

# Praktikum Pemrograman Berbasis Objek

Pertemuan 9

# Review Time



1. Exception & Error Class
2. Exception Handling : Try-Catch
3. Exception Handling : Throw & Throws
4. Custom Exception Class

# Materi Pertemuan 9

Yang akan dipelajari hari ini

# Materi Pertemuan 9

## 01

### **Collections Framework**

Mengenal class-class collections yang sudah disediakan Java

## 02

### **Set, List, dan Map**

Mengenal perbedaan antara Set, List dan Map dalam Java

## 03

### **Iterator pada Collections**

Mengenal cara menggunakan iterator ketika memakai collections pada Java

01

# Collections Framework

# Collection

**Collection** atau sering juga disebut sebagai **container** adalah sebuah object sederhana yang menampung lebih dari satu elemen di dalam satu kesatuan.

Collection digunakan untuk **menyimpan**, **mengambil** dan **memanipulasi data**, juga **mentransmisikan data** dari satu method ke method lain.

Collection merepresentasikan item-item data yang **membentuk grup** seperti poker (kumpulan dari kartu), direktori (kumpulan dari file atau folder), kotak surat (kumpulan dari surat-surat), dll.

# Collections Framework

**Collections Framework** adalah Arsitektur yang merepresentasikan dan memanipulasi collection.

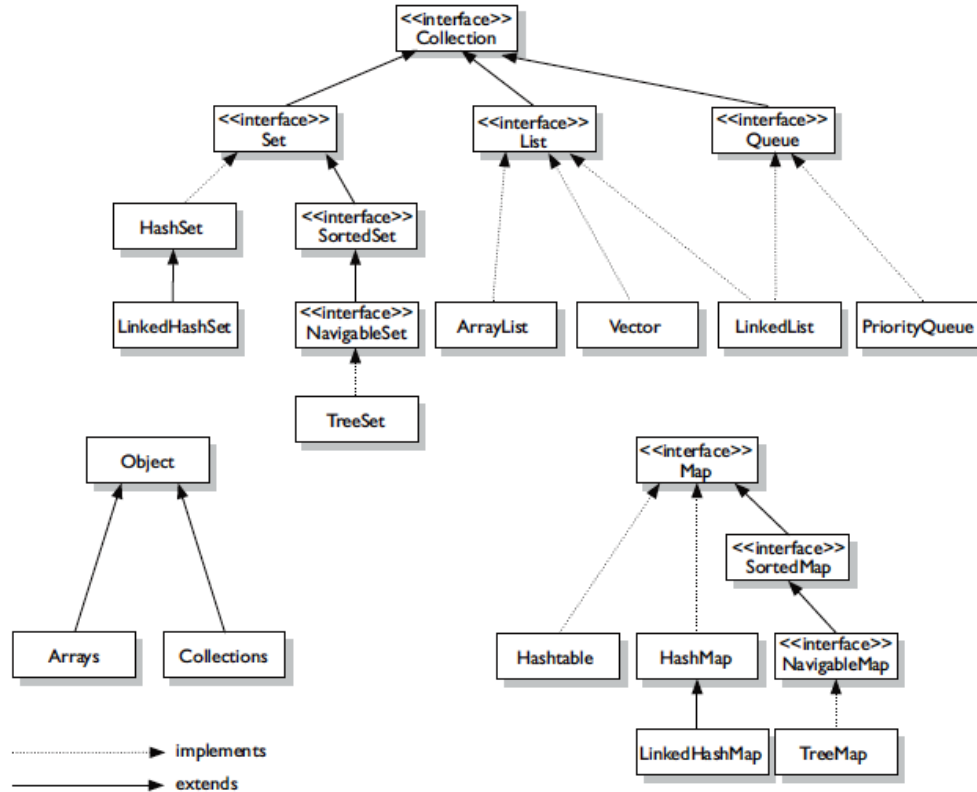
Di dalam Java, Collections Framework ada yang berupa :

