

# PRAKTIKUM 3

## KONSTRUKTOR DAN OVERLOADING

### A. TUJUAN

1. Konstruktor
2. Overloading terhadap konstruktor
3. Overloading pada metode

### B. DASAR TEORI

#### Deklarasi constructor (konstruktor)

Constructor (konstruktor) adalah suatu method yang pertama kali dijalankan pada saat pembuatan suatu obyek. Konstruktor mempunyai ciri yaitu :

- mempunyai nama yang sama dengan nama class
- tidak mempunyai return type (seperti void, int, double dll)

**Setiap class pasti mempunyai konstruktor.** Jika kita membuat suatu class tanpa menuliskan konstruktornya, maka kompiler dari Java akan menambahkan sebuah konstruktor kosong. Misalnya saja kita mempunyai suatu class Siswa seperti dibawah ini:

```
public class Siswa {  
  
}
```

Disana kita tidak mendeklarasikan konstruktornya secara eksplisit. Ketika proses kompilasi, Kompiler Java akan menambahkan konstruktor kosong sehingga class Siswa tersebut akan tampak sebagai berikut :

```
public class Siswa {  
    public Siswa() {  
    }  
}
```

Karena konstruktor adalah method yang pertama kali dijalankan pada saat suatu obyek dibuat, maka konstruktor sangat berguna untuk **menginisialisasi data member**. Misalnya saja class Siswa diatas mempunyai data member. Kita dapat menginisialisasi nrp di dalam konstruktor yang kita deklarasikan secara ekslipit, seperti yang tampak dibawah ini:

```
public class Siswa {
    private int nrp;
    public Siswa() {
        nrp=0;
    }
}
```

Kita juga dapat menginisialisasi suatu data member dengan nilai yang diinginkan oleh user dengan cara memasukkannya pada parameter konstruktor. Misalnya class Siswa diatas dapat kita modifikasi sebagai berikut :

```
public class Siswa {
    private int nrp;
    public Siswa(int n) {
        nrp=n;
    }
}
```

Dengan mendeklarasikan konstruktor seperti itu, user dapat membuat obyek dengan menginisialisasi nrp sesuai yang ia kehendaki, misalnya saja seperti berikut :

```
Siswa TA1 = new Siswa();
Siswa TA2 = new Siswa(5);
Siswa TA3 = new Siswa(7, "Andi");
```

### **Overloading constructor**

Suatu class dapat mempunyai lebih dari 1 konstruktor dengan syarat daftar parameternya tidak boleh ada yang sama. Misalnya saja kita ingin menginisialisasi data member nrp dengan 2 cara. Pertama, jika user tidak memberikan nilai inisialisasi nrp, maka nrp akan diset dengan nilai 0. Kedua, jika user ingin menginisialisasi nrp sesuai dengan nilai yang diinginkan, maka nrp akan diisi sesuai nilai yang diinginkan oleh user. Sehingga class Siswa diatas dapat kita deklarasikan 2 buah konstruktor seperti yang tampak sebagai berikut :

```
public class Siswa {
    private int nrp;
    private String nama;

    public Siswa() {
        nrp=0;
    }

    public Siswa(int n) {
        nrp=n;
    }

    public Siswa(int n, String s) {
        nrp=n;
        nama = s;
    }
}
```

### **Overloading Terhadap Metode**

Aturan pendeklarasian overloading terhadap metode:

- Nama method harus sama
- Daftar parameter harus berbeda
- Return type boleh sama, juga boleh berbeda

### **Overloading Terhadap Metode**

Perbedaan daftar parameter bukan hanya terjadi pada perbedaan banyaknya parameter, tetapi juga urutan dari parameter tersebut.

Misalnya saja dua buah parameter berikut ini :

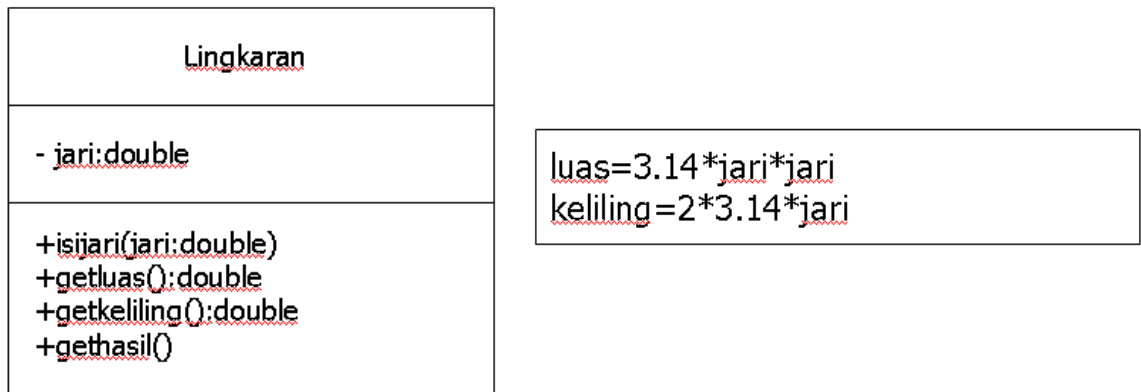
```
function_member(int x, String n)
```

```
function_member(String n, int x)
```

Dua parameter tersebut juga dianggap berbeda daftar parameternya.

