

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERBASIS OBJEK**

**PRAKTIKUM W03
Enkapsulasi pada Pemrograman Berorientasi Objek**



**Oleh:
JESSICA AMELIA
2341760185
SIB 2F/15**

**PROGRAM STUDI SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2023/2024**

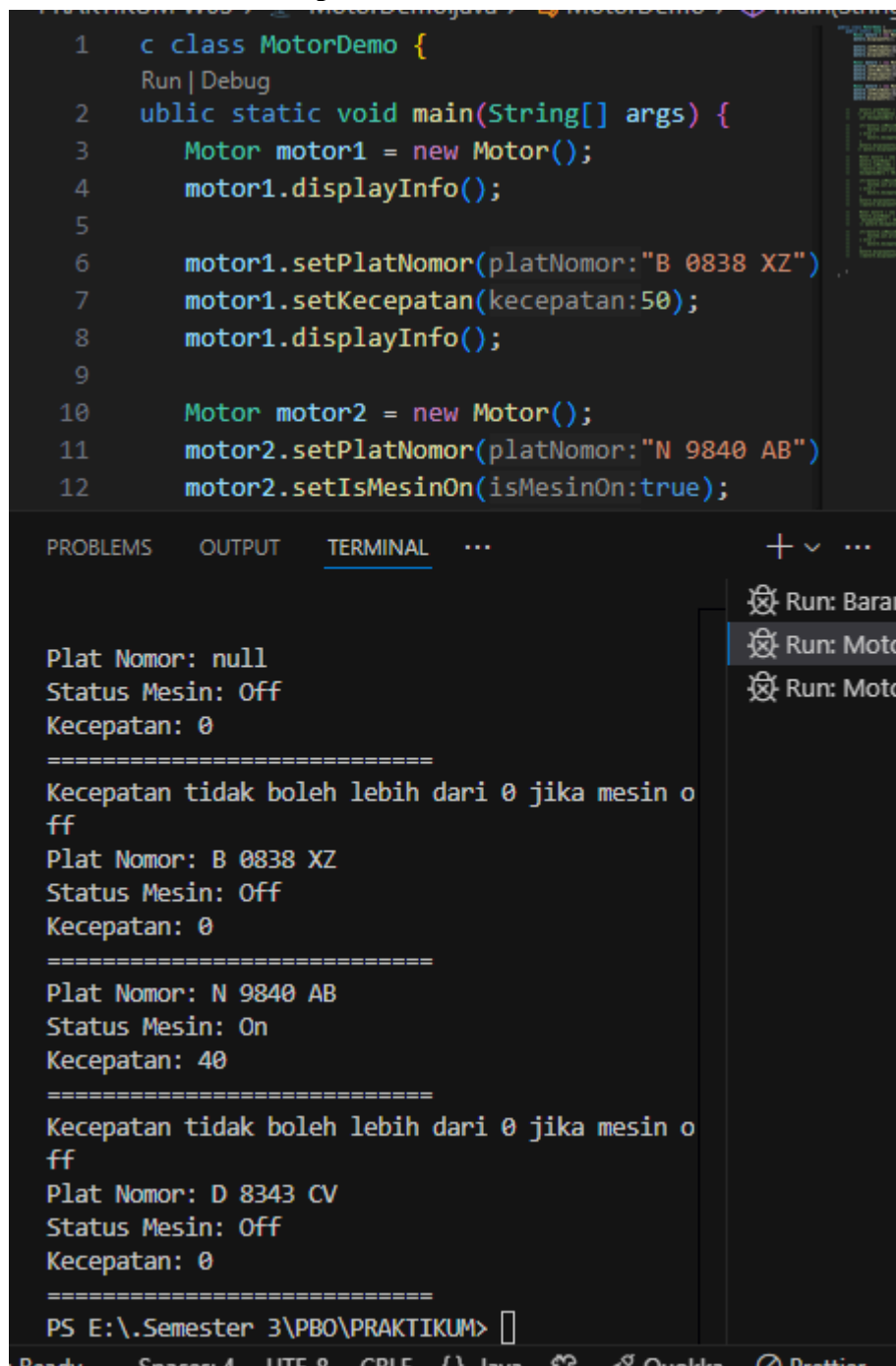
3.1 Percobaan 1

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

```
Plat Nomor: B 0838 XZ
Status Mesin: Off
Kecepatan: 50
=====
Plat Nomor: N 9840 AB
Status Mesin: On
Kecepatan: 40
=====
Plat Nomor: D 8343 CV
Status Mesin: Off
Kecepatan: 60
=====
PS E:\.Semester 3\PBO\PRAKTIKUM> █
```

```
ab\rednat.java\jdt_ws\PRAKTIKUM_Suecc000\bin
'MotorDemo'
Plat Nomor: null
Status Mesin: Off
Kecepatan: 0
=====
Kecepatan tidak boleh lebih dari 0 jika mesin
off
Plat Nomor: B 0838 XZ
Status Mesin: Off
Kecepatan: 0
=====
Plat Nomor: N 9840 AB
Status Mesin: On
Kecepatan: 40
=====
Kecepatan tidak boleh lebih dari 0 jika mesin off
Plat Nomor: D 8343 CV
Status Mesin: Off
Kecepatan: 0
=====
PS E:\.Semester 3\PBO\PRAKTIKUM> █
```

3.2 Percobaan 2 - Enkapsulasi



```
1  c class MotorDemo {
    Run | Debug
2  public static void main(String[] args) {
3      Motor motor1 = new Motor();
4      motor1.displayInfo();
5
6      motor1.setPlatNomor(platNomor:"B 0838 XZ")
7      motor1.setKecepatan(kecepatan:50);
8      motor1.displayInfo();
9
10     Motor motor2 = new Motor();
11     motor2.setPlatNomor(platNomor:"N 9840 AB")
12     motor2.setIsMesinOn(isMesinOn:true);
