PERTEMUAN KEDUABELAS

Java Lambda

Fungsi Anonymous

Anonymous menggambarkan sesuatu yang tidak memiliki nama. Di Java terdapat fungsi *anonymous* dan *class anonymous* seperti yang telah dibahas pada materi sebelumnya.

Fungsi *anonymous* adalah fungsi yang tidak memiliki nama. Fungsi *anonymous* di Java dikenal dengan nama ***Lambda Expression***.

Fungsi anonymous biasanya dibuat hanya untuk sekali pakai.

Artinya, saat kita membuat fungsi anonymous, kita akan mengeksekusinya saat itu juga. Tidak bisa dipanggil lagi seperti fungsi biasa.

Fungsi ini mulai ditambahkan pada JDK 8.

Membuat Fungsi Anonymous di Java

Bentuk umum *lambda expression* adalah

```
(int param1, int param2 ) -> {
    // kode fungsi
}
```

Simbol yang perlu diingat adalah

```
() -> { }
```

Keterangan:

- () tempat menaruh paramater
- -> adalah operator lambda yang menandakan bahwa fungsi ini adalah anonymous
- {} body fungsinya

Contoh:

```
(int x, int y) -> { return x + y };
```

Fungsi anonymous dapat dibuat di berbagai tempat seperti:

- Deklarasi variable

```
int variabel = () -> { return 0 };
```

- Pengisian variable dan array

```
int variabel;
int arr;
variabel = () -> { return 0 };
arr = () -> { return {0,4,3,2,1} };
```

- Saat mengembalikan nilai dengan return

```
int methodName(){
    return () -> { return 0 };
}
```

- Body lambda

```
() -> {
    return () -> 5 + 2;
};
```

- Ekspresi kondisional

```
String jawab = (int x) -> { x < 10} ? () -> return "yes": () -> return
"no";
```

Kenapa Menggunakan Fungsi Anonymous?

Lambda expression atau fungsi anonymous sebenarnya hadir untuk menyempurnakan class anonymous. Class anonymous biasanya digunakan untuk mengimplementasikan interface dan class abstrak. Tapi kendalanya saat interface hanya memiliki satu method saja untuk diimplementasikan. Kita harus membuat class (anonymous) baru. Padahal hanya dibutuhkan method-nya saja. Ini lah saatnya untuk menggunakan fungsi anonymous atau lambda expression.

Contoh Program Fungsi Anonymous

Interface Clickable

```
public interface Clickable {
    void onClick();
}
```

Class Button

```
public class Button {
    private Clickable action;

    public void setClickAction(Clickable action){
        this.action = action;
    }

    public void doClick(){
        action.onClick();
    }
}
```

Main Class dengan Anonymous Class

```
public class Main {
    public static void main(String[] args) {
        Button btn = new Button();
        String name = "Jagoan Neon";

        // membuat class anonymous untuk implementasi interface
        btn.setClickAction(new Clickable() {
            @Override
            public void onClick() {
                System.out.println("Tombol sudah diklik!");
                System.out.println("Yay!");
                System.out.println("Hello, " + name + "!");
            }
        });

        // mencoba klik tombol
        btn.doClick();
    }
}
```

Output

```
Tombol sudah diklik!
Yay!
Hello, Jagoan Neon!
```

Main Class dengan Lambda Expression

```
public class Main {
    public static void main(String[] args) {
        Button btn = new Button();
        String name = "Jagoan Neon";

        // membuat lambda expression untuk implementasi interface
        btn.setClickAction(() -> {
            System.out.println("Tombol sudah diklik!");
            System.out.println("Yay!");
            System.out.println("Hello, " + name + "!");
        });

        // mencoba klik tombol
        btn.doClick();
    }
}
```

Output

```
Tombol sudah diklik!
Yay!
Hello, Jagoan Neon!
```