

# **OBJECT ORIENTED PROGRAM PRACTICUM**

## **JOB SHEET 3: ENKAPSULASI**



BY :

**ATHRIYA GENFERIN**

**D4 INFORMATICS ENGINEERING (2I)**

**2241720075**

**(03)**

**State Polytechnic of Malang**

**Soekarno Hatta street No.9, Malang, East Java 65141**

**2022/2023**



## Praktikum 1

```
package jobsheet3.motorencapsulation;

public class Motor {
    public int kecepatan = 0;
    public boolean kontakOn = false;

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

package jobsheet3.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
```

## Praktikum 2

```
package jobsheet3.motorencapsulation;

public class Motor {
    private int kecepatan = 0;
    private boolean kontakOn = false;

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



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

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

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

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

package jobsheet3.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();  
    }  
}
```

```
PS E:\SEMESTER 3\OOP\OOP> java MotorDemo  
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  
  
PS E:\SEMESTER 3\OOP\OOP>
```

#### Pertanyaan

1. Pada class TestMobil, saat kita menambah kecepatan untuk pertama kalinya, mengapa muncul peringatan “Kecepatan tidak bisa bertambah karena Mesin Off!”?

Pada class MotorDemo, saat kita menambah kecepatan untuk pertama kalinya, muncul peringatan "Kecepatan tidak bisa bertambah karena Mesin Off!" karena pada awalnya, mesin motor dalam keadaan mati (kontakOff). Ketika kita memanggil metode tambahKecepatan() untuk pertama kali, ia akan memeriksa apakah mesin dalam keadaan menyala (kontakOn == true) sebelum menambah kecepatan. Karena kontak masih mati, pesan peringatan tersebut ditampilkan.

2. Mengapa atribut kecepatan dan kontakOn diset private?

Atribut kecepatan dan kontakOn diset sebagai private untuk menerapkan prinsip enkapsulasi dalam pemrograman berorientasi objek (OOP). Dengan mengatur atribut sebagai private, kita mengontrol akses ke atribut tersebut, sehingga hanya metode-metode dalam kelas Motor yang dapat mengubah atau membaca nilai-nilai tersebut. Ini membantu dalam mencegah perubahan yang tidak sah dan menjaga konsistensi data dalam objek Motor.

3. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

```
public void tambahKecepatan() {  
    if (kontakOn == true) {  
        if (kecepatan < 100){  
            kecepatan += 5;  
        }  
    }  
}
```



```
        } else {  
            System.out.println("Kecepatan sudah mencapai maksimum (100)!");  
        }  
    } else {  
        System.out.println("Kecepatan tidak bisa bertambah karena Mesin  
Off! \n");  
    }  
}
```

### Praktikum 3

```
package jobsheet3.KoperasiGetterSetter;  
  
public class Anggota {  
    private String nama;  
    private String alamat;  
    private float simpanan;  
  
    public void setNama(String nama) {  
        this.nama = nama;  
    }  
  
    public void setAlamat(String alamat) {  
        this.alamat = alamat;  
    }  
  
    public String getNama() {  
        return nama;  
    }  
  
    public String getAlamat() {  
        return alamat;  
    }  
  
    public float getSimpanan() {  
        return simpanan;  
    }  
  
    public void setor(float uang) {  
        simpanan += uang;  
    }  
  
    public void pinjam(float uang) {  
        simpanan -= uang;  
    }  
}
```



```
package jobsheet3.KoperasiGetterSetter;

public class KoperasiDemo {
    public static void main(String[] args) {
        Anggota anggota1 = new Anggota();
        anggota1.setNama("Iwan Setiawan");
        anggota1.setAlamat("Jalan Sukarno Hatta no 10");
        anggota1.setor(100000);
        System.out.println("Simpanan " + anggota1.getNama() + " : Rp " +
anggota1.getSimpanan());
        anggota1.pinjam(5000);
        System.out.println("Simpanan " + anggota1.getNama() + " : Rp " +
anggota1.getSimpanan());
    }
}
```

```
Simpanan Iwan Setiawan : Rp 100000.0
Simpanan Iwan Setiawan : Rp 95000.0
PS E:\SEMESTER 3\OOP\OOP>
```

