

Assignment Paper

**OBJECT ORIENTED PROGRAMMING (OOP)
PRACTICUM
MOCHAMAD WILDANI AZIZI**

Percobaan 1: Enkapsulasi

Didalam percobaan enkapsulasi, buatlah class Motor yang memiliki atribut kecepatan dan kontakOn, dan memiliki method printStatus() untuk menampilkan status motor. Seperti berikut:

1. Buka Netbeans, buat project MotorEncapsulation.
2. Buat class Motor. Klik kanan pada package motorencapsulation – New – Java Class.
3. Ketikkan kode class Motor dibawah ini.

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

4. Kemudian buat class MotorDemo, ketikkan kode berikut ini.

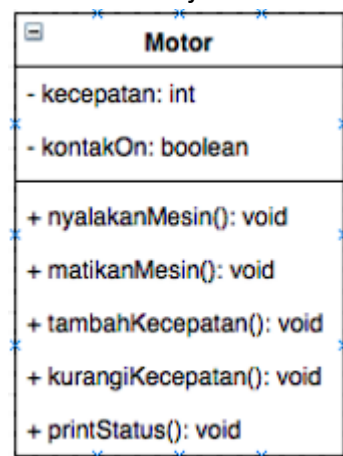
```
1 package motorencapsulation;
2
3 public class MotorDemo {
4     public static void main(String[] args) {
5         Motor motor = new Motor();
6         motor.printStatus();
7         motor.kecepatan = 50;
8         motor.printStatus();
9     }
10 }
11
```

5. Hasilnya adalah sebagai berikut

```
PS C:\Users\Moch Wil  
cuments\3rd Semester  
ges' '-cp' 'C:\Users  
ava\jdt_ws\JS3_5373d  
Kontak Off  
Kecepatan 0  
  
Kontak Off  
Kecepatan 50
```

Percobaan 2: Access Modifier

1. Ubah cara kerja class motor sesuai dengan UML class diagram berikut.



2. Berdasarkan UML class diagram tersebut maka class Motor terdapat perubahan, yaitu:
- Ubah access modifier kecepatan dan kontakOn menjadi private
 - Tambahkan method nyalakanMesin, matikanMesin, tambahKecepatan, kurangiKecepatan.

Implementasi class Motor adalah sebagai berikut:

```

1  package motorencapsulation;
2
3  public class Motor {
4      private int kecepatan = 0;
5      private boolean kontakOn = false;
6
7      public void nyalakanMesin(){
8          kontakOn = true;
9      }
10     public void matikanMesin(){
11         kontakOn = false;
12         kecepatan = 0;
13     }
14     public void tambahKecepatan(){
15         if (kontakOn == true) {
16             kecepatan += 5;
17         } else {
18             System.out.println("Kecepatan tidak bisa bertambah karena Mesin Off! \n");
19         }
20     }
21     public void kurangiKecepatan(){
22         if (kontakOn == true) {
23             kecepatan -= 5;
24         } else {
25             System.out.println("Kecepatan tidak bisa berkurang karena Mesin Off! \n");
26         }
27     }
28     public void printStatus(){
29         if (kontakOn == true) {
30             System.out.println("Kontak On");
31         } else {
32             System.out.println("Kontak Off");
33         }
34         System.out.println("Kecepatan " + kecepatan + "\n");
35     }
36 }
37

```

3. Kemudian pada class MotorDemo, ubah code menjadi seperti berikut:

```

1  package motorencapsulation;
2
3  public class MotorDemo {
4      public static void main(String[] args) {
5          Motor motor = new Motor();
6          motor.printStatus();
7          motor.tambahKecepatan();
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     }
24 }
25

