

LAPORAN PRAKTIKUM
Pemrograman Berbasis Objek

Jobsheet - 4: Relasi



Nama : Ghoffar Abdul Ja'far

NIM : 2341720035

Kelas : 2F

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2023/2024

Percobaan 1

- **Code:**

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

    public Laptop () {
    }

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

    public void setProc (Processor proc) {
        this.proc = proc;
    }
    public void setMerk (String merk) {
        this.merk = merk;
    }

    }

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

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

    public Processor () {
    }
    public Processor (String merk, double cache) {
        this.merk = merk;
        this.cache = cache;
    }
    public String getMerk () {
        return merk;
    }
    public void setMerk (String merk) {
        this.merk = merk;
    }
    public double getCache () {
        return cache;
    }
    public void setCache (double cache) {
        this.cache = cache;
    }
    public void info () {
        System.out.printf ("Merk Processor = %s\n", merk);
        System.out.printf ("Cache Memory = %.2f\n", cache);
    }
}

public class JavaKomputer {

    /**
     * @param args the command line arguments
     */
    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 ();
    }
}
```

- **Hasil:**

```
Merk Laptop = Thinkpad
Merk Processor = Intel i5
  Cache Memory = 3,00
Merk Laptop = Thinkpad
Merk Processor = Intel i5
  Cache Memory = 4,00
```

Pertanyaan:

1. Di dalam class Processor dan class Laptop , terdapat method setter dan getter untuk masing-masing atributnya. Apakah gunanya method setter dan getter tersebut ?
Jawab: setter digunakan untuk mengubah atau memasukkan data, sedangkan getter digunakan untuk mengambil data
2. Di dalam class Processor dan class Laptop, masing-masing terdapat konstruktor default dan konstruktor berparameter. Bagaimanakah beda penggunaan dari kedua jenis konstruktor tersebut ?

Jawab: konstruktor default perlu getter dan setter untuk pengolahan data, sedangkan konstruktor berparameter sudah mencakup getter dan setter, bisa langsung diinputkan saat melakukan instansiasi object

3. Perhatikan class Laptop, di antara 2 atribut yang dimiliki (merk dan proc), atribut manakah yang bertipe object ?

Jawab: atribut proc

4. Perhatikan class Laptop, pada baris manakah yang menunjukkan bahwa class Laptop memiliki relasi dengan class Processor ?

Jawab: pada saat deklarasi atribut proc pada class Laptop, karena atribut proc merupakan object

5. Perhatikan pada class Laptop, Apakah guna dari sintaks proc.info() ?

Jawab: digunakan untuk menampilkan informasi processor

6. Pada class MainPercobaan1, terdapat baris kode:

Laptop l = new Laptop("Thinkpad", p);.

Apakah p tersebut ?

Jawab: p sendiri adalah object dari class Processor

Dan apakah yang terjadi jika baris kode tersebut diubah menjadi:

Laptop l = new Laptop("Thinkpad", new Processor("Intel i5", 3));

Bagaimanakah hasil program saat dijalankan, apakah ada perubahan ?

Jawab: hasilnya akan tetap sama, hanya saja instansiasi object preocessor dilakukan langsung pada saat instansiasi object laptop

Percobaan 2

- Code:

```
public class Pelanggan {
    private String nama;
    private int hari;
    Mobil mobil;
    Sopir sopir;

