

Jobsheet 02

Class and Object

Name : Fitri cahyaniati(11)

Class : SIB2G

NIM : 2341760198

1. Competence

- Students can understand the descriptions of classes and objects
- Students understand the implementation of the class
- Students can understand the implementation of the attribute
- Students can understand the implementation of the method
- Students can understand the implementation of the intansi process

2. Introduction

2.1 Classes and Objects

In a nutshell, a class is an abstraction of an object (real or unreal) (Roger S Pressman). If we want to create a **student** class, then we need to identify the student object regarding the characteristics/attributes and behaviors/actions that represent the object. One example of an attribute from a student is **NIM** (Student Identification Number) and the behavior/action that can be done by students is **to follow final Exam**.

After we understand the meaning of classes and objects, the next step is to implement classes through the Object Oriented Programming approach (in this course using the java programming language). Here is the syntax of the class declaration in java programming:

```
<modifier> class <nama_class>{  
    //deklarasi atribut dan method  
}
```

The rules for writing a class are as follows:

1. In the form of nouns,
2. Starting with **a capital letter**,
3. If it consists of more than 1 word, then each word is concatenated, and the initial letter of each word uses **a capital letter**.

The Access Modifier is not covered in this jobsheet, but will be discussed in the next jobsheet.

Example class declaration:

```
public class Mahasiswa {  
  
}
```

2.2 Attribute

To declare *attributes*, you can do it with the following syntax:

```
<modifier> <tipe_data> <nama_atribut>;
```

The rules for writing attributes are as follows:

1. In the form of nouns or adjectives,
2. Starting with **a lowercase letter**,
3. If it consists of more than 1 word, then each word is concatenated, and the initial letter of each word uses **a capital letter**

Example attribute declaration:

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

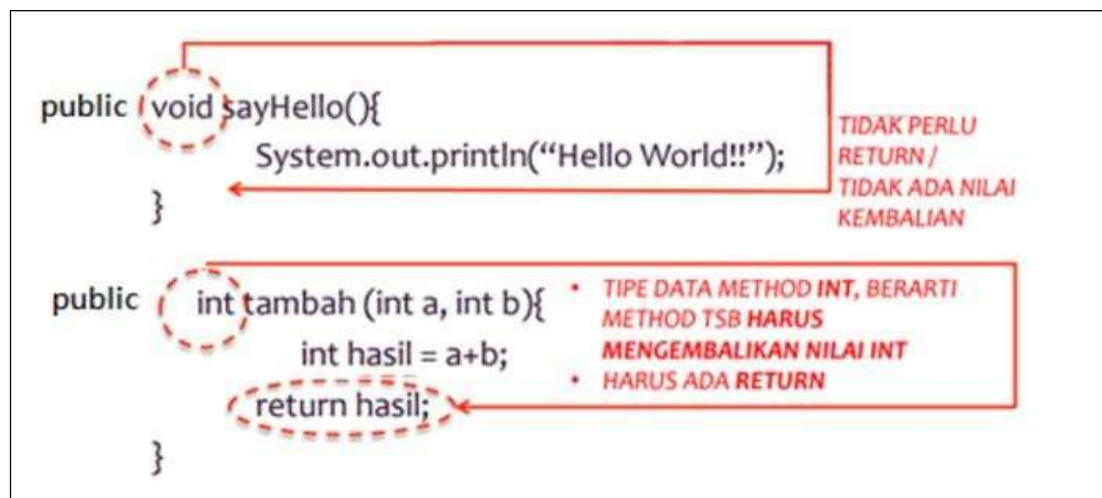
2.3 Method

The method on an object represents the behavior of the object or the action/function/procedure/process that can be performed. A method is implemented as a block that contains a statement or line of program code.

The method is declared with the following syntax:

```
<modifier> <return_type> <nama method>(param1, param2, ...)  
{  
    //statements  
}
```

A method with a *return type* **void** means it does not have a *return value*, so it does not require a *return* keyword in it. While methods with a *return type* **other than void** mean that they require a *return value*, so there must be a value returned with the *return* keyword in it.



