|  |
| --- |
| **Java** |
| **Catatan Belajar Java** |

# Java Dasar

## Sejarah Java

Java merupakan Bahasa pemrograman yang banyak digunakan oleh developer dalam mengembangkan aplikasi desktop, mobile dan juga website. Java dikenal dengan motto *“Write Once, Run Anywhere”* artinya java merupakan Bahasa pemrograman yang mampu diciptakan di berbagai platform tanpa harus penyeusaian ulang di platformnya.

Alasan Java dapat dijalankan di berbagai platform karena Bahasa ini merupakan Bahasa Tingkat tinggi. Di java terdapat **JVM** (*Java Virtual Machine*) Dimana akan merubah syntax menjadi kode numeric (*bytecode*) sehingga dapat dijalankan diberbagai perangkat.

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

Komponen – Komponen JAVA :

* JVM (*Java Virtual Machine*)

= Jantung dari java yang merubah code menjadi kode byte.

* JDK (*Java Development Kit*)

= Komponen utama dari java di dalamnya mencakup JRE dan juga JVM.

* JRE (*Java Runtime Environment*)

= Menjalankan aplikasi yang sudah dikompilasi kedalam bytecode & JRE tidak termasuk compiler karena hanya menjalankan aplikasi java yang sudah jadi.

### Menjalankan Kode Java

File Java itu berekstensi file .java Ketika kita menjalankan java terdapat 2 cara, seperti berikut :

* Compile ke dalam ekstensi .class

Anggap kita mempunyai sebuah file java “HelloWorld.java”

***javac HelloWorld.java***

= akan menghasilkan hasil kompilasi yaitu file .class

***java Hello World***

= akan menjalankan file .class

* Langsung dijalankan

***java HelloWorld.java***

= langsung dikompilasi dan dijalankan

Berikut adalah contoh Gambaran Proses Development Java.

A diagram of a computer program

Description automatically generated

## Instalasi Java (JDK)

Pada url tersebut <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 JDK java pada url berikut <https://jdk.java.net/>

Installasi :

* Setelah download install dan simpan di dalam folder C:
* Kemudian copy path urlnya seperti berikut “*C:\openjdk-22.0.1\jdk-22.0.1*”
* Buka setting sistem environtment & pergi untuk menambahkan *System Variables* baru
* Klik New dengan nama **JAVA\_HOME** dan value nya diisi path tersebut.
* Kemudian Pilih di *System Variables* Path klik edit, klik new dan tambahkan “*%JAVA\_HOME%\bin*”
* Buka cmd lakukan java –version untuk mengecek JRE dan javac –version untuk mengecek compiler atau JVM.

### 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 yang biasanya diguanakn untuk membuat code program JAVA :

* Intelli JDEA Ultimate / Community
* Eclipse
* NetBeans
* JDeveloper
* Visual Studio Code

## Program Awal - Hello World

A screen shot of a computer program

Description automatically generated

Sebelumnya buat folder dengan nama **java\_dasar** didalamnya akan berisi kode kode program java dasar. Karena disimpan di dalam sebuah folder **java\_dasar** kita harus memanggil foldernya menggunakan *package*.

Di dalam java kita harus membat sebuah Class utama yang didalam nya berisi sebuah ***main*** function, agar Ketika dijalankan maka main function tersebut yang akan dijalankan. Pembuatan main function :

***public static void main (String[] args) { … }***

System.out.println() = digunakan untuk mencetak data dan memberikan new line.

System.out.print() = sama dengan println namun tidak memberikan new line.

## Tipe Data Number

### Integer Number

Integer Number, atau kita anggap adalah bilangan bulat di dalam java terdapat banyak kita bisa menggunakan *byte, short, int, dan long*. Semuanya punya min max pada nilainya yang berbeda beda, semakin besar maka semakin banyak juga memakan memory.

A computer screen shot of numbers and lines

Description automatically generatedA table with numbers and a number

Description automatically generated

Kita dapat memberikan sebuah Keyword (***L***) pada long sebagai menandakan bahwa variable tersebut menggunakan tipe data *long.*

### Floating Point Number

Sedangkan floating point merupakan sebuah nilai (number) yang memiliki nilai di belakang koma, seperti 3,14 maka kita gunakan floating number baik itu *float* atau *double*.

