

Praktikum Pemrograman Berbasis Objek



Muhammad Nurfauzi Ikhsan

2241720076

2I Class

18

**STUDY PROGRAM D-IV INFORMATIC ENGINEER
INFORMATION TECHNOLOGY DEPARTMENT
MALANG STATE POLYTECHNIC**

Soekarno Hatta Street No.9, Jatimulyo, Lowokwaru District, Malang City, East Java
65141

3.1 Percobaan 1: Enkapsulasi

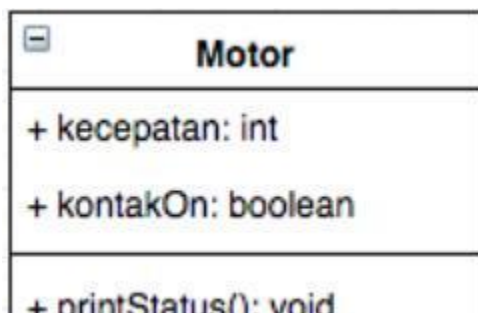
Didalam percobaan enkapsulasi, buatlah class **Motor** yang memiliki atribut kecepatan dan kontakOn, dan memiliki method `printStatus()` untuk menampilkan status motor. Seperti berikut:

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        }
11        else{
12            System.out.println("Kontak Off");
13        }
14        System.out.println("Kecepatan " + kecepatan+"\n");
15    }
16 }
```

```
public class motor {
    public int kecepatan = 0;
    public boolean kontakOn = false;
    public void printStatus() {
        if (kontakOn == true) {
            System.out.println("Kontak On");
        } else {
            System.out.println("Kontak Off");
        }
        System.out.println("Kecepatan " + kecepatan + "\n");
    }
}
```

bentuk UML class diagram class **Motor** adalah sebagai berikut:



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 }
```

```
public class Main {
    public static void main(String[] args) {
        motor motor = new motor();
        motor.printStatus();
        motor.kecepatan = 50;
        motor.printStatus();
    }
}
```

5. Hasilnya adalah sebagai berikut:

```
Kontak Off
Kecepatan 0

Kontak Off
Kecepatan 50

Process finished with exit code 0
```

Dari percobaan 1 - enkapsulasi, menurut anda, adakah yang janggal?

Yaitu, kecepatan motor tiba-tiba saja berubah dari 0 ke 50. Lebih janggal lagi, posisi kontak motor masih dalam kondisi OFF. Bagaimana mungkin sebuah motor bisa sekejap berkecepatan dari nol ke 50, dan itupun kunci kontaknya OFF?

Nah dalam hal ini, akses ke atribut motor ternyata tidak terkontrol. Padahal, objek di dunia nyata selalu memiliki batasan dan mekanisme bagaimana objek tersebut dapat digunakan. Lalu, bagaimana kita bisa memperbaiki class Motor diatas agar dapat digunakan dengan baik? Kita bisa pertimbangkan beberapa hal berikut ini:

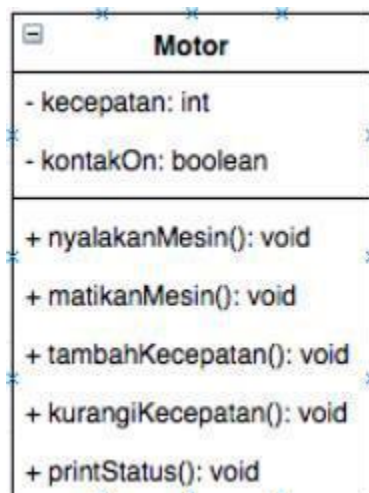
1. Menyembunyikan atribut internal (kecepatan, kontakOn) dari pengguna (class lain)
2. Menyediakan method khusus untuk mengakses atribut.

Untuk itu mari kita lanjutkan percobaan berikutnya tentang Access Modifier

3.2 Percobaan 2 – Access Modifier

Pada percobaan ini akan digunakan access modifier untuk memperbaiki cara kerja class Motor pada percobaan ke-1.

