

STATEMENT INPUT / OUTPUT

Disain & Bulkyow Dosen : Sulistyowati, ST., M.Kom.

Standar Input Output Java

Standar input output yang biasa ditangani oleh pemrograman Java adalah :

- **System.in**, berguna untuk menangani pembacaan input user dari keyboard (standar input). Standar input ini digunakan untuk membentuk berbagai macam metode input keyboard.
- **System.out**, berguna untuk mengirimkan keluaran (output) ke layar (standar output), biasanya dikombinasikan dengan metode **println()** atau **print()**.
- **System.err**, digunakan untuk mengirimkan pesan kesalahan. Biasanya diletakkan pada Exception Handling. Sehingga sistem ini dikenal dengan sebutan standar error.

Statement Output

- Statement output pada Java ada 2, yaitu :
 - ✓ `System.out.print ()` , yaitu mencetak tampilan tanpa enter
 - ✓ `System.out.println ()`, yaitu mencetak tampilan dengan enter
- Contoh :

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

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

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

Maka hasilnya adalah :

belajarJava

mudah

Statement Input

- Pada Java terdapat beberapa cara yang dapat dilakukan untuk memasukkan data dari keyboard, yaitu menggunakan:
 - Class Scanner
 - Class BufferedReader
 - GUI (Graphical User Interface) JOptionPane
- **Perbedaan Scanner, BufferedReader dan GUI :**
 - Pada Scanner, ketika ada perhitungan matematika, maka variabel yang akan digunakan dalam perhitungan tidak perlu di konversikan lagi, bisa langsung dihitung.
 - Pada BufferedReader → dengan fungsi (InputStreamReader + BufferedReader), variabel perlu dikonversikan terlebih dahulu sebelum dilakukan perhitungan matematika.
 - JOptionPane yang merupakan packages dari javax.swing, digunakan untuk input dan output data berbasis GUI swing. Tampilannya memudahkan pengguna karena berupa dialog box.

Statement Input - Scanner

- Deklarasi umum penggunaan class Scanner adalah sebagai berikut:

Scanner var1 = new Scanner(System.in);

Keterangan :

- var1 : adalah suatu variabel bertipe Scanner
 - new Scanner(System.in) : menciptakan suatu objek bertipe Scanner
- Class Scanner berada pada paket java.util, maka harus mengimpornya terlebih dahulu pada baris sebelum deklarasi *class* dengan sintaks:

import java.util.Scanner;

Statement Input – Scanner

Metode-metode untuk objek Scanner :

Metode	Penjelasan
<code>nextByte()</code>	Membaca suatu integer bertipe byte
<code>nextShort()</code>	Membaca suatu integer bertipe short
<code>nextInt()</code>	Membaca suatu integer bertipe Int
<code>nextLong()</code>	Membaca suatu integer bertipe long
<code>nextFloat()</code>	Membaca suatu angka pecahan bertipe float
<code>nextDouble()</code>	Membaca suatu angka pecahan bertipe double
<code>next()</code>	Membaca suatu string yang berakhir dengan karakter spasi
<code>nextLine()</code>	Membaca sebaris teks (suatu string yang berakhir dengan enter)

Contoh Statement Input – Scanner

```
import java.util.*;  
public class Input_Output_Bilangan  
{  
    public static void main(String[] args)  
    {  
        int a; float b;  
        Scanner input=new Scanner(System.in);  
        System.out.print("Masukkan sebuah bilangan bulat = ");  
        a=input.nextInt();  
        System.out.print("Masukkan sebuah bilangan desimal = ");  
        b=input.nextFloat();  
        System.out.println("Bilangan bulat yang diinputkan adalah "+a);  
        System.out.println("Bilangan desimal yang diinputkan adalah "+b);  
    }  
}
```

Tanda * digunakan jika akan menggunakan semua class yang ada dalam paket java.util

Contoh Statement Input – Scanner

```
import java.util.*;

public class Input_Output_String
{
    public static void main(String[] args)
    {
        String a;

        Scanner input=new Scanner(System.in);
        System.out.print("Masukkan sebuah kalimat = ");
        a=input.next();

        System.out.println("Pembacaan string dengan next() : "+a);
        a=input.nextLine();

        System.out.print("Pembacaan string dengan nextLine() : "+a);

    }
}
```


Statement Input – BufferedReader

- Deklarasi umum penggunaan class BufferedReader adalah sebagai berikut:

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

Keterangan :

- ✓ A : adalah suatu variabel bertipe BufferedReader
- ✓ new BufferedReader (new InputStreamReader(System.in)) :
menciptakan suatu objek bertipe BufferedReader
- Setiap inputan dari keyboard, dianggap sebagai string.
Sehingga jika ingin menginputkan data berupa numerik, harus dilakukan konversi data terlebih dulu.

