

OBJECT ORIENTED PROGRAM PRACTICUM

JOBSHEET 4 : Relation Class



BY :

ATHRIYA GENFERIN

D4 INFORMATICS ENGINEERING (2I)

2241720075

(03)

State Polytechnic of Malang

Soekarno Hatta street No.9, Malang, East Java 65141

2022/2023



Percobaan 1

```
package jobsheet4.PBO.relationclass.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 void setMerk(String merk) {  
        this.merk = merk;  
    }  
  
    public void setCache(double cache) {  
        this.cache = cache;  
    }  
  
    public String getMerk() {  
        return merk;  
    }  
  
    public double getCache() {  
        return cache;  
    }  
  
    public void info() {  
        System.out.printf("Merk Processor = %s\n", merk);  
        System.out.printf("Cache Memory = %.2f\n", cache);  
    }  
}
```

```
package jobsheet4.PBO.relationclass.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();
    }
}

package jobsheet4.PBO.relationclass.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.printf("Merk Laptop = %s\n", merk);
        proc.info();
    }

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

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

    public String getMerk() {
        return merk;
    }

    public Processor getProc() {
        return proc;
    }
}
```



```
PS E:\SEMESTER 3\OOP\OOP> & 'C:\Program Files\Java\jdk1.8.0_121\bin\java.exe' -Xmx6e70\redhat.java\jdt_ws\OOP_71cf1347\bin 'jobsheet4.PBO.relationc
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 3.00
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 4.00
PS E:\SEMESTER 3\OOP\OOP>
```

Pertanyaan

Berdasarkan percobaan 1, jawablah pertanyaan-pertanyaan yang terkait:

1. Di dalam class **Processor** dan class **Laptop** , terdapat method setter dan getter untuk masing-masing atributnya. Apakah gunanya method setter dan getter tersebut ?

Method setter dan getter memiliki fungsi berikut dalam class **Processor** dan class **Laptop**:

Setter (setMerk, setCache):

- Method setter digunakan untuk mengatur atau mengubah nilai dari atribut (variabel) objek.
- Dalam class **Processor**, setMerk digunakan untuk mengatur nilai atribut merk, dan setCache digunakan untuk mengatur nilai atribut cache.
- Dalam class **Laptop**, setMerk digunakan untuk mengatur nilai atribut merk laptop, dan setProc digunakan untuk mengatur nilai atribut processor laptop.
- Setter memungkinkan pengguna objek untuk mengubah nilai atribut dengan kontrol tertentu, seperti validasi atau logika bisnis, sebelum nilai atribut diubah.

Getter (getMerk, getCache):

- Method getter digunakan untuk mengambil nilai dari atribut (variabel) objek.
- Dalam class **Processor**, getMerk digunakan untuk mengambil nilai atribut merk, dan getCache digunakan untuk mengambil nilai atribut cache.
- Dalam class **Laptop**, getMerk digunakan untuk mengambil nilai atribut merk laptop, dan getProc digunakan untuk mengambil objek processor yang terkait dengan laptop.
- Getter memungkinkan pengguna objek untuk mengakses nilai atribut tanpa harus mengaksesnya langsung, sehingga data atribut dapat dienkapsulasi dan diatur hak aksesnya.

2. Di dalam class **Processor** dan class **Laptop**, masing-masing terdapat konstruktor default dan konstruktor berparameter. Bagaimanakah beda penggunaan dari kedua jenis konstruktor tersebut ?

Konstruktor default adalah konstruktor yang tidak memiliki parameter atau argumen saat objek dibuat. Ini berarti objek akan diinisialisasi dengan nilai default untuk setiap atributnya. Pada contoh di atas, **Processor** memiliki konstruktor default yang tidak menginisialisasi nilai merk dan cache, sedangkan **Laptop** juga memiliki konstruktor default yang tidak menginisialisasi nilai merk dan processor.



Konstruktor berparameter adalah konstruktor yang menerima satu atau lebih parameter saat objek dibuat. Parameter-parameter ini digunakan untuk menginisialisasi atribut-atribut objek dengan nilai yang diberikan. Pada contoh di atas, Processor memiliki konstruktor berparameter yang menerima merk dan cache sebagai argumen, dan Laptop juga memiliki konstruktor berparameter yang menerima merk dan objek Processor sebagai argumen. Dengan konstruktor berparameter, Anda dapat mengatur nilai awal objek sesuai dengan kebutuhan saat objek dibuat.

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

Pada class Laptop, atribut yang bertipe objek adalah proc. Atribut proc merupakan objek dari class Processor. Dengan atribut proc, Laptop dapat memiliki informasi tentang processor yang digunakan oleh laptop tersebut.

