

Object Oriented Programming
Jobsheet 3
Enkapsulasi



BY :

D4 INFORMATICS ENGINEERING (1I)
Shofa Yasmin Fauziah (22)

State Polytechnic of Malang
Soekarno Hatta street No.9, Malang, East Java 65141
2022/2023

1. Percobaan 1 – Enkapsulasi

```
1 package motorencapsulation;
2 public class Motor {
3     public int kecepatan = 0;
4     public boolean kontakOn = false;
5
6     public void printStatus() {
7         if (kontakOn == true) {
8             System.out.println("Kontak On");
9         }
10        else {
11            System.out.println("Kontak Off");
12        }
13        System.out.println("Kecepatan "+kecepatan+"\n");
14    }
15 }
16
```

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

Kontak Off
Kecepatan 0

Kontak Off
Kecepatan 50

2. Percobaan 2 - Access Modifier

```

1 package motorencapsulation;
2 public class Motor {
3     private int kecepatan = 0;
4     private boolean kontakOn = false;
5
6     public void nyalakanMesin() {
7         kontakOn = true;
8     }
9
10    public void matikanMesin() {
11        kontakOn = false;
12        kecepatan = 0;
13    }
14
15    public void tambahKecepatan() {
16        if (kontakOn == true) {
17            kecepatan += 5;
18        }
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        }
28        else{
29            System.out.println(×: "Kecepatan tidak bisa berkurang karena Mesin Off! \n");
30        }
31    }
32
33    public void printStatus() {
34        if (kontakOn == true) {
35            System.out.println(×: "Kontak On");
36        }
37        else{
38            System.out.println(×: "Kontak Off");
39        }
40        System.out.println("Kecepatan "+kecepatan+"\n");
41    }
42 }

```

```

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

```

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

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

Untuk menghindari peringatan tersebut, kita harus memastikan mesin (kontak) telah dinyalakan sebelum menambah kecepatan dengan menyalakan mesin dengan menggunakan method nyalakanMesin() dalam class MotorDemo.

2. Mengapa atribut kecepatan dan kontakOn diset private?
Mengontrol akses ke atribut tersebut dari luar class, untuk memastikan bahwa data dalam objek tidak dapat dengan mudah diubah oleh kode lain
3. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

```
15 public void tambahKecepatan() {  
16     if (kontakOn == true) {  
17         kecepatan += 90;  
18         if (kecepatan > 100) {  
19             System.out.println("Anda mencapai kecepatan maksimal");  
20         }  
21     }  
22     else {  
23         System.out.println("Kecepatan tidak bisa bertambah karena Mesin Off! \n");  
24     }  
25 }
```

Kontak Off
Kecepatan 0

Kecepatan tidak bisa bertambah karena Mesin Off!

Kontak On
Kecepatan 0

Kontak On
Kecepatan 90

Anda mencapai kecepatan maksimal
Kontak On
Kecepatan 180

Anda mencapai kecepatan maksimal
Kontak On
Kecepatan 270

Kontak Off
Kecepatan 0

3. Percobaan 3 - Getter dan Setter

```

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

```

```

1 package kopersigettersetter;
2 public class KoperasiDemo {
3     public static void main(String[] args) {
4         Anggota anggotal = new Anggota();
5         anggotal.setNama(nama: "Iwan Setiawan");
6         anggotal.setAlamat(alamat: "Jalan Sukarno Hatta no 10");
7         anggotal.setor(uang: 100000);
8         System.out.println("Simpanan "+anggotal.getNama()+ " : Rp "+anggotal.getSimpanan());
9
10        anggotal.pinjam(uang: 5000);
11        System.out.println("Simpanan "+anggotal.getNama()+ " : Rp "+anggotal.getSimpanan());
12    }
13 }
14

```

Simpanan Iwan Setiawan : Rp 100000.0

Simpanan Iwan Setiawan : Rp 95000.0

4. Percobaan 4 - Konstruktor, Instansiasi

```

1 package koperasigettersetter;
2 public class KoperasiDemo {
3     public static void main(String[] args) {
4         Anggota anggota = new Anggota( nama: "Iwan", alamat: "Jalan Mawar");
5         System.out.println("Simpanan "+anggota.getNama()+" : Rp "+anggota.getSimpanan());
6
7         anggota.setNama( nama: "Iwan Setiawan");
8         anggota.setAlamat( alamat: "Jalan Sukarno Hatta no 10");
9         anggota.setor( uang: 100000);
10        System.out.println("Simpanan "+anggota.getNama()+" : Rp "+anggota.getSimpanan());
11
12        anggota.pinjam( uang: 5000);
13        System.out.println("Simpanan "+anggota.getNama()+" : Rp "+anggota.getSimpanan());
14    }
15 }

