**LAPORAN PRAKTIKUM**

49

**ALGORITMA DAN PEMROGRAMAN LANJUT**

**Pertemuan Ke – 10**



**DISUSUN OLEH :**

**HELDA LUDYA SAFITRI**

**175410186**

**TEKNIK INFORMATIKA**

**STMIK AKAKOM**

**YOGYAKARTA**

**2018**

**PERTEMUAN KE-10**

**GENERIC**

1. **TUJUAN**
2. Memahami mengenai konsep generic.
3. Menerapkan teknik generic functional programing pada metode dan kelas.
4. **DASAR TEORI**

Kelas generic adalah tipe khusus dari sebuah kelas yang dapat diasosiasikan pada satu atau lebih tipe data didalam java yang tidak spesifik saat instansiasi. Generic merupakan cara java dalam melakukan generalisasi terhadap tipe data tanpa mengurangi kemampuan java dalam menjaga keamanan penggunaan tipe data.

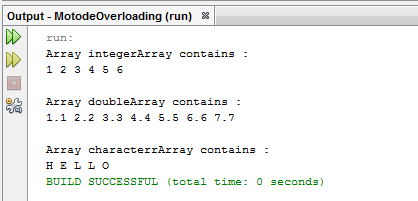
1. **PEMBAHASAN PRAKTIK**

**Praktik 1**

|  |
| --- |
| package motodeoverloading;  public class MotodeOverloading {  public static void main(String[] args) {  Integer[] integerArray={1,2,3,4,5,6};  Double[] doubleArray={1.1,2.2,3.3,4.4,5.5,6.6,7.7};  Character[] characterArray={'H','E','L','L','O'};  System.out.println("Array integerArray contains :");  printArray(integerArray);  System.out.println("\nArray doubleArray contains :");  printArray(doubleArray);  System.out.println("\nArray characterrArray contains :");  printArray(characterArray);  }  public static void printArray(Integer[] inputArray){  for(Integer element : inputArray)  System.out.printf("%s ", element);  System.out.println();  }  public static void printArray(Double[] inputArray){  for(Double element : inputArray)  System.out.printf("%s ", element);  System.out.println();  }  public static void printArray(Character[] inputArray){  for(Character element : inputArray)  System.out.printf("%s ", element);  System.out.println();  }  } |

Pada program ini membuat sebuah program non generic dengan membuat sebuah element dan menggunakan array dan perulangan for sehingga akan menampilkan sebuah nilai array intger, double dan character dengan sebuah nilai yang berbeda dan nilai integer dibuat metode sendiri dan double dibuat metode sendiri, character dibuat metode sendiri.

Output :

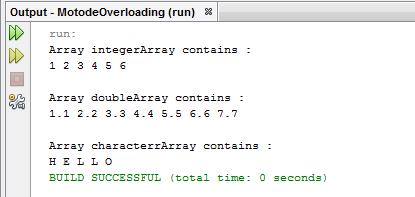


**Praktik 2**

|  |
| --- |
| package motodeoverloading;  public class GenericMetode {  public static void main(String[] args) {  Integer[] integerArray={1,2,3,4,5,6};  Double[] doubleArray={1.1,2.2,3.3,4.4,5.5,6.6,7.7};  Character[] characterArray={'H','E','L','L','O'};  System.out.println("Array integerArray contains :");  printArray(integerArray);  System.out.println("\nArray doubleArray contains :");  printArray(doubleArray);  System.out.println("\nArray characterrArray contains :");  printArray(characterArray);  }  public static <T> void printArray(T[] inputArray){  for(T element : inputArray)  System.out.printf("%s ", element);  System.out.println();  }  } |

Pada program ini membuat sebuah program dengan generic outputnya sama dengan program yang sebelumnya tetapi pada program ini menggunakan generic dengan membuat sebuah element dan membuat metode sendiri sehingga akan menampilkan nilai integer, double dan character.

