

# Daftar ISI

[Daftar ISI 2](#_heading=h.gjdgxs)

[1.](#_heading=h.30j0zll) Pengenalan Java 3

[2.](#_heading=h.1fob9te) Instalasi JDK 5

[3.](#_heading=h.3znysh7) Instalasi Intellij IDEA 8

[4.](#_heading=h.2et92p0) Instalasi PostgreSQL 9

[5.](#_heading=h.tyjcwt) Instalasi Postman 11

[6.](#_heading=h.3dy6vkm) Instalasi Git 12

[7.](#_heading=h.1t3h5sf) Variabel 14

[8.](#_heading=h.4d34og8) Tipe Data 16

[a.](#_heading=h.2s8eyo1) Tipe Data Primitif 16

[b.](#_heading=h.17dp8vu) Tipe Data Non Primitif (Reference Tipe Data) 19

[9.](#_heading=h.3rdcrjn) Function 24

[a.](#_heading=h.26in1rg) Cara Membuat Fungsi di Java 24

[b.](#_heading=h.lnxbz9) Cara Memanggil/Eksekusi Fungsi 24

[c.](#_heading=h.35nkun2) Fungsi dengan Parameter 25

[d.](#_heading=h.1ksv4uv) Fungsi yang Mengembalikan Nilai 26

[e.](#_heading=h.44sinio) Pemanggilan Fungsi di Fungsi Lain 26

[f.](#_heading=h.2jxsxqh) Fungsi Static dan Non-Static 27

[10.](#_heading=h.z337ya) Input & Output 29

[a.](#_heading=h.3j2qqm3) Mengambil Input dengan Class Scanner 29

[b.](#_heading=h.1y810tw) Mengambil Input dengan Class BufferReader 31

[c.](#_heading=h.4i7ojhp) Menampilkan Output 32

[Daftar Pustaka 33](#_heading=h.2xcytpi)

# Pengenalan Java

Java adalah bahasa pemrograman yang biasa digunakan untuk mengembangkan bagian back-end dari software, aplikasi Android, dan juga website.

Java juga dikenal memiliki moto “Write Once, Run Anywhere”. Artinya, Java mampu dijalankan di berbagai platform tanpa perlu disusun ulang menyesuaikan platformnya. Misalnya, berjalan di Android, Linux, Windows, dan lainnya.

Hal itu dapat terjadi karena Java memiliki sistem syntax atau kode pemrograman level tinggi. Di mana ketika dijalankan, syntax akan di-compile dengan Java Virtual Machine (JVM) menjadi kode numeric (bytescode) platform. Sehingga aplikasi Java bisa dijalankan di berbagai perangkat.

Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code(bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM[)](https://id.wikipedia.org/wiki/Mesin_virtual_java). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, “Tulis sekali, jalankan di mana pun”. Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi.

Pada mulanya, untuk belajar ataupun mengenal bahasa pemrograman Java ini sendiri memerlukan beberapa software pendukung yaitu adalah JDK dan juga JRE. JDK ( Java Development Kit ) adalah perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode java ke bytecode yang dapat dimengerti dan dapat dijalankan oleh JRE (Java Runtime Envirotment). Sedangkan JRE adalah perangkat lunak yang digunakan untuk menjalankan aplikasi yang dibangun menggunakan java. Versi JRE harus sama atau lebih tinggi dengan JDK yang dugunakan untuk membangun aplikasi java.

Banyak orang menyebut bahwa Java merupakan bahasa pemrograman yang powerful. Hal ini mengingat sifatnya yang multiplatform dan dikenal memiliki library yang lengkap.

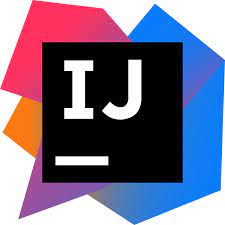
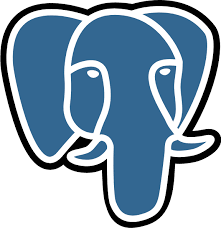
Oleh sebab itu, tak heran bila Java menjadi salah satu bahasa pemrograman yang banyak digunakan saat ini.

Java dikenal memiliki moto *“Write Once, Run Anywhere”.* Hal ini menunjukkan bahwa setiap kode program yang ditulis menggunakan Java bisa langsung dijalankan pada *platform* berbeda-beda tanpa perlu menyusun ulang.

