

Jobsheet Bahasa Pemrograman

06 Input / Output

A. TUJUAN

1. Siswa mampu memahami input dan output.
2. Siswa mampu menerapkan input dan output pada sebuah program.

B. DASAR TEORI

Input / Output

Seperti yang diketahui, program komputer terdiri dari tiga komponen utama, yaitu: input, proses, dan output.

- Input: nilai yang dimasukkan ke program
- Proses: langkah demi langkah yang dilakukan untuk mengelola input menjadi sesuatu yang berguna.
- Output: hasil pemrosesan/pengolahan.

Semua bahasa pemrograman telah menyediakan fungsi-fungsi untuk melakukan input dan output. Java sendiri sudah menyediakan tiga class untuk mengambil input:

- 1) *Class Scanner;*
- 2) *Class BufferedReader;*
- 3) dan *Class Console.*

Tiga class tersebut untuk mengambil input pada program berbasis teks (console). Sedangkan untuk output Java menyediakan fungsi `print()`, `println()`, dan `format()`.

1. Input dengan Class Scanner

Scanner merupakan *class* yang menyediakan fungsi-fungsi untuk mengambil input dari keyboard. Agar kita bisa menggunakan *Scanner*, kita perlu mengimpornya ke dalam kode: `import java.util.Scanner;`

Berikut ini adalah langkah-langkah yang perlu dilakukan untuk mendapatkan input dari console dengan *Scanner*:

1. Menambahkan

```
import java.util.Scanner;
```

2. Membuat objek dari class *Scanner* `Scanner input = new Scanner(System.in);`

3. Memasukan nilai dari masing-masing variabel dengan menggunakan object dari *Scanner*

```
System.out.print("Masukan nama : ");  
String nama = input.nextLine();  
System.out.println(nama);
```

Latihan 1

Ketiklah program berikut pada notepad atau lembar kerja pada netbeans dengan nama class DataSiswa.java, pembuatan class pada netbeans mengikuti langkah langkah pada jobsheet sebelumnya yang sudah dipelajari.

```
import java.util.Scanner;

public class DataSiswa {

    public static void main(String[] args) {
        // deklarasi variabel
        String nama, alamat;
        int usia, absen;
        // membuat scanner baru
        Scanner keyboard = new Scanner(System.in);
        // Tampilkan output ke user
        System.out.println("### DATA SISWA SMK TELKOM MALANG ###");
        System.out.print("Nama Siswa: ");
        // menggunakan scanner dan menyimpan apa yang diketik di variabel nama
        nama = keyboard.next();
        // Tampilkan output lagi
        System.out.print("Alamat: ");
        // menggunakan scanner lagi
        alamat = keyboard.next();
        System.out.print("Absen: ");
        usia = keyboard.nextInt();
        System.out.print("Usia: ");
        absen = keyboard.nextInt();
        // Menampilkan apa yang sudah simpan di variabel
        System.out.println("-----");
        System.out.println("Nama Siswa: " + nama);
        System.out.println("Alamat: " + alamat);
        System.out.println("Absen: " + usia);
        System.out.println("Usia: " + absen+ " tahun");
    }
}
```

Perhatikan: Penggunaan fungsi untuk mengambil data bergantung dari tipe data yang digunakan. Misal, tipe datanya adalah **String**, maka fungsi atau method yang dipakai adalah next().

Begitu juga dengan tipe data lain, **Integer** menggunakan nextInt(), **Double** menggunakan nextDouble(), dan sebagainya. Setelah selesai mengetik program, silahkan dijalankan.

2. Input dengan Class *BufferedReader*

Class BufferedReader sebenarnya tidak hanya untuk mengambil input dari keyboard, namun *class* ini juga dapat digunakan untuk membaca input dari file dan jaringan. *Class BufferedReader* ini terletak di dalam paket java.io. Untuk memakai class ini perlu diimpor classnya seperti halnya pada class Scanner.

```
import java.io.BufferedReader;
```

Berikut ini adalah langkah-langkah yang perlu dilakukan untuk mendapatkan input :

1. Tambahkan di bagian paling atas code

```
anda: import java.io.*;
```

2. Tambahkan statemen berikut pada method main:

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

3. Deklarasikan variabel String temporary dan gunakan fungsi `readLine()` untuk mendapatkan input serta ketikkan pada blog *try-catch*.

```
try {  
    String temp = dataIn.readLine();  
} catch ( IOException e ){  
    System.out.println("Terjadi kesalahan saat anda melakukan input ")  
}
```

