

Mata Kuliah : PBO – TI – S1

Pertemuan : Soal UTS Teori Tahun 2024

NIM : A11.2022.14532

Nama : Najma Aura Dias Prameswari

1. Jelaskan dan berikan contoh apa yang dimaksud dengan :
 - a. Class : Sebuah blueprint untuk membuat objek. Ini mendefinisikan attribute(data) & metode (fungsi) yang akan dimiliki oleh objek dari kelas tersebut.
 - b. Object : Perwujudan konkret dari sebuah kelas. Objek memiliki atribut yang sesuai dengan kelasnya dan dapat melakukan tindakan yang didefinisikan dalam metode kelasnya.
 - c. Attribut : Data yang disimpan di dalam sebuah objek. Mempresentasikan karakteristik dari objek tersebut. Seperti nama, usia, warna dll.
 - d. Method : Fungsi yang terkait dengan objek tertentu. Digunakan untuk melakukan tindakan tertentu pada objek / mengubah attribute objek.
 - e. Encapsulation : Konsep dalam pemrograman yang mengizinkan pembungkusan atribut dan metode dalam sebuah kelas sehingga mereka tidak dapat diakses dari luar kelas tersebut. Ini membantu dalam menyembunyikan detail implementasi dan mencegah akses yang tidak diinginkan, sehingga meningkatkan keamanan dan keandalan kode.
 - f. Inheritance : Suatu class dapat mewariskan atribut dan method kepada class lain(subclass) serta membentuk class hierarchy (hirarki).
 - g. Overloading : Penggunaan satu nama untuk beberapa method yang berbeda parameter.
 - h. Contoh code dari soal 2 :

```
public class soal_teoris_uts {
    public static void main(String[] args)
    {
        Nilai obj = new Nilai(90, 90, 85); //object
        obj.nim = "A.11.";
        obj.nama = "Najma";
        obj.na = (obj.tgs + obj.uts + obj.uas)/3;
        obj.cetak1();
        obj.cetak1("Nilai");
    }
}

class mhs{ // class & konstruktor
    String nim, nama; //attribut
    float tgs, uts, uas; //attribut
    mhs(float a, float b, float c)
    {
        tgs = a;
        uts = b;
        uas = c;
    }
    void cetak1() //method & overloading
    {
        System.out.println("Nama : "+nama);
        System.out.println("Tugas : "+tgs);
    }
}
```

```

        System.out.println("UTS : "+uts);
        System.out.println("UAS : "+uas);
    }
    void cetak1(String s)
    {
        System.out.println(s);
    }
}

class Nilai extends mhs{ //merupakan class & inheritance
    float na;
    Nilai(float l, float m, float n)
    {
        super(l,m,n); na=0;
    }
    void cetak1()//overriding
    {
        super.cetak1("Mahasiswa");
        super.cetak1();
    }
    void cetak1(String x)
    {
        System.out.println(x);
        System.out.println("Nim : "+super.nim);
        System.out.println("Nama : "+super.nama);
        System.out.println("Tugas : "+tgs);
        System.out.println("UTS : "+uts);
        System.out.println("UAS : "+uas);
        System.out.println("NA : "+na);
    }
}

```

contoh encapsulation dari public berubah ke private :

```

class mhs {
    private String nim, nama;
    private float tgs, uts, uas;
    // ...
}

```

2. Tentukan :

a. Konstruktor :

```

class mhs {
    mhs(float a, float b, float c) {
        // ...
    }
}

```

```

class Nilai extends mhs {
    Nilai(float l, float m, float n) {
        super(l, m, n);
        // ...
    }
}

```

b. Inheritance :

```
class Nilai extends mhs {  
    // ...  
}
```

c. Overloading, Overriding :

```
class mhs {  
    // ...  
    void cetak1() {  
        // ...  
    }  
    void cetak1(String s) {  
        // ...  
    }  
}
```

Overriding :

```
class Nilai extends mhs {  
    // ...  
    void cetak1() {  
        // ...  
    }  
    void cetak1(String x) {  
        // ...  
    }  
}
```

d. Hasil program

Mahasiswa

Nama : Najma

Tugas : 90

UTS : 90

UAS : 85

Nilai

Nim : A11.

