

IF34348 - PEMROGRAMAN LANJUT

GENERIC

10

Oleh : Andri Heryandi, M.T.

MATERI HARI INI

IF34348 - Pemrograman Lanjut

- Definisi Generic
- Membuat Class Generic



Oleh : Andri Heryandi, M.T.

CONTOH KASUS

IF34348 - Pemrograman Lanjut

- Anda membutuhkan sebuah class untuk menukarkan isi 2 buah data.
- Contoh class :



Oleh : Andri Heryandi, M.T.

CONTOH KASUS

IF34348 - Pemrograman Lanjut

■ Contoh Class Penukaran int:

```
public class TukarInt {
    private int v1;
    private int v2;
    public TukarInt(int var1, int var2){
        v1=var1;
        v2=var2;
    }
    public void tukarkan(){
        int temp=v1;
        v1=v2;
        v2=temp;
    }
    public int getV1(){ return v1;};
    public int getV2(){ return v2;};
}
```



CONTOH KASUS

IF34348 - Pemrograman Lanjut

■ Contoh Penggunaan Class Penukaran int:

```
public class TukarIntTester {  
    public static void main(String[] args) {  
        int a1=10,a2=15;  
        System.out.println("A1 : "+a1+" A2 : "+a2);  
        TukarInt t=new TukarInt(a1,a2);  
        t.tukarkan();  
        a1=t.getV1();  
        a2=t.getV2();  
        System.out.println("A1 : "+a1+" A2 : "+a2);  
    }  
}
```

Hasil RUN:

```
-----  
A1 : 10 A2 : 15  
A1 : 15 A2 : 10
```



CONTOH KASUS

IF34348 - Pemrograman Lanjut

- Pertanyaannya :
- Apakah class tersebut dapat digunakan untuk menukarkan data dengan tipe data yang berbeda? Jawabannya adalah **TIDAK**. Anda harus membuat class baru dengan menyesuaikan tipe datanya.
- Solusinya adalah : **GENERIC**



GENERIC

IF34348 - Pemrograman Lanjut

- Generic disebut pula sebagai parametered type (tipe berparameter).
- Generic memperbolehkan pemberian tipe (class/interface) sebagai parameter ketika membuat class.
- Dengan menggunakan konsep class generic, maka programmer dapat menghilangkan type-casting sehingga dapat menghasilkan aplikasi yang memiliki penanganan data yang baik.
- Generic juga dapat diterapkan dalam pendeklarasian method (generic method).



Oleh : Andri Heryandi, M.T.

GENERIC CLASS

IF34348 - Pemrograman Lanjut

■ Contoh Class Penukaran versi Generic:

```
public class Penukaran<T> {  
    private T v1;  
    private T v2;  
    public Penukaran(T var1, T var2){  
        v1=var1;  
        v2=var2;  
    }  
    public void tukarkan(){  
        T temp=v1;  
        v1=v2;  
        v2=temp;  
    }  
    public T getV1(){ return v1;};  
    public T getV2(){ return v2;};  
}
```



GENERIC CLASS

IF34348 - Pemrograman Lanjut

■ Contoh Penggunaan Class Penukaran versi Generic:

```
public class PenukaranTester {
    public static void main(String[] args) {
        Integer i1=50,i2=80;
        Double d1=55.5,d2=70.2;
        System.out.println("I1 : "+i1+", I2 : "+i2);
        System.out.println("D1 : "+d1+", D2 : "+d2);
        Penukaran<Integer> t1=new Penukaran<Integer>(i1,i2);
        Penukaran<Double> t2=new Penukaran<Double>(d1,d2);
        t1.tukarkan();
        i1=t1.getV1();
        i2=t1.getV2();
        System.out.println("I1 : "+i1+", I2 : "+i2);
        t2.tukarkan();
        d1=t2.getV1();
        d2=t2.getV2();
        System.out.println("D1 : "+d1+", D2 : "+d2);
    }
}
```

Hasil RUN:

```
-----
I1 : 50, I2 : 80
D1 : 55.5, D2 : 70.2
I1 : 80, I2 : 50
D1 : 70.2, D2 : 55.5
```



GENERIC METHOD

IF34348 - Pemrograman Lanjut

- Contoh penggunaan generic method adalah ketika ingin menggunakan method dengan parameter yang berbeda jenisnya.



GENERIC METHOD

IF34348 - Pemrograman Lanjut

■ Contoh Generic Method:

```
public class KelolaArray {  
    public static <E> void cetakArray(E[] data) {  
        for(E isi:data)  
            System.out.print(isi+" ");  
        System.out.println();  
    }  
}
```



GENERIC METHOD

IF34348 - Pemrograman Lanjut

■ Contoh Penggunaan Generic Method:

```
public class KelolaArrayTester {  
    public static void main(String[] args) {  
        Integer[] i={5,3,7,8,3};  
        Double[] d={5.7,2.3,5.7,9.8,3.5};  
        Character[] c={'U','N','I','K','O','M'};  
        String[] teman={"Aang","Budi","Cuplis","Dede"};  
        KelolaArray.cetakArray(i);  
        KelolaArray.cetakArray(d);  
        KelolaArray.cetakArray(c);  
        KelolaArray.cetakArray(teman);  
    }  
}
```

Hasil RUN:

```
-----  
5 3 7 8 3  
5.7 2.3 5.7 9.8 3.5  
U N I K O M  
Aang Budi Cuplis Dede
```

Oleh : Andri Heryandi, M.T.



