**PRAKTIKUM 3**
**KONSTRUKTOR DAN OVERLOADING**

## A. TUJUAN

1. Mahasiswa memahami konsep pengertian dan implementasi konstruktor
2. Mahasiswa memahami konsep overloading terhadap konstruktor
3. Mahasiswa memahami konsep overloading pada metode

## B. DASAR TEORI

### Deklarasi contructor (konstruktor)

Contructor (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 kontruktor 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. Jelaskan perbedaan antara konstruktor dengan metode ?
2. Permasalahan penggunaan konstruktor dan metode

   Mengimplementasikan UML class diagram dalam program untuk class Segitiga

| Segitiga |
| --- |
| - alas:float<br>- tinggi:float<br>- luas:double |
| +setAlas():float<br>+setTinggi():float<br>+getLuas():double<br>+cetakLuasSegitiga() |

$$luas = \frac{1}{2} * alas * tinggi$$

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

```
27  public class HitungSegitiga {
28      public static void main(String args[]){
29          Segitiga sgt = new Segitiga();
30          sgt.setAlas(2);
31          sgt.setTinggi(5);
32          sgt.cetakLuasSegitiga();
33      }
34  }
```

   Tampilan yang diharapkan

```
run:
Luas Segitiga = 5.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. Permasalahan overloading terhadap konstruktor.

   Mengimplementasikan UML class diagram dalam program untuk class Lingkaran

| Lingkaran |
| --- |
| - jari:double<br>- luas:double<br>- keliling:double |
| +Lingkaran()<br>+Lingkaran(jari:double)<br>+setJari(jari:double)<br>+getLuas():double<br>+getKeliling():double<br>+cetak() |

$$luas = 3.14 * jari * jari$$
$$keliling = 2 * 3.14 * jari$$

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

```
33  public class HitungLingkaran {
34      public static void main(String args[]){
35          Lingkaran a = new Lingkaran();
36          a.setJari(10.0);
37          a.cetak();
38          Lingkaran b = new Lingkaran(2.03);
39          b.cetak();
40      }
41  }
```

Tampilan yang diharapkan seperti di bawah ini.

```
run:
Luas = 314.0 dan Keliling = 62.800000000000004
Luas = 12.939625999999997 dan Keliling = 12.748399999999998
BUILD SUCCESSFUL (total time: 0 seconds)
```

Berdasarkan UML class diagram di atas terdapat konstruktor Lingkaran() dan Lingkaran(double jari). Jelaskan perbedaan antara keduanya.

4. Permasalahan overloading terhadap metode.
   Mengimplementasikan UML class diagram dalam program untuk class DataMahasiswa

| DataMahasiswa |
| --- |
| - nama:String<br>- nilai:float |
| +getNama():String<br>+getNilai():float<br>+cetak(nama:String,nilai:float)<br>+cetak(nilai:float,nama:String) |

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

```
26  public class CetakMahasiswa {
27      public static void main(String args[]){
28          DataMahasiswa mhs = new DataMahasiswa();
29          mhs.cetak("Budi Surya Wicaksana", 85);
30          mhs.cetak(90, "Adinda Putri Lestari");
31      }
32  }
```

Tampilan yang diharapkan

```
run:
Pada Praktikum Dasar Pemrograman 2, Budi Surya Wicaksana mendapatkan nilai 85.0
Pada Praktikum Dasar Pemrograman 2, Adinda Putri Lestari mendapatkan nilai 90.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Berdasarkan program di atas terdapat dua metode cetak() dengan parameter berbeda. Jelaskan perbedaan antara keduanya.