



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI\_1H

## Laporan Jobsheet 3

### Pemrograman Berorientasi Objek (PBO)

#### Percobaan 1

1. Buat class Motor

```
1 package motorencapsulasi;
2
3 public class Motor {
4     public int kecepatan = 0;
5     public boolean kontakOn = false;
6
7     public void printStatus(){
8         if (kontakOn = true) {
9             System.out.println("Kontak On");
10        }
11        else{
12            System.out.println("Kontak Off");
13        }
14        System.out.println("Kecepatan " + kecepatan + "\n");
15    }
16 }
```

2. Buat class MotorDemo

```
1 package motorencapsulasi;
2
3 public class MotorDemo {
4     Run | Debug
5     public static void main(String[] args) {
6         Motor motor = new Motor();
7         motor.printStatus();
8         motor.kecepatan = 50;
9         motor.printStatus();
10    }
11 }
```

3. Jalankan hasil

```
MotorDemo
Kontak On
Kecepatan 0

Kontak On
Kecepatan 50
```



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI\_1H

## Percobaan 2

### 1. Modifikasi class Motor

```
public void nyalakanMesin(){
    kontakOn = true;
}

public void matikanMesin(){
    kontakOn = false;
    kecepatan = 0;
}

public void tambahKecepatan(){
    if (kontakOn == true) {
        kecepatan += 5;
    }
    else{
        System.out.println(x:"Kecepatan tidak bisa bertambah karena Mesin Off\n");
    }
}

public void kurangiKecepatan(){
    if (kontakOn == true) {
        kecepatan -= 5;
    }
    else{
        System.out.println(x:"Kecepatan tidak bisa berkurang karena Mesin Off\n");
    }
}

public void printStatus(){
    if (kontakOn == true) {
        System.out.println(x:"Kontak On");
    }
    else{
        System.out.println(x:"Kontak Off");
    }
    System.out.println("Kecepatan " + kecepatan + "\n");
}
```

### 2. Modifikasi Class MotorDemo

```
1 package motorencapsulasi;
2
3 public class MotorDemo {
4     Run | Debug
5     public static void main(String[] args) {
6         Motor motor = new Motor();
7         motor.printStatus();
8         motor.tambahKecepatan();
9
10        motor.nyalakanMesin();
11        motor.printStatus();
12
13        motor.tambahKecepatan();
14        motor.printStatus();
15
16        motor.tambahKecepatan();
17        motor.printStatus();
18
19        motor.tambahKecepatan();
20        motor.printStatus();
21
22        motor.matikanMesin();
23        motor.printStatus();
24    }
```



Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI\_1H

### 3. Jalankan Program

```
Kontak Off
Kecepatan 0

Kecepatan tidak bisa bertambah karena Mesin Off

Kontak On
Kecepatan 0

Kontak On
Kecepatan 5

Kontak On
Kecepatan 10

Kontak On
Kecepatan 15

Kontak Off
Kecepatan 0
```

Pertanyaan :

1. Pada class MotorDemo, saat kita menambah kecepatan untuk pertama kalinya, mengapa muncul peringatan “Kecepatan tidak bisa bertambah karena Mesin Off!”?  
  
= karena mesin awal mati yang menghasilkan munculnya peringatan kecepatan tidak bisa bertambah karena mesin of, setelah mesin dihidupkan baru kecepatan bisa bertambah
2. Mengapa atribut kecepatan dan kontakOn diset private?  
  
= karena menerapkan konsep enkapsulasi yaitu menyembunyikan detail internal dari sebuah objek dan hanya memperlihatkan fungsionalitas(Method) yang diperlukan.
3. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

```
public void tambahKecepatan(){
    if (kontakOn == true) {
        if (kecepatan >= 100) {
            System.out.println(x:"Kecepatan sudah maksimal (100)");
        } else{
            kecepatan += 5;
        }
    }
    else{
        System.out.println(x:"Kecepatan tidak bisa bertambah karena
```



Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI\_1H

### Percobaan 3

1. Buat folder KoperasiGetterSetter dan di dalamnya buat file class Anggota

```
1 package KoperasiGetterSetter;
2
3 public class Anggota {
4     private String nama;
5     private String alamat;
6     private float simpanan;
7
8     public void setNama(String nama){
9         this.nama = nama;
10    }
11    public void setAlamat(String alamat){
12        this.alamat = alamat;
13    }
14    public String getNama(){
15        return nama;
16    }
17    public String getAlamat(){
18        return alamat;
19    }
20    public float getSimpanan(){
21        return simpanan;
22    }
23    public void setor(float uang){
24        simpanan += uang;
25    }
26    public void pinjam(float uang){
27        simpanan -= uang;
28    }
29 }
```