#### Praktikum 4

```
package jobsheet3.KoperasiGetterSetter;

public class KoperasiDemo {
    public static void main(String[] args) {
        Anggota anggota1 = new Anggota();
        System.out.println("Simpanan " + anggota1.getNama() + " : Rp " +
anggota1.getSimpanan());

        anggota1.setNama("Iwan Setiawan");
        anggota1.setAlamat("Jalan Sukarno Hatta no 10");
        anggota1.setor(100000);
        System.out.println("Simpanan " + anggota1.getNama() + " : Rp " +
anggota1.getSimpanan());
        anggota1.pinjam(5000);
        System.out.println("Simpanan " + anggota1.getNama() + " : Rp " +
anggota1.getSimpanan());
    }
}
```

```
Simpanan null : Rp 0.0
Simpanan Iwan Setiawan : Rp 100000.0
Simpanan Iwan Setiawan : Rp 95000.0
PS E:\SEMESTER 3\OOP\OOP>
```



```
package jobsheet3.KoperasiGetterSetter;

public class KoperasiDemo {
    public static void main(String[] args) {
        Anggota anggota1 = new Anggota("Iwan", "Jalan Mawar");
        System.out.println("Simpanan " + anggota1.getNama() + " : Rp " +
            anggota1.getSimpanan());

        anggota1.setNama("Iwan Setiawan");
        anggota1.setAlamat("Jalan Sukarno Hatta no 10");
        anggota1.setor(100000);
        System.out.println("Simpanan " + anggota1.getNama() + " : Rp " +
            anggota1.getSimpanan());
        anggota1.pinjam(5000);
        System.out.println("Simpanan " + anggota1.getNama() + " : Rp " +
            anggota1.getSimpanan());
    }
}
```

```
package jobsheet3.KoperasiGetterSetter;

public class Anggota {
    private String nama;
    private String alamat;
    private float simpanan;

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

    public void setNama(String nama) {
        this.nama = nama;
    }

    public void setAlamat(String alamat) {
        this.alamat = alamat;
    }

    public String getNama() {
        return nama;
    }

    public String getAlamat() {
        return alamat;
    }
}
```



```
public float getSimpanan() {  
    return simpanan;  
}  
  
public void setor(float uang) {  
    simpanan += uang;  
}  
  
public void pinjam(float uang) {  
    simpanan -= uang;  
}  
}
```

```
Simpanan Iwan : Rp 0.0  
Simpanan Iwan Setiawan : Rp 100000.0  
Simpanan Iwan Setiawan : Rp 95000.0  
PS E:\SEMESTER 3\OOP\OOP>
```

Pertanyaan – Percobaan 3 dan 4

**1. Apa yang dimaksud getter dan setter?**

Metode yang digunakan dalam pemrograman untuk mengakses (getter) dan mengubah (setter) nilai-nilai atribut atau properti dalam sebuah objek. Getter digunakan untuk mengambil nilai atribut, sedangkan setter digunakan untuk mengganti nilai atribut.

**2. Apa kegunaan dari method getSimpanan()?**

Tujuan dari method getSimpanan() adalah untuk mengambil nilai atribut simpanan pada objek Anggota. Dalam konteks ini, method ini digunakan untuk mendapatkan saldo atau simpanan anggota.

**3. Method apa yang digunakan untuk menambah saldo?**

Untuk menambah saldo, kita menggunakan method setor(float uang). Method ini digunakan untuk menambahkan uang ke saldo simpanan anggota.

