

Jobsheet 3 – Enkapsulasi Pemrograman Web



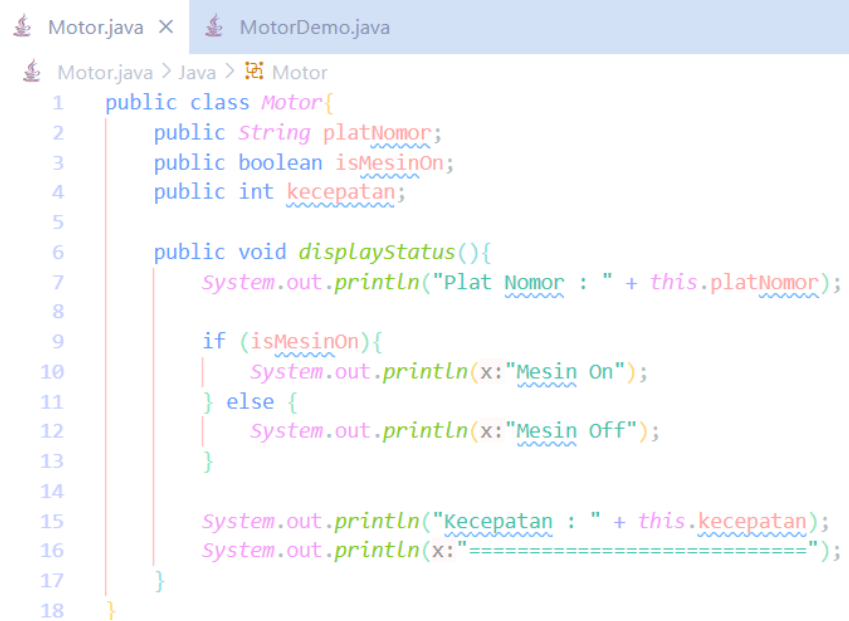
244107060144

Maria Savira

D-IV Sistem Informasi Bisnis / 2C

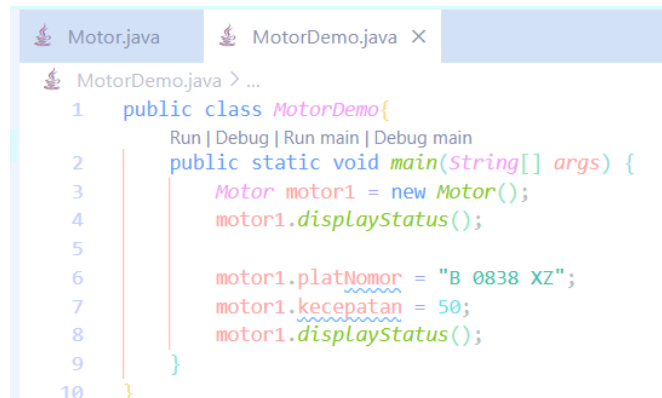
Percobaan 1

3. Membuat class Motor.java



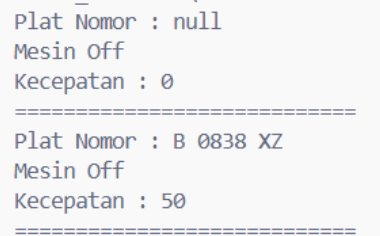
```
Motor.java X MotorDemo.java
Motor.java > Java > Motor
1 public class Motor{
2     public String platNomor;
3     public boolean isMesinOn;
4     public int kecepatan;
5
6     public void displayStatus(){
7         System.out.println("Plat Nomor : " + this.platNomor);
8
9         if (isMesinOn){
10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

4. Membuat class MotorDemo.java



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

5. Hasil output :



```
-----
Plat Nomor : null
Mesin Off
Kecepatan : 0
=====
Plat Nomor : B 0838 XZ
Mesin Off
Kecepatan : 50
=====
```

6. Menambahkan 2 objek baru :

```
Motor.java  MotorDemo.java X
MotorDemo.java > ...
1
2 public class MotorDemo {
3
4     Run | Debug | Run main | Debug main
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        Motor motor2 = new Motor();
14        motor2.platNomor = "N 9840 AB";
15        motor2.isMesinOn = true;
16        motor2.kecepatan = 40;
17        motor2.displayStatus();
18
19        Motor motor3 = new Motor();
20        motor3.platNomor = "D 8343 CV";
21        motor3.kecepatan = 60;
22        motor3.displayStatus();
23    }
```

7. Output