The rules for writing methods are as follows:

1. In the form of verbs,
2. Starting with **a lowercase letter**,
3. If it consists of more than 1 word, then each word **is concatenated**, and the initial letter of each word uses **a capital letter**.

2.4 Object

Once the class is created, the next step is to create the object. The process of creating an Object from a Class is called **instantiation** using *the keyword new*. The basic syntax of the agency is as follows:

```
NamaClass namaObject = new NamaClass();
```

Example:

```
Mahasiswa mhs = new Mahasiswa();
Mahasiswa ani = new Mahasiswa();
Mahasiswa mahasiswa = new Mahasiswa();
Random r = new Random();
Pegawai pegawai1 = new Pegawai();
```

In the first line of the example above, a new object is created with the name *Mhs* which is of type *Student*.

3. Experiment

3.1 Experiment 1: Object instantiation, accessing attributes, calling methods

Working steps:

1. Open development tools, e.g. Netbeans, Visual Code, etc.
2. Type the following program code:

```
public class Mahasiswa {
    public String nim;
    public String nama;
    public String alamat;
    public String kelas;

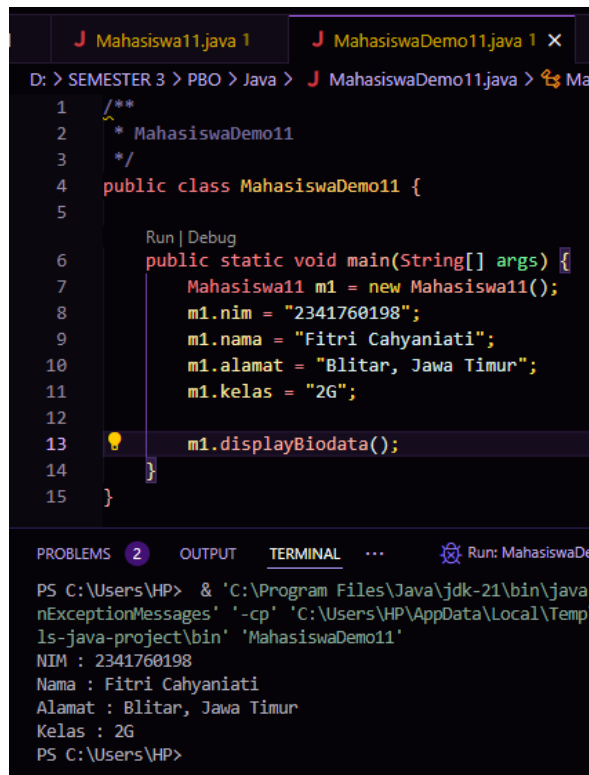
    public void displayBiodata() {
        System.out.println("NIM      : " + nim);
        System.out.println("Nama      : " + nama);
        System.out.println("Alamat   : " + alamat);
        System.out.println("Kelas   : " + kelas);
    }
}
```

3. Save with the file name Mahasiswa.java.
4. To create a new object with a student type, a student class institution is carried out as in the following example:

```
public class MahasiswaDemo {
    public static void main(String[] args) {
        Mahasiswa m1 = new Mahasiswa();
        m1.nim = "023432";
        m1.nama = "Yansy Ayuningtyas";
        m1.alamat = "Nias, Sumatera Utara";
        m1.kelas = "2A";

        m1.displayBiodata();
    }
}
```

5. Save files with MahasiswaDemo.java
6. Run class MahasiswaDemo.java



```

1  /**
2   * MahasiswaDemo11
3   */
4  public class MahasiswaDemo11 {
5
6      public static void main(String[] args) {
7          Mahasiswa11 m1 = new Mahasiswa11();
8          m1.nim = "2341760198";
9          m1.nama = "Fitri Cahyaniati";
10         m1.alamat = "Blitar, Jawa Timur";
11         m1.kelas = "2G";
12
13         m1.displayBiodata();
14     }
15 }
  
```

Run: MahasiswaDemo11

```

PS C:\Users\HP> & 'C:\Program Files\Java\jdk-21\bin\java.
nExceptionMessages' '-cp' 'C:\Users\HP\AppData\Local\Temp\
ls-java-project\bin' 'MahasiswaDemo11'
NIM : 2341760198
Nama : Fitri Cahyaniati
Alamat : Blitar, Jawa Timur
Kelas : 2G
PS C:\Users\HP>
  
```

7. At what point is the attribute declaration process in the above program?

In the attribute section of the student class11

- **public String nim;**
- **public String nama;**
- **public String alamat;**
- **public String kelas;**

8. In what part of the method declaration process in the program above?

The method declaration process occurs in the Mahasiswa11 class. The method declared is displayBiodata().

