

LAPORAN PRAKTIKUM (Java Collection Framework)

Diajukan untuk memenuhi salah satu tugas praktikum Mata kuliah Komunikasi Data dan Jaringan



Disusun Oleh:
Nalendra Praja Bredtyopati Yudo (231511056)
Jurusan Teknik Komputer dan Informatika

Program Studi D-3 Teknik Informatika
Politeknik Negeri Bandung
2024

Link GitHub : <https://github.com/Cobaltrt2311/OOP-Praktik.git>

a. Penjelasan Aplikasi

Aplikasi ini merupakan demonstrasi penggunaan java collection dalam java menggunakan List, Set, dan Map untuk mengelola dan manipulasi data. Aplikasi ini mencakup berbagai operasi seperti menambahkan, menghapus, mencetak elemen serta melakukan cek dalam List, Set dan Map menggunakan loop dan iterator.

b. Source Code

Example.java	<pre>import java.util.*; public class Example { public static void main(String[] args) throws Exception { List<String> Name = new ArrayList<>(); Name.add("Nalendra Praja"); Name.add("Ananda Dania"); System.out.println(Name); System.out.println(Name.get(0)); System.out.println("Ukuran Array : " + Name.size()); Name.remove(1); System.out.println("Array setelah dihapus : " + Name); for (String name : Name) { System.out.println(name); } for (int i = 0; i < Name.size(); i++) { System.out.println(Name.get(i)); } //===== Set<String> GoatSet = new HashSet<>(); GoatSet.add("Cristiano Ronaldo"); GoatSet.add("Lionel Messi"); GoatSet.add("Lionel Messi"); System.out.println(GoatSet); System.out.println("Ukuran Set : " + GoatSet.size()); for (String name : GoatSet) { System.out.println(name + GoatSet.contains("TOTY")); } GoatSet.remove("Cristiano Ronaldo"); System.out.println("Set setelah dihapus : " + GoatSet); System.out.println(GoatSet.contains("Lionel Messi")); Set<String> TOTY = GoatSet; TOTY.add("N'Golo Kante"); TOTY.add("Toni Kroos"); TOTY.add("Lionel Messi"); for (String name : TOTY) { System.out.println(name + " " + GoatSet.containsAll(TOTY)); } GoatSet.addAll(TOTY); System.out.println("Set GoatSet + TOTY : " + GoatSet); //===== Map<Integer, String> GoatMap = new HashMap<>(); GoatMap.put(7, "Cristiano Ronaldo"); GoatMap.put(10, "Lionel Messi"); GoatMap.put(9, "Marco Van Basten"); System.out.println("Key 10 " +GoatMap.get(10)); System.out.println(GoatMap); GoatMap.remove(9); System.out.println("GoatMap after remove 9 : " +GoatMap);</pre>
--------------	---

	<pre> for (Integer key : GoatMap.keySet()) { System.out.println(key + " -> " + GoatMap.get(key)); } for (Map.Entry<Integer, String> entry : GoatMap.entrySet()) { System.out.println(entry.getKey() + " -> " + entry.getValue()); } //===== Iterator<Integer> it = GoatMap.keySet().iterator(); while (it.hasNext()) { Integer key = it.next(); String value = GoatMap.get(key); System.out.println(key + " -> " + value); if (key == 10) it.remove(); } String nama = Name.get(0); System.out.println(nama); } } </pre>
--	---

Praktik :

```

import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;

// Generic Interface
interface MinMax<T extends Comparable<T>> {
    T min();
    T max();
}

// Generic Class implementing Generic Interface
class MyClass<T extends Comparable<T>> implements MinMax<T> {
    private T[] vals;

    public MyClass(T[] o) {
        this.vals = o;
    }

    @Override
    public T min() {
        T v = vals[0];
        for (int i = 1; i < vals.length; i++) {
            if (vals[i].compareTo(v) < 0) {
                v = vals[i];
            }
        }
        return v;
    }

    @Override
    public T max() {
        T v = vals[0];
        for (int i = 1; i < vals.length; i++) {
            if (vals[i].compareTo(v) > 0) {
                v = vals[i];
            }
        }
        return v;
    }
}

// Generic Method
class Utility {

