



# Reflection Test

Raditya Prirahmadian – 140810160009

Syafira Fitra Annisa – 140810160047

Patricia Joanne - 140810160065

# Soal A

## 1. *Apakah yang dimaksud dengan overloading?*

**Overloading** adalah suatu keadaan dimana beberapa method sekaligus dapat mempunyai nama yang sama, akan tetapi mempunyai fungsionalitas yang berbeda. Contoh penggunaan overloading dilihat di bawah ini:

- Gambar(Titik t1) → 1 parameter titik, untuk menggambar titik
- Gambar(Titik t1, Titik t2) → 2 parameter titik, untuk menggambar garis
- Gambar(Titik t1, Titik t2, Titik t3) → 3 parameter titik, untuk menggambar segitiga
- Gambar(Titik t1, Titik t2, Titik t3, Titik t4) → 4 parameter titik, untuk menggambar persegi empat

Overloading ini dapat terjadi pada class yang sama atau pada suatu parent class dan subclass-nya. Overloading mempunyai ciri-ciri sebagai berikut:

- Nama method harus sama
- Daftar parameter harus berbeda
- Return type boleh sama, juga boleh berbeda

## 2. *Apakah yang dimaksud dengan overriding?*

**Overriding** adalah suatu keadaan dimana method pada subclass menolak method pada parent class-nya. Overriding mempunyai ciri-ciri sebagai berikut:

- Nama method harus sama
- Daftar parameter harus sama
- Return type harus sama

### 3. ***Jelaskan aturan tentang overridden!***

Method yang terkena override (overridden method) diharuskan tidak boleh mempunyai modifier yang lebih luas aksesnya dari method yang meng-override (overriding method).

### 4. ***Apakah yang dimaksud dengan polimorfisme?***

Polimorfisme adalah kemampuan untuk mempunyai beberapa bentuk class yang berbeda. Polimorfisme ini terjadi pada saat suatu obyek bertipe parent class, akan tetapi pemanggilan constructornya melalui subclass.

5. ***Jelaskan proses terjadinya Virtual Method Invocation!***

**Virtual Method Invocation (VMI)** bisa terjadi jika terjadi polimorfisme dan overriding. Pada saat obyek yang sudah dibuat tersebut memanggil overridden method pada parent class, kompiler Java akan melakukan invocation (pemanggilan) terhadap overriding method pada subclass, dimana yang seharusnya dipanggil adalah overridden method.

6. ***Apakah yang dimaksud dengan polymorphic arguments?***

**Polymorphic arguments** adalah tipe suatu parameter yang menerima suatu nilai yang bertipe subclass-nya.

### 7. *Apakah kegunaan kata kunci instanceof?*

Pernyataan **instanceof** sangat berguna untuk mengetahui tipe asal dari suatu polymorphic arguments. Seringkali pemakaian instanceof diikuti dengan casting object dari tipe parameter ke tipe asal.

### 8. *Apa perbedaan antara Abstract dan Interface?*

Perbedaan	Abstract	Interface
<b>Attribute</b>	Tipe data apa saja	Hanya berupa konstanta
<b>Method</b>	Boleh deklarasi, boleh method lengkap	Berupa deklarasi
<b>Syntax</b>	Sebagian abstract	Seluruhnya abstract

# Soal B

## ***1. Apakah yang Anda ketahui tentang Inner Class?***

**Inner Class** adalah kelas yang disisipkan di dalam kelas yang lain. Fungsi kelas sisipan ini adalah mendukung suatu proses yang akan dijalankan oleh kelas utamanya. Inner Class bersifat tersarang terhadap kelas-kelas utamanya, seperti halnya blok penyeleksian (if, for) yang tersarang pada blok penyeleksian lainnya atau method yang tersarang pada method lainnya.

Inner Class dapat dianalogikan sebagai hubungan antara manusia dan paru-paru. Setiap manusia pasti bernafas dengan menggunakan paru-paru. Dalam hal ini berarti kinerja dari paru-paru turut mendukung/menentukan kinerja dari manusia. Dalam bahasa pemrograman berorientasi objek manusia dapat dikatakan sebagai outer class dan paru-paru sebagai inner class-nya.



