

# PRAKTIKUM 12

## OVERLOADING DAN OVERRIDING

### A. TUJUAN PEMBELAJARAN

1. Memahami tentang overloading
2. Memahami tentang overriding
3. Memahami aturan tentang overridden

### B. DASAR TEORI

**Overloading** adalah suatu keadaan dimana beberapa method sekaligus dapat mempunyai nama yang sama, akan tetapi mempunyai fungsionalitas yang berbeda.

Contoh penggunaan overloading dilihat dibawah ini:

`Gambar(int t1)` → 1 parameter titik, untuk menggambar titik

`Gambar(int t1,int t2)` → 2 parameter titik, untuk menggambar garis

`Gambar(int t1,int t2,int t3)` → 3 parameter titik, untuk menggambar segitiga

`Gambar(int t1,int t2,int t3,int t4)` → 4 parameter titik, untuk menggambar persegi empat

**Overloading** ini dapat terjadi pada class yang sama atau pada suatu parent class dan subclass-nya. Overloading mempunyai ciri-ciri sebagai berikut:

1. **Nama** method harus sama
2. Daftar **parameter** harus berbeda
3. **Return type** boleh sama, juga boleh berbeda

**Overriding** adalah suatu keadaan dimana method pada subclass menolak method pada parent class-nya. Overriding mempunyai ciri-ciri sebagai berikut :


1. **Nama** method harus sama
2. Daftar **parameter** harus sama

### 3. Return type harus sama

Berikut ini contoh terjadinya overriding dimana method Info() pada class Child meng-override method Info() pada class parent:

```
class Parent {  
    public void Info() {  
        System.out.println("Ini class Parent");  
    }  
}  
  
class Child extends Parent {  
    public void Info() {  
        System.out.println("Ini class Child");  
    }  
}
```

child -> [meng-override] -> parrent



Method yang terkena override (**overridden method**) diharuskan tidak boleh mempunyai modifier yang lebih luas aksesnya dari method yang meng-override (**overriding method**).

## C. TUGAS PENDAHULUAN

1. Memahami tentang overloading
2. Memahami tentang overriding
3. Memahami aturan tentang overridden

## D. PERCOBAAN

### Melakukan overloading pada method

Tulislah listing program berikut ini dan amati yang terjadi pada saat terjadinya overloading pada method.

```
import java.awt.Point;  
  
public class Segiempat {  
    int x1 = 0;
```

```

int y1 = 0;
int x2 = 0;
int y2 = 0;

public void buatSegiempat(int x1, int y1, int x2, int y2) {
    this.x1 = x1;
    this.y1 = y1;
    this.x2 = x2;
    this.y2 = y2;
}

public void buatSegiempat(Point topLeft, Point bottomRight) {
    x1 = topLeft.x;
    y1 = topLeft.y;
    x2 = bottomRight.x;
    y2 = bottomRight.y;
}

public void buatSegiempat(Point topLeft, int w, int h) {
    x1 = topLeft.x;
    y1 = topLeft.y;
    x2 = (x1 + w);
    y2 = (y1 + h);
}

void cetakSegiempat(){
    System.out.print("Segiempat: <" + x1 + ", " + y1);
    System.out.println(", " + x2 + ", " + y2 + ">");
}

public static void main(String[] arguments) {
    Segiempat rect = new Segiempat();
    System.out.println("Buat segiempat dengan koordinat (25,25)
                        dan (50,50)");
    rect.buatSegiempat(25, 25, 50, 50);
    rect.cetakSegiempat();
    System.out.println();
    System.out.println("Buat segiempat dengan point (10,10) dan

```

```

        point (20,20):");
    rect.buatSegiempat(new Point(10,10), new Point(20,20));
    rect.cetakSegiempat();
    System.out.println();
    System.out.print("Buat segiempat dengan 1 point (10,10),
    koodinat (50,50)");
    rect.buatSegiempat(new Point(10,10), 50, 50);
    rect.cetakSegiempat();
}
}

