

Pemrograman Berorientasi Objek

Generic Programming



**Object-Oriented
Programming:**
The Basic Building Blocks



Adam Mukharil Bachtiar
Teknik Informatika UNIKOM





Generic Programming, Collection, dan Multithreading

1. Definisi Generic Programming
2. Penjelasan Generic Programming
3. Kelas Object
4. Generic Class
5. Generic Method
6. Pewarisan Generic

Definisi Generic Programming

Konsep pemrograman yang memungkinkan penggunaan class (pembentukan objek) dengan tipe data yang berbeda-beda.

Kenapa Generic Programming?

```
public class KoleksiString {  
    private String data;  
  
    public String getData() {  
        return data;  
    }  
  
    public void setData(String data) {  
        this.data = data;  
    }  
}
```

```
public class KoleksiInteger {  
    private int data;  
  
    public int getData() {  
        return data;  
    }  
  
    public void setData(int data) {  
        this.data = data;  
    }  
}
```

Kalau 10 Tipe Data?

Pemecahan Masalah

1. Penggunaan **Object**
2. Penggunaan **Generic Programming**

Contoh Penggunaan Object (1)

```
public class Objek {  
    private Object data;  
  
    public Object getData() {  
        return data;  
    }  
  
    public void setData(Object data) {  
        this.data = data;  
    }  
}
```

Contoh Penggunaan Object (2)

```
public class TesterPenggunaanObjek {  
    public static void main(String[] args) {  
        Objek x=new Objek();  
        Integer y=10;  
        x.setData(y);  
        System.out.println(x.getData());  
  
        String z="Halo saya objek";  
        x.setData(z);  
        System.out.println(x.getData());  
    }  
}
```

Format Generic Class

Format Deklarasi:

```
<hak_akses> class Koleksi <T> {  
    private T data;
```

```
    public T getData() {  
        return data;  
    }
```

```
    public void setData(T data) {  
        this.data = data;  
    }
```

```
}
```



Bisa diganti dengan huruf lain

Contoh Generic Class (1)

```
public class Generic <E>{  
    private E data;  
  
    public E getData() {  
        return data;  
    }  
  
    public void setData(E data) {  
        this.data = data;  
    }  
}
```

Contoh Generic Class (2)

```
public class TesterGeneric {  
    public static void main(String[] args) {  
        Generic<Integer> x=new Generic<Integer>();  
        x.setData(10);  
        System.out.println("Data yang diinput adalah : "  
            +x.getData());  
  
        Generic<String> y=new Generic<String>();  
        y.setData("Halo saya generic!");  
        System.out.println("Data yang diinput adalah : "  
            +y.getData());  
    }  
}
```

Definisi Generic Method

Selain digunakan pada level class, generic pun bisa digunakan pada level method.

Format Generic Method

Format Deklarasi:

```
<hak_akses> <T> void <nama_void> (T t) {  
  
}
```

Atau

```
<hak_akses> <T> T<nama_void> (T t) {  
  
}
```

Contoh Generic Method (1)

```
public class GenericMethod <E>{  
    public E data;  
  
    public E getData() {  
        return data;  
    }  
  
    public void setData(E data) {  
        this.data = data;  
    }  
  
    public <U> U tampil(U data){  
        return data;  
    }  
}
```

Contoh Generic Method (2)

```
public class TesterGenericMethod {  
    public static void main(String[] args) {  
        GenericMethod<Integer> x=new GenericMethod<Integer>();  
        x.setData(10);  
        Integer tampil = x.tampil(11);  
        System.out.println(tampil);  
    }  
}
```

Pewarisan Generic Programming

Dalam generic programming berlaku juga hukum pewarisan. Secara default simbol pada Generic Programming merupakan turunan kelas Object, sehingga tipe data apapun dapat masuk ke kode generic tersebut.

THE END