



NAMA : SILVIA PRADA APRILIA
NIM : 2041720141
KELAS : TI 2C
MATERI : RELASI KELAS

Percobaan 1

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 *getter* digunakan untuk mendapatkan nilai dari atribut yang berada di class Laptop dan Processor. Sedangkan method *setter* digunakan untuk mengeset nilai atribut yang berada di class Laptop dan Processor.

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: Pada konstruktor default tidak terdapat parameter dan tidak berisi kode program apapun. Sedangkan pada konstruktor default terdapat parameter dan berisi inisialisasi nilai atribut. Konstruktor default digunakan untuk objek yang tidak memiliki Passing Parameter. Sedangkan konstruktor berparameter digunakan untuk objek yang memiliki passing parameter sehingga tidak perlu memanggil method setter untuk mengeset nilai awal.

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

Jawab: Atribut *proc* bertipe object yaitu objek class Processor.

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

Jawab: `private Processor_Silvia proc;` menandakan bahwa terdapat atribut yang bertipe objek dari class Processor.

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

Jawab: Sintaks `proc.info()` digunakan untuk mengakses method `info` pada class Processor yang berfungsi untuk menampilkan info pada class Processor.

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 ?

Jawab: *p* adalah objek pada class Processor yang berarti parameter tersebut menggunakan atribut objek dari class Processor. Tidak terjadi error saat kode program diubah dan Tidak terjadi perubahan setelah kode program dijalankan



NAMA : SILVIA PRADA APRILIA
NIM : 2041720141
KELAS : TI 2C
MATERI : RELASI KELAS

Percobaan 2

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

Jawab:

private Mobil_Silvia mobil;

private Sopir_Silvia sopir;

Karena atribut tersebut bertipe objek dari *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: Method tersebut memiliki argument hari karena didalam method tersebut dilakukan perhitungan biaya yang melibatkan hari yang diinputkan dari main.

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

Jawab: Kode tersebut digunakan untuk menghitung total biaya dari *class* Mobil dan *class* Sopir. Sehingga mejadi total biaya yang harus dibayar.

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

Jawab: Sintaks tersebut digunakan untuk insialisasi atribut sopir dan mobil pada *class* Pelanggan.

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

Jawab: Sintaks tersebut diguanakn untuk memanggil *method* hitungBiayaTotal pada *class* Pelanggan yang berfungsi untuk menghitung biaya total dari *class* Mobil dan Sopir.

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 tersebut digunakan untuk menampilkan Merk mobil dari *class* mobil yang dipanggil pada *class* Pelanggan.



NAMA : SILVIA PRADA APRILIA
NIM : 2041720141
KELAS : TI 2C
MATERI : RELASI KELAS

Percobaan 3

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

Jawab: Baris kode program tersebut digunakan untuk memanggil *method* `info` pada *class* `pegawai` yang digunakan untuk menampilkan data dari pegawai dengan posisi `masinis`.

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:

```
3 public class MainPertanyaan {  
4     public static void main(String[] args){  
5         Pegawai_Silvia masinis = new Pegawai_Silvia("1234", "Spongebob Squarepants");  
6         KeretaApi_Silvia keretaApi = new KeretaApi_Silvia("Gaya Baru", "Bisnis", masinis);  
7         System.out.println(keretaApi.info());  
8     }  
9 }
```

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

Jawab: Hal ini terjadi karena tidak ada *passing* parameter untuk *asisten* sehingga perlu ditambahkan objek *asisten* untuk menjalankan program.

Output - Jobsheet4 (run) x

```
run:  
Exception in thread "main" java.lang.RuntimeException: Uncompilable source code - con  
required: java.lang.String,java.lang.String,silvia.relasiclass.percobaan3.Pegawai_S  
found: java.lang.String,java.lang.String,silvia.relasiclass.percobaan3.Pegawai_Silv  
reason: actual and formal argument lists differ in length  
at silvia.relasiclass.percobaan3.MainPertanyaan.main(MainPertanyaan.java:6)  
Java Result: 1  
BUILD SUCCESSFUL (total time: 3 seconds)
```

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

Jawab:

```
1 package silvia.relasiclass.percobaan3;  
2  
3 public class MainPertanyaan {  
4     public static void main(String[] args){  
5         Pegawai_Silvia masinis = new Pegawai_Silvia("1234", "Spongebob Squarepants");  
6         Pegawai_Silvia asisten = new Pegawai_Silvia("5678", "Shandy Chiks");  
7         KeretaApi_Silvia keretaApi = new KeretaApi_Silvia("Gaya Baru", "Bisnis", masinis, asisten);  
8         System.out.println(keretaApi.info());  
9     }  
10 }
```



