Generic Class

Brigida Arie Minartiningtyas, M.Kom.

Generic

- Tipe generic adalah sebuah kelas generic yang bisa digunakan dalam tipe data apa saja.
- Kita bisa mendefinisikan suatu parameter/variabel baik dalam kelas maupun fungsi dengan tipe data yang bersifat umum atau tidak terikat hanya pada satu tipe data saja.
- Contoh terikat pada satu tipe data:
  - Private String Nama;
  - Private int umur;
  - Pravate float IP;
  - String getTheMostclosetoIP(float IP);
- Generic merupakan cara Java dalam melakukan generalisasi terhadap tipe data tanpa mengurangi kemampuan Java dalam menjaga keamanan penggunaan tipe data

Kegunaan Generic Type

- Prinsip REUSE lah yang ingin digunakan.
- Digunakan agar tidak melakukan code duplication hanya karena tipe data yang berbeda saja
- Contoh
  - Penggunaan fungsi add pada linkedlist
  - Fungsi add and dequeuer/pop pada stack dan queue

NON GENERIC CLASS

```java
public class Box {
    private Object object;

    public void add (Object object){
        this.object = object;
    }

    public Object get(){
        return object;
    }
}
```

Sembarang object dapat dimasukkan karena parameter pada method add adalah Class Object

Tapi pada saat mengambil object tersebut harus diubah sesuai dengan tipe dari object tersebut

integerBox diberikan object Integer

Contoh 1

```java
public class BoxDemo1 {
    public static void main (String [] args){

        Box integerbox = new Box();

        integerbox.add(new Integer (10));
        Integer someInteger = (Integer)integerbox.get();
        System.out.println(someInteger);
    }
}
```

Pada saat mengambil harus diubah menjadi Integer

```
ut - JavaApplication1 (run)
run:
10
BUILD SUCCESSFUL (total time: 0 seconds)
```

Contoh 1

Pada object IntegerBox dimasukkan object 10 tapi dengan tipe String,

```
public class BoxDemo1 {
    public static void main (String [] args){

        Box integerbox = new Box();

        integerbox.add("10");
        Integer someInteger = (Integer)integerbox.get();
        System.out.println(someInteger);
    }
}
```

Tapi pada saat mengambil object, diubah menjadi tipe Integer.

out - JavaApplication1 (run)

```
run:
Exception in thread "main" java.lang.ClassCastException: java.lang.String cannot be cast to java.lang.Integer
        at Generic.BoxDemo1.main(BoxDemo1.java:18)
C:\Users\Tyas\AppData\Local\NetBeans\Cache\8.2\executor-snippets\run.xml:53: Java returned: 1
BUILD FAILED (total time: 0 seconds)
```

Tipe data tidak sesuai sehingga error

Apa yang menyebabkan Error??

Contoh 2

```java
public class NonGen {
    Object ob;
    NonGen(Object o) {
        ob = o; }Object getob() {
        return ob;
        }

        void showType() {
            System.out.println("Type of ob is " + ob.getClass().getName()); }
}
```

```java
public class NonGenDemo {
    public static void main(String args[]) {
        NonGen integerObject = new NonGen(88);

        integerObject.showType();

        int v = (Integer) integerObject.getob();

        System.out.println("value: " + v);

        NonGen strOb = new NonGen("10");
        strOb.showType();

        String str = (String) strOb.getob();
        System.out.println("value: " + str);

        //menyebabkan exception
        Integer i = (Integer) strOb.getob();
    }
}
```

- JavaApplication1 (run)

```
run:
Type of ob is java.lang.Integer
value: 88
Type of ob is java.lang.String
value: 10
Exception in thread "main" java.lang.ClassCastException: java.lang.String cannot be cast to java.lang.Integer
        at Generic.NonGenDemo.main(NonGenDemo.java:29)
C:\Users\Tyas\AppData\Local\NetBeans\Cache\8.2\executor-snippets\run.xml:53: Java returned: 1
BUILD FAILED (total time: 0 seconds)
```

```java
public class NonGen {
    Object ob;
    NonGen(Object o) {
        ob = o; }Object getob() {
        return ob;
        }

        void showType() {
            System.out.println("Type of ob is " + ob.getClass().getName()); }
}
```