**A close-up of a number

Description automatically generated**

**A number and lines on a black background

Description automatically generated**

Float juga mirip seperti Long dapat diberikan penanda dengan (***F***) bahwa nilai floating point tersebut adalah float.

### Literals Number

Literals Number merupakan sebuah nilai (number) yang biasanya digunakan saat kondisi tertentu seperti membuat nilai hexadecimal yang diawali dengan (***0x***), sedangkan untuk bilangan binary dapat diawali dengan (***0b***).

A computer screen shot of a number

Description automatically generated

### Underscore Number

Terkadang jika nilai kita berupa angka Panjang kita memisahkannya menggunakan tanda . (titik) namun di java tidak seperti itu di java kita menggunakan tanda \_ (underscore) untuk nilai yang panjang karena . (titik) digunakan untuk floating point number.

**A number and numbers on a black background

Description automatically generated**

### Konversi Tipe Data Number

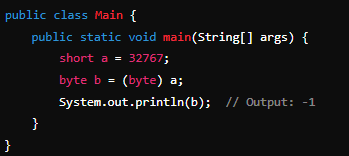
Di dalam Konversi Tipe Data Number terdapat dua cara dalam melakuan konversi, ada *Widening Casting* (Konversi secara otomatis) dan ada juga *Narrowing Casting* (Konversi secara manual). Lantas apa perbedaan diantara keduanya?

***Widening Casting*** *kita dapat merubah tipe data yang lebih kecil ke tipe data yang lebih besar.*

***Narrowing Casting*** merupakan konversi tipe data yang lebih besar ke tipe data yang lebih kecil, ini harus dilakukan secara eksplisit karena ada kemungkinan kehilangan data atau lebih dikenal dengan *Downcasting*.

*\*perlu diketahui Narrowing Casting nilai dari sebuah tipe data akan dibuat ke dalam byte code hal ini akan menyebabkan kehilangan data / perubahan nilai aslinya.*

Contoh Narrowing Casting :



Nilai awal *a* adalah 32767. *A* merupakan tipe data short yang sizenya merupakan 2 byte. ***1 byte adalah 8 bit.***

Ketika *a* diubah kedalam bit code maka menghasilkan 0111 1111 1111 1111 (16 bit).

Sedangkan tipe data byte hanya bernilai 1 byte atau 8 bit maka, Ketika dikonversi dari short ke bit akan terjadi perubahan data karena nilai yang diambil hanya pada 1111 1111 (8 bit).

Lalu bagaimana dengan hasilnya? Sebelum kesana kita harus mengetahui matematika ***kompenen***. Dimana kompenen ini sangat dipakai dalam menghitung byte.

#### Menghitung Nilai Kompenen

Nilai komplemen pada byte ada 2, ada one’s complement dan two’s complement. Komplemen ini akan ***mereverse*** dari nilai awalnya. Kita beri contoh perhitungannya seperti berikut :

Tadi nilai yang didapat adalah 1111 1111

*One’s Complements (reverse to binary)*

1111 1111 -> 0000 0000 = 0

*Two’s Complements (reverse to binary)*

1111 1111 -> 0000 0000 + 1

0000 0001 = -1

*Jika diawali dengn nilai 0 maka bilangan tersebut merupakan bilangan negative sedangkan jika diawali dengan 1 maka bilangan tersebut merupakan bilangan positif.*

Kesimpulannya, Java itu menerapkan two’s complement dalam melakukan narrowing casting, ini dilakukan karena untuk menghitung nilai wal yang memiliki nilai yang besar Ketika dikonveri ke nilai yang kecil.

*\*jika konversi nilainya tidak sesuai dari max / min tipe data konversinya maka akan terjadi ‘number overflow’ artinya nilainya dapat direpresentasikan oleh tipe dengan bilangan tertentu.*

A screenshot of a computer program

Description automatically generated

## Tipe Data Character

Di java tipe data ***char*** merupakan tipe data untuk 1 buah karakter (huruf) dengan ditandai menggunakan ‘ ’ (petik satu).