```
Plat Nomor : null
Mesin Off
Kecepatan : 0
=====
Plat Nomor : B 0838 XZ
Mesin Off
Kecepatan : 50
=====
Plat Nomor : N 9840 AB
Mesin On
Kecepatan : 40
=====
Plat Nomor : D 8343 CV
Mesin Off
Kecepatan : 60
=====
```

9. Menambahkan validasi

```
Run | Debug | Run main | Debug main
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(x:"Kecepatan tidak boleh lebih dari 0 jika mesin off");
13     } else {
14         motor1.kecepatan = kecepatanBaru;
15     }
16
17     motor1.displayStatus();
18 }
```

10. Menambahkan validasi untuk 2 objek lainnya

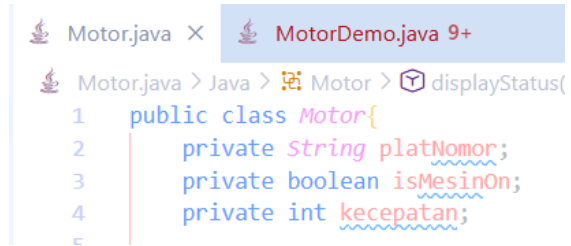
```
19     Motor motor2 = new Motor();
20     motor2.platNomor = "N 9840 AB";
21     motor2.isMesinOn = true;
22     kecepatanBaru = 60;
23
24     if (!motor2.isMesinOn && kecepatanBaru > 0) {
25         System.out.println(x: "Kecepatan tidak boleh lebih dari 0 jika mesin off");
26     } else {
27         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(x: "Kecepatan tidak boleh lebih dari 0 jika mesin off");
38     } else {
39         motor3.kecepatan = kecepatanBaru;
40     }
41
42     motor3.displayStatus();
43 }
44 }
```

11. 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
=====
```

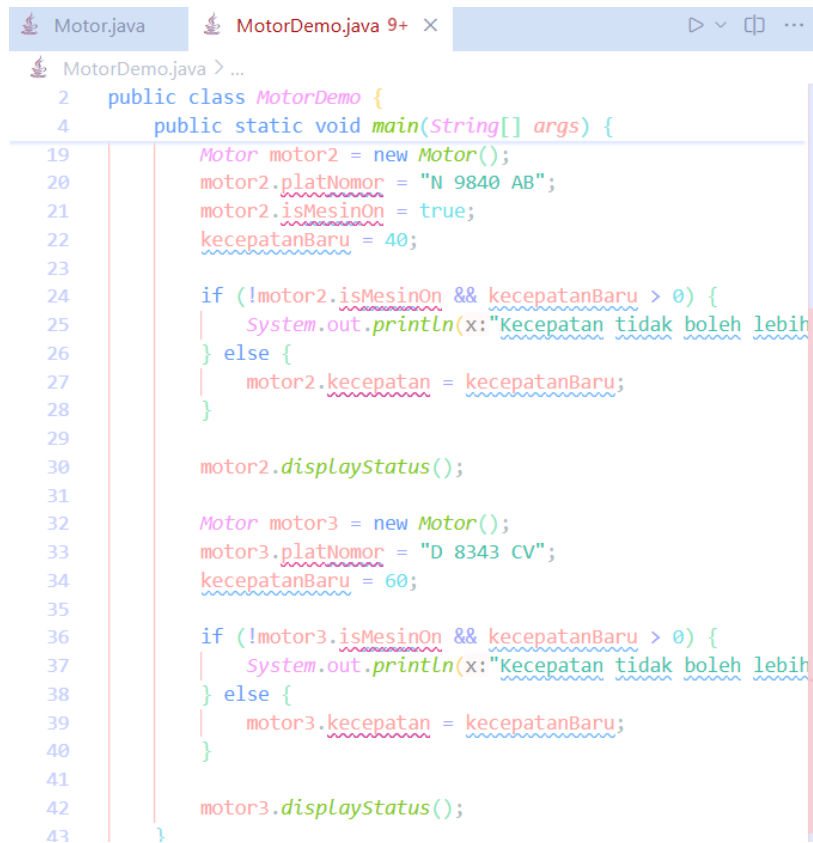
Percobaan 2

3. Mengubah atribut public menjadi private



```
Motor.java X MotorDemo.java 9+
Motor.java > Java > Motor > displayStatus(
1 public class Motor{
2     private String platNomor;
3     private boolean isMesinOn;
4     private int kecepatan;
5 }
```

