

**JOBSHEET 3**  
**OBJECT ORIENTED PROGRAMMING**



Arranged By :

Lenka Melinda Florianka

2241720074

Class 2I

**INFORMATION TECHNOLOGY**  
**D-IV INFORMATICS ENGINEERING**  
**MALANG STATE POLYTECHNIC**  
**2023**

## Practicum 1

Code :

```
1 package jobsheet3.motorencapsulation;
2
3 public class MotorDemo {
4     public static void main(String[] args) {
5         Motor motor = new Motor();
6         motor.printStatus();
7         motor.kecepatan = 50;
8         motor.printStatus();
9     }
10 }
11
```

```
1 package jobsheet3.motorencapsulation;
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        } else {
11            System.out.println("Kontak off");
12        }
13        System.out.println("Kecepatan " + kecepatan + "\n");
14    }
15 }
```

Output :

```
Kontak off
Kecepatan 0

Kontak off
Kecepatan 50

PS C:\Dev\00P>
```

## Practicum 2

### Code :

```
1 package jobsheet3.motorencapsulation;
2
3 public class MotorDemo {
4     public static void main(String[] args) {
5         Motor motor = new Motor();
6         motor.printStatus();
7         motor.tambahKecepatan();
8
9         motor.nyalakanMesin();
10        motor.printStatus();
11
12        motor.tambahKecepatan();
13        motor.printStatus();
14
15        motor.tambahKecepatan();
16        motor.printStatus();
17
18        motor.tambahKecepatan();
19        motor.printStatus();
20
21        motor.matikanMesin();
22        motor.printStatus();
23    }
24 }
25
26
```

```
1 package jobsheet3.motorencapsulation;
2
3 public class Motor {
4     private int kecepatan = 0;
5     private boolean kontakOn = false;
6
7     public void nyalakanMesin(){
8         kontakOn = true;
9     }
10
11    public void matikanMesin(){
12        kontakOn = false;
13        kecepatan = 0;
14    }
15
16    public void tambahKecepatan(){
17        if (kontakOn == true){
18            kecepatan += 5;
19        } else {
20            System.out.println("Kecepatan tidak bisa bertambah karena Mesin Off! \n");
21        }
22    }
23
24    public void kurangiKecepatan(){
25        if (kontakOn == true){
26            kecepatan -= 5;
27        } else {
28            System.out.println("Kecepatan tidak bisa berkurang karena Mesin Off! \n");
29        }
30    }
31
32    public void printStatus(){
33        if (kontakOn == true){
34            System.out.println("Kontak On");
35        } else {
36            System.out.println("Kontak off");
37        }
38        System.out.println("Kecepatan " + kecepatan + "\n");
39    }
40 }
```

### Output :

```
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 C:\Dev\00P>
```

## 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!

```
1 public void tambahKecepatan() {
2     if (kontakOn == true) {
3         if (kecepatan < 100) {
4             kecepatan += 5;
5         } else {
6             System.out.println("Kecepatan sudah mencapai maksimum (100)!");
7         }
8     } else {
9         System.out.println("Kecepatan tidak bisa bertambah karena Mesin Off!\n");
10    }
11 }
```

## Practicum 3

Code :

```
1 package jobsheet3.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
12    public void setAlamat(String alamat){
13        this.alamat = alamat;
14    }
15
16    public String getNama(){
17        return nama;
18    }
19
20    public String getAlamat(){
21        return alamat;
22    }
23
24    public float getSimpanan(){
25        return simpanan;
26    }
27
28    public void setor(float uang){
29        simpanan += uang;
30    }
31
32    public void pinjam(float uang){
33        simpanan -= uang;
34    }
35 }
```

```
1 package jobsheet3.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
12    public void setAlamat(String alamat){
13        this.alamat = alamat;
14    }
15
16    public String getNama(){
17        return nama;
18    }
19
20    public String getAlamat(){
21        return alamat;
22    }
23
24    public float getSimpanan(){
25        return simpanan;
26    }
27
28    public void setor(float uang){
29        simpanan += uang;
30    }
31
32    public void pinjam(float uang){
33        simpanan -= uang;
34    }
35 }
```

Output :

```
Simpanan Iwan Setiawan : Rp 100000.0  
Simpanan Iwan Setiawan : Rp 95000.0  
PS C:\Dev\OOP>
```

## Practicum 4

Code :

```
1 public static void main(String[] args) {  
2     Anggota anggota1 = new Anggota();  
3     System.out.println("Simpanan " + anggota1.getNama() + " : Rp " + anggota1.getSimpanan());  
4 }
```

```
1 Anggota anggota1 = new Anggota("Iwan", "Jalan Mawar");  
2     System.out.println("Simpanan " + anggota1.getNama() + " : Rp " + anggota1.getSimpanan());
```

```
1 Anggota (String nama, String alamat){  
2     this.nama = nama;  
3     this.alamat = alamat;  
4     this.simpanan = 0;  
5 }
```

Output :

```
Simpanan null : Rp 0.0  
Simpanan Iwan Setiawan : Rp 100000.0  
Simpanan Iwan Setiawan : Rp 95000.0  
PS C:\Dev\OOP>
```

## Pertanyaan

1. Apa yang dimaksud getter dan setter?

Getter dan setter adalah metode dalam pemrograman yang digunakan untuk mengakses (getter) dan mengubah (setter) nilai-nilai atribut atau properti dalam sebuah objek. Getter digunakan untuk mendapatkan nilai dari atribut, sementara setter digunakan untuk mengubah nilai atribut.

## 2. Apa kegunaan dari method `getSimpanan()`?

Method `getSimpanan()` digunakan untuk mendapatkan (mengambil) nilai dari atribut simpanan pada objek Anggota. Dalam hal ini, method ini digunakan untuk mendapatkan nilai saldo atau simpanan anggota.