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

private Processor proc;

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

Sintaks proc.info() dalam class Laptop digunakan untuk memanggil metode info() dari objek Processor yang disimpan dalam atribut proc dalam class Laptop. Sehingga dapat mendapatkan informasi tentang processor yang digunakan dalam laptop tersebut.

6. Pada class MainPercobaan1, terdapat baris kode:

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

Apakah p tersebut ?

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 ?

Jika diubah baris kode tersebut menjadi Laptop l = new Laptop("Thinkpad", new Processor("Intel i5", 3));, maka akan langsung membuat objek Processor baru sebagai argumen konstruktor Laptop. Ini akan menggantikan penggunaan objek p yang telah dibuat sebelumnya. Dan akan menghasilkan output yang sama, yang menampilkan informasi tentang laptop dan processor yang digunakan dalam laptop tersebut. Jadi, dalam hal ini, tidak ada perubahan yang signifikan dalam hasil program



Percobaan 2

```
package jobsheet4.PBO.relationclass.Percobaan2;  
  
public class Mobil {  
    private String merk;  
    private int biaya;  
  
    public Mobil() {  
    }  
  
    public Mobil(String merk, int biaya) {  
        this.merk = merk;  
        this.biaya = biaya;  
    }  
  
    public void setMerk(String merk) {  
        this.merk = merk;  
    }  
  
    public void setBiaya(int biaya) {  
        this.biaya = biaya;  
    }  
  
    public String getMerk() {  
        return merk;  
    }  
  
    public int getBiaya() {  
        return biaya;  
    }  
  
    public int hitungBiayaMobil(int hari) {  
        return biaya * hari;  
    }  
}
```

```
package jobsheet4.PBO.relationclass.Percobaan2;  
  
public class Pelanggan {  
    private String nama;  
    private Mobil mobil;  
    private Sopir sopir;  
    private int hari;
```



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



```
package jobsheet4.PBO.relationclass.Percobaan2;  
  
public class Sopir {  
    private String nama;  
    private int biaya;  
  
    public Sopir() {  
    }  
  
    public Sopir(String nama, int biaya) {  
        this.nama = nama;  
        this.biaya = biaya;  
    }  
  
    public void setName(String nama) {  
        this.nama = nama;  
    }  
  
    public void setBiaya(int biaya) {  
        this.biaya = biaya;  
    }  
  
    public String getName() {  
        return nama;  
    }  
  
    public int getBiaya() {  
        return biaya;  
    }  
  
    public int hitungBiayaSopir(int hari) {  
        return biaya * hari;  
    }  
}
```

```
package jobsheet4.PBO.relationclass.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("Jane Doe");  
p.setMobil(m);  
p.setSopir(s);  
p.setHari(2);  
System.out.println("Biaya Total = " + p.hitungBiayaTotal());  
}  
}
```

```
\bin\java.exe' '-cp' 'C:\Users\WINDOWS 11\AppData  
a\Roaming\Code\User\workspaceStorage\22ef0bb066b  
2e04bb7720e106b736e70\redhat.java\jdt_ws\OOP_71c  
f1347\bin' 'jobsheet4.PBO.relationclass.Percobaan  
n2.MainPercobaan2'  
Biaya Total = 1100000  
PS E:\SEMESTER 3\OOP\OOP>  
Ln 20, Col 2  Spaces: 4  UTF-8  CRLF  {} Java  Go Live
```

Pertanyaan

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

Class Pelanggan menunjukkan relasinya dengan class Mobil dan class Sopir pada baris berikut:

```
private Mobil mobil; (Menggunakan objek Mobil sebagai atribut)  
private Sopir sopir; (Menggunakan objek Sopir sebagai atribut)
```

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

Method hitungBiayaSopir pada class Sopir dan method hitungBiayaMobil pada class Mobil memerlukan argumen hari karena biaya sopir dan biaya mobil mungkin berubah berdasarkan jumlah hari penyewaan. Dengan adanya argumen hari, Anda dapat menghitung biaya yang sesuai dengan jumlah hari yang ditentukan oleh pelanggan.

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

Dalam class Pelanggan, perintah mobil.hitungBiayaMobil(hari) dan sopir.hitungBiayaSopir(hari) digunakan untuk menghitung biaya total dari penyewaan mobil dan sopir. mobil.hitungBiayaMobil(hari) akan mengembalikan biaya mobil berdasarkan jumlah hari, dan sopir.hitungBiayaSopir(hari) akan mengembalikan biaya sopir berdasarkan jumlah hari.



