

LAPORAN PRAKTIKUM

OOP

JOBSHEET 4



Disusun oleh:

Pascalis Dewangga S. L. 2241720140

TI - 2D

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2023

1. Percobaan 1

a. Kode Program

Class Processor

```
public class Processor {
    private String merk;
    private double cache;

    Processor(String merk, double cache) {
        this.merk = merk;
        this.cache = cache;
    }

    public Processor() {
    }

    public String getMerk() {
        return merk;
    }

    public double getCache() {
        return cache;
    }

    public void setMerk(String newMerk) {
        merk = newMerk;
    }

    public void setCache(double newCache) {
        cache = newCache;
    }

    public void info() {
        System.out.printf("Merk Processor = %s\n", merk);
        System.out.printf("Cache Memory = %.2f\n", cache);
    }
}
```

Class Laptop

```
public class Laptop {
    private String merk;
    private Processor proc;

    Laptop(String merk, Processor proc) {
        this.merk = merk;
        this.proc = proc;
    }

    public Laptop() {
    }
}
```

```

    }

    public String getMerk() {
        return merk;
    }

    public Processor getProc() {
        return proc;
    }

    public void setMerk(String newMerk) {
        merk = newMerk;
    }

    public void setProc(Processor newProc) {
        proc = newProc;
    }

    public void info() {
        System.out.println("Merk Laptop = " + merk);
        proc.info();
    }
}

```

Class MainPercobaan1

```

public class MainPercobaan1 {
    public static void main(String[] args) {
        Processor p = new Processor("Intel i5", 3);
        Laptop L = new Laptop("Thinkpad", p);

        L.info();

        Processor p1 = new Processor();
        p1.setMerk("Intel i5");
        p1.setCache(4);
        Laptop l1 = new Laptop();
        l1.setMerk("Thinkpad");
        l1.setProc(p1);
        l1.info();
    }
}

```

b. Hasil

```

Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 3.00
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 4.00

```

c. Pertanyaan

1. Untuk metode setter sendiri berfungsi untuk melakukan set sebuah isian array dengan parameter yang diinginkan/ melakukan pengisian pada sebuah atribut. Pengisian tersebut dilakukan dari class main.
2. Untuk metode setter sendiri berfungsi untuk melakukan set sebuah isian array dengan parameter yang diinginkan/ melakukan pengisian pada sebuah atribut. Pengisian tersebut dilakukan dari class main.
3. Yang bertipe object adalah proc, dikarenakan merupakan sebuah objek dari class Processor.
4. Baris 3 yang berisikan Private Processor proc;
5. proc.info() pada method info di class Laptop berfungsi untuk memanggil method info dari class Processor yang berisikan print dari merk processor dan cache memory dari laptop
6. Setelah dirubah, hasilnya:

```
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 3.00
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 4.00
```

Tidak ada perubahan.

2. Percobaan 2

a. Kode Program

Class Mobil

```
public class Mobil {
    private String merk;
    private int biaya;

    public Mobil() {
    }

    public String getMerk() {
        return merk;
    }

    public int getBiaya() {
        return biaya;
    }

    public void setMerk(String newMerk) {
        merk = newMerk;
    }

    public void setBiaya(int newBiaya) {
        biaya = newBiaya;
    }
}
```

```
        public int hitungBiayaMobil(int hari) {  
            return biaya * hari;  
        }  
    }  
}
```

Class Sopir

```
public class Sopir {  
    private String nama;  
    private int biaya;  
  
    Sopir() {  
    }  
  
    public String getNama() {  
        return nama;  
    }  
  
    public int getBiaya() {  
        return biaya;  
    }  
  
    public void setNama(String newName) {  
        nama = newName;  
    }  
  
    public void setBiaya(int newBiaya) {  
        biaya = newBiaya;  
    }  
  
    public int hitungBiayaSopir(int hari) {  
        return biaya * hari;  
    }  
}
```

Class Pelanggan

```
public class Pelanggan {  
    private String nama;  
    private Mobil mobil;  
    private Sopir sopir;  
    private int hari;  
  
    Pelanggan() {  
    }  
  
    public String getNama() {  
        return nama;  
    }  
}
```

```

public Mobil getMobil() {
    return mobil;
}

public Sopir getSopir() {
    return sopir;
}

public int getHari() {
    return hari;
}

public void setNama(String newNama) {
    nama = newNama;
}

public void setMobil(Mobil newMobil) {
    mobil = newMobil;
}

public void setSopir(Sopir newSopir) {
    sopir = newSopir;
}

public void setHari(int newHari) {
    hari = newHari;
}

public int hitungBiayaTotal() {
    return mobil.hitungBiayaMobil(hari) + sopir.hitungBiayaSopir(hari);
}
}

```

Class MainPercobaan2

```

public class MainPercobaan2 {
    public static void main(String[] args) {
        Mobil m = new Mobil();
        m.setMerk("Avanza");
        m.setBiaya(350000);
        Sopir s = new Sopir();
        s.setNama("John Doe");
        s.setBiaya(200000);
        Pelanggan p = new Pelanggan();
        p.setMobil(m);
        p.setSopir(s);
        p.setHari(2);
    }
}