## ***2. Apakah yang Anda ketahui tentang Wrapper?***

Java mengenal 8 buah tipe data primitif dan tidak dapat dibuat objek. Karena hal ini mengalami kesulitan dalam penggunaannya karena beberapa class di library Java hanya dapat berinteraksi dengan objek. Untuk mengatasi masalah ini, Java menyediakan tipe data class untuk tipe data primitif. Class ini membungkus tipe data primitif agar dapat digunakan sebagai objek. Class ini disebut dengan type Wrapper.

Tipe Data Primitif	Class Wrapper	Argument Constructor
Boolean	Boolean	boolean / String
Byte	Byte	byte / String
Char	Character	char
Double	Double	double / String
Float	Float	float / double / String
Int	Integer	int / String
Long	Long	long / String
Short	Short	short / String

Setiap data primitif dalam Java memiliki Wrapper Class yang bersesuaian. Sebagai contoh, Wrapper class untuk **int** adalah **Integer**, untuk **float** adalah **Float**, dan sebagainya. Yang patut diingat adalah nama dari Wrapper Class suatu data primitif adalah sama dengan tipe data primitif yang bersesuaian dengan huruf pertamanya dalam bentuk huruf besar, kecuali untuk tipe data **char** dengan Wrapper Class **Character**, dan **int** dengan Wrapper Class **Integer**.

### 3. *Apakah yang Anda ketahui tentang Enum?*

Sebelum J2SE 5.0, solusi untuk menangani masalah sekumpulan nilai konstanta, seperti di bawah ini:

Jenis credit card yang bisa diterima oleh aplikasi

- `public static final int VISA = 1;`
- `public static final int MASTER_CARD = 2;`
- `public static final int AMERICAN_EXPRESS = 3;`

Permasalahan yang muncul adalah tidak ada yang mengikat ketiga nilai menjadi semacam set dan kita bisa memberikan nilai yang salah pada variabel-variabel tersebut. Kondisi seperti ini disebut not type safe. Kondisi ini dapat diperbaiki dengan membuat tipe yang relatif aman dengan mendefinisikan suatu class, yaitu class *AllowedCreditCard*. Class tersebut mendefinisikan:

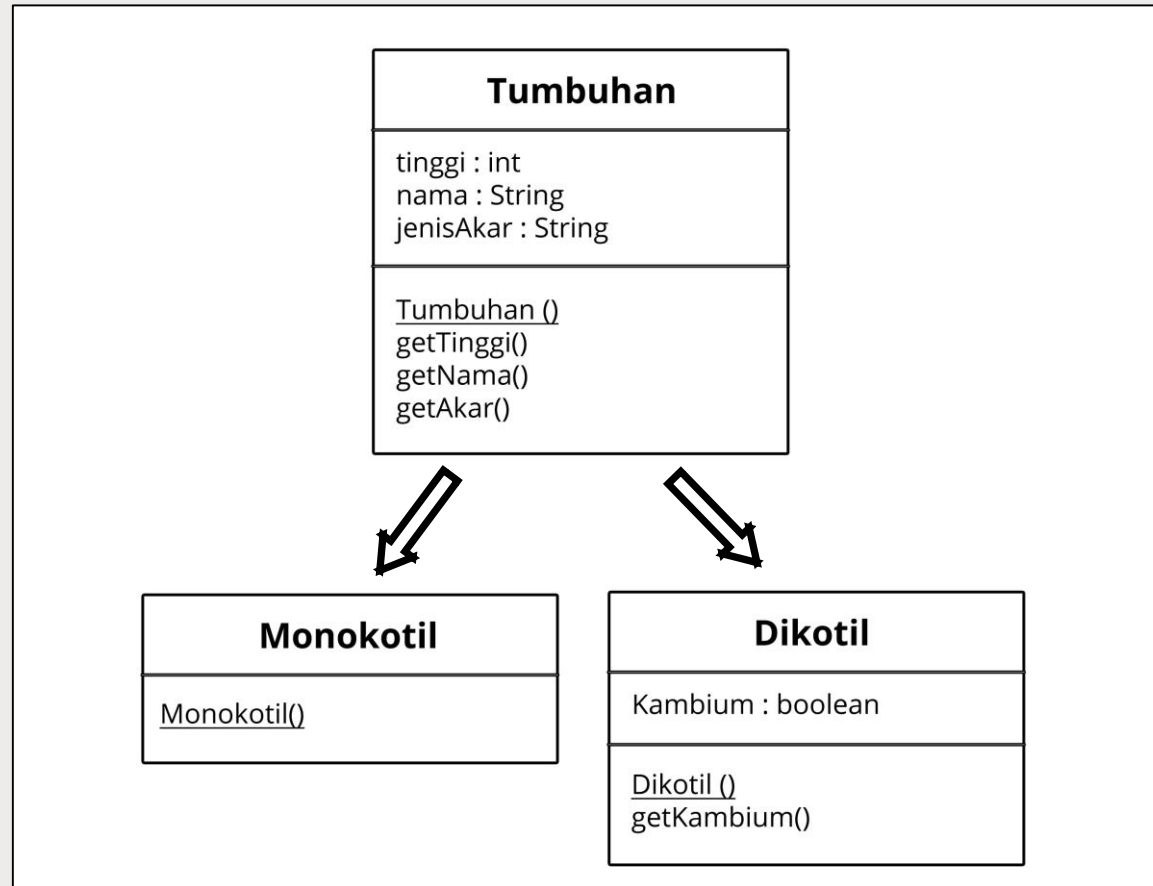
- Konstanta-konstanta di dalam kelas
- Variabel untuk menyatakan state object dari class tersebut.
- Constructor private untuk mengeset state.

