Nama : Restu Lestari Mulianingrum

NIM : A11.2022.14668

Kelompok : A11.4415

1. Jelaskan dan berikan contoh apa yang dimaksud dengan :

   a. Inheritance

   Inheritance (pewarisan) adalah konsep di mana sebuah class dapat mewarisi atribut dan metode dari class lain. Class yang mewarisi disebut subclass atau child class, sedangkan class yang memberikan warisan disebut superclass atau parent class. Contoh:

```
// Superclass (Parent Class)
public static class Animal {
    private String name;

    public Animal(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

}

// Subclass Dog
public static class Dog extends Animal {
    public Dog(String name) {
        super(name);
    }
}

// Main method untuk menguji pewarisan
public static void main(String[] args) {
    Dog dog = new Dog("Molly");

    System.out.println(dog.getName());
}
```

   b. Class, Object, Attribut, Method

   - Class

Class adalah blueprint atau cetakan untuk membuat objek. Class mendefinisikan atribut dan metode yang akan dimiliki oleh objek.

Contoh penulisan class dalam bahasa Java:

```
class Laptop {
    // Isi dari class Laptop...
}
```

- Object

  Objek adalah sebuah variabel yang merupakan instance atau perwujudan dari Class. Jika menggunakan analogi class Laptop, maka objek dari class Laptop bisa berupa “laptopRestu”

```
public class Laptop {
    public static void main(String[] args) {
        // membuat object dengan nama 'laptopRestu'
        Laptop laptopRestu = new Laptop();
        laptopRestu.cetakNama("Grey");
    }
}
```

- Attribut

  Attribut adalah variabel yang terdapat dalam class. Atribut menyimpan data atau keadaan dari sebuah objek.

  Contoh attribut dari class Laptop bisa berupa merk, warna, jenis processor, ukuran layar, dan lain-lain. Contoh penulisan class dengan attribut:

```
class Laptop {
    String pemilik;
    String merk;
    double ukuranLayar;
}
```

- Method

  Method adalah fungsi yang terdapat dalam class. Method mendefinisikan perilaku atau aksi yang bisa dilakukan oleh objek.

  Contoh method pada class Laptop bisa berupa hidupkanLaptop() untuk menghidupkan laptop, matikanLaptop(), gantiCover().

```
class Laptop {
```

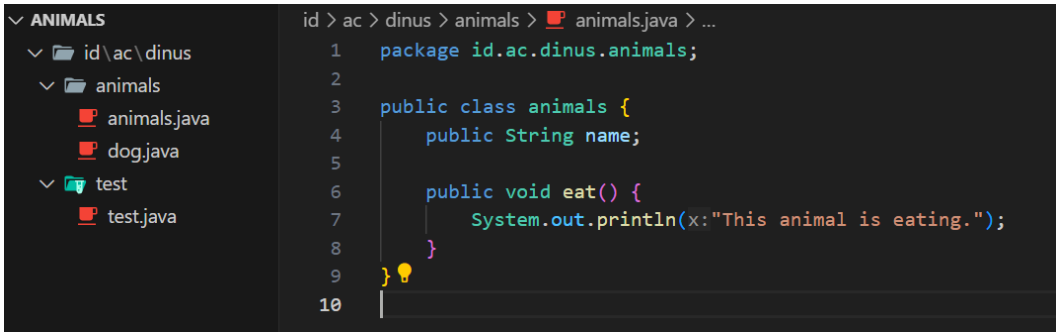
```
    String merk;
    String warna;
    int tahunProduksi;

    void hidupkanLaptop() {
        System.out.println("Laptop dinyalakan.");
    }
}

public class Main {
    public static void main(String[] args) {
        // Membuat objek laptop
        Laptop laptopRestu = new Laptop();
        laptopRestu.merk = "Lenovo";
        laptopRestu.warna = "Grey";
        laptopRestu.tahunProduksi = 2022;

        // Menggunakan metode pada objek
        laptopRestu.hidupkanLaptop();
    }
}
```

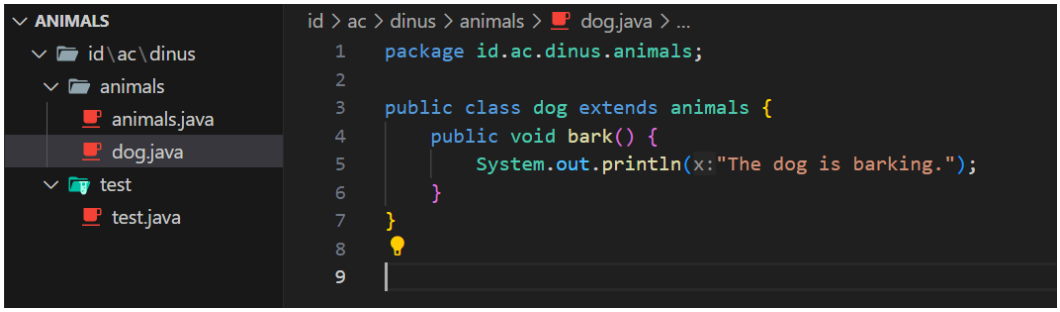
c. Package

Package dalam Java adalah cara untuk mengelompokkan kelas-kelas dan antarmuka-antarmuka yang terkait dalam satu grup. Package membantu mengelola namespace dan mencegah konflik nama, serta mempermudah pengorganisasian kode proyek.