```

```

        System.out.println("Biaya Total = " + p.hitungBiayaTotal());
    }
}

```

b. Hasil

```

Biaya Total = 1100000

```

c. Pertanyaan

1. - Baris 3 dan 4 (private Mobil mobil; private Sopir sopir;)
 - Baris 14 – 19 (public void setMobil(Mobil mobil){ this.mobil = mobil; } public void setSopir(Sopir sopir){ this.sopir = sopir; })
 - Baris 28 – 34 (public Mobil getMobil(){ return mobil; }) public Sopir sopir(){ return sopir; } } public String nama(){ return nama; } public int biaya(){ return biaya; } } Page 12 of 10)
 - Baris 40 – 44 (public int hitungBiayaTotal(){ return mobil.hitungBiayaMobil(hari) + sopir.hitungBiayaSopir(hari); })
2. Dikarenakan dalam menghitung tarif mobil dan supir bergantung pada jumlah hari. Semakin lama maka semakin mahal tarifnya.
3. Untuk menghitung biaya total pembayaran ketika kita menyewa 2 class tersebut
4. Fungsi nya adalah untuk melakukan set Mobil yang ingin disewa yaitu Mobil m. m tersebut mengacu pada Avanza dengan biaya 350000 per hari dan Sopir s yang mengacu pada sopir dengan nama John Doe dengan harga sewa 200000 per hari.
5. Untuk memanggil method pada class Pelanggan dimana di dalamnya terdapat perhitungan dari biaya mobil + biaya sopir.
6. Pertama tama, fungsi P.getmobil() adalah untuk memanggil method getmobil dengan isian dari instansiasi variabel P pada class Pelanggan, lalu isian dari method tersebut adalah untuk mengembalikan pemanggilan class Mobil. Selanjutnya terdapat getMerk() yang berfungsi untuk mereturn merk mobil pada Class mobil yang sudah di tetapkan oleh instansiasi dari variabel P yaitu Avanza (p.setMobil(m) dimana m.setmerk("Avanza"))

3. Percobaan 3

a. Kode Program

Class Pegawai

```

public class Pegawai {
    private String nip;
    private String nama;

    Pegawai(String nip, String nama) {
        this.nip = nip;
        this.nama = nama;
    }

    public String getNip() {
        return nip;
    }
}

```

```

public String getNama() {
    return nama;
}

public void setNip(String newNip) {
    nip = newNip;
}

public void setNama(String newNama) {
    nama = newNama;
}

public String info() {
    String info = " ";
    info += "Nip : " + this.nip + "\n";
    info += "Nama: " + this.nama + "\n";
    return info;
}
}

```

Class KeretaApi

```

public class KeretaApi {
    private String nama;
    private String kelas;
    private Pegawai Masinis;
    private Pegawai Asisten;

    KeretaApi(String nama, String kelas, Pegawai Masinis) {
        this.nama = nama;
        this.kelas = kelas;
        this.Masinis = Masinis;
    }

    KeretaApi(String nama, String kelas, Pegawai Masinis, Pegawai Asisten) {
        this.nama = nama;
        this.kelas = kelas;
        this.Masinis = Masinis;
        this.Asisten = Asisten;
    }

    public String getNama() {
        return nama;
    }

    public String getKelas() {
        return kelas;
    }
}

```



```

public void setNama(String newNama) {
    nama = newNama;
}

public void setKelas(String newKelas) {
    kelas = newKelas;
}

public String info() {
    String info = "";
    info += "Nama    : " + this.nama + "\n";
    info += "Kelas   : " + this.kelas + "\n";
    info += "Masinis: " + this.Masinis.info() + "\n";
    info += "Asisten: " + this.Asisten.info() + "\n";
    return info;
}
}

```

Class MainPercobaan3

```

public class MainPercobaan3 {
    public static void main(String[] args) {
        Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");
        Pegawai asisten = new Pegawai("4567", "Patrick Star");
        KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis,
asisten);

        System.out.println(keretaApi.info());
    }
}

```

b. Hasil

```

Nama    : Gaya Baru
Kelas   : Bisnis
Masinis: Nip : 1234
Nama: Spongebob Squarepants

Asisten: Nip : 4567
Nama: Patrick Star

```

c. Pertanyaan

1. Baris `this.masinis.info()` berfungsi untuk memanggil method `info()` pada class pegawai dikarenakan masinis memiliki tipe data class Pegawai sehingga method tersebut menuju ke class pegawai. Lalu isi method `info()` pada class pegawai

adalah melakukan print nip dan nama yang sudah di instansiasi pada class MainPercobaan3 dengan nama instan Masinis.

