

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJECT
JOBSHEET 3**



Sahrul Ramadhani

244107020058

2H

D-IV Teknik Informatika

Teknologi Informasi

Politeknik Negeri Malang

2025

Percobaan 1

Hasil RUN:

```
\Pemrograman Berbasis Object_b3ef0170\bin' 'motor
Kontak Off
Kecepatan 0

Kontak Off
Kecepatan 50
```

Percobaan 2

Hasil RUN:

```
Kontak Off
Kecepatan 0

Kecepatan tidak bisa bertambah karena mesin Off
Kontak On.
Kecepatan 0

Kontak On.
Kecepatan 5

Kontak On.
Kecepatan 10

Kontak On.
Kecepatan 15

Kontak Off
Kecepatan 0

PS D:\Pemrograman Berbasis Object>
```

Pertanyaan Percobaan 2

1. Pada class TestMobil?? motorDemo ya?, saat kita menambah kecepatan untuk pertama kalinya, mengapa muncul peringatan “Kecepatan tidak bisa bertambah karena Mesin Off!”? ^[1]_[SEP]
2. Mengapad atribut kecepatan dan kontakOn diset private? ^[1]_[SEP]
3. Ubah class Motor sehingga kecepatan maksimalnya adalah 100!

Jawaban:

1. Karena di Class Motor ada If Else

```
public void tambahKecepatan () {
    if (kontakOn == true) {
        kecepatan += 5;
    }
    else {
        System.out.println("Kecepatan tidak bisa bertambah karena
mesin Off ");
    }
}
```

```
}
```

Artinya jika kontakOn == False maka akan ada peringatan Kecepatan tidak bisa bertambah karena mesin Off

2. Ya agar User ketika menambahkan kecepatan tidak bisa langsung instan, misalnya dari 0-100 jadi secara bertahap dengan method tambahKecepatan setiap memanggil method akan bertambah +=5
3. Hanya perlu ubah pada method tambahKecepatan sebagaiberikut:

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

Percobaan 3

Hasil RUN:

```
mo'  
Simpanan Iwan Setiawan : Rp. 100000.0  
Simpanan Iwan Setiawan : Rp. 95000.0  
PS D:\Pemrograman Berbasis Object>
```

Percobaan 4

Hasil RUN:

1.

```
Simpanan null : Rp. 0.0  
Simpanan Iwan Setiawan : Rp. 100000.0  
Simpanan Iwan Setiawan : Rp. 95000.0  
PS D:\Pemrograman Berbasis Object>
```

2. Ditambahkan kontruktur dengan access modifier default

```
Berbasis Object_b3ef0170\bin' 'koperasiDemo'  
Simpanan Iwan : Rp. 0.0  
Simpanan Iwan Setiawan : Rp. 100000.0  
Simpanan Iwan Setiawan : Rp. 95000.0  
PS D:\Pemrograman Berbasis Object>
```

Pertanyaan Percobaan 3 & Percobaan 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 passing parameter?
8. Apa perbedaan atribut class dan instansiasi atribut?
9. Apa perbedaan class method dan instansiasi method?

JAWABAN:

1. - Getter = method untuk mengambil/melihat isi data (misalnya saldo, nama).
- Setter = method untuk mengubah atau mengisi data.
2. Untuk melihat berapa jumlah saldo/tabungan yang dimiliki anggota saat ini.
3. Method setor() digunakan untuk menambah saldo/simpanan.
4. Konstruktor adalah method khusus yang otomatis dipanggil saat membuat objek baru. Fungsinya untuk mengisi data awal.
5. Nama konstruktor harus sama dengan nama class.
 - Tidak boleh punya tipe kembalian (void, int, dll).
 - Boleh punya parameter (opsional).
6. Boleh, tapi sangat jarang dipakai. Biasanya digunakan untuk membatasi pembuatan objek, misalnya di Singleton Pattern (supaya hanya ada 1 objek).
7. Saat kita ingin mengisi data awal langsung dari luar class ketika objek dibuat.
Contoh: `new Anggota("Budi", "Malang", 100000);`
8. - Atribut class (static) → data milik bersama semua objek (contoh: tarif bunga koperasi).
- Atribut instansiasi (non-static) → data milik masing-masing objek (contoh: saldo tiap anggota berbeda-beda).
9. - Class method (static) → bisa dipanggil langsung pakai nama class, tanpa bikin objek.
- Instansiasi method (non-static) → harus lewat objek yang sudah dibuat.

