

TPI4202
e-tp.ub.ac.id

Input and Output (IO)

Lecture 4
Mas'ud Effendi

Output

- Menampilkan informasi kepada user lewat display
- Dengan perintah 'print' dan 'println'
 - `print`: menampilkan output (kursor tetap di baris yang sama)
 - `println`: menampilkan output (kursor berpindah ke baris selanjutnya)

Output

- **Format**

```
System.out.print("informasi");
```

ATAU

```
System.out.println("informasi");
```

Output Example

```
Class tampil{
Public static void main(String[] args){
    System.out.println("Selamat Belajar");
    System.out.print("Pemrograman ");
    System.out.print("Java");
}
}
```

Output (Variables And Constants)

- **Format:**

```
System.out.print(nama_variabel/konstanta);
```

atau

```
System.out.println(nama_variabel/konstanta);
```

Output Variables And Constants

Example

```
Class tampil{
Public static void main(String[] args) {
    int umurku=19;
    String keterangan="Mahasiswa";
    System.out.println("Aku "+keterangan);
    System.out.print("Umurku "+umurku);
}
}
```

Input

- Program komputer mendapat input dari user
- Dukungan paket *java.io*
- Program Java melakukan I/O melalui *stream*
- Dilakukan dengan menggunakan banyak cara, diantaranya:
 - Kelas `BufferedReader`
 - GUI Kelas `JOptionPane`

Class BufferedReader

- Bentuk kode objek:

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

- Atau

```
InputStreamReader isr = new  
InputStreamReader(System.in);  
BufferedReader br = new BufferedReader(isr);
```


Class BufferedReader

Class **BufferedReader**

- Terdapat di java.io package
- Digunakan untuk mendapatkan input

Langkah-langkah Mendapatkan Input

1. Tambahkan baris code ini pada awal *source code* Anda:

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

2. Tambahkan *statement* berikut ini:

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

Langkah-langkah Mendapatkan Input

3. Deklarasikan variabel sementara String untuk mendapatkan input, dan menjalankan perintah **readLine()** untuk mendapatkan input dari keyboard.
Anda harus menuliskannya di dalam blok try-catch.

```
try{
    String temp = dataIn.readLine();
}

catch( IOException e ){
    System.out.println("Error in getting input");
}
```

Contoh program

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

public class BacaInput {

    public static void main( String[] args ) {
        BufferedReader dataIn = new BufferedReader(new InputStreamReader(
            System.in ) );

        String name = "";
        System.out.print("Masukkan Nama Anda:");
        try {
            name = dataIn.readLine();
        }
        catch( IOException e ) {
            System.out.println("Error!");
        }
        System.out.println("Hello " + name + "!");
    }
}
```

Contoh program

- Pada baris,

```
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.

- Statement di atas dapat juga ditulis seperti,

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

Contoh program

- Java Application Programming Interface (API) memiliki ratusan class yang sudah dibuat dan dapat digunakan pada program Anda.
- Class-class tersebut dikumpulkan ke dalam package.
- **Package** mengandung class-class yang memiliki fungsi-fungsi tertentu.

Contoh program

- Pada statement,

```
public class BacalInput {
```

berarti kita mendeklarasikan sebuah class bernama **BacalInput**.

- Statement selanjutnya adalah mendeklarasikan method **main()**.

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

Contoh program

Pada statement,

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

mendeklarasikan variabel bernama **dataIn**,
dengan tipe class **BufferedReader**.

Contoh program

- Pada statement,

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

mendeklarasikan variabel **name** dengan identifier bernama **String**.

- Statement selanjutnya,

```
System.out.print("Masukkan Nama Anda:");
```

menghasilkan output String pada layar yang menanyakan nama user.

Contoh program

Baris/blok di bawah ini merupakan blok try-catch.

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

Code di atas digunakan untuk menangkap exception yang mungkin terjadi pada statement :

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

Contoh program

- Pada statement,

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

method **dataIn.readLine()** akan dipanggil, untuk mendapatkan input dari user dan memberikan *return value* dalam String.

- *value* tersebut kemudian disimpan di dalam variabel **name**, yang akan digunakan pada statement akhir, untuk menyapa user,

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

Class JOptionPane

- Cara lain untuk mendapatkan input dari user adalah dengan menggunakan class **JOptionPane** yang terdapat pada package **javax.swing**.
- JOptionPane dapat membuat dialog box standard yang menginformasikan kepada user untuk memasukkan sebuah nilai.

