

Materi 1 : Pengenalan Java, Output, Tipe Data & Variabel, Operator, dan Input

DAFTAR ISI

1.1	Pengenalan Java?	1
1.2	Yang Harus Disiapkan	1
1.2.1	JDK (Java Development Kit)	1
1.2.1	IDE	2
1.3	Cara Menjalankan Java	2
1.4	Method Output	3
1.5	Tipe Data	3
1.5.1	Primitif	3
1.5.2	Komposit	3
1.6	Variabel	5
1.7	Method Input	6
1.7.1	BufferedReader	6
1.7.2	Scanner	7
1.8	Operator	8

1.1 Pengenalan Java?



Java adalah bahasa pemrograman berorientasi objek yang dirancang oleh **James Gosling** dan **Sun Microsystems** (sekarang Oracle Corporation) pada tahun 1995. Java adalah bahasa pemrograman yang populer dan banyak digunakan untuk mengembangkan aplikasi web, aplikasi desktop, dan aplikasi seluler. Sintak java hampir sama dengan bahasa pemrograman C/C++ karena java terinspirasi dari kedua bahasa tersebut.

Bahasa ini lahir ketika perusahaan Sun Microsystems merasa bahwa bahasa C/C++ tidak lagi mencukupi kebutuhan proyek yang sedang mereka kerjakan. Java pada awalnya bernama **Oak** yang terinspirasi dari pohon oak yang ada di depan kantor James Gosling. Namun karena nama oak sudah terdaftar sebagai merek dagang di perusahaan lain, maka James Gosling menggantinya dengan nama “**Java**” yang terinspirasi dari “**Java Coffee**” atau kopi Jawa. Karena itulah logo bahasa Java berupa gambar secangkir kopi.

1.2 Yang Harus Disiapkan

Berikut ini adalah beberapa hal yang harus disiapkan (*install*) untuk belajar pemrograman java:

1.2.1 JDK (Java Development Kit)

JDK adalah paket perangkat lunak yang berisi semua yang Anda butuhkan untuk mengembangkan dan menjalankan aplikasi Java. JDK mencakup Java Virtual Machine (JVM), Java Compiler, Java Debugger, dan alat-alat lain yang diperlukan untuk mengembangkan aplikasi Java.

Berikut ini adalah langkah-langkah untuk menginstall JDK:

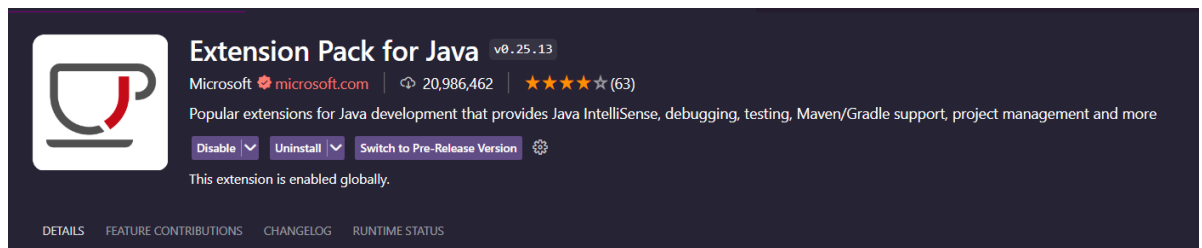
1. Buka website <https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>, kemudian pilih versi LTS (Long-Term Support) yang terbaru yaitu Java SE Development Kit 17.0.8, dan pilih Windows x64 Installer.

Linux x64 Debian Package	149.34 MB	https://download.oracle.com/java/17/archive/jdk-17.0.8_linux-x64_bin.deb (sha256)
Linux x64 RPM Package	173.66 MB	https://download.oracle.com/java/17/archive/jdk-17.0.8_linux-x64_bin.rpm (sha256)
macOS Arm 64 Compressed Archive	168.12 MB	https://download.oracle.com/java/17/archive/jdk-17.0.8_macos-aarch64_bin.tar.gz (sha256)
macOS Arm 64 DMG Installer	167.55 MB	https://download.oracle.com/java/17/archive/jdk-17.0.8_macos-aarch64_bin.dmg (sha256)
macOS x64 Compressed Archive	170.56 MB	https://download.oracle.com/java/17/archive/jdk-17.0.8_macos-x64_bin.tar.gz (sha256)
macOS x64 DMG Installer	169.98 MB	https://download.oracle.com/java/17/archive/jdk-17.0.8_macos-x64_bin.dmg (sha256)
Windows x64 Compressed Archive	172.38 MB	https://download.oracle.com/java/17/archive/jdk-17.0.8_windows-x64_bin.zip (sha256)
Windows x64 Installer	153.48 MB	https://download.oracle.com/java/17/archive/jdk-17.0.8_windows-x64_bin.exe (sha256)
Windows x64 MSI Installer	152.27 MB	https://download.oracle.com/java/17/archive/jdk-17.0.8_windows-x64_bin.msi (sha256)

2. Jika sudah menginstall, maka klik tombol next sampai selesai kemudian close.
3. Untuk mengetahui apakah JDK sudah terinstall atau belum. Maka perlu membuka Command Prompt kemudian ketik perintah `java -version`. Jika JDK sudah terinstall, maka akan muncul versi JDK yang terinstall. Jika JDK belum terinstall, maka akan menunjukkan pesan kesalahan.

