

PEMROGRAMAN BERBASIS OBJEK

Praktikum Enkapsulasi



Lavina/2341760062

SIB 2D

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

POLITEKNIK NEGERI MALANG

2024

Percobaan 1 – Tanpa Enkapsulasi

Motor.java

```
jobsheet3 > Motor.java > Motor
1  package jobsheet3;
2
3  public class Motor {
4      public String platNomor;
5      public boolean isMesinOn;
6      public int kecepatan;
7
8      public void displayStatus() {
9          System.out.println("Plat Nomor: " + this.platNomor);
10
11         if (isMesinOn) {
12             System.out.println(x:"Mesin On");
13         } else {
14             System.out.println(x:"Mesin Off");
15         }
16
17         System.out.println("Kecepatan: " + this.kecepatan);
18         System.out.println(x:"=====");
19     }
20 }
```

MotorDemo.java

```
jobsheet3 > MotorDemo.java > MotorDemo
1  package jobsheet3;
2
3  public class MotorDemo {
4      Run | Debug
5      public static void main(String[] args) {
6          Motor motor1 = new Motor();
7          motor1.displayStatus();
8
9          motor1.platNomor = "B 0838 XZ";
10         motor1.kecepatan = 50;
11         motor1.displayStatus();
12     }
13 }
```

Output:

```
● Plat Nomor: null
Mesin Off
Kecepatan: 0
=====
Plat Nomor: B 0838 XZ
Mesin Off
=====
```

Pada motor1 dengan plat "B 0838 XZ", kecepatannya dapat berubah dari 0 ke 50 padahal mesin motor masih dalam kondisi Off. Bagaimana mungkin atribut kecepatan bernilai 50 padahal mesin masih Off? Hal ini karena belum tersedia kontrol/batasan terhadap atribut kecepatan. Padahal, objek di dunia nyata selalu memiliki batasan dan mekanisme bagaimana objek tersebut dapat digunakan. Misalnya motor yang harus dalam keadaan menyala ketika kecepatan lebih dari 0. Kejanggalan ini juga terjadi pada motor ketiga dengan plat nomor "D 8343 CV".

Hasil modifikasi MotorDemo.java

```
1 package jobsheet3;
2
3 public class MotorDemo {
4     public static void main(String[] args) {
5         Motor motor1 = new Motor();
6         motor1.displayStatus();
7
8         motor1.platNomor = "B 0838 XZ";
9         int kecepatanBaru = 50;
10
11         if (!motor1.isMesinOn && kecepatanBaru > 0) {
12             System.out.println("Kecepatan tidak boleh lebih dari 0 jika mesin off");
13         } else {
14             motor1.kecepatan = kecepatanBaru;
15         }
16
17         motor1.displayStatus();
18
19         Motor motor2 = new Motor();
20         motor2.platNomor = "N 9840 AB";
21         motor2.isMesinOn = true;
22         kecepatanBaru = 40;
23
24         if (!motor2.isMesinOn && kecepatanBaru > 0) {
25             System.out.println("Kecepatan tidak boleh lebih dari 0 jika mesin off");
26         } else {
27             motor2.kecepatan = kecepatanBaru;
28         }
29
30         motor2.displayStatus();
31
32         Motor motor3 = new Motor();
33         motor3.platNomor = "D 8343 CV";
34         kecepatanBaru = 60;
35
36         if (!motor3.isMesinOn && kecepatanBaru > 0) {
37             System.out.println("Kecepatan tidak boleh lebih dari 0 jika mesin off");
38         } else {
39             motor1.kecepatan = kecepatanBaru;
40         }
41
42         motor3.displayStatus();
43     }
44 }
```

Ouput:

```
● Plat Nomor: null
Mesin Off
Kecepatan: 0
=====
Kecepatan tidak boleh lebih dari 0 jika mesin off
Plat Nomor: B 0838 XZ
Mesin Off
Kecepatan: 0
=====
Plat Nomor: N 9840 AB
Mesin On
Kecepatan: 40
=====
Kecepatan tidak boleh lebih dari 0 jika mesin off
Plat Nomor: D 8343 CV
Mesin Off
Kecepatan: 0
=====
```

