

Praktikum 10

Ekspresi Lambda

1. Tujuan

- Mahasiswa mampu menerapkan ekspresi lambda dengan functional interface.
- Mahasiswa mampu menerapkan ekspresi lambda dalam Collection.

2. Landasan Teori

Ekspresi Lambda merupakan fitur yang baru ditambahkan pada Java SE versi 8. Fitur ini merupakan usaha untuk membuat fungsi anonim atau *anonymous function*, yaitu fungsi yang tidak memiliki nama dan tidak dimiliki oleh sebuah *class*. Hal tersebut bermanfaat ketika akan mengimplementasikan sebuah method yang dimiliki oleh interface.

Ekspresi lambda ditulis menggunakan lambda operator (`->`). Sebelah kiri operator adalah parameter, dan sebelah kanan merupakan ekspresi atau *block statement*. Syntax:

`parameter -> expression`

Sebagai contoh, `(x, y) -> x+y`, adalah sebuah ekspresi lambda untuk menjumlahkan bilangan x dengan y.

Sebelum ada ekspresi lambda, salah satu cara untuk mengimplementasikan *abstract method* pada interface adalah membuat *anonymous class*. *Anonymous class*, yaitu *class* yang tidak memiliki nama, yang akan mengimplementasikan *abstract method*. Sebagai gambaran, tabel 1 menunjukkan perbandingan antara penggunaan anonymous class dan anonymous function.

Anonymous class	Anonymous function
<pre>IDiskon diskonMerdeka = new IDiskon(){ public double hitungDiskon(int harga){ return harga - (harga * 0.3); } };</pre>	<pre>IDiskon diskonBiasa = (harga) -> { return harga - (harga * 0.1); };</pre>

Untuk menggunakan ekspresi lambda, bisa memanfaatkan functional interface yang didefinisikan sendiri maupun yang telah didefinisikan (*predefined*).

3. Langkah Praktikum

3.1. Dasar Ekspresi Lambda

- Tulis kode berikut:

```

/**
 * File DiskonLambda.java
 * Deskripsi: Ekspresi lambda dasar, digunakan untuk menghitung diskon.
 */
interface IDiskon{
    public double hitungDiskon(int harga);
}

public class DiskonLambda {
    public static void main(String[] args) {
        //tanpa lambda
        IDiskon diskonMerdeka = new IDiskon(){
            public double hitungDiskon(int harga){
                return harga - (harga * 0.3);
            }
        };
        //dengan lambda
        IDiskon diskonLebaran = (harga) -> harga - (harga * 0.4);
        //dengan lambda dengan blok statement
        IDiskon diskonBiasa = (harga) -> {
            return harga - (harga * 0.1);
        };
        System.out.println("Diskon Merdeka: "+diskonMerdeka.hitungDiskon(45000));
        System.out.println("Diskon Lebaran: "+diskonLebaran.hitungDiskon(45000));
        System.out.println("Diskon Biasa: "+diskonBiasa.hitungDiskon(45000));
    }
}

```

b. Compile dan jalankan kode tersebut ! Dapatkah anda membedakan antara bagaimana diskonLebaran dan diskonBiasa diimplementasikan ?

3.2. Ekspresi Lambda pada Collection

a. Tulis kode berikut:

```

import java.util.ArrayList;

/**
 * File: Lambdalist.java
 * Deskripsi: Implementasi lambda pada List, digunakan sebagai parameter
 *           pada method.
 */
public class Lambdalist {
    public static void main(String[] args) {
        ArrayList<String> mahasiswaList = new ArrayList<>();
        mahasiswaList.add("Adi");
        mahasiswaList.add("Bambang");
        mahasiswaList.add("Cici");
        mahasiswaList.add("Didi");

        //lambda digunakan sebagai parameter
        mahasiswaList.forEach((nama) -> System.out.println(nama));
    }
}

```

b. Compile dan jalankan kode tersebut !

Latihan

Buatlah sebuah program yang digunakan untuk menampilkan key dan value dari sebuah Map, dimana key merupakan NIM dan value merupakan nama mahasiswa, menggunakan ekspresi lambda ! Gunakan kode pada latihan 3.2 sebagai referensi.