```

4. Hasilnya dari class MotorDemo adalah sebagai berikut:

```
PS C:\Users\Moch Wildani Azizi\OneDrive\Documents\Documents\3rd Semester\JS\p00P\JS3; & 'C:\Program Files' '-cp' 'C:\Users\Moch Wildani Azizi\AppData\Roaming\jdts\JS3_5373df6f\bin' 'motorencapsulation.M
Kontak Off
Kecepatan 0

Kecepatan tidak bisa bertambah karena Mesin Off!

Kontak On
Kecepatan 0

Kontak On
Kecepatan 5

Kontak On
Kecepatan 10

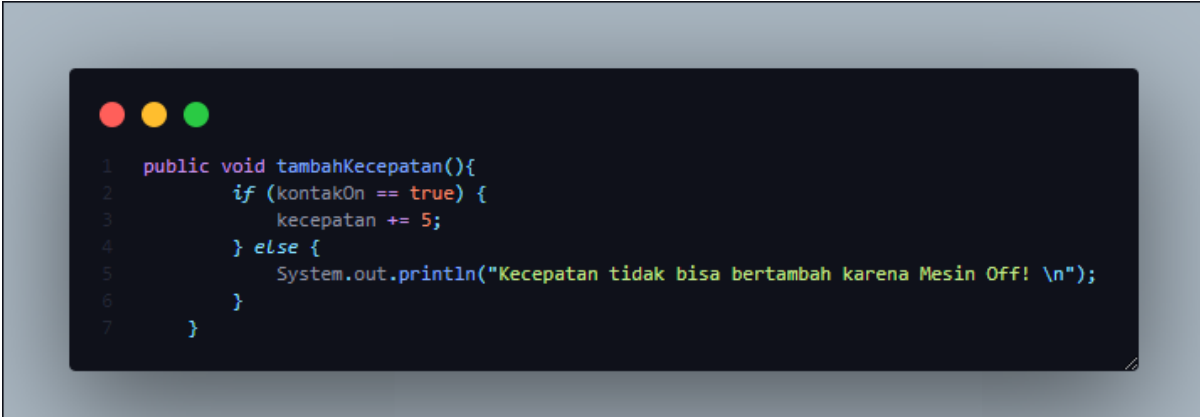
Kontak On
Kecepatan 15

Kontak Off
Kecepatan 0
```

Pertanyaan

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

Jawab: karena Peringatan "Kecepatan tidak bisa bertambah karena Mesin Off!" muncul saat kita mencoba menambah kecepatan untuk pertama kalinya dalam class MotorDemo karena pada saat itu mesin motor belum dinyalakan. Di dalam metode tambahKecepatan() di class Motor, terdapat kondisi berikut:



```
1 public void tambahKecepatan(){
2     if (kontakOn == true) {
3         kecepatan += 5;
4     } else {
5         System.out.println("Kecepatan tidak bisa bertambah karena Mesin Off! \n");
6     }
7 }
```

Kondisi ini memeriksa apakah kontakOn (status mesin) adalah true (nyala). Jika mesin dalam keadaan mati (kontakOff), maka pernyataan else akan dieksekusi, dan pesan "Kecepatan tidak bisa bertambah karena Mesin Off!" akan dicetak ke layar.

2. Mengapa atribut kecepatan dan kontakOn diset private?

Jawab: karena Atribut kecepatan dan kontakOn diset sebagai private dalam class Motor untuk menerapkan konsep enkapsulasi dalam pemrograman. Enkapsulasi adalah salah satu prinsip utama dalam paradigma pemrograman berorientasi objek (OOP) yang bertujuan untuk menyembunyikan detail implementasi internal dari suatu objek dan hanya memungkinkan akses ke objek tersebut melalui antarmuka yang ditentukan.

3. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

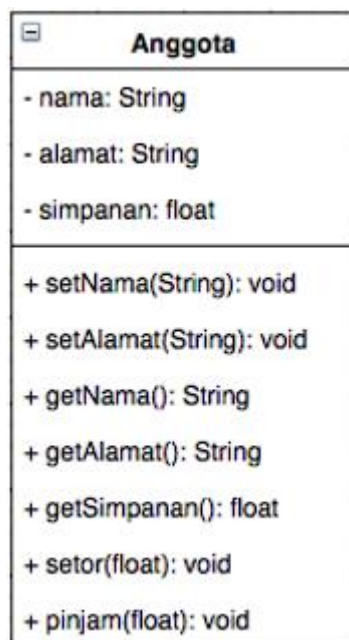
Jawab:

```
1 package motorencapsulation;
2
3 public class Motor {
4     private int kecepatan = 0;
5     private boolean kontakOn = false;
6     private static final int kec_max = 100;
7
8     public void nyalakanMesin(){
9         kontakOn = true;
10    }
11    public void matikanMesin(){
12        kontakOn = false;
13        kecepatan = 0;
14    }
15    public void tambahKecepatan(){
16        if (kontakOn == true) {
17            if (kecepatan + 5 <= kec_max) {
18                kecepatan += 5;
19            } else {
20                kecepatan += kec_max;
21            }
22        } else {
23            System.out.println("Kecepatan tidak bisa bertambah karena Mesin Off! \n");
24        }
25    }
26    public void kurangiKecepatan(){
27        if (kontakOn == true) {
28            if (kecepatan - 5 >= 0) {
29                kecepatan -= 5;
30            } else {
31                kecepatan = 0;
32            }
33        } else {
34            System.out.println("Kecepatan tidak bisa berkurang karena Mesin Off! \n");
35        }
36    }
37    public void printStatus(){
38        if (kontakOn == true) {
39            System.out.println("Kontak On");
40        } else {
41            System.out.println("Kontak Off");
42        }
43        System.out.println("Kecepatan " + kecepatan + "\n");
44    }
45 }
46 }
```

Percobaan 3 : Getter dan Setter

Misalkan di sebuah sistem informasi koperasi, terdapat class Anggota. Anggota memiliki atribut nama, alamat dan simpanan, dan method setter, getter dan setor dan pinjam. Semua atribut pada anggota tidak boleh diubah sembarangan, melainkan hanya dapat diubah melalui method setter, getter, setor dan tarik. Khusus untuk atribut simpanan tidak terdapat setter karena simpanan akan bertambah ketika melakukan transaksi setor dan akan berkurang ketika melakukan peminjaman/tarik.

1. Berikut ini UML class buatlah class Mahasiswa pada program:



2. Sama dengan percobaan 1 untuk membuat project baru
 - a. Buka Netbeans, buat project KoperasiGetterSetter.
 - b. Buat class Anggota. Klik kanan pada package kopersigettersetter – New –Java Class.
 - c. Ketikkan kode class Anggota dibawah ini.


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

3. Selanjutnya buatlah class KoperasiDemo untuk mencoba class Anggota.

```

1 package kopersigettersetter;
2
3 public class KoperasiDemo {
4     public static void main(String[] args) {
5
6         Anggota anggota1 = new Anggota();
7         anggota1.setNama("Iwan Setiawan");
8         anggota1.setAlamat("Jalan Sukarno Hatta no 10");
9         anggota1.setor(100000);
10        System.out.println("Simpanan " + anggota1.getNama() + " : Rp." + anggota1.getSimpanan());
11
12        anggota1.pinjam(5000);
13        System.out.println("Simpanan " + anggota1.getNama() + " : Rp." + anggota1.getSimpanan());
14    }
15 }
16

```

4. Hasil dari main method pada langkah ketiga adalah

```

PS C:\Users\Moch Wildani Azizi\OneDrive\Desktop> java -cp 'C:\Program Files\Java\jdk-18.0.2\bin' kopersigettersetter.KoperasiDemo
Simpanan Iwan Setiawan : Rp.100000.0
Simpanan Iwan Setiawan : Rp.95000.0

```

Percobaan 4 : Konstruktor, Instantiasi

1. Langkah pertama percobaan 4 adalah ubah class KoperasiDemo seperti berikut

```

1 package kopersigettersetter;
2
3 public class KoperasiDemo {
4     public static void main(String[] args) {
5         Anggota anggota1 = new Anggota();
6
7         System.out.println("Simpanan " + anggota1.getNama() + " : Rp." + anggota1.getSimpanan());
8
9         anggota1.setNama("Iwan Setiawan");
10        anggota1.setAlamat("Jalan Sukarno Hatta no 10");
11        anggota1.setor(100000);
12        System.out.println("Simpanan " + anggota1.getNama() + " : Rp." + anggota1.getSimpanan());
13
14        anggota1.pinjam(5000);
15        System.out.println("Simpanan " + anggota1.getNama() + " : Rp." + anggota1.getSimpanan());
16    }
17 }
18