- **Interface** (eg. Set, List, Queue, Deque, dll)
- **Class** (eg. ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, dll)

Tipe - Tipe Collections Framework, diantaranya:

- **Set**
- **List**
- **Map**
- **Queue**

# Hirarki Collection Framework







## Set, List dan Map

# Set

Interface **Set** tidak membolehkan duplikasi data di dalam collection

Dua class penting yang ada dalam Java Collections Framework yang mengimplement interface Set adalah: **HashSet** dan **TreeSet**

- **HashSet** merupakan class yang sering digunakan untuk menyimpan collection yang bebas duplikasi
- **TreeSet** merupakan class yang sering digunakan untuk mengekstrak elemen dari collection dalam urutan tertentu

# HashSet

```
import java.util.HashSet;

public class ContohHashSet {
    public static void main(String[] args) {
        HashSet hs = new HashSet(5, 0.5f);
        System.out.println(hs.add("one"));
        System.out.println(hs.add("two"));
        System.out.println(hs.add("three"));
        System.out.println(hs.add("four"));
        System.out.println(hs.add("five"));

        System.out.println(hs);

        Boolean b = hs.add("one");
        System.out.println("Duplicate item allowed = " + b);
        System.out.println(hs);
    }
}
```

# TreeSet

```
import java.util.Set;
import java.util.TreeSet;

public class ContohTreeSet {
    public static void main(String[] args) {
        Set ts = new TreeSet();
        ts.add("one");
        ts.add("two");
        ts.add("three");
        ts.add("four");
        ts.add("five");

        System.out.println(ts);

        Set ts2 = new TreeSet();
        ts2.add(1);
        ts2.add(2);
        ts2.add(3);
        ts2.add(4);
        ts2.add(2);

        System.out.println(ts2);
    }
}
```

# List

Interface **List** digunakan untuk mengoleksi data dalam bentuk terurut dan membolehkan duplikasi

Dua class penting yang ada dalam Java Collections Framework yang mengimplement interface Set adalah: **ArrayList** dan **LinkedList**

- **ArrayList** menyimpan data seperti array (diakses dengan index) namun ukurannya dapat bertambah secara fleksibel
- **LinkedList** merupakan implementasi dari algoritma LinkedList yang telah dipelajari di Struktur data (menyimpan data secara berantai)

# ArrayList

```
import java.util.ArrayList;
import java.util.ListIterator;

public class ContohArrayList {
    public static void main(String[] args) {
        ArrayList al = new ArrayList(2);
        System.out.println(al + ", size = " + al.size());

        al.add("P");
        al.add("B");
        al.add("O");
        al.add(new String("Java"));
        al.add(3, Integer.valueOf(2022));
        System.out.println(al + ", size = " + al.size());

        al.remove("Java");
        System.out.println(al + ", size = " + al.size());
    }
}
```

```
Boolean b = al.contains("Java");
System.out.println("List memiliki item Java = " +
b);
b = al.contains("N");
System.out.println("List memiliki item N = " + b);
b = al.contains(Integer.valueOf(2022));
System.out.println("List memiliki item 2022 = " +
b);

ListIterator li = al.listIterator();
while (li.hasNext()) {
    System.out.println("listIterator ->" + li.next());
}

Object a[] = al.toArray();
for (int i = 0; i < a.length; i++) {
    System.out.println("array -> " + a[i]);
}
}
```

