

Pembahasan UTS

Pemrograman Berbasis Objek 2023

Jawaban hanyalah sebuah referensi, belum tentu benar!

Made with  by Diklat

1.

```
//kode 1-1
class Insertion{
    try{
        //eksekusi insert ke dalam database
    }catch(SQLException exp){
        System.out.println("Terjadi kesalahan SQL.");
    }catch(Exception exp){
        System.out.println("Terjadi kesalahan program.");
    }catch(Throwable thr){
        System.out.println("Terjadi kesalahan");
    }
}
```

2.

```
//kode 2-1
interface ILuasPermukaan{
    public double hitungLuasPermukaan();
}
```

a. Kelas Kubus

```
public class Kubus extends ILuasPermukaan{
    private double sisi;

    public Kubus(double sisi){
        this.sisi = sisi;
    }

    public double getSisi(){
        return sisi;
    }

    public void setSisi(double sisi){
        this.sisi = sisi;
    }

    public double hitungLuasPermukaan(){
        return sisi * sisi;
    }
}
```

b. Kelas Main

```

public class Main{
    public static void main(String[] args){
        Kubus k1 = new Kubus(10);
        System.out.println("Luas dari kubus dengan panjang " +
            k1.getSisi() + " satuan adalah " + k1.hitungLuasPermukaan());
    }
}

```

3. Perbedaan atribut sebagai instance member vs class member. Berikan contoh deklarasi dan pemanggilannya menggunakan Java

a. **Instance member:**

- Atribut dan method merupakan anggota dari setiap objek.
- Perlu instansiasi untuk mengakses

Class member:

- Atribut dan method merupakan anggota dari sebuah class.
- Tidak perlu instansiasi untuk mengakses, karena setiap objek dapat mengakses ke static member

Contoh:

```

//Deklarasi
public class Circle{
    /*class member*/
    private double radius;
    /*instance member*/
    private static int numberOfCircles = 0;

    public Circle(double radius){
        this.radius = radius;
        numberOfCircle++;
    }

    /*class member*/
    public double getRadius(){
        return radius;
    }

    /*instance member*/
    public static int getNumberOfCircles(){
        return numberOfCircle;
    }
}

//Pemanggilan
public class Main{
    public static void main(String[] args){
        Circle c1 = new Circle(20);
        /*class member*/
        c1.getRadius();
        /*instance member*/
        Circle.getNumberOfCircles();
    }
}

```

- b. **Overloading**: Ketika method memiliki nama yang sama dalam sebuah class. Method yang overloading akan memiliki parameter dan return type yang berbeda.

```
public class Math1{
    public int add(int a, int b){
        return a + b;
    }

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

Overriding: Mendefinisikan ulang method yang ada di superclass. Method yang di override di subclass harus memiliki signature dan return type yang sama dengan method di superclass-nya.

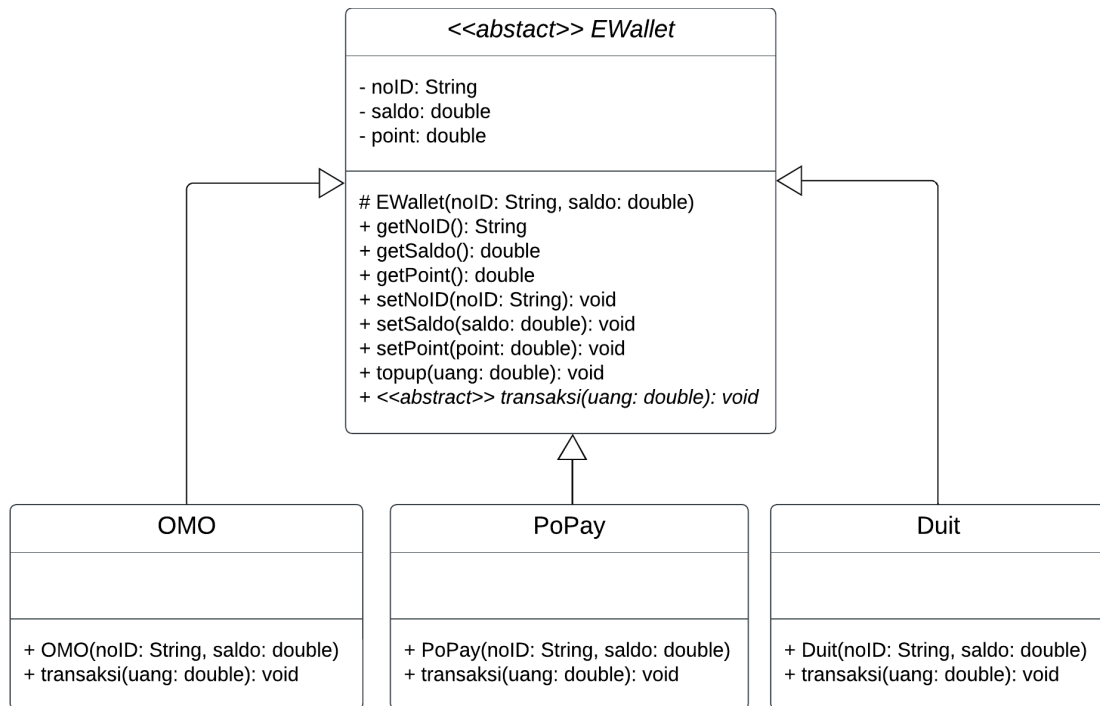
```
public class Poligon{
    protected int jumlahSisi;

