



NAMA : Abdulah Syahrany Kurniawan
NIM : 2041720037
KELAS : TI-2C
MATERI : Enkapsulasi Pada Pemrograman Berorientasi Objek

Percobaan 2

```
Output - Jobsheet3Rony (run) x
run:
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
BUILD SUCCESSFUL (total time: 0 seconds)
```

Pertanyaan

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

Jawab : Karena belum memanggil method nyalakan mesin, dan hasilnya saat di run maka muncul kecepatan tidak bisa bertambah karenan mesin off

2. Mengapat atribut kecepatan dan kontakOn diset private?

Jawab : Agar dapat diakses didalam kelas yang sama

3. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

Jawab :

```
public void tambahKecepatan(){
    if(kontakOn == true){
        if(kecepatan<=100){
            kecepatan += 5;
        }
        else{
            System.out.println("Kecepatan Maksimal adalah 100");
        }
    }
    else{
        System.out.println("Kecepatan tidak bisa bertambah karena Mesin Off!!!");
    }
}
```

Percobaan 3 dan Percobaan 4

```
Output - Jobsheet3Rony (run) x
run:
Simpanan Rony : Rp 0.0
Simpanan Rony : Rp 100000.0
Simpanan Rony : Rp 95000.0
BUILD SUCCESSFUL (total time: 0 seconds)
```



NAMA : Abdulah Syahrony Kurniawan
NIM : 2041720037
KELAS : TI-2C
MATERI : Enkapsulasi Pada Pemrograman Berorientasi Objek

Pertanyaan

1. Apa yang dimaksud getter dan setter?

Jawab : Getter adalah public method dan memiliki tipe data return, yang berfungsi untuk mendapatkan nilai dari atribut private. Sedangkan setter adalah public method yang tidak memiliki tipe data return, yang berfungsi untuk memanipulasi nilai dari atribut private.

2. Apa kegunaan dari method getSimpanan()?

Jawab : untuk menampilkan simpanan

3. Method apa yang digunakan untuk menambah saldo?

Jawab : method setor

4. Apa yang dimaksud konstruktor?

Jawab : konstruktor adalah suatu method yang akan memberikan nilai awal **pada** saat suatu objek dibuat.

5. Sebutkan aturan dalam membuat konstruktor?

Jawab : konstruktor tidak boleh memiliki tipe kembalian

6. Apakah boleh konstruktor bertipe private?

Jawab : boleh

7. Kapan menggunakan parameter dengan passing parameter?

Jawab : Setelah menambah konstruktor pada class Anggota maka atribut nama dan alamat secara otomatis harus diset terlebih dahulu dengan melakukan passing parameter jika melakukan instansiasi class Anggota.

8. Apa perbedaan atribut class dan instansiasi atribut?

Atribut class adalah objek, dan instansiasi atribut adalah proses pembuatan objek yang sudah ada, atau yang biasa disebut mendeklarasikan variabel pada objek

9. Apa perbedaan class method dan instansiasi method?

Jawab : Perbedaan method class dengan instansiasi method, kalau class tidak perlu instansiasi objek untuk mengaksesnya. Kalau method instansiasi harus membuat objek terlebih dahulu



NAMA : Abdulah Syahrony Kurniawan
NIM : 2041720037
KELAS : TI-2C
MATERI : Enkapsulasi Pada Pemrograman Berorientasi Objek

TUGAS

1. Cobalah program dibawah ini dan tuliskan hasil outputnya

```
run:
Name : Rony
Age : 30
BUILD SUCCESSFUL (total time: 0 seconds)
```

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 pada proses pemilihan saya memberi perintah Ketika newAge melebihi 30 maka hasilnya akan 30

3. Ubah program diatas agar atribut age dapat diberi nilai maksimal 30 dan minimal 18.

Jawab :

```
run:
Name : Rony
Age : 18
Name : Rony
Age : 30
Name : Rony
Age : 21
BUILD SUCCESSFUL (total time: 0 seconds)

    }
    public void setAge(int newAge) {
        if(newAge > 30) {
            age = 30;
        }
        else if(newAge <= 18) {
            age = 18;
        }
        else{
            age = newAge;
        }
    }
}
```



NAMA : Abdulah Syahrony Kurniawan
NIM : 2041720037
KELAS : TI-2C
MATERI : Enkapsulasi Pada Pemrograman Berorientasi Objek

```
// TODO code application logic here
EncapDemo encap = new EncapDemo();
EncapDemo encap1 = new EncapDemo();
EncapDemo encap2 = new EncapDemo();

encap.setName("Rony");
encap.setAge(17);
System.out.println("Name : " + encap.getName());
System.out.println("Age : " + encap.getAge());

encap1.setName("Rony");
encap1.setAge(35);
System.out.println("Name : " + encap1.getName());
System.out.println("Age : " + encap1.getAge());

encap2.setName("Rony");
encap2.setAge(21);
System.out.println("Name : " + encap2.getName());
System.out.println("Age : " + encap2.getAge());
```