```

2. Hasil dari program tersebut adalah sebagai berikut

```
PS C:\Users\Moch Wildani Azizi\OneDrive\Desktop> java -XX:+ShowCodeDetailsInExceptionMessages -jar C:\Users\Moch Wildani Azizi\OneDrive\Desktop\92c5001846fcf5723a4eedd72aee2172\redh.jar
Simpanan null : Rp.0.0
Simpanan Iwan Setiawan : Rp.100000.0
Simpanan Iwan Setiawan : Rp.95000.0
```

3. Ubah class Anggota menjadi seperti berikut

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

4. Selanjutnya ubah class KoperasiDemo sebagai berikut

```
1 package koperasigettersetter;
2
3 public class KoperasiDemo {
4     public static void main(String[] args) {
5         Anggota anggota1 = new Anggota("Iwan", "Jalan Mawar");
6
7         System.out.println("Simpanan " + anggota1.getNama() + " : Rp." + anggota1.getSimpanan());
8
9         anggota1.setNama("Iwan Setiawan");
10        anggota1.setAlamat("Jalan Sukarno Hatta no 10");
11        anggota1.setor(100000);
12        System.out.println("Simpanan " + anggota1.getNama() + " : Rp." + anggota1.getSimpanan());
13
14        anggota1.pinjam(5000);
15        System.out.println("Simpanan " + anggota1.getNama() + " : Rp." + anggota1.getSimpanan());
16    }
17 }
18
```

5. Hasil dari program tersebut adalah sebagai berikut

```
cuments\3rd Semester\JS\p00P\JS3'; &
ges' '-cp' 'C:\Users\Moch Wildani Azi
ava\jdt_ws\JS3_5373df6f\bin' 'koperas
Simpanan Iwan : Rp.0.0
Simpanan Iwan Setiawan : Rp.100000.0
Simpanan Iwan Setiawan : Rp.95000.0
```

Pertanyaan : Percobaan 3 dan 4

1. Apa yang dimaksud getter dan setter?

Jawab: Getter dan Setter adalah metode dalam pemrograman objek yang digunakan untuk mengakses (getter) dan mengubah (setter) nilai dari atribut atau properti suatu objek. Getter digunakan untuk mengambil nilai dari atribut, sedangkan setter digunakan untuk mengubah nilai atribut tersebut.

2. Apa kegunaan dari method getSimpanan()?

Jawab: Method getSimpanan() mungkin digunakan untuk mengambil nilai atau informasi tentang jumlah simpanan dari suatu objek atau entitas dalam konteks program yang lebih besar.

3. Method apa yang digunakan untuk menambah saldo?

Jawab: Untuk menambah saldo menggunakan method yang disebut setor(*jumlah uang*). Method ini akan menerima jumlah yang ingin ditambahkan ke saldo saat ini dan mengubah saldo sesuai dengan jumlah yang diberikan.

4. Apa yang dimaksud konstruktor?

Jawab: Konstruktor adalah sebuah method khusus dalam pemrograman berorientasi objek yang digunakan untuk menginisialisasi objek dari suatu kelas. Konstruktor biasanya

dipanggil saat objek baru dibuat dari kelas tersebut. Ini digunakan untuk mengatur nilai awal atribut atau melakukan inisialisasi lain yang diperlukan.

5. Sebutkan aturan dalam membuat konstruktor?

Jawab: Aturan dalam membuat konstruktor biasanya melibatkan beberapa hal berikut:

- Nama konstruktor harus sama dengan nama kelas.
- Konstruktor dapat memiliki parameter atau tidak.
- Konstruktor dapat memiliki aksesibilitas seperti public, private, atau protected.
- Konstruktor dapat melakukan inisialisasi atribut objek.