**Apa yang akan dipelajari ?**

Target pembelajaran kedepan akan mempelajari bagaimana membangun REST API menggunakan Bahasa JAVA. Oleh sebab itu, dibutuhkan beberapa tools dan framework dalam pengembangan REST API. Secara garis besar tools dan framework tersebut diantaranya.

* Intelij IDEA : Editor Java
* PostgreSQL : Basis Data
* Postman : Testing Rest API
* Gitlab : Version Control
* Maven – Spring boot Framework

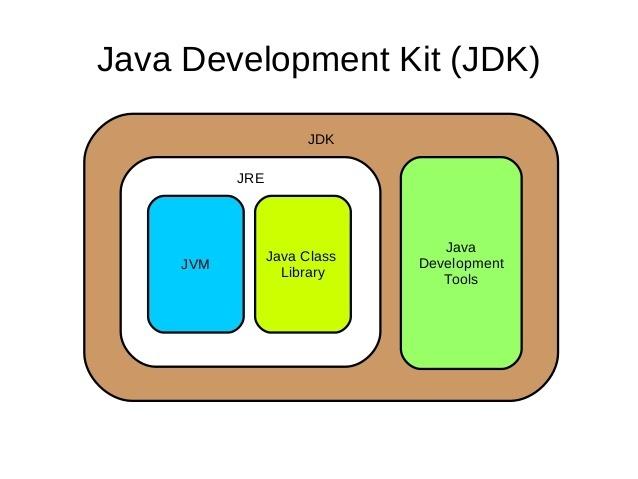


Gambar : Logo Tools

# Instalasi JDK

Java JDK adalah sebuah perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode java ke bytecode yang dapat dimengerti dan dapat dijalankan oleh JRE (Java Runtime Envirotment).

JDK wajib terinstall pada komputer yang akan melakukan proses pembuatan aplikasi berbasis java, namun tidak wajib terinstall di komputer yang akan menjalankan aplikasi yang dibangun dengan java.

JDK adalah sebuah paket aplikasi yang berisi JVM (Java Virtual Machine) + JRE (Java Runtime Environment) + berbagai aplikasi untuk proses pembuatan kode program Java.

Isi dari JDK – Java Development Kit (sumber gambar quora.com)

**Download** dan Lakukan **instalasi** pada laptop teman-teman :

<https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html>

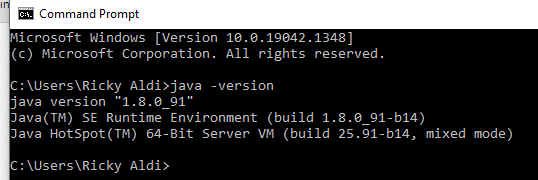
**Mengatur Path di System WIndows**

Dalam pengertian secara umum di dalam dunia komputer, path adalah istilah untuk menyebut alamat sebuah file di dalam directory komputer. Misalnya kita memiliki dokumen belajar\_java.doc. Alamat pathnya bisa saja tersimpan di C:\MyDocument\belajar\_java.doc.

Namun jika disebut “mengatur path di sistem operasi Windows“, itu maksudnya kita menginput sebuah alamat folder agar cmd Windows bisa menjalankan aplikasi yang tersimpan dari mana saja.

**Memeriksa apakah PATH Java sudah aktif atau belum**

* Buka CMD
* Ketikkan java –version

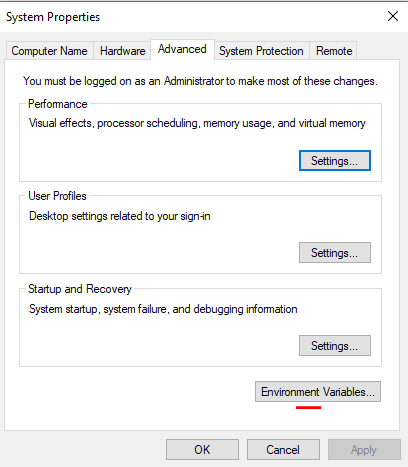


Gambar Version Java

Pada terminal diatas menunjukkan version java pada laptop teman-teman sudah terinstal. Jika terjadi eror, maka teman-teman perlu setting path java env di system windows

