

LAPORAN PRAKTIKUM 10
MATA KULIAH PEMROGRAMAN BERBASIS OBJEK



Disusun oleh:

Ananda Rizky Pratama

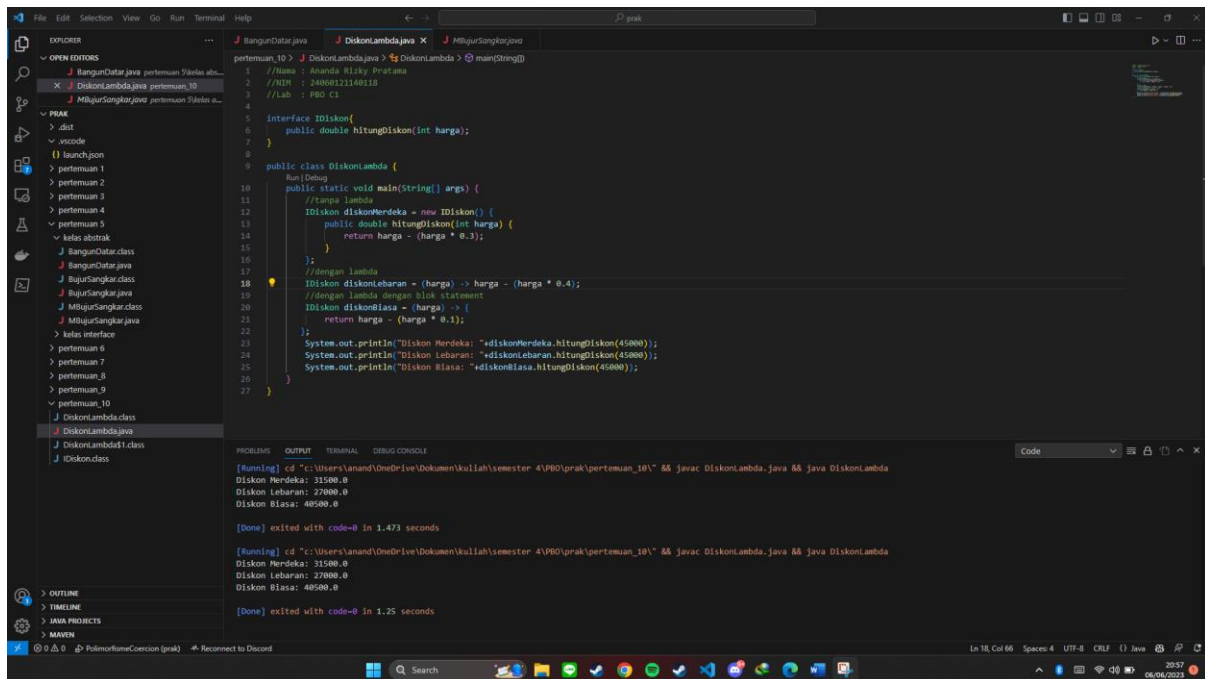
24060121140118

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN
MATEMATIKA UNIVERSITAS
DIPONEGORO SEMARANG
2023

Tugas Praktikum 10

3.1 Dasar Ekspresi Lambda

a. Source code



The screenshot shows an IDE with the following code in `DiskonLambda.java`:

```
1 //Nama : Ananda Rizky Pratama
2 //NIM : 24060121140118
3 //Lab : PBO C1
4
5 interface IDiskon{
6     public double hitungDiskon(int harga);
7 }
8
9 public class DiskonLambda {
10     //tanpa lambda
11     public static void main(String[] args) {
12         IDiskon diskonMerdeka = new IDiskon() {
13             public double hitungDiskon(int harga) {
14                 return harga - (harga * 0.3);
15             }
16         };
17         //dengan lambda
18         IDiskon diskonLebaran = (harga) -> harga - (harga * 0.4);
19         //dengan lambda dengan blok statement
20         IDiskon diskonBiasa = (harga) -> {
21             return harga - (harga * 0.1);
22         };
23         System.out.println("Diskon Merdeka: "+diskonMerdeka.hitungDiskon(45000));
24         System.out.println("Diskon Lebaran: "+diskonLebaran.hitungDiskon(45000));
25         System.out.println("Diskon Biasa: "+diskonBiasa.hitungDiskon(45000));
26     }
27 }
```

