

UAS IF2210 2022: Bagian 2

Ujian Akhir Semester IF2210 Pemrograman Berorientasi Objek, Semester II 2021/2022

Petunjuk

- Waktu: **135** menit
- Untuk jawaban (A), submit dalam bentuk zip file semua source code Java yang dibuat, dengan nama file <NIM>.zip, misalnya 13589010.zip.
- Tuliskan jawaban dalam satu file (pdf atau docx) untuk jawaban (B) dan (C), dengan nama file <NIM>.pdf, misalnya 13589010.pdf.

SOAL: Mudik Wakanda

Hanya ada satu Mudik Wakanda di dunia ini, dengan jumlah kapasitas jalur tertentu. Untuk melewati sebuah jalur saat mudik, kendaraan dikenai biaya.

Mudik Wakanda biasanya dijalani oleh berbagai jenis kendaraan, seperti: mobil dan kereta api. Jenis kendaraan lainnya mungkin saja ditambahkan suatu saat. Mobil bisa bergerak, bisa berhenti kapan saja dan di mana saja, bisa putar balik dan comfortable. Kereta dapat bergerak maju mundur, berhenti di stasiun tertentu sesuai jadwal, tidak bisa putar balik, dan comfortable. Salah satu dari kedua jenis kendaraan ini bisa menyerupai yang lainnya.

Mudik ini sangat diminati pengendara sehingga informasi ketersediaan jalur mudik ini selalu diberitakan kepada kendaraan yang terlibat (apakah jalur penuh atau masih ada jalur tersedia). Statistik pengguna jalur juga diinfokan (misal: berapa mobil dan kereta api yang sudah lewat)

Setiap jalur memiliki biaya dasar, dan bisa ditambahkan biaya penunjang lainnya setelah kendaraan melewati jalur.

Jenis jalur mudik Wakanda dibagi menjadi:

- Arteri: Rp 150000 per hari
- Tol: Rp 750000 per hari
- Kereta/One Way: Rp 1000000 per hari

Penunjang yang disediakan adalah:

- Food street: Rp 150000 per hari
- Go Massage: Rp 50000 per hari
- Netkliks: Rp 75000 per hari

Tagihan jalur dihitung berdasarkan tarif jalur, jumlah hari dan penunjang yang digunakan. Setelah kendaraan sampai tujuan, jalur akan available (bertambah kapasitasnya).

Tugas anda adalah:

- A. Implementasikan **interface** (bobot 25 %) dan **minimal dua design pattern** (bobot 25 %) dari kasus Mudik Wakanda ini.
- B. Sebutkan prinsip **SOLID** mana yang paling relevan dengan kasus ini (1 prinsip saja) dan jelaskan bagaimana kelas-kelas yang Anda buat sudah memenuhi prinsip tersebut (bobot 25 %)
- C. (bobot 25 %) Jika kelas mobil berbentuk seperti berikut ini:

```
import java.util.*;

class Mobil {
    private int harga;
    private List mirip;

    Mobil(int salary) {
        this.harga = harga;
        mirip = new ArrayList();
    }

    private void tambahMobilToMirip(String name) {
        this.mirip.add(name);
    }
}
```

dengan data harga mobil dan juga daftar mobil yang mirip harganya. Kemudian diperlukan informasi berapa harga tiap mobil dan juga menambahkan mobil ke dalam daftar mobil yang mirip harganya. Perhatikan bahwa atribut harga dan method untuk menambahkan nama mobil bersifat private di kelas Mobil, sehingga kita harus menggunakan **reflection** untuk mengaksesnya. Buatlah kelas MobilDecoder yang memiliki interface sebagai berikut.

```
import java.lang.reflect.*;

public class MobilDecoder {

    MobilDecoder(Mobil mobil) {

    }

    public void tambahMobilToMirip (String name) throws Exception {

    }

    public int getHarga() throws Exception {

    }
}
```

Hint: Panggil `setAccessible(true)` pada Field/Method untuk mengakses Field/Method yang private.