

Praktikum Pemrograman Berorientasi Objek

Modul 1 – Class, Object, dan UML

Case Study

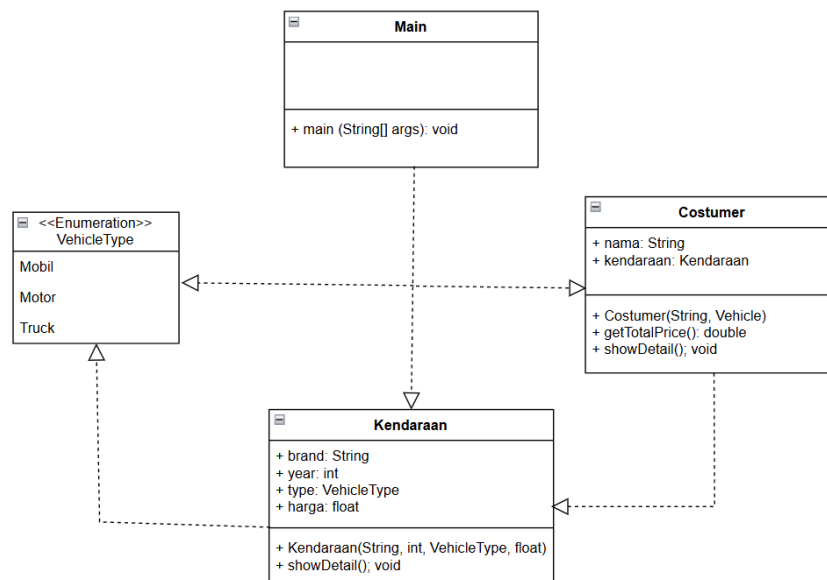
Catatan Penting

- ❖ Pastikan codingan kalian dapat di compile sebelum melakukan commit dan push ke remote repository.
- ❖ Pada UML, constructor dengan parameter kosong dapat diabaikan karena itu merupakan default constructor.
- ❖ Pada UML, method-method pada Enum dapat diabaikan karena itu merupakan default method dari Enum.

Buatlah *class* ataupun *enum* sesuai dengan yang sudah tertera di bawah, pada package yang baru dibuat.

Pastikan *access modifier*, *type*, dan modifier lainnya sesuai dengan UML. Jika bingung saat membaca UML dapat membuka **Standar dan Aturan Penulisan Program** dalam OOP di EMAS.

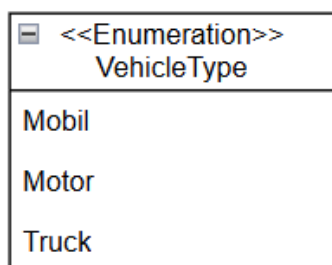
1. Buka *project* IntelliJ bernama “CS_[Hari]_[KodeAslab]_[Nama]_[NPM]_OOP[NoModul]” yang sudah Anda buat sebelumnya pada saat Pre-CS.
2. Berikut adalah UML yang akan digunakan pada case study ini:



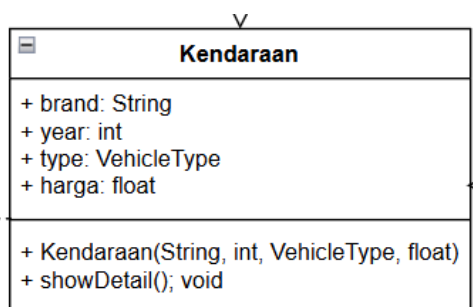
Anda bisa melihat link berikut untuk membaca UML yang diberikan:

<https://pages.cs.wisc.edu/~hasti/cs302/examples/UMLdiagram.html>

3. Buatlah *enumeration* bernama VehicleType dengan isi mengikuti UML berikut:



4. Buatlah *class* bernama Kendaraan dengan isi mengikuti UML berikut:

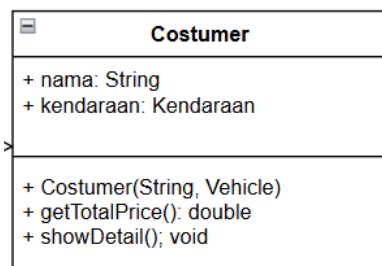


- *Constructor* Kendaraan berfungsi untuk mengisi seluruh variabel dari *class* Kendaraan, yaitu brand, year, type, dan harga dengan parameter yang diberikan di UML.