GENERIC

IF34348 - Pemrograman Lanjut

- Contoh umum penggunaan class generic adalah class untuk merepresentasikan struktur data, contoh :
 - Array
 - LinkedList

Struktur-struktur data di atas bisa dibuat generic sehingga struktur datanya tidak hanya bisa digunakan untuk satu jenis tipe data saja tetapi bisa berisi berbagai macam tipe.

Contoh : Sebuah LinkedList generic bisa digunakan untuk menampung data Angka (Number) atau String atau Class-class lain.



JAVA COLLECTION

IF34348 - Pemrograman Lanjut

- Java sudah menyediakan class/interface untuk menampung struktur data dengan menggunakan konsep Generic.
- Class-class tersebut disebut sebagai **Java Collection**.
- Beberapa class generic yang merupakan class-class Java Collection :
 - **ArrayList** : Class generic yang dapat digunakan untuk menampung data dalam bentuk array yang dinamis.
 - **LinkedList** : Class generic yang dapat digunakan untuk menampung data dalam bentuk linked list.
 - Class lain dapat dilihat di referensi javadoc.



ARRAYLIST

IF34348 - Pemrograman Lanjut

- **ArrayList** adalah class yang dapat digunakan untuk menyimpan objek dalam bentuk array yang dinamis.
- **Method yang ada diantaranya :**
 - **add()** : menambah elemen diakhir
 - **remove()** : menghapus elemen tertentu. Jika parameter berisi angka (int) maka akan menghapus data posisi tersebut. Posisi awal adalah 0.
 - **get()** : mereturnkan data pada posisi index tertentu. Paramter fungsi ini adalah indeksinya.
 - **clear()** : mengosongkan array
 - **size()** : mereturnkan banyaknya elemen yang ada dalam array



ARRAYLIST

IF34348 - Pemrograman Lanjut

```
import java.util.ArrayList;
public class ArrayListTester {
    public static void main(String[] args) {
        ArrayList<Integer> ai=new ArrayList<Integer>();
        ai.add(30);
        ai.add(50);
        ai.add(70);
        for(int i=0;i<ai.size();i++){
            System.out.println(ai.get(i));
        }
        ai.remove(1);// hapus data ke-2
        System.out.println(ai);
        ai.clear();// kosongkan array
        System.out.println(ai);
    }
}
```

Hasil RUN:

```
30
50
70
[30, 70]
[]
```

Oleh : Andri Heryandi, M.T.



LINKEDLIST

IF34348 - Pemrograman Lanjut

- **LinkedList** adalah class yang dapat digunakan untuk menyimpan objek dalam bentuk linked list.
- **Method** yang ada diantaranya :
 - **add()** : menambah elemen diakhir
 - **remove()** : menghapus pertama. Jika ada parameter berisi angka (int) maka akan menghapus data posisi tersebut. Posisi awal adalah 0.
 - **get()** : mereturnkan data pada posisi index tertentu. Paramter fungsi ini adalah indeksny.
 - **clear()** : mengosongkan array
 - **size()** : mereturnkan banyaknya elemen yang ada dalam array
 - **addFirst()** : menambah elemen baru di depan
 - **addLast()** : menambah elemen baru di belakang
 - **removeFirst()** : menghapus elemen di depan
 - **removeLast()** : menghapus elemen di belakang



LINKEDLIST

IF34348 - Pemrograman Lanjut

```
import java.util.LinkedList;
public class LinkedListTester {
    public static void main(String[] args) {
        LinkedList<String> ll = new LinkedList<String>();
        ll.add("B");
        ll.add("C");
        ll.add("D");
        ll.add("E");
        System.out.println(ll);
        ll.addLast("Z");
        ll.addFirst("A");
        System.out.println(ll);
        String a=ll.removeFirst();
        System.out.println("Pertama : "+a);
        System.out.println("Kedua : "+ll.get(1));
        ll.clear();
        System.out.println(ll);
    }
}
```

Hasil RUN:

```
-----
[B, C, D, E]
[A, B, C, D, E, Z]
Pertama : A
Kedua : C
[]
```

Oleh : Andri Heryandi, M.T.



LATIHAN

IF34348 - Pemrograman Lanjut

- **Buatlah program yang menggunakan class Queue dan Stack yang merupakan bawaandari Java (collection).**
- **Referensi, lihat Javadoc.**



Oleh : Andri Heryandi, M.T.