

## Jobsheet 3 – Encapsulation

### 3.1 Percobaan 1 – Enkapsulasi

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

```
src > MotorDemoJava > ...
1: public class MotorDemo {
2:     public static void main(String[] args) {
3:         Motor motor = new Motor();
4:         motor.printStatus();
5:         motor.kecepatan = 50;
6:         motor.printStatus();
7:     }
8: }
```

```
Kontak Off
Kecepatan 0

Kontak Off
Kecepatan 50
```

### 3.2 Percobaan 2 - Access Modifier

```
src > J Motor.java > Motor
1 public class Motor {
2     private int kecepatan = 0;
3     private boolean kontakOn = false;
4
5     public void nyalakanMesin(){
6         kontakOn = true;
7     }
8
9     public void matikanMesin(){
10        kontakOn = false;
11        kecepatan = 0;
12    }
13
14    public void tambahKecepatan(){
15        if (kontakOn == true){
16            kecepatan += 5;
17        }
18        else{
19            System.out.println("Kecepatan tidak bisa bertambah karena Mesin Off!\n");
20        }
21    }
22
23    public void kurangiKecepatan(){
24        if (kontakOn == true){
25            kecepatan -= 5;
26        }
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        }
36        else{
37            System.out.println("Kontak Off");
38        }
39        System.out.println("Kecepatan " + kecepatan + "\n");
40    }
41 }
```

```
src > J MotorDemo.java > -
1 public class MotorDemo {
2     Run (Debug
3     public static void main(String[] args) {
4         Motor motor = new Motor();
5         motor.printStatus();
6         motor.tambahKecepatan();
7         motor.printStatus();
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    }
```

```
Kontak Off
Kecepatan 0

Kecepatan tidak bisa bertambah karena Mesin Off!

Kontak Off
Kecepatan 0

Kontak On
Kecepatan 0

Kontak On
Kecepatan 5

Kontak On
Kecepatan 10

Kontak On
Kecepatan 15

Kontak Off
Kecepatan 0
```

### 3.3 Pertanyaan

1. Because the engine (kontakOn) is still **off**, so increasing speed is not allowed.
2. They are set **private** to follow encapsulation, so attributes cannot be accessed directly from outside the class.
3. Add a **limit check** in tambahKecepatan() so the speed cannot go above 100.

```
14 public void tambahKecepatan(){
15     if (kontakOn == true){
16         if (kecepatan < 100){
17             kecepatan += 5;
18         }
19     } else {
20         System.out.println("Kecepatan tidak bisa bertambah karena Mesin Off!\n");
21     }
```

### 3.4 Percobaan 3 - Getter dan Setter

```
1 public class Anggota {
2     private String nama;
3     private String alamat;
4     private float simpanan;
5
6     public void setName(String nama){
7         this.nama = nama;
8     }
9
10    public void setAlamat(String alamat){
11        this.alamat = alamat;
12    }
13
14    public String getName(){
15        return nama;
16    }
17
18    public String getAlamat(){
19        return alamat;
20    }
21
22    public float getSimpanan(){
23        return simpanan;
24    }
25
26    public void setor(float uang){
27        simpanan += uang;
28    }
29
30    public void pinjam(float uang){
31        simpanan -= uang;
32    }
33 }
```

```
src > J KoperasiDemo.java > % KoperasiDemo > 5) main(String[])
1 public class KoperasiDemo {
2     RunIDemp
3     public static void main(String[] args) {
4         Anggota anggota1 = new Anggota();
5         anggota1.setName("Iwan Setiawan");
6         anggota1.setAlamat("Jalan Sukarno Hatta no 10");
7         anggota1.setor(100000);
8         System.out.println("Simpanan " + anggota1.getName() + " : Rp " + anggota1.getSimpanan());
9
10        anggota1.pinjam(5000);
11        System.out.println("Simpanan " + anggota1.getName() + " : Rp " + anggota1.getSimpanan());
12    }
13 }
```

Simpanan Iwan Setiawan : Rp 100000.0

Simpanan Iwan Setiawan : Rp 95000.0

### 3.5 Percobaan 4 – Konstruktor, Instalasi

```
src > J KoperasiDemo.java > ...
1 public class KoperasiDemo {
2     public static void main(String[] args) {
3         Anggota anggota1 = new Anggota();
4         System.out.println("Simpanan " + anggota1.getName() + " : Rp " + anggota1.getSimpanan());
5
6         anggota1.setName("Iwan Setiawan");
7         anggota1.setAlamat("Jalan Sukarno Hatta no 10");
8         anggota1.setor(100000);
9         System.out.println("Simpanan " + anggota1.getName() + " : Rp " + anggota1.getSimpanan());
10
11        anggota1.pinjam(5000);
12        System.out.println("Simpanan " + anggota1.getName() + " : Rp " + anggota1.getSimpanan());
13    }
14 }
```