The output window shows the following results:

```
[Running] cd "C:\Users\anand\OneDrive\Documents\kuliah\semester 4\PBO\prak\pertemuan_10\" && javac DiskonLambda.java && java DiskonLambda
Diskon Merdeka: 31500.0
Diskon Lebaran: 27000.0
Diskon Biasa: 40500.0
[Done] exited with code=0 in 1.473 seconds

[Running] cd "C:\Users\anand\OneDrive\Documents\kuliah\semester 4\PBO\prak\pertemuan_10\" && javac DiskonLambda.java && java DiskonLambda
Diskon Merdeka: 31500.0
Diskon Lebaran: 27000.0
Diskon Biasa: 40500.0
[Done] exited with code=0 in 1.25 seconds
```

```
//Nama : Ananda Rizky Pratama
//NIM : 24060121140118
//Lab : PBO C1

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?

Dalam implementasi 'diskonLebaran', sebuah lambda expression '(harga) -> harga - (harga * 0.4)' digunakan untuk menggantikan implementasi langsung dari metode 'hitungDiskon()'. Dalam hal ini, lambda expression tersebut menerima argumen 'harga' dan mengembalikan hasil pengurangan harga awal dengan diskon 40%.

Sementara itu, dalam implementasi 'diskonBiasa', sebuah lambda expression dengan blok statement '(harga) -> { return harga - (harga * 0.1); }' digunakan. Lambda expression ini juga menerima argumen 'harga', tetapi berbeda dengan 'diskonLebaran', lambda expression ini memiliki blok statement yang menggunakan kata kunci 'return' untuk mengembalikan hasil pengurangan harga awal dengan diskon 10%.

Pada edua kasus tersebut menggunakan lambda expression untuk mengimplementasikan metode 'hitungDiskon()', namun gaya penulisannya berbeda. 'diskonLebaran' menggunakan lambda expression sederhana tanpa blok statement, sementara 'diskonBiasa' menggunakan lambda expression dengan blok statement. Meskipun demikian, hasil keluaran dari kedua implementasi tersebut sama, yaitu menghitung diskon berdasarkan persentase yang diberikan.

The screenshot shows an IDE with a Java file named 'Lambdahasmap.java'. The code defines a class 'Lambdahasmap' with a 'main' method. It uses a 'HashMap' to store student data and iterates over it to print names and NIMs. The output console shows the following results:

```

Priscilla : 24060121140781
Boy : 24060121140118
Kinanti : 24060121140101

```

The code in the IDE is as follows:

```

import java.util.HashMap;

//Nama : Ananda Rizky Pratama
//NIM : 24060121140118
//Lab : PBO C1

public class Lambdahasmap {
    public static void main(String[] args) {
        HashMap<String, String> mahasiswaMap = new HashMap<>();
        mahasiswaMap.put("Dienara", "24060121140118");
        mahasiswaMap.put("Priscilla", "24060121137171");
        mahasiswaMap.put("Boy", "24060121140781");
        mahasiswaMap.put("Kinanti", "24060121130191");

        // menggunakan lambda
        mahasiswaMap.forEach((nama, nim) -> System.out.println(nama + " : " + nim));
    }
}

```

```

import java.util.HashMap;

//Nama : Ananda Rizky Pratama
//NIM : 24060121140118

```

```
//Lab : PBO C1

public class LambdaHashMap {
    public static void main(String[] args) {
        HashMap<String, String> mahasiswaMap = new HashMap<>();
        mahasiswaMap.put("Dienara", "24060121140118");
        mahasiswaMap.put("Priscilla", "24060121137171");
        mahasiswaMap.put("Boy", "24060121140781");
        mahasiswaMap.put("Kinanti", "24060121120191");

        // menggunakan lambda
        mahasiswaMap.forEach((nama, nim) -> System.out.println(nama + " : "
+ nim));
    }
}
```