Nama : Najwa

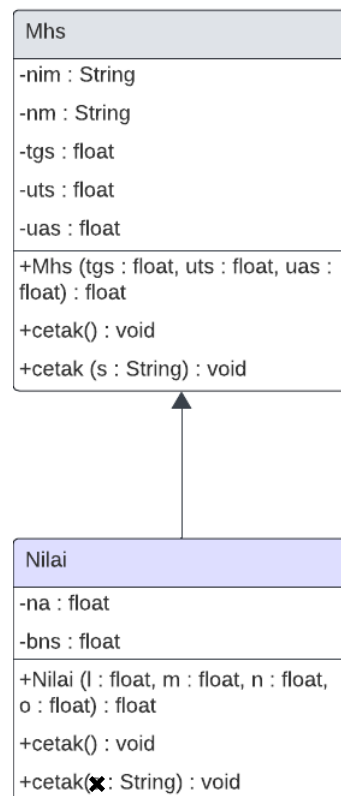
Tugas : 90

UTS : 90

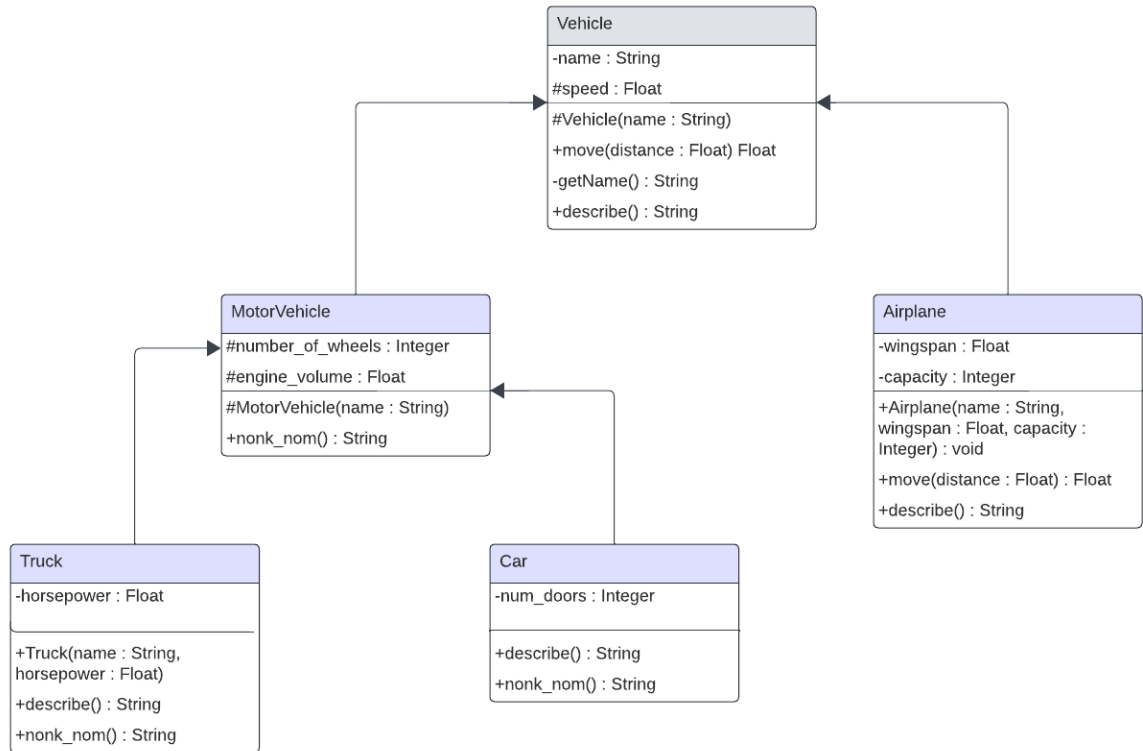
UAS : 85

NA : 88

e. Gambar class diagram :



