

Tugas Praktikum 3

Enkapsulasi



Rafi Ody Prasetyo
(2341720180)

D-IV Teknik Informatika
Politeknik Negeri Malang
Semester 3
2024

Percobaan 1 & 2

Code:

```
package Jobsheet3.MotorEncapsulation;

public class Motor {
    // Percobaan 1:
    // public int kecepatan = 0;
    // public boolean kontakOn = false;

    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
karenam Mesin off! \n");
        }
    }

    public void kurangiKecepatan() {
        if (kontakOn == true) {
            kecepatan -= 5;
        } else {
            System.out.println("Kecepatan tidak bisa berkurang
karena mesin off! \n");
        }
    }

    public void printStatus() {
        if (kontakOn == true) {
            System.out.println("Kontak On");
        } else {
            System.out.println("Kontak Off");
        }
        System.out.println("Kecepatan: "+ kecepatan + "\n");
    }
}
```

```

package Jobsheet3.MotorEncapsulation;
public class MotorDemo {
    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();
    }
}

```

Output:

```

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

Jawab:

Karena dalam method tambahKecepatan() terdapat statement if dengan syarat apabila kontakOn == True maka kecepatan dapat ditambahkan, namun apabila program tidak memenuhi syarat tersebut maka kecepatan tidak dapat bertambah karena kontakOn == False.

2. Mengapat atribut kecepatan dan kontakOn diset private?

Jawab:

Karena variabel tersebut hanya dibutuhkan pada class Motor.

3. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

Jawab:

Code:

```
public void tambahKecepatan() {  
    if (kontakOn == true) {  
        if (kecepatan >= 100) {  
            System.out.println("Kecepatan sudah  
maksimal! \n");  
        } else {  
            kecepatan += 50;  
        }  
    } else {  
        System.out.println("Kecepatan tidak bisa  
bertambah karenam Mesin off! \n");  
    }  
}
```

Pada method tambahKecepatan() ditambahkan nested if agar ketika kecepatan mencapai 100 program akan mengeksekusi print kecepatan sudah mencapai maksimal.

Percobaan 3 & 4

Code:

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

```

public class KoperasiDemo {

    public static void main(String[] args) {
        Anggota anggota1 = new Anggota("Iwan", "Jalan Rawar");
        System.out.println("Simpanan "+ anggota1.getNama() + " :
Rp. "+ anggota1.getSimpanan());
        anggota1.setNama("Iwan Setiawan");
        anggota1.setAlamat("Jalan Sukarno 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());
    }

}

```

Output:

```

Simpanan Iwan : Rp. 0.0
Simpanan Iwan Setiawan : Rp. 100000.0
Simpanan Iwan Setiawan : Rp. 95000.0

```

Pertanyaan 3 & 4

1. Apa yang dimaksud getter dan setter?

Jawab:

Getter merupakan method yang digunakan untuk mengambil nilai dari variabel, sedangkan setter merupakan method yang digunakan untuk mengatur atau mengubah nilai dari variabel.

2. Apa kegunaan dari method getSimpanan()?

Jawab:

Method yang digunakan untuk mengambil nilai variabel simpanan.

3. Method apa yang digunakan untuk menambah saldo?

Jawab:

Method setor() dapat digunakan untuk menambah saldo.

4. Apa yang dimaksud konstruktor?

Jawab:

Method default yang sudah ada sejak class terbuat.

5. Sebutkan aturan dalam membuat konstruktor?

Jawab:

- Nama konstruktor dibuat sama dengan nama class.
- Tidak ada nilai kembalian pada konstruktor.
- Konstruktor default dibuat otomatis.

6. Apakah boleh konstruktor bertipe private?

Jawab:

Boleh, konstruktor privat akan ada untuk mencegah kelas tersebut dibuat.

7. Kapan menggunakan parameter dengan passing parameter?

Jawab:

Menggunakan parameter dengan passing parameter dilakukan ketika sebuah metode atau konstruktor membutuhkan informasi tambahan dari luar untuk menjalankan tugasnya.

8. Apa perbedaan atribut class dan instansiasi atribut?

Jawab:

Instansiasi atribut dilakukan pada konstruktor, sedangkan atribut class didefinisikan diluar method apapun namun masih berada di dalam class.

9. Apa perbedaan class method dan instansiasi method?

Jawab:

Class method dapat mengakses dan memodifikasi atribut instance, sedangkan Tidak dapat mengakses atribut instance, tetapi bisa bekerja dengan atribut kelas (class attributes).

Tugas

1. Cobalah program dibawah ini dan tuliskan hasil outputnya

```
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 pada method setAge() terdapat statement if dengan syarat apabila newAge (set age) > 30 maka nilai yang tersimpan pada variabel age adalah 30. Itulah mengapa output yang dihasilkan pada age adalah 30.