Contoh :

Misalkan terdapat dua package, ‘id.ac.dinus.animals’ untuk mendefinisikan kelas hewan dan ‘id.ac.dinus.test’ untuk menguji kelas hewan.





d. **Overriding :** konsep di mana sebuah method dalam class anak (subclass) menggantikan (meng-"override") method dengan nama yang sama dari class induknya (superclass).

Contoh:

```
class Animal {
    void sound() {
        System.out.println("Animal makes a sound");
    }
}

class Dog extends Animal {
    @Override
    void sound() {
        System.out.println("Dog barks");
    }
}
```

Overloading : adalah kemampuan untuk mendefinisikan beberapa metode dengan nama yang sama di dalam sebuah class. Metode-metode ini memiliki parameter yang berbeda. Java membedakan metode berdasarkan jumlah dan tipe parameter.

Contoh:

```
class Kalkulator {
    int tambah(int a, int b) {
```

```
        return a + b;
    }

    double tambah(double a, double b) {
        return a + b;
    }
}
```

2. Tentukan :

   a. Konstruktor

Class Mhs	Class Nilai

`Mhs(float a, float b, float c) {`
`tgs = a;`
`uts = b;`
`uas = c;`
`}`
	`Nilai(float l, float m, float n, float o) {`
	`super(l, m, n);`
	`na = 0;`
	`bonus = 0;`
	`}`

   b. Inheritance

   Class Nilai mewarisi class Mhs

```
class Nilai extends Mhs {
    // algoritma
}
```

   c. Overloading dan Overriding:

   - Method Overloading:

     1. Terdapat dua metode cetak() dalam kelas Mhs. Yang pertama tidak memiliki parameter dan yang kedua memiliki parameter String cetak(String s).

```
void cetak() {
    System.out.println("Nama :" + nm);
    System.out.println("Tgs :" + tgs);
    System.out.println("Uts :" + uts);
    System.out.println("Uts :" + uas);
}

void cetak(String s) {
    System.out.println(s);
}
```

2. Pada kelas Nilai, terdapat dua metode cetak(), yang merupakan metode overriding dari kelas Mhs: Metode cetak() tanpa parameter dan Metode cetak() dengan parameter String x.

```
void cetak() {
    super.cetak("Mahasiswa");
    super.cetak();
}

void cetak(String x) {
    System.out.println(x);
    System.out.println("Nim = " + nim);
    System.out.println("Nama = " + nm);
    System.out.println("Tgs = " + tgs);
    System.out.println("Uts = " + uts);
    System.out.println("Uas = " + uas);
    System.out.println("NA = " + na);
}
```

- Method Overriding

1. Metode cetak() di kelas Nilai mengoverride metode cetak() dari kelas Mhs. Ini memanggil metode cetak(String) dengan argumen "Mahasiswa" dan memanggil metode cetak() dari kelas Mhs sendiri.

```
// Override metode cetak() dari kelas Mhs
@Override
void cetak() {
    super.cetak("Mahasiswa");
    super.cetak();
}
```

2. Metode cetak(String) di kelas Nilai mengoverride metode cetak(String) dari kelas Mhs. Ini mencetak string yang diberikan (x) dan juga mencetak informasi nama, NIM, tugas, UTS, UAS, dan nilai akhir (na) dari mahasiswa.

```
// Override metode cetak(String x) dari kelas Mhs
@Override
void cetak(String x) {
    System.out.println(x);
    System.out.println("Nim = " + nim);
    System.out.println("Nama = " + nm);
    System.out.println("Tgs = " + tgs);
    System.out.println("Uts = " + uts);
```

```
        System.out.println("Uas = " + uas);
        System.out.println("NA = " + na);
    }
```

d. Hasil Program :

