**Java**

**Catatan Code Java…**

# Java Dasar

## Sejarah Java

Java dibuat oleh *James Gosling* saat bekerja di *Sun Microsystem*, java mulai dirilis public saat tahun 1995.

Java adalah Bahasa pemrograman berorientasi object dan mendukung pengelolaan memori secara otomatis.

Java terkenal dengan *write once*, *run anywhere*, karena binary program Java di generate secara independent dan bisa dijalankan di *Java Virtual Machine* yang terinstal di berbagai system operasi.

**Teknologi Java**

* Java Standard Edition
* Java Enterprise Edition
* Java Micro Edition

**JRE** singkatan dari Java Runtime Environment (Jika aplikasi-nya sudah jadi dan ingin dijalankan).

JRE biasanya disimpan di server production.

**JDK** singkatan dari Java Development Kit (Untuk melakukan Development Aplikasi Java & Compiler java terdapat di dalam JDK).

**Java Virtual Machine**

Otak dibalik java sebenarnya adalah *Java Virtual Machine* yang merupakan program yang digunakan untuk mengeksekusi binary file Java. JVM hanya mengerti binary file sehingga sudah terbukti stabil dan bagus.

**Proses Development Java**

A diagram of a computer program

Description automatically generated

## Instalasi Java (JDK)

<https://openjdk.org/> merupakan salah satu implementasi JDK yang open-source dan gratis.

Terdapat alternatif lain selain OpenJDK yaitu Oracle JDK, Amazon Corrreto, dan Zulu.

Download pada url berikut <https://jdk.java.net/>

Ekstrak, dan simpan di C:/

Selanjutnya masuk ke dalam path bin C:\jdk-22.0.1\bin, dan buka settingan *environment* pada bagian system variable klik *new* dengan nama variabelnya *JAVA\_HOME* dan value nya berisi path jdk.

Selanjutnya tambahkan juga di menu *path* lalu klik edit, lalu tambahkan %JAVA\_HOME%\bin atau C:\jdk-22.0.1\bin. Ini untuk meregistrasikan folder bin di dalam jdk ke dalam environment variable path tersebut. Kegunaanya agar binary di dalam folder bin dapat diakses melalui terminal.

Test version

*java –version* = digunakan untuk mengecek version java / JRE.

*javac --version* = java compiler untuk memastikan java compiler sudah terinstal dengan baik.

**Integreted Development Environment (IDE)**

IDE adalah smart editor yang digunakan untuk mengedit kode program.

IDE juga digunakan untuk melakukan otomatisasi proses kompilasi kode program dan otomatisasi proses menjalankan program.

IDE untuk JAVA :

* Intelli JDEA Ultimate / Community
* Eclipse
* NetBeans
* JDeveloper

## Hello World

A screen shot of a computer program

Description automatically generated

Karena disimpan di dalam folder java\_dasar maka kita harus panggil sebuah *package java\_dasar;*

Untuk membuat sebuah program Hello World kita awali dengan membuat sebuah *class*. Di Java nama *class* harus sama dengan nama file.

Kita harus membuat sebuah ***main function*** Ketika ingin menjalankan Java, dan pembuatannya harus seperti itu *‘ public static void main(String[] args) { .. } ’*

Untuk menampilkan text kita gunakan

*System.out.println(‘Hello World’);* = *ln* membuat baris baru.

*System.out.print(‘Hello World');* = menambhakan di baris yang sama.

Kompilasi Kode Java

Cara 1 :

* javac HelloWorld.java
* java HelloWorld

Dengan begini kita melakukan sebuah kompilasi pada file HelloWorld.java sehingga nanti akan dibuat bytecode dan menghasilkan sebuah file .class

Oleh karena itu kita bisa langsung memanggil HelloWorld kode akan langsung dieksekusi.

Cara 2 :

* java HelloWorld.java

Proses ini melakukan kompilasi oleh JVM secara langsung ke bytecode tanpa menghasilkan file class dan kemudian langsung mengeksekusinya.

## Tipe Data Number

**Integer Number**

A table with numbers and a number

Description automatically generated

A computer screen shot of numbers and lines

Description automatically generatedKita bisa memberikan sebuah Keyword ***L*** pada long sebagai menandakan bahwa variable tersebut menggunakan tipe data *long.*

**Floating Point Number**

**A close-up of a number

Description automatically generated**

**A number and lines on a black background

Description automatically generated**Float juga sama seperti *long* kita bisa memberikan keyword ***F*** sebagai penanda bahwa itu adalah tipe data float.