Latihan 2

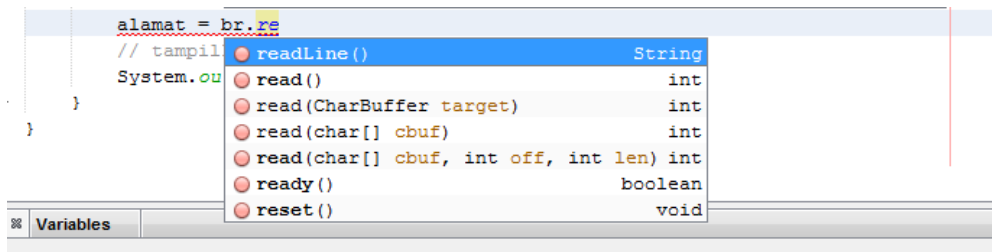
Ketiklah program berikut pada notepad atau lembar kerja pada netbeans dengan nama class `ContohBufferReader.java`, pembuatan class pada netbeans mengikuti langkah langkah pada jobsheet sebelumnya yang sudah dipelajari.

```
package Latihan;  
  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStreamReader;  
  
public class ContohBufferReader {  
    public static void main(String[] args) throws IOException {  
        String nama;  
        // Membuat objek inputstream  
        InputStreamReader isr = new InputStreamReader(System.in);  
        // membuat objek bufferreader  
        BufferedReader br = new BufferedReader(isr);  
        // Mengisi variabel nama dengan Bufferreader  
        System.out.print("Inputkan nama: ");  
        nama = br.readLine();  
        // tampilkan output isi variabel nama  
        System.out.println("Nama kamu adalah " + nama);  
    }  
}
```

Perhatikan: Class *BufferReader* tidak bisa bekerja sendirian, tapi juga butuh class yang lain yaitu: class *InputStreamReader* dan class *IOException*. Kedua class ini juga perlu diimport.

Perbedaan *BufferReader* dengan *Scanner* terlihat dari fungsi atau method yang dipakai. *Scanner* menggunakan *next()*, sedangkan *BufferReader* menggunakan *readLine()*.

Lalu untuk tipe data integer, *BufferReader* menggunakan fungsi *read()* saja. Hal ini dapat kita lihat dalam *hint autocomplete*. Tekan `Ctrl+Spasi` saat menulis kode untuk menampilkan *hint autocomplete*.



Setelah selesai mengetik program, Silahkan dijalankan programnya.



3. Input dengan *Class Console*

Class Console hampir sama dengan *BufferedReader*. Dia juga menggunakan fungsi `readLine()` untuk mengambil input. Namun *Class* ini hanya bisa digunakan di lingkungan console saja, seperti Terminal dan CMD. *Class Console* tidak bisa digunakan langsung di Netbeans. Maka dari itu, harus kompilasi secara manual. Untuk menggunakan class ini, perlu mengimpor terlebih dahulu.

```
import java.io.Console;
```

Latihan 3

Ketiklah program berikut pada notepad atau lembar kerja pada netbeans dengan nama class `ContohInputConsole.java`, pembuatan class pada netbeans mengikuti langkah langkah pada jobsheet sebelumnya yang sudah dipelajari.

```
package Latihan;
import java.io.Console;
public class ContohInputConsole {
    public static void main(String[] args) {
        String nama;
        int usia;
        String alamat;
        // membuat objek console
        Console con = System.console();
        // mengisi variabel nama usia alamat dengan console
        System.out.print("Inputkan nama anda: ");
        nama = con.readLine();
        System.out.print("Inputkan usia: ");
        usia = Integer.parseInt(con.readLine());
        System.out.print("Inputkan alamat anda: ");
        alamat = con.readLine();
        // menampilkan isi variabel nama usia dan alamat
        System.out.println("Nama kamu adalah: " + nama);
        System.out.println("Saat ini berusia " + usia + " tahun");
        System.out.println("Alamat kamu berada di: " + alamat);
    }
}
```

Perhatikan: kode di atas, kita menggunakan fungsi `Integer.parseInt(con.readLine())` untuk tipe data `integer`. Artinya, kita merubah tipe data `String` ke `Integer`.

Karena Console tidak memiliki nilai kembalian berupa integer untuk fungsi `read()`.

Setelah selesai mengetik program, Silahkan dijalankan programnya dengan masuk ke direktori tempat menyimpan kodenya. Ketik perintah: `javac ContohInputConsole.java` untuk melakukan kompilasi, kemudian ketik perintah `java InputConsole` untuk menjalankannya.