**4. Apa yang dimaksud konstruktor?**

Konstruktor adalah metode khusus dalam kelas yang digunakan untuk menginisialisasi objek saat objek tersebut dibuat (diinstansiasi). Konstruktor memiliki nama yang sama dengan nama kelas. Dalam kasus Anggota, terdapat konstruktor yang digunakan untuk menginisialisasi nama, alamat, dan simpanan anggota saat objek anggota dibuat.

**5. Sebutkan aturan dalam membuat konstruktor?**

- Nama konstruktor harus sama dengan nama kelas.
- Konstruktor tidak memiliki tipe pengembalian (void, int, dll.).
- Kita dapat memiliki beberapa konstruktor dalam satu kelas dengan parameter yang berbeda (overloading).

**6. Apakah boleh konstruktor bertipe private?**





Konstruktor boleh memiliki tipe akses private. Konstruktor dengan tipe akses private akan membuat kelas tersebut tidak dapat diakses secara langsung dari luar kelas, kecuali jika kelas tersebut memiliki metode lain yang dapat mengakses konstruktor tersebut.

**7. Kapan menggunakan parameter dengan passing parameter?**

Parameter dengan passing parameter digunakan saat ingin mengirim data atau informasi ke suatu metode agar dapat digunakan di dalam metode tersebut. Ini berguna ketika ingin melakukan operasi atau mengubah sesuatu dalam metode dengan menggunakan nilai yang berasal dari luar metode.

**8. Apa perbedaan atribut class dan instansiasi atribut?**

Atribut kelas adalah atribut yang bersifat statis dan digunakan bersama oleh semua objek dari kelas yang sama. Mereka dideklarasikan dengan kata kunci static. Sementara atribut instansiasi adalah atribut yang dimiliki oleh setiap objek dari kelas dan nilainya dapat berbeda antar objek.

**9. Apa perbedaan class method dan instansiasi method?**

Class method (juga disebut static method) adalah metode yang berkaitan dengan kelas itu sendiri, bukan dengan instance objek dari kelas tersebut. Mereka dideklarasikan dengan kata kunci static dan dapat diakses melalui nama kelas tanpa harus membuat objek. Sedangkan instance method adalah metode yang berhubungan dengan instance objek dari kelas tersebut. Mereka dapat mengakses dan mengubah nilai-nilai atribut objek. Untuk mengakses instance method, Anda perlu membuat objek dari kelas tersebut terlebih dahulu.

## Tugas

**1. . Cobalah program dibawah ini dan tuliskan hasil outputnya**

```
package jobsheet3.Tugas1;

public class EncapDemo {
    private String name;
    private int age;

    public String getName() {
        return name;
    }

    public void setName(String newName) {
        name = newName;
    }

    public int getAge() {
        return age;
    }
}
```



```
public void setAge(int newAge) {  
    if (newAge > 30) {  
        age = 30;  
    } else {  
        age = newAge;  
    }  
}  
  
}  
package jobsheet3.Tugas1;  
  
public class EncapTest {  
  
    public static void main(String[] args) {  
        EncapDemo encap = new EncapDemo();  
        encap.setName("James");  
        encap.setAge(35);  
  
        System.out.println("Name : " + encap.getName());  
        System.out.println("Age : " + encap.getAge());  
    }  
}
```

```
Name : James  
Age : 30  
PS E:\SEMESTER 3\OOP\OOP>
```

2. Pada program diatas, pada class EncapTest kita mengeset age dengan nilai 35, namun pada saat ditampilkan ke layar nilainya 30, jelaskan mengapa

Terdapat batasan pada setter setAge(int newAge) dalam class EncapDemo. Dalam setter ini, ada kondisi yang mengatur bahwa jika nilai newAge lebih dari 30, maka nilai age akan diatur menjadi 30. Ketika mengeset usia (age) dengan nilai 35 di class EncapTest, nilai tersebut akan dikonfirmasi oleh setter setAge di class EncapDemo. Karena 35 lebih besar dari 30, maka setter akan mengatur nilai age menjadi 30 sesuai dengan aturan yang telah ditentukan dalam setter tersebut.

