JOBSHEET 3

ENKAPSULASI PBO 1

Hawa Esanda

TI-2I_11_2241720079

PERCOBAAN 1 - ENKAPSULASI

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

    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");
    }
}
```

```java
public class MotorDemo {
    Run | Debug
    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
```

PERCOBAAN 2 - ACCESS MODIFIER

```java
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 tambahKecapatan(){
        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();
        }
        System.out.println("Kecepatan "+kecepatan+"\n");
    }
}
```

```java
public class MotorDemo {
    Run | Debug
    public static void main(String[] args) {
        Motor motor = new Motor();
        motor.printStatus();
        motor.tambahKecapatan();

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

        motor.tambahKecapatan();
        motor.printStatus();

        motor.tambahKecapatan();
        motor.printStatus();

        motor.tambahKecapatan();
        motor.printStatus();

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

```
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

A. Pada class TestMobil, saat kita menambah kecepatan untuk pertama kalinya, mengapa muncul peringatan “Kecepatan tidak bisa bertambah karena Mesin Off!”?
Jawab : Karena mesin mati, perintah motor.tambahKecapatan(); di dalam method main pertama kali dipanggil akan menjalankan blok kode di bagian else dalam method tambahKecapatan(), yang mencetak pesan kesalahan tersebut.

B. Mengapat atribut kecepatan dan kontakOn diset private?
Jawab : untuk menerapkan konsep enkapsulasi yaitu salah satu prinsip dalam pemrograman berorientasi objek untuk menyembunyikan detail internal dari suatu objek dan hanya memberikan akses melalui metode-metode yang ditentukan. Dengan mengatur atribut ke private, kita dapat membatasi akses langsung ke atribut-atribut tersebut dari luar kelas Motor yang membantu dalam menghindari perubahan yang tidak diinginkan atau manipulasi data.

C. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!
Jawab :

```java
public void tambahKecapatan(){
    if (kontakOn){
        if (kecepatan < 100){
            kecepatan += 5;
        }else{
            System.out.println(x:"Kecepatan telah mencapai maksimal 100!");
        }
    } else{
        System.out.println(x:"Kecepatan tidak bisa bertambah karena mesin off! \n");
    }
}

public void kurangiKecepatan(){
    if(kontakOn && kecepatan > 0){
        kecepatan -= 5;
    } else if (!kontakOn){
        System.out.println(x:"Kecepatan tidak bisa berkurang karena mesin off! \n");
    } else {
        System.out.println(x:"Kecepatan sudah mencapai 0!");
    }
}
```

```
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
```

PERCOBAAN 3 - GETTER DAN SETTER

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

```java
public class KoperasiDemo {
    Run | Debug
    public static void main(String[] args) {
        Anggota a1 = new Anggota();
        a1.setNama(nama:"Iwan Setiawan");
        a1.setAlamat(alamat:"Jalan Sukarno Hatta no 10");
        a1.setor(uang:100000);
        System.out.println("Simpanan "+a1.getNama()+ " : Rp "+a1.getSimpan

        a1.pinjam(uang:5000);
        System.out.println("Simpanan "+a1.getNama()+ " : Rp " +a1.getSimpa
    }
}
```

```
Simpanan Iwan Setiawan : Rp 100000.0
Simpanan Iwan Setiawan : Rp 95000.0
```

PERCOBAAN 4 - KONSTRUKTOR, INSTANSIASI

```
System.out.println("Simpanan "+a1.getNama()+ ": "+a1.getSimpanan()
```

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