Percobaan 2 – Enkapsulasi

Motor.java

```
1  package jobsheet3;
2
3  public class Motor {
4      private String platNomor;
5      private boolean isMesinOn;
6      private int kecepatan;
7
8      public String getPlatNomor() {
9          return platNomor;
10     }
11
12     public void setPlatNomor(String platNomor) {
13         this.platNomor = platNomor;
14     }
15
16     public boolean isMesinOn() {
17         return isMesinOn;
18     }
19
20     public void setMesinOn(boolean mesinOn) {
21         isMesinOn = mesinOn;
22     }
23
24     public int getKecepatan() {
25         return kecepatan;
26     }
27
28     public void setKecepatan(int kecepatan) {
29         if (!this.isMesinOn && kecepatan > 0) {
30             System.out.println("Kecepatan tidak boleh lebih dari 0 jika mesin off");
31         } else {
32             this.kecepatan = kecepatan;
33         }
34     }
35 }
```

```

36 public void displayStatus() {
37     System.out.println("Plat Nomor: " + this.platNomor);
38
39     if (isMesinOn) {
40         System.out.println("Mesin On");
41     } else {
42         System.out.println("Mesin Off");
43     }
44
45     System.out.println("Kecepatan: " + this.kecepatan);
46     System.out.println("=====");
47 }
48 }

```

MotorDemo.java

```

jobsheet3 > MotorDemo.java > MotorDemo
1 package jobsheet3;
2
3 public class MotorDemo {
4     Run | Debug
5     public static void main(String[] args) {
6         Motor motor1 = new Motor();
7         motor1.displayStatus();
8
9         motor1.setPlatNomor(platNomor:"B 0838 XZ");
10        motor1.setKecepatan(kecepatan:50);
11        motor1.displayStatus();
12
13        Motor motor2 = new Motor();
14        motor2.setPlatNomor(platNomor:"N 9840 AB");
15        motor2.setMesinOn(mesinOn:true);
16        motor2.setKecepatan(kecepatan:40);
17        motor2.displayStatus();
18
19        Motor motor3 = new Motor();
20        motor3.setPlatNomor(platNomor:"D 8343 CV");
21        motor3.setKecepatan(kecepatan:60);
22        motor3.displayStatus();
23    }

```

Output:

```

Plat Nomor: null
Mesin Off
Kecepatan: 0
=====
Kecepatan tidak boleh lebih dari 0 jika mesin off
Plat Nomor: B 0838 XZ
Mesin Off
Kecepatan: 0
=====
Plat Nomor: N 9840 AB
Mesin On
Kecepatan: 40
=====
Kecepatan tidak boleh lebih dari 0 jika mesin off
Plat Nomor: D 8343 CV
Mesin Off
Kecepatan: 0
=====

```

Pertanyaan

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

Jawab: Karena sudah diatur *conditional statementnya* pada method setKecepatan() dimana jika !isMesinOn (mesin off) bernilai true, maka akan muncul peringatan tersebut.

2. Mengapa atribut merek, kecepatan, dan statusMesin diset private?

Jawab: Dibat private agar tidak dinisialisasi sembarangan dari class lain, harus mengikuti alur algoritma yang sudah dibuat di method-method yang ada di class Motor yang mana hanya di class tersebut atribut-atribut private diatas bisa diakses.

3. Apa fungsi dari setter dan getter?

Jawab: Fungsi setter berguna untuk mengisi/menginisialisasi value dari atribut private sedangkan getter berfungsi untuk mendapatkan/mengakses value tersebut.

