

# LAPORAN PRAKTIKUM

## PEMROGRAMAN BERORIENTASI OBJEK

Modul : 3 Tanggal praktikum : 06-04-2023 (Format: DD-MM-YYYY)

Nama : Muhamad Fahmi

NIM : 20220810029

Kelas : TINFC-2022-02 (contoh: TINFC-2022-01)

### JUDUL/TEMA MODUL

#### 1. Tujuan Pembelajaran

Tuliskan tujuan pembelajaran/praktikum. Sesuaikan dengan tujuan pembelajaran pada modul praktikum.

1. Memahami konsep OOP
2. Memahami konsep Objek, Class, Atribut, Method, This, Access Modifier, Instance of Class.

#### 2. Dasar Teori

Pada bagian ini, silahkan kalian tulis/tambahkan materi pendukung untuk kegiatan praktikumnya. Tambahkan gambar / tabel/ flowchart apabila diperlukan.

##### 1. Definisi Objek

Setiap Object (obyek) dibangun dari sekumpulan data (atribut) yang disebut variabel (untuk menjabarkan karakteristik khusus dari obyek) dan juga terdiri dari sekumpulan *method* (menjabarkan tingkah laku dari obyek). **Objek** merupakan sebuah instance dari class. jika class secara umum mempresentasikan template sebuah objek, sebuah instance adalah representasi nyata dari class itu sendiri.

Cara pembuatan objek: menggunakan perintah **new**

```
NamaKelas NamaObjek = new NamaKonstruktor();
```

Atau

```
Nama Kelas NamaObjek;
```

Contoh: dari class Motor dapat dibuat objek honda, yamaha, suzuki, dll

**Method** adalah suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu objek atau dapat dikatakan *method* adalah perilaku dari objek.

Method memiliki 4 (empat) bagian dasar, antara lain:

1. Nama method
2. Tipe Objek atau tipe primitive yang dikembalikan metode
3. Daftar parameter
4. Badan atau isi method

**Konstruktor** merupakan method khusus yang digunakan untuk menginisialisasi objek dan masing-masing class pasti memiliki minimal satu konstruktor. Perbedaan method biasa dengan konstruktor adalah nama method. Konstruktor memiliki nama method sama persis dengan nama classnya, tidak bertipe dan tidak memiliki nilai kembalian. Pemanggilan Konstruktor dapat dilakukan dengan referensi `this()`.

## 2. Method Accessor Dan Method Mutator

**Method Accessor** merupakan method yang digunakan untuk mengambil nilai variabel pada suatu class. Method accessor disebut juga dengan **getter**. Sedangkan **Method Mutator** merupakan method yang digunakan untuk memberikan nilai pada variabel. Method mutator disebut juga dengan **setter**. Kedua metode tersebut menggunakan access modifier `public`. Variabel yang digunakan untuk getter dan setter menggunakan access modifier `private`.

Contoh cara penulisan getter :

```

public class orang {
    private String nama;

    //mendeklarasikan getter dengan nama getName() dan bertipe
    String
    public String getName(){
        //karena non-void maka harus ada nilai kembalian (return
        value)
        return nama;
    }
}

```

Contoh cara penulisan Setter

```

//mendeklarasikan setter dengan nama setName() dan bertipe
Void
//method setName() memiliki 1 parameter "nama" dan bertipe
String
public void setName(String nama){
    this.nama = nama;
}
}

```

### 3. Overloading Method

Overloading Method adalah terdapat method yang sama pada satu class namun berbeda paramater. Berikut ini contoh implementasi overloading method

### 3. Hasil dan Pembahasan

- Bagian ini berisi hasil praktikum yang telah anda kerjakan yang meliputi *pretest*, tahapan praktikum, tugas, dan *posttest*.
- Lampirkan hasil pekerjaan berupa tangkapan layar (*schreenshoot*) kegiatan praktikum sesuai dengan langkah kerja yang ada.