```

```

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

```

5. Pertanyaan – Percobaan 3 dan 4

1. Apa yang dimaksud getter dan setter?
Getter adalah metode yang digunakan untuk mengambil nilai dari sebuah atribut atau properti objek.
Setter adalah metode yang digunakan untuk mengubah nilai dari sebuah atribut atau properti objek.
2. Apa kegunaan dari method getSimpanan()?
Untuk mengambil nilai yang merupakan hasil dari memasukkan nilai uang ke method setor() atau nilai uang ke method pinjam()
3. Method apa yang digunakan untuk menambah saldo?
Method setor()
4. Apa yang dimaksud konstruktor?
Konstruktor adalah method memiliki nama yang sama dengan class yang digunakan untuk menginisialisasi objek dari class tersebut serta memiliki parameter yang digunakan untuk menginisialisasi atribut-atribut objek.
5. Sebutkan aturan dalam membuat konstruktor?
 - Konstruktor harus memiliki nama yang sama dengan class
 - Tidak ada return atau void karena hanya untuk menginisialisasi objek
6. Apakah boleh konstruktor bertipe private?
Boleh
7. Kapan menggunakan parameter dengan passing parameter?

Ketika membutuhkan data tambahan atau tidak akan digunakan lagi pada kontes yang berbeda.

8. Apa perbedaan atribut class dan instansiasi atribut?

Class Attributes

- Mencakup class secara keseluruhan.
- Dideklarasikan di dalam class, tetapi di luar konstruktor menggunakan kata kunci 'static' .

Instance Attributes

- Mencakup objek individu yang dibuat dari class.
- Dideklarasikan di dalam class, tetapi di luar konstruktor tanpa menggunakan kata kunci 'static' lalu diinstansiasi dalam konstruktor.

9. Apa perbedaan class method dan instansiasi method?

Class Method

- Mencakup class secara keseluruhan tidak perlu membuat objek terlebih dahulu untuk menggunakannya.
- Tidak memiliki akses langsung ke atribut atau metode instansiasi.

Instance Method

- Mencakup objek individu yang dibuat dari class yang hanya dapat diakses melalui objek.
- Memiliki akses ke atribut yang dimiliki oleh objek

6. Tugas

1. Cobalah program dibawah ini dan tuliskan hasil outputnya

```
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());
    }
}
```



```
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;
        }
    }
}
```

```

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

```

```

1  package encap;
2
3  public class EncapTest {
4
5      public static void main(String[] args) {
6          EncapDemo encap = new EncapDemo();
7          encap.setName( newName: "James");
8          encap.setAge( newAge: 35);
9
10         System.out.println("Name : "+ encap.getName());
11         System.out.println("Age : "+ encap.getAge());
12     }
13 }
14

```

Name : James

Age : 30

BUILD SUCCESS

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

Karena pada method setAge() terdapat if(newAge > 30){age = 30}; yang berarti jika kita memasukkan nilai newAge lebih dari 30 maka ia akan mencetak age 30 dengan kata lain method ini membatasi nilai attribute age untuk tidak lebih dari 30.

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

```
public void setAge(int newAge){
    if(newAge <= 30 && newAge >= 18 ){
        age = newAge;
    }
    else {
        System.out.println("Umur tidak memenuhi syarat minimal : 18 dan maksimal : 30 ");
    }
}
```

The first screenshot shows the code with the following changes:

```
package encap;

public class EncapTest {

    public static void main(String[] args) {
        EncapDemo encap = new EncapDemo();
        encap.setName("James");
        encap.setAge(18);

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

The output shows:

```
Running NetBeans Compile On Save execution. Phase execution is skipped and output di
Scanning for projects...

-----com.mycompany:Encap >-----
Building Encap 1.0-SNAPSHOT
-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Encap ---
Name : James
Age : 18
BUILD SUCCESS
Total time: 1.234 s
Finished at: 2023-09-18T23:06:57+07:00
```

The second screenshot shows the code with the following changes:

```
package encap;

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());
    }
}
```

The output shows:

```
Running NetBeans Compile On Save execution. Phase execution is skipped and output di
Scanning for projects...