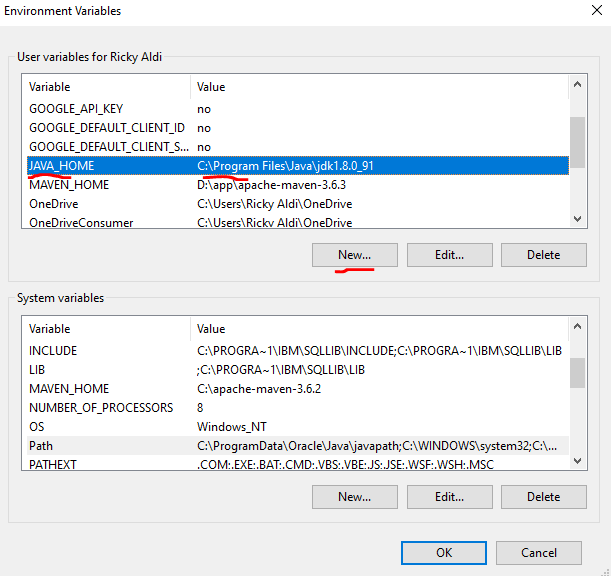
**Mengubah Pengaturan Path (environment variable) untuk Java JDK**

Ketikkan “**Advanced System Settings”** pada pencarian windows  
Pilih Environment Variabel



Gambar Advanced System Setting

Tambahkan Path JDK teman-teman pada user variable JAVA\_HOME pada gambar dibawah ini:



Gambar User Variabel

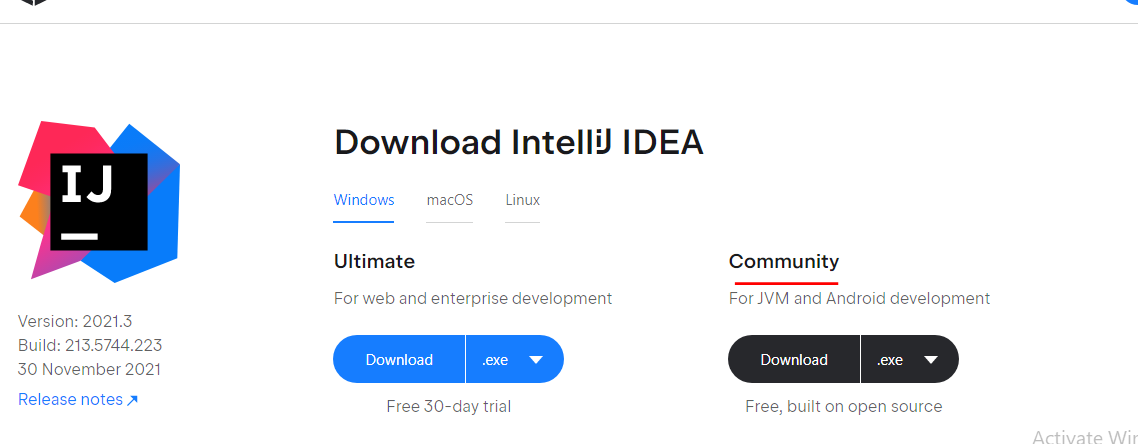
# Instalasi Intellij IDEA

***Berkenalan Dengan IntelliJ IDEA***

Intellij IDEA adalah sebuah IDE yang dikembangkan oleh Jetbrains. IntelliJ IDEA tersedia dalam 2 (dua) versi, yaitu versi Community Edition dengan lisensi *Apache 2 Licensed* yang bisa kita gunakan secara gratis dan versi Ultimate Edition untuk penggunaan komersial.

**Silahkan download pada link berikut , dan pilih versi commmunity:**

[https://www.jetbrains.com/idea/download/#section=windows](https://www.jetbrains.com/idea/download/)



Gambar Intellij IDEA

# Instalasi PostgreSQL

**Apakah itu PostgreSQL ?**

PostgreSQL adalah sistem database objek-relasional open source yang sangat kuat, dimana memiliki lebih dari 15 tahun fase pengembangan aktif dan arsitektur yang terbukti membuatnya mendapatkan reputasi yang baik untuk keandalan, integritas data, dll.

PostgreSQL mendukung empat bahasa prosedural standar, yang memungkinkan pengguna untuk menulis kode mereka sendiri dalam salah satu bahasa pemrograman dan itu dapat dieksekusi oleh server database PostgreSQL. Diantaranya adalah - Perl, Python, pgSQL. Selain itu, bahasa prosedural non-standar lainnya seperti PHP, V8, Ruby, Java, dll.

**Silahkan download dan install PostgreSQL pada link dibawah ini :**

<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

**Fitur-Fitur PostgreSQL**

* *View*
* *Transaction*
* Pilihan sub SQL
* Slony,
* Streaming replikasi
* DB mirror,
* SQL query kompleks

MVCC atau multiversion concurrency control, dan masih banyak lagi.

