



MODUL PERKULIAHAN

PEMROGRAMAN IV (Object I)

**Analisis Top-Down (Runtunan/Sequential), Input
& Output.**

Fakultas
Teknik

Program Studi
Informatika

Tatap Muka

07

Kode MK
06610005

Disusun Oleh

Ardiles Sinaga, S.T., M.T.
Kurnia Jaya Eliazar, S.T., M.T.

Abstract

Modul Pertemuan 06 Berisi Mengenai control structure pada java terutama yang berkaitan dengan Analisis Top-Down (Runtunan/Sequential), Input & Output.

Kompetensi

Mahasiswa memiliki kemampuan untuk menerapkan control structure pada java terutama yang berkaitan dengan Analisis Top-Down (Runtunan/Sequential), Input & Output.

Pertemuan 06 – Analisis Top-Down (Runtunan/Sequential), Input & Output, Pemilihan (Selection), Looping.

Struktur dan Aturan Penulisan Sintaks Java

Setiap bahasa pemrograman memiliki struktur dan aturan penulisan sintaks yang berbeda-beda.

Java merupakan bahasa pemrograman yang dikembangkan dari bahasa C dan tentunya akan banyak mengikuti gaya penulisan C.

Saat baru pertama kali melihat program Java, mungkin kamu akan bertanya-tanya.

Ini apa? itu apa?

Contoh:

Coba perhatikan program berikut:

```
package com.program.test;

class Test {
    public static void main(String args[]){
        System.out.println("Hello World");
    }
}
```

Banyak hal yang kita belum ketahui.

Apa itu *package*?

Apa itu *class*?

...dan mengapa ditulis seperti itu?

Oleh sebab itu, kita perlu belajar sintaks dasar dan struktur program Java.

Struktur Dasar Program Java

Struktur program Java secara umum dibagi menjadi 4 bagian:

1. Deklarasi Package
2. Impor Library
3. Bagian Class
4. Method Main

Mari kita lihat contohnya:

```
package com.petanikode.program; //<- 1. deklarasi package

import java.io.File; //<- 2. Impor library

class Program { //<- 3. Bagian class

    public static void main(String args[]){ //<- 4. Method main
        System.out.println("Hello World");
    }
}
```

1. Deklarasi Package

Package merupakan sebuah folder yang berisi sekumpulan program Java.

Deklarasi package biasanya dilakukan saat membuat program atau aplikasi besar.

Contoh deklarasi package:

```
package com.petanikode.program;
```

Biasanya nama package mengikuti nama domain dari sebuah vendor yang mengeluarkan program tersebut.

Pada contoh di atas, com.petanikode adalah nama domain dari petani kode.

Aturannya: nama domain dibalik, lalu diikuti nama programnya.

Bagaimana kalau kita tidak mendeklarasikan package?

Boleh-boleh saja dan programnya akan tetap bisa jalan.

Kita wajib mendeklarasikan package.

2. Bagian Impor

Pada bagian ini, kita melakukan impor library yang dibutuhkan pada program.

Library merupakan sekumpulan class dan fungsi yang bisa kita gunakan dalam membuat program.

Contoh impor library:

```
import java.io.File;
```

Pada contoh tersebut, kita mengimpor class *Scanner* dari package *java.util*

3. Bagian Class

Java merupakan bahasa pemrograman yang menggunakan paradigma OOP (*Object Oriented Programming*).

Setiap program harus dibungkus di dalam class agar nanti bisa dibuat menjadi objek.

Kalau kamu belum paham apa itu OOP?

Cukup pahami class sebagai deklarasi nama program.

```
class NamaProgram {  
    public static void main(String args[]){  
        System.out.println("Hello World");  
    }  
}
```

Ini adalah blok class.

Blok class dibuka dengan tanda kurung kurawal { kemudian ditutup atau diakhiri dengan }.

Di dalam blok class, kita dapat mengisinya dengan method atau fungsi-fungsi dan juga variabel.

Pada contoh di atas, terdapat method *main()*.

4. Method Main

Method *main()* atau fungsi *main()* merupakan blok program yang akan dieksekusi pertama kali.

Ini adalah entri point dari program.

Method *main()* wajib kita buat. Kalau tidak, maka programnya tidak akan bisa dieksekusi.

