

LAPORAN JOBSHEET 1

PENGANTAR KONSEP PBO

Mata Kuliah : Praktikum Pemrograman Berbasis Objek

Dosen : Irsyad Arif Mashudi, S.Kom., M.Kom



Ilham Dharma Atmaja
24410702020
Kelas :TI 2D

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG TAHUN 2025

A. Tujuan

1. Memahami perbedaan paradigma pemrograman struktural dan berorientasi objek.
2. Mengenal konsep dasar Pemrograman Berorientasi Objek (PBO).
3. Mampu membuat class, objek, dan memanggil method dalam Java.

B. Dasar Teori

Pemrograman Berorientasi Objek (PBO) adalah paradigma pemrograman yang memecah program menjadi objek, yang masing-masing memiliki state (atribut) dan behavior (method).

Konsep Dasar PBO:

- Object: Representasi nyata dalam program. Contoh: Objek Sepeda memiliki atribut merek, kecepatan, dan gear.
- Class: Blueprint atau cetak biru dari objek.
- Encapsulation: Penyembunyian detail internal objek untuk mempermudah penggunaannya.
- Inheritance: Pewarisan sifat/fitur dari satu class ke class lainnya.
- Polymorphism: Kemampuan objek untuk memiliki bentuk/metode berbeda dengan antarmuka yang sama.

Dengan OOP, kode menjadi lebih modular, mudah dipelihara, dan dapat digunakan kembali.

C. Percobaan 1

Membuat class Sepeda dan SepedaDemo untuk mendemonstrasikan objek, atribut, dan method.

1. Class Sepeda

```
SepedaDemo > Sepeda.java > Sepeda > gear
1 package SepedaDemo;
2 public class Sepeda
3 {
4     private String merek;
5     private int kecepatan;
6     private int gear;
7
8     public void setMerek(String newValue)
9     {
10         merek = newValue;
11     }
12
13     public void gantiGear(int newValue)
14     {
15         gear = newValue;
16     }
17
18     public void tambahKecepatan(int increment)
19     {
20         kecepatan = kecepatan + increment;
21     }
22
23     public void rem(int decrement)
24     {
25         kecepatan = kecepatan - decrement;
26     }
27
28     public void cetakStatus()
29     {
30         System.out.println("Merek: " + merek);
31         System.out.println("Kecepatan: " + kecepatan);
32         System.out.println("Gear: " + gear);
33     }
34 }
```

2. Class SepedaDemo

```
SpedaDEmo > J SepedaDemo.java > SepedaDemo > main(String[])
1  package SepedaDEmo;
2  public class SepedaDemo
3  {
4      Run | Debug
      public static void main(String[] args)
5      {
6          // dua buah objek sepeda
7          Sepeda spd1 = new Sepeda();
8          Sepeda spd2 = new Sepeda();
9
10         // Panggil method di dalam objek sepeda
11         spd1.setMerek(newValue:"Polygon");
12         spd1.tambahKecepatan(increment:10);
13         spd1.gantiGear(newValue:2);
14         spd1.cetakStatus();
15
16         spd2.setMerek(newValue:"Wiim Cycle");
17         spd2.tambahKecepatan(increment:10);
18         spd2.gantiGear(newValue:2);
19         spd2.tambahKecepatan(increment:10);
20         spd2.gantiGear(newValue:3);
21         spd2.cetakStatus();
22     }
23 }
24
```

3. Verifikasi Hasil

```
Merek: Polygon
Kecepatan: 10
Gear: 2
Merek: Wiim Cycle
Kecepatan: 20
Gear: 3
PS C:\Users\ILHAM DHARMA A\OneDrive\Documents\SEMESTER 3\PBO Pak Irsyad\PRAKTIKUM\PROGRAM>
```

D. Percobaan 2

1. Class SepedaGunung

```
SpedaDEmo > J SepedaGunung.java > ...
1  public class SepedaGunung extends Sepeda
2  {
3      private String tipeSuspensi;
4
5      public void setTypeSuspensi(String newValue)
6      {
7          tipeSuspensi = newValue;
8      }
9
10     public void cetakStatus()
11     {
12         super.cetakStatus();
13         System.out.println("Tipe suspensi: " + tipeSuspensi);
14     }
15 }
16
```

