BAB 5

Mendapatkan Input dari Keyboard

5.1 Tujuan

Kita telah mempelajari konsep dasar pada Java dan menulis beberapa program sederhana. Sekarang kita akan mencoba membuat program kita lebih interaktif dengan menggunakan input dari keyboard. Pada bab ini, kita akan mempelajari dua cara memberikan input, yang pertama adalah menggunakan class BufferedReader dan melalui GUI (Graphical User Interface) dengan menggunakan class JOptionPane.

Pada akhir pembahasan, diharapkan pembaca dapat :

- Membuat program Java yang interaktif yang bisa mendapatkan input dari keyboard
- Menggunakan class BufferedReader untuk mendapatkan input dari keyboard melalui layar console
- Menggunakan class JOptionPane untuk mendapatkan input dari keyboard menggunakan GUI

5.2 Menggunakan BufferedReader untuk mendapatkan input

Pada bagian ini, kita akan menggunakan class BufferedReader yang berada di package `java.io` untuk mendapatkan input dari keyboard.

Berikut ini adalah langkah-langkah yang diperlukan untuk mendapatkan input dari keyboard:

1. Tambahkan di bagian paling atas code Anda:

   import java.io.*;

2. Tambahkan statement berikut:

   ```
   BufferedReader dataIn = new BufferedReader(new InputStreamReader( System.in) );
   ```

3. Deklarasikan variabel String temporary untuk mendapatkan input, dan gunakan fungsi readLine() untuk mendapatkan input dari keyboard. Anda harus mengetikkannya di dalam blok try-catch:

   ```
   try{
        String temp = dataIn.readLine();
   }
   catch( IOException e ){
        System.out.println("Error in getting input");
   }
   ```

Berikut ini adalah source code lengkapnya:

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;


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

       BufferedReader dataIn = new BufferedReader(new
                InputStreamReader( System.in) );

       String name = "";

       System.out.print("Please Enter Your Name:");

       try{
                name = dataIn.readLine();
                }catch( IOException e ){
                System.out.println("Error!");
       }

       System.out.println("Hello " + name +"!");
    }
}
```

Berikutnya akan penjelasan setiap baris dari code tersebut:

Statement,

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
```

menjelaskan bahwa kita akan menggunakan class **BufferedReader**, **InputStreamReader** dan **IOException** yang berada di **java.io package**. Java Application Programming Interface (API) berisi ratusan class yang sudah didefinisikan sebelumnya yang dapat digunakan untuk program Anda. Class-class tersebut dikumpulkan di dalam **packages.**

Packages berisi class yang mempunyai fungsi yang saling berhubungan. Seperti pada contoh di atas, **java.io package** mengandung class-class yang memungkinkan program untuk melakukan input dan output data. Pernyataan di atas juga dapat ditulis sebagai berikut,

```
import java.io.*;
```

yang akan mengeluarkan semua class yang berada dalam package, dan selanjutnya kita bisa menggunakan class-class tersebut dalam program kita.

Dua statement selanjutnya,

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

kita sudah mempelajari pada pelajaran sebelumnya. Pernyataan ini mendeklarasikan class bernama `GetInputFromKeyboard` dan kita mendeklarasikan method `main`.

Dalam statement,

```
BufferedReader dataIn = new BufferedReader(new
                            InputStreamReader( System.in) );
```

kita mendeklarasikan sebuah variabel bernama **dataIn** dengan tipe class **BufferedReader**. Jangan mengkhawatirkan tentang maksud dari syntax saat ini. Kita akan menjelaskannya pada akhir pembahasan.

Sekarang, kita akan mendeklarasikan variabel String dengan identifier name,

```
String name = "";
```

Pernyataan di atas merupakan tempat untuk menyimpan input dari user. Nama variabel diinisialisasi sebagai String kosong "". Sebaiknya kita selalu menginisialisasi sebuah variabel setelah kita mendeklarasikannya.

