



NAMA : OKTAVIAN VANDI TRI SHAKTI

NIM : 2041720208

KELAS : 2C

Pertanyaan percobaan 2

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

Jawab:

Karena pada class MotorDemo, method yang dipanggil pertama kali langsung method tambahKecepatan(). Seharusnya sebelum itu, kita harus memanggil method nyalakanMesin(). Pada method nyalakanMesin() kontakOn diset bernilai true.

Kemudian pada method tambahKecepatan, jika kontakOn bernilai false maka akan tampil “Kecepatan tidak bisa bertambah karena Mesin Off!”. Untuk itu, method yang dipanggil sebelum tambahKecepatan() pertama kalinya, itu seharusnya method nyalakanMesin().

2. Mengapa atribut kecepatan dan kontakOn diset private?

Jawab:

Karena atribut kecepatan dan kontakOn tidak bisa diakses oleh pengguna dan diganti nilainya secara sembarangan. Ketika kita ingin mencoba menambah kecepatan saat posisi kontak masih OFF, maka akan muncul notifikasi bahwa mesin OFF. Untuk mengakses atau manipulasi atribut kecepatan dan kontakOn hanya bisa melalui method tambahKecepatan() dan kurangiKecepatan().

3. Ubah class Motor sehingga kecepatan maksimalnya adalah 150!

Jawab:

```
12 public void tambahKecepatan() {
13     if(kontakOn==true) {
14         kecepatan +=5;
15         if(kecepatan <=150) {
16             kecepatan +=5;
17         }else{
18             System.out.println("Kecepatan maksimal adalah 150");
19         }
20     }else{
21         System.out.println("Kecepatan tidak bisa bertambah karena mesin off! \n");
22     }
23 }
```



NAMA : OKTAVIAN VANDI TRI SHAKTI

NIM : 2041720208

KELAS : 2C

Pertanyaan – Percobaan 3 dan 4

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.
- 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 menyimpan, menampilkan dan mengupdate jumlah simpanan.

3. Method apa yang digunakan untuk menambah saldo?

Jawab:

Method setor(float uang);

4. Apa yang dimaksud konstruktor?

Jawab:

Konstruktor mirip dengan method cara deklarasinya akan tetapi tidak memiliki tipe return dan konstruktor dieksekusi ketika instansiasi dari objek dibuat. Jadi setiap kali sebuah objek dibuat dengan keyword new() maka konstruktor akan dieksekusi.

5. Sebutkan aturan dalam membuat konstruktor?

Jawab:

- Nama konstruktor harus sama dengan nama class
- Konstruktor tidak memiliki tipe data return
- Konstruktor tidak boleh menggunakan modifier abstract, static, final, dan synchronized

6. Apakah boleh konstruktor bertipe private?

Jawab:

Boleh, tetapi hasilnya atribut/method tidak akan bisa diakses oleh class lain.

7. Kapan menggunakan parameter dengan passing parameter?

Jawab:

Passing parameter dilakukan saat instansiasi class. Hal ini biasa dilakukan untuk atribut yang membutuhkan nilai yang spesifik. Jika tidak membutuhkan nilai spesifik dalam konstruktor tidak perlu parameter.

8. Apa perbedaan atribut class dan instansiasi atribut?

Jawab:

Pada atribut class, nilai antar object pasti sama dan untuk mengaksesnya tidak perlu instansiasi object. Sedangkan pada instansiasi atribut, nilai antar objectnya bisa berbeda-beda dan untuk mengaksesnya perlu dilakukan instansiasi object terlebih dahulu.



NAMA :OKTAVIAN VANDI TRI SHAKTI
NIM : 2041720208
KELAS : 2C

9. Apa perbedaan class method dan instansiasi method?

Jawab:

Pada class method, nilai antar object pasti sama dan untuk mengaksesnya tidak perlu instansiasi object. Sedangkan pada intansiasi method, nilai antar objectnya bisa berbeda-beda dan untuk mengaksesnya perlu dilakukan instansiasi object terlebih dahulu.