```

## E. LATIHAN

### Latihan 1. Overriding

**Apa yang terjadi bila program berikut ini dikompile dan dijalankan? Jelaskan !**

```

class Base{
    private void amethod(int iBase){
        System.out.println("Base.amethod");
    }
}

```

```

class Over extends Base{
    public static void main(String argv[]){
        Over o = new Over();
        int iBase=0;
        o.amethod(iBase);
    }

    public void amethod(int iOver){
        System.out.println("Over.amethod");
    }
}

```

### Latihan 2. Overloading

**Apa yang terjadi bila program berikut ini dikompile dan dijalankan? Jelaskan !**

```

class MyParent {
    int x, y;
    MyParent(int x, int y){
        this.x = x;
        this.y = y;
    }

    public int addMe(int x, int y){
        return this.x + x + y + this.y;
    }

    public int addMe(MyParent myPar){
        return addMe(myPar.x, myPar.y);
    }
}

```

```

class MyChild extends MyParent{
    int z;

    MyChild (int x, int y, int z) {
        super(x,y);
        this.z = z;
    }

    public int addMe(int x, int y, int z){
        return this.x + x + this.y + y + this.z + z;
    }

    public int addMe(MyChild myChi){
        return addMe(myChi.x, myChi.y, myChi.z);
    }

    public int addMe(int x, int y){
        return this.x + x + this.y + y;
    }
}

```

```

public class MySomeone{
    public static void main(String args[]) {
        MyChild myChi = new MyChild(10, 20, 30);
        MyParent myPar = new MyParent(10, 20);
        int x = myChi.addMe(10, 20, 30);
        int y = myChi.addMe(myChi);
        int z = myPar.addMe(myPar);
        System.out.println(x + y + z);
    }
}

```

### Latihan 3. Overloading

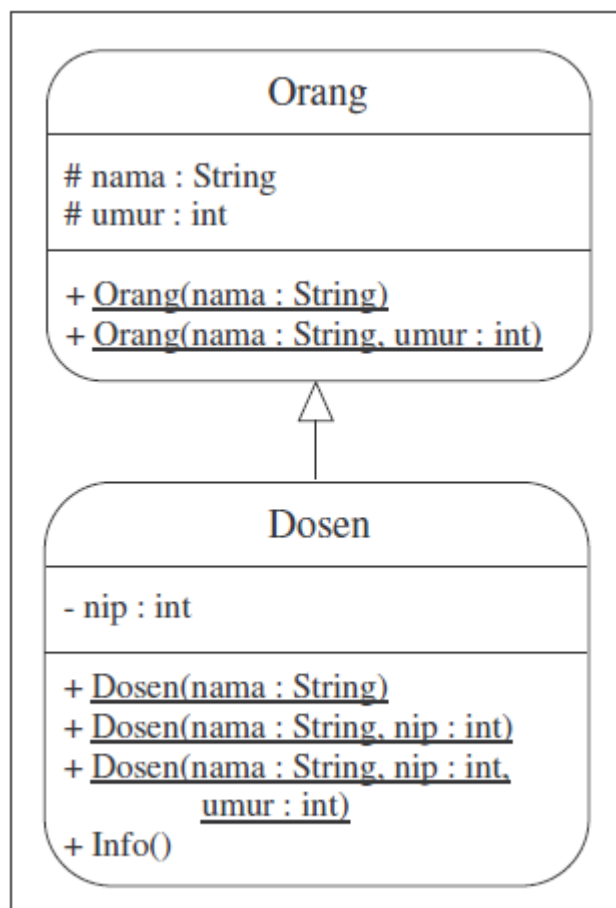
Apa yang terjadi bila program berikut ini dikompile dan dijalankan? Jelaskan !

```
class MyClass{
    void myMethod(int i) {
        System.out.println("int version");
    }

    void myMethod(String s) {
        System.out.println("String version");
    }

    public static void main(String args[]){
        MyClass obj = new MyClass();
        char ch = 'c';
        obj.myMethod(ch);
    }
}
```