```

```

    public static <T> void printCollection(Collection<T> collection) {
        for (T item : collection) {
            System.out.println(item);
        }
    }

    public static <T extends Comparable<T>> boolean isSorted(T[] array) {
        for (int i = 1; i < array.length; i++) {
            if (array[i].compareTo(array[i - 1]) < 0) {
                return false;
            }
        }
        return true;
    }
}

// Main Class
public class GenericDemo {
    public static void main(String[] args) {
        // Example 1: Using Generic Class with MinMax
        Integer nums[] = {3, 5, 1, 8, 4};
        MyClass<Integer> obj = new MyClass<>(nums);
        System.out.println("Min: " + obj.min());
        System.out.println("Max: " + obj.max());

        // Example 2: Using Generic Method with Collections
        ArrayList<String> list = new ArrayList<>();
        Collections.addAll(list, "Apple", "Orange", "Banana", "Grape");
        System.out.println("\nCollection Items:");
        Utility.printCollection(list);

        // Example 3: Checking if Array is Sorted
        Integer sortedArray[] = {1, 2, 3, 4, 5};
        System.out.println("\nIs array sorted? " + Utility.isSorted(sortedArray));

        Integer unsortedArray[] = {5, 3, 1, 4, 2};
        System.out.println("Is array sorted? " + Utility.isSorted(unsortedArray));
    }
}

```

c. Penjelasan Source Code

Fungsi yang digunakan :

1. List

```

List<String> Name = new ArrayList<>();
//Membuat objek name dengan tipe list dimana isi list ini tipe datanya adalah string
Name.add("Nalendra Praja");
Name.add("Ananda Dania");

//Menambahkan elemen Nalendra Praja dan Ananda Dania ke dalam List Name

System.out.println(Name);
System.out.println(Name.get(0));
System.out.println("Ukuran Array : " + Name.size());
Name.remove(1);
System.out.println("Array setelah dihapus : " + Name);

// Print 1 : Melakukan pencetakan seluruh elemen yang terdapat pada list
// Print 2 : Melakukan pencetakan elemen berdasarkan indeks
// Print 3 : MelakuaKn pencetakan ukuran list menggunakan fungsi Name.size()
// Fungsi Name.remove(1) : Melakukan penghapusan elemen list pada indeks 1
// Print 4 : Melakukan pencetakan setelah dilakukan penghapusan elemen list pada
indeks 1

```

```

for (String name : Name) {
    System.out.println(name);
}

```

```

for (int i = 0; i < Name.size(); i++) {
    System.out.println(Name.get(i));
}

// Perulangan pertama : untuk menecetak setiap elemen yang terdapat pada name
// Perulangan kedua : untuk mencetak setiap elemen yang terdapat pada name dengan
menggunakan parameter iterasi indeks pada name

```

2. Set

```

Set<String> GoatSet = new HashSet<>();
// Mendeklarasikan objek GoatSet sebagai set bertipe string

GoatSet.add("Cristiano Ronaldo");
GoatSet.add("Lionel Messi");
GoatSet.add("Lionel Messi");
//Menambahkan elemen pada set

System.out.println(GoatSet);

//Menampilkan seluruh elemen yang terdapat pada set, dikarenakan pada set itu tidak
boleh ada duplikasi elemen maka Lionel Messi akan ditampilkan sekali dan disimpan
sekali saja, dimana ukuran set tersebut sebesar 1.

System.out.println("Ukuran Set : " + GoatSet.size());

//Menampilkan ukuran set , dimana ukuran set tersebut berisi dua elemen sehingga
besaran ukuran set tersebut adalah 2.

GoatSet.remove("Cristiano Ronaldo");

// Menghapus elemen "Cristiano Ronaldo" pada set GoatSet

System.out.println("Set setelah dihapus : " + GoatSet);

System.out.println(GoatSet.contains("Lionel Messi"));

// Print 1 : Menampilkan Set setelah penghapusan elemen "Cristiano Ronaldo"
// Print 2 : Menampilkan Goatset yang berisi elemen "Lionel Messi"

Set<String> TOTY = GoatSet;

TOTY.add("N'Golo Kante");

TOTY.add("Toni Kroos");

TOTY.add("Lionel Messi");

// membuat set bernama TOTY dengan type data String dan menambahkan elemen ke dalam
set TOTY

Set<String> intersection = new HashSet<>(GoatSet); // Salin GoatSet

intersection.retainAll(TOTY); // Menyisakan elemen yang ada di kedua set

System.out.println("Irisan GoatSet dan TOTY: " + intersection); // Menampilkan hasil
irisian antara kedua set

// Mencari irisan antara GoatSet dan TOTY dengan menggunakan fungsi retainAll