    public Pelanggan() {
    }

    public String getNama() {
        return nama;
    }

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

    public int getHari() {
        return hari;
    }

    public void setHari(int hari) {
        this.hari = hari;
    }

    public Mobil getMobil() {
        return mobil;
    }

    public void setMobil(Mobil mobil) {
        this.mobil = mobil;
    }
}
```

```
public Sopir getSopir() {
    return sopir;
}

public void setSopir(Sopir sopir) {
    this.sopir = sopir;
}

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

```

public class Sopir {
    private String nama;
    private int biaya;

    public Sopir() {
    }

    public String getNama() {
        return nama;
    }

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

    public int getBiaya() {
        return biaya;
    }

    public void setBiaya(int biaya) {
        this.biaya = biaya;
    }

    public int hitungBiayaSopir (int hari) {
        return biaya * hari;
    }
}

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

    public Mobil() {
    }

    public String getMerk() {
        return merk;
    }

    public void setMerk(String merk) {
        this.merk = merk;
    }

    public int getBiaya() {
        return biaya;
    }

    public void setBiaya(int biaya) {
        this.biaya = biaya;
    }

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

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.setNama ("Jane Doe");
        p.setMobil (m);
        p.setSopir (s);
        p.setHari (2);
        System.out.println ("Biaya Total = " +
            p.hitungBiayaTotal () );
    }
}

```

- **Hasil:**

```

Biaya Total = 1100000
BUILD SUCCESSFUL (total time: 0 seconds)

```

Pertanyaan:

1. Perhatikan class Pelanggan. Pada baris program manakah yang menunjukkan bahwa class Pelanggan memiliki relasi dengan class Mobil dan class Sopir ?
Jawab: pada saat mendeklarasikan atribut mobil dan sopir, yang mana itu mengambil dari object class mobil dan sopir
2. Perhatikan method hitungBiayaSopir pada class Sopir, serta method hitungBiayaMobil pada class Mobil. Mengapa menurut Anda method tersebut harus memiliki argument hari?
Jawab: karena perentalan mobil juga menghitung seberapa lama mobil itu disewa
3. Perhatikan kode dari class Pelanggan. Untuk apakah perintah mobil.hitungBiayaMobil(hari) dan sopir.hitungBiayaSopir(hari) ?
Jawab: untuk menghitung total biaya mobil dan sopir perharinya
4. Perhatikan class MainPercobaan2. Untuk apakah sintaks p.setMobil(m) dan p.setSopir(s)?

Jawab: digunakan untuk mengisi value dari atribut mobil dan sopir dari object yang telah dibuat dengan menggunakan setter

5. Perhatikan class MainPercobaan2. Untuk apakah proses p.hitungBiayaTotal() tersebut ?

Jawab: digunakan untuk menghitung total biaya sewa mobil yang harus ditanggung pelanggan

6. Perhatikan class MainPercobaan2, coba tambahkan pada baris terakhir dari method main dan amati perubahan saat di-run!

```
System.out.println(p.getMobil().getMerk());
```

Jadi untuk apakah sintaks p.getMobil().getMerk() yang ada di dalam method main tersebut?

```
Biaya Total = 1100000
Avanza
BUILD SUCCESSFUL (total time: 0 seconds)
```

Hasil:

Jawab: syntax tersebut digunakan untuk menampilkan merk dari sebuah mobil yang disewa oleh pelanggan

Percobaan 3

- **Code:**

```
public class KeretaApi {
    private String nama;
    private String kelas;
    private Pegawai masinis;
    private Pegawai asisten;

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

    public KeretaApi(String nama, String kelas, Pegawai masinis, Pegawai asisten) {
        this.nama = nama;
        this.kelas = kelas;
        this.masinis = masinis;
        this.asisten = asisten;
    }

    public String getName() {
        return nama;
    }

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

    public String getKelas() {
        return kelas;
    }

    public void setKelas(String kelas) {
        this.kelas = kelas;
    }

    public Pegawai getMasinis() {
        return masinis;
    }

    public void setMasinis(Pegawai masinis) {
        this.masinis = masinis;
    }

    public Pegawai getAsisten() {
        return asisten;
    }

    public void setAsisten(Pegawai asisten) {
        this.asisten = asisten;
    }

    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;
    }
}

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

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

    public String getNip() {
        return nip;
    }

    public void setNip(String nip) {
        this.nip = nip;
    }

    public String getName() {
        return nama;
    }

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

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

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());
    }
}
```

- Hasil:

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

Asisten: Nip: 4567
Nama: Patrick Star

BUILD SUCCESSFUL (total time: 0 seconds)
```

Pertanyaan:

1. Di dalam method info() pada class KeretaApi, baris this.masinis.info() dan this.asisten.info() digunakan untuk apa ?

Jawab: digunakan untuk menampilkan data object class pegawai yang tersimpan pada atribut object masinis dan asisten

2. Buatlah main program baru dengan nama class MainPertanyaan pada package yang sama. Tambahkan kode berikut pada method main() !