4. Error dalam MotorDemo.java karena sekarang atribut menjadi private



```
Motor.java MotorDemo.java 9+ X
MotorDemo.java > ...
2 public class MotorDemo {
4     public static void main(String[] args) {
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(x:"Kecepatan tidak boleh lebih
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(x:"Kecepatan tidak boleh lebih
38         } else {
39             motor3.kecepatan = kecepatanBaru;
40         }
41
42         motor3.displayStatus();
43     }
```

5. Menambahkan setter dan getter dalam class Motor.java

```
6      public String getPlatNomor(){
7          |   return platNomor;
8      }
9
10     public boolean isMesinOn(){
11         |   return isMesinOn;
12     }
13
14     public int getKecepatan(){
15         |   return kecepatan;
16     }
17
18     public void setIsMesinOn(boolean isMesinOn){
19         |   this.isMesinOn = isMesinOn;
20     }
21
22     public void setKecepatan(int kecepatan){
23         |   this.kecepatan = kecepatan;
24     }
25
26     public void setPlatNomor(String platNomor){
27         |   this.platNomor = platNomor;
28     }
```

6. Menyesuaikan getter dan setter dalam class MotorDemo.java

```
Run | Debug | Run main | Debug main
public static void main(String[] args) {
    Motor motor1 = new Motor();
    motor1.displayStatus();

    motor1.setPlatNomor(platNomor:"B 0838 XZ");
    motor1.setKecepatan(kecepatan:50);
    motor1.displayStatus();

    Motor motor2 = new Motor();
    motor2.setPlatNomor(platNomor:"N 9840 AB");
    motor2.setIsMesinOn(isMesinOn:true);
    motor2.setKecepatan(kecepatan:40);
    motor2.displayStatus();

    Motor motor3 = new Motor();
    motor3.setPlatNomor(platNomor:"D 8343 CV");
    motor3.setKecepatan(kecepatan:60);
    motor3.displayStatus();
}
```

7. Menambahkan validasi untuk kecepatan pada method setKecepatan class Motor.java

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

8. 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!"?
2. Mengapa atribut merek, kecepatan, dan statusMesin diset private?
3. Apa fungsi dari setter dan getter?
4. Ubah class Motor sehingga kecepatan maksimalnya adalah 100
5. Ubah class Motor sehingga kecepatan nya tidak boleh nilai negatif

Jawaban

1. Karena apabila value dari isMesinOn mulanya adalah false, maka yang didapat adalah pesan error tersebut jika menambahkan kecepatan. Apabila ingin menambah kecepatan, value dari isMesinOn harus true.
2. Beberapa atribut tersebut dilakukan enkapsulasi agar objek terlindungi dan tidak bisa sembarangan diubah. Sehingga, kontrol akses terhadap atribut hanya dapat melalui Getter dan Setter saja. Selain itu, dengan Private maka suatu objek dapat menampilkan apa saja yang perlu diketahui, tanpa membocorkan detail internal.
3. Setter berguna untuk mengubah atau set suatu value dari atribut Class yang didapat dari Getter. Sedangkan Getter berguna untuk mengambil value dari atribut Class tersebut.
4. Menambahkan validasi untuk maksimal dan minimal kecepatan dalam Class Motor:

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

Perubahan atribut objek dalam Class MotorDemo :

```
2 public class MotorDemo {
3
4     Run | Debug | Run main | Debug main
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.setIsMesinOn(isMesinOn:true);
11        motor1.setKecepatan(kecepatan:110);
12        motor1.displayStatus();
13
14        Motor motor2 = new Motor();
15        motor2.setPlatNomor(platNomor:"N 9840 AB");
16        motor2.setIsMesinOn(isMesinOn:true);
17        motor2.setKecepatan(-2);
18        motor2.displayStatus();
19
20        Motor motor3 = new Motor();
21        motor3.setPlatNomor(platNomor:"D 8343 CV");
22        motor3.setKecepatan(kecepatan:60);
23        motor3.displayStatus();
24    }
```

Outputnya :

```
Plat Nomor : null
Mesin Off
Kecepatan : 0
=====
Maksimal kecepatan adalah 100!
Plat Nomor : B 0838 XZ
Mesin On
Kecepatan : 0
=====
Kecepatan tidak boleh negatif!
Plat Nomor : N 9840 AB
Mesin On
Kecepatan : 0
=====
Kecepatan tidak boleh lebih dari 0 jika mesin off
Plat Nomor : D 8343 CV
Mesin Off
Kecepatan : 0
=====
```


