

# **Tugas - Inheritance**

## **Pertemuan ke 5**

Diajukan untuk memenuhi salah satu tugas praktikum  
Mata kuliah PBO



**Disusun Oleh:**  
**Alqan Nazra (231511068)**

**Jurusan Teknik Komputer dan Informatika**

**Program Studi D-3 Teknik Informatika**  
**Politeknik Negeri Bandung**  
**2024**

## Judul : Tugas - Inheritance

Hari, Tanggal : 23 September 2024

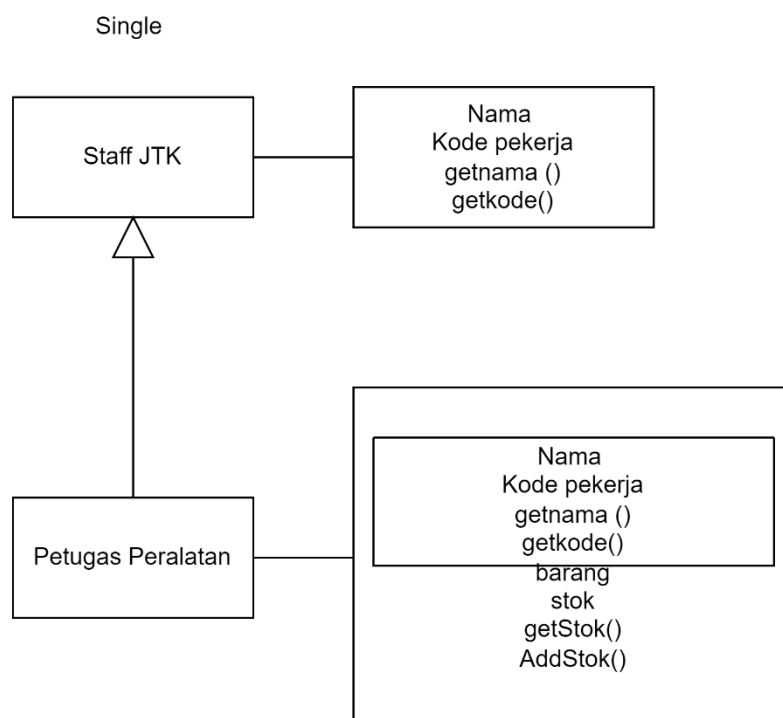
### SOAL

Silahkan jawab beberapa pertanyaan berikut ini:

1. Buatlah contoh hirarki yang menerapkan single inheritance (sesuaikan dengan kasus anda di matkul PRPL) ?
2. Buatlah contoh hirarki yang menerapkan multiple level single inheritance (sesuaikan dengan kasus anda di matkul PRPL)?
3. Buatlah contoh hirarki yang menerapkan multiple inheritance (jangan menggunakan contoh yang sudah diberikan) ?
4. Jelaskan mengapa inheritance dapat meningkatkan aspek "reuse" ?
5. Bagaimana pendapat anda terhadap banyaknya level hirarki pada inheritance ?
6. Apakah permasalahan yang muncul pada kasus multiple inheritance pada Java ?
7. Bagaimana solusi yang mungkin dilakukan untuk menangani multiple inheritance pada Java ?