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

Method `setor(float uang)` digunakan untuk menambahkan saldo (uang) ke simpanan anggota. Itu memungkinkan anggota untuk menyetor uang ke akun mereka.

## 4. Apa yang dimaksud konstruktor?

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

## 5. Sebutkan aturan dalam membuat konstruktor?

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 menjadi 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 `passsing` parameter?

Parameter dengan `passsing` parameter digunakan ketika Anda ingin mengirim data atau informasi ke suatu metode agar dapat digunakan di dalam metode tersebut. Hal ini berguna ketika kita ingin melakukan operasi atau mengubah sesuatu dalam method dengan menggunakan nilai yang berasal dari luar metode.

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

Atribut kelas (class attributes) adalah atribut yang bersifat statis dan bersama-sama digunakan oleh semua objek dari kelas yang sama. Mereka dideklarasikan dengan kata kunci `static`. Sedangkan atribut instansiasi (instance attributes) 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 (atau disebut juga `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

### Code :

```
1 package jobsheet3.tugas;
2
3 public class EncapTest {
4     public static void main(String[] args) {
5         EncapDemo encap = new EncapDemo();
6         encap.setName("James");
7         encap.setAge(35);
8
9         System.out.println("Name : " + encap.getName());
10        System.out.println("Age : " + encap.getAge());
11    }
12 }
```

```
1 package jobsheet3.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 void setName(String newName){
12        name = newName;
13    }
14
15    public int getAge(){
16        return age;
17    }
18
19    public void setAge(int newAge){
20        if (newAge > 30){
21            age = 30;
22        } else {
23            age = newAge;
24        }
25    }
26 }
```

### Output :

```
Name : James
Age : 30
PS C:\Dev\00P>
```

## Pertanyaan

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

Pada program di atas, ketika Anda mengeset umur (age) dengan nilai 35 melalui metode `setAge(35)`, ada pengecekan dalam metode tersebut. Jika nilai yang diberikan melebihi 30, maka umur akan diatur menjadi 30. Oleh karena itu, ketika Anda mencetak umur dengan `encap.getAge()`, nilainya menjadi 30 karena telah dibatasi oleh logika di dalam metode `setAge(30)`.

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

```
1 public void setAge(int newAge) {
2     if (newAge ≥ 18 && newAge ≤ 30) {
3         age = newAge;
4     } else if (newAge < 18) {
5         age = 18;
6     } else {
7         age = 30;
8     }
9 }
10 }
```

3. 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.

Code :

```
1 public class TestKoperasi {
2     public static void main(String[] args) {
3         Anggota donny = new Anggota("111333444", "Donny", 5000000);
4         System.out.println("Nama Anggota: " + donny.getNama());
5         System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());
6         System.out.println("\nMeminjam uang 10.000.000 ... ");
7         donny.pinjam(10000000);
8         System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
9         System.out.println("\nMeminjam uang 4.000.000 ... ");
10        donny.pinjam(4000000);
11        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
12        System.out.println("\nMembayar angsuran 1.000.000");
13        donny.angsur(1000000);
14        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
15        System.out.println("\nMembayar angsuran 3.000.000");
16        donny.angsur(3000000);
17        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
18    }
19 }
```



```

1 package jobsheet3.tugas;
2
3 public class Anggota {
4     private String nomorKTP;
5     private String nama;
6     private double limitPeminjaman;
7     private double jumlahPinjaman;
8
9     public Anggota(String nomorKTP, String nama, double limitPeminjaman) {
10         this.nomorKTP = nomorKTP;
11         this.nama = nama;
12         this.limitPeminjaman = limitPeminjaman;
13         this.jumlahPinjaman = 0;
14     }
15
16     public String getNomorKTP() {
17         return nomorKTP;
18     }
19
20     public String getNama() {
21         return nama;
22     }
23
24     public double getLimitPinjaman() {
25         return limitPeminjaman;
26     }
27
28     public double getJumlahPinjaman() {
29         return jumlahPinjaman;
30     }
31
32     public void pinjam(double jumlah) {
33         if (jumlah <= limitPeminjaman - jumlahPinjaman) {
34             jumlahPinjaman += jumlah;
35             System.out.println("Jumlah pinjaman saat ini: " + jumlahPinjaman);
36         } else {
37             System.out.println("Maaf, jumlah pinjaman melebihi limit.");
38         }
39     }
40
41     public void angsur(double jumlah) {
42         if (jumlah <= jumlahPinjaman) {
43             jumlahPinjaman -= jumlah;
44         } else {
45             System.out.println("Maaf, jumlah angsuran melebihi jumlah pinjaman.");
46         }
47     }
48 }

```

Output :

```

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
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:\Dev\00P>

```

4. 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".

```
1 public void angsur(double jumlah) {
2     double minimumAngsuran = 0.1 * jumlahPinjaman; // 10% dari jumlah pinjaman saat ini
3     if (jumlah >= minimumAngsuran) {
4         jumlahPinjaman -= jumlah;
5     } else {
6         System.out.println("Maaf, angsuran harus 10% dari jumlah pinjaman.");
7     }
8 }
9 }
```

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

```
1 import java.util.Scanner;
2
3 public class TestKoperasi {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         Anggota donny = new Anggota("111333444", "Donny", 5000000);
8         System.out.println("Nama Anggota: " + donny.getNama());
9         System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());
10
11         System.out.print("\nMeminjam uang: ");
12         double pinjaman = scanner.nextDouble();
13         donny.pinjam(pinjaman);
14         System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
15
16         System.out.print("\nMembayar angsuran: ");
17         double angsuran = scanner.nextDouble();
18         donny.angsur(angsuran);
19         System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
20
21         scanner.close();
22     }
23 }
```