#### PRETEST

1. Buatlah kelas mahasiswa dengan atribut/variable = nama dan nim dan mempunyaimethod: tampilkanNama, tampilkanNim, belajar, olahraga, makan, minum.

Jawab:

```
class Mahasiswa {

    String nama;

    int nim;

    public void tampilkanNama(){

        System.out.println("Tampilkan Nama");

    }

    public int tampilkanNim(){

        System.out.println("Tampilkan Nim");

        return 0;

    }

    public void belajar(){

        System.out.println("Sedang belajar");

    }

    public void olahraga() {

        System.out.println("Sedang olahraga");

    }

    public void makan(){

        System.out.println("Makan");

    }

    public void minum() {

        System.out.println("Minum");

    }

}
```

}

## PRAKTIKUM DAN PEMBAHASAN

### PRAKTIKUM1

1. Buatlah Project Baru di Netbeans dengan nama **Modul3PBO**
2. Buat File Empty Java File Baru
3. Ketikkan source code berikut ini dan simpan dengan nama file **DataDiri.java**

```

1  package DataDiri;
2  class Orang {
3      String nama, gender;
4      int usia;
5      public String tampilNama() {
6          return nama;
7      }
8      public void caraBerjalan() {
9          System.out.println("Tangan disimpan dibelakang");
10     }
11     int usiaSekarang(){
12         return usia;
13     }
14 }
15 public class DataDiri{
16     public static void main(String args[]){
17         Orang org = new Orang();
18         org.nama = "Suci";
19         org.usia = 25;
20         System.out.println("Nama : "+org.tampilNama());
21         System.out.println("Nama : "+org.usiaSekarang());
22         System.out.println("Cara berjalan ");
23         org.caraBerjalan();
24     }
25 }

```

4. Jalankan File **DataDiri.java** tersebut, amati hasilnya!

```

: Output - modul3PBO (run)

run:
Nama : Suci
Nama : 25
Cara berjalan
Tangan disimpan dibelakang
BUILD SUCCESSFUL (total time: 0 seconds)

```

Penjelasan:

Pada praktikum 1 ini belajar praktik tentang pembuatan object dan class dalam satu file di file java **DataDiri.java** sehingga file tersebut bisa langsung dijalankan.

**Class** : cetak biru atau *blueprint* dari **object**. Class digunakan hanya untuk membuat kerangka dasar.

**Object** atau **objek** adalah hasil cetak dari class, atau bisa juga disebut hasil konkrit dari class. **Property** (atau kadang juga dengan atribut atau field) adalah data yang terdapat dalam sebuah class.

**Method** adalah tindakan yang bisa dilakukan di dalam class.

## PRAKTIKUM 2

### Penggunaan Method Setter dan Getter

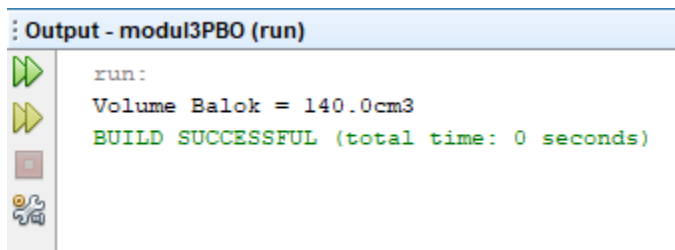
1. Buat File Empty Java File Baru
2. Ketikkan Source Code berikut ini disimpan dengan nama file **Balok.java**

```

1 // Deklarasi class
2 public class Balok {
3 // Deklarasi Atribut
4 double panjang, lebar, tinggi;
5 // Deklarasi setter
6 public void setPanjang(double panjang){this.panjang=panjang;}
7 public void setLebar(double lebar){this.lebar=lebar;}
8 public void setTinggi(double tinggi){this.tinggi=tinggi;}
9 // Deklarasi getter
10 public double getPanjang(){return panjang;}
11 public double getLebar(){return lebar;}
12 public double getTinggi(){return tinggi;}
13 public double getVolume(){return panjang*lebar*tinggi;}
14 public static void main(String args[]){
15     double volume;
16     Balok bl = new Balok();
17     bl.panjang = 7;
18     bl.lebar = 4;
19     bl.tinggi = 5;
20     volume = bl.panjang * bl.lebar * bl.tinggi;
21     System.out.println("Volume Balok = " + volume + "cm3");
22 }
23 }

