double hitungVolume() {
        return getLuas() * tinggi;
    }

    void printInfo() {
        super.printInfo();
        System.out.println(", volume = " + hitungVolume());
    }
}

public class Soal3 {
    public static void main(String[] args) {
        Silinder silinder = new Silinder(tinggi: 10, radius: 5, warna: "Merah");
        silinder.printInfo();
    }
}
```

Berikut penjelasan kode saya

1. Kelas Silinder (Cylinder):

- Ini adalah kelas turunan (subkelas) dari kelas Lingkaran. Dalam OOP, ini menggambarkan hubungan "is-a" di mana silinder adalah jenis khusus dari lingkaran.
- Mempunyai atribut tambahan, yaitu tinggi (tinggi silinder).

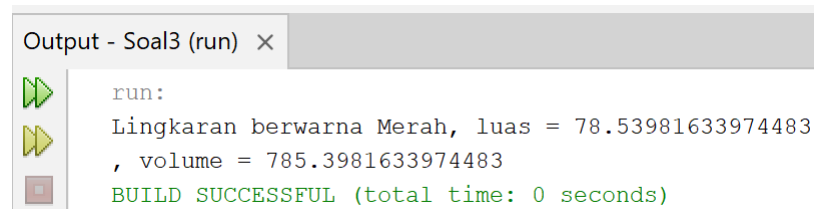
- Terdapat sebuah konstruktor khusus untuk kelas Silinder yang menerima tiga parameter (tinggi, radius, dan warna). Konstruktor ini menggunakan super untuk memanggil konstruktor kelas dasar (Lingkaran) untuk menginisialisasi atribut-atribut kelas Lingkaran dan atribut tambahan kelas Silinder.
- Terdapat metode hitungVolume yang menghitung volume silinder dengan mengalikan luas dasar (diperoleh dari kelas Lingkaran) dengan tinggi.
- Terdapat metode printInfo yang meng-override metode dari kelas Lingkaran dan mencetak informasi tentang silinder, termasuk warna, luas, dan volume.

2. Kelas Soal3 (Main Class):

- Ini adalah kelas utama yang berisi metode main, yang merupakan titik masuk eksekusi program.
- Dalam metode main, sebuah objek silinder dibuat dengan parameter 10 (tinggi), 5 (radius), dan "Merah" (warna).
- Kemudian, metode printInfo dari objek silinder dipanggil, yang mencetak informasi tentang silinder, termasuk warna, luas, dan volume.

Kode ini membantu untuk memahami konsep pewarisan dan penggunaan kelas dalam OOP, di mana kelas Silinder adalah turunan dari kelas Lingkaran, sehingga dapat mengakses atribut dan metode dari kelas dasar, dan menggantinya dengan metode yang lebih spesifik sesuai kebutuhan.

Berikut program Ketika dijalankan



```

Output - Soal3 (run) ×
run:
Lingkaran berwarna Merah, luas = 78.53981633974483
, volume = 785.3981633974483
BUILD SUCCESSFUL (total time: 0 seconds)
  
```