Tugas

1.

```
Source History
1 package Tugas;
2
3 public class EncapDemo {
4
5     private String name;
6     private int age;
7
8     public String getName() {
9         return name;
10    }
11
12    public void setName(String newName) {
13        name = newName;
14    }
15
16    public int getAge() {
17        return age;
18    }
19
20    public void setAge(int newAge) {
21        if (newAge > 30) {
22            age = 30;
23        } else {
24            age = newAge;
25        }
26    }
27
28 }
```

Tugas.EncapDemo 27:2 INS Windows (CRLF)

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

15:1 INS Windows (CRLF)



NAMA : OKTAVIAN VANDI TRI SHAKTI
NIM : 2041720208
KELAS : 2C

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Encap ---  
Name : James  
Age : 30  
-----  
BUILD SUCCESS  
-----
```

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 class EncapDemo pada method setAge, terdapat pemilihan jika age yang dimasukkan lebih dari 30 maka nilai age = 30. Karena itulah walaupun pada class EncapTest age yang diset adalah 35, tetapi pada output yang ditampilkan nilainya adalah 30.

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

```
20 public void setAge(int newAge) {  
21     if (newAge >= 18 && newAge <= 30) {  
22         age = newAge;  
23     } else {  
24         System.out.println("Usia minimal harus 15 dan maksimal 30 tahun ");  
25     }  
26 }  
27 }
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Encap ---  
Usia minimal harus 15 dan maksimal 30 tahun  
Name : James  
Age : 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



NAMA :OKTAVIAN VANDI TRI SHAKTI
NIM : 2041720208
KELAS : 2C

```
Source History
private String noKtp;
private String nama;
private int limitPinjaman;
private int 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 int getJumlahPinjaman(){
    return jumlahPinjaman;
}

public void pinjam(int pinjam){
    if ((jumlahPinjaman + pinjam) <= limitPinjaman){
        jumlahPinjaman += pinjam;
    }else {
        System.out.println("Maaf, jumlah pinjaman melebihi batas");
    }
}

public void angsur(int angsur){
    jumlahPinjaman -= angsur;
}
```

```
Source History
package Tugas;

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());
    }
}
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Encap ---
Nama Anggota: Donny
Limit Pinjaman: 5000000

Meminjam uang 10.000.000...
Maaf, jumlah pinjaman melebihi batas
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 SUCCESS
```



NAMA :OKTAVIAN VANDI TRI SHAKTI
NIM : 2041720208
KELAS : 2C

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".

```
34 public void angsur(int angsur){  
35     if (angsur > (0.10 * jumlahPinjaman)){  
36         jumlahPinjaman -= angsur;  
37     }else {  
38         System.out.println("Maaf, angsuran harus 10% dari jumlah pinjaman");  
39     }  
40 }  
41 }
```

```
Nama Anggota: Donny  
Limit Pinjaman: 5000000  
  
Meminjam uang 10.000.000...  
Maaf, jumlah pinjaman melebihi batas  
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 SUCCESS
```

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

```
Source History  
1 package Tugas;  
2 import java.util.Scanner;  
3 public class TestKoperasi {  
4  
5     public static void main(String[] args) {  
6         Scanner sc = new Scanner(System.in);  
7         int jumlahPinjam, jumlahAngsur;  
8         Anggota donny = new Anggota("111333444", "Donny", 5000000);  
9  
10        System.out.println("Nama Anggota: " + donny.getNama());  
11        System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());  
12  
13        System.out.println("\nMeminjam uang");  
14        jumlahPinjam = sc.nextInt();  
15        donny.pinjam(jumlahPinjam);  
16        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());  
17  
18        System.out.println("\nMeminjam uang");  
19        jumlahPinjam = sc.nextInt();  
20        donny.pinjam(jumlahPinjam);  
21        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());  
22        System.out.println("\nMembayar Angsuran");  
23        jumlahAngsur = sc.nextInt();  
24        donny.angsur(jumlahAngsur);  
25        System.out.println("Jumlah Pinjaman saat ini: " + donny.getJumlahPinjaman());  
26  
27        System.out.println("\nMembayar angsuran");  
28        jumlahAngsur = sc.nextInt();  
29        donny.angsur(jumlahAngsur);  
30        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());  
31    }  
32 }  
33  
Transferring Maven repository index: Central Repository 9% 33:1 INS Windows (CRLF)
```



NAMA :OKTAVIAN VANDI TRI SHAKTI
NIM : 2041720208
KELAS : 2C

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Encap ---  
Nama Anggota: Donny  
Limit Pinjaman: 5000000  
  
Meminjam uang  
10000000  
Maaf, jumlah pinjaman melebihi batas  
Jumlah pinjaman saat ini: 0  
  
Meminjam uang  
4000000  
Jumlah pinjaman saat ini: 4000000  
  
Membayar Angsuran  
1000000  
Jumlah Pinjaman saat ini: 3000000  
  
Membayar angsuran  
3000000  
Jumlah pinjaman saat ini: 0  
-----  
BUILD SUCCESS  
-----
```