```

3. Jalankan File **Balok.java** tersebut, Amati hasilnya!



```

Output - modul3PBO (run)
run:
Volume Balok = 140.0cm3
BUILD SUCCESSFUL (total time: 0 seconds)

```

Penjelasan:

Pada praktikum 2 ini program yang dibuat menerapkan system OOP dengan menggunakan method **Getter** dan **Setter** untuk file java balok.java sehingga file tersebut hanya menggunakan 1 class saja. Method Getter dan Setter digunakan untuk mengambil dan mengisi data ke dalam objek.

**Getter:** Method ini memberi penamaan pada method diawali dengan kata **Get** kemudian mengembalikan nilai dengan return sesuai atribut dari yang digunakan. Method Get berfungsi untuk mengambil nilai.

**Setter:** Penamaan pada method ini diawali dengan **Set** yang gunanya untuk mengisi data ke atribut class. Method setter memiliki parameter yang digunakan untuk mengisi nilai tersebut ke atributnya.

### PRAKTIKUM 3

1. Buat File empty java file baru
2. Ketikan SourceCode berikut ini dan simpan dengan nama file **Manusia.java**

```

1 public class Manusia{
2     public String nama;
3     public void manusia(String n){
4         this.nama = n;
5     }
6     public String tampilkanNama(){
7         return nama;
8     }
9     public void makan(){
10        System.out.println("Nyam...Nyam...Nyam...");
11    }
12    public void kerja(){
13        System.out.println("Kerja...kerja...");
14    }
15 }

```

3. Buat File empty java file baru
4. Ketikan SourceCode berikut ini dan simpan dengan nama file **Jaka.java**

```

1 class Jaka{
2     public static void main(String args[]){
3         Manusia andy = new Manusia();
4
5         System.out.println("Nama "+andy.tampilkanNama());
6         andy.makan();
7     }
8 }

```

5. Jalankan file **Jaka.java** Tersebut, amati hasilnya!

```

: Output - modul3PBO (run)

run:
Nama null
Nyam...Nyam...Nyam...
BUILD SUCCESSFUL (total time: 0 seconds)

```



Penjelasan:

Pada praktikum 3 ini membuat program class dan object di file terpisah. Class adalah file **manusia.java** dan objeknya **Jaka.java**. Pada saat menjalankan program dari file Jaka.java terlihat pada nama terdapat null karena pada saat pemanggilan method tampilkan nama tidak memasukan parameter sehingga dianggap null.

## PRAKTIKUM 4:

1. Buat File Empty Java File Baru
2. Ketikan source code berikut ini dan simpan dengan nama file **SPP.java**

```

1  public class SPP {
2      String nama, nim;
3      int spt, spv, sks;
4      //    Konstruktor dengan parameter
5      public SPP(String nama, String nim, int spt, int spv, int sks){
6          this.nama = nama;
7          this.nim = nim;
8          this.spt = spt;
9          this.spv = spv;
10         this.sks = sks;
11     }
12     //    Method hitung dengan tipe int
13     public int hitung(){
14         //    Method non-void harus ada nilai kembalian/return value
15         return (spt+(spv*sks));
16     }
17 }

