

**PRAKTIKUM 10**  
**PEMROGRAMAN BERORIENTASI OBJEK**  
**LAPORAN**



**DISUSUN OLEH :**  
**BONIFASIUS BRYAN SURYANINGTYAS**  
**(24060121130066)**

**ASISTEN PRAKTIKUM :**  
**1. M. DRAJAT DANY DARMAWAN**  
**2. AGUNG RAMADHANI**

**PROGRAM STUDI INFORMATIKA LAB C1**  
**FAKULTAS SAINS DAN MATEMATIKA**  
**UNIVERSITAS DIPONEGORO**  
**SEMARANG**  
**2023**

## A. DASAR EKSPRESI LAMBDA

### 1. Source Code

```
// Nama : Bonifasius Bryan Suryaningtyas
// NIM : 24060121130066
// Pertemuan 10 Pemrograman Berorientasi Objek Lab C1
// Hari, tanggal : Rabu, 30 Mei 2023
// Nama 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) {
        IDiskon diskonMerdeka = new IDiskon() {
            public double hitungDiskon(int harga) {
                return harga - (harga * 0.3);
            }
        };

        IDiskon diskonLebaran = (harga) -> harga - (harga * 0.4);

        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));
    }
}
```

### 2. Hasil

```
Diskon Merdeka: 31500.0
Diskon Lebaran: 27000.0
Diskon Biasa: 40500.0
```

### 3. Pembahasan Soal

Soal : *Compile* dan jalankan kode tersebut! Dapatkah Anda membedakan antara bagaimana diskonLebaran dan diskonBiasa diimplementasikan?

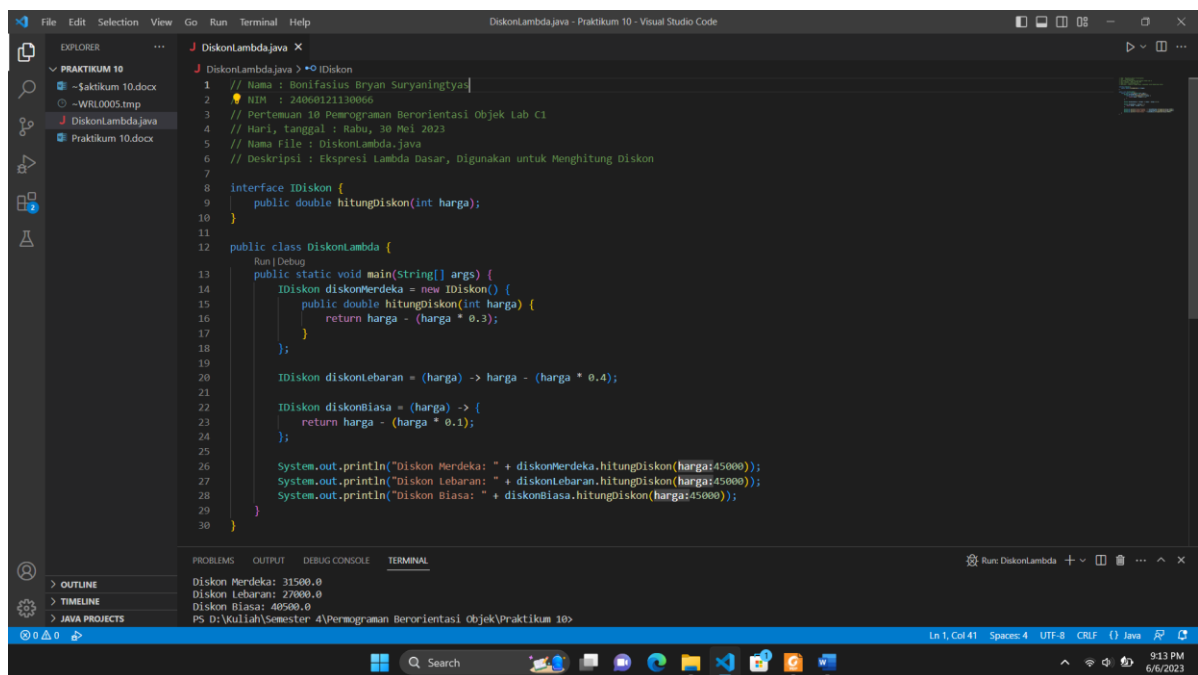
Pembahasan :