**Kelebihan PostgreSQL**

* **Terdapat fitur OO:**Mempunyai fitur OO berarti juga keunggulan dari PostgreSQL untuk dapat mendefinisikan tabel-tabel dan mewarisi table yang lain untuk dapat digunakan.
* **Memiliki Arstitektur Multiproses:** Arsitektur multiproses (forking) yang dimiliki oleh PostgreSQL menjadikan PostgreSQL mempunyai stabilitas yang tinggi.
* **Memiliki Kecepatan Meski Dalam Load Tinggi:** PostgreSQL didapuk memiliki kecepatan yang tinggi bahkan hingga mengalahkan kecepatan MySQL dalam hal [query](https://dosenit.com/php/fungsi-mysqli-query-php) dengan klausa [JOIN](https://dosenit.com/php/fungsi-string-join-pada-php) dengan tingkat yang kompleks
* **Memiliki Tipe Data Geometri:** PostgreSQL juga akan semakin memudahkan pengguna dikarenakan database system ini telah support berbagai tipe data geometri.
* **Menyediakan Seluruh Fitur Database:** PostgreSQL juga menyediakan bebagai macam fitur database yang pada umumnya juga terdapat pada produk-produk database komersial lain.

# Instalasi Postman

**Apa itu Postman ?**

**Postman** adalah sebuah aplikasi yang berfungsi sebagai REST Client untuk uji coba REST API. Postman biasa digunakan oleh developer pembuat API sebagai tools untuk menguji API yang telah mereka buat

**Postman** merupakan *tool* untuk melakukan proses *development* API, untuk saat ini sudah banyak fitur-fitur yang sangat membantu dalam proses *development* API, diantaranya :

* **Collection**

Pengelompokan *request* API yang bisa disimpan atau diatur dalam bentuk *folder*. Memudahkan untuk pengelompokan request sesuai dengan proyek yang di kerjakan.

* **Environment**

Semacam *config* untuk menyimpan *attribute dan attribute* tersebut dapat digunakan ataupun dimanipulasi dalam proses *request* API.

* **Response**

*Developer* dapat membuat Mockup API sebelum benar-benar mengimplementasikan ke dalam proyek.

**Silahkan** download dan install postman pada link dibawah ini :

https://www.postman.com/downloads/

# Instalasi Git

**Apa itu Git** ?

* Git adalah salah satu sistem pengontrol versi *(Version Control System)* pada proyek perangkat lunak yang diciptakan oleh Linus Torvalds.
* Pengontrol versi bertugas mencatat setiap perubahan pada file proyek yang dikerjakan oleh banyak orang maupun sendiri.
* Git dikenal juga dengan *distributed revision control* (VCS terdistribusi), artinya penyimpanan database Git tidak hanya berada dalam satu tempat saja.

**Kenapa Git Penting Bagi Programmer?**

Beberapa menfaat yang akan kamu rasakan setelah bisa menggunakan Git.

* Bisa menyimpan seluruh versi source code;
* Bisa paham cara kolaborasi dalam proyek;
* Bisa ikut berkontribusi ke poryek open-source;
* Lebih aman digunakan untuk kolaborasi, karena kita bisa tahu apa yang diubah dan siapa yang mengubahnya.
* Bisa memahami cara *deploy* aplikasi modern;
* Bisa membuat blog dengan SSG.
* dan sebagainya…

**Apa itu Gitlab ?**

* Git adalah sistem versi kode sumber yang memungkinkan Anda melacak perubahan secara lokal dan mendorong atau menarik perubahan dari remote resources.
* GitLab, GitHub, dan Bitbucket : layanan yang menyediakan remote access  ke Git repositories. Selain meng-hosting kode Anda, layanan ini menyediakan fitur tambahan yang dirancang untuk membantu mengelola siklus pengembangan software. Fitur-fitur tambahan ini termasuk mengelola pembagian kode antara orang-orang yang berbeda, bug tracking, wiki space, dan alat-alat lain untuk ‘social coding’
* GitHub adalah layanan gratis yang tersedia untuk umum yang membutuhkan semua kode (kecuali Anda memiliki akun berbayar) dibuat terbuka. Siapa pun dapat melihat kode yang Anda push ke GitHub dan menawarkan saran untuk peningkatan. GitHub saat ini meng-host kode sumber untuk puluhan ribu proyek open source.