**Literals Number**

A computer screen shot of a number

Description automatically generatedLiterals number ini biasanya digunakan saat kondisi tertentu.

Misalkan jika kita ingin membuat sebuah hexa decimal maka diawali dengan ***0x***. Sedangkan untuk bilangan binary dapat diawali dengan ***0b.***

**Underscore Number**

**A number and numbers on a black background

Description automatically generated**Terkadang didunia nyata pemisah angka biasanya menggunakan ‘.’ (titik) di java pemisah angka dapat dilakukan dengan menggunakan *underscore (\_).*

### Konversi Tipe Data

Widening Casting (Otomatis) : byte -> short -> int -> long -> float -> double

Narrowing Casting (Manual) : double -> float -> long -> int -> short -> byte

A screenshot of a computer program

Description automatically generatedKenapa Narrowing dilakukan secara manual ? karena belum tentu nilai yang awalnya memiliki angka yang besar dapat dikonversi menjadi nilai yang kecil.

*\*Jika konversi nilainya tidak sesuai dengan max / min dari tipe data number maka akan terjadi ‘****number overflow****’, jadi setelah dikonversi akan balik lagi kebelakang.*

## Tipe Data Character

Data karakter (huruf) di java direpresentasikan oleh tipe *char*.

Untuk membuat char di java kita bisa gunakan (petik satu) di awal dan diakhir character.

A computer screen shot of a computer code

Description automatically generated

## Tipe Data Boolean

Boolean di java direpresentasikan dengan nama *boolean*. Default nilai dari Boolean di java adalah false.

A computer screen shot of a number of text

Description automatically generated with medium confidence

## Tipe Data String

Tipe data String adalah tipe data yang berisikan karakter atau sederhananya adalah teks. Di java ini direpresentasikan dengan *String*. Default nilai String adalah null dan untuk membuatnya pastikan menggunakan petik 2.

A computer code with colorful text

Description automatically generated

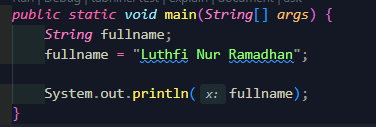
*Untuk menggabungkan lebih dari 1 string dapat menggunakan* ***+*** .

## Variabel

Variabel adalah tempat untuk menyimpan data.

Java adalah Bahasa yang static type, sehingga sebuah variable hanya bisa digunakan untuk menyimpan tipe data yang sama, tidak bisa berubah ubah.

Untuk membuat variable di Java kita bisa menggunakan nama tipe data lalu diikuti dengan nama variabelnya.

**

### Keyword Var

Java sekarang sudah mendukung kata kunci var sehingga kita tidak perlu menyebutkan tipe datanya.

Namun perlu diingat, saat kita membuat kata kunci var untuk membuat variable, kita harus menginisialisasi value secara langsung.

A screen shot of a computer code

Description automatically generated

### Keyword Final

Secara default variable datanya dapat diubah ubah nilainya.

Jika kita ingin membuat sebuah variable yang datanya tidak boleh diubah ubah setelah pertama kali dibuat kita bisa menggunakan kata kunci *Final (constant).*

A computer screen with colorful text

Description automatically generated

## Tipe Data Bukan Primitif

Tipe data primitive adalah tipe data bawaan Bahasa pemorgraman. Tipe data ini tidak bisa diubah lagi.

Tipe data Number, Char, Boolean adalah tipe data primitive. Tipe data primitive selalu memiliki default value. Sedangkan String bukan termasuk tipe data primitive, Tipe data yang bukan primitive tidak memiliki default value dan bisa bernilai Null.

*Jika data bukan primitive bisa memiliki sebuah method / function.*

Di Java, semua tipe data primitive memiliki representasi tipe data bukan primitive nya.

Sebelah kiri merupakan tipe data primitive sedangkan disebelah kanan merupakan tipe data bukan primitive adalah ***object***.

A screenshot of a computer

Description automatically generated

A graph with red line

Description automatically generated with medium confidence

*A screen shot of a computer code

Description automatically generated*

### Konversi ke Bukan Primitif

Jika tipe data awal kita primitive dan ingin menggunakan Object dari tipe data yang bukan primitive, kita bisa lakukan seperti berikut.

A computer screen shot of a math equation

Description automatically generated

Bagaimana jika ingin konversi ke Object bukan primitive lainnya?

A computer screen shot of a code

Description automatically generatedPemanggilan method menggunakan .

Untuk melakukan konversi dapat kita panggil method *namaTipeDataValue()* untuk mengkonversi value nya.