Statement Input – BufferedReader

- Untuk menggunakan BufferedReader perlu diimportkan terlebih dahulu library berikut agar berjalan dengan sempurna:
 - `import java.io.BufferedReader;`
 - `import java.io.InputStreamReader;`
 - `import java.io.IOException;`
- Java Application Programming Interface (API) sudah berisi ratusan kelas yang bisa digunakan untuk program. Kelas-kelas tersebut dikumpulkan ke dalam packages/paket.

Packages memiliki kelas yang mempunyai fungsi yang saling berhubungan. Seperti pada contoh diatas, java.io packages mengandung kelas-kelas yang memungkinkan program untuk melakukan input dan output data.

Statement diatas juga dapat ditulis : `import java.io.*;`

Statement Input – BufferedReader

❖ INPUT DATA STRING

Untuk menginputkan data berupa string harus digunakan method **readLine()**.

Bentuk deklarasinya :

```
var1 = A.readLine();
```

Dimana :

- ✓ var1 : nama variabel penampung data string
- ✓ A : nama variabel yang bertipe BufferedReader

Contoh Statement Input – BufferedReader

```
import java.io.*;
import java.util.*;
public class Input_Output_BuffRead {
    public static void main (String[] args) {
        int a; float b; String data;
        BufferedReader input = new BufferedReader (new InputStreamReader(System.in));
        try {
            System.out.print("Masukkan sebuah kalimat = ");
            data=input.readLine();
            System.out.println("Kalimat yang diinputkan adalah : "+data);
        }
        catch(Exception e) {
            System.out.println("Error pada saat input data");
        }
    }
}
```

Statement Input – BufferedReader

❖ INPUT DATA NUMERIK

Karena data yang diinputkan dianggap sebagai string, maka perlu adanya konversi tipe data, dari string menjadi numerik. Yaitu dengan menggunakan static method **valueOf()** atau **Parse()**.

Bentuk deklarasinya :

```
string → integer : var1=Integer.parseInt(var2) or var1=Integer.valueOf(var2)
string → short   : var1=Short.parseShort(var2) or var1=Short.valueOf(var2)
string → byte    : var1=Byte.parseByte(var2) or var1=Byte.valueOf(var2)
string → long    : var1=Long.parseLong(var2) or var1=Long.valueOf(var2)
string → float   : var1=Float.parseFloat(var2) or var1=Float.valueOf(var2)
string → double  : var1=Double.parseDouble(var2) or var1=Double.valueOf(var2)
```

Dimana :

- var1 : nama variabel penampung nilai setelah konversi
- var2 : nama variabel yang menampung nilai string sebelum konversi

Contoh Statement Input – BufferedReader

```
import java.io.*;
import java.util.*;
public class Input_Output_BuffRead {
    public static void main (String[] args) {
        int a; float b; String data;
        BufferedReader input = new BufferedReader (new InputStreamReader(System.in));
        try {
            System.out.print("Masukkan sebuah bilangan bulat = ");
            data=input.readLine();
            a=Integer.parseInt(data);
            System.out.print("Masukkan sebuah bilangan desimal = ");
            data=input.readLine();
            b=Float.parseFloat(data);
            System.out.println("Bilangan bulat yang diinputkan adalah : "+a);
            System.out.println("Bilangan desimal yang diinputkan adalah : "+b);
        }
        catch(Exception e) {
            System.out.println("Error pada saat input data");
        }
    }
}
```

Statement Input – BufferedReader

❖ INPUT DATA KARAKTER

Untuk membaca input berupa karakter, digunakan method **read()**.

Bentuk deklarasinya :

```
var1 = (char) A.read();
```

Dimana :

- ✓ var1 : nama variabel penampung data karakter
- ✓ A : nama variabel yang bertipe BufferedReader

Contoh Statement Input – BufferedReader

```
import java.io.*;
public class demo {
    public static void main(String[] args) {
        BufferedReader data = new BufferedReader(new InputStreamReader(System.in));
        char inisial = 0;
        try {
            inisial = (char) data.read();
        }
        catch (Exception e) {
            System.out.println("ERROR");
        }
        System.out.println(" Karakter = " + inisial);
    }
}
```