**Silahkan download an install Instalasi gitlab pada link berikut :**

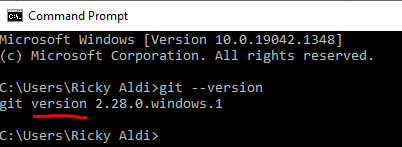
<https://docs.gitlab.com/ee/topics/git/how_to_install_git/>

**Bagaimana Cek versi git ?**

Jika ingin mengetahui apakah git sudah terinstal, maka teman-teman dapat melakukan cek versi dengan cara sebagai beikut ini

* Buka terminal
* Ketikkan “git –version”

Sehingga akan menampilkan gambar berikut.



Gambar Cek Versi Git

Berikut cara menambahkan file ke gitlab step-by-step:

## Register Gitlab

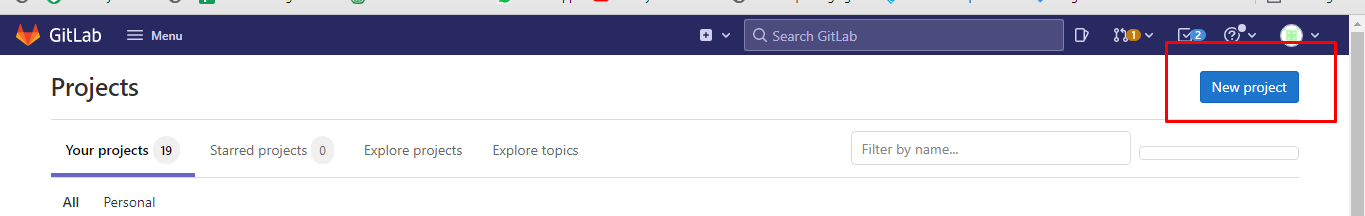
Teman-teman dapat register via link berikut : <https://gitlab.com/users/sign_up>

