Date _____ UAS Pemograman Berorientasi Objek

Salma Nora Renada
2 90601 20130050
Informatika B

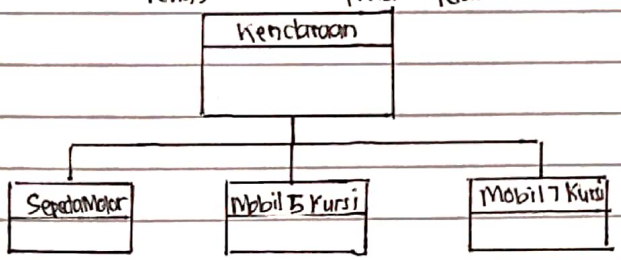
1.) Saya, Salma Nora Renada . NIM : 2 90601 20130050 mengerjakan ujian dibantu orang lain bernama Tanda Tangan : [redacted]

2) a. Rancanglah sebuah kelas Kendaraan yang berisi informasi nomor plat kendaraan, nama pengendara, dan jumlah max penumpang, disertai metode yang relevan!

Kendaraan
- noPlat : String
- namaPengendara : varChar
- maxPenumpang : int
+ Kendaraan ()
+ get noPlat () : String
+ get namaPengendara () : varChar
+ get maxPenumpang () : int
+ set noPlat (plat : string)
+ set namaPengendara (nama : varChar)
+ set maxPenumpang (max : int)

b. Rancanglah 3 kelas anak dari kelas Kendaraan!

Penggambaran Anak kelas




```
class sepedaMotor extends Kendaraan {
    public SepedaMotor (String noPlat, varChar namaPengendara, int maxPenumpang) {
        super (noPlat, namaPengendara, maxPenumpang);
    }
}

class mobil5kursi extends Kendaraan {
    public Mobil5kursi (String noPlat, varChar namaPengendara, int maxPenumpang) {
        super (noPlat, namaPengendara, maxPenumpang);
    }
}

class mobil7kursi extends Kendaraan {
    public mobil7kursi (String noPlat, varChar namaPengendara, int maxPenumpang) {
        super (noPlat, namaPengendara, maxPenumpang);
    }
}
```

Date

Salma Nora Penada
2408012001300 50

C. Buatlah deklarasi dan instantiasi setiap kelas anak dengan konsep Polimorfisme!

```
SepedaMotor sepedaMotor = new SepedaMotor ("H3201C", "Iki", 2);
MobilsKursi MobilsKursi = new MobilsKursi ("B2011D", "Nada", 5);
Mobil7Kursi Mobil7Kursi = new Mobil7Kursi ("F01213C", "Taski", 7);
```

d. Berikan contoh pengisian nilai dan cara akses informasi setiap instans kelas anak!

```
// isi nilai
SepedaMotor. set noPlat = "H 3201 C";
MobilsKursi. set noPlat = "B 2011D")
Mobil7Kursi. set noPlat = "F01213C";
SepedaMotor. set namaPengendara = "Iki";
MobilsKursi. set namaPengendara = "Nada";
Mobil7Kursi. set namaPengendara = "Taski";
SepedaMotor. set maxPenumpang = 2;
MobilsKursi. set maxPenumpang = 5;
Mobil7Kursi. set maxPenumpang = 7;

// get / akses informasi
SepedaMotor. get noPlat();
MobilsKursi. get noPlat();
Mobil7Kursi. get noPlat();
SepedaMotor. get namaPengendara();
MobilsKursi. get namaPengendara();
Mobil7Kursi. get namaPengendara();
SepedaMotor. get maxPenumpang();
MobilsKursi. get maxPenumpang();
Mobil7Kursi. get maxPenumpang();
```

Date

Salma Nara Renada
2406012013005

3) a) Public class koleksi (T extends kendaraan) {

public ArrayList (T) NIK ;

public ~~String~~ varChar namaPenumpang ;

public koleksi (~~String~~ varChar namaPenumpang) {

this. namaPenumpang = namaPenumpang ;

this. NIK = new ArrayList (T) ;

}

public ArrayList (T) get.NIK () {

return this. NIK ;

}

public void add NIK (kendaraan. kendaraan) {

this. NIK add (kendaraan)

}

}

b) koleksi (kendaraan) penumpang = new koleksi (kendaraan) : ("3373021512134")

c) koleksi (kendaraan) penumpang : new koleksi (kendaraan)

penumpang. add NIK ("3373021512134")

d) Kendaraan sepedaMotor = new sepedaMotor ()

Kendaraan mobil5Kursi = new mobil5Kursi ()

mobil5Kursi. get maxPenumpang

koleksi. kendaraan (mobil5kursi)

koleksi. addNIK (sepedaMotor)

Date

Salma Nara Peracta
24060120130050

4.) Dalam pemograman berorientasi objek, dikenal 5 prinsip utama yang biasa disingkat SOLID. Lakukan analisis atas kemungkinan cara penerapan setiap prinsip dalam perancangan kelas** untuk kasus aplikasi BONCENG!

▲ SOLID : SRP, OCP, LSP, ISP, DIP

* SRP > masing** dari superclass dan subclass harus dan hanya memiliki 1 tanggung jawab yaitu menyimpan data jumlah kasus & tanggal dari kasus tsb.
* OCP > kasus harus dapat diwariskan method & atributnya kepada kelas anak. Tetapi kelas anak tdk boleh diubah oleh superclass.
* LSP > subclass diimplementasikan maka program tdk boleh mengalami kesalahan & tetap berjalan dg semestinya walau superclass diganti.
* ISP > u/ menerapkan ISP kita dapat membuatkan interface u/ masing** subclass kasus.
* DIP > Kelass yg memiliki keruntutan tinggi tidak boleh memiliki ketergantungan dg kelas yg keruntutannya lebih rendah.