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



2. dasarkan UML class diagram tersebut maka class Motor terdapat perubahan, yaitu:

- a. Ubah access modifier kecepatan dan kontakOn menjadi private

- b. Tambahkan method nyalakanMesin, matikanMesin, tambahKecepatan, kurangiKecepatan.

Implementasi class Motor adalah sebagai berikut:

```
public class motor {  
  
    private int kecepatan = 0;  
  
    private boolean kontakOn = false;  
  
    public void nyalakanMesin() {  
  
        kontakOn = true;  
  
    }  
  
    public void matikanMesin() {  
  
        kontakOn = false;  
  
        kecepatan = 0;  
  
    }  
  
    public void tambahKecepatan() {  
  
        if (kontakOn == true) {  
  
            kecepatan += 5;  
  
        } else {  
  
            System.out.println("Kecepatan tidak bisa bertambah karena");  
  
        }  
  
    }  
  
    public void kurangiKecepatan() {  
  
        if (kontakOn == true) {
```

```

        kecepatan -= 5;

    } else {

        System.out.println("Kecepatan tidak bisa berkurang karena Mesin ");

    }

}

public void printStatus() {

    if (kontakOn == true) {

        System.out.println("Kontak On");

    } else {

        System.out.println("Kontak Off");

    }

    System.out.println("Kecepatan " + kecepatan + "\n");

}

}

```

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 }

```

```

public class Main {
    public static void main(String[] args) {
        motor motor = new motor();
    }
}

```

```

        motor.printStatus();
        motor.tambahKecepatan();

        motor.nyalakanMesin();
        motor.printStatus();

        motor.tambahKecepatan();
        motor.printStatus();

        motor.tambahKecepatan();
        motor.printStatus();

        motor.tambahKecepatan();
        motor.printStatus();

        motor.matikanMesin();
        motor.printStatus();

    }
}

```

4. Hasilnya dari class MotorDemo adalah sebagai berikut:

```

Kontak Off
Kecepatan 0

Kecepatan tidak bisa bertambah karena
Kontak On
Kecepatan 0

Kontak On
Kecepatan 5

Kontak On
Kecepatan 10

Kontak On
Kecepatan 15

Kontak Off
Kecepatan 0

Process finished with exit code 0

```

Dari percobaan diatas, dapat kita amati sekarang atribut kecepatan tidak bisa diakses oleh pengguna dan diganti nilainya secara sembarangan. Bahkan ketika mencoba menambah kecepatan saat posisi kontak masih OFF, maka akan muncul notifikasi bahwa mesin OFF. Untuk

mendapatkan kecepatan yang diinginkan, maka harus dilakukan secara gradual, yaitu dengan memanggil method `tambahKecepatan()` beberapa kali. Hal ini mirip seperti saat kita mengendarai motor.

3.3 Pertanyaan

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

Jawab:

Hal ini dikarenakan nilai dari atribut `kontakOn` bernilai *false*, sehingga pada saat dilakukan percabangan *if else* dengan indikator `kontakOn == true` tidak dapat masuk ke dalam penambahan kecepatan dan akan masuk ke dalam percabangan *else*.

2. Mengapa atribut kecepatan dan `kontakOn` diset `private`?

Jawab:

Hal ini untuk membuat atribut tersebut hanya dapat di akses dari kelas tersebut dan tidak dapat di akses dari luar kelas *Motor.java*

3. Ubah class `Motor` sehingga kecepatan maksimalnya adalah 100!

Jawab:

Tambahkan atribut `maxSpeed`:

```
private int maxSpeed = 100;
```

Tambahkan kondisi pada method `tambahKecepatan`:

```
public void tambahKecepatan() {  
  
    if (kontakOn == true && kecepatan < maxSpeed) { kecepatan += 5;
```



```

    } else if (kontakOn == false) {

        System.out.println("Kecepatan tidak bisa bertambah karena Mesin

            Off! \n");

    } else {

        System.out.println("Kecepatan melebihi batas! \n");

    }

}

```

Melakukan looping untuk mengetahui hasil akhir:

```

for (int i = 0; i < 22; i++) {

    motor.tambahKecepatan();

    if (i == 20) {

        motor.printStatus();