# LinkedList

```
import java.util.LinkedList;

public class ContohLinkedList {
    public static void main(String[] args) {
        LinkedList list = new LinkedList();
        list.add(Integer.valueOf(1));
        list.add(Integer.valueOf(2));
        list.add(Integer.valueOf(3));
        list.add(Integer.valueOf(1));
        System.out.println(list + ", size = " + list.size());

        list.addFirst(Integer.valueOf(0));
        list.addLast(Integer.valueOf(4));
        System.out.println(list + ", size = " + list.size());
        System.out.println(list.getFirst() + ", " + list.size());
        System.out.println(list.get(2) + ", " + list.get(3));

        list.removeFirst();
        list.removeLast();
        System.out.println(list + ", size = " + list.size());
```

```
list.remove(Integer.valueOf(1));
System.out.println(list + ", size = " + list.size());

String s = new String("Jatinangor");
list.add(s);
System.out.println(list + ", size = " + list.size());
System.out.println("Index item Jatinangor dalam list adalah " + list.indexOf(s));

list.remove(2);
System.out.println(list + ", size = " + list.size());

list.set(1, "one");
System.out.println(list + ", size = " + list.size());

LinkedList clone = (LinkedList) list.clone();
clone.add(0, new String("Clone"));
list.add(0, new String("Original"));
System.out.println(list + ", size = " + list.size());
System.out.println(clone + ", size = " + clone.size());
    }
}
```

# Map

Interface **Map** mendeskripsikan pemetaan dari key ke value, dan key tidak boleh duplikat

Dua class penting yang ada dalam Java Collections Framework yang mengimplement interface Map adalah: **HashMap** dan **TreeMap**

- **HashMap** memberikan kemudahan dan kecepatan dalam melakukan insert, hapus dan mencari data dalam Collection
- **TreeMap** dapat menjelajahi key secara terurut

PS. Akan lebih cepat jika menambah data dalam HashMap dulu, dan jika perlu dijelajahi secara terurut, barulah konversikan ke TreeMap



# HashMap

```
import java.util.Map;
import java.util.HashMap;

public class ContohHashMap {
    public static void main(String[] args) {
        String name[] = {
            new String("A"), new String("B"), new String("C"),
            new String("D"), new String("E"), new String("A"),
            new String("B"), new String("A"),
        };

        Map m = new HashMap();

        for (int i = 0; i < name.length; i++) {
            Integer freq = (Integer) m.get(name[i]);

            m.put(name[i], (freq == null ? Integer.valueOf(1) : Integer.valueOf(freq.intValue() + 1)));
        }

        System.out.println(m.size() + " string unik dideteksi");

        System.out.println(m);
    }
}
```

# TreeMap

```
import java.util.Map;
import java.util.TreeMap;

public class ContohTreeMap {
    public static void main(String[] args) {
        String name[] = {
            new String("A"), new String("B"), new String("C"),
            new String("D"), new String("E"), new String("A"),
            new String("B"), new String("A"),
        };

        Map m = new TreeMap();

        for (int i = 0; i < name.length; i++) {
            Integer freq = (Integer) m.get(name[i]);

            m.put(name[i], (freq == null ? Integer.valueOf(1) : Integer.valueOf(freq.intValue() + 1)));
        }

        System.out.println(m.size() + " string unik dideteksi");

        System.out.println(m);
    }
}
```

03

## Iterator pada Collections

# Iterator

**Iterator** merupakan objek yang memungkinkan untuk mengunjungi setiap unsur pada collection satu per satu

Contohnya untuk mengakses seluruh elemen yang ada pada **ArrayList**, kita menggunakan perulangan seperti `for each`, dengan menggunakan **Iterator**, kita dapat menambahkan beberapa method tambahan lainnya

# Iterator

```
import java.util.Collection;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Iterator;
import java.text.DateFormatSymbols;

public class ContohIterator {
    Collection safeMonths;

    public Collection filter(Collection c) {
        Collection filteredCollection = new ArrayList();
        for (Iterator i = c.iterator(); i.hasNext();) {
            String s = (String) i.next();
            if (condition(s)) {
                filteredCollection.add(s);
            }
        }
        return filteredCollection;
    }

    public boolean condition(String s) {
        if (s.contains("r")) return true;
    }
}
```

```
public static void main(String[] args) {
    ContohIterator ci = new ContohIterator();

    DateFormatSymbols dfs = new DateFormatSymbols();
    String[] monthArray = dfs.getMonths();

    Collection months = Arrays.asList(monthArray);

    ci.safeMonths = ci.filter(months);

    System.out.println("The following months are safe");
    System.out.println(ci.safeMonths);
}
}
```

