



NAMA : OKTAVIAN VANDI TRI SHAKTI

NIM : 2041720208

KELAS : 2C

Pertanyaan 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:

- Getter adalah public method dan memiliki tipe data return, yang berfungsi untuk mendapatkan nilai dari atribut private.
- Setter adalah public method yang tidak memiliki tipe data return, yang berfungsi untuk memanipulasi nilai dari atribut private.

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:

- Default konstruktor adalah jenis konstruktor yang disebut secara otomatis ketika programmer belum mendefinisikan konstruktor dalam program.
- Sebaliknya, konstruktor berparameter adalah jenis konstruktor yang ditentukan oleh pemrogram dengan melewati parameter untuk memberikan nilai awal pada variabel instan di kelas. Itu adalah perbedaan utama antara konstruktor default dan parameter.

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

Jawab:

Atribut proc mempunyai tipe data object yaitu Processor.

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

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

Pada bagian tersebut terlihat jelas pada class Laptop telah dideklarasikan atribut proc yang memiliki tipe data object Processor dari class Processor.

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

Jawab:

Fungsinya untuk memanggil method info() dari 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:



NAMA : OKTAVIAN VANDI TRI SHAKTI

NIM : 2041720208

KELAS : 2C

P tersebut merupakan objek dari konstruktor Processor yang telah dibuat sebelumnya. Apabila diganti kode program hasilnya tetap sama (tidak ada perubahan):

Pertanyaan Percobaan 2

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

Jawab:

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

Pada bagian tersebut terlihat jelas pada class Pelanggan telah dideklarasikan atribut mobil dan sopir yang memiliki tipe data object Mobil dan Sopir 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:

Karena untuk menghitung biaya sopir diperlukan juga untuk menghitung berapa hari mereka bekerja lalu mengalikannya dengan biaya. Sehingga akan diperoleh hasil akhir dari berapa lama waktu mereka bekerja dengan biaya per harinya.

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

Jawab:

Untuk menghitung berapa banyak biaya yang harus pelanggan keluarkan. Dan biaya itu berasal dari total biaya mobil ditambah dengan total biaya sopir.

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

Jawab:

Untuk memanggil method setter (setMobil dan setSopir) dengan parameter masing-masing m dan p. Dimana m merupakan objek dari konstruktor Mobil dan s merupakan objek dari konstruktor Sopir.

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

Jawab:

Untuk memanggil method hitungBiayaTotal() dari class Pelanggan. Method tersebut digunakan untuk menghitung berapa banyak biaya (total biaya) yang harus pelanggan keluarkan.



NAMA :OKTAVIAN VANDI TRI SHAKTI

NIM : 2041720208

KELAS : 2C

- 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:

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

Sintaks tersebut berfungsi untuk mencetak merk mobil dengan cara memanggil method `getMobil()` dari class `Pelanggan`. Karena class `Pelanggan` dan `Mobil` terhubung, kita juga bisa langsung memanggil method `getMerk()`

Pertanyaan Percobaan 3

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

Jawab:

Kode program tersebut digunakan untuk memanggil method `info` dari class `Pegawai`. Hal ini bisa dilakukan karena atribut `masinis` dan `asisten` memiliki tipe data `Pegawai`.

- 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());`

```
package Oktavian.relasticlass.percobaan3;
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());
    }
}
```

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

Jawab:

Terjadi error

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ jti.polinema.relasticlass.percobaan1 ---
Exception in thread "main" java.lang.NullPointerException
    at Oktavian.relasticlass.percobaan3.KeretaApi.info(KeretaApi.java:58)
    at Oktavian.relasticlass.percobaan3.MainPertanyaan.main(MainPertanyaan.java:6)
Command execution failed.
```

Hal tersebut terjadi karena pada baris terakhir kita memanggil method `info()` dari class `KeretaApi`.

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

Di dalam method `info()`, terdapat kode program untuk memanggil atribut `asisten` juga. Sedangkan pada class `MainPertanyaan` yang kita buat, kita hanya memanggil atribut/objek `masinis`

- Perbaiki class `KeretaApi` sehingga program dapat berjalan!

Jawab:

Untuk memperbaiki errornya dengan menambahkan satu method yaitu `info2()` pada



NAMA :OKTAVIAN VANDI TRI SHAKTI

NIM : 2041720208

KELAS : 2C

class KeretaApi yang di dalamnya tidak dipanggil atribut asisten

```
public String info2() {  
    String info2 = "";  
    info2 += "Nama: " + this.nama + "\n";  
    info2 += "Kelas: " + this.kelas + "\n";  
    info2 += "Masinis: " + this.masinis.info() + "\n";  
    return info2;  
}
```

Pada class MainPertanyaan, kita juga ganti method yang dipanggil menjadi info2()

```
System.out.println(keretaApi.info2());
```

Output yang dihasilkan seperti berikut :

```
--- exec-maven-plugin:3.0.0:exec (def  
Nama: Gaya Baru  
Kelas: Bisnis  
Masinis: Nip: 1234  
Nama: Spongebob Squarepants
```

Pertanyaan Percobaan 4

1. Pada main program dalam class MainPercobaan4, berapakah jumlah kursi dalam Gerbong A ?
Jawab :
Pada gerbong A tidak ada jumlah kursi. hal ini dikarenakan pada class main hanya ada instansiasi dari gerbong C
2. Perhatikan potongan kode pada method info() dalam class Kursi. Apa maksud kode tersebut ?
Jawab :
Maksudnya adalah apabila atribut telah terisi nilai maka nilai akan ditampilkan
3. Mengapa pada method setPenumpang() dalam class Gerbong, nilai nomor dikurangi dengan angka 1 ?
Jawab :
Nilai nomor dikurangi dengan angka 1 karena array dimulai dari indeks ke-0
4. Instansiasi objek baru budi dengan tipe Penumpang, kemudian masukkan objek baru tersebut pada gerbong dengan gerbong.setPenumpang(andi, 5). Apakah yang terjadi ?
Jawab :

```
package Oktavian.relasiclass.percobaan4;  
public class MainPercobaan4 {  
    public static void main(String[] args) {  
        Penumpang andi = new Penumpang("731234984", "Bagus Raharjo");  
        Penumpang budi = new Penumpang("123456789", "Budianto");  
        Gerbong gerbong = new Gerbong("C", 20);  
        gerbong.setPenumpang(andi, 5);  
        gerbong.setPenumpang(budi, 5);  
        System.out.println(gerbong.info());  
    }  
}
```



NAMA :OKTAVIAN VANDI TRI SHAKTI
NIM : 2041720208
KELAS : 2C

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ jti.polinema.relasticlass.percobaan1 ---  
Kode: C  
Nomor: 1  
Nomor: 2  
Nomor: 3  
Nomor: 4  
Nomor: 5  
Penumpang: Ktp: 123456789  
Nama: Budianto  
  
Nomor: 6  
Nomor: 7  
Nomor: 8  
Nomor: 9  
Nomor: 10  
Nomor: 11  
Nomor: 12  
Nomor: 13  
Nomor: 14  
Nomor: 15  
Nomor: 16  
Nomor: 17  
Nomor: 18  
Nomor: 19  
Nomor: 20  
  
-----  
BUILD SUCCESS  
-----
```

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

Jawab :