```

3. Buat File Empty Java File Baru
4. Ketikan source code berikut ini dan simpan dengan nama file **BayarSPP.java**

```

1 import java.io.*;
2 public class BayarSPP{
3     public static void main(String args[]) throws IOException {
4         // Deklarasi variable
5         String nama, nim;
6         int spt, spv, sks;
7         // Proses instansiasi objek dengan inputan manual
8         SPP bayar = new SPP("Irfan", "12.121234", 950000, 70000, 24);
9         // dengan inputan
10        // Langkah 1: Menginisialisasi BuferedReader
11        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
12        // Langkah 2: Proses Input
13        System.out.println("Nama \t\t: "); nama = br.readLine();
14        System.out.println("NIM \t\t: "); nim = br.readLine();
15        System.out.println("SPP Tetap \t: "); spt = Integer.parseInt(br.readLine());
16        System.out.println("SPP Variable \t: "); spv = Integer.parseInt(br.readLine());
17        System.out.println("SKS \t\t: "); sks = Integer.parseInt(br.readLine());
18        // Proses instansiasi objek dengan inputan
19        SPP bayar2 = new SPP(nama, nim, spt, spv, sks);
20        // Output perhitungan dengan inputan manual
21        System.out.println("\n\n\n----- PROGRAM PERHITUNGAN SPP MANUAL ----- \n\n");
22        System.out.println("Nama \t\t: " + bayar.nama);
23        System.out.println("NIM \t\t: " + bayar.nim);
24        System.out.println("SPP Tetap \t: " + bayar.spt);
25        System.out.println("SPP Variable \t: " + bayar.spv);
26        System.out.println("SKS \t\t: " + bayar.sks);
27        System.out.println("Total SPP yang harus dibayarkan : Rp " + bayar.hitung() + ", -\n");
28        System.out.println("----- TERIMA KASIH ----- \n\n");
29        // Output perhitungan dengan inputan
30        System.out.println("\n\n\n----- PROGRAM MENGHITUNG SPP OTOMATIS ----- \n\n");
31        System.out.println("Nama \t\t: " + bayar2.nama);
32        System.out.println("NIM \t\t: " + bayar2.nim);
33        System.out.println("SPP Tetap \t: " + bayar2.spt);
34        System.out.println("SPP Variable \t: " + bayar2.spv);
35        System.out.println("SKS \t\t: " + bayar2.sks);
36        System.out.println("Total SPP yang harus dibayarkan : Rp " + bayar2.hitung() + ", -\n");
37        System.out.println("----- TERIMA KASIH ----- \n\n");
38    }

```

## 5. Jalankan File BayarSPP.java tersebut, amati hasilnya!

```

run:
Nama      :
Rifki
NIM       :
2208
SPP Tetap :
89000
SPP Variable :
12000
SKS       :
23

----- PROGRAM PERHITUNGAN SPP MANUAL -----

Nama      : Irfan
NIM       : 12.121234
SPP Tetap : 950000
SPP Variable : 70000
SKS       : 24
Total SPP yang harus dibayarkan : Rp 2630000, -

----- TERIMA KASIH -----

----- PROGRAM MENGHITUNG SPP OTOMATIS -----

Nama      : Rifki
NIM       : 2208
SPP Tetap : 89000
SPP Variable : 12000
SKS       : 23
Total SPP yang harus dibayarkan : Rp 365000, -

----- TERIMA KASIH -----

```

Penjelasan:

Pada praktikum 5 membuat program Class dan objek dimana file **Spp.java** sebagai file class dan file **BayarSpp.java** sebagai object dari class. Program ini terdapat 2 method yaitu SPP digunakan sebagai method construct dan hitung untuk menghitung pembayaran SPP.

