



NAMA : Alfinza Sanjaya Putra
NIM : 2041720186
KELAS : 2C
MATKUL : Praktikum PBO

Pertanyaan Percobaan 1-2

1. Pada class TestMobil, saat kita menambah kecepatan untuk pertama kalinya, mengapa muncul peringatan “Kecepatan tidak bisa bertambah karena Mesin Off!”? [1] [SEP]

Jawab:

karena pada method tambahKecepatan ada kondisi if dimana kontakOn == true maka kecepatan akan ditambahkan 5, sedangkan kita belum memberikan perintah nyalakan mesin yang berarti kontakOn == false, maka yang akan dieksekusi yaitu else yang akan menampilkan “Kecepatan tidak bisa bertambah karena Mesin Off!”.

2. Mengapa atribut kecepatan dan kontakOn diset private? [1] [SEP]

Jawab:

agar atribut tersebut hanya bisa diakses di dalam class Motor dan jika ingin merubah value dari atribut tersebut tinggal memanggil method yang sudah dibuat pada class main.

3. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

Jawab:

- Kode program

```
16 public void tambahKecepatan(){
17     if (kontakOn == true) {
18         kecepatan += 50;
19         if (kecepatan > 100) {
20             System.out.println("Kecepatan melebihi batas!!");
21             kecepatan -= kecepatan;
22         }
23     }
```

- Output

```
run:
Kontak Off
Kecepatan 0

Kecepatan tidak bisa bertambah karena Mesin Off!

Kontak On
Kecepatan 0

Kontak On
Kecepatan 50

Kontak On
Kecepatan 100

Kecepatan melebihi batas!!
Kontak On
Kecepatan 0

Kontak Off
Kecepatan 0

BUILD SUCCESSFUL (total time: 0 seconds)
```



NAMA : Alfinza Sanjaya Putra
NIM : 2041720186
KELAS : 2C
MATKUL : Praktikum PBO

Pertanyaan Percobaan 3-4

1. Apa yang dimaksud getter dan setter?

Jawab:

adalah public method dimana getter memiliki tipe data return untuk mendapatkan nilai dari atribut private, sedangkan setter tidak memiliki tipe data return karena untuk memanipulasi nilai dari atribut private.

2. Apa kegunaan dari method getSimpanan()?

Jawab:

untuk mendapatkan nilai dari atribut simpanan.

3. Method apa yang digunakan untuk menambah saldo?

Jawab:

method setor, karena terdapat parameter uang yang digunakan untuk menambah nilai dari atribut simpanan.

4. Apa yang dimaksud konstruktor?

Jawab:

suatu method yang tidak memiliki return value dan digunakan untuk memberikan nilai pada suatu objek dengan cara melakukan instansiasi.

5. Sebutkan aturan dalam membuat konstruktor?

Jawab:

- **nama konstruktor harus sama dengan nama class.**
- **tidak memiliki tipe data return.**
- **tidak boleh menggunakan modifier abstract, static, final, dan synchronized.**

6. Apakah boleh konstruktor bertipe private?

Jawab:

boleh, tetapi tidak bisa diinstansiasi di dalam class lain.

7. Kapan menggunakan parameter dengan passing parameter?

Jawab:

passing parameter digunakan untuk atribut yang membutuhkan nilai yang spesifik, jika tidak membutuhkan nilai spesifik dalam konstruktor maka tidak perlu parameter.

8. Apa perbedaan atribut class dan instansiasi atribut?

Jawab:

Atribut class adalah atribut yang berada dalam class-nya sendiri sedangkan instansiasi atribut adalah atribut yang diakses dalam class yang lain.

9. Apa perbedaan class method dan instansiasi method?

Jawab:

class method adalah method yang berada dalam class-nya sendiri sedangkan instansiasi method adalah method yang diakses dalam class yang lain.



NAMA : Alfinza Sanjaya Putra
NIM : 2041720186
KELAS : 2C
MATKUL : Praktikum PBO

Tugas

1. Cobalah program dibawah ini dan tuliskan hasil outputnya

```
public class EncapDemo
{
    private String name;
    private int age;

    public String getName()
    {
        return name;
    }

    public void setName(String newName)
    {
        name = newName;
    }

    public int getAge()
    {
        return age;
    }

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

```
public class EncapTest
{
    public static void main(String args[])
    {
        EncapDemo encap = new EncapDemo();
        encap.setName("James");
        encap.setAge(35);

        System.out.println("Name : " + encap.getName());
        System.out.println("Age : " + encap.getAge());
    }
}
```

Jawab:

output seperti di bawah ini

```
run:
Name : James
Age : 30
BUILD SUCCESSFUL (total time: 0 seconds)
```

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 kondisi jika parameter newAge > 30 maka nilai atribut age menjadi 30

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

Jawab:

```
package tugas3;

public class EncapTest {
    public static void main(String[] args) {
        EncapDemo encap = new EncapDemo();
        EncapDemo encap2 = new EncapDemo();

        encap.setName("James");
        encap.setAge(35);
        encap2.setName("Agus");
        encap2.setAge(10);

        System.out.println("Name : " + encap.getName());
        System.out.println("Age : " + encap.getAge());
        System.out.println(" ");
        System.out.println("Name : " + encap2.getName());
        System.out.println("Age : " + encap2.getAge());
    }
}

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



NAMA : Alfinza Sanjaya Putra
NIM : 2041720186
KELAS : 2C
MATKUL : Praktikum PBO

```
run:
Name : James
Age : 30

Name : Agus
Age : 18
BUILD SUCCESSFUL (total time: 0 seconds)
```