Terjadi exception karena pada object strOb dimasukkan object 10 tapi dengan tipe String

Tapi pada saat mengambil object, diubah menjadi tipe Integer

Tipe data tidak sesuai sehingga menyebabkan terjadinya exception

Contoh 2

```java
public class NonGenDemo {
    public static void main(String args[]) {
        NonGen integerObject = new NonGen(88);

        integerObject.showType();

        int v = (Integer) integerObject.getob();

        System.out.println("value: " + v);

        NonGen strOb = new NonGen("10");
        strOb.showType();

        String str = (String) strOb.getob();
        System.out.println("value: " + str);

        //menyebabkan exception
        Integer i = (Integer) strOb.getob();
    }
}
```

- JavaApplication1 (run)

```
run:
Type of ob is java.lang.Integer
value: 88
Type of ob is java.lang.String
value: 10
Exception in thread "main" java.lang.ClassCastException: java.lang.String cannot be cast to java.lang.Integer
        at Generic.NonGenDemo.main(NonGenDemo.java:29)
C:\Users\Tyas\AppData\Local\NetBeans\Cache\8.2\executor-snippets\run.xml:53: Java returned: 1
BUILD FAILED (total time: 0 seconds)
```

Permasalahan?

- No homogeneous collections – memerlukan banyak casting
- Tidak ada pengecekan pada saat kompile, kesalahan baru bisa terdeteksi pada saat runtime.

```
LinkedList list = new LinkedList();
list.add(new Integer(0));
Integer i = (Integer) list.get(0);
String  s = (String)  list.get(0);
```

fine at compile-time, but fails at runtime

casts required

GENERIC CLASS

Cara Mendeklarasikan Class Generic

Ubah class Box Non Generics menjadi class Box Generic.

Pendeklarasian type generics dengan mengubah public class Box() menjadi public class Box <T>

- T biasanya disebut parameter type formal (formal type parameter)
- T adalah type parameter yang akan diganti dengan tipe sebenarnya (Type dari T bisa berupa class, interface atau tipe variabel lainnya).
- T adalah nama dari type parameter

```java
public class Gen<T> {
    T ob; // declare an object of type T

    Gen(T o) { ob = o;
    }

    T getob() { return ob;
    }
    void showType() {
        System.out.println("Type of T is " + ob.getClass().getName()); }
}
```

Objek iOb adalah objek dari class Generic, menggunakan tipe data Integer

T yang ada didefinisikan sebagai tipe data yang tipe nya diserahkan pada kelas lain yang akan membentuknya

Contoh 1

```java
public class GenDemo {
    public static void main(String args[]) {
        Gen<Integer> iOb;
        iOb = new Gen<Integer>(88);
        iOb.showType();

        int v = iOb.getob();
        System.out.println("value: " + v);
        System.out.println();

        Gen<String> strOb = new Gen<String>("Generics Test");

        String v2 = strOb.getob(); }
}
```

sehingga pada saat mengambil objek menggunakan fungsi get(), tidak perlu proses casting

```
ut - JavaApplication1 (run)
run:
Type of T is java.lang.Integer
value: 88

BUILD SUCCESSFUL (total time: 0 seconds)
```

```java
public class Boxku <T>{
    private T t;

    public void add (T t){
        this.t = t;
    }
    public T get(){
        return t;
    }
}
```

Tidak perlu proses casting pada saat menggunakan fungsi get().

```java
public class BoxDemo3 {
    public static void main (String [] args){
        Boxku<Integer> integerBox = new Boxku<Integer>();
        integerBox.add(new Integer(10));
        Integer someInteger = integerBox.get();
        System.out.println(someInteger);

    }
}
```

