Tugas Praktikum Pertemuan 1

Pengantar Konsep Pemrograman Berorientasi Objek



Rafi Ody Prasetyo  
(2341720180) / TI-2D

D-IV Teknik Informatika  
Politeknik Negeri Malang  
Semester 3  
2024

**Percobaan 1**

Code:

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 += increment;

}

public void rem(int decrement) {

kecepatan -= decrement;

}

public void cetakStatus() {

System.out.println("Merek: "+merek);

System.out.println("Kecepatan: "+kecepatan);

System.out.println("Gear: "+gear);

}

}

Output:

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("Whim Cycle");

spd2.tambahKecepatan(10);

spd2.gantiGear(2);

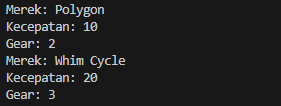
spd2.tambahKecepatan(10);

spd2.gantiGear(3);

spd2.cetakStatus();

}

}



Github: <https://github.com/rafiody16/PBO/tree/main/Jobsheet1/sepedaDemo>

**Percobaan 2**

Code:

public class SepedaDemo {

public static void main(String[] args) {

Sepeda spd1 = new Sepeda();

Sepeda spd2 = new Sepeda();

SepedaGunung spd3 = new SepedaGunung();

...

spd3.setMerek("Klinee");

spd3.tambahKecepatan(5);

spd3.gantiGear(7);

spd3.setTipeSuspensi("Gas Suspension");

spd3.cetakStatus();

}

}

public class SepedaGunung extends Sepeda {

private String tipeSuspensi;

public void setTipeSuspensi(String newValue) {

tipeSuspensi = newValue;

}

public void cetakStatus() {

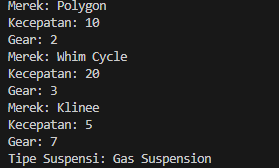
super.cetakStatus();

System.out.println("Tipe Suspensi: "+tipeSuspensi);

}

}

Output:



Github: <https://github.com/rafiody16/PBO/blob/main/Jobsheet1/sepedaDemo/SepedaGunung.java>

**Pertanyaan**

1. **Jelaskan perbedaan antara objek dengan class!**

**Jawab:**

Class berfungsi untuk menampung program yang akan dijalankan, sedangkan objek merupakan definisi dari benda nyata yang di representasikan kedalam program.

1. **Jelaskan alasan warna dan tipe mesin dapat menjadi atribut dari objek mobil!**

**Jawab:**

Karena warna dan tipe mesin merupakan bagian dari mobil, sehingga dapat dijadikan sebagai atribut.

1. **Sebutkan salah satu kelebihan utama dari pemrograman berorientasi objek dibandingkan dengan pemrograman struktural!**

**Jawab:**

Pemrograman berorientasi objek lebih muda dibaca dan dipahami dibandingkan dengan pemrograman struktural.

1. **Apakah diperbolehkan melakukan pendefinisian dua buah atribut dalam satu baris kode seperti “public String nama,alamat;”?**

**Jawab:**

Boleh, apabila memiliki tipe data yang sama.

1. **Pada class SepedaGunung, jelaskan alasan atribut merk, kecepatan, dan gear tidak lagi ditulis didalam class tersebut!**

**Jawab:**

Karena pada class SepedaGunung mewarisi/terhubung class Sepeda sehingga beberapa atribut tidak lagi ditulis.

**Praktikum**

1. **Handphone**

Code:

public class Handphone {

private String merek, warna, os;

private int sound;

public void setMerek(String newValue) {

merek = newValue;

}

public void setWarna(String newValue) {

warna = newValue;

}

public void setOs(String newValue) {

os = newValue;

}

public void upVolume(int increment) {

sound += increment;

}

public void downVolume(int decrement) {

sound -= decrement;

}

public void cetakStatus() {

System.out.println("Merek: "+merek);

System.out.println("Warna: "+warna);

System.out.println("OS: "+os);

System.out.println("Volume: "+sound);

}

}

Output:

public class HandphoneMain {

public static void main(String[] args) {

Handphone hp1 = new Handphone();

hp1.setMerek("Samsung");

hp1.setWarna("Hitam");

hp1.setOs("Android");

hp1.upVolume(5);

hp1.downVolume(3);

hp1.upVolume(2);

hp1.cetakStatus();

}

}



1. **Kulkas**

Code:

public class Kulkas {

private String merek, warna;

private int suhu;

public void setMerek(String newValue) {

merek = newValue;

}

public void setWarna(String newValue) {

warna = newValue;

}

public void upSuhu(int increment) {

suhu += increment;

}

public void downSuhu(int decrement) {

suhu -= decrement;

}

public void cetakStatus() {

System.out.println("Merek: "+merek);

System.out.println("Warna: "+warna);

System.out.println("Suhu: "+suhu);

}

}

Output:

public class KulkasMain {

public static void main(String[] args) {

Kulkas ks1 = new Kulkas();

ks1.setMerek("LG");

ks1.setWarna("Putih");

ks1.upSuhu(3);

ks1.downSuhu(2);

ks1.cetakStatus();

}

}