- *Method* `showDetail()` berfungsi untuk menampilkan beberapa informasi dari kendaraan, yaitu brand, year, type, dan harga. Berikut contoh *output*-nya:

```
Brand: Honda Supra  
Year: 1998  
Type: Motor  
Harga: 3000.0
```

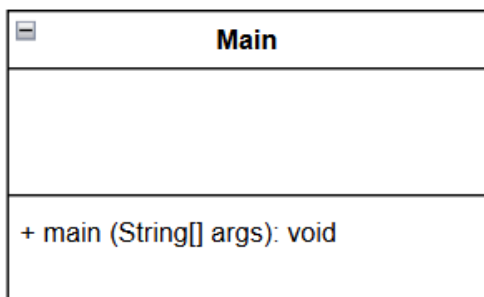
5. Buatlah *class* bernama *customer* dengan isi mengikuti UML berikut:



- *Constructor* *customer* berfungsi untuk mengisi seluruh variabel dari *class* *customer*, yaitu nama dan kendaraan dengan parameter yang diberikan di UML.
- *Method* `getTotalPrice` berfungsi untuk mengembalikan total harga kendaraan.
- *Method* `showDetail()` berfungsi untuk menampilkan informasi nama *customer* dan informasi kendaraan. Berikut contoh *output*-nya:

```
Customer Name: Bapak Joko  
Brand: Honda Supra  
Year: 1998  
Type: Motor  
Harga: 3000.0
```

6. Buatlah *class* bernama *Main* dengan isi mengikuti UML berikut:



7. Pada *class* Main, buatlah 3 *object* **Kendaraan** dengan ketentuan seperti berikut:

Nama Object	Brand	Year	Type	Harga
kendaraanSupraBapak	Honda Supra	1998	Motor	3000
kendaraanKalcer	VW Beetle	1998	Mobil	200000
kendaraanGuede	Isuzu Giga	2011	Truck	300000

Dan buatlah 3 *object* **customer** dengan ketentuan seperti berikut:

Nama Object	Nama customer	Kendaraan
customer1	Tentukan sendiri	kendaraanSupraBapak
customer2	Tentukan sendiri	kendaraanKalcer
customer3	Tentukan sendiri	kendaraanGuede

8. Jalankan program yang sudah dibuat dengan syarat program Anda harus menampilkan:

- Informasi customer1 dengan kendaraan yang dibeli beserta total harganya.
- Informasi customer2 dengan kendaraan yang dibeli beserta total harganya.
- Informasi customer3 dengan kendaraan yang dibeli beserta total harganya.

Buktikan hasil program Anda dengan *screenshot output-nya*!

Contoh *output* keseluruhan program:

```
-----  
Customer Name: Bapak Joko  
Brand: Honda Supra  
Year: 1998  
Type: Motor  
Harga: 3000.0  
-----  
-----
```

```
Customer Name: Ibu Siti  
Brand: VW Beetle  
Year: 1998  
Type: Mobil  
Harga: 200000.0  
-----  
-----
```

```
Customer Name: Bapak Budi  
Brand: Isuzu Giga  
Year: 2011  
Type: Truk  
Harga: 300000.0  
-----
```

9. Jangan lupa untuk *push* program yang sudah Anda buat ke *repository* Github Anda. Instruksi untuk *push* dapat dilihat di PreCS bagian PART II nomor 14 - 16.

Dengan format "<CS><No.Modul> - <deskripsi>".