Simpanan null : Rp 0.0

Simpanan Iwan Setiawan : Rp 100000.0

Simpanan Iwan Setiawan : Rp 95000.0

```
src > J Anggota.java > Anggota > setAlamat(String)
1 public class Anggota {
2     private String nama;
3     private String alamat;
4     private float simpanan;
5
6     Anggota(String nama, String alamat){
7         this.nama = nama;
8         this.alamat = alamat;
9         this.simpanan = 0;
10    }
11
12    public void setName(String nama){
13        this.nama = nama;
14    }
15
16    public void setAlamat(String alamat){
17        this.alamat = alamat;
18    }
19
20    public String getName(){
21        return nama;
22    }
23
24    public String getAlamat(){
25        return alamat;
26    }
27
28    public float getSimpanan(){
29        return simpanan;
30    }
31
32    public void setor(float uang){
33        simpanan += uang;
34    }
35
36    public void pinjam(float uang){
37        simpanan -= uang;
38    }
39 }

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

### 3.6 Pertanyaan – Percobaan 3 dan 4

1. **Getter and Setter** are methods used to read (get) and modify (set) private attributes of a class.
2. getSimpanan() is used to **retrieve the current balance (savings)** value.
3. The method to add balance is **setor()**.
4. A **constructor** is a special method that initializes objects when they are created.
5. Rules: constructor has the **same name as the class**, no return type, can be overloaded.
6. Yes, a constructor can be **private**, usually for singleton or restricted object creation.
7. Use parameters with passing parameter when you need to **initialize an object with specific values** at creation.
8. A **class attribute** is shared by all objects (static), while an **instance attribute** belongs to each object separately.
9. A **class method** (static) can be called without creating an object, while an **instance method** requires an object to be called.

## 4 Kesimpulan

A

## 5 Tugas

1. Cobalah program dibawah ini dan tuliskan hasil outputnya

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

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

```
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.

**Answer:** The age shows 30 instead of 35 because in setAge, if the value is greater than 30, it is forced to 30.

3. Modify setAge to limit the value between 18 and 30:

```
17 public void setAge(int newAge) {
18     if (newAge > 30) {
19         age = 30;
20     } else if (newAge < 18) {
21         age = 18;
22     } else {
23         age = newAge;
24     }
25 }
```

```
5 encap.setAge(newAge:15);
```

```
Name : James
Age : 18
```

4. 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.

```

src > J Anggotajava > Anggota
1 public class Anggota {
2     private String noKtp; The value of the field Anggota.noKtp is not used
3     private String nama;
4     private int limitPinjaman;
5     private int jumlahPinjaman;
6
7     public Anggota(String noKtp, String nama, int limitPinjaman) {
8         this.noKtp = noKtp;
9         this.nama = nama;
10        this.limitPinjaman = limitPinjaman;
11        this.jumlahPinjaman = 0;
12    }
13
14    public String getNama() {
15        return nama;
16    }
17
18    public int getLimitPinjaman() {
19        return limitPinjaman;
20    }
21
22    public int getJumlahPinjaman() {
23        return jumlahPinjaman;
24    }
25
26    public void pinjam(int jumlah) {
27        if (jumlahPinjaman + jumlah > limitPinjaman) {
28            System.out.println("Maaf, jumlah pinjaman melebihi limit!");
29        } else {
30            jumlahPinjaman += jumlah;
31        }
32    }
33
34    public void angsur(int jumlah) {
35        if (jumlah == jumlahPinjaman) {
36            jumlahPinjaman = 0;
37        } else if (jumlah > jumlahPinjaman) {
38            System.out.println("Maaf, jumlah angsuran melebihi jumlah pinjaman!");
39        } else {
40            jumlahPinjaman -= jumlah;
41        }
42    }
43 }

```

```

Nama Anggota: Donny
Limit Pinjaman: 5000000

Meminjam uang 10.000.000...
Maaf, jumlah pinjaman melebihi limit!
Jumlah pinjaman saat ini: 0

Meminjam uang 4.000.000...
Jumlah pinjaman saat ini: 4000000

Membayar angsuran 1.000.000
Jumlah pinjaman saat ini: 3000000

Membayar angsuran 3.000.000
Jumlah pinjaman saat ini: 0