# Exercise!

Semua bisa karena terbiasa

# Exercise

Test.java

Diberikan array integer, cari jumlah pasangan integer yang memiliki perbedaan sejumlah `k`

Misalnya, diberikan array {1, 7, 5, 9, 2, 12, 3} dan  $k = 2$ . Jawabannya adalah 4 karena ada 4 pasangan integer yang memiliki besar perbedaan 2.

Daftar pasangan:

- 1 dan 3
- 3 dan 5
- 5 dan 7
- 7 dan 9

Hint: Pakai collection non duplikasi

INPUT

```
7
1 7 5 9 2 12 3
2
```

- baris pertama: int (panjang array)
- baris kedua : int (bilangan isi array dipisahkan spasi)
- baris ketiga : int (besar perbedaan nilai/ `k`)

Jumlah pasangan bilangan

OUTPUT

```
4
```

# Assignment!

Seperti biasa, setiap selesai praktikum pasti ada tugas



# Assignment 9 Soal 1

Pelajari kedua collection class di bawah lalu berikan contoh penggunaannya dalam bentuk kode dan penjelasannya dalam bentuk komentar dalam kode.

- Vector
- EnumSet

Folder : soal1, File : VectorTest.java, EnumSetTest.java

# Assignment 9 Soal 2

Cari tahu mengenai generic type dalam java. Berikan contoh implementasinya (generic class dan generic method [generic method boleh dibuat di Test.java]) serta jelaskan dalam bentuk komentar dalam kode.

Folder : soal2, File : (Class generic buatan kalian).java, Test.java

# Assignment 9 Soal 3

Selesaikan soal berikut.

## Left Rotation

Operasi left rotation adalah operasi menggeser elemen kedua menjadi yang pertama, ketiga menjadi kedua, dst. Untuk elemen yang pertama akan pindah menjadi elemen yang terakhir.

Contoh input & output sebagai berikut:

```
// complete the rotLeft function below
static int[] rotLeft(int[] array, int numberOfRotation) {

}
```

INPUT

```
5 4
1 2 3 4 5
```

OUTPUT

```
5 1 2 3 4
```

Folder : `soal3`, File : `LeftRotation.java`

# Tugas Kelompok!

Silakan isi daftar anggota kelompok dalam spreadsheet di bawah untuk persiapan proyek PBO.

<https://s.id/1nDKY>

Kolom Ide Proyek dapat diisi setelah info proyek disampaikan melalui discord.

# Teknis Pengumpulan

Pengerjaan dan pengumpulan tugas akan dilakukan di Github Classroom

Kelas A:

Link Tugas Kelas A

Kelas B:

Link Tugas Kelas B

Accept assignment terlebih dahulu lalu link akun Github dengan slot nama yang sesuai di Github Classroom

# Teknis Pengumpulan

Format setiap file `.java` didahulukan dengan Nama, NPM, Kelas, Tanggal, dan Deskripsi

Cara menambah comment di java

```
// untuk single line
/* untuk multiple line */
```

Contoh Format

```
/*
  Nama   : Jane Doe
  NPM    : 99
  Kelas  : A
  Tanggal : 1 September 2021
  Deskripsi : Class jawaban exercise-01 soal-01
*/
```

# Deadline Pengumpulan

Kelas A:

14 November 2022, 23:59 WIB

Kelas B:

15 November 2022, 23:59 WIB

**Waktu yang dilihat adalah waktu last commit.**

Jika ada yang commit melewati deadline walaupun sudah commit sebelumnya akan dianggap telat

# Terima Kasih!

Do you have any questions? Please use respective class discussion channel on Discord.

Semangat terus menjalani kuilahnya!! 🔥🔥🔥