

OBJECT ORIENTED PROGRAMMING JOBSHEET 4 – RELASI KELAS



HAWA ESANDA
2241720079_11_TI-2I

Percobaan 1

```
package jti.polinema.relasiclass.percobaan1;

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 info(){
        System.out.println("Merk Laptop = "+merk);
        proc.info();
    }
    public String getMerk() {
        return merk;
    }
    public void setMerk(String merk) {
        this.merk = merk;
    }
    public Processor getProc() {
        return proc;
    }
    public void setProc(Processor proc) {
        this.proc = proc;
    }
}
```

```
package jti.polinema.relasiclass.percobaan1;

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);
    }
}
```

```

package jti.polinema.relasticlass.percobaan1;

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();
    }
}

```

```

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 : Method *setter* dan *getter* digunakan untuk mengakses dan memodifikasi nilai atribut (variabel) dalam suatu class.

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 memberikan nilai default dan memungkinkan pembuatan objek tanpa memberikan nilai awal.
- Konstruktor berparameter memungkinkan inisialisasi objek dengan nilai yang disediakan oleh pengguna.

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

Jawab : Atribut yang bertipe objek dalam class **Laptop** adalah **proc**. **proc** adalah objek dari class **Processor**. Dalam hal ini, **Processor** adalah suatu kelas yang digunakan sebagai tipe data untuk atribut **proc**. Jadi, **proc** adalah objek dari kelas **Processor**.

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

Jawab : `private Processor proc;`

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

Jawab : Sintaks **proc.info()** pada class **Laptop** digunakan untuk memanggil method **info()** yang ada pada objek **proc**, yang merupakan objek dari class **Processor**.

6. Pada *class* MainPercobaan1, terdapat baris kode:
`Laptop l = new Laptop("Thinkpad", p);`
Apakah `p` tersebut ?

Jawab : `p` merupakan 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 : Hasil yang dikeluarkan tetap sama tidak ada yang berubah

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

Percobaan 2

```
package jti.polinema.relasticlass.percobaan2;

public class Mobil {
    private String merk;
    private int biaya;
    public Mobil() {
    }

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

    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;
    }
}
```

```
package jti.polinema.relasticlass.percobaan2;

public class Sopir {
    private String nama;
    private int biaya;
    public Sopir() {
    }

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

    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;
    }
}
```

```
package jti.polinema.relasticlass.percobaan2;

public class Pelanggan {
    private String nama;
    private Mobil mobil;
    private Sopir sopir;
    private int hari;
    public String getName() {
        return nama;
    }
    public void setName(String nama) {
        this.nama = nama;
    }
    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 getHari() {
        return hari;
    }
    public void setHari(int hari) {
        this.hari = hari;
    }
    public Pelanggan() {
    }
    public int hitungBiayaTotal(){
        return mobil.hitungBiayaMobil(hari) + sopir.hitungBiayaSopir(hari);
    }
}
```



```

package jti.polinema.relasiclass.percobaan2;

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

```

Biaya Total = 1100000

Pertanyaan

1. Perhatikan *class* Pelanggan. Pada baris program manakah yang menunjukkan bahwa *class* Pelanggan memiliki relasi dengan *class* Mobil dan *class* Sopir ?

Jawab : Class **Pelanggan** memiliki relasi dengan class **Mobil** dan **Sopir**. Atribut **mobil** dan **sopir** merupakan objek yang digunakan dalam class **Pelanggan**, menunjukkan adanya hubungan antara objek-objek dari kelas tersebut.

2. Perhatikan *method* hitungBiayaSopir pada class Sopir, serta method hitungBiayaMobil pada class Mobil. Mengapa menurut Anda *method* tersebut harus memiliki argument hari ?

Jawab : Karena biaya yang dikenakan untuk sopir dan mobil mungkin berubah berdasarkan jumlah hari pemakaian.

3. Perhatikan kode dari *class* Pelanggan. Untuk apakah perintah mobil.hitungBiayaMobil(hari) dan sopir.hitungBiayaSopir(hari) ?

Jawab : Perintah **mobil.hitungBiayaMobil(hari)** dan **sopir.hitungBiayaSopir(hari)** dalam method **hitungBiayaTotal()** pada class **Pelanggan** digunakan untuk menghitung biaya total sewa mobil dan sopir berdasarkan jumlah hari (**hari**).

4. Perhatikan *class* MainPercobaan2. Untuk apakah sintaks p.setMobil(m) dan p.setSopir(s) ?

Jawab : Sintaks **p.setMobil(m)** dan **p.setSopir(s)** dalam class **MainPercobaan2** digunakan untuk mengatur objek mobil (**m**) dan sopir (**s**) untuk objek pelanggan (**p**). Mereka menetapkan mobil dan sopir yang akan digunakan oleh pelanggan dalam suatu transaksi sewa.

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

Jawab : Proses **p.hitungBiayaTotal()** dalam class **MainPercobaan2** bertujuan untuk menghitung biaya total sewa yang harus dibayar oleh pelanggan (**p**)

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?

Jawab : Sintaks **p.getMobil().getMerk()** yang ada di dalam method **main** digunakan untuk mengakses dan mencetak merk mobil yang dimiliki oleh objek pelanggan (**p**).

Percobaan 3

```
package jti.polinema.relasiclass.percobaan3;

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

```

package jti.polinema.relasticlass.percobaan3;

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 getNama() {
        return nama;
    }
    public void setNama(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;
    }
}

```

```

package jti.polinema.relasticlass.percobaan3;

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());
    }
}

```

```

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

Asisten: Nip: 4567
Nama: Patrick Star

```

Pertanyaan

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

Jawab : Digunakan untuk mendapatkan informasi detail tentang masinis dan asisten yang bekerja pada kereta api tersebut.

