



NAMA : Annisa Aulia Nadhila
NIM : 2041720023
KELAS : TI-2C
MATERI : Enkapsulasi

PERCOBAAN

1. L

```
public class Motor {  
    public int kecepatan = 0;  
    public boolean kontakOn = false;  
  
    public void printStatus(){  
        if(kontakOn == true){  
            System.out.println("Kontak On");  
        }else{  
            System.out.println("Kontak Off");  
        }  
        System.out.println("Kecepatan " + kecepatan + "\n");  
    }  
}
```

```
public class MotorDemo {  
    public static void main(String[] args) {  
        Motor motor = new Motor();  
        motor.printStatus();  
        motor.kecepatan = 50;  
        motor.printStatus();  
    }  
}
```

```
run:  
Kontak Off  
Kecepatan 0  
  
Kontak Off  
Kecepatan 50  
  
BUILD SUCCESSFUL (total time: 0 seconds)
```

2.



NAMA : Annisa Aulia Nadhila
NIM : 2041720023
KELAS : TI-2C
MATERI : Enkapsulasi

```
public class Motor {  
    private int kecepatan = 0;  
    private boolean kontakOn = false;  
    public void nyalakanMesin() {  
        kontakOn = true;  
    }  
    public void matikanMesin() {  
        kontakOn = false;  
        kecepatan = 0;  
    }  
    public void tambahecepatan() {  
        if(kontakOn == true) {  
            kecepatan += 5;  
        } else {  
            System.out.println("Kecepatan tidak bisa bertambah karna Mesin Off! \n");  
        }  
    }  
    public void kurangiKecepatan() {  
        if(kontakOn == true) {  
            kecepatan -= 5;  
        } else {  
            System.out.println("Kecepatan tidak bisa berkurang karna Mesin Off! \n");  
        }  
    }  
    public void printStatus() {  
        if(kontakOn == true) {  
            System.out.println("Kontak On");  
        } else {  
            System.out.println("Kontak Off");  
        }  
        System.out.println("Kecepatan " + kecepatan + "\n");  
    }  
}
```



NAMA : Annisa Aulia Nadhila
NIM : 2041720023
KELAS : TI-2C
MATERI : Enkapsulasi

```
public class MotorDemo {  
    public static void main(String[] args) {  
        Motor motor = new Motor();  
        motor.printStatus();  
        motor.tambahekecepatan();  
  
        motor.nyalakanMesin();  
        motor.printStatus();  
  
        motor.tambahekecepatan();  
        motor.printStatus();  
  
        motor.tambahekecepatan();  
        motor.printStatus();  
  
        motor.tambahekecepatan();  
        motor.printStatus();  
  
        motor.matikanMesin();  
        motor.printStatus();  
    }  
}
```

run:

Kontak Off
Kecepatan 0

Kecepatan tidak bisa bertambah karna Mesin Off!

Kontak On
Kecepatan 0

Kontak On
Kecepatan 5

Kontak On
Kecepatan 10

Kontak On
Kecepatan 15

Kontak Off
Kecepatan 0

BUILD SUCCESSFUL (total time: 0 seconds)

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

Jawab : karna meskipun mesin telah dinyalakan tapi kecepatan belum ditambahkan.

2. Mengapa atribut kecepatan dan kontakOn diset private?



NAMA : Annisa Aulia Nadhila
NIM : 2041720023
KELAS : TI-2C
MATERI : Enkapsulasi

Jawab : agar atribut tersebut tidak berubah-ubah dan hanya bisa diakses diclass tersebut

3. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

Jawab :

```
public void tambahecepatan(){
    if(kontakOn == true){
        kecepatan += 5;
        if (kecepatan > 100) {
            System.out.println("Kecepatan tidak bisa ditambahkan lagi karna sudah maksimal");
            kecepatan = 100;
        }
    }else{
        System.out.println("Kecepatan tidak bisa bertambah karna Mesin Off! \n");
    }
}
```

3.

```
public class Anggota {
    public String nama;
    public String alamat;
    private float simpanan;
    public void setName(String nama) {
        this.nama = nama;
    }

    public void setAddress(String alamat) {
        this.alamat = alamat;
    }

    public String getName() {
        return nama;
    }

    public String getAddress() {
        return alamat;
    }

    public float getSimpanan() {
        return simpanan;
    }

    public void setor(float uang){
        simpanan += uang;
    }
    public void pinjam(float uang){
        simpanan -= uang;
    }
}
```