```

- 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".

```

34 public void angsur(int jumlah) {
35     if (jumlah < jumlahPinjaman * 0.1) {
36         System.out.println("Maaf, angsuran harus 10% dari jumlah pinjaman");
37     } else if (jumlah == jumlahPinjaman) {
38         jumlahPinjaman = 0;
39     } else if (jumlah > jumlahPinjaman) {
40         System.out.println("Maaf, jumlah angsuran melebihi jumlah pinjaman!");
41     } else {
42         jumlahPinjaman -= jumlah;
43     }
44 }

```

```

18 System.out.println("\nMembayar angsuran 50.000");
19 donny.angsur(jumlah:50000);

```

```

Membayar angsuran 50.000
Maaf, angsuran harus 10% dari jumlah pinjaman
Jumlah pinjaman saat ini: 3000000

```

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

```
src > J TestKoperasi.java > TestKoperasi
1  import java.util.Scanner;
2  public class TestKoperasi {
    Run | Debug
3      public static void main(String[] args) {
4          Scanner scanner = new Scanner(System.in);    Resource leak: 'scanner' is never closed
5          System.out.print(s:"Masukkan No KTP: ");
6          String noKtp = scanner.nextLine();
7          System.out.print(s:"Masukkan Nama: ");
8          String nama = scanner.nextLine();
9          System.out.print(s:"Masukkan Limit Pinjaman: ");
10         int limitPinjaman = scanner.nextInt();
11
12         Anggota donny = new Anggota(noKtp, nama, limitPinjaman);
13
14         while (true) {
15             System.out.println(x:"=== Menu Koperasi ===");
16             System.out.println(x:"1. Pinjam");
17             System.out.println(x:"2. Angsur");
18             System.out.println(x:"3. Keluar");
19             System.out.print(s:"Pilih menu (1-3): ");
20             int choice = scanner.nextInt();
21             switch (choice) {
22                 case 3:
23                     break;
24                 case 1:
25                     System.out.print(s:"Masukkan Jumlah Pinjaman: ");
26                     int jumlahPinjam = scanner.nextInt();
27                     donny.pinjam(jumlahPinjam);
28                     System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
29                     break;
30                 case 2:
31                     System.out.print(s:"Masukkan Jumlah Angsuran: ");
32                     int jumlahAngsur = scanner.nextInt();
33                     donny.angsur(jumlahAngsur);
34                     System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
35                     break;
36                 default:
37                     System.out.println(x:"Pilihan tidak valid. Silakan coba lagi.");
38             }
39         }
40     }
41 }
```

```
src > J Anggota.java > Anggota
1 public class Anggota {
2     private String noKtp;    The value of the field Anggota.noKtp is not used
3     private String nama;
4     private int limitPinjaman;
5     private int jumlahPinjaman;
6
7     public Anggota(String noKtp, String nama, int limitPinjaman) {
8         this.noKtp = noKtp;
9         this.nama = nama;
10        this.limitPinjaman = limitPinjaman;
11        this.jumlahPinjaman = 0;
12    }
13
14    public String getNama() {
15        return nama;
16    }
17
18    public int getLimitPinjaman() {
19        return limitPinjaman;
20    }
21
22    public int getJumlahPinjaman() {
23        return jumlahPinjaman;
24    }
25
26    public void pinjam(int jumlah) {
27        if (jumlahPinjaman + jumlah > limitPinjaman) {
28            System.out.println(x:"Maaf, jumlah pinjaman melebihi limit!");
29        } else {
30            jumlahPinjaman += jumlah;
31        }
32    }
33
34    public void angsur(int jumlah) {
35        if (jumlah < jumlahPinjaman * 0.1) {
36            System.out.println(x:"Maaf, angsuran harus 10% dari jumlah pinjaman");
37        } else if (jumlah == jumlahPinjaman) {
38            jumlahPinjaman = 0;
39        } else if (jumlah > jumlahPinjaman) {
40            System.out.println(x:"Maaf, jumlah angsuran melebihi jumlah pinjaman!");
41        } else {
42            jumlahPinjaman -= jumlah;
43        }
44    }
45 }
```



```
Masukkan No KTP: 123
Masukkan Nama: qwe
Masukkan Limit Pinjaman: 1000000
=== Menu Koperasi ===
1. Pinjam
2. Angsur
3. Keluar
Pilih menu (1-3): 1
Masukkan Jumlah Pinjaman: 2000000
Maaf, jumlah pinjaman melebihi limit!
Jumlah pinjaman saat ini: 0
=== Menu Koperasi ===
1. Pinjam
2. Angsur
3. Keluar
Pilih menu (1-3): 2
Masukkan Jumlah Angsuran: 1
Maaf, jumlah angsuran melebihi jumlah pinjaman!
Jumlah pinjaman saat ini: 0
=== Menu Koperasi ===
```