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());

```

Jawab :

```

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

Asisten: null

```

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

Jawab : Terjadi error **NullPointerException** menunjukkan bahwa ada upaya untuk memanggil metode **info()** pada objek yang memiliki nilai null, yaitu **this.asisten**. Dalam kasus ini, metode **info()** pada class **KeretaApi** mencoba memanggil **this.asisten.info()**, tetapi karena asisten belum diinisialisasi, maka terjadi **NullPointerException**.

```
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "jti.polinema.relasiclass.percobaan3.Pegawai.info()" because "this.asisten" is null
    at jti.polinema.relasiclass.percobaan3.KeretaApi.info(KeretaApi.java:48)
    at jti.polinema.relasiclass.percobaan3.mainPertanyaan.main(mainPertanyaan.java:7)
```

4. Perbaiki *class* **KeretaApi** sehingga program dapat berjalan !

Jawab :

```
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";

    // mainPertanyaan : Periksa apakah asisten sudah diinisialisasi sebelum memanggil metode info()
    if (this.asisten != null) {
        info += "Asisten: " + this.asisten.info() + "\n";
    } else {
        info += "Asisten: null\n";
    }
    return info;
}
```

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

Percobaan 4

```
package jti.polinema.relasticlass.percobaan4;

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

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

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

    public String getKtp() {
        return ktp;
    }

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

    public String getName() {
        return nama;
    }

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

```
package jti.polinema.relasticlass.percobaan4;

public class Kursi {
    private String nomor;
    private Penumpang penumpang;

    public Kursi(String nomor){
        this.nomor = nomor;
    }
    public void setNomor(String nomor){
        this.nomor = nomor;
    }
    public String getNomor(){
        return nomor;
    }
    public void setPenumpang(Penumpang penumpang){
        this.penumpang = penumpang;
    }
    public Penumpang getPenumpang(){
        return penumpang;
    }
    public String info(){
        String info = "";
        info += "Nomor: " + nomor + "\n";
        if(this.penumpang != null){
            info += "Penumpang: " +penumpang.info()+ "\n";
        }
        return info;
    }
}
```



```
package jti.polinema.relasticlass.percobaan4;

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

    private void initkursi(){
        for (int i = 0; i < arrayKursi.length; i++){
            this.arrayKursi[i] = new Kursi(String.valueOf(i + 1));
        }
    }
    public Gerbong(String kode, int jumlah){
        this.kode = kode;
        this.arrayKursi = new Kursi[jumlah];
        this.initkursi();
    }
    public void setKode(String kode){
        this.kode = kode;
    }
    public String getKode(){
        return kode;
    }
    public void setPenumpang(Penumpang penumpang, int nomor){
        this.arrayKursi[nomor - 1].setPenumpang(penumpang);
    }
    public Kursi[] getArrayKursi(){
        return arrayKursi;
    }
    public String info(){
        String info = "";
        info += "Kode: " +kode+ "\n";
        for(Kursi kursi : arrayKursi){
            info += kursi.info();
        }
        return info;
    }
}
```

```

package jti.polinema.relasticlass.percobaan4;

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());
    }
}

```

```

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 : Gerbong memiliki jumlah kursi 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 : Kode tersebut digunakan untuk menambahkan informasi tentang penumpang ke dalam string **info** hanya jika objek **penumpang** tidak **null**.

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

Jawab : Karena indeks array atau list dimulai dari 0, sehingga nomor kursi yang dimasukkan oleh pengguna yang dimulai dari 1 harus disesuaikan dengan indeks array yang dimulai dari 0.

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

Jawab : Instansiasi objek baru **budi** hasilnya tergantung pada implementasi method **setPenumpang()**. Jika kursi nomor 1 belum diisi maka budi ditempatkan dikursi nomor 1 dan jika kursi nomor 1 sudah diisi maka hasilnya akan tetap diisi oleh penumpang pertama yaitu **Mr. Krab**.

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

Jawab :

```
public void setPenumpang(Penumpang penumpang, int nomor) {  
    if (nomor >= 1 && nomor <= arrayKursi.length) {  
        if (arrayKursi[nomor - 1].getPenumpang() == null) {  
            arrayKursi[nomor - 1].setPenumpang(penumpang);  
        } else {  
            System.out.println("Kursi nomor " + nomor + " sudah terisi.");  
        }  
    } else {  
        System.out.println(x:"Nomor kursi tidak valid.");  
    }  
}
```

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