    }

}

```

Output di halaman selanjutnya

```
Kontak Off
Kecepatan 0

Kecepatan tidak bisa bertambah karena Mesin
Kontak On
Kecepatan 0

Kontak On
Kecepatan 100

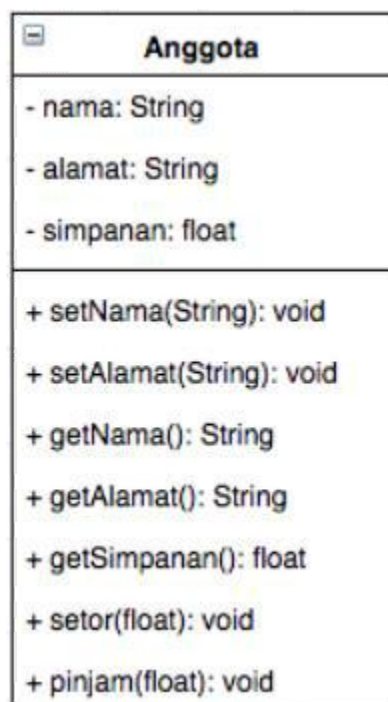
Kontak Off
Kecepatan 0

Process finished with exit code 0
```

3.4 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 **koperasigettersetter** – New – Java Class.
- c. Ketikkan kode class Anggota dibawah ini

```
1 package koperasigettersetter;
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    public void setor(float uang){
23        simpanan +=uang;
24    }
25    public void pinjam(float uang){
26        simpanan -=uang;
27    }
28 }
```

setter, getter
nama dan alamat

getter
simpanan

```
public class Anggota {

    private String nama;

    private String alamat;
```

```
private float simpanan;

public void setNama(String nama) {

    this.nama = nama;

}

public void setAlamat(String alamat) {

    this.alamat = alamat;

}

public String getNama() {

    return nama;

}

public String getAlamat() {

    return alamat;

}

public float getSimpanan() {

    return simpanan;

}

public void setor(float uang) {

    simpanan += uang;

}

public void pinjam(float uang) {

    simpanan -= uang;

}
```

```
}
```

Jika diperhatikan pada class Anggota, atribut nama dan alamat memiliki masing-masing 1 getter dan setter. Sedangkan atribut simpanan hanya memiliki getSimpanan() saja, karena seperti tujuan awal, atribut simpanan akan berubah nilainya jika melakukan transaksi setor() dan pinjam/tarik().

3. Selanjutnya buatlah class KoperasiDemo untuk mencoba class Anggota.

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

```
public class Main {
    public static void main(String[] args) {
        Anggota anggota1 = new Anggota();

        anggota1.setNama("Iwan Setiawan");

        anggota1.setAlamat("Jalan Soekarno Hatta No 10");

        anggota1.setor(100000);

        System.out.println("Simpanan " + anggota1.getNama() + " Rp " +
            anggota1.getSimpanan());

        anggota1.pinjam(5000);

        System.out.println("Simpanan " + anggota1.getNama() + " Rp " +
            anggota1.getSimpanan());

    }
}
```

4. Hasil dari main method pada langkah ketiga adalah

```
run:
Simpanan Iwan Setiawan : Rp 100000.0
Simpanan Iwan Setiawan : Rp 95000.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
-Dsun.stderr.encoding=UTF-8 -classpath E:\KULIAH\c
Simpanan Iwan Setiawan Rp 100000.0
Simpanan Iwan Setiawan Rp 95000.0

Process finished with exit code 0
```

Dapat dilihat pada hasil percobaan diatas, untuk mengubah simpanan tidak dilakukan secara langsung dengan mengubah atribut simpanan, melainkan melalui method setor() dan pinjam(). Untuk menampilkan nama pun harus melalui method getName(), dan untuk menampilkan simpanan melalui getSimpanan().

3.5 Percobaan 4 - Konstruktur, Instansiasi

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

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

```
public class Main {
    public static void main(String[] args) { Anggota anggota1 = new Anggota();