6. Apakah boleh konstruktor bertipe private?

Jawab: konstruktor bisa memiliki tipe akses private. Dalam hal ini, konstruktor private mencegah pembuatan objek lainnya di luar kelas itu sendiri.

7. Kapan menggunakan parameter dengan passing parameter?

Jawab: Parameter dengan passing parameter digunakan ketika ingin mengirim data atau informasi tambahan ke sebuah method atau fungsi. Ini digunakan ketika perlu memberikan input yang diperlukan untuk melakukan operasi tertentu.

8. Apa perbedaan atribut class dan instansiasi atribut?

Jawab: Perbedaan antara atribut kelas (class attribute) dan atribut instansiasi (instance attribute) adalah:

- Atribut kelas adalah atribut yang terkait dengan kelas itu sendiri dan berlaku untuk semua instance dari kelas tersebut. Mereka didefinisikan di dalam kelas tetapi di luar method.
- Atribut instansiasi adalah atribut yang dimiliki oleh setiap instance individu dari kelas. Mereka didefinisikan dalam method konstruktor atau method lain di dalam kelas, dan setiap instance memiliki salinan atribut ini yang independen.

9. Apa perbedaan class method dan instansiasi method?

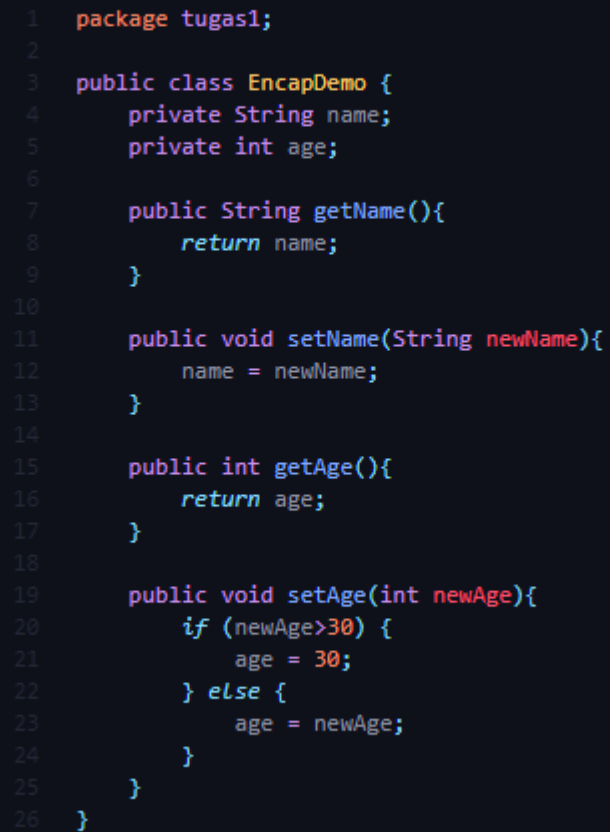
Jawab: Perbedaan antara class method dan instansiasi method (biasanya disebut instance method) adalah:

- Class method adalah method yang terkait dengan kelas dan tidak bergantung pada instance objek. Mereka dapat diakses melalui kelas itu sendiri tanpa membuat objek dari kelas tersebut.
- Instansiasi method adalah method yang berkaitan dengan instance objek dari kelas. Mereka bekerja pada data yang terkait dengan objek spesifik dan biasanya memerlukan objek sebagai argumen pertama (biasanya disebut self dalam Python).

Tugas

1. Cobalah program dibawah ini dan tuliskan hasil outputnya

Jawab:



```
1  package tugas1;
2
3  public class EncapDemo {
4      private String name;
5      private int age;
6
7      public String getName(){
8          return name;
9      }
10
11     public void setName(String newName){
12         name = newName;
13     }
14
15     public int getAge(){
16         return age;
17     }
18
19     public void setAge(int newAge){
20         if (newAge>30) {
21             age = 30;
22         } else {
23             age = newAge;
24         }
25     }
26 }
```

```
1  package tugas1;
2
3  public class EncapTest {
4      public static void main(String[] args) {
5          EncapDemo encap = new EncapDemo();
6          encap.setName("James");
7          encap.setAge(35);
8
9          System.out.println("Name: " +encap.getName());
10         System.out.println("Age: " +encap.getAge());
11     }
12 }
13
```

