



**IPB University**  
— Bogor Indonesia —

KOM120C -- PEMROGRAMAN

# Object Oriented Programming

- Java Interface
- Java Collections Framework

---

Tim Pengajar Pemrograman IPB University

# Interface

Interface dalam Java didefinisikan sebagai tipe abstrak yang digunakan untuk menentukan *behaviour* dari suatu class.

Java interface berisi *static constants* dan *abstract methods* (*pure virtual function* dalam C++).

Interface tidak memiliki *constructor*.

Sintaks:

```
interface <interface-identifier> {  
    // declare constant fields  
    // declare abstract methods  
}
```

# Class vs Interface

## Class

Dengan class, kita DAPAT menginstansiasi variabel dan membuat objek.

Class dapat berisi method yang kongkrit (memiliki implementasi).

Access specifier anggota class adalah private, protected, dan public.

## Interface

Interface, kita TIDAK DAPAT menginstansiasi variabel atau membuat objek.

Interface TIDAK berisi method yang kongkrit. Interface dapat berisi abstract method.

Access specifier anggota interface hanya public.

# Implementasi Interface

Implementasi interface dalam Java menggunakan keyword **implements**

Contoh kasus: objek kendaraan (motor dan mobil) memiliki fungsi yang umum, yaitu (1) mengubah "gigi", dan (2) menambah/mengurangi kecepatan.

```
interface Kendaraan {  
    // abstract methods.  
    void ubahGigi(int a);  
    void tambahKecepatan(int a);  
    void kurangiKecepatan(int a);  
}
```

Definisi class Mobil mirip dengan definisi class Motor.

```
class Motor implements Kendaraan {  
    int kecepatan;  
    int gigi;  
  
    @Override  
    public void ubahGigi(int g)  
    { gigi = g; }  
  
    @Override  
    public void tambahKecepatan (int s)  
    { kecepatan = kecepatan + s; }  
  
    @Override  
    public void kurangiKecepatan (int s)  
    { kecepatan = kecepatan - s; }  
  
    public void show() {  
        System.out.println("kecepatan: " +  
            kecepatan + " gigi: " + gigi); }  
}
```

# Java Collections Framework (JCF)

Mirip dengan STL (*Standard Template Library*) dalam C++.

`std::set`

`std::unordered_set`

`std::vector`

`std::map`

Kita dapat meneruskan (pass) fungsi sebagai parameter.

C++ tidak mengenal interface.

`java.util.HashSet`

`java.util.LinkedHashSet`

`java.util.ArrayList`

`java.util.TreeMap`

Tidak dapat.

Tergantung pada Interface.

# Java Set

## Definisi:

```
// Obj adalah tipe yang akan disimpan ke dalam Set
Set<Obj> set = new HashSet<Obj>();
```

```
import java.util.*;
public class ContohSet {
    public static void main(String[] args) {
        Set<String> hs = new HashSet<String>();

        hs.add("B");
        hs.add("A");
        hs.add("B");
        hs.add("D");
        hs.add("C");

        System.out.println(hash_Set);
    }
}
```

Output:

[A, B, C, D]

# Set of Class

Contoh penggunaan Set untuk mengelola koleksi buku.

```
import java.util.*;

class Buku {
    int id;
    String judul, pengarang, penerbit;
    int jumlah;
    public Buku(int id, String judul, String pengarang,
                String penerbit, int jumlah)
    {
        this.id = id;
        this.judul = judul;
        this.pengarang = pengarang;
        this.penerbit = penerbit;
        this.jumlah = jumlah;
    }
}
```

# Set of Class

Contoh penggunaan Set untuk mengelola koleksi buku.

```
public class Koleksi {  
    public static void main(String[] args) {  
        LinkedHashSet<Buku> kol=new LinkedHashSet<Buku>();  
  
        Buku b1=new Buku(101,"Programming","Allan B Tucker","CSB",8);  
        Buku b2=new Buku(102,"Data Communications","Forouzan","MGH",4);  
        Buku b3=new Buku(103,"Operating System","Galvin","Wiley",6);  
  
        kol.add(b1);  
        kol.add(b2);  
        kol.add(b3);  
  
        for(Buku b:kol) {  
            System.out.println(b.id+" "+b.judul+" "+b.jumlah);  
        }  
    }  
}
```



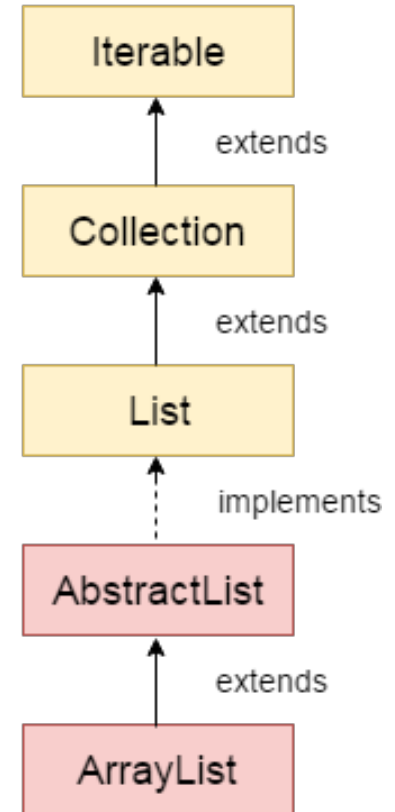
# Java ArrayList

Dynamic array dalam Java

Seperti Vector dalam C++.

Tidak seperti Set, ArrayList dapat berisi elemen yang sama (tidak unik).

Lebih fleksibel dibanding array biasa. Mengapa?