        System.out.println("Simpanan " + anggota1.getName() + " : Rp " +
    anggota1.getSimpanan());
```

```

    anggota1.setNama("Iwan Setiawan");

    anggota1.setAlamat("Jalan Soekarno Hatta No 10"); anggota1.setor(100000);

    System.out.println("Simpanan " + anggota1.getNama() + " Rp " +
    anggota1.getSimpanan());

    anggota1.pinjam(5000);

    System.out.println("Simpanan " + anggota1.getNama() + " Rp " +
    anggota1.getSimpanan());

}
}

```

2. Hasil dari program tersebut adalah sebagai berikut

```

run:
Simpanan null : Rp 0.0
Simpanan Iwan Setiawan : Rp 100000.0
Simpanan Iwan Setiawan : Rp 95000.0
BUILD SUCCESSFUL (total time: 0 seconds)

```

```

-Dsun.stderr.encoding=UTF-8 -classpath E:\
Simpanan null : Rp 0.0
Simpanan Iwan Setiawan Rp 100000.0
Simpanan Iwan Setiawan Rp 95000.0

Process finished with exit code 0

```

Dapat dilihat hasil running program, ketika dilakukan pemanggilan method `getNama()` hasilnya hal ini terjadi karena atribut nama belum diset nilai defaultnya. Hal ini dapat ditangani dengan membuat kontruktor.

3. Ubah class Anggota menjadi seperti berikut

```
1 package koperasigettersetter;  
2 public class Anggota {  
3     private String nama;  
4     private String alamat;  
5     private float simpanan;  
6  
7     Anggota(String nama, String alamat){  
8         this.nama = nama;  
9         this.alamat = alamat;  
10        this.simpanan = 0;  
11    }  
12  
13    public void setNama(String nama){  
14        this.nama = nama;  
15    }  
16    public void setAlamat(String alamat){  
17        this.alamat = alamat;  
18    }  
19    public String getNama(){  
20        return nama;  
21    }  
22    public String getAlamat(){  
23        return alamat;  
24    }  
25    public float getSimpanan(){  
26        return simpanan;  
27    }  
28    public void setor(float uang){  
29        simpanan +=uang;  
30    }  
31    public void pinjam(float uang){  
32        simpanan -=uang;  
33    }  
34 }  
35
```

```
public class Anggota {private String nama;  
  
    private String alamat;  
  
    private float simpanan;  
  
    Anggota(String nama, String alamat) {  
  
        this.nama = nama;  
  
        this.alamat = alamat;
```



```
        this.simpanan = 0;

    }

    public void setName(String nama) {

        this.nama = nama;

    }

    public void setAddress(String alamat) {

        this.alamat = alamat;

    }

    public String getName() {

        return nama;

    }

    public String getAddress() {

        return alamat;

    }

    public float getSimpanan() {

        return simpanan;

    }

    public void setOr(float uang) {
```

```

        simpanan += uang;

    }

    public void pinjam(float uang) {

        simpanan -= uang;

