

## BAB 5

# Mendapatkan Input dari Keyboard

## 5.1 Tujuan

Kita sudah mempelajari konsep mendasar 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 kelas `BufferedReader` dan melalui GUI (Graphical User Interface) dengan menggunakan kelas `JOptionPane`.

Pada akhir bab ini, para siswa diharapkan mampu:

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

## 5.2 Menggunakan `BufferedReader` untuk mendapatkan input

Pada bagian ini, kita akan menggunakan kelas `BufferedReader` yang berada di `java.io` package 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 ini:

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

3. Deklarasikan variabel `String` temporer 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 selengkapnya:

```
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 dijelaskan setiap baris dari code:

Statement,

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

menjelaskan bahwa kita akan menggunakan kelas **BufferedReader**, **InputStreamReader** dan **IOException** yang berada di **java.io package**. Java Application Programming Interface (API) sudah berisi ratusan kelas yang bisa digunakan untuk program anda. Kelas-kelas tersebut dikumpulkan ke dalam **packages**.

**Packages** memiliki kelas yang mempunyai fungsi yang saling berhubungan. Seperti pada contoh di atas, **java.io package** mengandung kelas-kelas yang memungkinkan program untuk melakukan input dan output data. Statement diatas juga dapat ditulis,

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

yang akan mengeluarkan semua kelas yang berada pada paket, dan selanjutnya kita bisa menggunakan kelas-kelas tersebut pada program kita.

Dua statement selanjutnya,

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

kita sudah mempelajari pada bab sebelumnya. Statement ini menyatakan bahwa kita mendeklarasikan sebuah class bernama `GetInputFromKeyboard` dan kita mendeklarasikan `main` method.

Pada statement,

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

kita mendeklarasikan sebuah variabel bernama **`dataIn`** dengan tipe kelas **`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 = "";
```

Statement diatas merupakan tempat untuk menyimpan input dari user. Variabel `name` diinisialisasi sebagai `String` kosong `""`. Sebaiknya kita selalu menginisialisasi sebuah variabel setelah kita mendeklarasikannya.

Baris berikutnya adalah memberikan output string pada layar menanyakan nama user.

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

Sekarang, block 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 statement

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

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

Selanjutnya statement,

```
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 kelas `JOptionPane` yang didapatkan dari `javax.swing` package. `JOptionPane` membuat kemudahan dengan memunculkan dialog box standar yang memberikan kepada user sebuah nilai atau menginformasikan sesuatu.

Berikan 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,





Statement pertama,

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

Menjelaskan bahwa kita mengimport kelas `JOptionPane` dari `javax.swing` package.

Bisa juga ditulis,

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

statement selanjutnya,

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

membuat sebuah `JOptionPane` input dialog, 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 variabel `msg`,

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

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

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

## 5.1 Latihan

### 5.1.1 Kata Terakhir (versi *BufferedReader*)

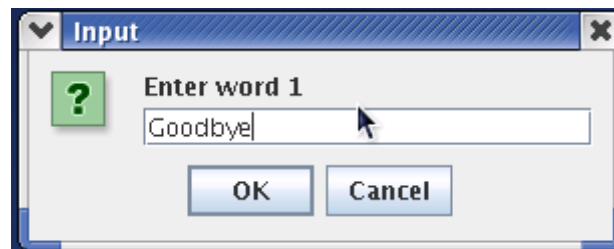
Menggunakan *BufferedReader*, tanyakan tiga kata dari user dan tampilkan output dari input user tersebut ke layar. Contoh,

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

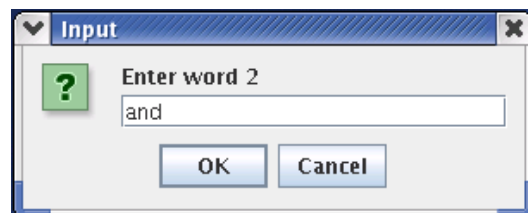
```
Goodbye and Hello
```

### 5.1.2 Kata Terakhir (versi *JOptionPane*)

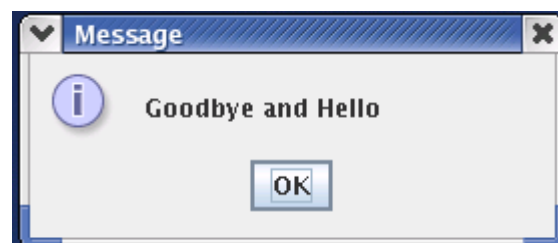
Menggunakan *JOptionPane*, tanyakan tiga kata dari user dan tampilkan output dari input user tersebut ke layar. Contoh



Gambar 1: Input Pertama



Gambar 2: Input Kedua



Gambar 3: Menampilkan Pesan