A computer screen shot of a computer code

Description automatically generated

## Tipe Data Boolean

Boolean di java direpresentasikan dengan nama tipe data ***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

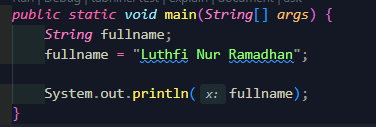
Description automatically generated

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

## Variabel

Variabel adalah tempat untuk menyimpan data. Java adalah Bahasa yang static type (*pendeklarasian tipe data saat kompilasi*), 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 bisa tidak perlu menyebutkan secara eksplisit 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 harus 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 berupa sebuah ***Object***.

A screenshot of a computer

Description automatically generated

A graph with red line

Description automatically generated with medium confidence

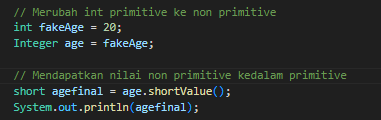
### Konversi ke Bukan Primitif

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

Jika ingin melakukan konversi dari Object / Bukan Primitive ke dalam Primitive dapat dengan menggunakan method namaTipeDataValue(), seperti berikut.



## 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. Diawali dengan nama tipe data nya, kemudian inisialisasi datanya dan berikan jumlah data yang bisa ditampung.

A computer screen shot of a program code

Description automatically generated

### Array Initializer

Cara membuat array tidak hanya seperti itu kita bisa membuat array dengan beberapa cara :

A computer screen shot of a code

Description automatically generated

*Cara biasa ini kita harus menambahkan keyword new dan nama tipe data beserta jumlah array yang bisa ditampungnya.*

A computer screen shot of a code

Description automatically generated

*Cara kedua ini hamper sama dengan yang pertama namun kita tidak menyisipkan jumlah arraynya oleh karena itu array ini tidak memiliki batasan namun kita harus membuat isi datanya secara langsung karena jika tidak maka akan error.*

A screenshot of a computer code

Description automatically generated

*Cara ketiga ini adalah cara yang paling simple karena kita cukup gunakan { } untuk mengisikan arraynya, namun array wajib diisi saat pertama kali dibuat.*

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 **null**.

### Array in Array

Ketika kita ingin membuat array di dalam array, caranya masih mirip dengan cara sebelumnya hanya saja diberikan 2 array di dalamnya contohnya seperti berikut.

A screen shot of a computer code

Description automatically generated

## Operasi Matematika

Operasi matematika adalah operasi digunakan untuk perhitungan di Java Berikut adalah symbol operasi Matematika di java

A screenshot of a computer

Description automatically generated

### Augmented Assignment

Augemented assignment merupkana cara / gaya penulisan agar lebih singkat lagi di dalam operasi matematika di Java.

A table with text and numbers

Description automatically generated

A screenshot of a computer program

Description automatically generated

### Unary Operator

Unary operator juga merupakan cara / style penulisan untuk mempersingkat operasi matematika di dalam blok kode.

A screenshot of a computer

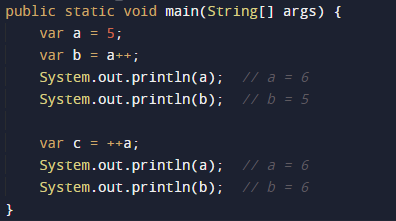
Description automatically generated

A screen shot of a computer program

Description automatically generated

Perlu diingat antara perbedaan a++ dengan ++a.

Jika *a++* biasanya dilakukan saat perulangan karena akan menambahkan value setiap datanya dilooping. Namun jika *++a* maka nilainya akan ditambahkan terlebihi dahulu +1 saat pertama kali. Sebagai contoh berikut.



## 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

Sebelumnya operasi perbandingan itu menghasilkan sebuah hasil true or false, jika operasi boolean itu melibatkan logika boolean seperti berikut.

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 tanda ; (titik koma).

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 lah 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 juga 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.