2. Modif Class SepedaDemo

```
SpedaDEmo > J SepedaDemo.java > SepedaDemo
1  public class SepedaDemo
2  {
3      Run | Debug
4      public static void main(String[] args)
5      {
6          // dua buah objek sepeda
7          Sepeda spd1 = new Sepeda();
8          Sepeda spd2 = new Sepeda();
9          SepedaGunung spd3 = new SepedaGunung();
10
11         // Panggil method di dalam objek sepeda
12         spd1.setMerek(newValue:"Polygon");
13         spd1.tambahKecepatan(increment:10);
14         spd1.gantiGear(newValue:2);
15         spd1.cetakStatus();
16
17         spd2.setMerek(newValue:"Wiim Cycle");
18         spd2.tambahKecepatan(increment:10);
19         spd2.gantiGear(newValue:2);
20         spd2.tambahKecepatan(increment:10);
21         spd2.gantiGear(newValue:3);
22         spd2.cetakStatus();
23
24         spd3.setMerek(newValue:"Klinee");
25         spd3.tambahKecepatan(increment:5);
26         spd3.gantiGear(newValue:7);
27         spd3.setTipeSuspensi(newValue:"Gas suspension");
28         spd3.cetakStatus();
29     }
30 }
```

3. Verifikasi Hasil

```
Merek: Polygon
Kecepatan: 10
Gear: 2
Merek: Wiim Cycle
Kecepatan: 20
Gear: 3
Merek: Klinee
Kecepatan: 5
Gear: 7
Tipe suspensi: Gas suspension
PS C:\Users\ILHAM DHARMA A\OneDrive\Documents\SEMESTER 3\PBO Pak Irsyad\PRAKTIKUM\PROGRAM>
```

E. Jawaban Pertanyaan

1. Perbedaan antara objek dengan class:

- **Class** adalah blueprint atau template yang mendefinisikan atribut dan perilaku (method) dari suatu entitas. Class belum memiliki data nyata.
- **Objek** adalah instance dari class, yaitu wujud nyata yang memiliki data/atribut sesuai dengan blueprint class.

2. Alasan warna dan tipe mesin dapat menjadi atribut dari objek mobil:

Karena **warna** dan **tipe mesin** adalah karakteristik atau ciri-ciri yang membedakan setiap mobil. Setiap objek mobil bisa memiliki warna dan tipe mesin yang berbeda-beda, sehingga cocok dijadikan atribut.

3. **Salah satu kelebihan utama OOP dibandingkan pemrograman struktural:**

Enkapsulasi: Data dan perilaku dibungkus dalam satu kesatuan (class), sehingga lebih mudah dalam pengelolaan, pemeliharaan, dan keamanan data.

4. **Pendefinisian dua atribut dalam satu baris seperti `public String nama,alamat;`**

Diperbolehkan. Kode tersebut akan membuat dua atribut bertipe String, yaitu nama dan alamat.

5. **Alasan atribut merk, kecepatan, dan gear tidak lagi ditulis di dalam class SepedaGunung:**

Karena class SepedaGunung **mewarisi** (inherit) atribut-atribut tersebut dari class induknya (Sepeda). Dengan pewarisan, atribut dan method dari class induk otomatis dimiliki oleh class turunan, sehingga tidak perlu dideklarasikan ulang.

F. Tugas Praktikum

1. **Foto Objek**

1. **Laptop**



2. Smartphone



3. Laptop Gaming



4. Laptop Ultrabook



2. Program

1. Class Laptop

```
Minggu 1 > SpedaDEmo > TugasPraktik > J Laptop.java > ...
1  public class Laptop {
2      protected String merek;
3      protected String prosesor;
4
5      public void setData(String merek, String prosesor) {
6          this.merek = merek;
7          this.prosesor = prosesor;
8      }
9
10     public void hidupkan() {
11         System.out.println(merek + " dengan prosesor " + prosesor + " sedang dinyalakan...");
12     }
13
14     public void cetakInfo() {
15         System.out.println("Merek: " + merek);
16         System.out.println("Prosesor: " + prosesor);
17     }
18 }
19
```