```
Anggota a1 = new Anggota(nama:"Iwan", alamat:"Jalan Mawar");
```

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

PERTANYAAN - Percobaan 3 dan 4

1. Apa yang dimaksud getter dan setter?
   Jawab : Getter adalah public method dan memiliki tipe data return, yang berfungsi untuk mendapatkan nilai dari atribut private. Sedangkan setter adalah public method yang tidak memliki tipe data return, yang berfungsi untuk memanipulasi nilai dari atribut private.

2. Apa kegunaan dari method getSimpanan()?
   Jawab : Metode tersebut digunakan untuk mengambil nilai dari variable simpanan. Metode tersebut merupakan contoh dari metode getter.

3. Method apa yang digunakan untk menambah saldo?
   Jawab : Metode yang digunakan adalah metode setor()

4. Apa yand dimaksud konstruktor?
   Jawab : Konstruktor mirip dengan method cara deklarasinya akan tetapi tidak memiliki tipe return. Dan konstruktor dieksekusi ketika instan dari objek dibuat. Jadi setiap kali sebuat objek dibuat dengan keyword new() maka konstruktor akan dieksekusi.

5. Sebutkan aturan dalam membuat konstruktor?
   Jawab :
   - Nama konstruktor harus sama dengan nama class
   - Konstruktor tidak memiliki tipe data return
   - Konstruktor tidak boleh menggunakan modifier abstract, static, final, dan synchronized

6. Apakah boleh konstruktor bertipe private?
   Jawab : Konstruktor boleh bertipe private asalkan kelas yang dipakai sama.

7. Kapan menggunakan parameter dengan passsing parameter?

Jawab : Parameter dengan passing parameter biasanya digunakan ketika Anda ingin mengirimkan nilai atau data dari satu fungsi ke fungsi lainnya.

8. Apa perbedaan atribut class dan instansiasi atribut?
   Jawab : Atribut class adalah variabel yang didefinisikan di tingkat kelas dan dibagi oleh semua instance kelas tersebut. Sedangkan instansiasi atribut (atau atribut instance) adalah variabel yang didefinisikan di tingkat instance dan unik untuk setiap instance kelas.

9. Apa perbedaan class method dan instansiasi method?
   Jawab : Class method adalah metode yang berhubungan dengan kelas itu sendiri dan bukan dengan instance kelas tersebut. Sedangkan instansiasi method (atau instance method) adalah metode yang berhubungan dengan instance kelas dan bukan dengan kelas itu sendiri

TUGAS

1. Cobalah program dibawah ini dan tuliskan hasil outputnya

```java
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;
        }
    }
}
```

```java
public class EncapTest {
    Run | Debug
    public static void main(String[] args) {
        EncapDemo encap = new EncapDemo();
        encap.setName(newName:"James");
        encap.setAge(newAge:35);

        System.out.println("Name : " +encap.getName());
        System.out.println("Age : " +encap.getAge());
    }
}
```

```
Name : James
Age : 30
```

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

Jawab : Pada program di atas, metode setAge(int newAge) dalam kelas EncapDemo memiliki logika untuk membatasi nilai age menjadi maksimal 30. Jadi, jika mengatur age dengan nilai yang lebih besar dari 30, seperti 35 maka, metode tersebut akan mengatur age menjadi 30.

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

```
public void setAge(int newAge){
    if (newAge >= 18 && newAge <= 30) {
        age = newAge;
    } else if (newAge < 18) {
        age = 18; // Set age to the minimum value (18)
    } else {
        age = 30; // Set age to the maximum value (30)
    }
}
```

```
Name : James
Age : 30
```

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.
   Jawab :

```java
public class Anggota1 {
    private String nomorKTP;
    private String nama;
    private double limitPeminjaman;
    private double jumlahPinjaman;

    public Anggota1(String nomorKTP, String nama, double limitPeminjaman) {
        this.nomorKTP = nomorKTP;
        this.nama = nama;
        this.limitPeminjaman = limitPeminjaman;
        this.jumlahPinjaman = 0; // Awalnya jumlah pinjaman adalah 0
    }

    public String getNomorKTP() {
        return nomorKTP;
    }

    public String getNama() {
        return nama;
    }

    public double getLimitPeminjaman() {
        return limitPeminjaman;
    }

    public double getJumlahPinjaman() {
        return jumlahPinjaman;
    }

    public void pinjam(double jumlah) {
        if (jumlah <= limitPeminjaman - jumlahPinjaman) {
            jumlahPinjaman += jumlah;
            System.out.println("Pinjaman sebesar Rp" + jumlah + " berhasil.");
        } else {
            System.out.println(x:"Maaf, jumlah pinjaman melebihi limit.");
        }
    }

    public void angsur(double jumlah) {
        if (jumlah <= jumlahPinjaman) {
            jumlahPinjaman -= jumlah;
            System.out.println("Angsuran sebesar Rp" + jumlah + " berhasil.");
        } else {
            System.out.println(x:"Maaf, jumlah angsuran melebihi jumlah pinjaman.");
        }
    }
}
```