-----com.mycompany:Encap >-----
Building Encap 1.0-SNAPSHOT
-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Encap ---
Umur tidak memenuhi syarat minimal : 18 dan maksimal : 30
Name : James
Age : 0
BUILD SUCCESS
Total time: 1.734 s
Finished at: 2023-09-18T23:08:20+07:00
```

- 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

```
package nomor4;
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("\nMeminjam uang 10.000.000...");
        donny.pinjam(10000000);
        donny.getJumlahPinjaman();
        System.out.println("Jumlah pinjaman saat ini : " + donny.getJumlahPinjaman());

        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());
    }
}
```

```

package nomor4;
public class Anggota {
    String nomorKTP;
    String nama;
    int limitPeminjaman = 0;
    int jumlahPinjaman = 0;

    public Anggota(String nomorKTP, String nama, int limitPeminjaman) {
        this.nomorKTP = nomorKTP;
        this.nama = nama;
        this.limitPeminjaman = limitPeminjaman;
    }

    public void setName(String nama){
        this.nama = nama;
    }
    public void setLimitPeminjaman(int limitPeminjaman){
        this.limitPeminjaman = limitPeminjaman;
    }
    public void setJumlahPinjaman(int jumlahPinjaman){
        this.jumlahPinjaman = jumlahPinjaman;
    }

    public String getName(){
        return nama;
    }
    public int getLimitPeminjaman(){
        return limitPeminjaman;
    }
    public int getJumlahPinjaman(){
        return jumlahPinjaman;
    }

    public void pinjam(int uang){
        if(uang <= limitPeminjaman){
            jumlahPinjaman += uang;
        } else {
            System.out.println("Maaf, jumlah pinjaman melebihi limit.");
        }
    }

    public void angsur(int uang){
        if(uang <= jumlahPinjaman){
            jumlahPinjaman -= uang;
        }
    }
}

```

Output

Nama Anggota :Donny

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

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”.

Code

Anggota Class

```
41
42 public void angsur(int uang) {
43     if(uang == 0.1*jumlahPinjaman){
44         jumlahPinjaman -= uang;
45     }else{
46         System.out.println("Maaf, angsuran harus 10% dari jumlah pinjaman");
47     }
48 }
49 }
50
```

TestKoperasi Class

```
1 package nomor4;
2 public class TestKoperasi {
3     public static void main(String[] args) {
4         Anggota donny = new Anggota(nomorKTP: "111333444", nama: "Donny", limitPeminjaman: 5000000);
5         System.out.println("Nama Anggota : " + donny.getNama());
6
7         System.out.println("Meminjam uang 10.000.000...");
8         donny.pinjam(uang: 10000000);
9         donny.getJumlahPinjaman();
10        System.out.println("Jumlah pinjaman saat ini : " + donny.getJumlahPinjaman());
11
12        System.out.println("Meminjam uang 4.000.000...");
13        donny.pinjam(uang: 4000000);
14        System.out.println("Jumlah pinjaman saat ini : " + donny.getJumlahPinjaman());
15
16        System.out.println("Membayar angsuran 400.000...");
17        donny.angsur(uang: 400000);
18        System.out.println("Jumlah pinjaman saat ini : " + donny.getJumlahPinjaman());
19    }
20 }
```

Output

Nama Anggota :Donny

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 400.000...

Jumlah pinjaman saat ini : 3600000

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

Code

```
1 package nomor4;
2
3 import java.util.Scanner;
4
5 public class TestKoperasi {
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8
9         Anggota donny = new Anggota(nomorKTP: "111333444", nama: "Donny", limitPeminjaman: 5000000);
10        System.out.println("Nama Anggota : " + donny.getNama());
11
12        System.out.print("Meminjam uang: ");
13        int pinjam = sc.nextInt();
14        donny.pinjam(uang: pinjam);
15        donny.getJumlahPinjaman();
16        System.out.println("Jumlah pinjaman saat ini : " + donny.getJumlahPinjaman());
17
18        System.out.print("Membayar angsuran: ");
19        int angsur = sc.nextInt();
20        donny.angsur(uang: angsur);
21        System.out.print("Jumlah pinjaman saat ini : " + donny.getJumlahPinjaman());
22    }
23 }
```

Output

Nama Anggota :Donny

Meminjam uang: 4000000

Jumlah pinjaman saat ini : 4000000

Membayar angsuran: 400000

Jumlah pinjaman saat ini : 3600000