```

3. Map

```

Map<Integer, String> GoatMap = new HashMap<>();
GoatMap.put(7, "Cristiano Ronaldo");
GoatMap.put(10, "Lionel Messi");
GoatMap.put(9, "Marco Van Basten");
System.out.println("Key 10: " + GoatMap.get(10));
System.out.println("GoatMap: " + GoatMap);
// Membuat Map dengan nama GoatMap dimana key bertipe Integer dan Value bertipe
string dan menambahkan elemen elemen ke dalam GoatMap
// Print 1 : Menampilkan elemen pada GoatMap dengan key 10 yaitu Lionel Messi
// Print 2 : Menampilkan seluruh elemen pada GoatMap
GoatMap.remove(9);
System.out.println("GoatMap setelah remove key 9: " + GoatMap);
// Menghapus elemen pada GoatMap dengan key value 9 lalu menampilkan hasil setelah
remove key value
for (Integer key : GoatMap.keySet()) {
    System.out.println(key + " -> " + GoatMap.get(key));
}
// Menampilkan seluruh elemen pada GoatMap dengan menggunakan iterasi keyset() dimana
akan menghasilkan sebuah set integer key yang berisi kunci pada GoatMap dengan
menggunakan fungsi Integer key : GoatMap.keySet() dan akan di tampilkan dengan
menggunakan fungsi System.out.println(key + " -> " + GoatMap.get(key));

// Iterasi menggunakan entrySet()
for (Map.Entry<Integer, String> entry : GoatMap.entrySet()) {
    System.out.println(entry.getKey() + " -> " + entry.getValue());
}
Menampilkan map dengan menggunakan entry set dimana akan terbentuk sebuah map entry
yang menampung seluruh elemen GoatMap dan menampilkan dengan fungsi
System.out.println(entry.getKey() + " -> " + entry.getValue());.

```

Praktik :

Jadi, program ini sebenarnya menggabungkan beberapa konsep **Generic Programming** dalam Java untuk menunjukkan fleksibilitas dan keamanan tipe dalam pengembangan kode. Pertama, ada kelas generik bernama `MyClass<T>` yang digunakan untuk mencari nilai minimum dan maksimum dari array apa saja. Asalkan elemen array tersebut bisa dibandingkan (harus implementasi `Comparable`), kelas ini bisa langsung bekerja tanpa perlu penyesuaian lagi. Contohnya, dengan

array `Integer`, program bisa langsung menghitung nilai terkecil dan terbesar hanya dengan memanfaatkan metode `min()` dan `max()` yang ada di kelas ini.

Lalu, ada kelas `*utility*` yang dirancang untuk tugas-tugas serbaguna, dengan dua metode utama: `printCollection` dan `isSorted`. Metode `printCollection` menggunakan wildcard (`<?>`) untuk mencetak elemen dari koleksi tipe apa saja, seperti `ArrayList`, `LinkedList`, atau `HashSet`. Jadi, nggak peduli apa jenis koleksi yang kamu gunakan, metode ini tetap bisa bekerja. Sementara itu, metode `isSorted` memeriksa apakah elemen-elemen dalam sebuah array sudah terurut dengan benar. Metode ini juga generik, sehingga bisa digunakan untuk berbagai jenis array selama elemen di dalamnya bersifat dapat dibandingkan.