Pada diskonLebaran, ekspresi lambda ditulis dalam bentuk singkat tanpa menggunakan tanda kurung kurawal {}. Ekspresi lambda ini langsung mengembalikan hasil perhitungan diskon. diskonLebaran menggunakan diskon

sebesar 40% dari harga yang dihitung dengan mengalikan harga dengan 0.4 dan mengurangnya dari harga awal. Implementasi ini langsung diwakili oleh ekspresi lambda “(harga) → harga - (harga \* 0.4)” tanpa membutuhkan nilai *return*.

Pada diskonBiasa, ekspresi lambda ditulis dalam bentuk blok kode dengan menggunakan tanda kurung kurawal {}. Di dalam blok kode ini, terdapat pernyataan *return* yang digunakan untuk mengembalikan nilai hasil perhitungan diskon. diskonBiasa menggunakan diskon sebesar 10% dari harga yang dihitung dengan mengalikan harga dengan 0.1 dan mengurangnya dari harga awal. Implementasi ini membutuhkan blok kode dengan pernyataan *return* untuk mengembalikan hasil perhitungan diskon.

#### 4. Screenshot



```
1 // Nama : Bonifasius Bryan Suryaningtyas
2 // NIM : 24060121130066
3 // Pertemuan 10 Pemrograman Berorientasi Objek Lab C1
4 // Hari, tanggal : Rabu, 30 Mei 2023
5 // Nama File : DiskonLambda.java
6 // Deskripsi : Ekspresi Lambda Dasar, Digunakan untuk Menghitung Diskon
7
8 interface IDiskon {
9     public double hitungDiskon(int harga);
10 }
11
12 public class DiskonLambda {
13     public static void main(String[] args) {
14         IDiskon diskonMerdeka = new IDiskon() {
15             public double hitungDiskon(int harga) {
16                 return harga - (harga * 0.3);
17             }
18         };
19
20         IDiskon diskonLebaran = (harga) -> harga - (harga * 0.4);
21
22         IDiskon diskonBiasa = (harga) -> {
23             return harga - (harga * 0.1);
24         };
25
26         System.out.println("Diskon Merdeka: " + diskonMerdeka.hitungDiskon(harga:45000));
27         System.out.println("Diskon Lebaran: " + diskonLebaran.hitungDiskon(harga:45000));
28         System.out.println("Diskon Biasa: " + diskonBiasa.hitungDiskon(harga:45000));
29     }
30 }
```

## B. EKPRESI LAMBDA PADA COLLECTION

### 1. Source Code

```
// Nama : Bonifasius Bryan Suryaningtyas
// NIM : 24060121130066
// Pertemuan 10 Pemrograman Berorientasi Objek Lab C1
// Hari, tanggal : Rabu, 30 Mei 2023
// Nama File : LambdaHashMap.java
// Deskripsi : Implementasi Lambda pada HashMap

import java.util.HashMap;

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

```

HashMap<String, String> mahasiswaMap = new HashMap<>();
mahasiswaMap.put("24060121130066", "Bonifasius Bryan");
mahasiswaMap.put("24060121120017", "Ananda Rizky");
mahasiswaMap.put("24060121140158", "Devin Januar");
mahasiswaMap.put("24060121130079", "Fiersa");

mahasiswaMap.forEach((nim, nama) ->
System.out.println("Mahasiswa dengan NIM " + nim + " bernama " +
nama));
}
}

```

## 2. Hasil

```

Mahasiswa dengan NIM 24060121130066 bernama Bonifasius Bryan
Mahasiswa dengan NIM 24060121130079 bernama Fiersa
Mahasiswa dengan NIM 24060121120017 bernama Ananda Rizky
Mahasiswa dengan NIM 24060121140158 bernama Devin Januar

```

## 3. Screenshot