Baris berikutnya adalah memberikan output sebuah String pada layar yang menanyakan nama user.

```
System.out.print("Please Enter Your Name:");
```

Sekarang, blok di bawah ini merupakan try-catch block,

```
try{
    name = dataIn.readLine();
}catch( IOException e ){
    System.out.println("Error!");
}
```

Pada baris ini menjelaskan bahwa kemungkinan terjadi error pada pernyataan,

```
name = dataIn.readLine();
```

akan ditangkap. Kita akan membahas tentang penanganan exception pada bab selanjutnya dari pembahasan ini, tetapi untuk sekarang, Anda cukup mencatat bahwa Anda perlu menambahkan kode ini untuk menggunakan method readLine() dari BufferedReader untuk mendapatkan input dari user.

Selanjutnya kembali ke pernyataan,

```
name = dataIn.readLine();
```

method diatas memanggil `dataIn.readLine()`, mendapatkan input dari user dan memberikan sebuah nilai String. Nilai ini akan disimpan ke dalam variabel `name`, yang akan kita gunakan pada statement akhir untuk menyambut user,

```
System.out.println("Hello " + name + "!");
```

5.1 Menggunakan JOptionPane untuk mendapatkan input

Cara lain untuk mendapatkan input dari user adalah dengan menggunakan class JoptionPane yang didapatkan dari javax.swing package. JoptionPane memudahkan memunculkan dialog box standard yang memberikan kepada user sebuah nilai atau menginformasikan sesuatu.

Diberikan kode berikut ini,

```
import javax.swing.JOptionPane;

public class GetInputFromKeyboard
{
     public static void main( String[] args ){
        String name = "";
        name = JoptionPane.showInputDialog("Please enter your
                                                        name");

        String msg = "Hello " + name + "!";
        JOptionPane.showMessageDialog(null, msg);
     }
}
```

Akan menghasilkan output,

Gambar 1: Mendapatkan Input menggunakan JOptionPane

Gambar 2: Input florence pada JOptionPane

Gambar 3: Menunjukkan Pesan Menggunakan JOptionPane

Statement pertama,

```
import javax.swing.JOptionPane;
```

Menjelaskan bahwa kita mengimpor class `JoptionPane` dari package `javax.swing`.

Bisa juga ditulis seperti,

```
import javax.swing.*;
```

Pernyataan,

```
name = JOptionPane.showInputDialog("Please enter your name");
```

membuat sebuah input dialog `JOptionPane`, yang akan menampilkan dialog dengan sebuah pesan, sebuah textfield dan tombol OK seperti pada gambar. Hasil dari dialog tersebut adalah String dan disimpan ke dalam variabel `name`.

Sekarang kita membuat pesan selamat datang, yang akan disimpan ke dalam variabe `msg`,

```
String msg = "Hello " + name + "!";
```

Baris selanjutnya adalah menampilkan sebuah dialog yang berisi sebuah pesan dan tombol OK,

```
JOptionPane.showMessageDialog(null, msg);
```

5.1 Latihan

5.1.1 Kata Terakhir (versi BufferedReader)

Menggunakan BufferedReader, tanyakan tiga kata dari user dan tampilkan output dari input user tersebut ke layar. Contoh,

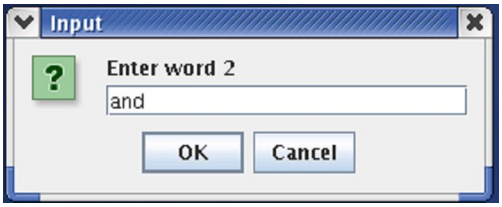
```
Enter word1:Goodbye
Enter word2:and
Enter word3:Hello

Goodbye and Hello
```

5.1.2 Kata Terakhir (versi JOptionPane)

Menggunakan JOptionPane, tanyakan tiga kata dari user dan tampilkan output dari input user tersebut ke layar. Contoh

Gambar 1: Input Pertama



Gambar 2: Input Kedua

Gambar 3: Menampilkan Pesan