4. Perhatikan class **MainPercobaan2**. Untuk apakah sintaks **p.setMobil(m)** dan **p.setSopir(s)** ?
Sintaks **p.setMobil(m)** dan **p.setSopir(s)** digunakan untuk mengatur objek Mobil dan Sopir dalam objek Pelanggan. Dengan demikian, Pelanggan memiliki informasi tentang mobil dan sopir yang digunakan selama penyewaan.
5. Perhatikan class **MainPercobaan2**. Untuk apakah proses **p.hitungBiayaTotal()** tersebut ?
Proses **p.hitungBiayaTotal()** digunakan untuk menghitung biaya total dari penyewaan mobil dan sopir oleh pelanggan. Hasil dari method ini adalah biaya total yang harus dibayar oleh 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?

Sintaks **p.getMobil().getMerk()** pada baris terakhir dari method main digunakan untuk mengakses atribut merk dari objek Mobil yang dimiliki oleh objek Pelanggan **p**. Dengan demikian, Anda dapat mendapatkan informasi tentang merk mobil yang disewa oleh pelanggan **p**. Dalam contoh ini, hasilnya akan mencetak merk mobil yang diset sebelumnya, yaitu "Avanza".

Percobaan 3

```
package jobsheet4.PBO.relationclass.Percobaan3;  
  
public class KeretaApi {  
    private String nama;  
    private String kelas;  
    private Pegawai masinis;  
    private Pegawai asisten;  
  
    public KeretaApi() {  
    }  
  
    public KeretaApi(String nama, String kelas, Pegawai masinis, Pegawai  
asisten) {  
        this.nama = nama;  
        this.kelas = kelas;  
        this.masinis = masinis;  
        this.asisten = asisten;  
    }  
  
    public void setNama(String nama) {  
        this.nama = nama;  
    }  
}
```



```
public void setKelas(String kelas) {  
    this.kelas = kelas;  
}  
  
public void setMasinis(Pegawai masinis) {  
    this.masinis = masinis;  
}  
  
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 jobsheet4.PBO.relationclass.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());  
    }  
}
```

```
package jobsheet4.PBO.relationclass.Percobaan3;  
  
public class Pegawai {  
    private String nip;  
    private String nama;  
  
    public Pegawai() {  
    }  
  
    public Pegawai(String nip, String nama) {  
        this.nip = nip;  
    }  
}
```



```
        this.nama = nama;
    }

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

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

    public String getNip() {
        return nip;
    }

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

```
2e04bb7720e106b736e70\redhat.java\jdt_ws\OOP_71c
f1347\bin' 'jobsheet4.PBO.relationclass.Percobaan
n3.MainPercobaan3'
Nama: Gaya Baru
Kelas: Bisnis
Masinis: Nip: 1234
Nama: Spongebob Squarepants

Asisten: Nip: 4567
Nama: Patrick Star

PS E:\SEMESTER 3\OOP\OOP> 
```

Pertanyaan

1. Di dalam method info() pada class KeretaApi, baris this.masinis.info() dan this.asisten.info() digunakan untuk apa ?
Di dalam method info() pada class KeretaApi, baris this.masinis.info() dan this.asisten.info() digunakan untuk mendapatkan informasi tentang masinis dan asisten kereta dengan memanggil method info() dari objek Pegawai yang mewakili 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



ATHRIYA GENFERIN
2I – 03 - 2241720075
D-IV INFORMATICS ENGINEERING

```
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 ?

Nama: Gaya Baru

Kelas: Bisnis

Masinis: Nip: 1234

Nama: Spongebob Squarepants

Asisten: Nip: null

Nama: null

4. Perbaiki class KeretaApi sehingga program dapat berjalan !

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

Percobaan 4

```
package jobsheet4.PBO.relationclass.Percobaan4;  
  
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();  
    }  
  
    private void initKursi() {  
        for (int i = 0; i < arrayKursi.length; i++) {  
            this.arrayKursi[i] = new Kursi(String.valueOf(i + 1));  
        }  
    }  
}
```



```
    }  
}  
  
public void setKode(String kode) {  
    this.kode = kode;  
}  
  
public void setPenumpang(Penumpang penumpang, int nomor) {  
    if (this.arrayKursi[nomor - 1].getPenumpang() != null) {  
        System.out.println("Kursi sudah terisi, silahkan pilih kursi  
lain");  
    } else {  
        this.arrayKursi[nomor - 1].setPenumpang(penumpang);  
    }  
}  
  
public String getKode() {  
    return kode;  
}  
  
public Kursi[] getArrayKursi() {  
    return arrayKursi;  
}  
  
public String info() {  
    String info = "";  
    info += "Kode: " + this.kode + "\n";  
    for (Kursi kursi : arrayKursi) {  
        info += kursi.info();  
    }  
    return info;  
}  
}
```

```
package jobsheet4.PBO.relationclass.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 void setPenumpang(Penumpang penumpang) {
    this.penumpang = penumpang;
}

public String getNomor() {
    return nomor;
}

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 jobsheet4.PBO.relationclass.Percobaan4;

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

    public Penumpang() {
    }

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

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

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



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

