

PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK



Nama : Muhammad Fahmi

Nim : 13020190019

Kelas : A1

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR
2021

A. Modul 1

1. Bagaimana cara membuat lebih dari satu objek class dengan class yang sama ?

Pertama deklarasi dulu objeknya yang harus menggunakan nama class yang sama dengan struktur yang sama yaitu “ NamaClass NamaObjek = new NamaClass”, kemudian pada pemberian nama objeknya harus berbeda.

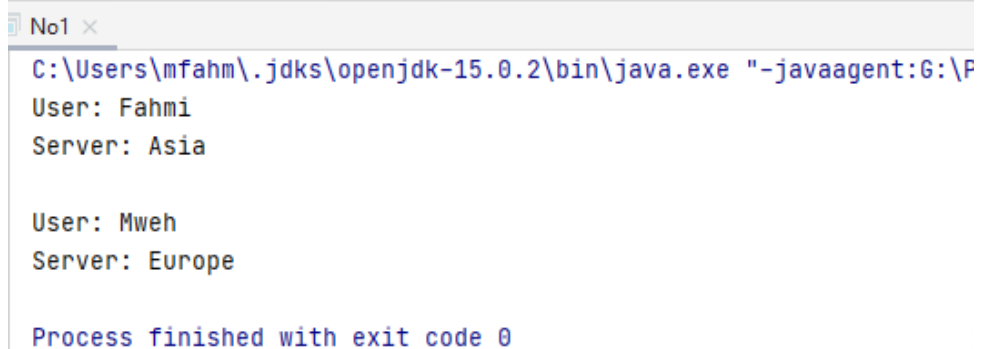
Listing Program:

```
public class No1 {
    String User, Server;

    public static void main(String[] args) {
        No1 a = new No1();
        a.User = "Fahmi";
        a.Server = "Asia";
        System.out.println("User: " + a.User);
        System.out.println("Server: " + a.Server);
        System.out.println("");

        No1 b = new No1();
        b.User = "Mweh";
        b.Server = "Europe";
        System.out.println("User: " + b.User);
        System.out.println("Server: " + b.Server);
    }
}
```

Output pada IntelliJ:



```
No1 x
C:\Users\mfahm\.jdk\openjdk-15.0.2\bin\java.exe "-javaagent:G:\F
User: Fahmi
Server: Asia

User: Mweh
Server: Europe

Process finished with exit code 0
```

2. Apakah bisa membuat lebih dari satu fungsi utama ? Jelaskan alasannya.

Fungsi utama yang biasanya ditulis seperti dibawah ini:

```
public static void main(String[] args) {  
}
```

Adalah java main method. Penyebutan method ini sama saja penyebutan function dalam bahasa C++ atau yang lainnya, keduanya memiliki makna yang sama hanya saja method memiliki objek sedangkan fungsi tidak. Ketika kita memulai program java, program akan mencari method utamanya.

Sebenarnya bisa membuat lebih dari satu fungsi hanya saja bisa terjadi atau disebut istilah overloading dalam java. Seperti dibawah ini:

```
public class no2 {  
    public static void main(String args[]){  
        main(69);  
        main('f');  
        main("Hi Fahmi!");  
    }  
  
    public static void main(int i){  
        System.out.println("Overloaded main() "+i);  
    }  
  
    public static void main(char i){  
        System.out.println("Overloaded main() "+i);  
    }  
  
    public static void main(String str){  
        System.out.println("Overloaded main() "+str);  
    }  
}
```

Output pada IntelliJ:

```
no2 x
C:\Users\mfahm\.jdk\openjdk-15.0.2\bin\java.exe "-ja
Overloaded main()69
Overloaded main()f
Overloaded main()Hi Fahmi!

Process finished with exit code 0
```

3. Buatlah sebuah program sederhana yang menghasilkan output berupa biodata diri sendiri. Sertakan dengan gambar output program.