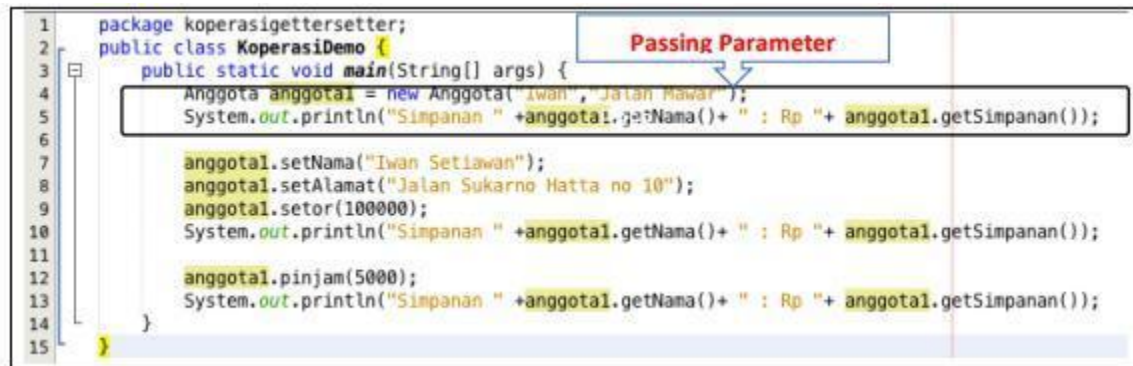
    }

}

```

Pada class Anggota dibuat konstruktor dengan access modifier default yang memiliki 2 parameter nama dan alamat. Dan didalam konstruktor tersebut dipastikan nilai simpanan untuk pertama kali adalah Rp. 0

4. Selanjutnya ubah class KoperasiDemo sebagai berikut



```

1 package koperasigettersetter;
2 public class KoperasiDemo {
3     public static void main(String[] args) {
4         Anggota anggota1 = new Anggota("Iwan", "Jalan Mawar");
5         System.out.println("Simpanan " + anggota1.getNama() + " : Rp " + anggota1.getSimpanan());
6
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 }

```

```

public class Main {
    public static void main(String[] args) {

        Anggota anggota1 = new Anggota("Iwan", "Jalan Mawar");

        System.out.println("Simpanan " + anggota1.getNama() + " : Rp " +
        anggota1.getSimpanan());

        anggota1.setNama("Iwan Setiawan");
    }
}

```

```

        anggota.setAlamat("Jalan Soekarno Hatta no 10"); anggota.setor(100000);

        System.out.println("Simpanan " + anggota.getNama() + " Rp " +
        anggota.getSimpanan());

        anggota.pinjam(5000);

        System.out.println("Simpanan " + anggota.getNama() + " Rp " +
        anggota.getSimpanan());

    }
}

```

5. Hasil dari program tersebut adalah sebagai berikut

```

run:
Simpanan Iwan : Rp 0.0
Simpanan Iwan Setiawan : Rp 100000.0
Simpanan Iwan Setiawan : Rp 95000.0
BUILD SUCCESSFUL (total time: 0 seconds)

```

```

D:\src>java -Duser.encoding=utf-8 -classpath E:\
Simpanan Iwan : Rp 0.0
Simpanan Iwan Setiawan Rp 100000.0
Simpanan Iwan Setiawan Rp 95000.0

Process finished with exit code 0

```

Setelah menambah konstruktor pada class Anggoata maka atribut nama dan alamat secara otomatis harus diset terlebih dahulu dengan melakukan passing parameter jika melakukan instansiasi class Anggota. Hal ini biasa dilakukan untuk atribut yang membutuhkan nilai yang spesifik. Jika tidak membutuhkan nilai spesifik dalam konstruktor tidak perlu parameter. Contohnya simpanan untuk anggota baru diset 0, maka simpanan tidak perlu untuk dijadikan parameter pada konstruktor.

3.6 Pertanyaan – Percobaan 3 dan 4

1. Apa yang dimaksud getter dan

setter? Jawab:

Getter merupakan sebuah aksi saat kita mengambil sebuah nilai/values dari suatu variable/object sedangkan setter adalah sebuah aksi saat kita memasukan sebuah nilai/values kedalam suatu variable/object.

2. Apa kegunaan dari method

getSimpanan()? Jawab:

getSimpanan() digunakan untuk memasukkan nilai simpanan (setting)

3. Method apa yang digunakan untuk menambah saldo?

Jawab:

setor()

4. Apa yang dimaksud

konstruktor? Jawab:

Suatu method yang akan memberikan nilai awal pada saat suatu objek dibuat.

5. Sebutkan aturan dalam membuat konstruktor?

Jawab:

Nama konstruktor harus sama dengan nama kelas, pada sebuah class hanya terdapat satu konstruktor, konstruktor harus bersifat public.

6. Apakah boleh konstruktor bertipe

private? Jawab:

Tidak diperbolehkan, karena konstruktor tidak dapat diakses dari luar class tersebut apabila dibuat private.

7. Kapan menggunakan parameter dengan passing parameter?

Jawab

Passing parameter digunakan pada saat setelah menambahkan konstruktor untuk menambahkan nilai spesifik dan ketika sebuah method membutuhkan nilai spesifik.

8. Apa perbedaan atribut class dan instansiasi

atribut? Jawab:

Atribut class merupakan atribut yang berada didalam atribut class, sedangkan instansiasi atribut adalah atribut yang dimiliki oleh sebuah objek dalam melakukan instansiasinya.

9. Apa perbedaan class method dan instansiasi

method? Jawab:

Class method adalah method yang berada pada suatu class dan belum dilakukan sedangkan instansiasi method adalah proses pemanggilan method setelah dilakukan instansiasi objek.

4. Kesimpulan

Dari percobaan diatas, telah dipelajari kosep dari enkapsulasi, konstruktor, access modifier yang terdiri dari 4 jenis yaitu public, protected, default dan private. Konsep atribut atau method class yang ada di dalam blok code class dan konsep instansiasi atribut atau method. Cara penggunaan getter dan setter beserta fungsi dari getter dan setter. Dan juga telah dipelajari atau memahami notasi UML.

5. Tugas

1. Cobalah program dibawah ini dan tuliskan hasil outputnya