4. Menampilkan Output

Kita sudah mengenal beberapa cara mengambil input dari keyboard untuk program berbasis teks. Kemudian bagaimana dengan outputnya??

Ada beberapa fungsi yang sudah disediakan oleh Java untuk output sebuah program:

1. Fungsi `System.out.print()`
2. Fungsi `System.out.println()`
3. Fungsi `System.out.format()`

Apa saja perbedaan dari fungsi-fungsi tersebut?

Fungsi `print()` vs `println()`

Fungsi `print()` dan `println()` sama-sama digunakan untuk menampilkan teks.

Fungsi **`print()`** akan menampilkan teks apa adanya. Sedangkan **`println()`** akan menampilkan teks dengan ditambah baris baru.

Sedangkan untuk menggabungkan String yang lebih kompleks, kita bisa menggunakan fungsi **`format()`**.

Latihan 4

Ketiklah program berikut pada notepad atau lembar kerja pada netbeans dengan nama class

`ContohPrintvsPrintln.java`.

```
6 package Latihan;
7
8 public class ContohPrintvsPrintln {
9     public static void main(String[] args) {
10
11         System.out.print("Ini teks yang dicetak dengan print()");
12         System.out.println("Sedangkan ini teks yang dicetak dengan println()");
13         System.out.print("Teks dicetak dengan print() lagi");
14     }
15 }
```

Setelah selesai mengetik program, Silahkan dijalankan programnya.

Latihan 5

Ketiklah program berikut pada notepad atau lembar kerja pada netbeans dengan nama class ContohFormatString.java.

```
6 package Latihan;
7
8 public class ContohFormatString {
9     public static void main(String[] args) {
10         String namaDepan = "Andi";
11         String namaBelakang = "Bagus";
12         String namaakhir = "Prasetya";
13         System.out.format("Nama saya %s %s %s", namaDepan, namaBelakang, namaakhir);
14     }
15 }
```

Setelah selesai mengetik program, Silahkan dijalankan programnya.

Latihan 6

Ketikkan dan simpan program berikut. Program berikut akan menerima inputan dari keyboard, dan menyimpannya ke dalam suatu variabel, dan kemudian menampilkannya ke layar. Untuk menggunakan masukan keyboard, digunakan library Scanner, dari paket java.util. Ketik dan jalankan program berikut.

```
public class Inputan {
    public static void main (String[] args) {
        Scanner masukan = new Scanner(System.in);
        int a, b;
        System.out.print("Masukkan nilai a :");
        a = masukan.nextInt();
        System.out.print("Masukkan nilai b :");
        b = masukan.nextInt();
        System.out.println();
        System.out.println("Nilai Variabel yang ada pada program :")
        System.out.println("a = " + a);
        System.out.println("b = " + b);
    }
}
```

Output Program

Jelaskan perbedaan `System.out.print` dan `System.out.println` pada program di atas!

Latihan 7

Program ini untuk menentukan suatu kelulusan siswa terhadap suatu matapelajaran. Program tersebut menggunakan pernyataan If untuk penentuan hasil akhir. Ketik dan jalankan program berikut!

```
import java.util.Scanner;

public class Kondisi{
    public static void main (String[] args) {
        Scanner masukan = new Scanner(System.in);
        int nilai;
        System.out.print("Masukkan nilai akhir PBO :");
        nilai = masukan.nextInt();
        if (nilai < 70)
            System.out.println("Siswa tidak lulus");
        if (nilai > 70)
            System.out.println("Siswa lulus");
    }
}
```

Hasil Program

Latihan 8

Tulis dan simpan program berikut.

```
import java.io.*;

public class Persegi {
    public static void main(String[] args) {
        BufferedReader data = new BufferedReader(new InputStreamReader(System.in));
        String x = " ";
        int hitungLuas=0;
        System.out.println("---Luas Persegi---");
        System.out.print("Masukkan nilai sisi : ");

        try {
            x = data.readLine();
            int angka = Integer.parseInt(x);
            hitungLuas = angka * angka;
        } catch (Exception e) {
            System.out.println("Error");
        }

        System.out.println("Luas persegi dengan sisi " + x + " adalah " + hitungLuas);
    }
}
```

Hasil program



C. Tugas Praktikum

Buatlah program kalkulator aritmatika sederhana menggunakan Java.

**Semangat itu ada
Ketika kita punya tujuan
Jangan pernah merasa tidak mampu
Ketika belum mencoba
--Siswa Telkom Luar Biasa--**