2. Buat file class KoperasiDemo

```
1 package KoperasiGetterSetter;
2
3 public class KoperasiDemo {
4     public static void main(String[] args) {
5         Anggota anggota1 = new Anggota();
6         anggota1.setNama(nama:"Iwan Setiawan");
7         anggota1.setAlamat(alamat:"Jalan Sukarno Hatta no 10");
8         anggota1.setor(uang:100000);
9         System.out.println("Simpanan " + anggota1.getNama() + " : Rp " + anggota1.getSimpanan());
10
11         anggota1.pinjam(uang:5000);
12         System.out.println("simpanan " + anggota1.getNama() + " : Rp " + anggota1.getSimpanan());
13     }
14 }
```

3. Jalankan Program

```
Simpanan Iwan Setiawan : Rp 100000.0
simpanan Iwan Setiawan : Rp 95000.0
PS C:\Users\fdlpr\Polinema\Semester 3\
```



Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI\_1H

#### Percobaan 4

1. Modifikasi class KoperasiDemo

```
Anggota anggota1 = new Anggota();  
System.out.println("simpanan " + anggota1.getNama() + " : Rp " + anggota1.getSimpanan());
```

2. Jalankan Kode Program

```
simpanan null : Rp 0.0  
Simpanan Iwan Setiawan : Rp 100000.0  
simpanan Iwan Setiawan : Rp 95000.0
```

3. Modifikasi class Anggota

```
Anggota(String nama, String alamat){  
    this.nama = nama;  
    this.alamat = alamat;  
    this.simpanan = 0;  
}
```

4. Modifikasi class KoperasiDemo

```
Anggota anggota1 = new Anggota(nama:"Iwan", alamat:"Jalan Mawar");  
System.out.println("simpanan " + anggota1.getNama() + " : Rp " + anggota1.getSimpanan());
```

5. Jalankan Kode Program

```
simpanan Iwan : Rp 0.0  
Simpanan Iwan Setiawan : Rp 100000.0  
simpanan Iwan Setiawan : Rp 95000.0  
PS C:\Users\fdlpr\Polinema\Semester 3
```

#### Pertanyaan Percobaan 3 & 4

1. Apa yang dimaksud getter dan setter?
  - **Getter**: Method yang digunakan untuk **mengambil** atau **membaca** nilai dari sebuah atribut. Sesuai konvensi, namanya diawali dengan get. Contoh dari kode Anda adalah getNama(), getAlamat(), dan getSimpanan().
  - **Setter**: Method yang digunakan untuk **mengatur** atau **mengubah** nilai sebuah atribut. Sesuai konvensi, namanya diawali dengan set. Contoh dari kode Anda adalah setName(String nama) dan setAlamat(String alamat).
2. Apa kegunaan dari method getSimpanan()?  
= untuk **mengambil dan menampilkan nilai terbaru** dari atribut simpanan yang bersifat private
3. Method apa yang digunakan untuk menambah saldo?  
= setor(float uang)
4. Apa yang dimaksud konstruktor?  
= sebuah method spesial yang namanya **sama persis** dengan nama class-nya