```

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;
        }
    }
}

```

```

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());
    }
}

```

jawab

EncapDemo.java

```

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;

        }

    }

}

```

EncapTest.java

```

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());

    }

}

```

```
}  
  
}
```

Output

```
-Dsun.stderr.encoding=UTF-8 -classpath .  
Name : James  
Age : 30  
  
Process finished with exit code 0
```

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

Jawab :

Hal tersebut dikarenakan adanya if else pada saat setAge yang memiliki argument jika newAge (inputan age) > 30 maka nilai age akan 30. Dikarenakan inputan 35 maka umur yang digunakan sistem adalah 30 sama dengan kondisi tersebut.

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

18. Jawab:

EncapDemo.java

```
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) {

            System.out.println("Umur melebihi batas maksimal. Batas maksimal Tahun");
        } else if (newAge < 18) {

            System.out.println("Umur kurang dari batas minimal. Batas minimal Tahun");
        } else {

            age = newAge;

        }

    }

}

```

EncapTest.java

```

public class EncapTest {

```

```
public static void main(String[] args) { EncapDemo encap = new EncapDemo();
EncapDemo encap2 = new EncapDemo(); EncapDemo encap3 = new EncapDemo();

    encap.setName("James");

    encap.setAge(17);

    System.out.println("Name : " + encap.getName());

    System.out.println("Age : " + encap.getAge());

    encap2.setName("Dhayu");

    encap2.setAge(33);

    System.out.println("Name : " + encap2.getName());

    System.out.println("Age : " + encap2.getAge());

    encap3.setName("uzi");

    encap3.setAge(27);

    System.out.println("Name : " + encap3.getName());

    System.out.println("Age : " + encap3.getAge());

}
}
```

Output

```
Umur kurang dari batas minimal. Batas minimal Tahun
Name : James
Age : 0
Umur melebihi batas maksimal. Batas maksimal Tahun
Name : Dhayu
Age : 0
Name : Wildan
Age : 27

Process finished with exit code 0
```

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

```
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("Limit Pinjaman: " + donny.getLimitPinjaman());

        System.out.println("\nMeminjam uang 10.000.000...");
        donny.pinjam(10000000);
        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());
    }
}
```

Hasil yang diharapkan

```
D:\MyJava>javac TestKoperasi.java

D:\MyJava>java TestKoperasi
Nama Anggota: Donny
Limit Pinjaman: 5000000

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

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
```

Anggota.java

```
public class Anggota {

    private String noKTP, nama;

    private int limitPinjaman, jumlahPinjaman;

    Anggota(String noKTP, String nama, int limitPinjaman) { this.noKTP = noKTP;

        this.nama = nama;

        this.limitPinjaman = limitPinjaman;

    }

    public String getNama() {

        return nama;

    }

    public int getLimitPinjaman() {
```

```

        return limitPinjaman;

    }

    public void pinjam(int pinjaman) {

        if (jumlahPinjaman + pinjaman < limitPinjaman) { jumlahPinjaman += pinjaman;