Listing Program:

```
public class no3 {
    public static void main(String[] args) {
        String name = "Muhammad Fahmi";
        String from = "Makassar";
        String street = "Jln. Toddopuli Raya Timur No.
171C";
        String id = "13020190019";
        int sm = 4;
        int height = 180;
        int weight = 76;

        System.out.println("Name: " + name);
        System.out.println("Place of birth: " + from);
        System.out.println("Adress: " + street);
        System.out.println("ID: " + id);
        System.out.println("Semester: " + sm + "th");
        System.out.println("Height: " + height + " cm");
        System.out.println("Weight: " + weight + " kg");
    }
}
```

Output pada IntelliJ:

```
no3 x
C:\Users\mfahm\.jdk\openjdk-15.0.2\bin\java.exe "-javaagent:G:\Prograi
Name: Muhammad Fahmi
Place of birth: Makassar
Adress: Jln. Toddopuli Raya Timur No. 171C
ID: 13020190019
Semester: 4th
Height: 180 cm
Weight: 76 kg

Process finished with exit code 0
```

4. Buatlah sebuah program dengan mengimplementasikan konstruktor berparameter ! Sertakan dengan gambar output.

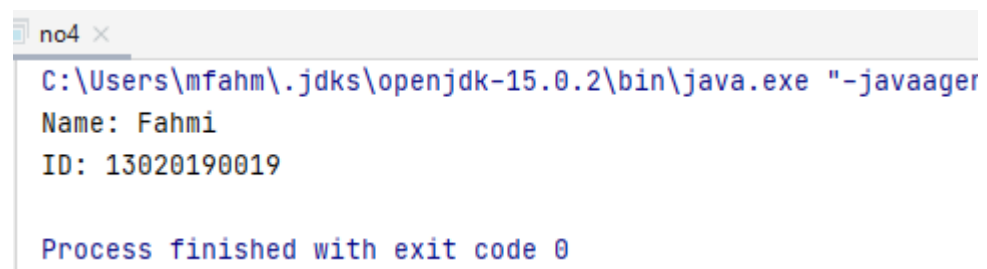
Listing Program:

```
public class no4 {
    String name, id;

    public no4(String inputname, String inputid) {
        name = inputname;
        id = inputid;
    }

    public static void main(String[] args) {
        no4 n = new no4("Name: Fahmi", "ID:
13020190019");
        System.out.println(n.name);
        System.out.println(n.id);
    }
}
```

Output pada IntelliJ:



```
no4 x
C:\Users\mfahm\.jdk\openjdk-15.0.2\bin\java.exe "-javaager
Name: Fahmi
ID: 13020190019

Process finished with exit code 0
```

5. Tuliskan kesimpulan praktikum pada kolom kesimpulan.

Bahasa yang dipelajari pada praktikum ini adalah bahasa Java yang merupakan bahasa berorientasi objek. Dalam Java untuk membuat sebuah program terlebih dahulu buat class lalu didalamnya terdapat Fungsi Utama agar bisa dijalankan. Setelah itu bisa ditambahkan objek, method, maupun konstruktor sesuai program bagaimana yang ingin dibuat.

B. Modul 2

1. Buatlah contoh program yang mengimplementasikan penggunaan variabel local, global dan static !

Listing Program:

```
package Mod2;

public class No1 {
    int oldWeight= 78; // Global Var
    static int addWeight= 10; //Static Var
    public No1() {
        String name = "Fahmi"; //Local Var
        System.out.println(name);
        System.out.println("Old Weight: " + oldWeight + "
kg");
    }
    public static void main(String[] args) {
        No1 n = new No1();
        int oldWeight= 78;
        int newWeight = oldWeight+addWeight;

        System.out.println("New Weight: " + newWeight + "
kg");
    }
}
```

Output pada IntelliJ:

```
No1 x
C:\Users\mfahm\.jdk\openjdk-15.0.2\bin\java.exe "-javaagent:
Fahmi
Old Weight: 78 kg
New Weight: 88 kg

Process finished with exit code 0
```

2. Buatlah contoh program yang mengimplementasikan method procedure dan function!

Listing Program:

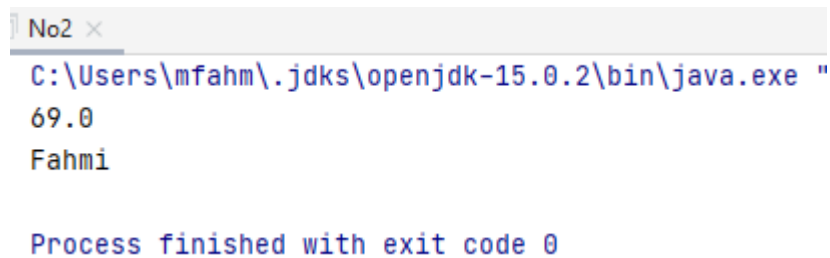
```
package Mod2;

public class No2 {
    public static void test() { //procedure
        float a = 6.9f;
        float b = 10;
        float c = a * b;
        System.out.println(c);
    }

    public static String name() { //function
        String name = "Fahmi";
        return name;
    }

    public static void main(String[] args) {
        test();
        System.out.println(name());
    }
}
```

Output pada IntelliJ:



```
No2 x
C:\Users\mfahm\.jdk\openjdk-15.0.2\bin\java.exe "
69.0
Fahmi

Process finished with exit code 0
```

3. Jelaskan kapan method berparameter dan method nonparameter digunakan

! Berikan contohnya !

Method biasa atau method nonparameter digunakan untuk keperluan tertentu dan akan jalan ketika dipanggil, biasa juga disebut functions. Misalnya method sangat berguna sekali ketika didefinisikan sekali, selanjutnya bisa dipakai banyak kali jadi dapat menyingkat coding maupun waktu.

Method parameter digunakan ketika ingin memberi data atau informasi kedalam method.

Listing Program:

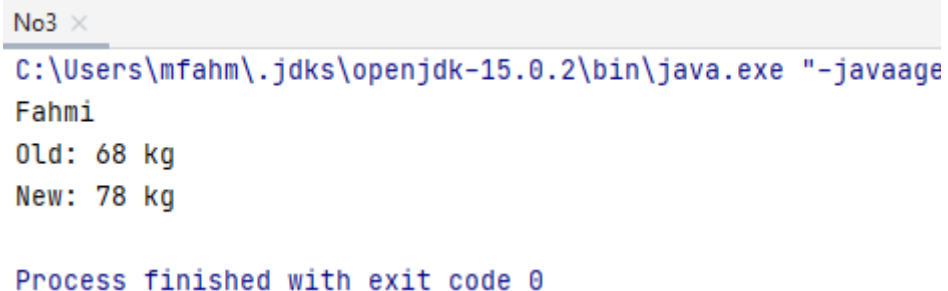
```
package Mod2;

public class No3 {
    static void name() { //nonparameter
        System.out.println("Fahmi");
    }

    static void id(String id) { //parameter
        System.out.println(id + " kg");
    }

    public static void main(String[] args) {
        name();
        id("Old: 68");
        id("New: 78");
    }
}
```

Output pada IntelliJ:



```
No3 x
C:\Users\mfahm\jdk\openjdk-15.0.2\bin\java.exe "-javaage
Fahmi
Old: 68 kg
New: 78 kg

Process finished with exit code 0
```

4. Pada program nomor 2, ubah semua method yang dibuat menjadi method static. Tuliskan pesan error yang terjadi dan jelaskan maksud error tersebut. Tidak error karena awalnya saya menggunakan static sesuai dengan method dan fungsi utama yang keduanya menggunakan static. Akan tetapi, jika didalam fungsi utama static terdapat atribut yang bukan static maka akan terjadi error.

5. Bagaimana cara memanggil atribut atau method yang bersifat static !
Jelaskan !

Seperti apa yang saya tulis pada coding nomor 2 dan 3, saya memanggil atribut dan method yang bersifat static dengan hanya memanggil nama kelas dan nama methodnya, jadi tidak perlu dibuatkan objeknya terlebih dahulu.

6. Tuliskan minimal 6 poin kesimpulan pada modul ini!
- a) Class atau kelas adalah kelompok objek-objek yang memiliki karakteristik yang sama (yang sejenis).
 - b) Setiap class pasti memiliki objek, satu kelas ini dapat terbentuk banyak objek.
 - c) Untuk membuat dan menginisialisasi sebuah object baru diperlukan konstruktor yang merupakan sebuah tipe khusus dari method yang digunakan
 - d) *Method* adalah kumpulan kode perintah untuk melakukan sebuah proses yang diberi nama yang disebut sebagai nama method.
 - e) Procedure merupakan method yang menggunakan void jadi tidak memiliki return atau nilai kembali, tidak seperti pada functions.