1.2.1 IDE

IDE yang akan kita gunakan adalah vscode dan intelliJ. Namun penggunaan intelliJ akan digunakan setelah UTS. Jika sudah menginstall vscode, maka perlu menginstall Extension Pack for Java.



1.3 Cara Menjalankan Java

Untuk memudahkan cara menjalankan java, ikutilah langkah-langkah berikut ini:

1. Buatlah sebuah file dengan nama `Main.java`
2. Kemudian tambahkan kode berikut ini ke dalam file `Main.java`

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

3. Jika sudah menambahkan kode di atas, maka save file tersebut
4. Jalankan perintah berikut di terminal

```
javac Main.java
```

5. Kode di atas berguna untuk mengkompilasi kode java ke bytecode. Kemudian jalankan perintah di bawah ini untuk menjalankan program java

```
java Main
```

6. Selamat Anda sudah menjalankan program pertama!

1.4 Method Output

Method output di java adalah fungsi yang digunakan untuk mencetak output ke konsol. Method output yang paling umum adalah `System.out.println()`. Method ini memiliki satu argumen yang dapat berupa teks atau objek dan menambahkan baris baru di ujungnya.

Selain `System.out.println()`, ada beberapa method lainnya di java, yaitu:

1. `System.out.print()`, method ini digunakan untuk mencetak teks ke konsol tanpa menambah baris baru. Contoh:

```
System.out.print("Hello, World!");  
System.out.print("Hello Universe!");
```

2. `System.out.printf()`, method ini digunakan untuk mencetak teks ke konsol dengan format tertentu. Contoh:

```
int age = 20;  
System.out.printf("Umur saya %d tahun", age);
```

1.5 Tipe Data

Ada beberapa 2 tipe data yang ada di java, yaitu :

1.5.1 Primitif

Adalah tipe data yang langsung disimpan oleh komputer. Contohnya adalah `byte`, `short`, `int`, `long`, `float`, `double`, `char`, dan `boolean`. Contoh penggunaannya adalah sebagai berikut:

```
char golonganDarah = 'A';  
int umur = 20;  
double tinggi = 1.74;  
boolean menikah = true;
```

1.5.2 Komposit

Adalah tipe data yang merupakan objek, yaitu `String`, `Array`, `Object` dan `Class`.

1. String

Penulisan string pada java harus diawali dengan huruf S besar "`String`", hal ini disebabkan karena String merupakan sebuah kelas di java. Berikut adalah contoh penggunaan string.

```
String nama = "Dimas Firmansyah";
```

String dapat dimanipulasi dengan berbagai method berikut ini:

- `charAt()`, untuk mengambil karakter tertentu pada string
- `length()`, untuk menghitung panjang string
- `concat()`, untuk menggabungkan 2 string
- `toLowerCase()`, untuk mengubah semua huruf dalam string menjadi huruf kecil
- `toUpperCase()`, untuk mengubah semua huruf dalam string menjadi huruf besar
- `equalsIgnoreCase()`, untuk memeriksa apakah 2 string sama, terlepas dari huruf besar ataupun kecil
- `equals()`, untuk memeriksa apakah 2 string sama
- dll.

2. Array

Adalah struktur data yang menyimpan sekumpulan data dengan tipe yang sama. Array di Java direpresentasikan menggunakan kelas Array. Contoh penggunaannya adalah:

```
int[] arr = { 1, 2, 3, 4, 5 };

int[] number = new int[5];
number[0] = 10;
number[1] = 20;
number[2] = 30;
System.out.println(number[0]);
```

Array dapat dimanipulasi dengan berbagai method seperti:

- `sort()`, untuk mengurutkan elemen array
- `copyOf()`, untuk menyalin sebuah array ke dalam array baru
- `fill()`, untuk mengisi array dengan nilai yang sama
- `binarySearch()`, untuk mencari nilai dalam array menggunakan pencarian biner
- `equals()`, untuk membandingkan dua array

