

**Tugas Praktikum Pertemuan 1- Pengantar Konsep
Pemrograman Berorientasi Objek**



**Muhammad Farrel Caesarian
TI-2D/16
2341720053
POLITEKNIK NEGERI MALANG**

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: Karena mesin dalam kondisi mati, sehingga tidak masuk akal jika kecepatan bisa bertambah.

2. Mengapa atribut kecepatan dan kontakOn diset private?

Jawab: Untuk membatasi akses atribut agar hanya bisa diakses di class itu saja.

3. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

Jawab:

```
public void tambahKecepatan(){
    if(kontakOn==true){
        if(kecepatan>=100){
            System.out.println(x:"Kecepatan sudah maksimal");
        }else{
            kecepatan+=5;
        }
    }else{
        System.out.println(x:"Mesin Mati");
    }
}
```

Mesin Mati
Kontak On
Kecepatan 90

Kontak On
Kecepatan 95

Kontak On
Kecepatan 100

Kecepatan sudah maksimal
Kontak On
Kecepatan 100

Kontak Off
Kecepatan 0

Hasil Percobaan 3&4

Simpanan Farrel : Rp 100000.0
Simpanan Farrel : Rp 50000.0

Simpanan Caesar : Rp 0.0
Simpanan Farrel : Rp 100000.0
Simpanan Farrel : Rp 50000.0

3.6 Pertanyaan – Percobaan 3 dan 4

1. Apa yang dimaksud getter dan setter?

Jawab: getter digunakan untuk menampilkan, sedangkan setter digunakan untuk menginisialisasikan.

2. Apa kegunaan dari method getSimpanan()?

Jawab: Menampilkan jumlah saldo.

3. Method apa yang digunakan untuk menambah saldo?

Jawab: Method void Setor().

4. Apa yang dimaksud konstruktor?

Jawab: Konstruktor adalah method khusus dalam suatu class yang dipanggil secara otomatis ketika sebuah objek dari class tersebut dibuat (diinstansiasi).

5. Sebutkan aturan dalam membuat konstruktor?

Jawab: Nama konstruktor harus sama dengan nama class, Konstruktor dapat memiliki parameter untuk menginisialisasi atribut dengan nilai tertentu.

6. Apakah boleh konstruktor bertipe private?

Jawab: Ya, boleh. Konstruktor bertipe private digunakan untuk membatasi pembuatan objek dari luar class. Ini biasanya dilakukan dalam pola desain seperti Singleton, di mana hanya satu instance objek yang boleh dibuat.

7. Kapan menggunakan parameter dengan passing parameter?

Jawab: Parameter dengan passing parameter digunakan ketika kita ingin mengirimkan data ke dalam method atau konstruktor, sehingga method atau konstruktor dapat bekerja berdasarkan nilai-nilai yang dikirimkan.

8. Apa perbedaan atribut class dan instansiasi atribut?

Jawab:

-Atribut class (atau variabel statis) adalah atribut yang dimiliki bersama oleh semua objek dari class tersebut. Atribut ini didefinisikan dengan kata kunci static, sehingga hanya ada satu salinan yang dimiliki seluruh objek.

-Atribut instansiasi (atau variabel instance) adalah atribut yang unik untuk setiap objek dari class tersebut. Setiap objek memiliki salinan tersendiri dari variabel instance.

9. Apa perbedaan class method dan instansiasi method?

Jawab:

-Class method (method statis) adalah method yang dapat dipanggil tanpa harus membuat instance dari class tersebut. Method ini dideklarasikan dengan kata kunci static dan biasanya digunakan untuk melakukan operasi yang tidak bergantung pada objek.

-Instansiasi method (method instance) adalah method yang hanya bisa dipanggil melalui objek yang telah dibuat dari class tersebut.

Tugas

1.

```

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

```

```

1  public class EncapTest{
2      Run | Debug
3      public static void main(String[] args) {
4          EncapDemo encap = new EncapDemo();
5          encap.setName(newName: "Caesar");
6          encap.setAge(newAge: 35);
7
8          System.out.println("Name : " + encap.getName());
9          System.out.println("Age : " + encap.getAge());
10     }

```

Name : Caesar
 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 class EncapDemo method setAge() kita telah menuliskan jika age lebih dari 30, maka age akan diubah menjadi 30. Oleh karena itu, jika kita mengeset age dengan nilai 35, maka yang akan ditampilkan 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;
    }
}

encap.setName(newName:"Caesar");
encap.setAge(newAge:15);
Name : Caesar
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.