2. Sudah
3. Muncul Error dikarenakan variabel Asisten yang masih bernilai null, sehingga tidak terdeteksi.

```
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "Pegawai.info()" because "this.Asisten" is null
    at KeretaApi.info(KeretaApi.java:41)
    at MainPercobaan3.main(MainPercobaan3.java:7)
```

4. Tambahkan pada Class KeretaApi

```
public String infoMasinis() {
    String info = "";
    info += "Nama: " + this.nama + "\n";
    info += "Kelas: " + this.kelas + "\n";
    info += "Masinis: " + this.Masinis.info() + "\n";
    return info;
}
```

Pemanggilan pada Main

```
System.out.println(keretaApi.infoMasinis());
```

Hasil:

```
Nama: Gaya Baru
Kelas: Bisnis
Masinis: Nip : 1234
Nama: Spongebob Squarepants
```

3. Percobaan 4
 - a. Kode Program
- Class Penumpang

```
public class Penumpang {
    private String ktp;
    private String nama;

    Penumpang(String ktp, String nama){
        this.ktp = ktp;
        this.nama = nama;
    }

    public String getKtp(){
        return ktp;
    }

    public String getNama(){
        return nama;
    }

    public void setKtp(String ktp){
```

```

        this.ktp = ktp;
    }

    public void setNama(String nama){
        this.nama = nama;
    }

    public String info(){
        String info = "";
        info += "Ktp: " + ktp + "\n";
        info += "Nama: " + nama + "\n";
        return info;
    }
}

```

Class Kursi

```

public class Kursi {
    private String nomor;
    Penumpang penumpang;

    Kursi(String nomor) {
        this.nomor = nomor;
    }

    public String getNomor() {
        return nomor;
    }

    public Penumpang getPenumpang() {
        return penumpang;
    }

    public void setPenumpang(Penumpang penumpang) {
        this.penumpang = penumpang;
    }

    public String info() {
        String info = "";
        info += "Nomor: " + nomor + "\n";
        if (this.penumpang != null) {
            info += "Penumpang: " + penumpang.info() + "\n";
        }
        return info;
    }
}

```

Class Gerbong

```
public class Gerbong {
    private String kode;
    private Kursi[] kursi;

    private void initKursi() {
        for (int i = 0; i < kursi.length; i++) {
            this.kursi[i] = new Kursi(String.valueOf(i + 1));
        }
    }

    public Gerbong(String kode, int jumlah) {
        this.kode = kode;
        this.kursi = new Kursi[jumlah];
        this.initKursi();
    }

    public String getKode() {
        return kode;
    }

    public Kursi[] kursi() {
        return kursi;
    }

    public void setPenumpang(Penumpang penumpang, int nomor) {
        this.kursi[nomor - 1].setPenumpang(penumpang);
    }

    public String info() {
        String info = "";
        info += "Kode: " + kode + "\n";
        for (Kursi kursi : kursi) {
            info += kursi.info();
        }
        return info;
    }
}
```

Class MainPercobaan4

```
public class MainPercobaan4 {
    public static void main(String[] args){
        Penumpang p = new Penumpang("12345", "Mr. Krab");
        Gerbong gerbong = new Gerbong("A", 10);
        gerbong.setPenumpang(p, 1);
    }
}
```

```
        System.out.println(gerbong.info());  
    }  
}
```

b. Hasil

```
Kode: A  
Nomor: 1  
Penumpang: Ktp: 12345  
Nama: Mr. Krab  
  
Nomor: 2  
Nomor: 3  
Nomor: 4  
Nomor: 5  
Nomor: 6  
Nomor: 7  
Nomor: 8  
Nomor: 9  
Nomor: 10
```

c. Pertanyaan Percobaan 4

1. Kursi berjumlah 10
2. If tersebut mengecek jika `this.penumpang != null` atau sudah memiliki isi, maka kita bisa melakukan print info penumpang dengan isian ktp dan nama penumpang itu sendiri. Print penumpang dilakukan dari class penumpang dengan method `info`.
3. Dikarenakan agar kita bisa mengisi array sebelumnya, seperti contoh jika nomor yang dimasukkan adalah 1 maka nomor 1 akan dikurangi – sehingga array tersebut menjadi indeks 0. Dan pastinya array akan dimulai dari indeks ke 0 dan akan bertahap hingga selanjutnya.
4. Hasil:

```
Kode: A  
Nomor: 1  
Penumpang: Ktp: 23456  
Nama: Budi  
  
Nomor: 2  
Nomor: 3  
Nomor: 4  
Nomor: 5  
Nomor: 6  
Nomor: 7  
Nomor: 8  
Nomor: 9  
Nomor: 10
```

5. Kode Program

Tambahan Class Gerbong

```
public void setPenumpang(Penumpang penumpang, int nomor) {  
    Kursi kursi = this.kursi[nomor - 1];  
    if (!kursi.isTerisi()) {  
        kursi.setPenumpang(penumpang);  
    } else {  
        System.out.println("Kursi nomor " + nomor + " sudah terisi.");  
    }  
}
```

Tambahan Class Kursi

```
public boolean isTerisi() {  
    return penumpang != null;  
}
```

Hasil:

```
Kursi nomor 1 sudah terisi.  
Kode: A  
Nomor: 1  
Penumpang: Ktp: 12345  
Nama: Mr. Krab  
  
Nomor: 2  
Nomor: 3  
Nomor: 4  
Nomor: 5  
Nomor: 6  
Nomor: 7  
Nomor: 8  
Nomor: 9  
Nomor: 10
```