## Tipe Data Array

Array adalah tipe data yang berisikan Kumpulan data dengan *tipe data yang sama*, jumlah data di array tidak bisa diubah sejak pertama kali dibuat.

A computer screen shot of a program code

Description automatically generatedDiawali dengan nama tipe data nya, kemudian inisialisasi datanya dan berikan jumlah data yang bisa ditampung.

### Array Initializer

A computer screen shot of a code

Description automatically generatedCara membuat array tidak hanya seperti itu kita bisa membuat array dengan beberapa cara :

Berikut adalah cara pertama yang sebelumnya dilakukan

A computer screen shot of a code

Description automatically generated

Cara ke-dua ini hampir sama dengan yang pertama hanya saja tidak memberikan Batasan pada isi array dan membuat isi data nya dengan menggunakan ***{ }***

A screenshot of a computer code

Description automatically generatedCara ke-tiga ini adalah cara yang paling simple karena kita tidak perlu melakukan inisialisasi Kembali pada tipe data arraynya, dan langsung dapat membuat isi array secara langsung.

*Di java tidak ada istilah menghapus data array, jika kita menggunakan Object maka kita bisa secara langsung memberikan nilai null agar mengkosongkan nilainya. Jika menggunakan tipe data bukan primitive maka kita bisa mengisikan nilainya menjadi 0.*

### Array in Array

Ketika kita ingin membuat array di dalam array, caranya mirip dengan cara sebelumnya.

A screen shot of a computer code

Description automatically generated

## Operasi Matematika

A screenshot of a computer

Description automatically generated

### Augmented Assignment

A table with text and numbers

Description automatically generated

A screenshot of a computer program

Description automatically generated

### Unary Operator

A screenshot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generatedPerlu diingat antara perbedaan a++ dengan ++a

Jika a++ biasanya dilakukan saat perulangan karena akan menambahkan value setiap datanya dilooping.

Jika kita tidak menggunakannya di perulangan maka nilainya akan sama pertama kali dan + 1 untuk berikutnya.

Sedangkan ++a akan langsung menambahkan + 1 setelah dipanggil.

## Operasi Perbandingan

Operasi perbandingan adalah operasi untuk membandingkan 2 buah data, dan menghasilkan nilai Boolean (true / false).

A screenshot of a computer

Description automatically generated

A screen shot of a computer code

Description automatically generated

## Operasi Boolean

A screenshot of a computer

Description automatically generated

**&&** jika true keduanya maka akan true, sedangkan jika ada false maka akan false.

**||** jika keduanya false maka akan false, sedangkan jika salah satunya true atau keduanya true maka akan true.

**!** jika true maka akan false, dan false akan jadi true.

A screen shot of a computer program

Description automatically generated

## Expression, Statement & Block

Ekspression adalah konstruksi dari variabel, operator dan pemanggilan method yang mengevaluasi menjadi sebuah single value.

*Expression adalah core component dari statement.*

A black background with white text and green lines

Description automatically generatedYang hijau merupakan sebuah expression dan juga statement.

Dan yang warna kuning merupakan sebuah statement.

Statement berisikan execution component, biasanya diakhiri dengan ;

Ada beberapa jenis statement.

* Assignment expression
* Penggunaan ++ dan –
* Method Invocation
* Object Create Expression

A computer screen shot of a code

Description automatically generated

Block adalah kumpulan statement yang terdiri dari 0 atau lebih statement.

Block diawali dan diakhiri dengan kurung kurawal **{ }**

A screen shot of a computer code

Description automatically generated

## If Statement

Dalam Java **If** statement itu merupakan kata kunci untuk percabangan. Percabangan artinya kita bisa mengeksekusi kode program tertentu ketika suatu kondisi terpenuhi.

**Else if** Statement

Kadang di dalam If kita membutuhkan beberapa kondisi, kita bisa gunakan else if untuk melakukan kondisi pengecekan sama seperti if.

**Else** Statement

Blok if akan dieksekusi ketika kondisi if bernilai true, kadang kita ingin melakukan eksekusi tertentu jika if bernilai false.

A screen shot of a computer program

Description automatically generated

## Switch Statement

Kadang kita hanya butuh menggunakan kondisi sederhana di if statement, seperti hanya menggunakan perbandingan ==.

Switch adalah statement percabangan yang sama dengan if, namun lebih sederhana pembuatannya.

Kondisi Switch statement hanya untuk perbandingan ==.