```

PROBLEMS OUTPUT **TERMINAL** ...

Plat Nomor: null
Status Mesin: Off
Kecepatan: 0
=====

Kecepatan tidak boleh lebih dari 0 jika mesin o
ff
Plat Nomor: B 0838 XZ
Status Mesin: Off
Kecepatan: 0
=====

Plat Nomor: N 9840 AB
Status Mesin: On
Kecepatan: 40
=====

Kecepatan tidak boleh lebih dari 0 jika mesin o
ff
Plat Nomor: D 8343 CV
Status Mesin: Off
Kecepatan: 0
=====

PS E:\.Semester 3\PBO\PRAKTIKUM> |

3.3 Pertanyaan

1. Pada class MotorDemo, saat kita menambah kecepatan untuk pertama kalinya, mengapa muncul peringatan “Kecepatan tidak bisa bertambah karena Mesin Off!”?
Jawab : Karena pada method setKecepatan dilakukan validasi nilai kecepatan terhadap status mesin yaitu dengan if else dimana kondisi if (!isMesinOn dan kecepatan lebih dari 0) terpenuhi maka muncul peringatan tersebut, jika kondisi tidak terpenuhi maka kecepatan dapat bertambah dan tandanya mesin on.
2. Mengapa atribut merek, kecepatan, dan statusMesin sebaiknya diset private?

Jawab : Karena dengan mengeset privat atribut tidak dapat diakses di luar kelas. Nah dengan begitu dibutuhkan setter dan getter untuk setiap atribut tersebut agar tidak muncul error

3. Apa fungsi dari setter dan getter?

Jawab : Fungsi setter dan getter untuk memanipulasi dan mengakses nilai atribut, sehingga apabila ada perubahan requirement pada atribut di setiap objek tidak perlu melakukan perubahan berulang kali, cukup melakukan perubahan pada setter ataupun getter.