Tipe data array memiliki kelemahan yaitu ukurannya tidak dapat berubah. Untuk mengatasi hal tersebut maka java membuat sebuah struktur data bernama ArrayList yang bisa menampung sekumpulan objek dengan tipe data yang sama, ukurannya lebih fleksibel dan bisa diubah sesuai kebutuhan. Berikut contoh penggunaan ArrayList:

```
// membuat ArrayList
ArrayList<String> listName = new ArrayList<>();
```

```
// menambahkan nilai ke elemen ArrayList
listName.add("Dilan");
listName.add("Milea");

// mengambil nilai ArrayList
System.out.println(listName.get(0));
System.out.println(listName.get(1));
```

ArrayList juga bisa dimanipulasi dengan berbagai method seperti:

- **add()**, digunakan untuk menambahkan objek
- **remove()**, digunakan untuk menghapus sebuah objek
- **get()**, untuk mengambil sebuah objek berdasarkan nilainya
- **set()**, untuk menetapkan objek baru ke indeks yang sudah ditentukan (update)
- **size()**, untuk mendapatkan ukuran ArrayList
- **clear()**, untuk menghapus semua objek dari ArrayList
- **isEmpty()**, digunakan untuk memeriksa apakah ArrayList kosong atau tidak
- **contains()**, digunakan untuk memeriksa apakah objek tertentu ada di dalam ArrayList atau tidak
- dll.

3. Object & Class

Meteri object dan class akan dibahas dipertemuan OOP.

1.6 Variabel

Di java terdapat beberapa jenis variabel berdasarkan cakupan, karakteristik, dan tujuan penggunaannya. Berikut ini adalah beberapa jenis variabel dan contohnya:

1. Variabel Lokal

Variabel lokal dideklarasikan di dalam blok kode seperti metode, konstruktor, atau blok lainnya. Variabel ini hanya bisa diakses di dalam blok tempat mereka dideklarasikan.

```
public class Main {
    public static void main(String[] args) {
        int number = 10; //variabel lokal
        System.out.println(number);
    }
}
```

2. Variabel Final

Variabel final adalah variabel yang nilainya tidak dapat diubah setelah di inisialisasi. Ini bisa digunakan untuk membuat konstanta.

```

public class Main {
    final int number = 0; // variabel final

    public static void main(String[] args) {
        Main example = new Main();
        System.out.println(example.number);
    }
}

```

3. Variabel Kelas & Instance

Materi ini akan dijelaskan di bagian OOP

1.7 Method Input

Di java, terdapat 2 cara untuk menerima inputan dari user yaitu Scanner dan BufferedReader. Scanner adalah class yang lebih baru dan lebih powerful, sedangkan BufferedReader adalah class yang lebih lama dan lebih sederhana.

1.7.1 BufferedReader

BufferedReader hanya bisa digunakan untuk membaca input dari file dan sifatnya yang lebih sederhana dibandingkan dengan Scanner sehingga cocok untuk tugas-tugas sederhana. Berikut contoh penggunaannya.

```

// yang harus di import
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader input =
            new BufferedReader(new InputStreamReader(System.in));
        String nama;
        System.out.print("Masukan nama : ");
        nama = input.readLine();
        System.out.println("Selamat datang " + nama);
    }
}

```

Karena hasil dari buffer adalah string, maka wajib melakukan parsing (mengubah tipe data) ke tipe data yang sesuai.

Konversi Ke-	Cara Konvensi
boolean	<code>Boolean.parseBoolean(String)</code>

float	<code>Float.parseFloat(String)</code>
double	<code>Double.parseDouble(String)</code>
byte	<code>Byte.parseByte(String)</code>
short	<code>Short.parseShort(String)</code>
int	<code>Integer.parseInt(String)</code>
long	<code>Long.parseLong(String)</code>

1.7.2 Scanner

Scanner dapat digunakan untuk menerima input dari berbagai sumber termasuk keyboard, file, dan URL. Scanner memiliki banyak method dibanding dengan `BufferedReader` sehingga scanner lebih fleksibel dan dapat digunakan untuk berbagai tugas. Berikut contoh penggunaannya.

```
// yang harus di import
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Nama : ");
        String nama = input.nextLine();
        System.out.println("Selamat Datang " + nama);
    }
}
```

Untuk melakukan inputan ke tipe data selain string, maka kita harus menggunakan beberapa method berikut:

Method	Penjelasan
<code>nextByte()</code>	Membaca tipe data byte
<code>nextShort()</code>	Membaca tipe data short
<code>nextInt()</code>	Membaca tipe data integer
<code>nextLong()</code>	Membaca tipe data long
<code>nextFloat()</code>	Membaca tipe data float

<code>nextDouble()</code>	Membaca tipe data double
<code>next()</code>	Membaca sebaris string
<code>nextLine()</code>	Membaca sebaris string dengan baris baru di akhir

1.8 Operator

Semua operator di java sama dengan operator yang ada di bahasa C, sehingga kalian bisa mencari tahu atau mengingat kembali materi yang telah diajarkan oleh para instruktur.