

## **OOP Jobsheet2**



**Oleh :**

**NAME : Maulana Dwi Cahyono**

**CLASS : 1I**

**NO.ABSENT: 14**

**Major : Information Technology**

**STUDY PROGRAM : Information  
Engineering**

### 3.6 Pertanyaan – Percobaan 3 dan 4

1. Apa yang dimaksud getter dan setter?
2. Apa kegunaan dari method `getSimpanan()`?
3. Method apa yang digunakan untuk menambah saldo?
4. Apa yang dimaksud konstruktor?
5. Sebutkan aturan dalam membuat konstruktor?
6. Apakah boleh konstruktor bertipe `private`?
7. Kapan menggunakan parameter dengan `passsing` parameter?
8. Apa perbedaan atribut `class` dan instansiasi atribut?
9. Apa perbedaan `class method` dan instansiasi `method`?

1. Getter dan setter adalah metode dalam pemrograman yang digunakan untuk mengakses (getter) dan mengubah (setter) nilai dari atribut (variabel) suatu objek. Getter digunakan untuk mendapatkan nilai dari atribut, sedangkan setter digunakan untuk mengubah nilai atribut.

2. Method ``getSimpanan()`` digunakan untuk mendapatkan (mengambil) nilai saldo simpanan dari objek yang bersangkutan.

3. Method ``setor(float uang)`` digunakan untuk menambah saldo simpanan dengan jumlah tertentu. Dalam kode yang diberikan, ``setor`` digunakan untuk menambah saldo simpanan anggota.

4. Konstruktor adalah metode khusus dalam sebuah kelas yang digunakan untuk menginisialisasi objek saat objek tersebut dibuat. Konstruktor memiliki nama yang sama dengan nama kelasnya.

5. Aturan dalam membuat konstruktor adalah:

- Nama konstruktor harus sama dengan nama kelas.
- Konstruktor dapat memiliki parameter atau tidak.
- Konstruktor dapat memiliki aksi atau kode yang dijalankan saat objek dibuat.

**6. Ya, konstruktor boleh memiliki akses modifier private. Ketika sebuah konstruktor memiliki akses modifier private, objek dari kelas tersebut hanya dapat dibuat di dalam kelas itu sendiri, dan ini digunakan dalam pola desain Singleton.**

**7. Parameter dengan passing parameter digunakan ketika Anda ingin mengirim nilai ke sebuah method atau konstruktor. Ini memungkinkan Anda untuk menginisialisasi objek dengan nilai-nilai yang diberikan saat objek dibuat atau saat pemanggilan method.**

**8. Atribut class adalah variabel yang bersifat statis dan terhubung dengan kelas itu sendiri, sedangkan instansiasi atribut adalah variabel yang terhubung dengan objek yang dibuat dari kelas tersebut. Atribut class bersifat bersama untuk semua objek dari kelas yang sama, sementara instansiasi atribut memiliki salinan yang berbeda untuk setiap objek.**

**9. Class method (atau disebut juga dengan static method) adalah method yang terkait dengan kelas itu sendiri, bukan dengan objek individu dari kelas tersebut. Mereka dapat diakses melalui nama kelas tanpa harus membuat objek. Instansiasi method, di sisi lain, adalah method yang berkaitan dengan objek individu dan dapat diakses melalui objek tersebut.**

## 5. Tugas

1. Cobalah program dibawah ini dan tuliskan hasil outputnya

```
Name : James  
Age : 30  
PS C:\Users\ASUS\Desktop\kuliah python\Kuliah Java_Sems 3>
```

2. Pada program diatas, pada class EncapTest kita mengeset age dengan nilai 35, namun pada saat ditampilkan ke layar nilainya 30, jelaskan mengapa.

Jawaban:

Pada program diatas, pada class EncapTest kita mengeset age dengan nilai 35, namun pada saat ditampilkan ke layar nilainya 30, jelaskan mengapa.

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

Jawaban:

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

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.

```

package Jobsheet_2.no4_5_6;

public class Anggota {
    private String nomorKTP;
    private String nama;
    private double limitPinjaman;
    private double jumlahPinjaman;

    public Anggota(String nomorKTP, String nama, double limitPinjaman) {
        this.nomorKTP = nomorKTP;
        this.nama = nama;
        this.limitPinjaman = limitPinjaman;
        this.jumlahPinjaman = 0;
    }

    public String getNomorKTP() {
        return nomorKTP;
    }

    public String getNama() {
        return nama;
    }

    public double getLimitPinjaman() {
        return limitPinjaman;
    }

    public double getJumlahPinjaman() {
        return jumlahPinjaman;
    }

    public boolean pinjam(double jumlah) {
        if (jumlah > 0 && jumlah ≤ (limitPinjaman - jumlahPinjaman)) {
            jumlahPinjaman += jumlah;
            System.out.println("Jumlah pinjaman saat ini: " + jumlahPinjaman);
            return true;
        } else {
            System.out.println("Maaf, Jumlah melebihi limit ");
            return false;
        }
    }

    public void angsur(double jumlah) {
        if (jumlah > 0 && jumlah ≤ jumlahPinjaman) {
            jumlahPinjaman -= jumlah;
            System.out.println("Jumlah pinjaman saat ini: " + jumlahPinjaman);
        } else {
            System.out.println("Maaf, Jumlah melebihi pinjaman atau jumlah tidak valid.");
        }
    }
}

```

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".

Jawaban:

```
public void angsur(double jumlah) {  
    double minimalAngsur = 0.1 * jumlahPinjaman;  
    if (jumlah > 0 && jumlah ≤ jumlahPinjaman) {  
        if (jumlah ≥ minimalAngsur) {  
            jumlahPinjaman -= jumlah;  
            System.out.println("Angsuran berhasil. Jumlah pinjaman saat ini: " + jumlahPinjaman);  
        } else {  
            System.out.println(x:"Maaf, angsuran harus 10% dari jumlah pinjaman.");  
        }  
    } else {  
        System.out.println(x:"Angsuran gagal. Jumlah melebihi pinjaman atau jumlah tidak valid.");  
    }  
}
```

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

Jawaban:

```
import java.util.Scanner;  
  
public class TestKoperasi {  
    Run | Debug  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        Anggota donny = new Anggota(nomorKTP:"111333444", nama:"Donny", limitPinjaman:5000000);  
  
        System.out.println("Nama Anggota: " + donny.getNama());  
        System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());  
  
        System.out.print(s:"\nMeminjam uang. Masukkan jumlah pinjaman: ");  
        double pinjaman = scanner.nextDouble();  
        donny.pinjam(pinjaman);  
        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());  
  
        System.out.print(s:"\nMembayar angsuran. Masukkan jumlah angsuran: ");  
        double angsuran = scanner.nextDouble();  
        donny.angsur(angsuran);  
        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());  
    }  
}
```