A screen shot of a computer program

Description automatically generated

*Keuntungan menggunakan switch itu alur nya lebih mudah untuk dipahami dibandingkan terjebak dengan if else hell.*

*Namun switch case, tidak bisa seperti if adanya operator perbandingan di dalamnya. Sehingga jika kita ingin melakukkan perbandingan cukup lakukan dengan if statement.*

### Switch Lambda

Saat java v.14 diperkenalkan switch expression dengan lambda.

Ini lebih mempermudah saat penyusunan switch expression kita tidak perlu lagi menggunakan kata kunci break.

A screen shot of a computer code

Description automatically generated

### Kata kunci Yeild

Di Java v.14 ada kata kunci baru yaitu yield, dimana kita menggunakan kata kunci yeild untuk mengembalikan nilai pada switch statement.

Ini akan sangat mempermudah kita ketika butuh membuat data berdasarkan kondisi switch statement.

*Switch tanpa yeild*

A computer screen with text

Description automatically generatedkita bisa inisialisasikan sebuah message kosong, kemudian nilainya akan diisi setelah switch berhasil.

Nanti kita set ulang nilai dari message supaya tidak null.

*Switch dengan yield*

*A screen shot of a computer program

Description automatically generated*Semuanya sama saja hanya dalam penulisan codenya saja yang berbeda, dan hasil yang ditampilkan juga tetap sama.

Yield itu seperti melakukan return value.

## Ternary Operator

Ternary Operator adalah operator sederhana dari if statement.

Ternary Operator terdiri dari kondisi yang dievaluasi, jika menghasilkan true maka mengembalikan nilai pertama, jika false maka akan mengembalikan nilai kedua.

Cara menggunakan ternary operator ini lebih cepat, jika if else hanya memiliki 1 kondisi.

A blue and green text

Description automatically generated

## For Loop

For adalah salah satu kata kunci yang bisa digunakan untuk melakukan perulangan. Blok kode yang ada di dalam for akan diulang hingga kondisi for terpenuhi.

A computer code with colorful text

Description automatically generated

Perulangan itu akan dilakukan hingga i bernilai sama dengan 10 dan itu akan berhenti mengulangi block yang ada di dalam for.

*Jika kita melakukan perulangan tanpa kondisi dan initstatement atau (;;) maka nanti akan menghadapi for tanpa henti.*

## While Loop

While loop adalah versi perulangan yang lebih sederhana dibandingkan forLoop.

Di dalam while loop hanya terdapat kondisi perulangan tanpa adanya *init statement* dan *post statement*.

A screen shot of a computer

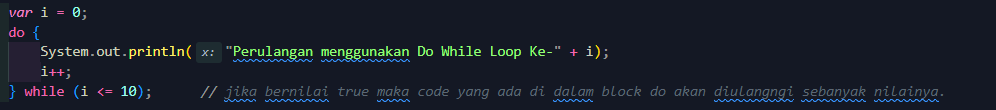
Description automatically generated

## Do While Loop

Do While loop adalah perulangan yang mirip dengan while.

Perbedaannya hanya saat melakukan pengecekan kondisi, Pengecekan kondisi pada while loop dilakukan diawal, sedangkan do while dilakukan setelah perulangan dilakukan.

Oleh karena itu meskipun nilainya tidak sesuai akan tetap diulangi 1x lalu melakukan looping selanjutnya.



## Break & Continue

Pada switch statement kita sudah mengenal kata kunci break, yang digunakan untuk menghentikan case. Sama juga pada perulangan break juga digunakan untuk menghentikan seluruh perulangan.

Namun berbeda dengan continue, continue akan digunakan untuk menghentikan perulangan saat ini dan akan melanjutkan perulangan selanjutnya.

Contoh break :

A screenshot of a computer screen

Description automatically generated

Contoh continue :

A screen shot of a computer program

Description automatically generated

Dapat dilihat kita membuat sebuah kondisi ketika nilai i modulus 2 maka continue, maksudnya berhentikan, lalu lanjutkan di loop berikutnya.

## ForEach

Kadang kita bisa mengakses data array menggunakan perulangan.

Mengakses data array menggunakan for loop itu cukup bertele tele, kita harus membuat counter lalu mengakses array menggunakan counter yang kita buat.

Namun terapat forEach yang bisa digunakan untuk mengakses seluruh data array secara otomatis.

