

Object Oriented Programming Encapsulation

**Name**

Virza Aulia Rachman

NIM

2241720078

Class

1i

Department

Information Technology

Study Program

D4 Informatics Engineering

Practicum 1

```
package MotorEncapsulation;

2 usages
public class motor {
    2 usages
    public int kecepatan = 0;
    1 usage
    public boolean kontakOn = false;

    2 usages
    public void printStatus(){
        if(kontakOn == true){
            System.out.println("Kontak on");
        }
        else{
            System.out.println("Kontak off");
        }
        System.out.println("Kecepatan " + kecepatan+"\n");
    }
}
```

```
package MotorEncapsulation;

public class MotorDemo {
    public static void main(String[]args){
        motor motor = new motor();
        motor.printStatus();
        motor.kecepatan=50;
        motor.printStatus();
    }
}
```

```
Kontak off
Kecepatan 0
```

```
Kontak off
Kecepatan 50
```

Practicum 2

```
package MotorEncapsulation;

2 usages
public class motor {
    4 usages
    public int kecepatan = 0;
    5 usages
    public boolean kontakOn = false;

    1 usage
    public void nyalakanMesin(){
        kontakOn = true;
    }
    1 usage
    public void matikanMesin(){
        kontakOn = false;
        kecepatan = 0;
    }
    4 usages
    public void tambahKecepatan(){
        if(kontakOn == true){
            kecepatan += 5;
        }
        else{
            System.out.println("Kecepatan tidak bisa bertambah karena mesin Off! \n");
        }
    }
    no usages
    public void kurangiKecepatan(){
        if(kontakOn == true){
            kecepatan -= 5;
        }
        else{
            System.out.println("Kecepatan tidak bisa berkurang karena Mesin Off! \n");
        }
    }
    6 usages
    public void printStatus(){
        if(kontakOn == true){
            System.out.println("Kontak on");
        }
        else{
            System.out.println("Kontak off");
        }
        System.out.println("Kecepatan " + kecepatan + "\n");
    }
}
```

```
package MotorEncapsulation;

public class MotorDemo {
    public static void main(String[] args){
        motor motor = new motor();
        motor.printStatus();
        motor.tambahKecepatan();

        motor.nyalakanMesin();
        motor.printStatus();

        motor.tambahKecepatan();
        motor.printStatus();

        motor.tambahKecepatan();
        motor.printStatus();

        motor.tambahKecepatan();
        motor.printStatus();

        motor.matikanMesin();
        motor.printStatus();
    }
}
```

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

Question

1. Pada class TestMobil, saat kita menambah kecepatan untuk pertama kalinya, mengapa muncul peringatan "Kecepatan tidak bisa bertambah karena Mesin Off!"?
2. Mengapa atribut kecepatan dan kontakOn diset private?
3. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

Answer:

1. Karena kondisi Boolean KontakOn = false; jadi sesuai dengan method tambahKecepatan();
2. Agar tidak dapat diakses class lain dan menjadi atribut pribadi objek motor.
- 3.

```
public void tambahKecepatan(){
    if(kontakOn == true){
        if (kecepatan < 100){
            kecepatan += 5;
        }else{
            System.out.println("pelan pelan pak sopir");
        }
    } else{
        System.out.println("Kecepatan tidak bisa bertambah karena mesin Off! \n");
    }
}
```

Practicum 3

```
package koperasigetsetter;

2 usages
public class Anggota {
    2 usages
    private String nama;
    2 usages
    private String alamat;
    3 usages
    private float simpanan;

    1 usage
    public void setName(String nama){
        this.nama = nama;
    }

    1 usage
    public void setAddress(String alamat){
        this.alamat = alamat;
    }

    2 usages
    public String getName(){
        return nama;
    }

    no usages
    public String getAddress(){
        return alamat;
    }

    2 usages
    public float getSimpanan(){
        return simpanan;
    }

    1 usage
    public void setor(float uang){
        simpanan += uang;
    }

    1 usage
    public void pinjam(float uang){
        simpanan -= uang;
    }
}
```

```
package koperasigetsetter;

public class koperasiDemo {
    public static void main(String[] args){
        Anggota anggota1 = new Anggota();
        anggota1.setName("Virza Aulia");
        anggota1.setAlamat("jalan kembang kertas");
        anggota1.setor(100000);
        System.out.println("Simpanan " + anggota1.getName() + " : Rp " + anggota1.getSimpanan());

        anggota1.pinjam( uang: 5000);
        System.out.println("Simpanan " + anggota1.getName() + " : Rp " + anggota1.getSimpanan());
    }
}
```

```
Simpanan Virza Aulia : Rp 100000.0
Simpanan Virza Aulia : Rp 95000.0
```