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI\_1H

5. Sebutkan aturan dalam membuat konstruktor?
  - Nama konstruktor harus **sama persis** dengan nama class.
  - Konstruktor **tidak memiliki tipe data kembalian (return type)**, bahkan tidak juga `void`.
  - Konstruktor bisa memiliki *access modifier* (seperti `public`, `private`) untuk mengatur dari mana saja ia bisa dipanggil.
  - Sebuah class bisa memiliki lebih dari satu konstruktor dengan daftar parameter yang berbeda (disebut *constructor overloading*).
6. Apakah boleh konstruktor bertipe private?  
= **Ya, boleh**. Sebuah konstruktor private berarti objek dari class tersebut tidak bisa dibuat dari luar class itu sendiri
7. Kapan menggunakan parameter dengan passing parameter?  
= Parameter digunakan pada method atau konstruktor ketika method tersebut **membutuhkan data dari luar** untuk dapat menjalankan tugasnya dengan benar. Parameter bertindak sebagai "input" untuk method.
8. Apa perbedaan atribut class dan instansiasi atribut?  
**Method Instance (Method Instansiasi):**
  - Dideklarasikan tanpa keyword static.
  - **Bekerja pada data milik objek (instance) tertentu.**
  - Hanya bisa dipanggil melalui sebuah objek (contoh: `anggota1.getSimpanan()`).
  - Dapat mengakses atribut instance (seperti nama, simpanan) dan juga atribut static.
  - Contoh di kode Anda: Semua method di class Anggota adalah method instance (`getNama`, `sektor`, `pinjam`, dll).**Method Class (Method Static):**
  - Dideklarasikan dengan keyword static.
  - **Tidak terikat pada objek manapun** dan bekerja pada level class.
  - Bisa dipanggil langsung dari nama class-nya tanpa perlu membuat objek (contoh: `Math.random()`).
  - Hanya bisa mengakses atribut static. Ia **tidak bisa** mengakses atribut instance secara langsung karena tidak tahu harus merujuk ke data milik objek yang mana.
  - Contoh di kode Anda: method main di class KoperasiDemo adalah method static.
9. Apa perbedaan class method dan instansiasi method?  
**Method Instance (Method Instansiasi):**
  - Dideklarasikan tanpa keyword static.
  - **Bekerja pada data milik objek (instance) tertentu.**
  - Hanya bisa dipanggil melalui sebuah objek (contoh: `anggota1.getSimpanan()`).
  - Dapat mengakses atribut instance (seperti nama, simpanan) dan juga atribut static.
  - Contoh di kode Anda: Semua method di class Anggota adalah method instance (`getNama`, `sektor`, `pinjam`, dll).**Method Class (Method Static):**
  - Dideklarasikan dengan keyword static.
  - **Tidak terikat pada objek manapun** dan bekerja pada level class.



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI\_1H

- Bisa dipanggil langsung dari nama class-nya tanpa perlu membuat objek (contoh: Math.random()).
- Hanya bisa mengakses atribut static. Ia **tidak bisa** mengakses atribut instance secara langsung karena tidak tahu harus merujuk ke data milik objek yang mana.
- Contoh di kode Anda: method main di class KoperasiDemo adalah method static.

## Tugas

1. Tulis kode program dan lihat hasilnya

```
1 package Tugas;
2
3 public class EncapDemo {
4     private String name;
5     private int age;
6
7     public String getName(){
8         return name;
9     }
10
11    public int getAge(){
12        return age;
13    }
14
15    public void setName(String newName){
16        name = newName;
17    }
18
19    public void setAge(int newAge){
20        if (newAge > 30) {
21            age = 30;
22        } else {
23            age = newAge;
24        }
25    }
26 }
27
```

```
Name : James
Age : 30
PS C:\Users\fdlpr\Polin
```

```
1 package Tugas;
2
3 public class EncapTest {
4     public static void main(String[] args) {
5         EncapDemo encap = new EncapDemo();
6         encap.setName(newName:"James");
7         encap.setAge(newAge:35);
8
9         System.out.println("Name : " + encap.getName());
10        System.out.println("Age : " + encap.getAge());
11    }
12 }
```

2. Pada class EncapTest, saat kita memanggil encap.setAge(35), method setAge di class EncapDemo akan dieksekusi. Di dalam method tersebut, terdapat sebuah kondisi if yang memeriksa apakah nilai masukan lebih besar dari 30. Karena 35 memenuhi kondisi tersebut, program akan menjalankan perintah untuk menetapkan nilai variabel age menjadi 30, sehingga nilai itulah yang ditampilkan



Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI\_1H

3. Ubah program diatas agar atribut age dapat diberi nilai maksimal 30 dan minimal 18.

```
public void setAge(int newAge){  
    if (newAge > 30) {  
        age = 30;  
        System.out.println(x:"Max Age = 30");  
    } else if (newAge < 18) {  
        age = 18;  
        System.out.println(x:"Min Age = 18");  
    }  
    else {  
        age = newAge;  
    }  
}
```