Mahasiswa

Nama: Ghiyatsi

Tgs: 95.0

Uts: 87.0

Uas: 85.0

Nilai Mhs

Nim = A11

Nama = Ghiyatsi

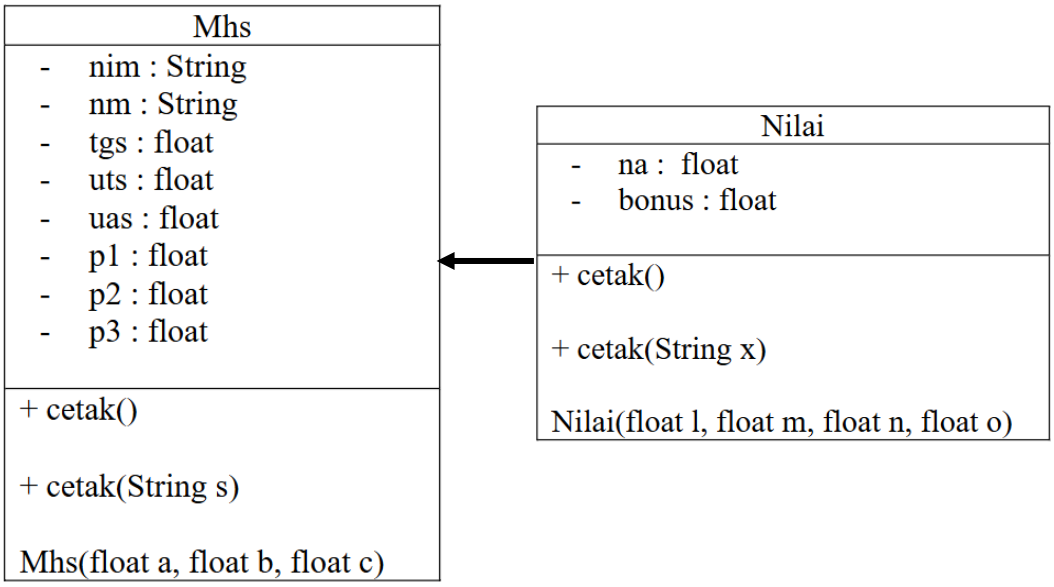
Tgs = 95.0

Uts = 87.0

Uas = 85.0

NA = 88.7

e. Gambar Class Diagram



3.

a. Buatlah design class dari class diagram di atas!

Class Pegawai

```
public class Pegawai {
    private double gajiPokok;
    private String nama;
    private String jabatan;

    Pegawai(String nama){
        this.nama = nama;
    }

    Pegawai(String nama, double gajiPokok, String jabatan){
        this.nama = nama;
        this.gajiPokok = gajiPokok;
        this.jabatan = jabatan;
    }

    Pegawai(String nama, String jabatan) {
        this.nama = nama;
        this.jabatan = jabatan;
    }

    public double getGajiPokok(){
        return gajiPokok;
    }

    public void setGajiPokok(double gajiPokok){
        this.gajiPokok = gajiPokok;
    }

    public String getNama(){
        return nama;
    }

    public String getJabatan(){
        return jabatan;
    }

    public double getTotalGaji(){
        return gajiPokok;
    }

    public void info(){
        System.out.println("Nama\t: " + nama);
        System.out.println("Jabatan\t: " + jabatan);
        System.out.println("Gaji Pokok\t: " + getGajiPokok());
```

```
    }
}
```

Class PegawaiBiasa.java

```
public class PegawaiBiasa extends Pegawai{
    PegawaiBiasa(String nama, double gajiPokok){
        super(nama, gajiPokok, "Pegawai Biasa");
    }
}
```

Class Sekretaris

```
public class Sekretaris extends Pegawai {
    private double upahLembur;

    Sekretaris(String nama, double gajiPokok, double upahLembur){
        super(nama, gajiPokok, "Sekretaris");
        this.upahLembur = upahLembur;
    }

    public double getUpahLembur(){
        return upahLembur;
    }

    public void setUpahLembur(double upahLembur){
        this.upahLembur = upahLembur;
    }

    public double getTotalGaji(){
        return getGajiPokok() + upahLembur;
    }

}
```

Class Manajer

```
public class Manajer extends Pegawai {
    private double bonus;

    Manajer(String nama, double gajiPokok, double bonus) {
        super(nama, gajiPokok, "Manajer");
        this.bonus = bonus;
    }
```

```
    public double getBonus(){
        return bonus;
    }

    public void setBonus(double bonus){
        this.bonus = bonus;
    }

    public double getTotalGaji(){
        return getGajiPokok() + bonus;
    }
}
```

Class EksekutifManajer

```
public class EksekutifManajer extends Manajer{
    private Sekretaris mySekretaris;

    EksekutifManajer(String nama, double gajiPokok, double bonus){
        super(nama, gajiPokok, bonus);
    }

    public Sekretaris getSekretaris(){
        return mySekretaris;
    }

    public void setSekretaris(Sekretaris sekr){
        this.mySekretaris = sekr;
    }

    public double getTotalGaji(){
        return super.getTotalGaji();
    }

    public void info(){
        super.info();
        System.out.println("Bonus\t: " + getBonus());
        System.out.println("Total Gaji\t: " + getTotalGaji());
    }

}
```

b. Berikan contoh penerapan untuk mencetak EksekutifManajer yang ada di design class pada soal (a) dengan output :

Nama :

Jabatan :

Gaji Pokok :

Bonus :

Total Gaji :

testEksekutifManajer.java

```
public class TestEksekutifManajer {
    public static void main(String[] args) {
        Sekretaris sekr = new Sekretaris("Restu", 8000000,
2500000);
        EksekutifManajer em = new EksekutifManajer("Restu
Lestari", 15000000, 5000000);
        em.setSekretaris(sekr);
        em.info();
    }
}
```

Output

```
Nama            : Restu Lestari
Jabatan         : Manajer
Gaji Pokok      : 1.5E7
Bonus           : 5000000.0
TotalGaji       : 2.0E7
```