

Soal Tugas Praktikum Modul 07

Dasar Pemrograman Berorientasi Objek – CCK1IAB3

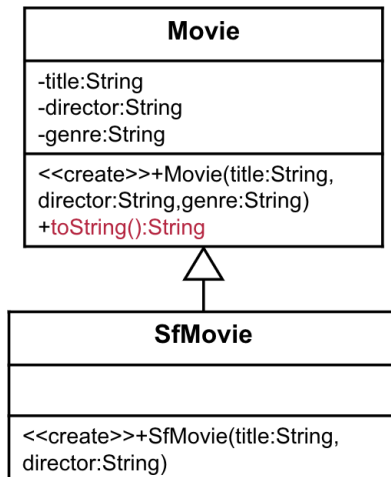
Abstract dan Interface

Petunjuk Umum

1. Tugas Praktikum bersifat individu, sehingga praktikan tidak diperbolehkan untuk berdiskusi satu sama lain, berdiri, pindah tempat duduk ataupun membuat kegaduhan selama Praktikum.
2. Tugas Praktikum terdiri dari sesi terbimbing, mohon membaca teknis kedua sesi tersebut!
 - a. **Sesi Terbimbing**
 - Asisten Praktikum (AsPrak) memberikan tutorial di depan kelas.
 - Praktikan mengikuti dan bertanya terkait tutorial yang diberikan
 - Praktikan mengerjakan soal terbimbing lainnya dibantu oleh AsPrak, baik secara logika maupun secara sintak.
3. Pengumpulan tugas melalui LMS mengikut petunjuk yang diberikan oleh AsPrak.

Jurnal

1. Buatlah program Java untuk menampilkan informasi tentang film ber-genre *science fiction*. Implementasi program ini mencakup beberapa kelas sesuai dengan diagram kelas yang diberikan.



Kelas Movie

- a. Tempatkan kelas pada package `dpbo.jurnal.week7`
- b. Atribut:
 - Title (String): Judul film.
 - Director (String): Nama sutradara film.
 - Genre (String): Genre film
- c. Konstruktor:

Movie (String title, String director, String genre): Menerima judul, nama sutradara, dan genre untuk menginisialisasi objek **Movie**.
- d. Metode:

toString(): Mengembalikan deskripsi lengkap dari film dalam format:

Judul: <title>, Director: <director>, Genre: <genre>

Kelas SfMovie (Turunan dari Movie)

- Tempatkan kelas pada package `dpbo.jurnal.week7`
- Konstruktor:
 - `SfMovie(String title, String director)`: Konstruktor yang hanya menerima judul dan sutradara film, sedangkan genre secara otomatis diatur ke “SF” (Science Fiction)
- Tidak ada metode tambahan di kelas ini, **override** method dari class parent.
- Gunakan keyword **super()**

Kelas MovieTest (Main)

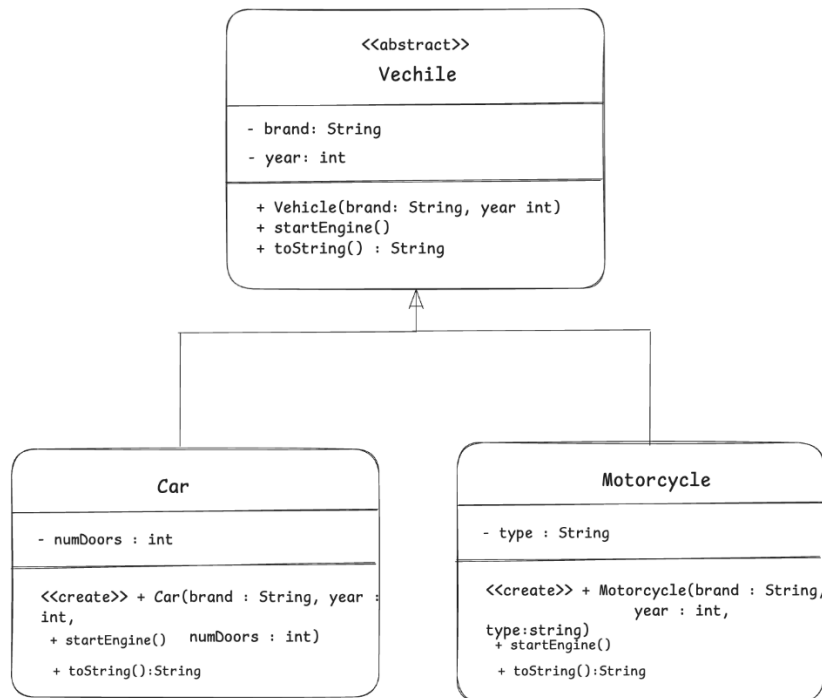
- Tempatkan kelas pada package `dpbo.jurnal.week.7`
- Implementasikan kelas **Movie** dan **SfMovie** berdasarkan deskripsi di atas dan diagram kelas yang diberikan.
- Gunakan kelas **Test/Main** untuk menguji program:
 - Buat dua objek **SfMovie**:
`SfMovie("Avatar", "James Cameron")`
`SfMovie("AI", "Steven Spielberg")`
 - Cetak informasi film untuk masing-masing objek menggunakan metode **toString()**.

```
public class MovieTest {  
    public static void main(String[] args) {  
        SfMovie[] sfMovies = {new SfMovie("Avatar", "James Cameron"),  
                               new SfMovie("AI", "Steven Spielberg")};  
  
        for (int inx = 0; inx < sfMovies.length; inx++) {  
            System.out.println(sfMovies[inx]);  
        }  
    }  
}
```