4. Ubah class Motor sehingga kecepatan maksimalnya adalah 100.

Jawab: Ditambahkan conditional statementnya seperti berikut:

```
public void setKecepatan(int kecepatan) {  
    if (!this.isMesinOn) {  
        System.out.println("Kecepatan tidak boleh lebih dari 0 jika mesin off");  
    } else if (kecepatan > 100) {  
        System.out.println("Kecepatan tidak boleh lebih dari 100");  
    } else {  
        this.kecepatan = kecepatan;  
    }  
}
```

Lalu, ketika kecepatannya diset lebih dari 100:

```
motor2.setKecepatan(140);
```

Hasilnya :

```
Kecepatan tidak boleh lebih dari 100  
Plat Nomor: N 9840 AB  
Mesin On  
Kecepatan: 0  
=====
```

5. Ubah class Motor sehingga kecepatan nya tidak boleh nilai negatif

Jawab: Ditambahkan conditional statementnya seperti berikut:

```
public void setKecepatan(int kecepatan) {
    if (!this.isMesinOn) {
        System.out.println("Kecepatan tidak boleh lebih dari 0 jika mesin off");
    } else if (kecepatan > 100) {
        System.out.println("Kecepatan tidak boleh lebih dari 100");
    } else if (kecepatan < 0) {
        System.out.println("Kecepatan tidak boleh kurang dari 0");
    } else {
        this.kecepatan = kecepatan;
    }
}
```

Jika kecepatannya dibuat negatif:

```
motor2.setKecepatan(-140);
```

Hasilnya:

```
Kecepatan tidak boleh kurang dari 0
Plat Nomor: N 9840 AB
Mesin On
Kecepatan: 0
=====
```

Percobaan 3 – Constructor

User.java

```
4 public class User {
5     public String username;
6     public String nama;
7     public String email;
8     public String alamat;
9     public String pekerjaan;
10
11     public User(String username, String nama, String email) {
12         this.username = username;
13         this.nama = nama;
14         this.email = email;
15     }
16
17     public void cetakInfor() {
18         System.out.println("Username: " + this.username);
19         System.out.println("Nama: " + this.nama);
20         System.out.println("Email: " + this.email);
21         System.out.println("Alamat: " + this.alamat);
22         System.out.println("Pekerjaan: " + this.pekerjaan);
23         System.out.println(x:"=====");
24     }
25 }
```

DemoUser.java

```
4 public class DemoUser {  
5  
6     Run | Debug  
7     public static void main(String[] args) {  
8         User user1 = new User(username:"lavina", nama:"lavina", email:"lavina@gmail.com");  
9         user1.cetakInfor();  
10    }
```

Output:

```
Username: lavina  
Nama: lavina  
Email: lavina@gmail.com  
Alamat: null  
Pekerjaan: null  
=====
```

Pertanyaan

1. Apa yang dimaksud constructor?

Jawab: constructor biasa digunakan jika kita ingin membuat banyak objek dengan tipe dan atribut yang sama sehingga mempermudah penginisialisasian atribut.

2. Sebutkan aturan dalam membuat constructor?

Jawab: Best practicenya, constructor dibuat dengan *upper-case first latter*, constructor bisa dibuat lebih dari satu tetapi tidak boleh memiliki parameter yang sama.

3. Lakukan analisa dan buat kesimpulan apakah constructor bisa bertipe private?

Jawab: Jika constructor dibuat private, maka constructor tersebut jadi tidak bisa digunakan, karena parameternya jadi tidak visible saat digunakan di class lain sedangkan saat instansiasi objek harus menyertakan constructor.

Tugas

1. 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 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 constructor sesuai dengan kebutuhan. Uji dengan TestKoperasi berikut ini untuk memeriksa apakah class Anggota yang anda buat telah sesuai dengan yang diharapkan. Perhatikan

bahwa nilai atribut pinjaman tidak dapat diubah secara random dari luar class, tetapi hanya dapat diubah melalui method pinjam() dan angsur().

```
public class TestKoperasi
{
    public static void main(String[] args)
    {
        Anggota anggota1 = new Anggota("111333444", "Donny", 5000000);

        System.out.println("Nama Anggota: " + anggota1.getNama());
        System.out.println("Limit Pinjaman: " + anggota1.getLimitPinjaman());

        System.out.println("\nMeminjam uang 10.000.000..");
        anggota1.pinjam(10000000);
        System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());

        System.out.println("\nMeminjam uang 4.000.000..");
        anggota1.pinjam(4000000);
        System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());

        System.out.println("\nMembayar angsuran 1.000.000");
        anggota1.angsur(1000000);
        System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());

        System.out.println("\nMembayar angsuran 3.000.000");
        anggota1.angsur(3000000);
        System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());
    }
}
```

