

**LAPORAN PRAKTIKUM JOBSHEET 2**  
**CLASS DAN OBJECT**  
**MATA KULIAH PEMROGRAMAN BERBASIS OBJEK**



**Disusun Oleh :**  
**Jami'atul Afifah (2341760102)**  
**SIB-2F**

**PROGRAM STUDI D4 SISTEM INFORMASI BISNIS**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**  
**2024**

## Percobaan 1: Instansiasi objek, mengakses atribut, memanggil method

Langkah kerja:

1. Bukalah development tools, misalkan Netbeans, Visual Code, dsb
2. Ketikkan kode program berikut ini:

```
J Mahasiswa.java > Mahasiswa
1 public class Mahasiswa {
2     public String nim;
3     public String nama;
4     public String alamat;
5     public String kelas;
6
7     public void displayBiodata() {
8         System.out.println("NIM      : " + nim);
9         System.out.println("Nama    : " + nama);
10        System.out.println("Alamat  : " + alamat);
11        System.out.println("Kelas  : " + kelas);
12    }
13 }
```

3. Simpan dengan nama file Mahasiswa.java.
4. Untuk membuat objek baru dengan tipe mahasiswa, dilakukan instansiasi class Mahasiswa seperti contoh berikut:

```
J MahasiswaDemo.java > MahasiswaDemo
1 public class MahasiswaDemo {
2     Run | Debug
3     public static void main(String[] args) {
4         Mahasiswa m1 = new Mahasiswa();
5         m1.nim = "023432";
6         m1.nama = "Yansy Ayuningtyas";
7         m1.alamat = "Nias, Sumatra Utara";
8         m1.kelas = "2A";
9
10        m1.displayBiodata();
11    }
12 }
```

5. Simpan file dengan MahasiswaDemo.java
6. Jalankan class MahasiswaDemo.java

```
PBO_4a8e0648\bin' 'MahasiswaDemo'
NIM      : 023432
Nama     : Yansy Ayuningtyas
Alamat   : Nias, Sumatra Utara
Kelas   : 2A
PS D:\Matkul\SEM 3\PBO> 
```

7. Pada bagian mana proses pendeklarasian atribut pada program di atas?  
Pendeklarasian dilakukan dalam class Mahasiswa

```
public String nim;
public String nama;
public String alamat;
public String kelas;
```

8. Pada bagian mana proses pendeklarasian method pada program di atas?

Pendeklarasian dilakukan dalam class Mahasiswa

```
public void displayBiodata() {  
    System.out.println("NIM      : " + nim);  
    System.out.println("Nama     : " + nama);  
    System.out.println("Alamat  : " + alamat);  
    System.out.println("Kelas  : " + kelas);  
}
```

9. Berapa banyak objek yang di instansiasi pada program di atas?

Hanya satu objek yang diinstansi

```
Mahasiswa m1 = new Mahasiswa();
```

10. Apakah yang sebenarnya dilakukan pada sintaks program “m1.nim=101” ?

Sintaks dalah proses pengisian nilai (assignment) pada atribut nim dari objek m1 dengan nilai "101". Atribut nim yang sebelumnya kosong akan terisi nilai "101".

11. Apakah yang sebenarnya dilakukan pada sintaks program “m1.displayBiodata()” ?

Sintaks memanggil method displayBiodata() dari objek m1. Ketika dipanggil, method ini akan mencetak informasi biodata mahasiswa berupa NIM, Nama, Alamat, dan Kelas yang tersimpan pada objek m1.

12. Instansiasi 2 objek mahasiswa baru pada class MahasiswaDemo

Ubah kode pada MahasiswaDemo.java seperti berikut:

```
1 public class MahasiswaDemo {  
2     Run | Debug  
3     public static void main(String[] args) {  
4         Mahasiswa m1 = new Mahasiswa();  
5         m1.nim = "023432";  
6         m1.nama = "Yansy Ayuningtyas";  
7         m1.alamat = "Nias, Sumatra Utara";  
8         m1.kelas = "2A";  
9  
10        Mahasiswa m2 = new Mahasiswa();  
11        m2.nim = "023433";  
12        m2.nama = "Devita Hwat";  
13        m2.alamat = "Tarik, Darjo";  
14        m2.kelas = "2F";  
15  
16        Mahasiswa m3 = new Mahasiswa();  
17        m3.nim = "023434";  
18        m3.nama = "Farhan Pentol";  
19        m3.alamat = "Sukun, Malang";  
20        m3.kelas = "2F";  
21  
22        m1.displayBiodata();  
23        m2.displayBiodata();  
24        m3.displayBiodata();  
25    }  
26 }
```