Result:

```
PS C:\Users\...
S3> & 'C:\P...
etailsInExce...
aming\Code\U...
.java\jdt_ws...
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.
Jawab: karena potongan kode pada class EncapDemo pada line 19-25 berfungsi apabila nilai yang diinputkan diatas/lebih besar dari 30 maka output yang akan keluar pasti angka 30.
3. Ubah program diatas agar atribut age dapat diberi nilai maksimal 30 dan minimal 18.



```
1 public void setAge(int newAge){
2     if (newAge > 30) {
3         age = 30;
4     } else if (newAge < 18) {
5         age = 18;
6     } else {
7         age = newAge;
8     }
9 }
```



```
1 package tugas1;
2
3 public class EncapTest {
4     public static void main(String[] args) {
5         EncapDemo encap = new EncapDemo();
6         encap.setName("James");
7         encap.setAge(35);
8         EncapDemo encap2 = new EncapDemo();
9         encap2.setName("Albert");
10        encap2.setAge(17);
11
12        System.out.println("Name: " +encap.getName());
13        System.out.println("Age: " +encap.getAge());
14        System.out.println("Name: " +encap2.getName());
15        System.out.println("Age: " +encap2.getAge());
16    }
17 }
18
```

Result:

```
PS C:\Users\Mo
Azizi\OneDrive
X:+ShowCodeDet
ceStorage\92c5
Name: James
Age: 30
Name: Albert
Age: 18
```


4. 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.

```
1 package tugas4;
2
3 public class Anggota {
4     private String nomorKTP;
5     private String nama;
6     private double limitPinjaman;
7     private double jumlahPinjaman;
8
9     public Anggota(String nomorKTP, String nama, double limitPinjaman) {
10         this.nomorKTP = nomorKTP;
11         this.nama = nama;
12         this.limitPinjaman = limitPinjaman;
13         this.jumlahPinjaman = 0; // Awalnya jumlah pinjaman adalah 0
14     }
15
16     public String getNomorKTP() {
17         return nomorKTP;
18     }
19
20     public String getNama() {
21         return nama;
22     }
23
24     public double getLimitPinjaman() {
25         return limitPinjaman;
26     }
27
28     public double getJumlahPinjaman() {
29         return jumlahPinjaman;
30     }
31
32     public void pinjam(double jumlahPinjaman) {
33         if (jumlahPinjaman <= 0) {
34             System.out.println("Maaf, jumlah pinjaman harus lebih dari 0.");
35         } else if (this.jumlahPinjaman + jumlahPinjaman > limitPinjaman) {
36             System.out.println("Maaf, jumlah pinjaman melebihi limit.");
37         } else {
38             this.jumlahPinjaman += jumlahPinjaman;
39         }
40     }
41
42     public void angsur(double jumlahAngsuran) {
43         if (jumlahAngsuran <= 0) {
44             System.out.println("Maaf, jumlah angsuran harus lebih dari 0.");
45         } else if (jumlahAngsuran > this.jumlahPinjaman) {
46             System.out.println("Maaf, jumlah angsuran melebihi jumlah pinjaman.");
47         } else {
48             this.jumlahPinjaman -= jumlahAngsuran;
49         }
50     }
51 }
52
53
```

```

1 package tugas4;
2
3 public class TestKoperasi {
4     public static void main(String[] args) {
5         Anggota donny = new Anggota("111333444", "Donny", 5000000);
6
7         System.out.println("Nama Anggota: " + donny.getNama());
8         System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());
9
10        System.out.println("\nMeminjam uang 10.000.000...");
11        donny.pinjam(10000000);
12        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
13
14        System.out.println("\nMeminjam uang 4.000.000...");
15        donny.pinjam(4000000);
16        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
17
18        System.out.println("\nMembayar angsuran 1.000.000...");
19        donny.angsur(1000000);
20        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
21
22        System.out.println("\nMembayar angsuran 3.000.000...");
23        donny.angsur(3000000);
24        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
25    }
26 }

```