TUGAS

1. Hasil RUN kode :

```
Name : James  
Age : 30  
PS D:\Pemrograman Berbasis Object>
```

2. Pada program diatas, pada class EncapTest kita mengeset age dengan nilai 35, namun pada saat ditampilkan ke layar nilainya 30, jelaskan mengapa?
Karena encapDemo ada If Else yang fungsinya jika memasukan Age > 30 maka akan di simpan 30

```

if (newAge > 30){
    age = 30;
}

```

3. Kode saya ubah Jika <17 atau >30 maka akan ada peringatan minimum maximum dan otomatis keluar program

```

public void setAge (int newAge){
    if (age <18 || age>30) {
        System.out.println("Minimum age is 18 and maximum age is 30");
        System.exit(0);
    }
    else {
        age = newAge;
    }
}

```

Hasil RUN :

```

n Berbasis Object_b3ef0170\bin' 'encapTest'
Minimum age is 18 and maximum age is 30
PS D:\Pemrograman Berbasis Object>

```

4. Berikut adalah hasil RUN dari Tugas untuk membuat sebuah sistem peminjaman, yang bisa di lihat kodenya pada folder Koperasi

```

Nama : Rama
Nomor KTP : 1234567890
Limit Pinjaman : 5000000.0

Maaf, jumlah pinjaman melebihi limit
Anda ingin meminjam : 9000000.0
Sedangkan Limit pinjaman Anda : 5000000.0

Meminjam Uang 4000000.0
Jumlah pinjaman saat ini : 4000000.0

Melakukan pembayaran sejumlah 1000000.0
Jumlah pinjaman saat ini : 3000000.0

Melakukan pembayaran sejumlah 3000000.0
Jumlah pinjaman saat ini : 0.0
PS D:\Pemrograman Berbasis Object>

```

5. Hasil Modify Kode pada method membayar:

```

public void membayar (float uang){
    if (uang < 0.1*pinjaman) {
        System.out.println("jumlah nominal angsuran minimal 10% dari totalpinjaman");
        System.out.println("anda harus membayar minimal : " + 0.1*pinjaman);
    } else {
        pinjaman -= uang;
        System.out.println("Melakukan pembayaran sejumlah " + uang);
        System.out.println("Jumlah pinjaman saat ini : " + pinjaman);
    }
}

```

Hasil RUN :

```
Nama : Rama
Nomor KTP : 1234567890
Limit Pinjaman : 5000000.0

Maaf, jumlah pinjaman melebihi limit
Anda ingin meminjam : 9000000.0
Sedangkan Limit pinjaman Anda : 5000000.0

Meminjam Uang 4000000.0
Jumlah pinjaman saat ini : 4000000.0

jumlah nominal angsuran minimal 10% dari totalpinjaman
anda harus membayar minimal : 400000.0

Melakukan pembayaran sejumlah 3000000.0
Jumlah pinjaman saat ini : 1000000.0
PS D:\Pemrograman Berbasis Object>
```

6. Hasil Modify Code dengan menambahkan Scanner dan Syntax Inputnya nya :

```
import java.util.Scanner;
public class testKoperasi {
    public static void main(String[] args) {

        float pinjam;
        float bayar;
        Scanner input= new Scanner(System.in);
        koperasi Rama = new koperasi ("1234567890", "Rama", 5000000);
        System.out.println("");
        System.out.println("Nama : " + Rama.getNama());
        System.out.println("Nomor KTP : " + Rama.getNomorKTP());
        System.out.println("Limit Pinjaman : " +
Rama.getLimitPinjaman());
        System.out.println("");

        System.out.println("");
        System.out.print("masukkan jumlah pinjaman : ");
        pinjam = input.nextFloat();
        Rama.memorpinjam(pinjam);

        System.out.println("");
        System.out.print("masukkan jumlah pinjaman : ");
        pinjam = input.nextFloat();
        Rama.memorpinjam(pinjam);

        System.out.println("");
        System.out.print("masukkan jumlah angsuran : ");
        bayar = input.nextFloat();
        Rama.membayar(bayar);

        System.out.println("");
        System.out.print("masukkan jumlah angsuran : ");
        bayar = input.nextFloat();
        Rama.membayar(bayar);

    }
}
```

HASIL RUN :

```
Nama : Rama
Nomor KTP : 1234567890
Limit Pinjaman : 5000000.0

masukkan jumlah pinjaman : 6000000
Maaf, jumlah pinjaman melebihi limit
Anda ingin meminjam : 6000000.0
Sedangkan Limit pinjaman Anda : 5000000.0

masukkan jumlah pinjaman : 4000000
Meminjam Uang 4000000.0
Jumlah pinjaman saat ini : 4000000.0

masukkan jumlah angsuran : 300000
jumlah nominal angsuran minimal 10% dari totalpinjaman
anda harus membayar minimal : 400000.0

masukkan jumlah angsuran : 2500000
Melakukan pembayaran sejumlah 2500000.0
Jumlah pinjaman saat ini : 1500000.0
PS D:\Pemrograman Berbasis Object> |
```