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;  
    } else if (newAge < 18) {  
        age = 18;  
    } else {  
        age = newAge;  
    }  
}
```



}

4. Pada sebuah sistem informasi koperasi simpan pinjam, terdapat class Anggota yang memiliki atribut antara lain nomor KTP, nama, limit peminjaman, dan jumlah pinjaman. Anggota dapat meminjam uang dengan batas limit peminjaman yang ditentukan. Anggota juga dapat mengangsur pinjaman. Ketika Anggota tersebut mengangsur pinjaman, maka jumlah pinjaman akan berkurang sesuai dengan nominal yang diangsur. Buatlah class Anggota tersebut, berikan atribut, method dan konstruktor sesuai dengan kebutuhan. Uji dengan TestKoperasi berikut ini untuk memeriksa apakah class Anggota yang anda buat telah sesuai dengan yang diharapkan

```
package jobsheet3.Tugas4;
```

```
public class Anggota {  
    private String nomorKTP;  
    private String nama;  
    private double limitPinjaman;  
    private double jumlahPinjaman;  
  
    public Anggota(String nomorKTP, String nama, double limitPinjaman) {  
        this.nomorKTP = nomorKTP;  
        this.nama = nama;  
        this.limitPinjaman = limitPinjaman;  
        this.jumlahPinjaman = 0; // Saat anggota dibuat, jumlah pinjaman  
        awalnya adalah 0  
    }  
  
    public String getNomorKTP() {  
        return nomorKTP;  
    }  
  
    public String getNama() {  
        return nama;  
    }  
  
    public double getLimitPinjaman() {  
        return limitPinjaman;  
    }  
  
    public double getJumlahPinjaman() {  
        return jumlahPinjaman;  
    }  
  
    public void pinjam(double jumlah) {  
        if (jumlah <= limitPinjaman) {  
            jumlahPinjaman += jumlah;  
            System.out.println("Pinjaman berhasil. Jumlah pinjaman saat  
            ini: " + jumlahPinjaman);  
        }  
    }  
}
```



```
        } else {  
            System.out.println("Maaf, jumlah pinjaman melebihi limit.");  
        }  
    }  
  
    public void angsur(double jumlah) {  
        if (jumlah > 0) {  
            if (jumlah <= jumlahPinjaman) {  
                jumlahPinjaman -= jumlah;  
                System.out.println("Angsuran berhasil. Jumlah pinjaman saat  
ini: " + jumlahPinjaman);  
            } else {  
                System.out.println("Maaf, jumlah angsuran melebihi jumlah  
pinjaman saat ini.");  
            }  
        } else {  
            System.out.println("Maaf, jumlah angsuran harus lebih dari  
0.");  
        }  
    }  
}
```

```
package jobsheet3.Tugas4;
```

```
public class TestKoperasi {  
    public static void main(String[] args) {  
        Anggota donny = new Anggota("111333444", "Donny", 5000000);  
        System.out.println("Nama Anggota: " + donny.getNama());  
        System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());  
  
        System.out.println("\nMeminjam uang 10.000.000...");  
        donny.pinjam(10000000);  
  
        System.out.println("\nMeminjam uang 4.000.000...");  
        donny.pinjam(4000000);  
        System.out.println("Jumlah pinjaman saat ini: " +  
donny.getJumlahPinjaman());  
  
        System.out.println("\nMembayar angsuran 1.000.000");  
        donny.angsur(1000000);  
        System.out.println("Jumlah pinjaman saat ini: " +  
donny.getJumlahPinjaman());  
  
        System.out.println("\nMembayar angsuran 3.000.000");  
        donny.angsur(3000000);  
    }  
}
```



```
        System.out.println("Jumlah pinjaman saat ini: " +  
donny.getJumlahPinjaman());  
    }  
}
```

```
Nama Anggota: Donny  
Meminjam uang 10.000.000...  
Maaf, jumlah pinjaman melebihi limit.  
  