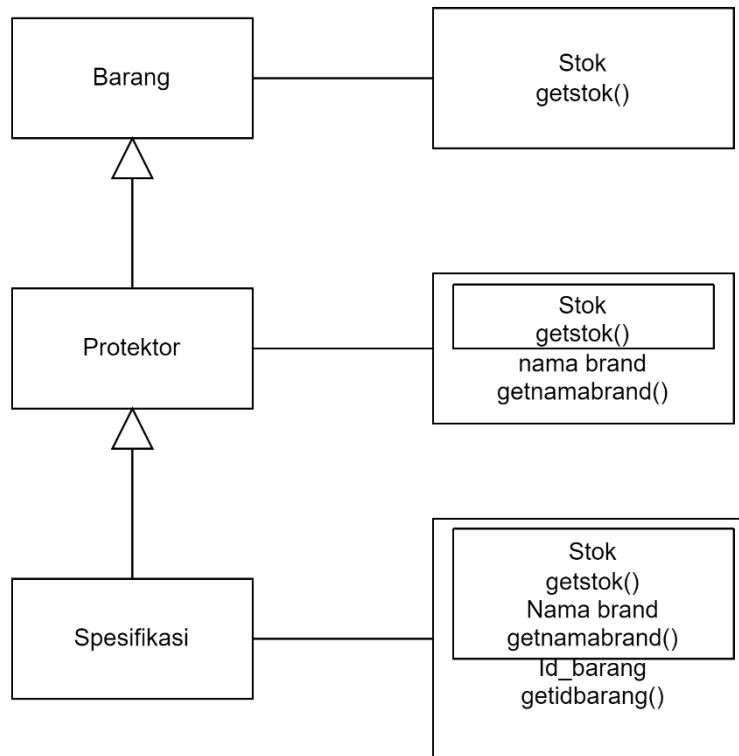
Jawaban

1. Contoh Hirarki single inheritance dengan kasus Petugas

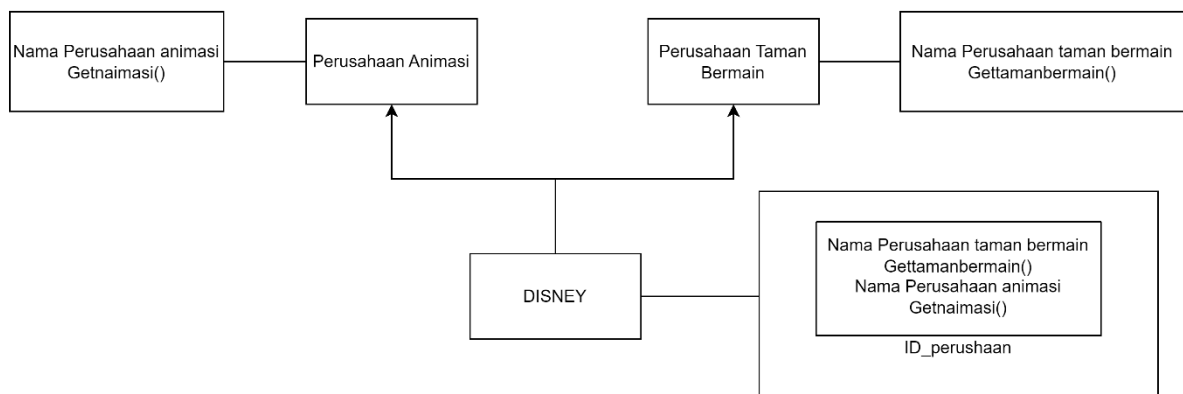


## 2. Contoh Multiple Inheritance Kasus Pemimjaman barang

### multiple level single inheritance



## 3. Contoh multiple inheritance kasus Perusahaan Disney



- Penyebab inheritance memiliki sifat penurunan dari superclass dan subclass sifat penurunan ini menurunkan seluruh objek dari superclass menuju subclass sehingga pada subclass memiliki method dan objek yang dimiliki oleh superclass, kelebihan ini membuat inheritance reuse karena method yang dibutuhkan oleh subclass tidak perlu dibuat Kembali jika memiliki proses yang sama dan output yang sama, membuat method yang dimiliki oleh superclass dapat digunakan kembali pada subclass dan seterusnya.
- Dengan banyaknya hirarki dapat memudahkan dalam mengembangkan sebuah program dan mengurangi urangan dalam aplikasi, namun bila level pada hirarki sudah terlalu banyak maka diperlukan spesifikasi dalam penentuan pewarisan ini karena semakin banyak level yang dimiliki pada suatu penurunan maka semakin kurang

efektif, karena penurunan memiliki tujuan untuk penggolongan pada kelas agar memudahkan proses pembuatan method dan object tanpa perlu membuat lagi dengan banyaknya level semisalnya 10 level maka dalam bentuk program akan tidak terlalu terasa namun dalam konteks pengembangan maka perlu waktu lama untuk program tersebut untuk dikembangkan karena skema pemetaan kelasnya sudah terlalu besar

6. Permasalahan multiple pewarisan merupakan pada syarat pewarisan dimana subclass hanya bisa memiliki 1 super class namun dalam kasus pewarisan ganda subclass memiliki 2 super class yang membuatnya pada proses pemanggilan pada saat extended menuliskan 2 superclass namun komputer hanya bisa membaca hanya satu,
7. Solusi pada pewarisan ganda ini merupakan interface dimana interface ini berbeda dengan class, interface adalah blueprint yang mendefinisikan serangkaian metode yang harus diimplementasikan oleh kelas yang menggunakan atau mewarisinya, interface hanya bisa menjalankan metode saja tidak bisa menjalankan properti sehingga pada saat ingin menggunakan interface pastikan mengambil metode-metodenya saja namun bila diperlukan sebuah variable untuk dapat menjalankan metode maka bisa menggunakan variable yang konstan (Tetap atau final), berikut contoh dari interface

```
interface Kendaraan {  
    void start();  
    void stop();  
}
```

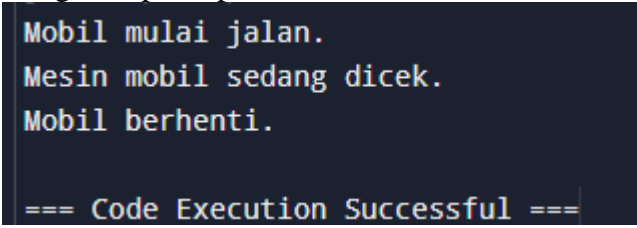
```
interface Mesin {  
    void cekMesin();  
}
```

```
class Mobil implements Kendaraan, Mesin {  
    @Override  
    public void start() {  
        System.out.println("Mobil mulai jalan.");  
    }  
  
    @Override  
    public void stop() {  
        System.out.println("Mobil berhenti.");  
    }  
  
    @Override  
    public void cekMesin() {  
        System.out.println("Mesin mobil sedang dicek.");  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Mobil mobilSaya = new Mobil();  
        mobilSaya.start();  
        mobilSaya.cekMesin();  
    }  
}
```

```
        mobilSaya.stop();  
    }  
}
```

Dengan output seperti ini



```
Mobil mulai jalan.  
Mesin mobil sedang dicek.  
Mobil berhenti.  
  
=== Code Execution Successful ===
```

walaupun tidak bisa menyelesaikan masalah multiple ini dengan membuat properti yang akan digunakan pada interface sehingga masalah ini dapat diselesaikan secara maksimal.