## Login Gitlab

Teman -teman dapat login pada link berikut : <https://gitlab.com/users/sign_in>

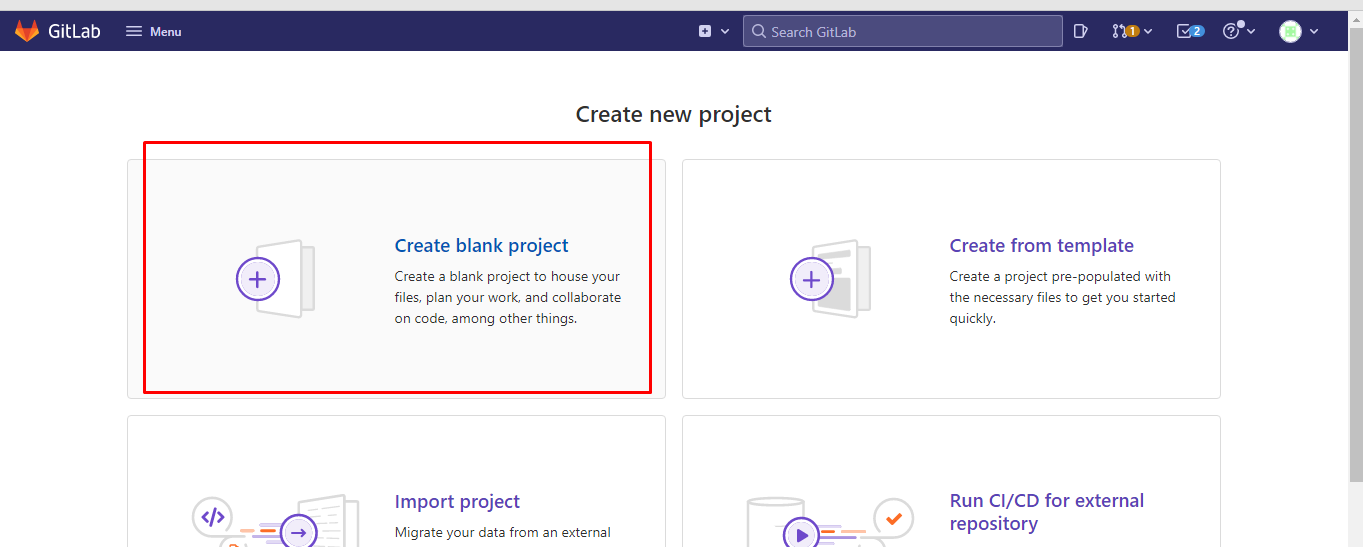
## Membuat project pada gitlab

* **Pilih new project**



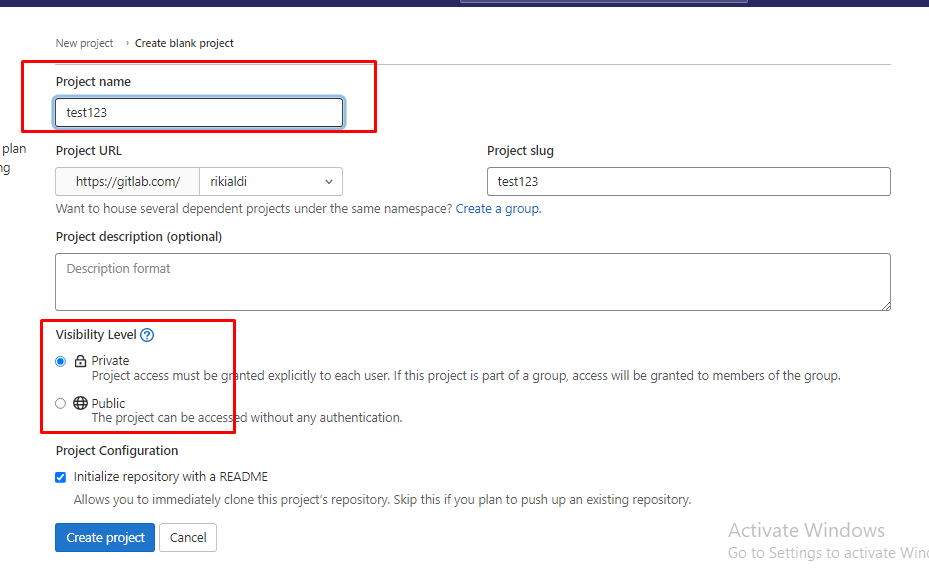
Gambar new project

* **Create blank project**



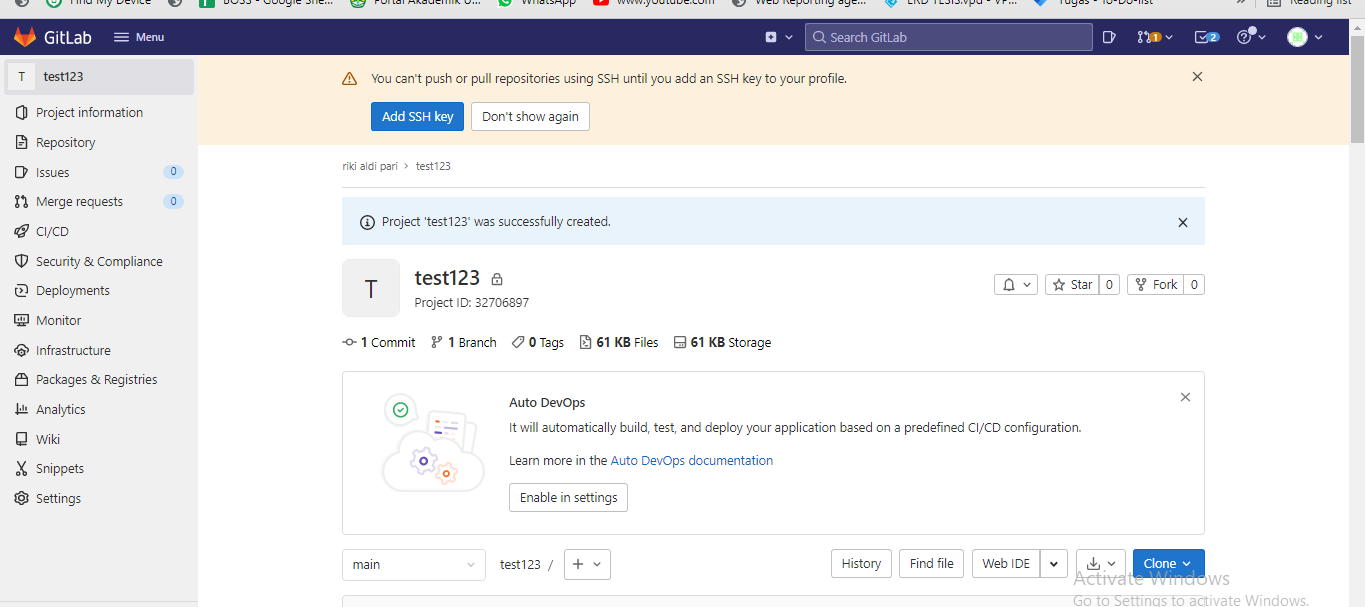
Gambar create blank project

* **Isi nama project dan visibility level**



Gambar nama project dan visibility level

* **Project akan terbentuk**

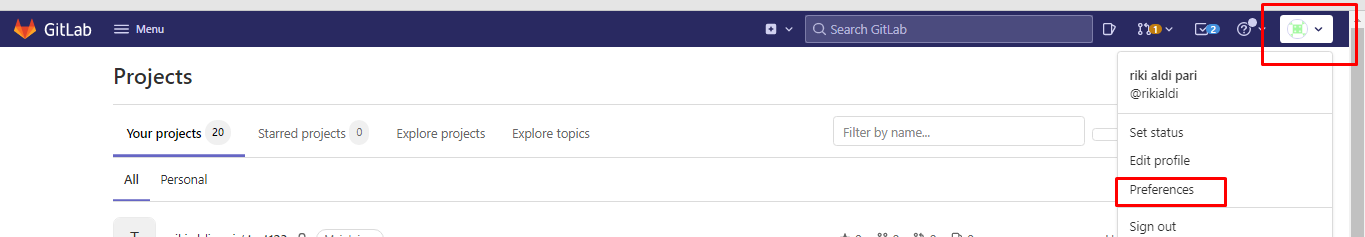


Gitlab project

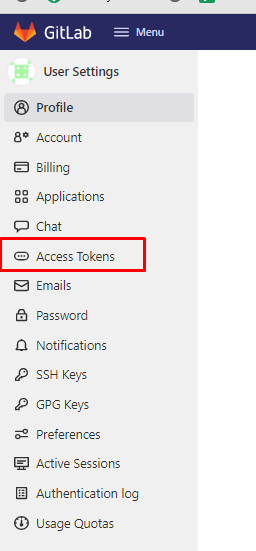
## Menambahkan project pada gitlab anda

* Membuat access token

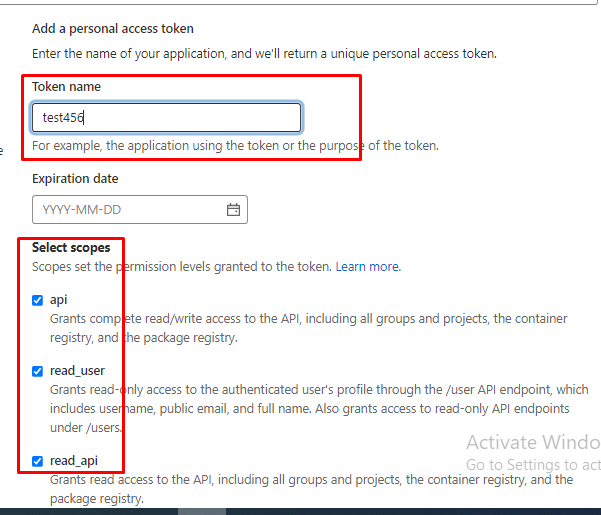
Pilih preferences



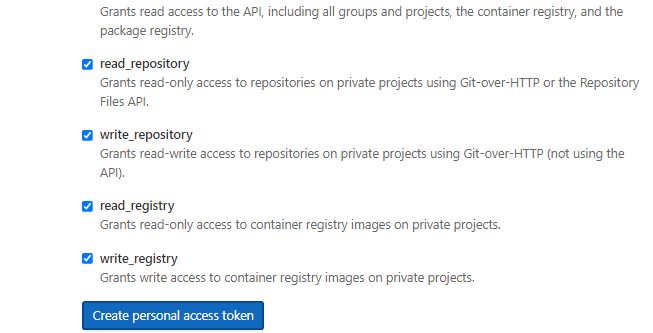
Pilih access token



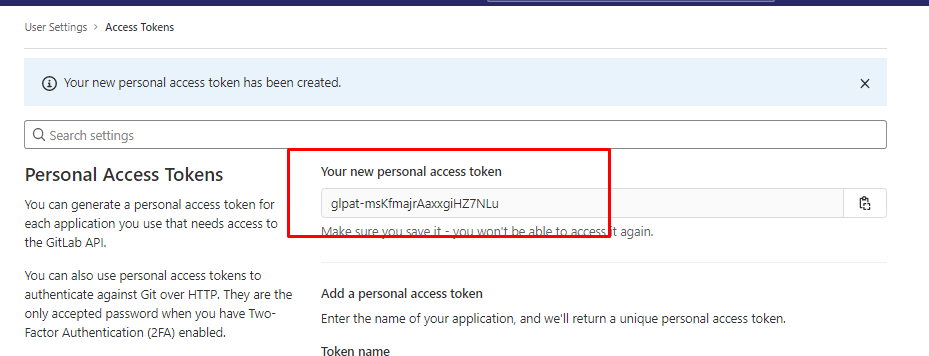