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

Jawab:

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

Output jika setAge > 30:

```
encap.setNama(newName: "James");  
encap.setAge(newAge: 40);  
System.out.println("Name: " + encap.getNama());  
System.out.println("Age: " + encap.getAge());
```

gas.EncapTest
Name: James
Age: 30

Output jika setAge < 18

```
encap.setNama(newName: "James");  
encap.setAge(newAge: 7);  
System.out.println("Name: " + encap.getNama());  
System.out.println("Age: " + encap.getAge());
```

gas.EncapTest
Name: James
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 menangsur 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.

Jawab:

Code:

```
public class Anggota {
    private String nama;
    // private String nik;
    private String nik;
    private int limit;
    private float pinjaman;
    private float simpanan;
    Anggota(String nik, String nama, int limit) {
        this.nama = nama;
        this.nik = nik;
        this.limit = limit;
    }
    public void setNama(String nama) {
        this.nama = nama;
    }
    public void setnik(String nik) {
        this.nik = nik;
    }
    public void setLimit(int limit) {
        this.limit = limit;
    }
    public String getNama() {
        return nama;
    }
    public String getNik() {
        return nik;
    }
    public int getLimit() {
        return limit;
    }
    public void pinjam(float uang) {
        if (uang > limit) {
            System.out.println("Maaf, jumlah pinjaman
melebihi limit.");
        } else {
            pinjaman += uang;
        }
    }
    public void angsur(float uang) {
        pinjaman -= uang;
    }
    public float getPinjaman() {
        return pinjaman;
    }
}
```

Output:

```
Nama Anggota: Donny
Limit Pinjaman: 5000000

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

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

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

Membayar angsuran Rp. 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”.

Jawab:

Code:

```
public void angsur(float uang) {
    float batasAngsur = 0.1f * pinjaman;
    if (uang < batasAngsur) {
        System.out.println("Maaf, angsuran harus 10%
dari jumlah pinjaman.");
    } else {
        pinjaman -= uang;
    }
}
```

Pada method angsur terdapat variabel batasAngsur yang memiliki nilai $0.1f * \text{jumlah pinjaman}$, apabila uang yang diangsur kurang dari 10% dari pinjaman maka tidak program tidak dapat mengurangi jumlah angsuran pinjaman.

Output:

```
Nama Anggota: Donny
Limit Pinjaman: 5000000

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

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

Membayar angsuran Rp. 1.000.000...
Maaf, angsuran harus 10% dari jumlah pinjaman.
Jumlah pinjaman saat ini: 4000000.0

Membayar angsuran Rp. 3.000.000...
Jumlah pinjaman saat ini: 1000000.0
```

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

Jawab:

Code:

```
public class KoperasiDemo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("NIK: ");
        String nik = sc.next();
        System.out.print("Nama: ");
        String nama = sc.next();
        System.out.print("Limit Pinjaman: ");
        int limit = sc.nextInt();
        sc.next();
        Anggota agt = new Anggota(nik, nama, limit);
        int pilih;
        do {
            System.out.println("1. Pinjam\n2. Angsur\n3.
Exit");
            pilih = sc.nextInt();
            switch (pilih) {
                case 1:
                    System.out.print("Nominal Pinjam: ");
                    int pjml = sc.nextInt();
                    agt.pinjam(pjml);
                    System.out.println("Jumlah pinjaman saat
ini: "+ agt.getPinjaman());
                    System.out.println();
                    break;
                case 2:
                    System.out.print("Nominal Angsur: ");
                    int angs = sc.nextInt();
                    agt.angsur(angsa);
                    System.out.println("Jumlah pinjaman saat
ini: "+ agt.getPinjaman());
                    System.out.println();
                    break;
                case 3:
                    System.exit(0);
                    break;
                default:
                    break;
            }
        } while (pilih != 3);
        sc.close();
    }
}
```

Output:

```
NIK: 101010101
Nama: Gunawan
Limit Pinjaman: 5000000

1. Pinjam
2. Angsur
3. Exit
Pilih: 1

Nominal Pinjam: 3000000
Jumlah pinjaman saat ini: 3000000.0

1. Pinjam
2. Angsur
3. Exit
Pilih: 1

Nominal Pinjam: 2000000
Jumlah pinjaman saat ini: 5000000.0

1. Pinjam
2. Angsur
3. Exit
Pilih: 1

Nominal Pinjam: 10000000
Maaf, jumlah pinjaman melebihi limit.
Jumlah pinjaman saat ini: 5000000.0

1. Pinjam
2. Angsur
3. Exit
Pilih: 2

Nominal Angsur: 1000
Maaf, angsuran harus 10% dari jumlah pinjaman.
Jumlah pinjaman saat ini: 5000000.0

1. Pinjam
2. Angsur
3. Exit
Pilih: 2

Nominal Angsur: 3000000
Jumlah pinjaman saat ini: 2000000.0

1. Pinjam
2. Angsur
3. Exit
Pilih: 2

Nominal Angsur: 2000000
Jumlah pinjaman saat ini: 0.0

1. Pinjam
2. Angsur
3. Exit
Pilih: 3
```