Jobsheet 02

Class andObject

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 intansiasi 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;
```
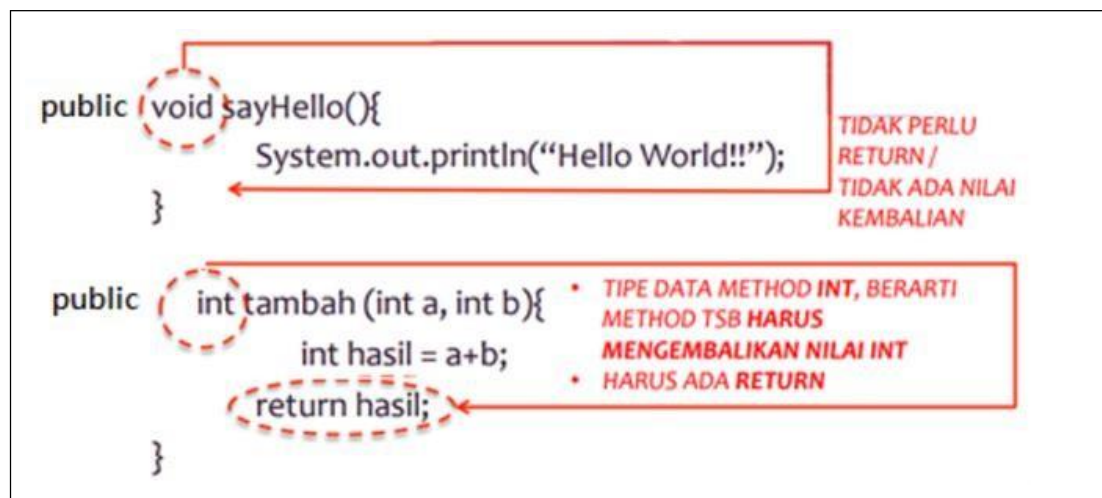
2.3 Method

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:

```java
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:

```java
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

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

```
1   public String nim;
2       public String nama;
3       public String alamat;
4       public String kelas;
```

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

```
1   public void displayBiodata() {
2           System.out.println("NIM     : " + nim);
3           System.out.println("Nama    : " + nama);
4           System.out.println("Alamat : " + alamat);
5           System.out.println("Kelas   : " + kelas);
6       }
```

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

```
1   Mahasiswa16 m1 = new Mahasiswa16();
```
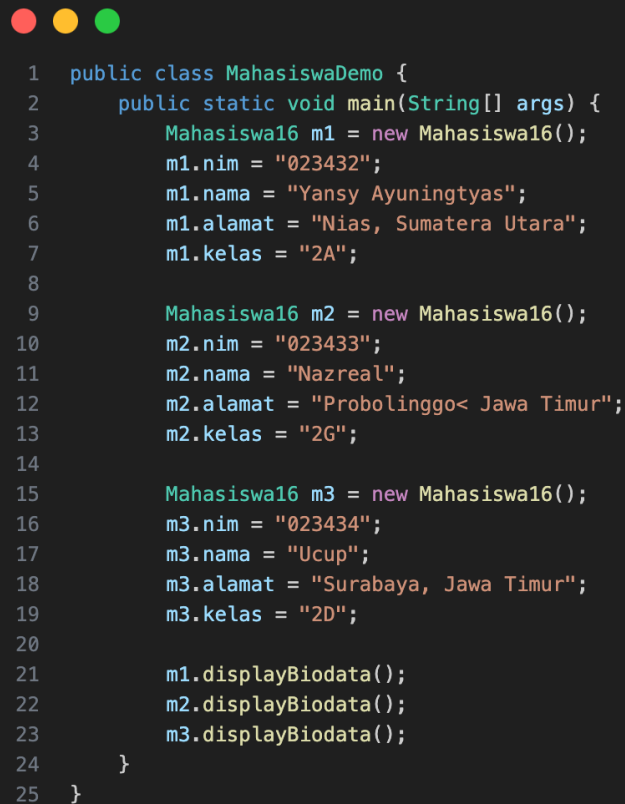
10. What does the "m1.nim=101" program syntax actually do?

- To populate the nim value for object m1 with "101"

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

- calls the displayBiodata() method on object m1. This method executes the command to display the values of the nim, name, address, and class attributes of object m1 to the screen.

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

```
1   public class MahasiswaDemo {
2       public static void main(String[] args) {
3           Mahasiswa16 m1 = new Mahasiswa16();
4           m1.nim = "023432";
5           m1.nama = "Yansy Ayuningtyas";
6           m1.alamat = "Nias, Sumatera Utara";
7           m1.kelas = "2A";
8
9           Mahasiswa16 m2 = new Mahasiswa16();
10          m2.nim = "023433";
11          m2.nama = "Nazreal";
12          m2.alamat = "Probolinggo< Jawa Timur";
13          m2.kelas = "2G";
14
15          Mahasiswa16 m3 = new Mahasiswa16();
16          m3.nim = "023434";
17          m3.nama = "Ucup";
18          m3.alamat = "Surabaya, Jawa Timur";
19          m3.kelas = "2D";
20
21          m1.displayBiodata();
22          m2.displayBiodata();
23          m3.displayBiodata();
24      }
25  }
```

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.