    public void printInfo(){
        System.out.println("Jumlah sisi: " + jumlahSisi);
    }
}

public class PersegiPanjang extends Poligon{
    private double panjang;
    private double lebar;

    public void printInfo(){
        super.printInfo();
        System.out.println("Panjang: " + panjang);
        System.out.println("Lebar: " + lebar);
    }
}
```

4. DENGAN NIM 240601xxxxx014



```

public abstract class EWallet{
    private String noID;
    private double saldo;
    private double point;

    protected EWallet(String noID, double saldo){
        this.noID = noID;
        this.saldo = saldo;
        this.point = 0;
    }

    public String getNoID(){
        return noID;
    }

    public double getSaldo(){
        return saldo;
    }

    public double getPoint(){
        return point;
    }

    public void setNoID(String noID){
        this.noID = noID;
    }

    public void setSaldo(double saldo){
        this.saldo = saldo;
    }

    public void setPoint(double point){
        this.point = point;
    }
}

```

```
public void topup(double uang){
    saldo = saldo + uang;
}

public abstract void transaksi(double uang);
}
```

```
public class OMO extends EWallet{
    public OMO(String noID, double saldo){
        super(noID,saldo);
        setPoint(0);
    }

    public void transaksi(double uang){
        assert(getSaldo() - uang >= 0):"Saldo tidak mencukupi!";
        if(getSaldo() - uang >= 0){
            double s = getSaldo() - uang;
            setSaldo(s);
            if(uang > 300000){
                double p = getPoint() + (1 * 10);
                setPoint(p);
            }
        }
    }
}

public class PoPay extends EWallet{
    public PoPay(String noID, double saldo){
        super(noID,saldo);
        setPoint(0);
    }

    public void transaksi(double uang){
        assert(getSaldo() - uang >= 0):"Saldo tidak mencukupi!";
        if(getSaldo() - uang >= 0){
            double s = getSaldo() - uang;
            setSaldo(s);
            if(uang > 300000){
                double p = getPoint() + (4 * 10);
                setPoint(p);
            }
        }
    }
}

public class Duit extends EWallet{
    public Duit(String noID, double saldo){
        super(noID,saldo);
        setPoint(0);
    }

    public void transaksi(double uang){
        assert(getSaldo() - uang >= 0):"Saldo tidak mencukupi!";
        if(getSaldo() - uang >= 0){
            double s = getSaldo() - uang;
            setSaldo(s);
            if(uang > 300000){
```

```
        double p = getPoint() + ((1 + 4) * 5);  
        setPoint(p);  
    }  
}  
}
```