Contoh 2

ut - JavaApplication1 (run)

```
run:
10
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
12  public class BoxDemo3 {
13      public static void main (String [] args){
            Boxku<Integer> integerBox = new Boxku<Integer>();
            integerBox.add("10");
16          Integer someInteger = integerBox.get();
17          System.out.println(someInteger);
18
19      }
20  }
```

Jika menambahkan ke box dengan tipe yang tidak sesuai (misal : String) maka pada saat di kompile akan mengalami error

Contoh 2

Aturan Penamaan Type Parameter

Nama type parameter biasanya satu huruf dan huruf besar.

Gunakan tanda < > setelah nama kelas untuk mendefinisikan tipe data yang akan dipakai

Gunakan tipe data parameter yang didefinisikan pada tanda tersebut untuk menjadi tipe data baru

Jenis nama tipe parameter yang sering digunakan :

- E - Element (biasanya digunakan untuk Collection Framework)
- K – Key
- N – Number
- T – Type
- V - Value
- S,U,V dll. - 2nd, 3rd, 4th types

```java
public class Gen<T> {
    T ob; // declare an object of type T

    Gen(T o) { ob = o;
    }

    T getob() { return ob;
    }
    void showType() {
        System.out.println("Type of T is " + ob.getClass().getName()); }
}
```

Apa outputnya???

```java
public class GenDemo {
    public static void main(String args[]) {
        Gen<Integer> iOb;
        iOb = new Gen<Integer>(88);
        iOb.showType();

        int v = iOb.getob();
        System.out.println("value: " + v);
        System.out.println();

        Gen<String> strOb = new Gen<String>("Generics Test");
        strOb.showType();

        String v2 = strOb.getob();
        System.out.println("value : "+ v2);
    }
}
```

```java
public class Gen<T> {
    T ob; // declare an object of type T

    Gen(T o) { ob = o;
    }

    T getob() { return ob;
    }
    void showType() {
        System.out.println("Type of T is " + ob.getClass().getName()); }
}
```

Outputnya…

```java
public class GenDemo {
    public static void main(String args[]) {
        Gen<Integer> iOb;
        iOb = new Gen<Integer>(88);
        iOb.showType();

        int v = iOb.getob();
        System.out.println("value: " + v);
        System.out.println();

        Gen<String> strOb = new Gen<String>("Generics Test");
        strOb.showType();

        String v2 = strOb.getob();
        System.out.println("value : "+ v2);
    }
}
```

```
t - JavaApplication1 (run)
run:
Type of T is java.lang.Integer
value: 88

Type of T is java.lang.String
value : Generics Test
BUILD SUCCESSFUL (total time: 0 seconds)
```

GENERIC CLASS DENGAN 2 PARAMETER

```java
public class TwoGen<T,V> {
    T ob1;
    V ob2;

    TwoGen(T o1, V o2){
        ob1 = o1;
        ob2 = o2;
    }

    void showTypes(){
        System.out.println("Type of T = "+ob1.getClass().getName());
        System.out.println("Type of V = "+ob2.getClass().getName());
    }

    T getob1(){
        return ob1;
    }

    V getob2(){
        return ob2;
    }
}
```

Apa outputnya???

```java
public class MainClass {
    public static void main (String [] args){
        TwoGen<Integer, String> tgObj = new TwoGen<Integer, String> (88, "Generic");
        tgObj.showTypes();

        int v = tgObj.getob1();
        System.out.println("value : "+v);

        String str = tgObj.getob2();
        System.out.println("value : "+str);
    }
}
```