Isi token name dan select semua scopes



Create personal access token



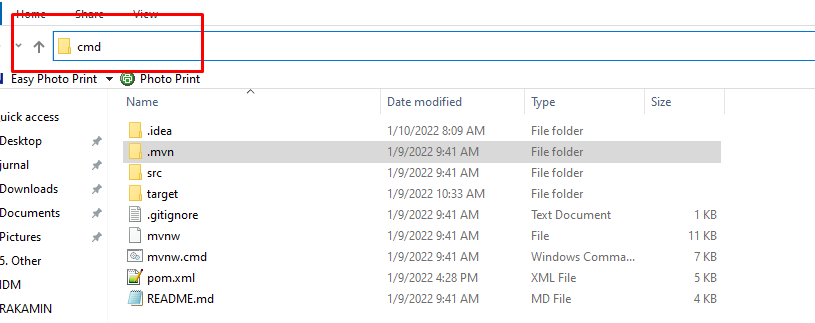
Token sudah digenerate dan simpan token tersebut



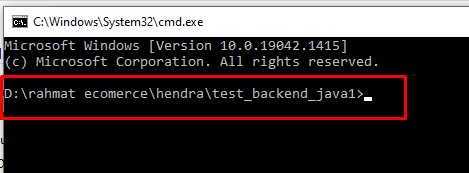
* Membuat Project yang akan ditambahkan pada gitlab

Buatlah project di local anda, kemudian masuk pada terminal :

Ketikkan “cmd” pada menu filter windows anda



Sehingga tampilan di terminal seperti berikut



Add project ke gitlab

Jika project sudah ada di gitlab

git remote set-url origin [https://[user]:[token] @gitlab.com/[user]/](https://[user]:[token]%20@gitlab.com/%5buser%5d/)[namaproject]

**contoh**

git remote set-url origin [https://rikialdi:sdvnmdkfvndfvdfnvdfbdfb@gitlab.com/rikialdi/](https://rikialdi:[token%20anda%20masukkan%20disini]@gitlab.com/rikialdi/%5bnama%20project%20anda%5d)test123.git

