13520034 – Bryan Bernigen

A)Interface : terdapat pada MajuMundur, Sampai, Putarbalik, Comfortable, dan KeluarMasukJalur

Design pattern yang dipakai:  
1. Composite karena kelas seperti mobil dan kereta di wrap oleh kelas kendaraan, lalu kelas arteri, tol, dan one way diwrap oleh jalur, lalu food street, go massage, dan netkliks diwrap oleh class penunjang

2. Observer karena terdapat sistem “publish” dan “subscribe” dimana setiap mobil yang masuk ke dalam sebuah jalur akan dimasukan sebagai subsriber dan sebuah jalur dapat mengpusblish jumlah kendaraan yang ada pada jalur tersebut dan jumlah total mobil dan kereta api yang sudah lewat (berapa banyak yang sudah keluar jalur dan sampai di tujuan mudik).

3.Chain of command karena terdapat sistem mencari apakah masih ada jalur yang kosong atau semua sudah penuh. Dengan mencampur design chain of command dan observer, kita dapat memberi tahu para pengendara apakah masih terdapat sebuah jalur yang kosong atau tidak. Optimalnya adalah mendapatkan informasi mengenai seluruh jalur, namun karena pada soal diminta apakah sebuah jalur penuh atau masih ada jalur tersedia, maka chain of command digunakan untuk mencari apakah masih terdapat jalur yang belum penuh oleh kendaraan.

B) Prinsip Solid yang paling relevan untuk dipakai pada persoalan kali ini adalah prinsip O yakni Open Closed Principle karena dengan prinsip tersebut, sebuah kelas akan mudah untuk ditambah namun tidak untuk di ubah. Untuk kasus ini, dengan konsep tersebut, maka akan mudah untuk menambahkan kendaraan baru, jalur baru, ataupun penunjang baru tanpa memengaruhi kode kelas-kelas lainnya. Kelas-kelas yang saya buat mengadaptasi prinsip tersebut dengan cara mengextend sebuah kelas parent ataupun mengimplement sebuah interface. Sebagai contoh, kelas Mobil mengextend kelas kendaraan dan kelas kereta mengextend kelas kendaraan. Lalu pada Jalur, kelas yang dimasukkan kedalam data jalur tersebut adalah kelas kendaraan sehingga jika nantinya terdapat sebuah kendaraan baru, akan mudah untuk menambahkan kendaraan tersebut ke dalam sebuah jalur. Kode pada kelas jalur tidak perlu diubah. Kendaraan baru tersebut cukup mengextend kelas kendaraan dan kendaraan baru tersebut siap untuk dimasukkan ke dalam lajur. Begitu pula dengan tipe jalur yang dapat dengan mudah ditambah dan tipe penunjang yang dapat ditambah dengan mudah. Hanya perlu mengextend kelas jalut atau kelas penunjang dan jalur ataupun penunjang baru tersebut siap untuk digunakan.

C)

Mobil.java

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);

    }

}

MobilDecoder.java

import java.lang.reflect.\*;

public class MobilDecoder {

    Mobil mobil;

    MobilDecoder(Mobil mobil) {

        this.mobil = mobil;

    }

    public void tambahMobilToMirip(String name) throws Exception {

        Class ClassMobil = mobil.getClass();

        Method method = ClassMobil.getDeclaredMethod("tambahMobilToMirip", String.class);

        method.setAccessible(true);

        method.invoke(mobil, name);

    }

    public int getHarga() throws Exception {

        Class ClassMobil = mobil.getClass();

        Field field = ClassMobil.getDeclaredField("harga");

        field.setAccessible(true);

        return field.getInt(mobil);

    }

}