NAMA : Annisa Aulia Nadhila
NIM : 2041720023
KELAS : TI-2C
MATERI : Enkapsulasi

```
public class KoprasiDemo {  
    public static void main(String[] args) {  
        Anggota anggotal = new Anggota();  
        anggotal.setNama("Iwan Setiawan");  
        anggotal.setAlamat("Jalan Sukarno Hatta no 10");  
        anggotal.setor(100000);  
        System.out.println("Simpanan "+anggotal.getNama()+" : Rp "+anggotal.getSimpanan());  
  
        anggotal.pinjam(5000);  
        System.out.println("Simpanan "+anggotal.getNama()+" : Rp "+anggotal.getSimpanan());  
    }  
}
```

```
run:  
Simpanan Iwan Setiawan : Rp 100000.0  
Simpanan Iwan Setiawan : Rp 95000.0  
BUILD SUCCESSFUL (total time: 1 second)
```

4.

```
public class KoprasiDemo {  
    public static void main(String[] args) {  
        Anggota anggotal = new Anggota();  
        System.out.println("Simpanan "+anggotal.getNama()+" : Rp "+anggotal.getSimpanan());  
  
        anggotal.setNama("Iwan Setiawan");  
        anggotal.setAlamat("Jalan Sukarno Hatta no 10");  
        anggotal.setor(100000);  
        System.out.println("Simpanan "+anggotal.getNama()+" : Rp "+anggotal.getSimpanan());  
  
        anggotal.pinjam(5000);  
        System.out.println("Simpanan "+anggotal.getNama()+" : Rp "+anggotal.getSimpanan());  
    }  
}
```

```
run:  
Simpanan null : Rp 0.0  
Simpanan Iwan Setiawan : Rp 100000.0  
Simpanan Iwan Setiawan : Rp 95000.0  
BUILD SUCCESSFUL (total time: 1 second)
```

Pertanyaan – Percobaan 3 dan 4

1. Apa yang dimaksud getter dan setter?

Jawab : getter setter adalah salah satu fungsi oop untuk memberi dan mengambil nilai dari atribut

2. Apa kegunaan dari method getSimpanan()?

Jawab : untuk mengambil nilai dari atribut simpan;

3. Method apa yang digunakan untuk menambah saldo?



NAMA : Annisa Aulia Nadhila
NIM : 2041720023
KELAS : TI-2C
MATERI : Enkaptulasi

Jawab : setor

4. Apa yang dimaksud konstruktor?

Jawab : konstruktor adalah method yang memiliki nama yang sama dengan nama class yang berfungsi untuk menghubungkan class satu dengan lainnya, meskipun konstruktor adalah method tapi konstruktor tidak punya return value;

5. Sebutkan aturan dalam membuat konstruktor?

Jawab : nama konstruktor sama dengan nama class, tidak memiliki return value. Konstruktor tidak boleh menggunakan modifier abstract, static, final, dan synchronized.

6. Apakah boleh konstruktor bertipe private?

Jawab : tidak boleh

7. Kapan menggunakan parameter dengan passing parameter?

Jawab : parameter digunakan saat ingin menggunakan suatu atribut tambahan dalam method. Sedangkan passing parameter digunakan ketika kita ingin menggunakan parameter tersebut di tempat lain.

8. Apa perbedaan atribut class dan instansiasi atribut?

Jawab : atribut adalah bagian yang ada dalam class, sedangkan instansiasi atribut adalah proses untuk membuat atribut

9. Apa perbedaan class method dan instansiasi method?

Jawab : class method adalah method yang ada dalam method, sedangkan instansiasi method adalah proses pembuatan / pemberian nilai pada method

TUGAS

1.

```
package pertemuan3;

/**
 *
 * @author HP
 */
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());
    }
}
```



NAMA : Annisa Aulia Nadhila
NIM : 2041720023
KELAS : TI-2C
MATERI : Enkaptulasi

```
package pertemuan3;

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

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

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

```
run:
Name = James
Age = 30
BUILD SUCCESSFUL (total time: 1 second)
```

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 dimethod setAge kita memberikan kondisi dimaan jika angka yang dimasukkan lebih dari 30 maka nilai umur adalah 30, sedangkan apabila nilai umur yang dimasukkan sama dengan atau kurng dari 30 maka nilai umur sama dengan umur yang dimasukkan

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

