

TUGAS INTERFACE



Oleh :

Siti Delfi Nurhuda

15 111 099

TIF RM 15 B

SEKOLAH TINGGI TEKNOLOGI BANDUNG

TEKNIK INFORMATIKA

2018

Pengertian interface

Interface adalah jenis khusus dari blok yang hanya berisi method signature (atau constant). Interface mendefinisikan sebuah(signature) dari sebuah kumpulan method tanpa tubuh. Interface mendefinisikan sebuah cara standar dan umum dalam menetapkan sifat-sifat dari class-class. Mereka menyediakan class-class, tanpa memperhatikan lokasinya dalam hirarki class, untuk mengimplementasikan sifat-sifat yang umum. Dengan catatan bahwa interface-interface juga menunjukkan polimorfisme, dikarenakan program dapat memanggil method interface dan versi yang tepat dari method yang akan dieksekusi tergantung dari tipe object yang melewati pemanggil method interface.

Perbedaan class Abstract dan interface

Salah satu yang membedakan class Abstract dengan Interface , yaitu ketika sebuah class dapat mengimplementasikan atau mewariskan lebih dari satu interface , ini biasa di sebut *Multiple inheritance* . Interface tidak dapat di instansiasi sama seperti class Abstract, dan juga method di Interface harus di set modifier ke **public**. Sebuah class yang mengimplementasi Interface , wajib hukumnya untuk mengimplementasi method-method yang tersedia di Interface. Interface juga menganut konsep menganut konsep Abstraksi.

Pengertian abstraksi

Abstraksi adalah proses menyembunyikan detail implementasi , dan hanya menampilkan fungsionalitas kepada pengguna. Jadi user atau pengguna tidak tahu-menahu proses pembuatannya , hanya secara fungsional kita dapat menggunakannya. Sebagai contoh kasus , saat kita mengetik text ,lalu mengirim SMS. Kalian tidak tahu detail , proses pengirimannya, yang kita tahu SMS ,tiba-tiba sudah terkirim atau tidak. Pada artikel belajar java kali ini ,kita akan mengenal contoh penggunaan class Interface di bahasa pemrograman Java.

Ciri-ciri dari interface adalah sebagai berikut..

- 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.

```
1 public interface [InterfaceName]
2 {
3     //beberapa method tanpa isi
4 }
```

Seperti penggunaan interface pada salah satu class interface berikut :

```
1 public class MyRemoteAllinOne {
2     public void sendRequest(int aksi, RemoteAllinOne rmt, int tombol) {
3         switch(aksi) {
4             case 1:
5                 rmt.PindahChannel(tombol);
6                 break;
7             case 2:
8                 rmt.perbesarvolume(tombol);
9                 break;
10            case 2:
11                rmt.perkecilvolume(tombol);
12                break;
13            case 2:
14                rmt.perbesarzoom(tombol);
15                break;
16            case 2:
17                rmt.perkecilzoom(tombol);
18                break;
19        }
20    }
21 }
22 }
```

Setelah itu kita menggunakan class yang akan mengimplementasi setiap method yang ada pada class interface RemoteAllInOne.

```
1 public class LG {
2     String[] menuTV = {"RCTI", "NET", "TRANSTV", "TRANS7", "MNCTV", "GTV", "SCTV"};
3
4     public void PindahChannel(int channel) {
5         System.out.println("LCD LG Pindah Channel ke - " + this.menuTV[channel]);
6     }
7
8     public void perbesarvolume(int kapasitas) {
9         System.out.println("LCD LG Perbesar Volume : " + kapasitas);
10    }
11
12    public void perkecilvolume(int kapasitas) {
13        System.out.println("LCD LG Perkecil Volume : " + kapasitas);
14    }
15
16    public void perbesarzoom(int kapasitas) {
17        System.out.println("LCD LG Perbesar Zoom : " + kapasitas);
18    }
19
20    public void perkecilzoom(int kapasitas) {
21        System.out.println("LCD LG Perkecil Zoom : " + kapasitas);
22    }
23 }
```

Setelah itu kita membuat class yang berfungsi sebagai perantara pemanggil method sebelum kita masuk ke class main, atau pembuatan instance baru sebuah objek supaya dapat digunakan setiap interface yang dibuat tadi.

```
1 public class MyRemoteAllinOne {
2     public void sendRequest(int aksi, RemoteAllinOne rmt, int tombol) {
3         switch(aksi) {
4             case 1:
5                 rmt.PindahChannel(tombol);
6                 break;
7             case 2:
8                 rmt.perbesarvolume(tombol);
9                 break;
10            case 2:
11                rmt.perkecilvolume(tombol);
12                break;
13            case 2:
14                rmt.perbesarzoom(tombol);
15                break;
16            case 2:
17                rmt.perkecilzoom(tombol);
18                break;
19        }
20    }
21 }
22 }
```

Dan berikut adalah class main yang dipakai waktu pengekseskuan program.

```
1 public class Main {
2
3     public static void main(String[] args) {
4         Ephon ep = new Ephon();
5         LG ll = new LG();
6         Mitsubishi mt = new Mitsubishi();
7         MyRemoteAllinOne my = MyRemoteAllinOne();
8         my.sendrequest(1, mt, 4);
9         my.sendrequest(1, ep, 5);
10        my.sendrequest(1, ll, 6);
11        my.sendrequest(1, mt, 4);
12    }
13 }
14 }
```