Practicum 4

```
package koperasigetsetter;  
  
2 usages  
public class Anggota {  
    3 usages  
    private String nama;  
    3 usages  
    private String alamat;  
    4 usages  
    private float simpanan;  
  
    1 usage  
    Anggota(String nama, String alamat){  
        this.nama= nama;  
        this.alamat =alamat;  
        this.simpanan = 0;  
    }  
  
    1 usage  
    public void setName(String nama){  
        this.nama = nama;  
    }  
  
    1 usage  
    public void setAddress(String alamat){  
        this.alamat = alamat;  
    }  
  
    3 usages  
    public String getName(){  
        return nama;  
    }  
  
    no usages  
    public String getAddress(){  
        return alamat;  
    }  
  
    3 usages  
    public float getSimpanan(){  
        return simpanan;  
    }  
  
    1 usage  
    public void setor(float uang){  
        simpanan += uang;  
    }  
  
    1 usage  
    public void pinjam(float uang){  
        simpanan -= uang;  
    }  
}
```

```
package koperasigetsetter;  
  
public class KoperasiDemo {  
    public static void main(String[] args){  
        Anggota anggota1 = new Anggota( nama: "Virza", alamat: "jalan merak");  
        System.out.println("Simpanan "+anggota1.getnama()+ " : Rp "+ anggota1.getSimpanan());  
  
        anggota1.setName("Virza Aulia");  
        anggota1.setAlamat("jalan kembang kertas");  
        anggota1.setor(100000);  
        System.out.println("Simpanan "+ anggota1.getnama()+ " : Rp "+anggota1.getSimpanan());  
  
        anggota1.pinjam( uang: 5000);  
        System.out.println("Simpanan "+ anggota1.getnama()+ " : Rp "+anggota1.getSimpanan());  
    }  
}
```

```
Simpanan Virza : Rp 0.0  
Simpanan Virza Aulia : Rp 100000.0  
Simpanan Virza Aulia : Rp 95000.0
```

Question

1. Apa yang dimaksud getter dan setter?
2. Apa kegunaan dari method `getSimpanan()`?
3. Method apa yang digunakan untuk menambah saldo?
4. Apa yang dimaksud konstruktor?
5. Sebutkan aturan dalam membuat konstruktor?
6. Apakah boleh konstruktor bertipe `private`?
7. Kapan menggunakan parameter dengan passing parameter?
8. Apa perbedaan atribut class dan instansiasi atribut?
9. Apa perbedaan class method dan instansiasi method?

Answer:

1. Getter untuk mengambil value dari atribut sedangkan setter untuk memberikan value pada atribut
2. Untuk mengambil value dari atribut simpanan.
3. Method `setor()`;
4. Konstruktor mirip dengan method cara deklarasinya akan tetapi tidak memiliki tipe return. Dan konstruktor dieksekusi ketika instan dari objek dibuat.
5.
 - a. Nama konstruktor harus sama dengan nama class
 - b. Konstruktor tidak memiliki tipe data return
 - c. Konstruktor tidak boleh menggunakan modifier `abstract`, `static`, `final`, dan `synchronized`
6. Iya bisa
7. Saat mengirim nilai atau data ke dalam suatu fungsi atau metode saat memanggilnya untuk digunakan dalam operasi yang dijalankan oleh fungsi atau metode tersebut.
8. Atribut Class bersifat bersama-sama untuk semua objek kelas, sedangkan atribut instansiasi ada pada tingkat objek dan berbeda antara satu objek dengan objek lainnya.
9. Class method diakses melalui kelas dan berhubungan dengan atribut kelas, sedangkan instance method diakses melalui objek dan beroperasi pada atribut objek.

Assignment

1. This is the output:

```
Name : Aulia  
Age : 30
```

2. Karena pada program dijelaskan apabila value diatas 30 maka output yang dihasilkan adalah 30 dan karena rantai program itu menggunakan if else jadi kondisi yang sesuai dieksekusi.
3. This is the code:

```
public void setAge(int newAge){  
    if(newAge > 30 ){  
        age = 30;  
    } else if (newAge < 18) {  
        System.out.println("dibawah umur");  
    }else {  
        age = newAge;  
    }  
}
```

4. This is the code :

```
package Assignment;  
  
no usages  
public class Anggota {  
    1 usage  
    String NoKtp;  
    1 usage  
    String nama;  
    3 usages  
    int limit;  
    4 usages  
    int pinjaman;  
  
    no usages  
    Anggota(String ktp,String nama,int limit){  
        NoKtp = ktp;  
        this.nama = nama;  
        this.limit = limit;  
    }  
    no usages  
    int getLimitPinjaman(){  
        return limit;  
    }  
    1 usage  
    int getJumlahPinjaman(){  
        return pinjaman;  
    }  
    no usages  
    void pinjam(int nominal){  
        if (nominal <= limit){  
            pinjaman += nominal;  
        }else{  
            System.out.println("Maaf, jumlah pinjaman melebihi limit");  
        }  
    }  
    no usages  
    void angsur(int nominal){  
        if (nominal <= pinjaman){  
            pinjaman -= nominal;  
        }else{  
            System.out.println("kelebihan pak, utang anda hanya "+getJumlahPinjaman());  
        }  
    }  
}
```


5. This is the code :

```
void angsur(int nominal){
    if (nominal > (pinjaman * 10/100)){
        pinjaman -= nominal;
    }else{
        System.out.println("Maaf, angsuran harus 10% dari pinjaman ");
    }
}
```

6. This is the code :

```
package Assignment;

import java.util.Scanner;
public class TestKoperasi {
    public static void main(String[]args){
        Anggota donny = new Anggota( ktp: "111333444", nama: "Donny", limit: 5000000);
        Scanner sc = new Scanner(System.in);

        System.out.println("Nama anggota : "+donny.getNama());
        System.out.println("Limit Pinjaman : "+donny.getLimitPinjaman());

        System.out.println("Pinjam berapa : ");int pinjaman1 = sc.nextInt();
        donny.pinjam(pinjaman1);
        System.out.println("Jumlah Pinjaman saat ini : "+donny.getjumlahPinjaman());

        System.out.println("Angsur berapa : ");int angsuran1 = sc.nextInt();
        donny.angsur(angsuran1);
        System.out.println("Jumlah Pinjaman saat ini : "+donny.getjumlahPinjaman());
    }
}
```