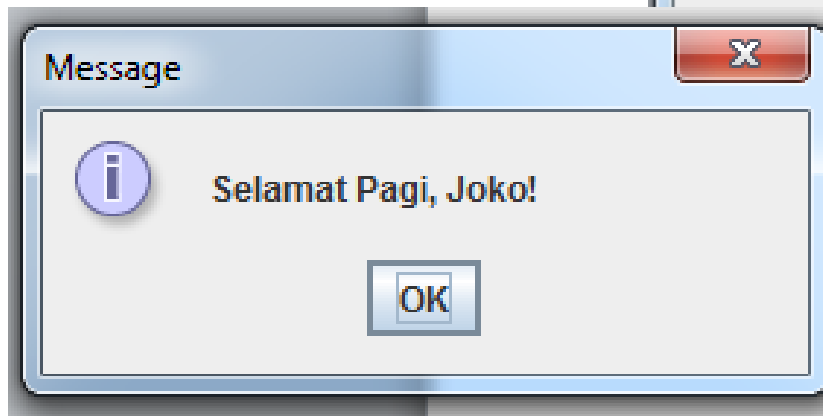
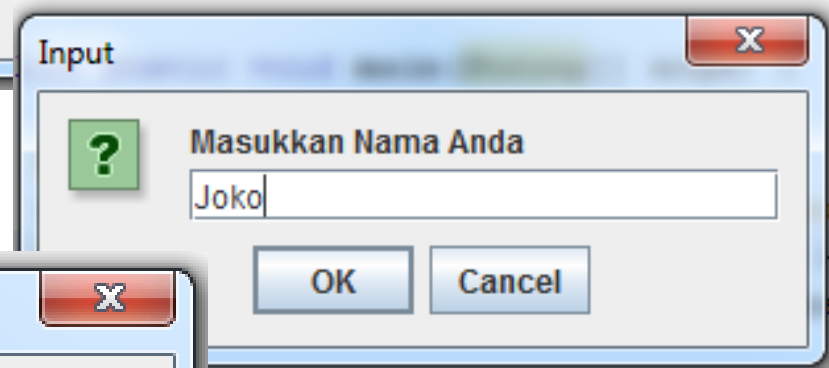
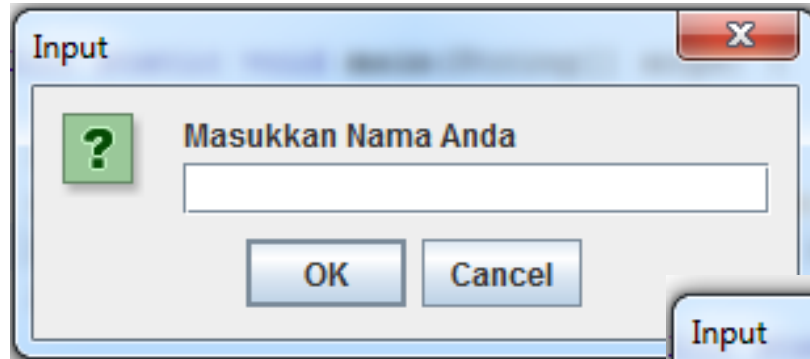
Contoh program

```
import javax.swing.JOptionPane;

public class BacalInput {

    public static void main( String[] args ){
        String name = "";
        name=JOptionPane.showInputDialog("Masukkan Nama Anda");
        String msg = "Selama Pagi, " + name + "!";
        JOptionPane.showMessageDialog(null, msg);
    }
}
```

Contoh output program



Contoh program

- Pada statement,

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

menginformasikan bahwa kita akan meng-import class **JOptionPane** dari package **javax.swing**.

- Kode di atas dapat juga ditulis,

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

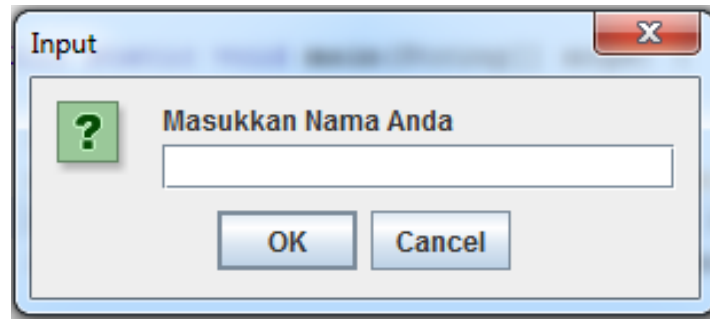
Contoh program

- Statement,

```
name=JOptionPane.showInputDialog("Masukkan Nama Anda");
```

digunakan untuk membuat input dialog **JOptionPane**, yang akan menampilkan dialog, yang terdiri atas sebuah **message**, sebuah **textfield** dan sebuah **button OK**, seperti terlihat pada gambar.

- Dialog tersebut akan memberikan *return value* String yang akan disimpan di variabel **name**.



Contoh program

Statement,

```
String msg = "Selamat Pagi," + name + "!";
```

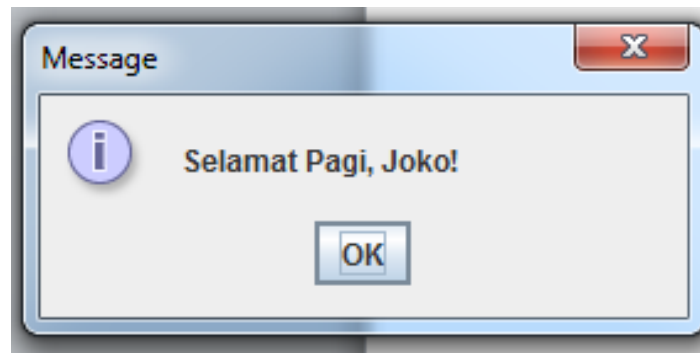
digunakan untuk membuat *welcome message*, yang akan disimpan di variabel **msg**.

Contoh program

Statement,

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

Akan menampilkan sebuah dialog yang berisi sebuah pesan dan sebuah OK button.



Kesimpulan

- Mempelajari dua cara untuk mendapatkan input dari user dengan menggunakan class:
 - **BufferedReader**
 - **JOptionPane**
- Penjelasan singkat tentang package
 - Merupakan sekumpulan class di Java
 - Class berada di dalam package
 - Class menggunakan package dengan cara mengimportnya