Contoh 1

Outputnya…

```java
public class TwoGen<T,V> {
    T ob1;
    V ob2;

    TwoGen(T o1, V o2){
        ob1 = o1;
        ob2 = o2;
    }

    void showTypes(){
        System.out.println("Type of T = "+ob1.getClass().getName());
        System.out.println("Type of V = "+ob2.getClass().getName());
    }

    T getob1(){
        return ob1;
    }

    V getob2(){
        return ob2;
    }
}
```

```java
public class MainClass {
    public static void main (String [] args){
        TwoGen<Integer, String> tgObj = new TwoGen<Integer, String> (88, "Generic");
        tgObj.showTypes();

        int v = tgObj.getob1();
        System.out.println("value : "+v);

        String str = tgObj.getob2();
        System.out.println("value : "+str);
    }
}
```

t - JavaApplication1 (run)

```
run:
Type of T = java.lang.Integer
Type of V = java.lang.String
value : 88
value : Generic
BUILD SUCCESSFUL (total time: 0 seconds)
```

Contoh 1

```java
public class Pair <KeyType, ValueType> {
    private  KeyType key;
    private  ValueType value;

    public Pair(KeyType aKey, ValueType aValue) {
        this.key = aKey;
        this.value = aValue;
    }

    public KeyType getKey() {
        return key;
    }

    public ValueType getValue() {
        return value;
    }

    public void setValue(ValueType value) {
        this.value = value;
    }
```

```java
Pair <Integer, String> p = new Pair <> (1, "A");
```

Cara membuat objek

```java
public class UtamaClass {
    public static void main (String [] args){
        Pair <Integer, String> p = new Pair <Integer, String> (1, "A");
        System.out.println(p.getKey().getClass().getName());
    }
}
```

Contoh 2

- JavaApplication1 (run)

```
run:
java.lang.Integer
BUILD SUCCESSFUL (total time: 0 seconds)
```

```java
public class NonGen {
  Object ob;

  NonGen(Object o) {
    ob = o;
  }

  Object getob() {
    return ob;
  }
  void showType() {
    System.out.println("Type of ob is " +
                       ob.getClass().getName());
  }
}
```

```java
public class NonGenDemo {
  public static void main(String args[]) {
    NonGen integerObject;
    integerObject = new NonGen(88);

    integerObject.showType();

    int v = (Integer) integerObject.getob();
    System.out.println("value: " + v);


    NonGen strOb = new NonGen("Non-Generics Test");
    strOb.showType();

    String str = (String) strOb.getob();
    System.out.println("value: " + str);

    //ini yang menyebabkan excepton
    integerObject = strOb;
    v = (Integer) integerObject.getob();

  }
}
```

Soal 1

Soal 2

```java
public class Gen<T> {
  T ob; // declare an object of type T

  Gen(T o) {
    ob = o;
  }

  T getob() {
    return ob;
  }

  void showType() {
    System.out.println("Type of T is " + ob.getClass().getName());
  }
}
```

```java
public class GenDemo {
  public static void main(String args[]) {
    Gen<Integer> iOb;
    iOb = new Gen<Integer>(88);

    iOb.showType();

    int v = iOb.getob();
    System.out.println("value: " + v);

    System.out.println();

    Gen<String> strOb = new Gen<String>("Generics Test");

    strOb.showType();

    String str = strOb.getob();
    System.out.println("value: " + str);
  }
}
```

```java
class TwoGen<T, V> {
  T ob1;

  V ob2;

  TwoGen(T o1, V o2) {
    ob1 = o1;
    ob2 = o2;
  }

  void showTypes() {
    System.out.println("Type of T is " + ob1.getClass().getName());
    System.out.println("Type of V is " + ob2.getClass().getName());
  }

  T getob1() {
    return ob1;
  }

  V getob2() {
    return ob2;
  }
}
```

Soal 3

```java
public class TwoGenDemo {
  public static void main(String args[]) {
    TwoGen<Integer, String> tgObj = new TwoGen<Integer, String>(88,
"Generics");
    tgObj.showTypes();
    int v = tgObj.getob1();
    System.out.println("value: " + v);
    String str = tgObj.getob2();
    System.out.println("value: " + str);
  }
}
```