Percobaan 3

1. Membuat class baru bernama User dan membuat constructor

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

2. Error dalam menginisialisasi Class User dikarenakan constructor default sudah tidak berlaku jika telah dibuat constructor baru yang memiliki parameter.

```
Motor.java User.java DemoUser.java 3 x
DemoUser.java > Language Support for Java(TM) by Red Hat >
1 public class DemoUser {
2     Run | Debug | Run main | Debug main
3     public static void main(String[] args) {
4         User user1 = new User();
5     }
6 }
```

3. Menginstansiasi objek baru secara benar dalam Class DemoUser

```
1 public class DemoUser {
2     Run | Debug | Run main | Debug main
3     public static void main(String[] args) {
4         User user1 = new User(username:"annisa.nadya", nama:"Annisa Nadya", email:"annisa.nadya@gmail.com");
5         user1.cetakInfo();
6     }
7 }
```

4. Output

```
Username : annisa.nadya
Nama : Annisa Nadya
Email : annisa.nadya@gmail.com
Alamat : null
Pekerjaan : null
=====
```

Pertanyaan

1. Apa yang dimaksud constructor?
2. Sebutkan aturan dalam membuat constructor?
3. Lakukan analisa dan buat kesimpulan apakah constructor bisa bertipe private?

Jawaban

1. Constructor adalah method yang digunakan saat pembuatan atau instansiasi object baru.
2. Saat Class baru dibuat, Class tersebut secara otomatis memiliki Constructor default tanpa parameter. Selain itu, terdapat beberapa aturan dalam membuat Constructor :
 - Nama Constructor harus sama dengan nama Class.
 - Constructor tidak boleh memiliki return type.
 - Constructor dapat memiliki parameter.
 - Dalam Constructor, boleh menggunakan this untuk instansiasi atribut agar dapat membedakan Class Variable dengan parameter.
3. Constructor dapat dibuat dengan tipe private. Jika Constructor bertipe private, maka objek tidak akan bisa dibuat secara langsung di luar kelas.

Tugas

Class Anggota.java

```
Motor.java User.java DemoUser.java Anggota.java 4 X TestKop
Anggota.java > Java > Anggota
1 public class Anggota {
2     private String nomorKTP;
3     private String nama;
4     private int limitPinjaman = 5000000;
5     private int jumlahPinjaman = 0;
6
7     public Anggota(String nomorKTP, String nama, int jumlahPinjaman){
8         this.nomorKTP = nomorKTP;
9         this.nama = nama;
10        this.limitPinjaman = limitPinjaman;
11    }
12    public String getNomorKTP(){
13        return nomorKTP;
14    }
15
16    public String getNama(){
17        return nama;
18    }
19
20    public int getLimitPinjaman(){
21        return limitPinjaman;
22    }
23
24    public void pinjam(int temp){
25        if (temp > limitPinjaman){
26            System.out.println(x:"Maaf, jumlah pinjaman melebihi limit.
27        } else {
28            jumlahPinjaman += temp;
29        }
30    }
31
32    public int angsur(int temp){
33        return jumlahPinjaman -= temp;
34    }
35
36    public int getJumlahPinjaman(){
37        return jumlahPinjaman;
38    }
39
```

Class TestKoperasi.java

```
1 public class TestKoperasi {  
    Run | Debug | Run main | Debug main  
2     public static void main(String[] args) {  
3         Anggota anggota1 = new Anggota(nomorKTP:"111333444", nama:"Donny", jumlahPinjaman:5000000);  
4  
5         System.out.println("Nama Anggota: " + anggota1.getNama());  
6         System.out.println("Limit Pinjaman: " + anggota1.getLimitPinjaman());  
7  
8         System.out.println(x:"\nMeminjam uang 10.000.000...");  
9         anggota1.pinjam(temp:10000000);  
10        System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());  
11  
12        System.out.println(x:"\nMeminjam uang 4.000.000...");  
13        anggota1.pinjam(temp:4000000);  
14        System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());  
15  
16        System.out.println(x:"\nMembayar angsuran 1.000.000");  
17        anggota1.angsur(temp:1000000);  
18        System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());  
19  
20        System.out.println(x:"\nMembayar angsuran 3.000.000");  
21        anggota1.angsur(temp:3000000);  
22        System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());  
23    }  
24 }
```

Output

```
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
```