```
public void setAge(int newAge) {
    if ( 18 <= newAge && newAge <= 30) {
        age = newAge;
    } else {
        System.out.println("Umur minimam 18 dan maksimal 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



NAMA : Annisa Aulia Nadhila
NIM : 2041720023
KELAS : TI-2C
MATERI : Enkapsulasi

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 Anggota {  
    public String nomorKtp;  
    public String nama;  
    public int limitPinjam;  
    public int jumlahPinjam;  
    public int jumlahAngsur;  
    public Anggota(String nomorKtp, String nama, int limitPinjam) {  
        this.nomorKtp = nomorKtp;  
        this.nama = nama;  
        this.limitPinjam = limitPinjam;  
    }  
    public String getName() {  
        return nama;  
    }  
    public int getLimit() {  
        return limitPinjam;  
    }  
    public int getJumlahPinjam() {  
        return jumlahPinjam - jumlahAngsur;  
    }  
    public void pinjam(int Pinjam) {  
        if (limitPinjam > Pinjam) {  
            this.jumlahPinjam = Pinjam;  
        }else{  
            System.out.println("Maaf, jumlah pinjam melebihi limit.");  
        }  
    }  
    public void angsur(int jumlahAngsur) {  
        this.jumlahAngsur = jumlahAngsur;  
    }  
}
```




NAMA : Annisa Aulia Nadhila
NIM : 2041720023
KELAS : TI-2C
MATERI : Enkapsulasi

```
package pertemuan3;

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 Peminjam = " + donny.getLimit());

        System.out.println("\nMeminjam uang 10.000.000...");
        donny.pinjam(10000000);

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

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

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

    }
}
```

```
run:
Nama Anggota : Donny
Limit Peminjam = 5000000

Meminjam uang 10.000.000...
Maaf, jumlah pinjam 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
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"

```
public void angsur(int Angsur) {
    if (Angsur >= (jumlahPinjam * 0.1)) {
        this.jumlahAngsur = Angsur;
    } else {
        System.out.println("Maaf, angsuran harus 10% dari jumlah pinjam");
    }
}
```



NAMA : Annisa Aulia Nadhila
NIM : 2041720023
KELAS : TI-2C
MATERI : Enkaptulasi

```
package pertemuan3;

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 Peminjam = " + donny.getLimit());

        System.out.println("\nMeminjam uang 10.000.000...");
        donny.pinjam(10000000);

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

        System.out.println("\nMembayar angsuran 1000");
        donny.angsur(1000);
        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjam());

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

```
run:
Nama Anggota : Donny
Limit Peminjam = 5000000

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

Meminjam uang 4.000.000...
Jumlah pinjaman saat ini: 4000000

Membayar angsuran 1000
Maaf, angsuran harus 10% dari jumlah pinjam
Jumlah pinjaman saat ini: 4000000

Membayar angsuran 3.000.000
Jumlah pinjaman saat ini: 1000000
BUILD SUCCESSFUL (total time: 0 seconds)
```

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



NAMA : Annisa Aulia Nadhila
NIM : 2041720023
KELAS : TI-2C
MATERI : Enkaptulasi

```
import java.util.Scanner;
public class TestKoperasi {
    public static void main(String[] args) {
        Anggota donny = new Anggota("111333444", "Donny", 5000000);
        Scanner annisa = new Scanner(System.in);
        System.out.println("Nama Anggota : " + donny.getNama());
        System.out.println("Limit Peminjam = " + donny.getLimit());

        //      System.out.println("\nMeminjam uang 10.000.000...");
        //      donny.pinjam(10000000);

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

        //      System.out.println("\nMembayar angsuran 1000");
        //      donny.angsur(1000);
        //      System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjam());
        System.out.print("Masukkan Jumlah Pinjaman : ");
        donny.pinjam(annisa.nextInt());
        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjam());
        System.out.println("");
        System.out.print("Masukkan jumlah angsuran : ");
        donny.angsur(annisa.nextInt());
        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjam());

    }
}
```

```
run:
Nama Anggota : Donny
Limit Peminjam = 5000000
Masukkan Jumlah Pinjaman : 4000000
Jumlah pinjaman saat ini: 4000000

Masukkan jumlah angsuran : 3000000
Jumlah pinjaman saat ini: 1000000
BUILD SUCCESSFUL (total time: 11 seconds)
```