Meminjam uang 4.000.000...  
Pinjaman berhasil. Jumlah pinjaman saat ini: 4000000.0  
  
Jumlah pinjaman saat ini: 4000000.0  
  
Membayar angsuran 1.000.000  
Angsuran berhasil. Jumlah pinjaman saat ini: 3000000.0  
  
Jumlah pinjaman saat ini: 3000000.0  
  
Membayar angsuran 3.000.000  
Angsuran berhasil. Jumlah pinjaman saat ini: 0.0  
Jumlah pinjaman saat ini: 0.0  
PS E:\SEMESTER 3\OOP\OOP>
```

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 jumlah) {  
    double minimumAngsuran = 0.1 * jumlahPinjaman; // 10% dari jumlah  
    pinjaman  
    if (jumlah >= minimumAngsuran) {  
        if (jumlah <= jumlahPinjaman) {  
            jumlahPinjaman -= jumlah;  
            System.out.println("Angsuran berhasil. Jumlah pinjaman saat  
ini: " + jumlahPinjaman);  
        } else {  
            System.out.println("Maaf, jumlah angsuran melebihi jumlah  
pinjaman saat ini.");  
        }  
    } else {  
        System.out.println("Maaf, angsuran harus 10% dari jumlah  
pinjaman.");  
    }  
}
```



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

```
package jobsheet3.Tugas4;

import java.util.Scanner;

public class TestKoperasi {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        Anggota donny = new Anggota("111333444", "Donny", 5000000);
        System.out.println("Nama Anggota: " + donny.getNama());
        System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());

        System.out.print("\nMeminjam uang. Masukkan jumlah pinjaman: ");
        double jumlahPinjaman = scanner.nextDouble();
        donny.pinjam(jumlahPinjaman);

        System.out.print("\nMeminjam uang. Masukkan jumlah pinjaman: ");
        jumlahPinjaman = scanner.nextDouble();
        donny.pinjam(jumlahPinjaman);
        System.out.println("Jumlah pinjaman saat ini: " +
donny.getJumlahPinjaman());

        System.out.print("\nMembayar angsuran. Masukkan jumlah angsuran:
");
        double jumlahAngsuran = scanner.nextDouble();
        donny.angsur(jumlahAngsuran);
        System.out.println("Jumlah pinjaman saat ini: " +
donny.getJumlahPinjaman());

        System.out.print("\nMembayar angsuran. Masukkan jumlah angsuran:
");
        jumlahAngsuran = scanner.nextDouble();
        donny.angsur(jumlahAngsuran);
        System.out.println("Jumlah pinjaman saat ini: " +
donny.getJumlahPinjaman());

        scanner.close();
    }
}
```



ATHRIYA GENFERIN

2I – 03 - 2241720075

D-IV INFORMATICS ENGINEERING

Nama Anggota: Donny

Limit Pinjaman: 5000000.0

Meminjam uang. Masukkan jumlah pinjaman: 2000000

Pinjaman berhasil. Jumlah pinjaman saat ini: 2000000.0

Meminjam uang. Masukkan jumlah pinjaman: 5000000

Pinjaman berhasil. Jumlah pinjaman saat ini: 7000000.0

Jumlah pinjaman saat ini: 7000000.0

Membayar angsuran. Masukkan jumlah angsuran: 1000000

Angsuran berhasil. Jumlah pinjaman saat ini: 6000000.0

Jumlah pinjaman saat ini: 6000000.0

Membayar angsuran. Masukkan jumlah angsuran: 6000000

Angsuran berhasil. Jumlah pinjaman saat ini: 0.0

Jumlah pinjaman saat ini: 0.0

PS E:\SEMESTER 3\OOP\OOP> █