Tanpa menggunakan forEach

A screen shot of a computer code

Description automatically generated

Menggunakan forEach

A screenshot of a computer code

Description automatically generated

*forEach di java cukup berbeda dengan beberapa bahasa pemrograman lainnya karena biasanya (datas as data) sedangkan di java (data : datas). Keduanya mirip hanya saja terbalik di java dan juga java membutuhkan tipe data sebelum pemanggilan data di dalam foreachnya.*

## Method & Method parameter

Metho adalah block kode yang berjalan ketika dipanggil. *System.out.println()* ini juga merupakan termasuk method yang betujuan untuk menampilkan data ke console.

Untuk membuat method di java kita juga bisa menggunakan *void* lalu diikuti *nama\_method­() dan diakhiri dengan block. Method* dapat kita sebut juga sebagai *function*.

Karena biasanya function berada di luar class, karena java semuanya berada di dalam class oleh karena itu *functionnya* disebut dengan *method*.

A screenshot of a computer program

Description automatically generated

*Karena main itu merupakan static maka hanya boleh memanggil method static juga.*

Di dalam main kita memanggil sebuah method yang berada di class yang sama dan itu static, dengan mengirimkan argument dan ditangkap oleh parameter.

*Argument* adalah data yang dikirimkan saat pemanggilan method, sednagkan *parameter* adalah deklarasi yang ditangkap oleh Method.

## Method Return Value

Secara default, method itu tidak menghasilkan value apapun, namun kita juga bisa melakukan dimana method mengembalikan return value.

Agar method bisa mengembalikan return value kita harus mengubah kata kunci *void* dengan *tipe\_data* yang akan dihasilkannya.

A screen shot of a computer program

Description automatically generated

## Method Variabel Argument

Kadang kita butuh mengirimkan data yang tidak pasti ke dalam sebuah method.

Biasanya, agar seperti ini, kita akan gunakan array sebagai parameter di method tersebut.

Namun di java, kita bisa gunakan variabel *argument*, untuk mengirimkan data yang berisi jumlah tak tentu, bisa nol atau lebih.

Parameter dengan tipe variabel *argument*, hanya bisa ditempatkan di posisi akhir parameter.

A screen shot of a computer

Description automatically generated

## Method Overloading

Method overloading adalah kemampuan untuk membuat method dengan nama yang sama lebih dari 1x.

Ketentuannya yaitu data parameter di method tersebut harus berbeda beda, entah jumlah atau tipe parameternya. Jika sama maka akan terjadi error.

A computer screen shot of text

Description automatically generated

## Recursive Method (Factorial)

Recursive Method adalah kemampuan method untuk memanggil method dirinya sendiri.

Kadang memang ada banyak problem, yang lebih mudah diselesaikan menggunakan recursive method, contoh kasusnya adalah factorial.

Factorial itu misal saya membuat angka 5 maka akan dikalikan 4, 3, 2, 1.

Berikut Contoh Factorial tanpa menggunakan recursive method

A screenshot of a computer program

Description automatically generated

Penjelasan nya sederhana, ketika valuenya adalah 1 maka kembalikan nilai 1, tapi jika tidak maka jalankan kembali functionnya dan valuenya dikali dan value di dalam method dikurang 1.

A screen shot of a computer code

Description automatically generated

**Problem dengan recursive**

Walaupun menggunakan recursive itu menarik, namun kita perlu berhati hati.

Jika recursive terlalu dalam, maka akan terjadi error StackOverflow, yaitu errornya dimana stack method terlalu banyak dijava.

Kenapa problem ini terjadi?

Karena ketika kita memanggil method, Java akan menyimpannya di dalam stack, jika method itu memanggil method lain, maka stack akan menumpuk terus menerus, dan jika terlalu dalam maka stack nya akan terlalu besar dan menyebabkan error StackOverflow.

Contoh kode untuk mengecek kendala rekursive

A screen shot of a computer code

Description automatically generated

## Scope

Di Java, variabel hanya bisa diakses di dalam area dimana mereka dibuat (Scope).

Contoh jika kita membuat variabel di dalam sebuah method maka variabel tersebut hanya bisa diakses di dalam block tersebut.

*A screen shot of a computer program

Description automatically generated*Contoh codenya hi error jika dipanggil diluar dari scope if karena itu variabel hanya bisa diakses di dalam if saja.

## Comment

Kadang kita membutuhkan sebuah komentar seperti memberikan sebuah dokumentasi, komentar ini akan dihiraukan saat melakukan kompilasi / dihapus.

Komentar multi line & single line.

Multi line lebih digunakan untuk dokumentasi, sedangkan single line hanya untuk memberikan informasi pada baris Code.

A blue background with white text

Description automatically generated