        } else {

            System.out.println("Maaf, jumlah pinjaman melebihi limit");

        }

    }

    public int getJumlahPinjaman() {

        return jumlahPinjaman;

    }

    public void angsur(int angsuran) {

        jumlahPinjaman -= angsuran;

    }

}

```

Testkoperasi.java

```

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("Limit Pinjaman : " + donny.getLimitPinjaman());
System.out.println("\nMeminjam uang 10000000"); donny.pinjam(10000000);

        System.out.println("Jumlah Pinjaman saat ini : " + donny.getJumlahPinjaman());

        System.out.println("\nMeminjam uang 4000000");

        donny.pinjam(4000000);

        System.out.println("Jumlah Pinjaman saat ini : " +

                donny.getJumlahPinjaman());

        System.out.println("Membayar angsuran 10000000");

        donny.angsur(10000000);

        System.out.println("Jumlah Pinjaman saat ini : " +

                donny.getJumlahPinjaman());

        System.out.println("Membayar angsuran 3000000");

        donny.angsur(3000000);

        System.out.println("Jumlah Pinjaman saat ini : " +

                donny.getJumlahPinjaman());

    }
}
```

Output

```
-Dsun.stderr.encoding=UTF-8 -classpath E:\KULIAH\oop\koperasigettersetter\c
Nama Anggota : uzi
Limit Pinjaman : 5000000

Meminjam uang 10000000
Maaf, jumlah pinjaman melebihi limit
Jumlah Pinjaman saat ini : 0

Meminjam uang 4000000
Jumlah Pinjaman saat ini : 4000000
Membayar angsuran 10000000
Jumlah Pinjaman saat ini : 3000000
Membayar angsuran 3000000
Jumlah Pinjaman saat ini : 0

Process finished with exit code 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”.

```
public class Anggota {

    private String noKTP, nama;

    private int limitPinjaman, jumlahPinjaman;

    Anggota(String noKTP, String nama, int limitPinjaman) { this.noKTP = noKTP;

        this.nama = nama;

        this.limitPinjaman = limitPinjaman;

    }

    public String getNama() {

        return nama;

    }

}
```

```
}

public int getLimitPinjaman() {

    return limitPinjaman;

}

public void pinjam(int pinjaman) {

    if (jumlahPinjaman + pinjaman < limitPinjaman) { jumlahPinjaman += pinjaman;

    } else {

        System.out.println("Maaf, jumlah pinjaman melebihi limit");

    }

}

public int getJumlahPinjaman() {

    return jumlahPinjaman;

}

public void angsur(int angsuran) {

    int minAngsuran = jumlahPinjaman * 10 / 100; if (angsuran > minAngsuran) {

        jumlahPinjaman -= angsuran;

    } else {

        System.out.println("Maaf, angsuran harus 10% dari jumlah pinjaman");

    }

}
```



```
}  
  
}
```

Testkoperasi.java

```
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("Limit Pinjaman : " + donny.getLimitPinjaman());  
        System.out.println("\nMeminjam uang 10000000"); donny.pinjam(10000000);  
  
        System.out.println("Jumlah Pinjaman saat ini : " + donny.getJumlahPinjaman());  
  
        System.out.println("\nMeminjam uang 4000000");  
  
        donny.pinjam(4000000);  
  
        System.out.println("Jumlah Pinjaman saat ini : " +  
  
            donny.getJumlahPinjaman());  
  
        System.out.println("Membayar angsuran 1000000");  
  
        donny.angsur(1000000);  
  
        System.out.println("Jumlah Pinjaman saat ini : " +
```

```

        donny.getJumlahPinjaman());

        System.out.println("Membayar angsuran 1000");

        donny.angsur(1000);

        System.out.println("Jumlah Pinjaman saat ini : " +

                donny.getJumlahPinjaman());

        System.out.println("Membayar angsuran 3000000");

        donny.angsur(3000000);

        System.out.println("Jumlah Pinjaman saat ini : " +

                donny.getJumlahPinjaman());

    }

}