### Latihan 4. Mengimplementasikan UML class diagram dalam program



Transformasikan class diagram diatas ke dalam bentuk program? Tulislah listing program berikut ini sebagai pengetesan.

```
public class TesLatihan{
    public static void main(String args[]){
        System.out.println("Memasukkan identitas dosen 1 : Agus");
        Dosen dosen1 = new Dosen("Agus");

        System.out.println("Memasukkan identitas dosen 2 : Budi,
                            NIP. 1458");
        Dosen dosen2 = new Dosen("Budi", 1458);

        System.out.println("Memasukkan identitas dosen 3 : Iwan,
                            NIP. 1215, umur 47");
        Dosen dosen3 = new Dosen("Iwan", 1215, 47);

        System.out.println();
        dosen1.Info();
        System.out.println();
        dosen2.Info();
        System.out.println();
        dosen3.Info();
    }
}
```

Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti diatas.

```
Masukkan identitas dosen 1 : Agus
Masukkan identitas dosen 2 : Budi, NIP. 1458
Masukkan identitas dosen 3 : Iwan, NIP. 1215, umur 47

Nama : Agus
NIP : -
Umur : -

Nama : Budi
NIP : 1458
Umur : -

Nama : Iwan
NIP : 1215
Umur : 47
```

## F. TUGAS

### Tugas 1: Mengimplementasikan UML class diagram dalam program

| RerataNilai                      |
|----------------------------------|
| + int average(int, int)          |
| + double average(double, double) |
| + int average(int, int, int)     |

Transformasikan class diagram diatas ke dalam bentuk program? Tulislah listing program berikut ini sebagai pengetesan

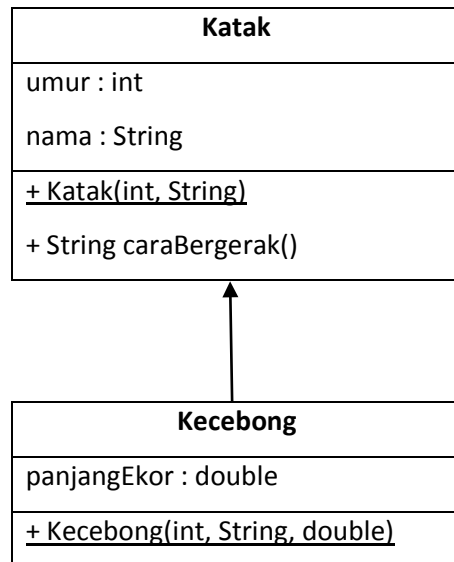
```
public class TestTugas1{
    public static void main(String args[]){
        RerataNilai rn = new RerataNilai();
        System.out.println("Rerata nilai 21 dan 13 adalah : " +
                           rn.average(21, 13));
        System.out.println("Rerata nilai 19.3 dan 39.5 adalah : " +
                           rn.average(19.3, 39.5));
        System.out.println("Rerata nilai 123, 567 dan 744
                           adalah : " + rn.average(123, 567, 744));
    }
}
```

Lakukan kompilasi pada program diatas dan jalankan. Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti diatas.

```
Rerata nilai 21 dan 13 adalah : 27
Rerata nilai 19.3 dan 39.5 adalah : 39.05
Rerata nilai 123, 567 dan 744 adalah : 938
```



## Tugas 2: Mengimplementasikan UML class diagram dalam program



Transformasikan class diagram diatas ke dalam bentuk program? Buatlah TesTugas2.java yang membuat obyek dengan karakteristik sebagaimana Tabel 13.1. Selanjutnya tampilkan informasi nama, umur dan cara bergerak obyek-obyek tersebut!

Tabel 13.1 Karakteristik obyek

| Obyek | umur | nama        | panjangEkor | caraBergerak |
|-------|------|-------------|-------------|--------------|
| O1    | 5    | Froggy      |             | melompat     |
| O2    | 2    | Junior Frog | 10          | berenang     |

## G. LAPORAN RESMI

Kumpulkan hasil latihan dan tugas di atas. Tambahkan analisa dalam laporan resmi.