```

PS C:\Users\Moch Wildani Azizi\OneDrive\Desktop> java -version
8.0.2.1\bin\java.exe' '-agentlib:jdwp=
+ShowCodeDetailsInExceptionMessages'
Storage\92c5001846fcf5723a4eedd72aee21
Nama Anggota: Donny
Limit Pinjaman: 5000000.0

Meminjam uang 10.000.000...
Maaf, jumlah pinjaman melebihi limit.
Jumlah pinjaman saat ini: 0.0

Meminjam uang 4.000.000...
Jumlah pinjaman saat ini: 4000000.0

Membayar angsuran 1.000.000...
Jumlah pinjaman saat ini: 3000000.0

Membayar angsuran 3.000.000...
Jumlah pinjaman saat ini: 0.0

```

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".

```

1 public void angsur(double jumlahAngsuran) {
2     if (jumlahAngsuran <= 0) {
3         System.out.println("Maaf, jumlah angsuran harus lebih dari 0.");
4     } else if (jumlahAngsuran < 0.1 * this.jumlahPinjaman) {
5         System.out.println("Maaf, angsuran harus minimal 10% dari jumlah pinjaman.");
6     } else if (jumlahAngsuran > this.jumlahPinjaman) {
7         System.out.println("Maaf, jumlah angsuran melebihi jumlah pinjaman.");
8     } else {
9         this.jumlahPinjaman -= jumlahAngsuran;
10        System.out.println("Angsuran berhasil. Jumlah pinjaman saat ini: " + this.jumlahPinjaman);
11    }
12 }

```

```

Nama Anggota: Donny
Limit Pinjaman: 5000000.0

Meminjam uang 10.000.000...
Maaf, jumlah pinjaman melebihi limit.
Jumlah pinjaman saat ini: 0.0

Meminjam uang 4.000.000...
Jumlah pinjaman saat ini: 4000000.0

Membayar angsuran 300.000...
Maaf, angsuran harus minimal 10% dari jumlah pinjaman.
Jumlah pinjaman saat ini: 4000000.0

Membayar angsuran 3.000.000...
Angsuran berhasil. Jumlah pinjaman saat ini: 1000000.0
Jumlah pinjaman saat ini: 1000000.0

```

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

```

1 package tugas4;
2
3 import java.util.Scanner;
4
5 public class TestKoperasi {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8
9         System.out.print("Masukkan nomor KTP: ");
10        String nomorKTP = scanner.nextLine();
11
12        System.out.print("Masukkan nama: ");
13        String nama = scanner.nextLine();
14
15        System.out.print("Masukkan limit pinjaman: ");
16        double limitPinjaman = scanner.nextDouble();
17        scanner.nextLine(); // Membersihkan newline character dari buffer
18
19        Anggota anggota1 = new Anggota(nomorKTP, nama, limitPinjaman);
20
21        System.out.println("Nomor KTP: " + anggota1.getNomorKTP());
22        System.out.println("Nama: " + anggota1.getNama());
23        System.out.println("Limit Pinjaman: Rp " + anggota1.getLimitPinjaman());
24
25        System.out.print("Masukkan jumlah pinjaman: ");
26        double jumlahPinjaman = scanner.nextDouble();
27        anggota1.pinjam(jumlahPinjaman);
28
29        System.out.print("Masukkan jumlah angsuran: ");
30        double jumlahAngsuran = scanner.nextDouble();
31        anggota1.angsur(jumlahAngsuran);
32
33        // Tutup scanner setelah digunakan
34        scanner.close();
35    }
36 }
37

```

```

Masukkan nomor KTP: 01
Masukkan nama: Abdul
Masukkan limit pinjaman: 5000000
Nomor KTP: 01
Nama: Abdul
Limit Pinjaman: Rp 5000000.0
Masukkan jumlah pinjaman: 4000000
Masukkan jumlah angsuran: 2000000
Angsuran berhasil. Jumlah pinjaman saat ini: 2000000.0

```