

PEMROGRAMAN BERORIENTASI OBYEK

Dosen Pengampu : Bpk. Andi Iwan Nurhidayat, S.Kom., M.T.



DISUSUN OLEH :

MUHAMMAD LADZI SAFRONI 19051397074

D4 Manajemen Informatika B

**UNIVERSITAS NEGERI SURABAYA
UNESA**

JURUSAN TEKNIK INFORMATIKA

Jl. Ketintang, Ketintang, Kec. Gayungan, Kota Surabaya, Jawa Timur 60231, Telp. (031) 8280009

SOAL

1. Tulislah dan jelaskan perbedaan antara Abstract Class dan Interface !

JAWABAN

1. Sebelum menulis perbedaan antara Abstract Class dan Interface, alangkah baiknya kita mengerti terlebih dahulu definisi tentang Abstract Class dan definisi tentang Interface.

- **Definisi Abstract Class dan Interface**

- a. **Abstract Class**

- Adalah sebuah class yang tidak bisa di-instansiasi (tidak bisa dibuat menjadi objek) dan berperan sebagai 'kerangka dasar' bagi class turunannya. Di dalam abstract class umumnya akan memiliki abstract method.

- Abstract class digunakan di dalam inheritance (pewarisan class) untuk 'memaksakan' implementasi method yang sama bagi seluruh class yang diturunkan dari abstract class. Abstract class digunakan untuk membuat struktur logika penurunan di dalam pemrograman objek.

- b. **Interface**

- Adalah wadah dari kumpulan method yang bersifat abstrak atau tidak memiliki implementasi. Sedangkan method yang didefinisikan di dalam interface tersebut akan diimplementasikan oleh class yang mengimplementasikan interface tersebut. Interface merupakan bentuk perluasan dari kelas abstrak.

- Selain method, interface juga dapat berisi sekumpulan variable, namun variable yang dideklarasikan di dalam interface harus bersifat final(nilainya tidak dapat diubah /konstan).

- Sebagai contoh : Dalam kehidupan nyata dapat diketahui ada manusia yang bekerja sebagai tentara, penyanyi, pengacara, dan sebagainya, tentunya manusia-manusia tersebut selain harus memiliki method standard sebagai seorang manusia, juga harus memiliki method yang sesuai dengan pekerjaannya. Dengan demikian untuk membuat objek manusia yang bekerja sebagai penyanyi, harus dibuat kelas yang merupakan turunan kelas manusia yang meng-implementasikan interface penyanyi.

- **Ciri – Ciri dan Beberapa aturan dalam penulisan Abstract Class dan Interface**

- a. Abstract Class**

- 1) Jika sebuah kelas memiliki method abstrak maka kelas itu harus menjadi kelas abstrak.
- 2) Sebuah kelas abstrak dapat saja memiliki method yang tidak abstrak.
- 3) Jika sebuah kelas abstrak diturunkan menjadi kelas konkret, maka semua method abstrak dari kelas abstrak haruslah ditulis ulang / dibuat ulang di kelas konkretnya dan diberi detail dari methodnya.
- 4) Jika method abstrak di turunkan dan kelas turunannya adalah kelas abstrak, maka tidak perlu menulis ulang method yang abstrak.

- b. Interface**

Method interface tidak punya tubuh, sebuah interface hanya dapat mendefinisikan konstanta dan interface tidak langsung mewariskan hubungan dengan class istimewa lainnya, mereka didefinisikan secara independent. Tidak bisa membuat instance atau objek baru dari sebuah interface. Ciri umum lain adalah baik interface maupun class dapat mendefinisikan method. Bagaimanapun, sebuah interface tidak punya sebuah kode implementasi sedangkan class memiliki salah satunya.

- **Perbedaan Abstract Class dan Interface**

Abstract Class dikatakan Abstract karena class tersebut tidak diimplementasikan secara langsung di class tersebut. Melainkan diimplementasikan di class lain yang merupakan class anak dari abstract class itu. Itulah sebabnya dinamakan abstract class. Berikut perbedaan antara class abstract dengan interface :

1. Abstract Class dapat diimpelentasikan dengan pewarisan (inheritance), Sedangkan Interface Class menggunakan implementasi.
2. Abstract Class memiliki member private dan protected, Sedangkan Interface Class harus publik semua.
3. Abstract Class kita bisa implementasikan kode method seperti class biasa, Sedangkan di Interface Class harus menggunakan default.
4. Abstract Class kita bisa membuat properti atau variabel, Sedangkan di Interface Class kita hanya bisa membuat konstanta saja.
5. Abstract Class itu seperti kelas biasa tetapi ada method Abstractnya (minimal 1), Sedangkan Interface Class lebih seperti rancangannya.
6. Abstract class boleh terdiri dari konstruktor dan destruktur, sedangkan Interface tidak boleh terdiri dari konstruktor dan destruktur.

- **Contoh Abstract Class :**

```
abstract class Robot {
    String nama;
    String pemilik;
    int tahun;

    public void setTahunPembuatan(int tahun){
        this.tahun = tahun;
    }

    public void setPemilik(String Pemilik){
        pemilik = Pemilik;
    }

    public abstract void setName(String nama);

    public void displayData(){
        System.out.println("Nama Robot      : "+nama);
        System.out.println("Pemilik Robot   : "+pemilik);
        System.out.println("Tahun        : "+tahun);
    }
}

class DoraMini extends Robot{

    public void sayDora(){
        System.out.println("Halo, Nama Saya Dora Mini.");
    }

    @Override
    public void setName(String nama) {
        this.nama = nama;
    }
}
```

Dari contoh di atas dapat kita lihat bahwa pada abstract class *Mahluk hidup* terdapat 1 method yang tidak memiliki implementasi. Implementasi method bernafas nanti akan diimplementasikan di class anak karena biasanya class abstract dapat menjadi class induk.

- **Contoh Interface Class :**

```
interface Robot {  
  
    public void setName(String nama);  
    public void setTahunPembuatan(int tahun);  
    public void displayData();  
}  
  
interface Doraemon {  
    public void sayDora();  
    public void displayKantongAjaib();  
}  
  
class DoraMini implements Robot, Doraemon{  
    String nama;  
    String pemilik;  
    int tahun;  
  
    public String getPemilik(){  
        return pemilik;  
    }  
}
```

Pada contoh interface class di atas dapat dilihat perbedaan antara abstract class dan interface class. Interface class memiliki method yang tidak memiliki implementasi.

- Terdapat perbedaan yang lainnya, sebagai berikut :

ABSTRACT CLASS	INTERFACE
Bisa berisi abstract dan non- abstract method.	Hanya boleh berisi abstract method.
Kita harus menuliskan sendiri modifiernya.	Kita tidak perlu susah-susah menulis public abstract di depan nama method. Karena secara implisit, modifier untuk method di interface adalah public dan abstract.
Bisa mendeklarasikan constant dan instance variable.	Hanya bisa mendeklarasikan constant. Secara implisit variable yang dideklarasikan di interface bersifat public, static, dan final.
Method boleh bersifat static.	Method tidak boleh bersifat static.
Method boleh bersifat final.	Method tidak boleh bersifat final.
Suatu abstract class hanya bisa meng-extend satu abstract class lainnya.	Suatu interface bisa meng-extend satu atau lebih interface lainnya.
Suatu abstract class hanya bisa meng-extend satu abstract class dan meng-impliment beberapa interface.	Suatu interface hanya bisa meng-extend interface lainnya. Dan tidak bisa meng-impliment class atau interface lainnya.