Pada pendekatan ini, terdapat 3 state dari class *AllowedCreditCard* yang dinyatakan oleh tiga object yang dicreate dengan memberikan nilai yang berbeda pada variabel card. Karena konstruktor private, sehingga tidak bisa create object di luar class. Design seperti ini masih dianggap type safe. Tapi pada J2SE 5.0 terdapat solusi yang bagus dengan mengenalkan tipe baru yaitu **enum**. Tipe data enum dikenalkan di J2SE 5.0 berguna untuk variabel yang berisi sekumpulan nilai.

Cara mendefinisikan variabel enum:

- Mendefinisikan tipe enum dengan sekumpulan nilai.
- Mendefinisikan variabel yang menyimpan satu dari nilai-nilai tersebut.

# UML DIAGRAM



# UML TO JAVA

```
public class Tumbuhan {  
    int tinggi;  
    String nama;  
    String jenisAkar;  
    public Tumbuhan(int tinggi, String nama){  
        this.tinggi = tinggi;  
        this.nama = nama;  
    }  
    public String getAkar(){  
        return jenisAkar;  
    }  
    public String getNama(){  
        return nama;  
    }  
    public int getTinggi(){  
        return tinggi;  
    }  
}
```

```
class Monokotil extends Tumbuhan {  
    public Monokotil(int tinggi, String nama){  
        super(tinggi,nama);  
    }  
}
```

```
class Dikotil extends Tumbuhan {  
    boolean kambium;  
    public Dikotil(int tinggi, String nama, boolean kambium){  
        super(tinggi,nama);  
        this.kambium = kambium;  
    }  
    public boolean getKambium(){  
        return kambium;  
    }  
}
```



# TEST JAVA

```
class TesTumbuhan {  
    public static String cariAkar(Tumbuhan tbh){  
        String jenisAkar = tbh.getAkar();  
        if(tbh instanceof Dikotil) jenisAkar = "Tunggang";  
        else jenisAkar = "Serabut";  
        return jenisAkar;  
    }  
  
    public static void main(String[] args){  
  
        Monokotil q1 = new Monokotil(20, "Kelapa");  
        Dikotil q2 = new Dikotil(12, "Apel", true);  
        System.out.println("Objek \t| Tinggi | Nama \t| Kambium | Akar");  
        System.out.println("-----");  
        System.out.print("O1 \t| " + q1.getTinggi() + " \t| " + q1.getNama() + " \t| ---- | ");  
        System.out.println(cariAkar(q1));  
        System.out.print("O2 \t| " + q2.getTinggi() + " \t| " + q2.getNama() + " \t| " + q2.getKambium() + " | ");  
        System.out.println(cariAkar(q2));  
    }  
}
```

# COMPILING JAVA

```
E:\DOCS\task.bbr\Programming\Java>javac TestTumbuhan.java
```

```
E:\DOCS\task.bbr\Programming\Java>java TestTumbuhan
```

Objek		Tinggi		Nama		Kambium		Akar
01		20		Kelapa		-----		Serabut
02		12		Apel		true		Tunggang

THANK YOU...