4. Ubah class Motor sehingga kecepatan maksimalnya adalah 100

Jawab :

```
3     public void setKecepatan(int kecepatan){
4         if (!this.isMesinOn && kecepatan > 0){
5             System.out.println(x:"Kecepatan tidak boleh lebih dari 0 jika mesin off");
6         } else if(kecepatan > 100){
7             System.out.println(x:"Kecepatan tidak boleh lebih dari 100");
8             this.kecepatan = 100; //set otomatis 100 jika melebihi
9         } else{
10            this.kecepatan = kecepatan;
11        }
12    }
13 }
```

```
1  public class MotorDemo {
2      public static void main(String[] args) {
11         Motor2.setPlatNomor(platNomor:"N 9840 AB");
12         motor2.setIsMesinOn(isMesinOn:true);
13         motor2.setKecepatan(kecepatan:40);
14         motor2.displayInfo();
15
16         Motor motor3 = new Motor();
17         motor3.setPlatNomor(platNomor:"D 8343 CV");
18         motor3.setIsMesinOn(isMesinOn:true);
19         motor3.setKecepatan(kecepatan:120);
20         motor3.displayInfo();
21     }
22 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

WINDOWS 10\AppData\Roaming\Code\User\workspace5
torage\04e323270ba3b92a998912b8ce16e0a8\redhat.
java\jdt_ws\PRAKTIKUM_5decc600\bin' 'MotorDemo'
Plat Nomor: null
Status Mesin: Off
Kecepatan: 0
=====

Kecepatan tidak boleh lebih dari 0 jika mesin off
Plat Nomor: B 0838 XZ
Status Mesin: Off
Kecepatan: 0
=====

Plat Nomor: N 9840 AB
Status Mesin: On
Kecepatan: 40
=====

Kecepatan tidak boleh lebih dari 100
Plat Nomor: D 8343 CV
Status Mesin: On
Kecepatan: 100
=====

PS E:\Semester 3\PBO\PRAKTIKUM>

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

```

33     public void setKecepatan(int kecepatan){
34         if (kecepatan < 0 ){
35             System.out.println(x:"Kecepatan tidak boleh bernilai negatif");
36         } else if(!this.isMesinOn && kecepatan >0){
37             System.out.println(x:"Kecepatan tidak boleh lebih dari 0 jika mesin off");
38         }else if(kecepatan > 100){
39             System.out.println(x:"Kecepatan tidak boleh lebih dari 100");
40             this.kecepatan = 100; //set otomatis 100 jika melebihi
41         } else{
42             this.kecepatan = kecepatan;
43         }
44     }
45 }
46

```

```

20     motor3.displayInfo();
21
22     Motor motor4 = new Motor();
23     motor4.setPlatNomor(platNomor:"AB 7353 TG");
24     motor4.setKecepatan(-20);
25     motor4.displayInfo();
--

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

Kecepatan tidak boleh lebih dari 0 jika mesin off
Plat Nomor: B 0838 XZ
Status Mesin: Off
Kecepatan: 0
=====
Plat Nomor: N 9840 AB
Status Mesin: On
Kecepatan: 40
=====
Kecepatan tidak boleh lebih dari 100
Plat Nomor: D 8343 CV
Status Mesin: On
Kecepatan: 100
=====
Kecepatan tidak boleh bernilai negatif
Plat Nomor: AB 7353 TG
Status Mesin: Off
Kecepatan: 0
=====
PS E:.\Semester 3\PBO\PRAKTIKUM>

```

nadya

3.4 Percobaan 3 - Constructor

```

\jdt_ws\PRAKTIKUM_5decc600\bin' 'DemoUse
r'
Username: annisa.nadya
Nama: Anisa Nadyanull
Email: annisa.nadya@gmail.com
Alamat: null
Pekerjaan: null
=====
PS E:.\Semester 3\PBO\PRAKTIKUM>

```

3.5 Pertanyaan

1. Apa yang dimaksud constructor?

Jawab : Constructor adalah method istimewa yang digunakan untuk melakukan instansiasi objek atau membuat objek baru.

2. Sebutkan aturan dalam membuat constructor

Jawab : Nama constructor harus sama dengan nama class, tidak memiliki return type termasuk void, constructor dapat memiliki parameter.

3. Lakukan analisa dan buat kesimpulan apakah constructor bisa memiliki access level modifier private?

Jawab : Bisa saja , constructor bisa menggunakan private untuk membatasi akses. Ini berarti kita bisa mencegah orang lain membuat objek dari kelas tersebut. Misalnya, jika kita ingin kelas itu hanya punya satu objek, kita bisa membuat constructor-nya private. Dengan cara ini, kita bisa mengontrol kapan dan bagaimana objek itu dibuat.

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()

