

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJECT
JOBSHEET 1



Sahrul Ramadhani

244107020058

2H

D-IV Teknik Informatika

Teknologi Informasi

Politeknik Negeri Malang

2025

PERCOBAAN 1

Didalam percobaan ini, kita akan mendemonstrasikan bagaimana membuat class, membuat object, kemudian mengakses method didalam class tersebut.

1. Buka Netbeans, buat project SepedaDemo.
2. Buat class Sepeda. Klik kanan pada package sepedademo – New – Java Class.
3. Ketikkan kode class Sepeda dibawah ini.

Berikut adalah Hasil Source Code nya :

Sepeda.Java

```
public class Sepeda {
    private String merek;
    private int kecepatan;
    private int gear;

    public void setMerek (String newValue)
    {
        merek = newValue;
    }

    public void gantiGear (int newValue)
    {
        gear = newValue;
    }

    public void tambahKecepatan (int increment)
    {
        kecepatan = kecepatan + increment;
    }

    public void rem (int decrement)
    {
        kecepatan = kecepatan - decrement;
    }

    public void cetakStatus ()
    {
        System.out.println("Merek: " + merek);
        System.out.println("Kecepatan: " + kecepatan);
        System.out.println("Gear: " + gear);
    }
}
```

SepedaDemo.java

```
public class sepedaDemo {  
    public static void main(String[] args) {  
        Sepeda spd1 = new Sepeda ();  
        Sepeda spd2 = new Sepeda ();  
  
        spd1.setMerek("Polygon");  
        spd1.tambahKecepatan(10);  
        spd1.gantiGear(2);  
        spd1.cetakStatus();  
  
        spd2.setMerek("Wiim Cycle");  
        spd2.tambahKecepatan(10);  
        spd2.gantiGear(2);  
        spd2.tambahKecepatan(10);  
        spd2.gantiGear(3);  
        spd2.cetakStatus();  
    }  
}
```

Berikut adalah Output dari Percobaan 1 :

```
PS D:\Pemrograman Berbasis Object> d:; cd 'd:\Pemrograman Berbasis Object'; &  
'C:\Program Files\Java\jdk-22\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMes  
sages' '-cp' 'C:\Users\syahr\AppData\Roaming\Code\User\workspaceStorage\546d3bd  
1710dca98398227c6fdf48017\redhat.java\jdt_ws\Pemrograman Berbasis Object_b3ef01  
70\bin' 'sepedaDemo'  
Merek: Polygon  
Kecepatan: 10  
Gear: 2  
Merek: Wiim Cycle  
Kecepatan: 20  
Gear: 3  
PS D:\Pemrograman Berbasis Object> █
```

PERCOBAAN 2

Source Code :

New Class

```
public class SepedaGunung extends Sepeda {
    private String tipeSuspensi;

    public void setTypeSuspensi (String newValue)
    {
        tipeSuspensi = newValue;
    }

    public void cetakStatus ()
    {
        super.cetakStatus();
        System.out.println("Tipe Suspensi: " + tipeSuspensi);
    }
}
```

Modify Demo :

```
public class sepedaDemo {
    public static void main(String[] args) {
        Sepeda spd1 = new Sepeda ();
        Sepeda spd2 = new Sepeda ();
        SepedaGunung spd3 = new SepedaGunung ();

        spd1.setMerek("Polygon");
        spd1.tambahKecepatan(10);
        spd1.gantiGear(2);
        spd1.cetakStatus();

        spd2.setMerek("Wiim Cycle");
        spd2.tambahKecepatan(10);
        spd2.gantiGear(2);
        spd2.tambahKecepatan(10);
        spd2.gantiGear(3);
        spd2.cetakStatus();

        spd3.setMerek("Kiinee");
        spd3.tambahKecepatan(5);
        spd3.gantiGear(7);
        spd3.setTypeSuspensi("Gas Suspension");
        spd3.cetakStatus();
    }
}
```

Output:

```
Merek: Polygon
Kecepatan: 10
Gear: 2
Merek: Wiim Cycle
Kecepatan: 20
Gear: 3
Merek: Kiinee
Kecepatan: 5
Gear: 7
Tipe Suspensi: Gas Suspension
PS D:\Pemrograman Berbasis Object>
```

JAWABAN PERTANYAAN

1. Perbedaan Class vs Objek

- Class adalah cetak biru, atau template yang belum ada isinya, nanti bisa di isi sebuah object
- Object adalah sebuah data yang di isikan di dalam class, setiap object mempunyai nilai masing masing

2. Ya karena atribut warna dan tipe mesin untuk membedakan tiap jenis jenis mobil Ada berbagai warna yang bisa di pilih oleh konsumen : merah, Putih, Hitam yang membedakan Tampilannya

Dan ada juga jenis tipe mesin yang di pilih Bensin, EV, Diesel yang membedakan jenis mesin dan pengisian bahan bakarnya

3. Lebih mudah dirawat & dipakai ulang (modular). Lalu bisa di gunakan kembali contohnya dengan inheritance

4. Boleh ketika 1 tipe data

```
public String nama, alamat;  
public String nama = "A", alamat="B";
```

tidak boleh ketika berbeda TD

```
public String nama, int umur;
```

Lebih baik 1 baris 1 TD saja sih menurut saya, biar lebih rapi dan terstruktur.

5. Ya karena dia sudah mewarisi semua atribut dari class Sepeda.

Pada kode ini

```
public class SepedaGunung extends Sepeda
```