Jawab:



```
1  public class Koperasi{
2      private String nama,NIK;
3      private float limit_peminjaman, jumlah,saldo;
4
5      Koperasi(String nama, String NIK, float limit_peminjaman){
6          this.nama= nama;
7          this.NIK = NIK;
8          this.limit_peminjaman = limit_peminjaman;
9          this.saldo = saldo;
10     }
11     public void setNama(String nama){
12         this.nama = nama;
13     }
14     public void setNIK(String NIK){
15         this.NIK = NIK;
16     }
17     public void setLimit(float limit_peminjaman){
18         this.limit_peminjaman = limit_peminjaman;
19     }
20     public String getNama(){
21         return nama;
22     }
23     public String getNIK(){
24         return NIK;
25     }
26     public float getPinjam(){
27         return saldo;
28     }
29     public float getLimit(){
30         return limit_peminjaman;
31     }
32     public void angsur(float uang){
33         saldo -=uang;
34     }
35     public void pinjam(float uang){
36         saldo +=uang;
37     }
38 }
```

```

1  public class TestKoperasi{
2      public static void main(String[] args) {
3          Koperasi farrel= new Koperasi("Farrel", "3573", 3000000);
4
5          System.out.println("Nama Anggota      : " +farrel.getNama() );
6          System.out.println("Limit Peminjaman : " +farrel.getLimit() );
7
8          System.out.println("\nMeminjam uang 1000000");
9          farrel.pinjam(1000000);
10         System.out.println("Jumlah Pinjaman saat ini : " +farrel.getPinjam());
11
12         System.out.println("\nMeminjam uang 2000000");
13         farrel.pinjam(2000000);
14         System.out.println("Jumlah Pinjaman saat ini : " +farrel.getPinjam());
15
16         System.out.println("\nMembayar Angsuran 1000000");
17         farrel.angsur(1000000);
18         System.out.println("Jumlah Pinjaman saat ini : " +farrel.getPinjam());
19
20         System.out.println("\nMembayar Angsuran 2000000");
21         farrel.angsur(2000000);
22         System.out.println("Jumlah Pinjaman saat ini : " +farrel.getPinjam());
23     }
24 }

```

```

Nama Anggota      : Farrel
Limit Peminjaman : 3000000.0

Meminjam uang 1000000
Jumlah Pinjaman saat ini : 1000000.0

Meminjam uang 2000000
Jumlah Pinjaman saat ini : 3000000.0

Membayar Angsuran 1000000
Jumlah Pinjaman saat ini : 2000000.0

Membayar Angsuran 2000000
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:

```

public void angsur(float uang){
    if(saldo/10 >uang){
        System.out.println(x:"Maaf,angsuran harus 10% dari jumlah pinjaman");
    }else{
        saldo -=uang;
    }
}

```



```

Nama Anggota      : Farrel
Limit Peminjaman : 3000000.0

Meminjam uang 1000000
Jumlah Pinjaman saat ini : 1000000.0

Meminjam uang 2000000
Jumlah Pinjaman saat ini : 3000000.0

Membayar Angsuran 1000
Maaf,angsuran harus 10% dari jumlah pinjaman
Jumlah Pinjaman saat ini : 3000000.0

Membayar Angsuran 2000000
Jumlah Pinjaman saat ini : 1000000.0

```

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

Jawab:

```

1  import java.util.Scanner;
2  public class TestKoperasi{
3      public static void main(String[] args) {
4          Scanner input = new Scanner(System.in);
5          Koperasi farrel= new Koperasi("Farrel", "3573", 3000000);
6
7          System.out.println("Nama Anggota      : " +farrel.getNama() );
8          System.out.println("Limit Peminjaman : " +farrel.getLimit() );
9
10         System.out.print("\nMasukkan jumlah uang yang akan dipinjam: ");
11         float jumlahPinjam = input.nextFloat();
12         farrel.pinjam(jumlahPinjam);
13         System.out.println("Jumlah Pinjaman saat ini : " + farrel.getPinjam());
14
15         System.out.print("\nMasukkan jumlah uang untuk angsuran: ");
16         float jumlahAngsur = input.nextFloat();
17         farrel.angsur(jumlahAngsur);
18         System.out.println("Jumlah Pinjaman saat ini : " + farrel.getPinjam());
19
20         System.out.print("\nMasukkan jumlah uang yang akan dipinjam lagi: ");
21         jumlahPinjam = input.nextFloat();
22         farrel.pinjam(jumlahPinjam);
23         System.out.println("Jumlah Pinjaman saat ini : " + farrel.getPinjam());
24
25         System.out.print("\nMasukkan jumlah uang untuk angsuran lagi: ");
26         jumlahAngsur = input.nextFloat();
27         farrel.angsur(jumlahAngsur);
28         System.out.println("Jumlah Pinjaman saat ini : " + farrel.getPinjam());
29     }
30 }

```


Nama Anggota : Farrel
Limit Peminjaman : 3000000.0

Masukkan jumlah uang yang akan dipinjam: 2000000
Jumlah Pinjaman saat ini : 2000000.0

Masukkan jumlah uang untuk angsuran: 100000
Maaf, angsuran harus 10% dari jumlah pinjaman
Jumlah Pinjaman saat ini : 2000000.0

Masukkan jumlah uang yang akan dipinjam lagi: 1000
Jumlah Pinjaman saat ini : 2001000.0

Masukkan jumlah uang untuk angsuran lagi: 300000
Jumlah Pinjaman saat ini : 1701000.0