*Contoh Switch dengan yield :*

*A screen shot of a computer program

Description automatically generated*

Yield itu seperti melakukan return value. Dimana sebelumnya kita simpan logika switch kedalam sebuah tipe data, dan mengembalikan value dari tipe data tersebut yang sudah diolah menggunakan switch.

## 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*. Namun while loop itu mirip seperti if else lalu loop, karena while melakukan pengecekan terlebih dahulu jika true maka loop akan dijalankan.

Dan untuk memberhentikan loop seperti biasa kita gunakan keyword ***break;***

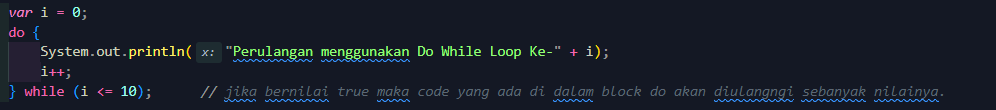
A screen shot of a computer

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.



## Keyword 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 terdapat 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

Method 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 itu identik berada di luar class, jika di java semuanya berada di dalam class oleh karena itu *function-nya* 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, sedangkan *parameter* adalah deklarasi / nilai 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

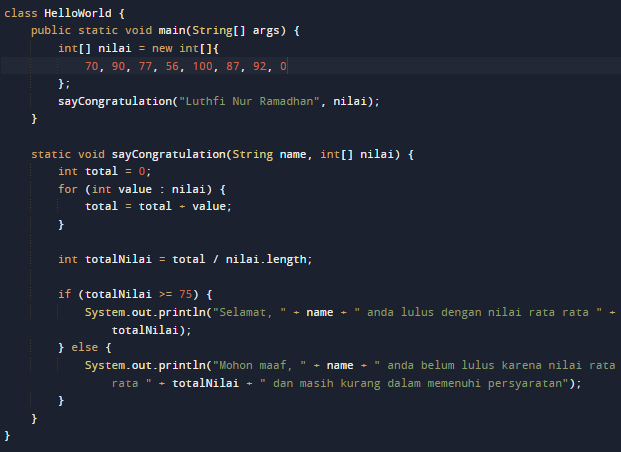
Description automatically generated

## Method Variabel Argument

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



## Method Overloading (Same name method)

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. Komentar 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

# Studi Kasus Java Dasar Aplikasi To Do List

## Prototype Aplikasi ToDoList

Prototype adalah bentuk dasar dari sebuah aplikasi. Prototype bisa digunakan agar memahami flow aplikasi yang akan kita buat, sehingga tidak salah ketika membuat aplikasi. Berikut adalah gambaran prototype aplikasi TodoList yang akan dibuat :

A screenshot of a cell phone

Description automatically generated

## Create Project Java

Untuk membuat project java, di vs code kita bisa dengan cara *CTRL + SHIFT + P*, kemudian cari java: create project. Pastikan sebelumnya sudah menginstall ekstension yang berhubungan dengan Java.

## Membuat Main Class

Main class merupakan class yang akan pertama kali / akan dijalankan di dalam program Java, Biasanya kita membuat 1 main class.

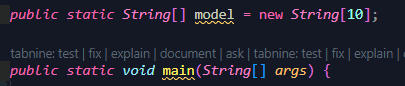
A computer screen shot of text

Description automatically generated

## Membuat Model

Model merupakan Representasi dari data. Nanti setiap data yang masuk akan dimasukkan ke dalam array untuk disimpan. Apabila kita ingin membuat model dengan nilai yang tidak tentu kita bisa ubah *new String[10]* menjadi *{ }*. Namun jika kita menggunakan *{ }* kita tidak bisa menyisipkan data array ke dalamnya.

Model dibuat secara static dan public diluar method main, kenapa diluar? Karena model ini akan diakses oleh method method yang lain.



## Menentukan Bussiness Logic

Sebelumnya kita tentukan di dalam aplikasi akan ada logika seperti apa di dalamnya, misal menampilkan data, create data, dan hapus data. Berikut adalah method method lainnya yang akan dibuat selain method main.



## Menentukan View

Sebelumnya merupakan logic / method untuk menjalankan logika, sedangkan method selanjutnya ini adalah untuk menampilkan view dari aplikasi meskipun nanti view nya hanya berupa command line.