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.

```
5  L  */
6  package Tugas;
7
8  /**
9   *
10  * @author LENOVO
11  */
12  public class Anggota {
13      String nama, nomorKtp;
14      int limitPinjam, jumlahPinjam;
15
16      Anggota(String nomorKtp, String nama, int limitPinjam){
17          this.nomorKtp = nomorKtp;
18          this.nama = nama;
19          this.limitPinjam = limitPinjam;
20      }
21
22      String getNama(){
23          return nama;
24      }
25      int getLimitPinjam(){
26          return limitPinjam;
27      }
28      int getJumlahPinjam(){
29          return jumlahPinjam;
30      }
31
32      public void pinjam(int pinjam){
33          if(jumlahPinjam + pinjam <= limitPinjam){
34              jumlahPinjam += pinjam;
35          }
36          else{
37              System.out.println("Maaf, Jumlah pinjaman melebihi limit!!!");
38          }
39      }
40
41      public void angsur(int angsur){
42          jumlahPinjam -= angsur;
43      }
44  }
```



NAMA : Abdulah Syahrony Kurniawan
NIM : 2041720037
KELAS : TI-2C
MATERI : Enkapsulasi Pada Pemrograman Berorientasi Objek

```
11  L  */
12  public class testKoperasi {
13
14  L  /**
15  L  * @param args the command line arguments
16  L  */
17  L  public static void main(String[] args) {
18  L  // TODO code application logic here
19  L  Anggota rony = new Anggota("111333444", "Rony", 5000000);
20
21  L  System.out.println("Nama Anggota: " + rony.getNama());
22  L  System.out.println("Limit Pinjaman: " + rony.getLimitPinjam());
23  L  System.out.println("\nMeminjam uang 10.000.000...");
24  L  rony.pinjam(10000000);
25  L  System.out.println("Jumlah pinjaman saat ini: " + rony.getJumlahPinjam());
26  L  System.out.println("\nMeminjam uang 4.000.000...");
27  L  rony.pinjam(4000000);
28  L  System.out.println("Jumlah pinjaman saat ini: " + rony.getJumlahPinjam());
29  L  System.out.println("\nMembayar angsuran 1.000.000");
30  L  rony.angsur(1000000);
31  L  System.out.println("Jumlah pinjaman saat ini: " + rony.getJumlahPinjam());
32  L  System.out.println("\nMembayar angsuran 3.000.000");
33  L  rony.angsur(3000000);
34  L  System.out.println("Jumlah pinjaman saat ini: " + rony.getJumlahPinjam());
35
36  L  }
37
```

Output - Jobsheet3Rony (run) X

```
run:
Nama Anggota: Rony
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
BUILD SUCCESSFUL (total time: 0 seconds)
```



NAMA : Abdulah Syahröny Kurniawan

NIM : 2041720037

KELAS : TI-2C

MATERI : Enkapsulasi Pada Pemrograman Berorientasi Objek

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(int angsur){
2     if(angsur > (0.01 * jumlahPinjam)){
3         jumlahPinjam -= angsur;
4     }
5     else{
6         System.out.println("Maaf, Angsuran harus 10% dari jumlah pinjaman");
7     }
8 }
9 }
```

```
run:
Nama Anggota: Rony
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
Maaf, Angsuran harus 10% dari jumlah pinjaman
Jumlah pinjaman saat ini: 3000000
BUILD SUCCESSFUL (total time: 0 seconds)
```

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

```
run:
Nama Anggota: Rony
Limit Pinjaman: 5000000
Meminjam Uang : 2000000
Jumlah pinjaman saat ini: 2000000
Membayar Angsuran : 400
Maaf, Angsuran harus 10% dari jumlah pinjaman
Jumlah pinjaman saat ini: 2000000
Membayar Angsuran : 1000000
Jumlah pinjaman saat ini: 1000000
BUILD SUCCESSFUL (total time: 1 minute 12 seconds)
```



NAMA : Abdulah Syahrony Kurniawan
NIM : 2041720037
KELAS : TI-2C
MATERI : Enkapsulasi Pada Pemrograman Berorientasi Objek

```
17  
18 public static void main(String[] args) {  
19     // TODO code application logic here  
20     Scanner input = new Scanner(System.in);  
21     int jumlahPinjam, jumlahAngsur;  
22     Anggota rony = new Anggota("111333444", "Rony", 5000000);  
23  
24     System.out.println("Nama Anggota: " + rony.getNama());  
25     System.out.println("Limit Pinjaman: " + rony.getLimitPinjam());  
26  
27     System.out.print("Meminjam Uang : ");  
28     jumlahPinjam = input.nextInt();  
29     rony.pinjam(jumlahPinjam);  
30     System.out.println("Jumlah pinjaman saat ini: " + rony.getJumlahPinjam());  
31  
32     System.out.print("Membayar Angsuran : ");  
33     jumlahAngsur = input.nextInt();  
34     rony.angsur(jumlahAngsur);  
35     System.out.println("Jumlah pinjaman saat ini: " + rony.getJumlahPinjam());  
36  
37     System.out.print("Membayar Angsuran : ");  
38     jumlahAngsur = input.nextInt();  
39     rony.angsur(jumlahAngsur);  
40     System.out.println("Jumlah pinjaman saat ini: " + rony.getJumlahPinjam());  
41  
42 }  
43  
44 }  
45
```