2. Class LaptopGaming

```
Minggu 1 > SpedaDEmo > TugasPraktik > J LaptopGaming.java > ...
1  public class LaptopGaming extends Laptop {
2      private String vga;
3      private int ram;
4
5      public void setGamingSpec(String vga, int ram) {
6          this.vga = vga;
7          this.ram = ram;
8      }
9
10     public void mainGame() {
11         System.out.println("Laptop gaming " + merek + " sedang digunakan untuk bermain game berat!");
12     }
13
14     @Override
15     public void cetakInfo() {
16         super.cetakInfo();
17         System.out.println("VGA: " + vga);
18         System.out.println("RAM: " + ram + "GB");
19     }
20 }
21
```

3. Class LaptopUltrabook

```

Minggu 1 > SpedaDEmo > TugasPraktik > J LaptopUltrabook.java > ...
1  public class LaptopUltrabook extends Laptop {
2      private double berat;
3      private double ketebalan;
4
5      public void setUltrabookSpec(double berat, double ketebalan) {
6          this.berat = berat;
7          this.ketebalan = ketebalan;
8      }
9
10     public void hematBaterai() {
11         System.out.println("Ultrabook " + merek + " dalam mode hemat baterai...");
12     }
13
14     @Override
15     public void cetakInfo() {
16         super.cetakInfo();
17         System.out.println("Berat: " + berat + " kg");
18         System.out.println("Ketebalan: " + ketebalan + " mm");
19     }
20 }
21

```

4. Class Smartphone

```

Minggu 1 > SpedaDEmo > TugasPraktik > J Smartphone.java > ...
1  public class Smartphone {
2      private String merek;
3      private int kapasitasBaterai;
4
5      public void setData(String merek, int baterai) {
6          this.merek = merek;
7          this.kapasitasBaterai = baterai;
8      }
9
10     public void telfon() {
11         System.out.println("Smartphone " + merek + " sedang melakukan panggilan...");
12     }
13
14     public void cetakInfo() {
15         System.out.println("Merek Smartphone: " + merek);
16         System.out.println("Kapasitas Baterai: " + kapasitasBaterai + "mAh");
17     }
18 }
19

```

5. Class Main


```

Minggu 1 > SpedaDEmo > TugasPraktik > Demo.java
1  public class Demo {
    Run | Debug
2      public static void main(String[] args) {
3
4          // Objek Laptop Gaming
5          LaptopGaming lg = new LaptopGaming();
6          lg.setData(merek:"Asus ROG", prosesor:"Intel i9");
7          lg.setGamingSpec(vga:"NVIDIA RTX 4080", ram:32);
8          lg.hidupkan();
9          lg.mainGame();
10         lg.cetakInfo();
11         System.out.println();
12
13         // Objek Laptop Ultrabook
14         LaptopUltrabook lu = new LaptopUltrabook();
15         lu.setData(merek:"Macbook Air", prosesor:"Apple M2");
16         lu.setUltrabookSpec(berat:1.2, ketebalan:13);
17         lu.hidupkan();
18         lu.hematBaterai();
19         lu.cetakInfo();
20         System.out.println();
21
22         // Objek Smartphone
23         Smartphone hp = new Smartphone();
24         hp.setData(merek:"Samsung Galaxy S24", baterai:5000);
25         hp.telfon();
26         hp.cetakInfo();
27     }
28 }
29

```

6. Verifikasi Hasil

```

Asus ROG dengan prosesor Intel i9 sedang dinyalakan...
Laptop gaming Asus ROG sedang digunakan untuk bermain game berat!
Merek: Asus ROG
Prosesor: Intel i9
VGA: NVIDIA RTX 4080
RAM: 32GB

Macbook Air dengan prosesor Apple M2 sedang dinyalakan...
Ultrabook Macbook Air dalam mode hemat baterai...
Merek: Macbook Air
Prosesor: Apple M2
Berat: 1.2 kg
Ketebalan: 13.0 mm

Smartphone Samsung Galaxy S24 sedang melakukan panggilan...
Merek Smartphone: Samsung Galaxy S24
Kapasitas Baterai: 5000mAh
PS C:\Users\ILHAM DHARMA A\OneDrive\Documents\SEMESTER 3\PBO Pak Irsyad\PRAKTIKUM\PROGRAM>

```