# Contoh ArrayList

```
import java.util.*;
public class ContohArrayList {
    public static void main(String args[]){
        ArrayList<String> list=new ArrayList<String>();
        list.add("Mangga");
        list.add("Apel");
        list.add("Pisang");
        list.add("Jambu");

        // Print
        System.out.println(list);

        Iterator itr=list.iterator();
        while(itr.hasNext()) { // memeriksa iterator memiliki nilai
            System.out.println(itr.next());
        }
    }
}
```

## Output:

```
[Mangga, Apel, Pisang, Jambu]
Mangga
Apel
Pisang
Jambu
```

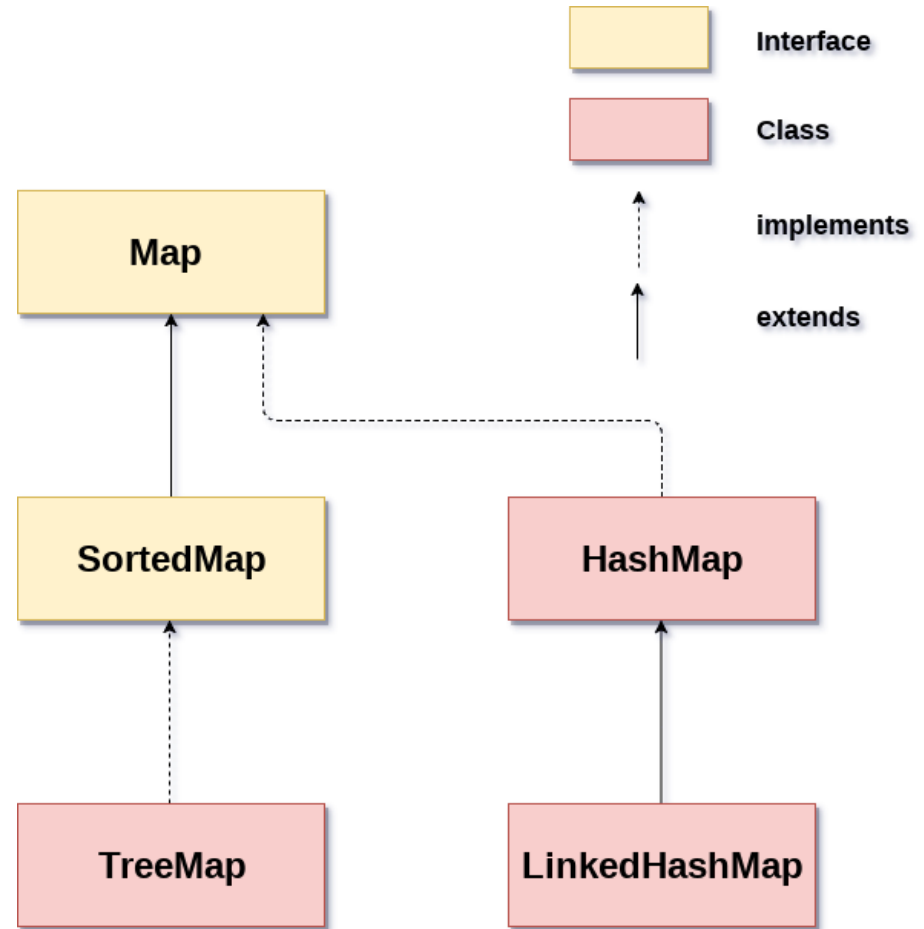
# Java Map

## The hierarchy of Java Map

Seperti Map dalam C++.

Mengandung key (tidak boleh duplikat) dan value.

Lebih fleksibel dibanding array biasa.  
Mengapa?



# Contoh Java Map

```
import java.util.*;
public class ContohMap {
    public static void main(String args[]){
        Map<Integer,String> map=new HashMap<Integer,String>();
        map.put(100,"Amir");
        map.put(102,"Rudi");
        map.put(101,"Shanti");
        map.put(100,"Abdullah");

        for(Map.Entry m:map.entrySet()) {
            System.out.println(m.getKey()+" "+m.getValue());
        }

        // sort by value
        map.entrySet()
            .stream()
            .sorted(Map.Entry.comparingByValue())
            .forEach(System.out::println);
    }
}
```

## Output:

```
100 Abdullah
101 Shanti
102 Rudi
100=Abdullah
102=Rudi
101=Shanti
```

# Latihan

Gunakan Java OOP

## Deskripsi

Buat program untuk membaca beberapa bilangan bulat, dan menghapus beberapa bilangan pada posisi tertentu.

## Format Masukan

Bagian pertama berisi beberapa bilangan bulat yang diakhiri dengan nilai -9 (nilai ini tidak ikut dalam pengolahan, hanya sebagai akhir data masukan).

Bagian kedua adalah beberapa bilangan terurut dari kecil ke besar yang menunjukkan elemen beberapa dari bilangan masukan yang dihapus. Input bagian ini juga diakhiri dengan -9 (tidak ikut diolah)

## Format Keluaran

Baris pertama adalah dua bilangan yang menunjukkan banyaknya bilangan awal dan banyaknya bilangan setelah dihapus beberapa elemennya. Baris kedua adalah dua nilai rata-rata yang dituliskan dalam dua digit di belakang tanda desimal dari bilangan-bilangan awal dan bilangan-bilangan setelah dihapus. Jika data dalam array kosong, maka nilai rata-rata dituliskan -9.99. Output program diakhiri

## Contoh Input

10 20 30 40 50 -9 1 3 4 -9

## Contoh Output

5 2  
30.00 35.00

dengan newline..