3. Perhatikan class diagram berikut :



a. Buatlah design class dari class diagram diatas !.

```

b. class vehicle {
    private String name;
    protected float speed;

    protected vehicle (String name)
    {
        this.name = name;
        this.speed = 0.0f; //klo default = 0.0f
    }

    public float move (float distance){
        if (speed > 0){
            return distance / speed;
        }else {
            return -1;
        }
    }

    public String getName()//dari soal pake -/private, tetapi
    tidak bisa memanggil jika dipake di class lain
    {
        return name;
    }

    public String describe()
    {
        return "Vehicle name : " + name + ", speed : " + speed +
  
```

```

    " units";
    }

}

class MotorVehicle extends vehicle{
    protected int number_of_wheels;
    protected float engine_volume;

    protected MotorVehicle(String name)
    {
        super(name);
        this.number_of_wheels = 0;
        this.engine_volume = 0.0f;
    }

    public String nonk_nom(){
        return "MotorVehicle : "+getName();
    }

    public void setEngine_volume(float engine_volume)
    {
        this.engine_volume = engine_volume;
    }

    public float getEngine_volume()
    {
        return engine_volume;
    }
}

class Truck extends MotorVehicle{
    private float horsepower;

    public Truck (String name, float horsepower)
    {
        super(name);
        this.horsepower = horsepower;
    }

    public String describe()
    {
        return "nama truck : " +getName()+ "\n"+
            "volume mesin truck : " +getEngine_volume()+
            "\n"+
            "deskripsi truck : " +horsepower+ "\n"+
            "horsepower truck : "+horsepower;
    }

    public String nonk_nom()
    {
        return "Truck : "+getName();
    }
}

class Car extends MotorVehicle{
    private int num_doors;

```

```

public Car(String name, int num_doors)
{
    super(name);
    this.num_doors = num_doors;
}

@Override
public String describe() {
    return "nama car : " + getName() + "\n"+
           "volume mesin car : " + getEngine_volume() + "\n"+
           "deskripsi car : " + num_doors + "\n"+
           "jumlah pintu : " + num_doors;
}
public String nonk_nom()
{
    return "Car :" + getName();
}
}

class Airplane extends vehicle{
    private float wingspan;
    private int capacity;
    private boolean landingGear;

    public Airplane(String name, float wingspan, int capacity)
    {
        super(name);
        this.wingspan = wingspan;
        this.capacity = capacity;
        this.landingGear = false;
    }

    void landingGear(boolean set) //awalnya pake private tetapi g
    bisa klo mau dipanggil di main
    {
        this.landingGear = set;
    }

    public float move(float distance)
    {
        if (speed > 0)
        {
            return distance / speed;
        }else {
            return -1;
        }
    }

    public String describe(){
        return "nama airplane: " + getName() + "\n"+
               "wingspan airplane: " + wingspan + "\n"+
               "deskripsi airplane: " + capacity + "\n"+
               "capacity airplane: " + capacity + "\n"+
               "landing gear airplane : " + (landingGear ?
               "Deployed" : "Retracted");
    }
}

```

```
}  
}
```

- c. Berikan contoh penerapan untuk mencetak **Car** yang ada di design class pada soal (a) dengan output :

Nama :

Volume mesin :

Deskripsi :

Jumlah pintu :

```
public String describe() {  
    return "nama car : " + getName() + "\n"+  
           "volume mesin car : " + getEngine_volume() + "\n"+  
           "deskripsi car : " + num_doors + "\n"+  
           "jumlah pintu : " + num_doors;  
}  
public class Main{  
    public static void main(String[] args)  
    {  
  
        Car car = new Car("Sedan", 4);  
        car.speed = 80;  
        car.setEngine_volume(2.5f);  
  
        System.out.println(car.describe());  
        float timee = car.move(160);  
        System.out.println("time to travels 160 units : " + timee + " hours");  
        System.out.println(car.nonk_nom());  
  
    }  
}
```