A screenshot of a computer program

Description automatically generated

## Menampilkan Todo List

Ketika kita ingin menampilkan todo list yang berupa array String, pastinya kita melakukan perulangan pada datanya, baik menggunakan forLoop biasa ataupun forEach. Berikut adalah contoh dalam menampilkan todoList

A screenshot of a computer program

Description automatically generated

Pada contoh menampilkan todo list, kita buat sebuah method tambahan untuk melakukan testing, kenapa harus melakukan testing? Karena agar pengujian yang dilakukan tidak pada code utama nya.

## Menambah Todo list

Berikutnya kita akan coba menambahkan todo list baru ke dalam model. Untuk menambahkan data ke dalam model pastikan kita buat sebuah pengecekan terlebih dahulu, jadi jika datanya belum penuh maka data tersebut akan disimpan ke dalam model, sedangkan jika sudah penuh ukuran model / total array-nya akan dikali 2 dengan ukuran array sekarang.

Lalu untuk menambahkan datanya kita lakukan perulangan Kembali, jika datanya masih null maka datanya null tersebut akan diisi oleh data yang sekarang diterima.

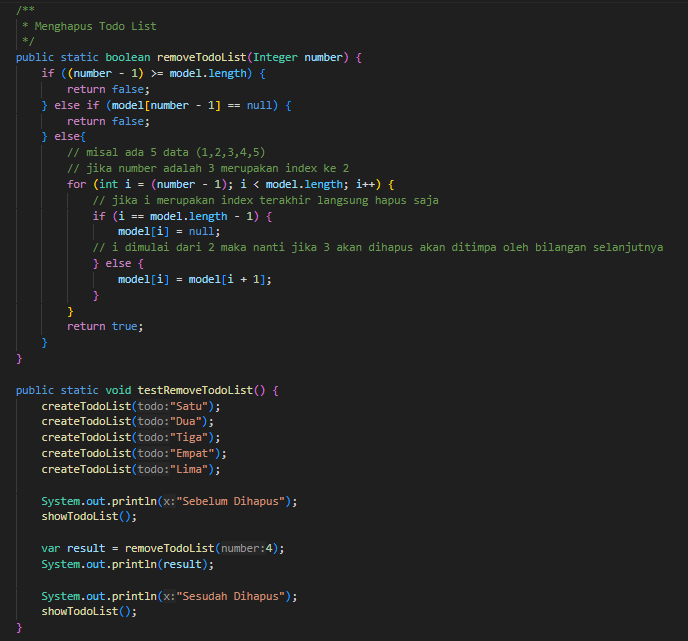
A screen shot of a computer program

Description automatically generated

Sama seperti menampilkan todo List disini kita juga harus membuat method untuk melakukan testing agar menghindari perubahan code pada method utamanya.

## Menghapus Todo List

Selanjutnya adalah menghapus Todo List berikut adalah contoh kode dalam menghapus todolist

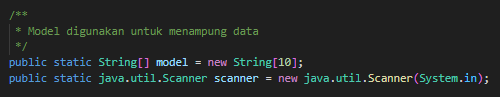


Di dalam method remove kita cek terlebih dahulu jika inputannya tidak sesuai / jika inputannya kosong, dan jika ada datanya dan mau dihapus maka buat data tersebut jadi null kemudian naikkan posisinya ke posisi data yang sudah dihapus, apabila menghapus data ditengah agar data selanjutnya naik. Pastikan membuat method testing agar melakukan percobaan di dalam method tersebut.

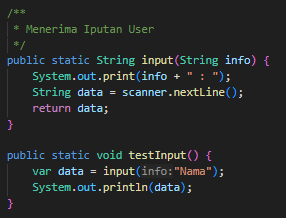
## Input Data

Terkadang kita tidak mungkin membuat inputan melalui codingan langsung, kadang kita sendiri yang ingin mengisikan nilai datanya. Oleh karena itu ikuti cara berikut.

