**Tugas Praktikum Pertemuan 4 - Relasi Kelas**



**Muhammad Farrel Caesarian**

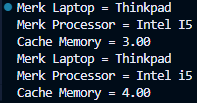
**TI-2D/16**

**2341720053**

**POLITEKNIK NEGERI MALANG**

Percobaan 1

Hasil :



**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 ?

Jawab: setter berfungsi untuk memberikan nilai pada attribut, sedangkan getter berfungsi untuk menampilkan attribut.

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 tidak menerima parameter, sehingga kita tidak harus memberikan nilai pada attribut saat membuat objek. Berbeda dengan konstruktor berparameter yang mengharuskan kita untuk menuliskan attribur saat membuat objek.

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

manakah yang bertipe *object* ?

Jawab: Attribut yang bertipe object adalah attribut proc, karena proc merupakan object dari kelas Processor yang menandakan bahwa proc bukanlah tipe data primitif seperti int atau String melainkan sebuah instance dari kelas lain.

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

Jawab:

private Processor proc;

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

Jawab: Untuk menampilkan informasi dari 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 object dari kelas Processor, jika kode diganti menjadi kode seperti diatas, outputnya sama dengan kode awal.

**Percobaan 2**

**Pertanyaan**

Hasil Percobaan:



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

Jawab:



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

Jawab:

Agar pethitungan biaya dapat diproses.

3. Perhatikan kode dari *class* Pelanggan. Untuk apakah perintah

mobil.hitungBiayaMobil(hari) dan sopir.hitungBiayaSopir(hari) ?

Jawab:

Untuk menghitung biaya total untuk pelanngan.

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

p.setSopir(s) ?

Jawab: Pada kelas MainPercobaan2, sintaks p.setMobil(m) dan p.setSopir(s) digunakan untuk menghubungkan objek Mobil dan objek Sopir dengan objek Pelanggan

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

tersebut ?

Jawab: menghitung biaya total untuk pelanggan.

6. Perhatikan class MainPercobaan2, coba tambahkan pada baris terakhir dari *method*

*main* dan amati perubahan saat di‑*run*!Page **6** of **10**

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

Jadi untuk apakah sintaks p.getMobil().getMerk() yang ada di dalam *method*

*main* tersebut?

Jawab:



Menampilkan merk mobil yang disewa pelanggan.

**Percobaan 3**

Hasil Percobaan



**Pertanyaan**

1. Di dalam *method* info() pada *class* KeretaApi, baris this.masinis.info() dan

this.asisten.info() digunakan untuk apa ?

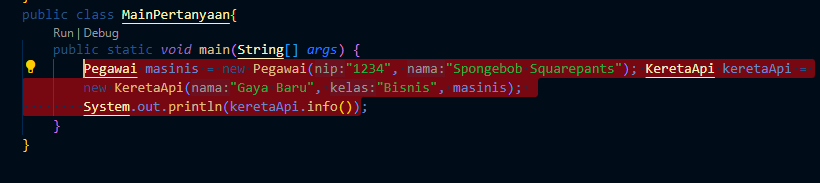
Jawab: Untuk menampilkan info tentang masinis dan asisten.

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());

Jawab:



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

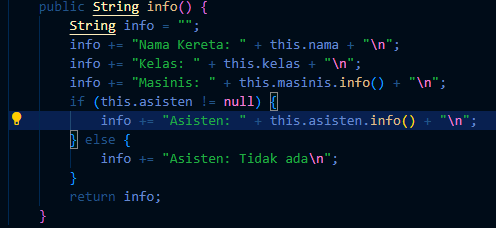
Jawab:

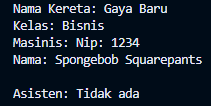


Hasilnya error karena

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

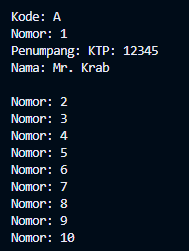
Jawab:





Percobaan 4

Hasil Percobaan



**Pertanyaan**

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

Gerbong A ?

Jawab: 10

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

...

**if** (**this**.penumpang != **null**) {

info += "Penumpang: " + penumpang.info() + "\n";

}

...

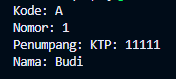
Jawab: saat ada penumpang, maka informasi penumpang akan ditambahkan.

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

Jawab: Ketika nomor kursi yang diberikan adalah 1 (kursi pertama), kita mengurangi 1 menjadi 0, sehingga kita mengakses elemen pertama dari arrayKursi. Jika nomor kursi yang diberikan adalah 2, kita mengakses elemen kedua, dan seterusnya.

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

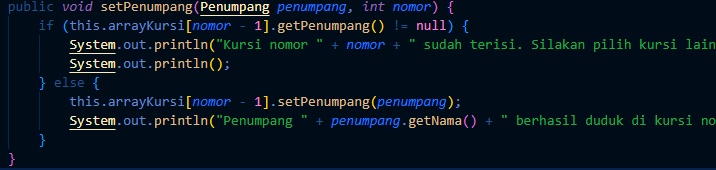
Jawab:

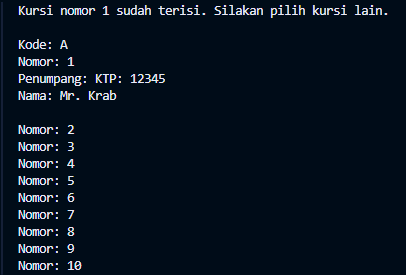


Mr. Krab akan terhapus dari list penumpang

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

Jawab:





**Tugas**

Class Diagram

