

OBJECT ORIENTED PROGRAMMING



By :

Sri Kresna Maha Dewa

2241720244

STUDY PROGRAM D-IV INFORMATIC ENGINEERING

INFORMATION TECHNOLOGY DEPARTMENT

MALANG STATE POLYTECHNIC

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

65141

PERTANYAAN PERCOBAAN 2

1. Pada TestMobil, saat kita menambah kecepatan untuk pertama kalinya, mengapa muncul peringatan “Kecepatan tidak bisa bertambah karena Mesin Off”?
Karena pada method TambahKecepatan terdapat selection yang memeriksa apakah atribut KontakOn sudah true atau tidak, karena attribute KontakOn masih false maka method akan memunculkan peringatan tersebut.
2. Mengapa attribute kecepatan dan kontakOn diset private?
Karena seperti yang dijelaskan sebelumnya, kita menerapkan information-hiding. Kita tidak perlu mengetahui kompleksitas yang ada di dalam sebuah objek.
3. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

```
public void tambahKecepatan() {  
    if (kontakOn == true) {  
        kecepatan += 5;  
    } else if (kecepatan >= 100) {  
        System.out.println(x:"Kecepatan tidak bisa bertambah karena melebihi batas maksimal! \n");  
    } else {  
        System.out.println(x:"Kecepatan tidak bisa bertambah karena Mesin Off! \n");  
    }  
}
```

PERTANYAAN PERCOBAAN 3 DAN 4

1. Apa yang dimaksud getter dan setter?
Getter adalah sebuah metode untuk mendapatkan nilai dari attribute private sebuah objek. Setter adalah sebuah metode untuk mengubah nilai dari attribute private sebuah objek
2. Apa kegunaan dari method getSimpanan()?
Mengembalikan nilai Simpanan yang telah di enkapsulasi
3. Method apa yang digunakan untuk menambah saldo?
Method setor
4. Apa yang dimaksud konstruktor?
Konstruktor adalah method yang dijalankan Ketika objek pertama kali dibuat
5. Sebutkan aturan dalam membuat konstruktor?
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?
Bisa, tetapi konstruktor hanya akan bisa diakses oleh class itu sendiri
7. Kapan menggunakan parameter dengan passing parameter?
Passing parameter dapat dipakai Ketika kita ingin menginisialisasi sesuatu terlebih dahulu sebelum menggunakannya.
8. Apa perbedaan atribut class dan instansiasi atribut?
Atribut dalam class terikat dengan class tersebut dan dapat digunakan oleh banyak objek yang diinstansiasi dengan class tersebut, juga attribute class dapat digunakan oleh class lain. Sedangkan instansiasi attribute biasanya hanya dimiliki oleh sebuah objek.
9. Apa perbedaan class method dan instansiasi method?

Class method dapat dimanfaatkan oleh class lain dan objek yang dinstansiasi oleh class nya, sedangkan instansiasi method adalah hal yang dapat dilakukan oleh sebuah objek.

TUGAS

1. Cobalah program dibawah ini dan tuliskan hasil outpunya

```
Name: James  
Age: 30  
PS D:\remote\code-for-edu>
```

2. Karena pada method setAge() Ketika nilainya lebih dari 30 akan diset menjadi 30.

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

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

```
27  public void setAge(int newAge) {  
28      if (newAge >= 18 && newAge <= 30) {  
29          age = newAge;  
30      } else {  
31          System.out.println(x:"Age must be between 18 and 30");  
32      }  
33  }  
34  }
```

4. CLASS ANGGOTA :

```
1 package oop_java.week3;
2
3 public class Anggota {
4     private String nomorkTP;
5     private String nama;
6     private int limitPinjaman;
7     private int jumlahPinjaman;
8
9     Anggota(String nomorkTP, String nama, int limitPinjaman) {
10         this.nomorkTP = nomorkTP;
11         this.nama = nama;
12         this.limitPinjaman = limitPinjaman;
13     }
14
15     public String getNama() {
16         return nama;
17     }
18
19     public int getLimitPinjaman() {
20         return limitPinjaman;
21     }
22
23     public int getJumlahPinjaman() {
24         return jumlahPinjaman;
25     }
26
27     public void pinjam(int pinjam) {
28         if (pinjam > limitPinjaman) {
29             System.out.println("Maaf, jumlah pinjaman melebihi limit.");
30         } else {
31             jumlahPinjaman += pinjam;
32         }
33     }
34
35     // public void angsur(int angsur) {
36     //     jumlahPinjaman -= angsur;
37     // }
38
39     public void angsur(int angsur) {
40         if (angsur < (jumlahPinjaman * 0.1)) {
41             System.out.println("Maaf, angsuran harus 10% dari jumlah pinjaman");
42         } else {
43             jumlahPinjaman -= angsur;
44         }
45     }
46
47
48 }
49
```

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 void angsur(int angsur) {  
    if (angsur < (jumlahPinjaman * 0.1)) {  
        System.out.println(x:"Maaf, angsuran harus 10% dari jumlah pinjaman");  
    } else {  
        jumlahPinjaman -= angsur;  
    }  
}
```

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

```
1  package oop_java.week3;  
2  
3  import java.util.Scanner;  
4  
5  public class TestKoperasi {  
6  
7      public static void main(String[] args) {  
8          Scanner sc = new Scanner(System.in);  
9          Anggota donny = new Anggota("111333444", "Donny", 5000000);  
10  
11         boolean menu = true;  
12  
13         while (menu) {  
14             System.out.println("Menu:");  
15             System.out.println("1. Pinjam");  
16             System.out.println("2. Angsur");  
17             System.out.println("3. Keluar");  
18             System.out.print("Pilih menu: ");  
19             int pilih = sc.nextInt();  
20  
21             switch (pilih) {  
22                 case 1:  
23                     System.out.print("Masukkan jumlah pinjaman: ");  
24                     int pinjam = sc.nextInt();  
25                     donny.pinjam(pinjam);  
26                     System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());  
27                     break;  
28                 case 2:  
29                     System.out.print("Masukkan jumlah angsuran: ");  
30                     int angsur = sc.nextInt();  
31                     donny.angsur(angsur);  
32                     System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());  
33                     break;  
34                 case 3:  
35                     menu = false;  
36                     break;  
37                 default:  
38                     System.out.println("Menu tidak tersedia");  
39                     break;  
40             }  
41         }  
42     }  
43 }  
44
```