```java
public class TestKoperasi {
    Run | Debug
    public static void main(String[] args) {
        Anggota1 donny = new Anggota1(nomorKTP:"111333444", nama:"Donny", limitPeminjaman:5000000);
        System.out.println("Nama Anggota: " +donny.getNama());
        System.out.println("Limit Pinjaman: " +donny.getLimitPeminjaman());

        System.out.println(x:"\nMeminjam uang 10.000.000...");
        donny.pinjam(jumlah:10000000);
        System.out.println("Jumlah pinjaman saat ini: " +donny.getJumlahPinjaman());

        System.out.println(x:"\nMeminjam uang 4.000.000...");
        donny.pinjam(jumlah:4000000);
        System.out.println("Jumlah pinjaman saat ini: " +donny.getJumlahPinjaman());

        System.out.println(x:"\nMeminjam uang 1.000.000...");
        donny.pinjam(jumlah:1000000);
        System.out.println("Jumlah pinjaman saat ini: " +donny.getJumlahPinjaman());

        System.out.println(x:"\nMeminjam uang 3.000.000...");
        donny.pinjam(jumlah:3000000);
        System.out.println("Jumlah pinjaman saat ini: " +donny.getJumlahPinjaman());
    }
}
```

```
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...
Pinjaman sebesar Rp4000000.0 berhasil.
Jumlah pinjaman saat ini: 4000000.0

Meminjam uang 1.000.000...
Pinjaman sebesar Rp1000000.0 berhasil.
Jumlah pinjaman saat ini: 5000000.0

Meminjam uang 3.000.000...
Maaf, jumlah pinjaman melebihi limit.
Jumlah pinjaman saat ini: 5000000.0
```

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”.

Jawab :

```java
public void angsur(double jumlah) {
    double minimalAngsuran = 0.1 * jumlahPinjaman; //10% dari jumlah pinjaman
    if (jumlah >= jumlahPinjaman) {
        jumlahPinjaman -= jumlah;
        System.out.println("Angsuran sebesar Rp" + jumlah + " berhasil.");
    } else {
        System.out.println(x:"Maaf, jumlah angsuran harus 10% dari jumlah pinjaman.");
    }
}
```

```java
System.out.println(x:"\nMengangsur uang 2.000.000...");
donny.angsur(jumlah:2000000);
System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
```

```
Meminjam uang 3.000.000...
Maaf, jumlah pinjaman melebihi limit.
Jumlah pinjaman saat ini: 5000000.0

Mengangsur uang 2.000.000...
Maaf, jumlah angsuran harus 10% dari jumlah pinjaman.
Jumlah pinjaman saat ini: 5000000.0
```

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

Jawab :

```java
import java.util.Scanner;

public class TestKoperasi {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Masukkan nomor KTP: ");
        String nomorKTP = sc.nextLine();

        System.out.print(s:"Masukkan nama: ");
        String nama = sc.nextLine();

        System.out.print(s:"Masukkan limit peminjaman: ");
        double limitPeminjaman = sc.nextDouble();

        Anggota1 donny = new Anggota1(nomorKTP:"111333444", nama:"Donny", limitPeminjaman:5000000);
        System.out.println("Nama Anggota: " +donny.getNama());
        System.out.println("Limit Pinjaman: " +donny.getLimitPeminjaman());

        System.out.print(s:"\nMasukkan jumlah pinjaman: ");
        double jumlahPinjaman = sc.nextDouble();
        donny.pinjam(jumlahPinjaman);
        System.out.println("Jumlah pinjaman saat ini: " +donny.getJumlahPinjaman());

        System.out.print(s:"\nMasukkan jumlah angsuran: ");
        double jumlahAngsuran = sc.nextDouble();
        donny.angsur(jumlahAngsuran);
        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
    }
}
```

```
Masukkan nomor KTP: 00123456
Masukkan nama: Donny
Masukkan limit peminjaman: 50000000
Nama Anggota: Donny
Limit Pinjaman: 5000000.0
Limit Pinjaman: 5000000.0

Masukkan jumlah pinjaman: 3000000
Pinjaman sebesar Rp3000000.0 berhasil.
Jumlah pinjaman saat ini: 3000000.0

Masukkan jumlah angsuran: 2000000
Maaf, jumlah angsuran harus 10% dari jumlah pinjaman.
Jumlah pinjaman saat ini: 3000000.0
```