9. How many objects are instantiated in the above program?

There is only one of the 11th grade students, namely *m1*

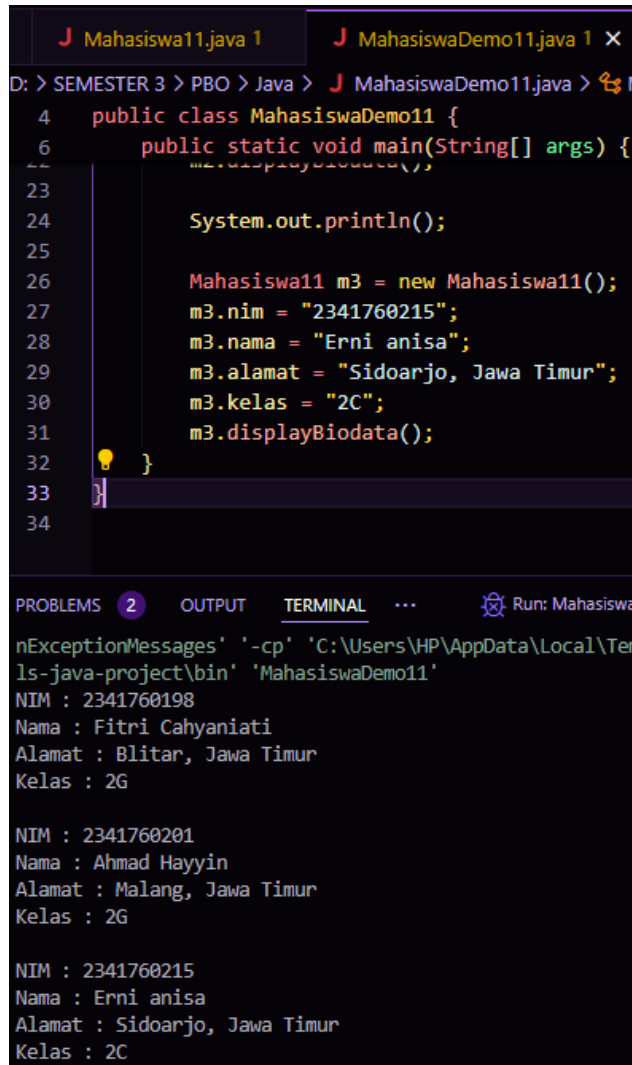
10. What does the "m1.nim=101" program syntax actually do?

This syntax sets the value of the nim attribute on object m1 to "101". After this command is executed, if we access m1.nim, its value becomes "101".

11. What does the "m1.displayBiodata()" program syntax actually do?

This syntax instructs the m1 object to display the data stored in it.

12. Institution of 2 new student objects in the StudentDemo class



The screenshot shows an IDE with two tabs: 'Mahasiswa11.java' and 'MahasiswaDemo11.java'. The 'MahasiswaDemo11.java' tab is active, displaying the following code:

```
4 public class MahasiswaDemo11 {
6     public static void main(String[] args) {
23         // ...
24         System.out.println();
25
26         Mahasiswa11 m3 = new Mahasiswa11();
27         m3.nim = "2341760215";
28         m3.nama = "Erni anisa";
29         m3.alamat = "Sidoarjo, Jawa Timur";
30         m3.kelas = "2C";
31         m3.displayBiodata();
32     }
33 }
34
```

The terminal window at the bottom shows the output of the program:

```
Run: MahasiswaDemo11
ExceptionMessages' '-cp' 'C:\Users\HP\AppData\Local\Temp\ls-java-project\bin' 'MahasiswaDemo11'
NIM : 2341760198
Nama : Fitri Cahyaniati
Alamat : Blitar, Jawa Timur
Kelas : 2G

NIM : 2341760201
Nama : Ahmad Hayyin
Alamat : Malang, Jawa Timur
Kelas : 2G

NIM : 2341760215
Nama : Erni anisa
Alamat : Sidoarjo, Jawa Timur
Kelas : 2C
```