Output yang diharapkan:

```
Judul: Avatar, Director: James Cameron, Genre: SF  
Judul: AI, Director: Steven Spielberg, Genre: SF
```

2. Buatlah sebuah program Java yang terdiri dari beberapa kelas untuk mengelola data kendaraan. Program ini akan mengimplementasikan konsep **super()**, **@Override**, kelas **Objek**, dan kelas **abstract**.



Kelas Abstract Vehicle

- Tempatkan kelas pada package `dpbo.jurnal.week7`
- Buat sebuah kelas abstrak bernama **Vehicle** dengan atribut berikut:
 - `brand`: Nama merek kendaraan (tipe `String`)
 - `year`: Tahun pembuatan kendaraan (tipe `int`)
- Tambahkan konstruktor `Vehicle(String brand, int year)` untuk menginisialisasi atribut `brand` dan `year`.
- Deklarasikan metode **abstrak public abstract void startEngine()** yang akan diimplementasikan oleh kelas turunan.
- Implementasikan metode **@Override public String toString()** untuk menampilkan informasi kendaraan dalam format:

Brand: <brand>, Year: <year>

Kelas Car

- Tempatkan kelas pada package `dpbo.jurnal.week7`
- Buat kelas **Car** yang merupakan turunan dari **Vehicle** dengan atribut tambahan:
 - `numDoors`: Jumlah pintu (tipe `int`)
- Buat konstruktor `Car(String brand, int year, int numDoors)` yang memanggil konstruktor **Vehicle** menggunakan `super(brand, year)` dan menginisialisasi **numDoors**.
- Implementasikan metode **startEngine()** untuk menampilkan pesan: **"Car engine started with button."**
- @Override** metode **toString()** untuk menampilkan informasi dalam format

Brand: <brand>, Year: <year>, Doors: <numDoors>

*Gunakan keyword **super()**

Kelas Motorcycle

- Tempatkan kelas pada package `dpbo.jurnal.week7`
- Buat kelas **Motorcycle** yang juga merupakan turunan dari **Vehicle** dengan atribut tambahan:
 - `type`: Jenis motor (contoh: "Sport", "matic", "cruiser") (tipe `String`)
- Buat konstruktor `Motorcycle(String brand, int year, String type)` yang memanggil konstruktor **Vehicle** menggunakan `super(brand, year)` dan menginisialisasi `type`.
- Implementasikan metode **startEngine()** untuk menampilkan pesan: **"Motorcycle engine started with key."**
- @Override** metode **toString()** untuk menampilkan informasi dalam format:

Brand: <brand>, Year: <year>, Type: <type>

*Gunakan keyword **super()**

Kelas VehicleTest/Main

- a. Tempatkan kelas pada package dpbo.jurnal.week7.vehicletest
- b. Disediakan kelas VehicleTest untuk menguji program.
- c. Di dalam metode main, telah dilakukan hal berikut:
 - Membuat objek Car dengan data: "Toyota", 2021, dan 4 pintu.
 - Membuat objek Motorcycle dengan data: "Yamaha", 2020, dan Sport.
 - Memanggil metode startEngine() dan toString() dari masing-masing objek, dan tampilkan hasilnya di konsol.

```
public class VehicleTest {  
    public static void main(String[] args) {  
        Vehicle car = new Car("Toyota", 2021, 4);  
  
        car.startEngine();  
        System.out.println(car.toString());  
  
        Vehicle motorcycle = new Motorcycle("Yamaha", 2020, "Sport");  
  
        motorcycle.startEngine();  
        System.out.println(motorcycle.toString());  
    }  
}
```

Output yang diharapkan

```
Car engine started with button.  
  
Brand: Toyota, Year: 2021, Doors: 4  
  
Motorcycle engine started with key.  
  
Brand: Yamaha, Year: 2020, Type: Sport
```