

Nama: Isri Amirul Haq  
NIM: 235150600111011

## **JAWABAN BAB 3 OVERLOAD DAN OVERLOADING METHOD**

Data dan Analisis hasil percobaan

A. Overloading Method

Pertanyaan

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

**Jawab:**

Terdapat kesalahan dalam pemanggilan method HitungLuas. Seharusnya kita memanggil method HitungLuas yang menerima parameter double, bukan yang menerima parameter integer.

2. Jika pada baris 7, pada parameter double value dan double value2 di hapus dan di ganti menjadi int a dan int b apa yang terjadi? Jelaskan!

**Jawab:**

Jika pada baris 7, parameter double value dan double value2 dihapus dan diganti menjadi int a dan int b, maka method tersebut akan menjadi overloading dari method HitungLuas yang sudah ada. Ini karena overloading memungkinkan kita untuk membuat method dengan nama yang sama tetapi berbeda dalam jumlah atau tipe parameter.

Jika kita menghapus parameter double dan menggantinya dengan int, maka kita akan memiliki dua method HitungLuas yang berbeda, satu menerima dua parameter integer dan satu menerima dua parameter double. Dalam hal ini, ketika method HitungLuas dipanggil, Java akan menentukan metode mana yang harus dipanggil berdasarkan tipe parameter yang diberikan.

3. Rubah method pada baris ketujuh menjadi method bertipe void, dan lakukan juga perubahan main method.

**Jawab:**

```

package Tugas4.Bab3;

import java.util.Scanner;

public class Overloading {
    public static void HitungLuas(int a, int b) {
        int nilai = a * b;
        System.out.println("maka hasil luas : " + nilai);
    }

    // public static double HitungLuas(double value, double value2) {
    //     double nilai = value * value2;
    //     return nilai;
    // }

    public static void HitungLuas(double value, double value2) {
        double nilai = value * value2;
        System.out.println("maka hasil luas : " + nilai);
    }

    Run | Debug
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print(s:"masukkan nilai integer 1 : ");
        int integer1 = in.nextInt();
        System.out.print(s:"masukkan nilai integer 2 : ");
        int integer2 = in.nextInt();
        HitungLuas(integer1, integer2);
        System.out.print(s:"masukkan nilai double 1 : ");
        double double1 = in.nextDouble();
        System.out.print(s:"masukkan nilai double 2 : ");
        double double2 = in.nextDouble();
        // HitungLuas(integer1, integer2);
        // System.out.println("Maka hasil luas : " + HitungLuas(double1, double2));
        HitungLuas(double1, double2);
    }
}

```

Dalam hal ini, method HitungLuas dengan parameter double telah diubah menjadi bertipe void dan cetakan hasil luas ditampilkan langsung di dalam method tersebut. Selain itu, pada main method, pemanggilan method HitungLuas juga sudah disesuaikan dengan perubahan tipe method tersebut.

## B. Overloading Constructor

### Pertanyaan

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

### Jawab:

Terdapat kesalahan penulisan nama class. Nama class yang seharusnya digunakan adalah Lingkaran, bukan lingkaran. Oleh karena itu, kita perlu memperbaiki penulisan nama class agar sesuai dengan konvensi penamaan Java.

```
// Before
// public lingkaran(int alas){
//     this.alas = alas;
// }

// public lingkaran(int alas, int tinggi){
//     this.alas = alas;
//     this.tinggi = tinggi;
// }

// Penulisan Class yang benar menggunakan "Lingkaran" bukan lingkaran
public Lingkaran(int alas) {
    this.alas = alas;
}

// no2
public Lingkaran(int alas, int tinggi) {
    this.alas = alas;
    this.tinggi = tinggi;
}

public static void main(String[] args) {
    // Lingkaran bukan "lingkaran"
    Lingkaran l = new Lingkaran(alas:3);
    l.setTinggi(tinggi:10);
    l.displayMessage();
    Lingkaran l2 = new Lingkaran(alas:4, tinggi:10);
    l2.displayMessage();
}
```

2. Pada class lingkaran Tambahkan constructor dengan parameter int tinggi, apa yang terjadi dan jelaskan!

### Jawab:

Untuk menambahkan constructor dengan parameter int tinggi, kita dapat menambahkan constructor baru seperti berikut:

```
// no2
public Lingkaran(int alas, int tinggi) {
    this.alas = alas;
    this.tinggi = tinggi;
}
```

Dengan menambahkan constructor ini, kita dapat menginisialisasi objek Lingkaran dengan memberikan nilai untuk alas dan tinggi sekaligus.

3. Pada class lingkaran tambahkan constrctor dengan tipe data String alas dan String tinggi, kemudian tambahkan method untuk melakukan parsing atau perubahan tipe data dari String menjadi integer. Setelah itu pada method main lakukan instansiasi objek dengan nama objek Lstring dengan memanggil constructor bertipe data String. Jelaskan!

### Jawab:

Untuk menambahkan constructor dengan tipe data String alas dan String tinggi, kita juga perlu menambahkan method untuk melakukan parsing atau perubahan tipe data dari String menjadi integer. Berikut adalah perubahan yang perlu dilakukan pada class Lingkaran:

```
// no 3
public Lingkaran(String alas, String tinggi) {
    this.alas = parseInt(alas);
    this.tinggi = parseInt(tinggi);
}

private int parseInt(String nilai) {
    return Integer.parseInt(nilai);
}
```

Berikut tampilan pemanggilan instansiasi objek Lingkaran dengan memanggil constructor yang menerima parameter bertipe data String dalam main method.

```
public class LingkaranMain {
    Run | Debug
    public static void main(String[] args) {
        // Lingkaran bukan "lingkaran"
        Lingkaran l = new Lingkaran(alas:3);
        l.setTinggi(tinggi:10);
        l.displayMessage();
        Lingkaran l2 = new Lingkaran(alas:4, tinggi:10);
        l2.displayMessage();

        // no 3
        Lingkaran lString = new Lingkaran(alas:"3", tinggi:"10");
        lString.displayMessage();
    }
}
```

Dalam pemanggilan constructor, kita memberikan nilai dalam bentuk String, dan constructor akan melakukan parsing atau perubahan tipe data dari String menjadi integer menggunakan method `parseInt`.