Pada bagian utama program, contoh aplikasinya diperlihatkan dengan jelas. Misalnya, dengan array `Integer`, program langsung memanfaatkan `MyClass` untuk mencari nilai minimum dan maksimum tanpa ribet. Lalu, `Utility.printCollection` dipakai untuk mencetak isi dari sebuah `ArrayList`, menunjukkan bagaimana metode generik bisa diaplikasikan secara fleksibel pada berbagai tipe koleksi. Hal ini membuktikan bahwa generics membuat kode lebih serbaguna tanpa mengorbankan keamanan tipe.

Terakhir, program ini juga menunjukkan bagaimana metode `isSorted` bekerja. Jika diberikan array yang terurut, seperti `[1, 2, 3, 4, 5]`, metode ini akan mengembalikan `true`. Sebaliknya, untuk array yang tidak terurut, seperti `[5, 3, 1, 4, 2]`, hasilnya adalah `false`. Dengan pendekatan seperti ini, `*Generic Programming*` tidak hanya membuat kode lebih efisien, tetapi juga memastikan tidak ada kesalahan tipe yang biasanya baru muncul di runtime. Intinya, generics membantu membuat program lebih aman, fleksibel, dan mudah digunakan.

d. ScreenShoot Program

```
List awal: [Nalendra Praja, Ananda Dania]
Nama pertama: Nalendra Praja
Ukuran Array: 2
List setelah dihapus: [Nalendra Praja]
Nalendra Praja
Nalendra Praja
GoatSet: [Cristiano Ronaldo, Lionel Messi]
Ukuran GoatSet: 2
GoatSet setelah dihapus: [Lionel Messi]
Apakah Lionel Messi ada di GoatSet? true
TOTY: [N'Golo Kante, Toni Kroos, Lionel Messi]
Irisan GoatSet dan TOTY: [Lionel Messi]
Key 10: Lionel Messi
GoatMap: {7=Cristiano Ronaldo, 9=Marco Van Basten, 10=Lionel Messi}
GoatMap setelah remove key 9: {7=Cristiano Ronaldo, 10=Lionel Messi}
7 -> Cristiano Ronaldo
10 -> Lionel Messi
7 -> Cristiano Ronaldo
10 -> Lionel Messi
7 -> Cristiano Ronaldo
10 -> Lionel Messi
Nama pertama di List: Nalendra Praja
```

```
PS D:\OneDrive - Politeknik Negeri Bandung\Politeknik Negeri Bandung\Semester 3\Pemrograman Berorientasi Objek\Teori\Teori 11 + P\Practice> & 'C:\Program Files\Java\jdk-22\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'D:\OneDrive - Politeknik Negeri Bandung\Politeknik Negeri Bandung\Semester 3\Pemrograman Berorientasi Objek\Teori\Teori 11 + P\Practice\bin' 'GenericDemo'
Min: 1
Max: 8

Collection Items:
Apple
Orange
Banana
Grape

Is array sorted? true
Is array sorted? false
PS D:\OneDrive - Politeknik Negeri Bandung\Politeknik Negeri Bandung\Semester 3\Pemrograman Berorientasi Objek\Teori\Teori 11 + P\Practice>
PS D:\OneDrive - Politeknik Negeri Bandung\Politeknik Negeri Bandung\Semester 3\Pemrograman Berorientasi Objek\Teori\Teori 11 + P\Practice> cd 'd:\OneDrive - Politeknik Negeri Bandung\Politeknik Negeri Bandung\Semester 3\Pemrograman Berorientasi Objek\Teori\Teori 11 + P\Practice'; & 'C:\Program Files\Java\jdk-22\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'D:\OneDrive - Politeknik Negeri Bandung\Politeknik Negeri Bandung\Semester 3\Pemrograman Berorientasi Objek\Teori\Teori 11 + P\Practice\bin' 'GenericDemo'
Min: 1
Max: 8

Collection Items:
Apple
Orange
Banana
Grape

Is array sorted? true
Is array sorted? false
PS D:\OneDrive - Politeknik Negeri Bandung\Politeknik Negeri Bandung\Semester 3\Pemrograman Berorientasi Objek\Teori\Teori 11 + P\Practice>
```