```
RAKTIKUM W03 > Anggota.java > Anggota > angsur(int)
1  public class Anggota {
2      private String nomorKTP;
3      private String nama;
4      private int limitPinjaman;
5      private int jumlahPinjaman;
6
7      public Anggota(String nomorKTP, String nama, int limitPinjaman){
8          this.nomorKTP = nomorKTP;
9          this.nama = nama;
10         this.limitPinjaman = limitPinjaman;
11         this.jumlahPinjaman = 0;
12     }
13     public String getName(){
14         return nama;
15     }
16
17     public String noKtp(){
18         return nomorKTP;
19     }
20
21     public int getJumlahPinjaman(){
22         return jumlahPinjaman;
23     }
24
25     public int getLimitPinjaman(){
26         return limitPinjaman;
27     }
28 }
```

```

9      public void pinjam(int jumlah){
10          if (jumlahPinjaman + jumlah > limitPinjaman) {
11              System.out.println(x:"Maaf, jumlah pinjaman melebihi limit.");
12          } else {
13              jumlahPinjaman += jumlah;
14          }
15      }
16
17      public void angsur(int jumlah){
18          if (jumlah >= jumlahPinjaman) {
19              jumlahPinjaman = 0;
20          } else {
21              jumlahPinjaman -= jumlah;
22          }
23      }
24  }
25

```

```

PRAKTIKUM W03 > TestKoperasi.java > TestKoperasi > main(String[])
1  public class TestKoperasi {
2      Run | Debug
3      public static void main(String[] args) {
4          Anggota anggota1 = new Anggota(nomorKTP:"1113333444", nama:"Donny", limitPinjaman:5000000);
5
6          System.out.println("Nama Anggota: " + anggota1.getNama());
7          System.out.println("Limit Pinjaman: " + anggota1.getLimitPinjaman());
8
9          System.out.println(x:"\nMeminjam uang 10.000.000...");
10         anggota1.pinjam(jumlah:10000000);
11         System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());
12
13         System.out.println(x:"\nMeminjam uang 4.000.000...");
14         anggota1.pinjam(jumlah:4000000);
15         System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());
16
17         System.out.println(x:"\nMembayar angsuran 1.000.000");
18         anggota1.angsur(jumlah:1000000);
19         System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());
20
21         System.out.println(x:"\nMembayar angsuran 3.000.000");
22         anggota1.angsur(jumlah:3000000);
23         System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());
24     }
25 }

```

```

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
PS E:\Semester 3\PBO\PRAKTIKUM>

```

Ready

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”

```
public void angsur(int jumlah){
    if (jumlah < jumlahPinjaman * 0.1) {
        System.out.println(x:"Maaf, angsuran harus 10% dari jumlah pinjaman.");
    } else {
        jumlahPinjaman -= jumlah;
        if (jumlahPinjaman < 0) {
            jumlahPinjaman = 0;
        }
    }
}
```

PRAKTIKUM W03 > TestKoperasi.java > TestKoperasi > main(String[])

```
1 public class TestKoperasi {
2     public static void main(String[] args) {
15
16         System.out.println(x:"\nMembayar angsuran 1.000.000");
17         anggota1.angsur(jumlah:1000000);
18         System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());
19
20         System.out.println(x:"\nMembayar angsuran 200.000");
21         anggota1.angsur(jumlah:200000);
22         System.out.println("Jumlah pinjaman saat ini: " + anggota1.getJumlahPinjaman());
23
24         System.out.println(x:"\nMembayar angsuran 3.000.000");
25         anggota1.angsur(jumlah:3000000);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

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 200.000
Maaf, angsuran harus 10% dari jumlah pinjaman.
Jumlah pinjaman saat ini: 3000000

Membayar angsuran 3.000.000
Jumlah pinjaman saat ini: 0

PS E:\Semester 3\PBO\PRAKTIKUM> □