NAMA : SILVIA PRADA APRILIA
NIM : 2041720141
KELAS : TI 2C
MATERI : RELASI KELAS

Output

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

Asisten: Nip: 5678
Nama: Shandy Chiks

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

Percobaan 4

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

Jawab: Kursi dalam gerbong A berjumlah 10 seperti yang di tuliskan dalam passing parameter objek gerbong `Gerbong gerbong = new Gerbong("A", 10);`

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

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

Jawab: Kode tersebut berfungsi untuk menjalankan perulangan untuk menampilkan data penumpang pada *class* `Penumpang` apabila penumpang tidak sama dengan `null`. Maka kode tersebut hanya akan dijalankan apabila terdapat penumpang saja.

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

Jawab: Karena penulisan index dalam array dimulai dari 0 sedangkan nomor yang dimasukkan tersebut bernilai mulai dari 1. Maka untuk menyesuaikan dengan array nilai nomor dikurangi 1 agar sesuai dengan tempat array index ke 0.



NAMA : SILVIA PRADA APRILIA
NIM : 2041720141
KELAS : TI 2C
MATERI : RELASI KELAS

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

Jawab: Kursi yang sebelumnya dimiliki oleh Mr. Krab menjadi milik Mr. Budi

```
public static void main(String[] args){  
    Penumpang_Silvia p = new Penumpang_Silvia("12345", "Mr. Krab");  
    Gerbong_Silvia gerbong = new Gerbong_Silvia("A", 10);  
    gerbong.setPenumpang(p, 1);  
    System.out.println(gerbong.info());  
  
    Penumpang_Silvia budi = new Penumpang_Silvia("678910", "Mr. Budi");  
    gerbong.setPenumpang(budi, 1);  
    System.out.println(gerbong.info());  
}
```

Output

```
run:  
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  
  
Kode: A  
Nomor: 1  
Penumpang: Ktp: 678910  
Nama: Mr. Budi  
  
Nomor: 2  
Nomor: 3  
Nomor: 4  
Nomor: 5  
Nomor: 6  
Nomor: 7  
Nomor: 8  
Nomor: 9  
Nomor: 10  
  
BUILD SUCCESSFUL (total time: 0 seconds)
```



NAMA : SILVIA PRADA APRILIA
NIM : 2041720141
KELAS : TI 2C
MATERI : RELASI KELAS

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

Jawab:

Modifikasi

```
public void setPenumpang(Penumpang_Silvia penumpang, int nomor) {  
    if (this.arrayKursi[nomor - 1].getPenumpang() != null) {  
        System.out.println("Kursi sudah ditempati\n");  
    } else {  
        this.arrayKursi[nomor - 1].setPenumpang(penumpang);  
    }  
}
```

Output

```
Output - Jobsheet4 (run) x Penumpang_Silvia.java x  
run:  
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  
  
Kursi sudah ditempati  
  
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  
  
BUILD SUCCESSFUL (total time: 0 seconds)
```

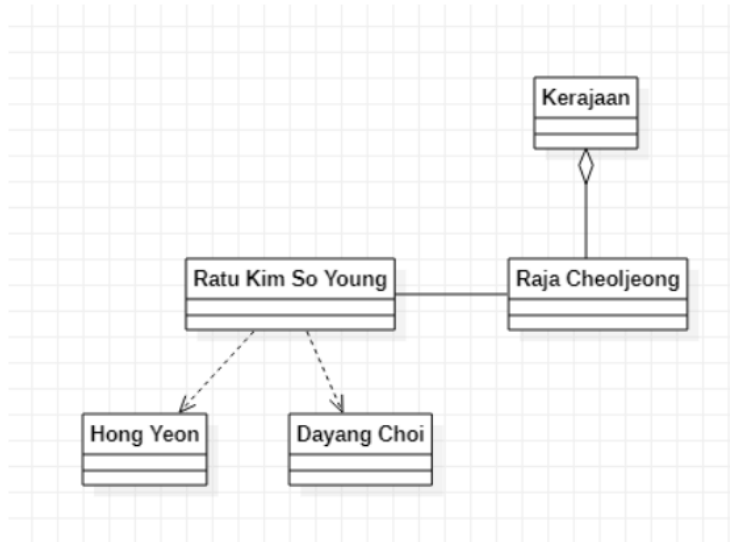


NAMA : SILVIA PRADA APRILIA
NIM : 2041720141
KELAS : TI 2C
MATERI : RELASI KELAS

Tugas Praktikum

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)

Jawab:



Kode Program sudah tercantum