Contoh method *main()*.

```
public static void main(String args[]){  
    System.out.println("Hello World");  
}
```

Penulisannya harus seperti ini...

Method *main()* memiliki parameter *args[]*. Parameter ini nanti akan menyimpan sebuah nilai dari argumen di command line.

Lalu di dalam method *main()*, terdapat statement atau fungsi:

```
System.out.println("Hello World");
```

Ini adalah fungsi untuk menampilkan teks ke layar monitor.

Statement dan Ekspresi pada Java

Statement dan ekspresi adalah bagian terkecil dalam program. Setiap statement dan ekspresi di Java, harus diakhiri dengan titik koma (;).

Contoh statemen dan ekspresi:

```
System.out.println("Hello World");  
System.out.println("Apa kabar?");  
var x = 3;  
var y = 8;  
var z = x + y;
```

Statemen dan ekspresi akan menjadi instruksi yang akan dikerjakan oleh komputer.

Pada contoh di atas, kita menyuruh komputer untuk menampilkan teks *"Hello World"*, dan *"Apa kabar?"*.

Lalu kita menyuruhnya untuk menghitung nilai $x + y$.

Blok Program Java

Blok program merupakan kumpulan dari statement dan ekspresi yang dibungkus menjadi satu.

Blok program selalu dibuka dengan kurung kurawal { dan ditutup dengan }.

Contoh blok program:

```
// blok program main
public static void main(String args[]){
    System.out.println("Hello World");
    System.out.println("Hello Kode");

    // blok program if
    if( true ){
        System.out.println('True');
    }

    // blok program for
    for ( int i = 0; i<10; i++){
        System.out.println("Perulangan ke"+i);
    }
}
```

Intinya: jika kamu menemukan kurung { dan }, maka itu adalah sebuah blok program.

Blok program dapat juga berisi blok program yang lain (nested).

Pada contoh di atas, blok program main() berisi blok if dan for.

Penulisan Komentar pada Java

Komentar merupakan bagian program yang tidak akan dieksekusi oleh komputer.

Komentar biasanya digunakan untuk:

- Memmmberi keterangan pada kode program;
- Menonaktifkan fungsi tertentu;
- Membuat dokumentasi;
- dll.

Penulisan komentar pada java, sama seperti pada bahasa C. Yaitu menggunakan:

1. Garis miring ganda (//) untuk komentar satu baris;
2. Garis miring bintang (/*...*/) untuk komentar yang lebih dari satu baris.

Contoh:

```
public static void main(String args[]){
    // ini adalah komentar satu baris
```

```
System.out.println("Hello World");

// komentar akan diabaikan oleh komputer
// berikut ini fungsi yang di-non-aktifkan dengan komentar
// System.out.println("Hello World");

/*
Ini adalah penulisan komentar
yang lebih dari
satu baris
*/
}
```

Input dan Output

Output

Untuk menampilkan tulisan pada terminal / console dapat menggunakan perintah :

System.out.print(...);

System.out.println(...);

Input

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

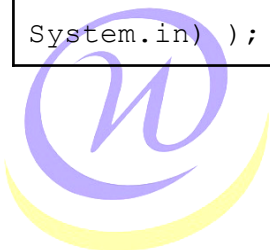
Berikut ini adalah langkah-langkah yang diperlukan untuk mendapatkan input dari keyboard:

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 + "!");
```



Daftar Pustaka

- [1] Deitel P.J., Deitel H.M., “Java: How to Program”, Prentice Hall, 2004
- [2] Holmes B. J., Joyce D.T., “Object-Oriented Programming With Java, Second Edition”, JONES AND BARTLETT PUBLISHERS, 2001
- [3] Keogh J., “JAVA DEMYSTIFIED”, McGraw-Hill/Osborne, 2004
- [4] Wu C.T., “AN INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING WITH JAVA™, FIFTH EDITION”, McGraw-Hill, 2010
- [5] Wu C.T., “A COMPREHENSIVE INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING WITH JAVA”, McGraw-Hill, 2008

- [6] Poo D., Kiong D., Ashok S., “Object-Oriented Programming and Java Second edition”, Springer, 2008
- [7] Sintes T., “Sams Teach Yourself Object Oriented Programming in 21 Days”, Sam Publishing, 2002