```java
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

```java
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!

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

- The return keyword is used to return the value of a method to its caller. Methods must use return if they have a return type other than void (such as int, double).
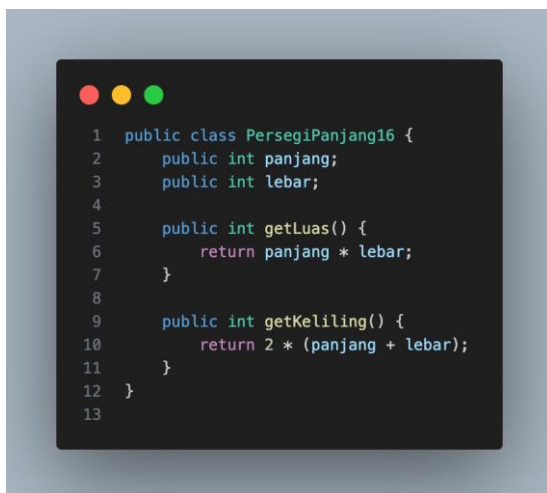
4.3 Assignment

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**.

```java
public class PersegiPanjang16 {
    public int panjang;
    public int lebar;

    public int getLuas() {
        return panjang * lebar;
    }

    public int getKeliling() {
        return 2 * (panjang + lebar);
    }
}
```

```
1   import java.util.Scanner;
2
3   public class Demo16 {
4       public static void main(String[] args) {
5           Scanner input = new Scanner(System.in);
6
7           PersegiPanjang16 persegi1 = new PersegiPanjang16();
8
9           System.out.print("Masukkan panjang: ");
10          persegi1.panjang = input.nextInt();
11          System.out.print("Masukkan lebar: ");
12          persegi1.lebar = input.nextInt();
13
14          System.out.println("");
15
16          System.out.println("Luas Persegi Panjang   : " + persegi1.getLuas());
17          System.out.println("Keliling Persegi Panjang: " + persegi1.getKeliling());
18
19          input.close();
20      }
21  }
22
```

```
Masukkan panjang: 5
Masukkan lebar: 6

Luas Persegi Panjang   : 30
Keliling Persegi Panjang: 22
nazril@Muhammads-MacBook-Air Class And Object %
```

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.

```
1   public class SmartPhone {
2       public String brand;
3       public String model;
4       public int batteryCapacity;
5       public String operatingSystem;
6       public float screenSize;
7
8       public void powerON() {
9           System.out.println(model + " is now ON.");
10      }
11
12      public void powerOFF() {
13          System.out.println(model + " is now OFF.");
14      }
15
16      public void photo() {
17          System.out.println("Taking a photo with " + model);
18      }
19
20      public void calling(String nomor) {
21          System.out.println("Calling " + nomor + " from " + model);
22      }
23
24      public void displayInfo() {
25          System.out.println("Brand: " + brand);
26          System.out.println("Model: " + model);
27          System.out.println("Battery Capacity: " + batteryCapacity + "mAh");
28          System.out.println("Operating System: " + operatingSystem);
29          System.out.println("Screen Size: " + screenSize + " inches");
30      }
31  }
32
```

```java
public class DemoSP {
    public static void main(String[] args) {
        System.out.println("");
        SmartPhone phone1 = new SmartPhone();
        phone1.brand = "Apple";
        phone1.model = "iPhone 13";
        phone1.batteryCapacity = 3240;
        phone1.operatingSystem = "iOS";
        phone1.screenSize = 6.1f;

        SmartPhone phone2 = new SmartPhone();
        phone2.brand = "Samsung";
        phone2.model = "Galaxy S21";
        phone2.batteryCapacity = 4000;
        phone2.operatingSystem = "Android";
        phone2.screenSize = 6.2f;

        System.out.println("Smartphone 1:");
        phone1.displayInfo();
        phone1.powerON();
        phone1.calling("08123456789");
        phone1.photo();
        phone1.powerOFF();

        System.out.println();

        System.out.println("Smartphone 2:");
        phone2.displayInfo();
        phone2.powerON();
        phone2.calling("08567891234");
        phone2.photo();
        phone2.powerOFF();

        System.out.println("");
    }
}
```

```
Smartphone 1:
Brand: Apple
Model: iPhone 13
Battery Capacity: 3240mAh
Operating System: iOS
Screen Size: 6.1 inches
iPhone 13 is now ON.
Calling 08123456789 from iPhone 13
Taking a photo with iPhone 13
iPhone 13 is now OFF.

Smartphone 2:
Brand: Samsung
Model: Galaxy S21
Battery Capacity: 4000mAh
Operating System: Android
Screen Size: 6.2 inches
Galaxy S21 is now ON.
Calling 08567891234 from Galaxy S21
Taking a photo with Galaxy S21
Galaxy S21 is now OFF.

nazril@Muhammads-MacBook-Air Class And Object %
```

----- Good Luck-----