

Jobsheet 3 Enkapsulasi PBO1

🍏 Type	Assignment
📅 Semester	Semester 3
🍀 Subject	Object Oriented Programming
📅 Time	@September 17, 2023

Pertanyaan percobaan 2

1. Pada class TestMobil, saat kita menambah kecepatan untuk pertama kalinya, mengapa muncul peringatan "Kecepatan tidak bisa bertambah karena Mesin Off!"?
 - Karena pada method tambahKecepatan terdapat sebuah percabangan untuk menambahkan kecepatan dengan syarat 'kontakOn' harus 'True'. dikarenakan pada awalnya keadaan 'kontakOn' itu false. maka akan diarahkan ke pilihan else dimana disitu akan menampilkan kalimat 'Kecepatan tidak bisa bertambah karena Mesin Off'
2. Mengapa atribut kecepatan dan kontakOn diset private?
 - Karena kita tidak perlu mengetahui kompleksitas yang ada di dalam sebuah object
3. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

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

Pertanyaan percobaan 3

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

- a. 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 outputnya

```
Name : James
Age : 30

Process finished with exit code 0
```

2. ada program diatas, pada class EncapTest kita mengeset age dengan nilai 35, namun pada saat ditampilkan ke layar nilainya 30, jelaskan mengapa.
- a. Karena method setAge pada class EncapDemo terdapat sebuah percabangan dimana jika 'age' diset lebih dari 30, maka secara otomatis 'age' akan diset dengan angka '30';
3. Ubah program diatas agar atribut age dapat diberi nilai maksimal 30 dan minimal 18.

```
public void setAge(int newAge){
    if (newAge > 30){
        System.out.println("Age can't be greater than 30");
    } else if (newAge < 18) {
        System.out.println("Age can't be less than 18");
    } else {
        age = newAge;
    }
}
```

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 No4;

2 usages
public class Anggota {
    1 usage
    private String nomorKTP;
    2 usages
    private String nama;
    3 usages
    private int limitPinjaman;
    4 usages
    private int jumlahPinjaman;

    1 usage
    Anggota(String nomorKTP, String nama, int limitPinjaman){
        this.nomorKTP = nomorKTP;
        this.nama = nama;
        this.limitPinjaman = limitPinjaman;
    }

    1 usage
    public String getNama(){
        return nama;
    }

    1 usage
    public int getLimitPinjaman(){
        return limitPinjaman;
    }
}
```

```
4 usages
public int getJumlahPinjaman(){
    return jumlahPinjaman;
}

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

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

```

Nama Anggota: Donny
Limit Pinjaman: 5000000

Meminjam uang 10.000.000...
Maaf, jumlah pinjamam melebihi limit
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

Process finished with exit code 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".

```

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

```

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

```

package No4;

import java.util.Scanner;

no usages
public class TestKoperasi {
    no usages
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Anggota donny = new Anggota( nomorKTP: "111333444", nama: "Donny", limitPinjaman: 5000000);

        boolean menu = true;

        while (menu){
            System.out.println("Menu");
            System.out.println("1. Pinjam");
            System.out.println("2. Angsur");
            System.out.println("3. Keluar");
            System.out.print("Masukkan pilihan Anda : ");
            int choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.println("Masukkan jumlah pinjaman: ");
                    int pinjam = sc.nextInt();
                    donny.pinjam(pinjam);
                    System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
                    break;

```

```

                    case 2:
                        System.out.println("Masukkan jumlah angsuran: ");
                        int angsur = sc.nextInt();
                        donny.angsur(angsur);
                        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
                        break;
                    case 3:
                        menu = false;
                        break;
                    default:
                        System.out.println("Masukkan pilihan yang benar!");
                        break;
                }
            }
        }
    }
}

```