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.

```
public class TestKoperasi
{
    public static void main(String[] args)
    {
        Anggota donny = new Anggota("111333444", "Donny", 5000000);

        System.out.println("Nama Anggota: " + donny.getNama()); System.out.println("Limit
        Pinjaman: " + donny.getLimitPinjaman());

        System.out.println("\nMeminjam uang 10.000.000...");
        donny.pinjam(10000000);
        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());

        System.out.println("\nMeminjam uang 4.000.000...");
        donny.pinjam(4000000);
        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());

        System.out.println("\nMembayar angsuran 1.000.000");
        donny.angsur(1000000);
        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());

        System.out.println("\nMembayar angsuran 3.000.000");
        donny.angsur(3000000);
        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
    }
}
```

Hasil yang diharapkan:

```
D:\MyJava>javac TestKoperasi.java
D:\MyJava>java TestKoperasi
Nama Anggota: Donny
Limit Pinjaman: 5000000

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

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
```



NAMA : Alfinza Sanjaya Putra
NIM : 2041720186
KELAS : 2C
MATKUL : Praktikum PBO

Jawab:

Class Anggota

```
1 package tugas4;  
2  
3 public class Anggota {  
4  
5     private String nama;  
6     private String noKtp;  
7     private int limitPinjaman;  
8     private int jumlahPinjaman;  
9  
10    Anggota(String nomorKTP, String Nama, int limitPinjaman) {  
11        this.noKtp = nomorKTP;  
12        this.nama = Nama;  
13        this.limitPinjaman = limitPinjaman;  
14    }  
15  
16    public String getNama() {  
17        return nama;  
18    }  
19  
20    public String getNomorKTP() {  
21        return noKtp;  
22    }  
23  
24    public int getLimitPinjaman() {  
25        return limitPinjaman;  
26    }  
27  
28    public int pinjam(int jumlahPinjam) {  
29        jumlahPinjaman += jumlahPinjam;  
30        if (jumlahPinjaman > getLimitPinjaman()) {  
31            System.out.println("Maaf, jumlah pinjaman melebihi limit!");  
32            jumlahPinjaman -= jumlahPinjam;  
33        }  
34        return jumlahPinjaman;  
35    }  
36  
37    public int angsur(int jumlahAngsur) {  
38        jumlahPinjaman -= jumlahAngsur;  
39        return jumlahPinjaman;  
40    }  
41  
42    public int getJumlahPinjaman() {  
43        return jumlahPinjaman;  
44    }  
45  
46 }
```

Output program

```
run:  
Nama Anggota: Donny  
Limit Pinjaman: 5000000  
  
Meminjam uang 10.000.000...  
Maaf, jumlah pinjaman 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  
BUILD SUCCESSFUL (total time: 0 seconds)
```

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:

Class Anggota

```
1 public int angsur(int jumlahAngsur) {  
2     if (jumlahAngsur < (0.1 * jumlahPinjaman)) {  
3         System.out.println("Maaf, angsuran harus 10% dari jumlah pinjaman");  
4         jumlahPinjaman += jumlahAngsur;  
5     }  
6     jumlahPinjaman -= jumlahAngsur;  
7     return jumlahPinjaman;  
8 }
```

Output program

```
run:  
Nama Anggota: Donny  
Limit Pinjaman: 5000000  
  
Meminjam uang 10.000.000...  
Maaf, jumlah pinjaman melebihi limit!  
Jumlah pinjaman saat ini: 0  
  
Meminjam uang 4.000.000...  
Jumlah pinjaman saat ini: 4000000  
  
Membayar angsuran 1.000.000  
Maaf, angsuran harus 10% dari jumlah pinjaman  
Jumlah pinjaman saat ini: 4000000  
  
Membayar angsuran 3.000.000  
Jumlah pinjaman saat ini: 1000000  
BUILD SUCCESSFUL (total time: 0 seconds)
```



NAMA : Alfinza Sanjaya Putra
NIM : 2041720186
KELAS : 2C
MATKUL : Praktikum PBO

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

Class TestKoperasi

```
1 package tugas6;  
2  
3 import java.util.Scanner;  
4  
5 public class TestKoperasi {  
6  
7     public static void main(String[] args) {  
8         Scanner alfin = new Scanner(System.in);  
9         int jumlah = 0;  
10        Anggota donny = new Anggota("111333444", "Donny", 5000000);  
11        System.out.println("Nama Anggota: " + donny.getName());  
12        System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());  
13        System.out.print("\nMeminjam uang : ");  
14        donny.pinjam(alfin.nextInt());  
15        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());  
16        System.out.print("\nMembayar uang : ");  
17        donny.pinjam(alfin.nextInt());  
18        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());  
19        System.out.print("\nMembayar angsuran : ");  
20        donny.angsur(alfin.nextInt());  
21        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());  
22        System.out.print("\nMembayar angsuran : ");  
23        donny.angsur(alfin.nextInt());  
24        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());  
25    }  
26 }
```

Output program

```
run:  
Nama Anggota: Donny  
Limit Pinjaman: 5000000  
  
Meminjam uang : 7000000  
Maaf, jumlah pinjaman melebihi limit!  
Jumlah pinjaman saat ini: 0  
  
Meminjam uang : 3000000  
Jumlah pinjaman saat ini: 3000000  
  
Membayar angsuran : 2000000  
Jumlah pinjaman saat ini: 1000000  
  
Membayar angsuran : 500000  
Jumlah pinjaman saat ini: 500000  
BUILD SUCCESSFUL (total time: 28 seconds)
```