**Construct:** Method khusus yang akan dieksekusi pada saat pembuatan objek (*instance*).

**Java.io:** Library java digunakan untuk melakukan input dan output.

**BufferedReader:** method yang digunakan untuk membaca String dan character.

**InputStream:** Digunakan untuk input Stream pada program atau sistem

## PRAKTIKUM 5

1. Buatlah file empty java file pada project tersebut dengan nama **Mobil.java**
2. Ketikan kode program berikut:

```

1  class Mobil {
2      // Deklarasi
3      private String merk, type, warna;
4      private double harga;
5
6      // Setter
7      public void setMerk(String merk){this.merk = merk;}
8      public void setType(String type){this.type = type;}
9      public void setWarna(String warna){this.warna = warna;}
10     public void setHarga(double harga){this.harga = harga;}
11
12     // Getter
13     public String getMerk(){return merk;}
14     public String getType(){return type;}
15     public String getWarna(){return warna;}
16     public double getHarga(){return harga;}
17
18     // Method tambahan (Cashback)
19     public double Cashback(){
20         double cashback = 0.1 * getHarga();
21         double total = getHarga() - cashback;
22         return total;
23     }
24     public void keterangan(){
25         System.out.println("Harga Mobil setelah dikurangi Cashback = "+Cashback());
26     }
27 }

```

3. Buatlah file empty java File Java pada project tersebut dengan nama **Utama.java**
4. Ketikan kode program berikut:

```

1 import java.io.*;
2
3 class Utama {
4     public static void main(String args[]) throws IOException {
5         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
6         // Instance Class
7         Mobil mb = new Mobil();
8
9         // Input
10        System.out.println("Masukan Merk Mobil");
11        String merk_mobil = br.readLine();
12        System.out.println("Masukan Type Mobil");
13        String type_mobil = br.readLine();
14        System.out.println("Masukan Warna Mobil");
15        String warna_mobil = br.readLine();
16        System.out.println("Masukan Harga Mobil");
17        double harga_mobil = Double.parseDouble(br.readLine());
18        mb.setMerk(merk_mobil);
19        mb.setType(type_mobil);
20        mb.setWarna(warna_mobil);
21        mb.setHarga(harga_mobil);
22
23        // Output
24        System.out.println("Daftar Harga, Merk, Type, dan Harga Mobil");
25        System.out.println("Merk Mobil = "+mb.getMerk());
26        System.out.println("Type Mobil = "+mb.getType());
27        System.out.println("Warna Mobil = "+mb.getWarna());
28        System.out.println("Harga Mobil sebelum di Diskon"+mb.getHarga());
29        mb.keterangan();
30    }
31 }

```

5. Jalankan File **Utama.java** tersebut, amati hasilnya!

```

Output - modul3PBO (run) #2 x
run:
Masukan Merk Mobil
Toyota
Masukan Type Mobil
Fortuner
Masukan Warna Mobil
BlackDove
Masukan Harga Mobil
5000000000
Daftar Harga, Merk, Type, dan Harga Mobil
Merk Mobil = Toyota
Type Mobil = Fortuner
Warna Mobil = BlackDove
Harga Mobil sebelum di Diskon5.0E9
Harga Mobil setelah dikurangi Cashback = 4.5E9
BUILD SUCCESSFUL (total time: 34 seconds)

```

Penjelasan:

Praktikum 5 membuat program OOP dengan menggunakan method Getter dan Setter untuk file java balok.java sehingga file tersebut hanya menggunakan 1 class saja. Method Getter dan Setter digunakan untuk mengambil dan mengisi data ke dalam objek.

## LATIHAN / POST TEST

1. Buatlah Program penerapan class, object, atribut, method seperti praktikum 3 dengan class/objek yang lain.

Jawab:

Source code file player (penerapan class, atribut dan method)

```

3  class player{
4      //ini adalah atribut atau variabel
5      String nama;
6      int kecepatan;
7
8      //ini adalah method atau fungsi
9      void jalan(){
10         System.out.println(nama + " sedang berlari");
11         System.out.println("dengan kecepatan " + kecepatan + " km/jam");
12     }
13
14     public static void main(String[] args){
15         //ini adalah objek baru
16         player orang = new player();
17
18         //ini adalah atribut
19         orang.nama="Fahmi";
20         orang.kecepatan=100;
21
22         //ini adalah method
23         orang.jalan();
24     }
25 }
26