4.2 Experiment 3: Method with return value

Working steps:

1. Open a text editor or IDE, e.g. Notepad++/netbeans.
2. Type the following program code:

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

3. Save with file name Barang.java
4. Create a method that calculates and returns the net price value based on the discount and gross price attributes

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

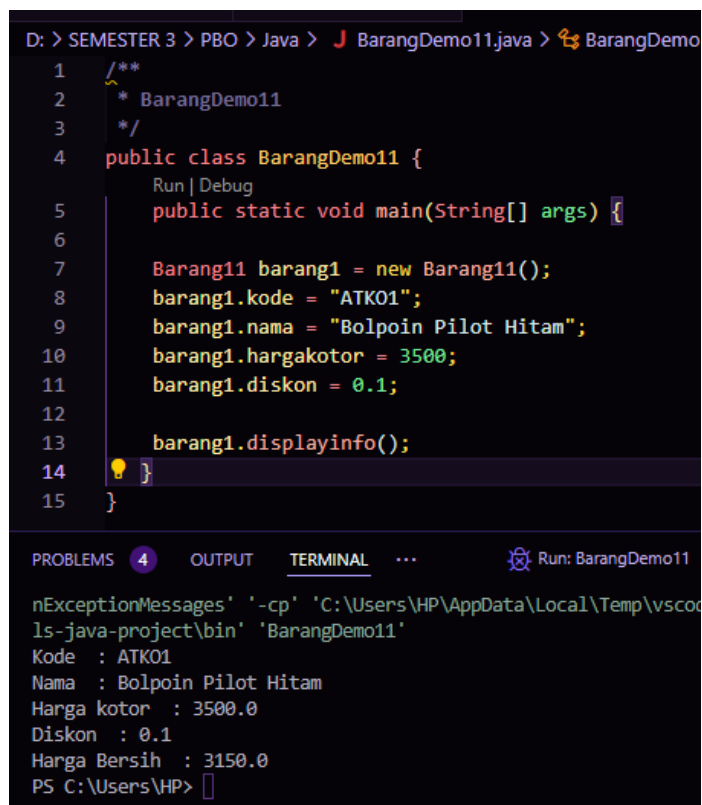
5. Create a method to print info from an item. The net price value is obtained by calling the `getHargaNet()` method.

```
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. Create a new file BarangDemo.java then instantiate the new item object

```
public class BarangDemo {  
    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. Run the program!



```
D: > SEMESTER 3 > PBO > Java > J BarangDemo11.java > BarangDemo11  
1  /**  
2   * BarangDemo11  
3   */  
4  public class BarangDemo11 {  
5      public static void main(String[] args) {  
6  
7          Barang11 barang1 = new Barang11();  
8          barang1.kode = "ATK01";  
9          barang1.nama = "Bolpoin Pilot Hitam";  
10         barang1.hargakotor = 3500;  
11         barang1.diskon = 0.1;  
12  
13         barang1.displayinfo();  
14     }  
15 }
```

PROBLEMS 4 OUTPUT TERMINAL ... Run: BarangDemo11

```
nExceptionMessages' '-cp' 'C:\Users\HP\AppData\Local\Temp\vscod  
ls-java-project\bin' 'BarangDemo11'  
Kode : ATK01  
Nama : Bolpoin Pilot Hitam  
Harga kotor : 3500.0  
Diskon : 0.1  
Harga Bersih : 3150.0  
PS C:\Users\HP>
```

8. Drawing conclusions about the usefulness of the return keyword, when should a method have a *return* keyword?

Function:

Returns the value of the method back to its caller. If the method has a return type, return returns a value of the appropriate type. If the method is void, return can be used to exit the method early.

Its use is:

Methods with return types must use return to return the appropriate value.

The void method is not mandatory, but can be used to stop the method before it is finished.