### C. TUGAS PENDAHULUAN

1. Mengimplementasikan UML class diagram dalam program untuk class Lingkaran



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

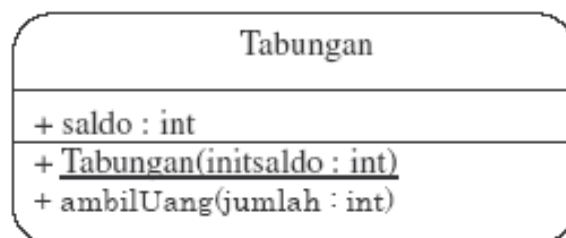
```
public class cobalagi {  
    public static void main(String args[])  
    {  
        Lingkaran a=new Lingkaran();  
        a.isijari(10.0);  
        a.gethasil();  
    }  
}
```

Sehingga didapat hasil:

```
jadi luasnya 314.0 sedangkan kelilingnya 62.800000000000004
```

### C. PERCOBAAN

1. mengimplementasikan UML class diagram dalam program untuk class Tabungan



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

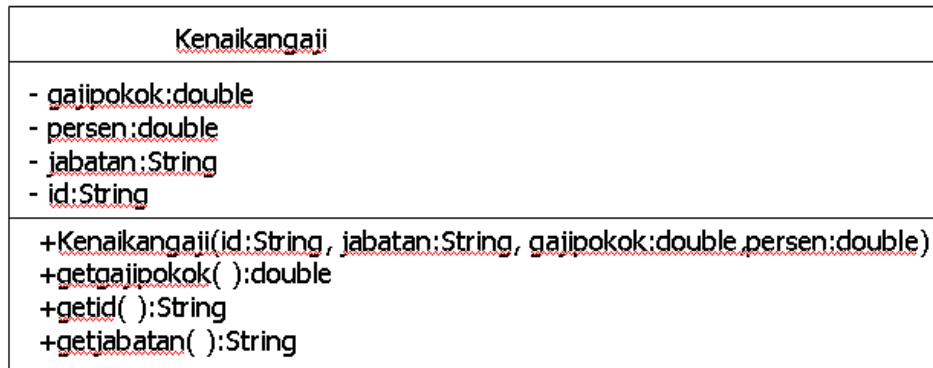
```
public class TesLatihan1 {
    public static void main(String args[]) {
        Tabungan tabungan=new Tabungan(5000);

        System.out.println("Saldo awal : "+tabungan.saldo);
        tabungan.ambilUang(2300);
        System.out.println("Jumlah uang yang diambil : 2300");
        System.out.println("Saldo sekarang : " + tabungan.saldo);
    }
}
```

Sehingga didapat hasil:

```
Saldo awal : 5000
Jumlah uang yang diambil : 2300
Saldo sekarang : 2700
```

2. Mengimplementasikan UML class diagram dalam program untuk class Kenaikangaji



```
gajibaru=gajipokok+(persen/100)*gajipokok
```

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

```

public class cobalagi {
    public static void main(String args[])
    {
        double gajisanti,gajironny,gajiindra;
        String jabatansanti,jabatanronny,jabatanindra,idsanti,idronny,
            idindra;
        Kenaikangaji Santi=new Kenaikangaji("SBY0001","Manager",4000000,5);
        Kenaikangaji Ronny=new Kenaikangaji("BDG0317","Staff",2000000,7.5);
        Kenaikangaji Indra=new Kenaikangaji("JKT0189","Direktur",8000000,3);

        gajisanti=Santi.getgajipokok();
        gajironny=Ronny.getgajipokok();
        gajiindra=Indra.getgajipokok();

        jabatansanti=Santi.getjabatan();
        jabatanronny=Ronny.getjabatan();
        jabatanindra=Indra.getjabatan();

        idsanti=Santi.getid();
        idronny=Ronny.getid();
        idindra=Indra.getid();

        System.out.println("Santi");
        System.out.println("no pegawai: "+idsanti);
        System.out.println("jabatan: "+jabatansanti);
        System.out.println("gaji baru: "+gajisanti);
        System.out.println("");

        System.out.println("Ronny");
        System.out.println("no pegawai: "+idronny);
        System.out.println("jabatan: "+jabatanronny);
        System.out.println("gaji baru: "+gajironny);
        System.out.println("");

        System.out.println("indra");
        System.out.println("no pegawai: "+idindra);
        System.out.println("jabatan: "+jabatanindra);
        System.out.println("gaji baru: "+gajiindra);
        System.out.println("");
    }
}

```

Tampilan yang dihasilkan:

```

Santi
no pegawai: SBY0001
jabatan: Manager
gaji baru: 4200000.0

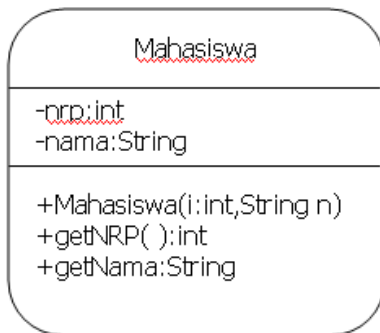
Ronny
no pegawai: BDG0317
jabatan: Staff
gaji baru: 2150000.0

indra
no pegawai: JKT0189
jabatan: Direktur
gaji baru: 8240000.0

```

## E. LAPORAN RESMI

### 1. Mengimplementasikan UML class diagram dalam program untuk class Mahasiswa



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

```
public class TesLatihan2 {
    public static void main(String args[]) {
        Mahasiswa mhs=new Mahasiswa(12345, "Jono");
        System.out.println("NRP : " + mhs.getNrp());
        System.out.println("Nama : " + mhs.getNama());
    }
}
```

Sehingga didapat hasil:

```
NRP : 12345
Nama : Jono
```