Sehingga didapatkan hasil output seperti berikut:

```
NIM      : 023432  
Nama     : Yansy Ayuningtyas  
Alamat   : Nias, Sumatra Utara  
Kelas   : 2A  
  
NIM      : 023433  
Nama     : Devita Hwat  
Alamat   : Tarik, Darjo  
Kelas   : 2F  
  
NIM      : 023434  
Nama     : Farhan Pentol  
Alamat   : Sukun, Malang  
Kelas   : 2F
```

## Percobaan 2: Method dengan return value

Langkah kerja:

1. Bukalah text editor atau IDE, misalnya Notepad ++ / netbeans.
2. Ketikkan kode program berikut ini:

```
public class Barang {  
    public String kode;  
    public String nama;  
    public double hargaKotor;  
    public double diskon;
```

3. Simpan dengan nama file Barang.java
4. Buat method yang menghitung dan mengembalikan nilai harga bersih berdasarkan atribut diskon dan hargaKotor

```
public double getHargaBersih() {  
    return hargaKotor - diskon * hargaKotor;  
}
```

5. Buat method untuk mencetak info dari suatu barang. Nilai harga bersih diperoleh dengan memanggil method getHargaBersih()

```
public void displayInfo() {  
    System.out.println("Kode      : " + kode);  
    System.out.println("Nama      : " + nama);  
    System.out.println("Harga Kotor : " + hargaKotor);  
    System.out.println("Diskon     : " + diskon);  
    System.out.println("Harga Bersih : " + getHargaBersih());  
}
```

6. Buat file baru BarangDemo.java kemudian instansiasi objek barang baru

```
public class BarangDemo {  
    Run | Debug  
    public static void main(String[] args) {  
        Barang barang1 = new Barang();  
        barang1.kode = "ATK01";  
        barang1.nama = "Bolpoin Pilot hitam";  
        barang1.hargaKotor = 3500;  
        barang1.diskon = 0.1;  
  
        barang1.displayInfo();  
    }  
}
```

7. Jalankan program tersebut!

```
Kode      : ATK01  
Nama      : Bolpoin Pilot hitam  
Harga Kotor : 3500.0  
Diskon     : 0.1  
Harga Bersih : 3150.0  
PS D:\Matkul\SEM 3\PBO>
```

8. Ambil kesimpulan tentang kegunaan dari kata kunci return, kapan suatu method harus memiliki kata kunci return?

- Mengembalikan Nilai dari Method:  
return digunakan ketika suatu method memiliki nilai yang ingin dikembalikan setelah melakukan perhitungan atau operasi tertentu. Dalam program di atas, method getHargaBersih() menghitung harga bersih dengan rumus tertentu dan mengembalikan hasil perhitungan tersebut:

```
public double getHargaBersih() {
    return hargaKotor - diskon * hargaKotor;
}
```

Method ini memiliki tipe pengembalian double, sehingga return diperlukan untuk mengembalikan nilai tipe tersebut.

- Kapan Suatu Method Harus Memiliki return:
  - Method Harus Mengembalikan Nilai: Suatu method harus memiliki return jika method tersebut dideklarasikan dengan tipe pengembalian selain void. Misalnya, jika sebuah method dideklarasikan dengan tipe int, double, atau String, maka method tersebut harus mengembalikan nilai dengan return yang sesuai dengan tipe pengembalian yang dideklarasikan.
  - Method dengan Tipe void: Method yang dideklarasikan dengan tipe void tidak memerlukan return karena method ini tidak mengembalikan nilai. Namun, dalam beberapa kasus, return; bisa digunakan untuk menghentikan eksekusi suatu method lebih awal.
- Mengakhiri Eksekusi Method:  
Selain mengembalikan nilai, return juga berfungsi untuk menghentikan eksekusi suatu method, sehingga kode setelah pernyataan return tidak akan dieksekusi.

## TUGAS

1. Implementasikan studi kasus berikut dengan paradigma PBO.

Class PersegiPanjang memiliki atribut panjang dan lebar dengan tipe data integer

Class tersebut juga memiliki tiga method:

- Method displayInfo() untuk menampilkan data panjang dan lebar
- Method getLuas() untuk menghitung luas dari
- Method getKeliling() untuk menghitung keliling

Tampilkan data persegi, nilai luas persegi dan keliling persegi pada class DemoPersegi.