Tambahkan code berikut untuk bisa menerima inputan



Kemudian buat method untuk menerima dan mengembalikan inputan

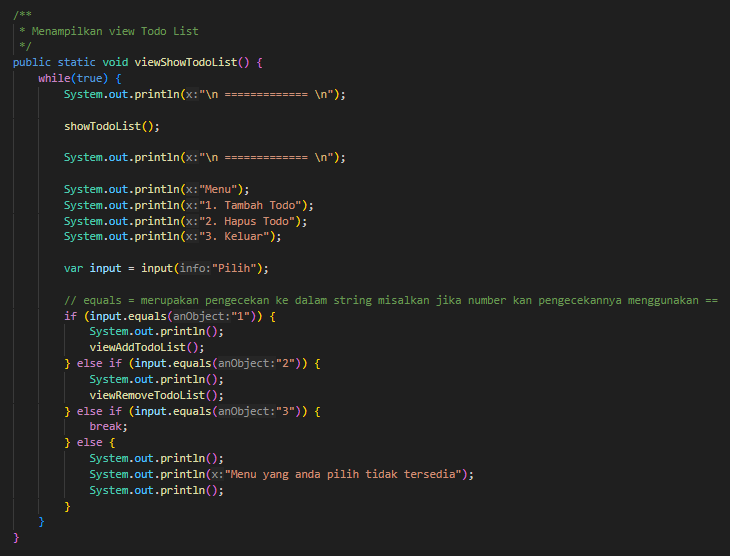


*scanner.nextLine() digunakan untuk mendapatkan inputan dari scanner.*

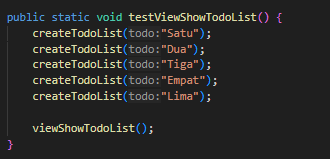
Pastikan juga kita tetap membuat method testing untuk menguji coba method input tersebut.

## View Menampilkan Todo List

Menampilkan todo list ini hanya berupa command line yang akan ditampilkan, berikut adalah method untuk menampilkan todo list.



Terdapat sebuah kondisi dengan menggunakan function equals(). Fungsi tersebut merupakan sebuah pengecekan pada inputan apabila terdapat / menginputkan berupa nilai angka 1, 2, atau 3 maka akan menjalankan method-nya.



Pastikan juga membuat sebuah method testing agar membuat code lebih clean dan menghindari bug.

## View Menambahkan Todo List

Dalam menambahkan todo list juga kita lakuan pengecekan pada inputannya, dan Ketika inputan sesuai maka data akan ditambahkan ke dalam model.



Pastikan kita membuat sebuah method testing juga, agar Ketika ingin melakukan uji coba tidak pada code utama.

## View Menghapus Todo List

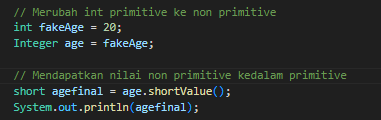
Untuk menghapus todo list kita membutuhkan id / nomor dari todo list nya, karena dengan nomor tersebut maka datanya akan dihapus. Berikut adalah contoh codenya.



Sebelumnya kita bisa melakukan pengecekan pada model dan memberikan message apabila modelnya masih kosong, dan tampilkan jika modelnya emang ada data di dalamnya.

Dikarenakan Ketika ingin melakukan remove adalah Tipe data non Primitive maka kita harus mendapatkan valuenya.

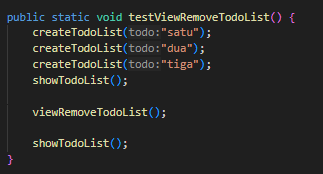
Sebelumnya kita pernah melakukan code berikut



Perlu diingat bahwa menggunakan cara tersebut untuk mendapatkan value dari tipe data non primitive ke primitive oleh karena itu karena *Integer* merupakan sebuah object maka akan memiliki fungsi seperti *shortValue().*

Sedangkan ***valueOf*** itu digunakan untuk mendapatkan value dari tipe data primitive ke non primitive. Oleh karena itu cara diatas menggunakan *valueOf* dikarenakan number disimpan ke dalam tipe data var, dan inputan mengembalikan String.

Pastikan juga untuk membuat method testingnya…



# Object Oriented Programming (OOP) Java

## Test

Test