Anggota.java

```
4 public class Anggota {
5     private String noKTP;
6     private String nama;
7     private int limitPinjaman;
8     private int pinjaman;
9
10    public Anggota(String noKTP, String nama, int limitPinjaman) {
11        this.noKTP = noKTP;
12        this.nama = nama;
13        this.limitPinjaman = limitPinjaman;
14    }
15
16    public String getNama() {
17        return nama;
18    }
19
20    public int getLimitPinjaman() {
21        return limitPinjaman;
22    }
23
24    public int getJumlahPinjaman() {
25        return this.pinjaman;
26    }
27
28    public void pinjam(int pinjaman) {
29        if (pinjaman > this.limitPinjaman) {
30            System.out.println("Maaf, jumlah pinjaman melebihi limit.");
31        } else {
```

```

32     this.pinjaman = pinjaman;
33     this.limitPinjaman -= pinjaman;
34 }
35 }
36
37 public void angsur(int angsuran) {
38     this.pinjaman -= angsuran;
39     this.limitPinjaman += angsuran;
40 }
41 }

```

TestKoperasi.java

```

4 public class TestKoperasi {
5
6     Run | Debug
7     public static void main(String[] args) {
8         Anggota anggota1 = new Anggota(noKTP:"1234567890", nama:"Lavina", limitPinjaman:5_000_000);
9
10        System.out.println("Nama Anggota: " + anggota1.getNama());
11        System.out.println("Limit Pinjaman: " + anggota1.getLimitPinjaman());
12
13        System.out.println(x:"\nMeminjam uang 10.000.000 ... ");
14        anggota1.pinjam(pinjaman:10_000_000);
15
16        System.out.println(x:"\nMeminjam 4.000.000 ... ");
17        anggota1.pinjam(pinjaman:4_000_000);
18        System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());
19
20        System.out.println(x:"\nMembayar angsuran 1.000.000 ... ");
21        anggota1.angsur(angsuran:1_000_000);
22        System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());
23
24        System.out.println(x:"\nMembayar angsuran 3.000.000 ... ");
25        anggota1.angsur(angsuran:3_000_000);
26        System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());
27    }
28 }

```

Output:

```

• Nama Anggota: Lavina
  Limit Pinjaman: 5000000

  Meminjam uang 10.000.000...
  Maaf, jumlah pinjaman melebihi limit.

  Meminjam 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

```

2. Modifikasi class Anggota 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"

Hasil modifikasi method angsur() :

```
public void angsur(int angsuran) {  
    if (angsuran ≥ this.pinjaman * 0.1) {  
        this.pinjaman -= angsuran;  
        this.limitPinjaman += angsuran;  
    } else {  
        System.out.println(x:"Maaf, angsuran harus 10% dari jumlah pinjaman");  
    }  
}
```

Untuk validasi, dibuat angsurannya menjadi hanya 10.000 :

```
System.out.println(x:"\nMembayar angsuran 10.000 ... ");  
anggota1.angsur(angsuran:10_000);  
System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());
```

Hasilnya :

```
Nama Anggota: Lavina  
Limit Pinjaman: 5000000  
  
Meminjam uang 10.000.000...  
Maaf, jumlah pinjaman melebihi limit.  
  
Meminjam 4.000.000...  
Jumlah pinjaman saat ini: 4000000  
  
Membayar angsuran 1.000.000...  
Jumlah pinjaman saat ini: 3000000  
  
Membayar angsuran 10.000...  
Maaf, angsuran harus 10% dari jumlah pinjaman  
Jumlah pinjaman saat ini: 3000000
```