```

Output

```

-DSON.Stderr.encoding=UTF-8 -classpath E:\KULIAH\OOD\KRO
Nama Anggota : Donny
Limit Pinjaman : 5000000

Meminjam uang 10000000
Maaf, jumlah pinjaman melebihi limit
Jumlah Pinjaman saat ini : 0

Meminjam uang 4000000
Jumlah Pinjaman saat ini : 4000000
Membayar angsuran 1000000
Jumlah Pinjaman saat ini : 3000000
Membayar angsuran 1000
Maaf, angsuran harus 10% dari jumlah pinjaman
Jumlah Pinjaman saat ini : 3000000
Membayar angsuran 3000000
Jumlah Pinjaman saat ini : 0

Process finished with exit code 0
|

```

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

Jawab:

Anggota.java

```
public class Anggota {  
  
    private String noKTP, nama;  
  
    private int limitPinjaman, jumlahPinjaman;  
  
    Anggota(String noKTP, String nama, int limitPinjaman) { this.noKTP = noKTP;  
  
        this.nama = nama;  
  
        this.limitPinjaman = limitPinjaman;  
  
    }  
  
    public String getNama() {  
  
        return nama;  
  
    }  
  
    public int getLimitPinjaman() {  
  
        return limitPinjaman;  
  
    }  
  
    public void pinjam(int pinjaman) {  
  
        if (jumlahPinjaman + pinjaman <= limitPinjaman) { jumlahPinjaman += pinjaman;  

```

```

    } else {

        System.out.println("Maaf, jumlah pinjaman melebihi limit");

    }

}

public int getJumlahPinjaman() {

    return jumlahPinjaman;

}

public void angsur(int angsuran) {

    int minAngsuran = jumlahPinjaman * 10 / 100; if (angsuran > minAngsuran) {

        jumlahPinjaman -= angsuran;

    } else {

        System.out.println("Maaf, angsuran harus 10% dari jumlah pinjaman");

    }

}

}

```

Testkoperasi.java

```

import java.util.Scanner;;

public class TestKoperasi {

```

```
public static void main(String[] args) {

    Anggota donny = new Anggota("111333444", "Donny", 5000000); Scanner sc = new
Scanner(System.in);

    System.out.println("Nama Anggota : " + donny.getNama());

    System.out.println("Limit Pinjaman : " + donny.getLimitPinjaman());

    System.out.print("\nInput Pinjaman : "); donny.pinjam(sc.nextInt());

    System.out.println("Jumlah Pinjaman saat ini : " + donny.getJumlahPinjaman());

    System.out.print("\nInput Pinjaman : ");

    donny.pinjam(sc.nextInt());

    System.out.println("Jumlah Pinjaman saat ini : " +

        donny.getJumlahPinjaman());

    System.out.print("Input Angsuran : ");

    donny.angsur(sc.nextInt());

    System.out.println("Jumlah Pinjaman saat ini : " + donny.getJumlahPinjaman());

    System.out.print("Input Angsuran : ");

    donny.angsur(sc.nextInt());

    System.out.println("Jumlah Pinjaman saat ini : " +

        donny.getJumlahPinjaman());

    System.out.print("Input Angsuran : ");

    donny.angsur(sc.nextInt());
```

```
        System.out.println("Jumlah Pinjaman saat ini : " + donny.getJumlahPinjaman());  
    }  
}
```

Output

```
Nama Anggota : Donny  
Limit Pinjaman : 5000000  
  
Input Pinjaman : 7000000  
Maaf, jumlah pinjaman melebihi limit  
Jumlah Pinjaman saat ini : 0  
  
Input Pinjaman : 5000000  
Jumlah Pinjaman saat ini : 5000000  
Input Angsuran : 300000  
Maaf, angsuran harus 10% dari jumlah pinjaman  
Jumlah Pinjaman saat ini : 5000000  
Input Angsuran : 2000  
Maaf, angsuran harus 10% dari jumlah pinjaman  
Jumlah Pinjaman saat ini : 5000000  
Input Angsuran : 2000000  
Jumlah Pinjaman saat ini : 3000000  
  
Process finished with exit code 0
```