**PersegiPanjang.java:**

```
PersegiPanjang.java > PersegiPanjang > getLuas()
1 public class PersegiPanjang {
2     public int panjang;
3     public int lebar;
4
5     public void displayInfo() {
6         System.out.println("TUGAS AFIFAH");
7         System.out.println("Panjang : " + panjang);
8         System.out.println("Lebar : " + lebar);
9     }
10
11     public int getLuas() {
12         return panjang * lebar;
13     }
14
15     public int getKeliling() {
16         return 2 * (panjang + lebar);
17     }
18 }
```

### DemoPersegi.java:

```
J DemoPersegi.java > DemoPersegi
1 public class DemoPersegi {
2     public static void main(String[] args) {
3         PersegiPanjang persegi = new PersegiPanjang();
4         persegi.panjang = 10;
5         persegi.lebar = 5;
6
7         persegi.displayInfo();
8
9         System.out.println("Luas      : " + persegi.getLuas());
10        System.out.println("Keliling  : " + persegi.getKeliling());
11    }
}
```

### Output:

```
TUGAS AFIFAH
Panjang      : 10
Lebar        : 5
Luas         : 50
Keliling     : 30
PS D:\Matkul\SEM 3\PBO>
```

- Implementasikan salah satu class yang telah dibuat pada tugas Teori PBO 02 ke dalam java dengan paradigma PBO. Instansiasikan 2 objek dari class tersebut pada class lain. Lakukan update nilai atribut dari setiap objek dan eksekusi method yang dimiliki.

FISH
Spesies : String Ukuran : Double Umur : Int Habitat : String
Berenang() : void Makan() : void Berkembangbiak() : Boolean

### Fish.java:

```
J Fish.java > ...
1 public class Fish {
2     public String spesies;
3     public double ukuran;
4     public int umur;
5     public String habitat;
6
7     public void berenang() {
8         System.out.println(spesies + " sedang berenang di habitat " + habitat);
9     }
10
11    public void makan() {
12        System.out.println(spesies + " sedang makan.");
13    }
14
15    public boolean berkembangbiak() {
16        if (umur >= 2) {
17            System.out.println(spesies + " bisa berkembang biak.");
18            return true;
19        } else {
20            System.out.println(spesies + " belum bisa berkembang biak.");
21            return false;
22        }
23    }
24 }
25
```

## FishDemo.java

```
J FishDemo.java
1  public class FishDemo {
    Run | Debug
2      public static void main(String[] args) {
3          Fish ikan1 = new Fish();
4          ikan1.spesies = "Ikan Hiu";
5          ikan1.ukuran = 300.5;
6          ikan1.umur = 5;
7          ikan1.habitat = "Laut";
8
9          Fish ikan2 = new Fish();
10         ikan2.spesies = "Ikan Koi";
11         ikan2.ukuran = 40.2;
12         ikan2.umur = 1;
13         ikan2.habitat = "Kolam";
14
15         System.out.println("Spesies: " + ikan1.spesies);
16         System.out.println("Ukuran : " + ikan1.ukuran + " cm");
17         System.out.println("Umur   : " + ikan1.umur + " tahun");
18         System.out.println("Habitat: " + ikan1.habitat);
19         ikan1.berenang();
20         ikan1.makan();
21         ikan1.berkembangbiak();
22
23         System.out.println("\nUpdate atribut objek ikan2...");
24         ikan2.umur = 2;
25         ikan2.habitat = "Akuarium";
26
27         System.out.println("Spesies: " + ikan2.spesies);
28         System.out.println("Ukuran : " + ikan2.ukuran + " cm");
29         System.out.println("Umur   : " + ikan2.umur + " tahun");
30         System.out.println("Habitat: " + ikan2.habitat);
31         ikan2.berenang();
32         ikan2.makan();
33         ikan2.berkembangbiak();
34     }
35 }
```

## Output:

```
Spesies: Ikan Hiu
Ukuran : 300.5 cm
Umur   : 5 tahun
Habitat: Laut
Ikan Hiu sedang berenang di habitat Laut
Ikan Hiu sedang makan.
Ikan Hiu bisa berkembang biak.

Update atribut objek ikan2...
Spesies: Ikan Koi
Ukuran : 40.2 cm
Umur   : 2 tahun
Habitat: Akuarium
Ikan Koi sedang berenang di habitat Akuarium
Ikan Koi sedang makan.
Ikan Koi bisa berkembang biak.
PS D:\Matkul\SEM 3\PB0>
```