4.3 Assignment

1. Implement the following case study with the PBO paradigm.

The Rectangle **class** has **long and wide** attributes with the integer data type The class also has three methods:

- method `displayInfo()` to display long and wide data
- Method `getArea()` to calculate the area of
- Method `getCircumference()` to calculate circumference

Display the square data, square area values and square circumference in the **DemoSquare class**.

```
SEMESTER 3 > PBO > Java > J PersegiPanjang11.java > PersegiPanjang11 > PersegiPanjang11.java
4  public class PersegiPanjang11 {
5
6      public int panjang;
7      public int lebar;
8
9      public PersegiPanjang11(int panjang, int lebar) {
10         this.panjang = panjang;
11         this.lebar = lebar;
12     }
13
14     public void displayInfo() {
15         System.out.println("Panjang: " + panjang);
16         System.out.println("Lebar: " + lebar);
17     }
18
19     public int getLuas() {
20         return panjang * lebar;
21     }
22
23     public int getKeliling() {
24         return 2 * (panjang + lebar);
25     }
26 }
```

```
D: > SEMESTER 3 > PBO > Java > DemoPersegi11.java > DemoPersegi11 > main(String[])
1  /**
2   * DemoPersegi11
3   */
4  public class DemoPersegi11 {
5      Run | Debug
6      public static void main(String[] args) {
7          PersegiPanjang11 persegi = new PersegiPanjang11(panjang:15, lebar:15);
8
9          persegi.displayInfo();
10
11         // Menghitung luas dan keliling
12         int luas = persegi.getLuas();
13         System.out.println("Luas: " + luas);
14
15         int keliling = persegi.getKeliling();
16         System.out.println("Keliling: " + keliling);
17     }
18 }
```

```
1\Temp\vscodesws_9bfb1\j
1'
Panjang: 15
Lebar: 15
Luas: 225
Keliling: 60
PS C:\Users\HP>
```

In the Barang11 class we define data and how to calculate the price after discount, then in the BarangDemo11 class we create and organize item data and display the information.

2. Implement **one of** the classes that has been created in the PBO Theory 02 task into java with the PBO paradigm. Instantiate 2 objects from that class on another class. Update the attribute values of each object and execute the methods it has.

```
> SEMESTER 3 > PBO > Java > J BarangDemo11.java > BarangDemo11 > main(String
4 public class BarangDemo11 {
5     public static void main(String[] args) {
6         barang2.kode = "ATK02";
7         barang2.nama = "Buku Binder A5";
8         barang2.hargakotor = 8000;
9         barang2.diskon = 0.05;
10
11        System.out.println(x:"\nInformasi Barang 1:");
12        barang1.displayInfo();
13
14        System.out.println(x:"\nInformasi Barang 2:");
15        barang2.displayInfo();
16
17        // Memperbarui nilai atribut //
18        barang1.hargakotor = 4000;
19        barang1.diskon = 0.2;
20
21        barang2.hargakotor = 1600;
22        barang2.diskon = 0.1;
23
24        System.out.println(x:"\nInformasi Barang 1 setelah update:");
25        barang1.displayInfo();
26
27        System.out.println(x:"\nInformasi Barang 2 setelah update:");
28        barang2.displayInfo();
29    }
30 }
```

```
Informasi Barang 1:
Kode : ATK01
Nama : Bolpoin Pilot Hitam
Harga kotor : 3500.0
Diskon : 0.1
Harga Bersih : 3150.0

Informasi Barang 2:
Kode : ATK02
Nama : Buku Binder A5
Harga kotor : 8000.0
Diskon : 0.05
Harga Bersih : 7600.0

Informasi Barang 1 setelah update:
Kode : ATK01
Nama : Bolpoin Pilot Hitam
Harga kotor : 4000.0
Diskon : 0.2
Harga Bersih : 3200.0

Informasi Barang 2 setelah update:
Kode : ATK01
Nama : Bolpoin Pilot Hitam
Harga kotor : 4000.0
Diskon : 0.2
Harga Bersih : 3200.0
PS C:\Users\HP>
```

In classDemoBarang11 we add two objects to Barang11, namely barang1 and barang2, then Set attribute values and display information for both objects and then Update attributes and display information after the update.

----- **Good Luck**-----