Output :

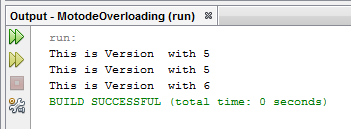


**Praktik 3**

|  |
| --- |
| package motodeoverloading;  public class Generics<T, T2> {  public T one;  public T2 two;  public Generics(T first, T2 second){  one=first;  two=second;  }  public String toString(){  return one +" with "+two;  }  public T getOne(){  return one;  }  public T2 getTwo(){  return two;  }  public void setOne(T temp){  one=temp;  }  public void setTwo(T2 temp){  two=temp;  }  public static void main(String[] args) {  Generics<String, Integer> version=new Generics<String, Integer>("This is Version ", 5);  System.out.println(version);  System.out.println(version.toString());  version.setTwo(6);  System.out.println(version);  }  } |

Program diatas adalah sebuah program dengan membuat kelas generic, jadi pada program ini kita membuat sebuah kelas dengan nama kelas generic<T> dengan sebuah public one dan two artinya dideklarasikan satu2 dengan membuat sebuah variabel version dengan menampilkan sebuah nilai dengan angka nilai 5, 5 dan 6 dengan value this is version.

Output :

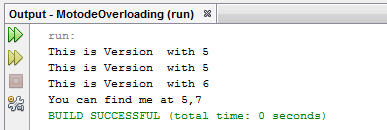


**Praktik 4**

|  |
| --- |
| package motodeoverloading;  public class Genericss {  public static void main(String[] args) {  Generics<String, Integer> version=new Generics<String, Integer>("This is Version ", 5);  System.out.println(version);  System.out.println(version.toString());  version.setTwo(6);  System.out.println(version);  int x=5;  int y=7;  Generics<Integer, Integer> coordinates=new Generics<Integer, Integer>(x,y);  System.out.println("You can find me at " +coordinates.getOne()+"," +coordinates.getTwo());  }  } |

Program diatas adalah membuat kelas generic dengan sebuah koordinat, jadi pada program tersebut sama dengan program yang pertama bedanya pada program ini menggunakan titik koordinat, jadi unntuk menampilkan sebuah titik koordinat kita membuat sebuah variabel coordint dan memanggil sebuah variabel one dan two.

Output :



1. **LATIHAN**

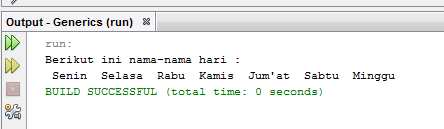
|  |
| --- |
| public class Generics<T, T2> {  public T one;  public T2 two;  public Generics(T first, T2 second){  one=first;  two=second;  }  public String toString(){  return one +" with "+two;  }  public T getOne(){  return one;  }  public T2 getTwo(){  return two;  }  public void setOne(T temp){  one=temp;  }  public void setTwo(T2 temp){  two=temp;  }  public static void main(String[] args) {  String judul="Falling in Love";  String pengarang="Helda Ludya Safitri";  Generics<String, String> buku=new Generics<String, String>(judul,pengarang);  System.out.println("Judul Buku : " +buku.getOne()+ " Nama Penulis " +buku.getTwo());  String namaku="Helda";  int umurku=19;  Generics<String, Integer> Nama=new Generics<String, Integer>(namaku, umurku);  System.out.println("Nama Siswa : " +Nama.getOne()+ " --- > umur : " +Nama.getTwo());  }  } |
|  |

Pada program ini membuat sebuah program dengan menamplkan judul buku dan nama siswa dengan cara membuat sebuah kelas generic dengan tipe integer dan string dengan nama variabel intger umurku dan nama variabel dari string adalah namaku.

|  |
| --- |
| package generics;  public class GenericMetode {  public static void main(String[] args) {  String[] stringArray={"Senin", "Selasa", "Rabu", "Kamis", "Jum'at", "Sabtu", "Minggu"};  System.out.println("Berikut ini nama-nama hari :");  printArray(stringArray);  }  public static <T> void printArray(T[] inputArray){  for(T element : inputArray)  System.out.printf(" %s ", element);  System.out.println();  }  } |

Pada program ini membuat sebuah program dengan menampilkam sebuah nama-nama hari, jadi yang digunakan adlah sebuah program yang bertipe string dengan sebuah array dan dibuat sebuah metode agar bisa tampil ketika program tersebut dipanggil. Dan dibuat sebuah element untuk bisa dieksekusi.

Output :



1. **KESIMPULAN**

Dari pembahasan diatas dapat disimpulkan bahwa generic merupakan cara java dalam melakukan generalisasi terhadap tipe data tanpa mengurangi kemampuan java dalam menjaga keamanan penggunaan tipe data.

1. **LISTING**

Terlampir.