```

```

run:
Fahmi sedang berlari
dengan kecepatan 100 km/jam
BUILD SUCCESSFUL (total time: 0 seconds)

```

### Penjelasan :

player sebagai class dan mendeklarasikan variabel nama menggunakan type data string dan untuk variabel kecepatan menggunakan type data int, dan untuk method void jalan, untuk objek nya orang atributnya nama player dan kecepatan yang di tempuh

## TUGAS INDIVIDU

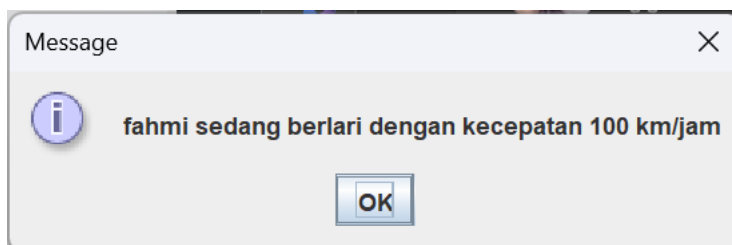
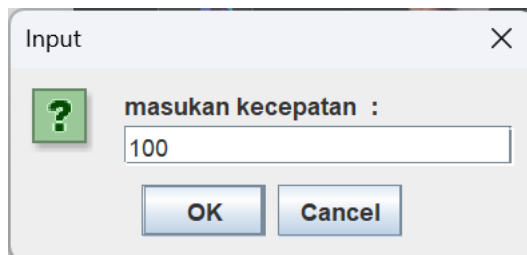
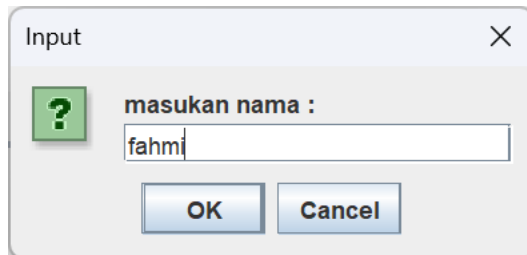
1. Buatlah Program Praktikum 3 dengan inputan data menggunakan JOptionPane.

Jawab:

```

1 import javax.swing.JOptionPane;
2
3 public class player_input
4 {
5
6     private static String nama;
7     public static void main(String[] args)
8     {
9         int kecepatan;
10        String nama;
11        String temp;
12
13        nama = JOptionPane.showInputDialog( "masukan nama :");
14        nama=nama;
15        temp = JOptionPane.showInputDialog( "masukan kecepatan :");
16        kecepatan =Integer.parseInt(temp);
17
18        JOptionPane.showMessageDialog(null,nama + " sedang berlari" + " dengan kecepatan " + kecepatan + " km/jam");
19    }
20 }

```



## 4. Kesimpulan

Buatlah kesimpulan dari pembahasan praktikum yang telah Anda kerjakan.

Dipraktikum pada modul 3 kita mempelajari tentang sebuah program dengan konsep Class dan object baik itu dalam satu file atau dua file terpisah antar class dan object. Kemudian mengenal dan memahami system konsep Getter dan Setter dari OOP(Object Oriented Programing) pada Bahasa Java

**Getter:** Method ini memberi penamaan pada method diawali dengan kata **Get** kemudian mengembalikan nilai dengan return sesuai atribut dari yang digunakan. Method Get berfungsi untuk mengambil nilai.

**Setter:** Penamaan pada method ini diawali dengan **Set** yang gunanya untuk mengisi data ke atribut class. Method setter memiliki parameter yang digunakan untuk mengisi nilai tersebut ke atributnya.

## 5. Referensi

Pada bagian ini cantumkan daftar pustaka/referensi yang ada gunakan dalam menyusun laporan praktikum ini.

Modul pembelajaran