```
package jobsheet4.PBO.relationclass.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());  
    }  
}
```

```
PS E:\SEMESTER 3\OOP\OOP> & 'C:\Program Files\Java\jdk1.8.0_121\bin\java.exe'  
'-cp' 'C:\Users\WINDOWS 11\AppData\Roaming\Code\User\workspaceStorage\22ef0bb  
066b2e04bb7720e106b736e70\redhat.java\jdt_ws\OOP_71cf1347\bin' 'jobsheet4.PBO.  
relationclass.Percobaan4.MainPercobaan4'  
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  
  
PS E:\SEMESTER 3\OOP\OOP>
```

Run: MainP...

Run: MainP...



Pertanyaan

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

Jumlah kursi dalam Gerbong A adalah 10. Hal ini ditentukan saat objek Gerbong "A" dibuat dengan `new Gerbong("A", 10);`.

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

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

Potongan kode pada method `info()` dalam class `Kursi` digunakan untuk mengecek apakah kursi sudah memiliki penumpang atau tidak. Jika penumpang (objek `Penumpang` yang ada di kursi) tidak null, artinya kursi sudah terisi oleh penumpang, maka informasi penumpang tersebut ditampilkan dalam hasil output.

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

Pada method `setPenumpang()` dalam class `Gerbong`, nilai nomor dikurangi 1 karena dalam pemrograman, indeks array dimulai dari 0. Dalam kode ini, kursi yang pertama memiliki nomor 1, tetapi array dalam objek `arrayKursi` dimulai dari indeks 0. Oleh karena itu, untuk menyesuaikan indeks array dengan nomor kursi, nomor dikurangi 1.

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

Pada instruksi `gerbong.setPenumpang(budi, 1);`, objek `budi` (tipe `Penumpang`) dimasukkan ke kursi pertama (indeks 0) dalam gerbong. Ini berarti bahwa kursi pertama sekarang ditempati oleh penumpang "Budi". Anda kemudian dapat memanggil `gerbong.info()` untuk melihat bahwa kursi pertama sudah ditempati oleh penumpang "Budi".

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

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



Tugas

Berdasarkan latihan di pertemuan teori, rancang dengan class diagram, kemudian implementasikan ke dalam program! Studi kasus harus mewakili relasi class dari percobaan-percobaan yang telah dilakukan pada materi ini, setidaknya melibatkan minimal 4 class (class yang berisi main tidak dihitung)

```
package jobsheet4.PBO.relationclass.tugas;

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

    public Mobil(String merk, int biaya) {
        this.merk = merk;
        this.biaya = biaya;
    }

    public String getMerk() {
        return merk;
    }

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

```
package jobsheet4.PBO.relationclass.tugas;

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

    public Pelanggan(String nama, int hari, Mobil mobil, Sopir sopir) {
        this.nama = nama;
        this.hari = hari;
        this.mobil = mobil;
        this.sopir = sopir;
    }

    public int hitungBiayaTotal() {
        int biayaMobil = mobil.hitungBiayaMobil(hari);
        int biayaSopir = sopir.hitungBiayaSopir(hari);
    }
}
```



```
        return biayaMobil + biayaSopir;
    }

    public String getNama() {
        return nama;
    }
}
```

```
package jobsheet4.PBO.relationclass.tugas;

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

    public Sopir(String nama, int biaya) {
        this.nama = nama;
        this.biaya = biaya;
    }

    public String getNama() {
        return nama;
    }

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

```
package jobsheet4.PBO.relationclass.tugas;

public class MainProgram {
    public static void main(String[] args) {
        Mobil mobil = new Mobil("Avanza", 350000);
        Sopir sopir = new Sopir("John Doe", 200000);
        Pelanggan pelanggan = new Pelanggan("Jane Doe", 2, mobil, sopir);

        System.out.println("Nama Pelanggan: " + pelanggan.getNama());
        System.out.println("Merk Mobil: " + mobil.getMerk());
        System.out.println("Biaya Total: " + pelanggan.hitungBiayaTotal());
    }
}
```



ATHRIYA GENFERIN
2I – 03 - 2241720075
D-IV INFORMATICS ENGINEERING

Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

```
PS E:\SEMESTER 3\OOP\OOP> & 'C:\Program Files\Java\jdk1.8.0_121\bin\java.exe' '-cp' 'C:\Users\WINDOWS 11\AppData\Roaming\Code\User\workspace\0bb066b2e04bb7720e106b736e70\redhat.java\jdt_ws\OOP_71cf1347\bin' 'jobsheet4.PBO.relationclass.tugas.MainProgram'
```

Nama Pelanggan: Jane Doe

Merk Mobil: Avanza

Biaya Total: 1100000

PS E:\SEMESTER 3\OOP\OOP>