```
Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");
KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis);
System.out.println(keretaApi.info());
```

```
public class MainPertanyaan {
    public static void main(String[] args) {
        Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");
        KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis);
        System.out.println(keretaApi.info());
    }
}
```

3. Apa hasil output dari main program tersebut ? Mengapa hal tersebut dapat terjadi ?

Jawab: terdapat error yang mengarah ke object keretaApi, yang mana pada saat memanggil fungsi info() pada class KeretaApi output menjadi NullPointerException karena pada class KeretaApi terkhusus fungsi info() menampilkan atribut asisten juga sebagai output, sedangkan atribut asisten tidak terisi

4. Perbaiki class KeretaApi sehingga program dapat berjalan !

Code:

```
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;
}
```

Hasil:

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

Percobaan 4

- Code:

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

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

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

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

```
public String getKode () {
    return kode;
}

public void setKode(String kode) {
    this.kode = kode;
}

public Kursi[] getArrayKursi () {
    return arrayKursi;
}

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

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

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

    public String getKtp() {
        return ktp;
    }

    public void setKtp(String ktp) {
        this.ktp = ktp;
    }

    public String getNama() {
        return nama;
    }

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

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

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

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

    public String getKtp() {
        return ktp;
    }

    public void setKtp(String ktp) {
        this.ktp = ktp;
    }

    public String getNama() {
        return nama;
    }

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

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

```
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()) ;
    }
}
```

- 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

```

Pertanyaan:

1. Pada main program dalam class MainPercobaan4, berapakah jumlah kursi dalam Gerbong A ?

Jawab: jumlah kursi pada Gerbong A adalah 10

2. Perhatikan potongan kode pada method info() dalam class Kursi. Apa maksud kode tersebut ?

```

...
if (this.penumpang != null) {
    info += "Penumpang: " + penumpang.info() + "\n";
}
...

```

Jawab: menunjukkan adanya kondisi jika value dari atribut penumpang tidak sama dengan null maka value dari penumpang akan ditampilkan

3. Mengapa pada method setPenumpang() dalam class Gerbong, nilai nomor dikurangi dengan angka 1 ?

Jawab: karena array dimulai dari 0 dan jika memilih index dari value atribut maka perlu dikurangi 1 karna jika pada value atribut angka dimulai dari 1

4. Instansiasi objek baru budi dengan tipe Penumpang, kemudian masukkan objek baru tersebut pada gerbong dengan gerbong.setPenumpang(budi, 1). Apakah yang terjadi ?

Jawab: penumpang pada kursi nomor 1 tergantikan oleh budi karena value pada kursi nomor 1 ditumpuk oleh budi

5. Modifikasi program sehingga tidak diperkenankan untuk menduduki kursi yang sudah ada penumpang lain !

• Code:

```

public void setPenumpang (Penumpang penumpang, int nomor) {
    if (nomor < 1 || nomor > arrayKursi.length) {
        System.out.println("Nomor kursi tidak valid.");
    } else if (this.arrayKursi[nomor - 1].getPenumpang() != null) {
        System.out.println("Penumpang " + penumpang.getNama() + "Kursi nomor " + nomor + " sudah ditempati oleh penumpang lain.");
    } else {
        this.arrayKursi[nomor - 1].setPenumpang(penumpang);
        System.out.println("Penumpang " + penumpang.getNama() + " berhasil ditempatkan di kursi nomor " + nomor);
    }
}

```

• Hasil:

```

Penumpang Mr. Krab berhasil ditempatkan di kursi nomor 1
Penumpang Budi arieKursi nomor 1 sudah ditempati oleh penumpang lain.
Kode: A
Nomor: 1
Penumpang: Ktp: 12345
Nama: Mr. Krab

```