4. Buat class objeknya

```
public class Anggota {  
    private String noKtp;  
    private String nama;  
    private double limitPinjaman;  
    private double jumlahPinjaman;  
  
    public Anggota(String noKtp, String nama, double limitPinjaman) {  
        this.noKtp = noKtp;  
        this.nama = nama;  
        this.limitPinjaman = limitPinjaman;  
        this.jumlahPinjaman = 0;  
    }  
  
    public String getNama() {  
        return nama;  
    }  
  
    public double getLimitPinjaman() {  
        return limitPinjaman;  
    }  
  
    public double getJumlahPinjaman() {  
        return jumlahPinjaman;  
    }  
  
    public void pinjam(double uang) {  
        if (uang + jumlahPinjaman > limitPinjaman) {  
            System.out.println("Maaf, jumlah pinjaman melebihi limit.");  
        } else {  
            jumlahPinjaman += uang;  
        }  
    }  
}
```





Name : Faatihurrizki Prasojo

NIM : 244107020142

Class : TI\_1H

```
public void angsur(double uang) {  
    if (uang < jumlahPinjaman * 0.1) {  
        System.out.println("Maaf, angsuran harus 10% dari jumlah pinjaman");  
    } else if (uang > jumlahPinjaman) {  
        System.out.println("Maaf, pembayaran Anda melebihi total pinjaman saat ini.");  
    } else {  
        jumlahPinjaman -= uang;  
    }  
}
```

```
Nama Anggota: Donny  
Limit Pinjaman: 5000000.0  
  
Meminjam uang 10.000.000...  
Maaf, jumlah pinjaman melebihi limit.  
Jumlah pinjaman saat ini: 0.0  
  
Meminjam uang 4.000.000...  
Jumlah pinjaman saat ini: 4000000.0  
  
Membayar angsuran 1.000.000  
Jumlah pinjaman saat ini: 3000000.0  
  
Membayar angsuran 3.000.000  
Jumlah pinjaman saat ini: 0.0  
PS C:\Users\fdlpr\Polinema\Semester 3\PB0\
```

5. Modifikasi soal no. 4 agar nominal yang dapat diangsur minimal adalah 10% dari jumlah pinjaman saat ini. Jika mengangsur kurang dari itu, maka muncul peringatan “Maaf, angsuran harus 10% dari jumlah pinjaman”.

```
public void angsur(double uang) {  
    if (uang < jumlahPinjaman * 0.1) {  
        System.out.println(x:"Maaf, angsuran harus 10% dari jumlah pinjaman");  
    } else if (uang > jumlahPinjaman) {  
        System.out.println(x:"Maaf, pembayaran Anda melebihi total pinjaman saat ini.");  
    } else {  
        jumlahPinjaman -= uang;  
    }  
}
```



Name : Faatihurrizki Prasajo

NIM : 244107020142

Class : TI\_1H

6. Modifikasi class TestKoperasi, agar jumlah pinjaman dan angsuran dapat menerima input dari console.

```
1 package Tugas;
2 import java.util.Scanner;
3
4 public class TestKoperasi {
5     Run | Debug
6     public static void main(String[] args) {
7         Scanner prasojo = new Scanner(System.in);
8
9         Anggota donny = new Anggota(nokTkp:"111333444", nama:"Donny", limitPinjaman:5000000);
10
11         System.out.println("Nama Anggota: " + donny.getNama());
12         System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());
13
14         System.out.print(s:"\nMasukkan jumlah uang yang akan dipinjam: ");
15         double jumlahPinjam = prasojo.nextDouble();
16         donny.pinjam(jumlahPinjam);
17         System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
18
19         while (donny.getJumlahPinjaman() > 0) {
20             System.out.println(x:"-----");
21             System.out.println("Jumlah pinjaman saat ini: Rp " + donny.getJumlahPinjaman());
22             System.out.print(s:"Masukkan jumlah angsuran: ");
23             double jumlahAngsuran = prasojo.nextDouble();
24             donny.angsur(jumlahAngsuran);
25
26             System.out.println(x:"-----");
27             System.out.println(x:"Selamat! Pinjaman Anda telah lunas.");
28             System.out.println("Jumlah pinjaman saat ini: Rp " + donny.getJumlahPinjaman());
29
30             prasojo.close();
31         }
32     }
```

```
Nama Anggota: Donny
Limit Pinjaman: 5000000.0

Masukkan jumlah uang yang akan dipinjam: 4500000
Jumlah pinjaman saat ini: 4500000.0
-----
Jumlah pinjaman saat ini: Rp 4500000.0
Masukkan jumlah angsuran: 1000000
-----
Jumlah pinjaman saat ini: Rp 3500000.0
Masukkan jumlah angsuran: 3500000
-----
Selamat! Pinjaman Anda telah lunas.
Jumlah pinjaman saat ini: Rp 0.0
```