Jika project belum ada di gitlab

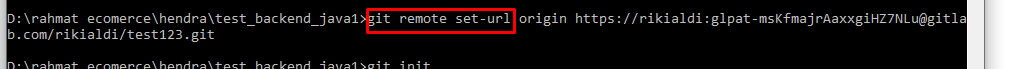
Step 1 : git init

Step 2: git remote add-url origin [https://[user]:[token] @gitlab.com/[user]/](https://[user]:[token]%20@gitlab.com/%5buser%5d/)[namaproject]

**contoh**

git remote set-url origin <https://rikialdi:sdvnmdkfvndfvdfnvdfbdfb@gitlab.com/rikialdi/test123.git>

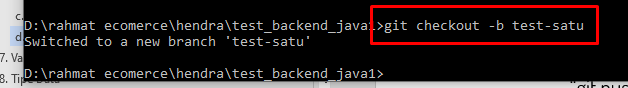
misal project di gitlab sudah pernah ada, maka syntax nya seperti berikut:



Perintah **membuat branch baru** dengan nama test-satu:

“git checkout -b test-satu”

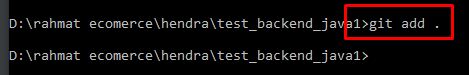
Perhatikan gambar dibawah ini :



Perintah **menambahkan** semua file :

“git add .”

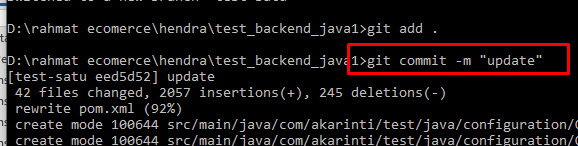
Perhatikan gambar dibawah ini :



Perintah **commit** semua file :

“git commit -m “push project” ”

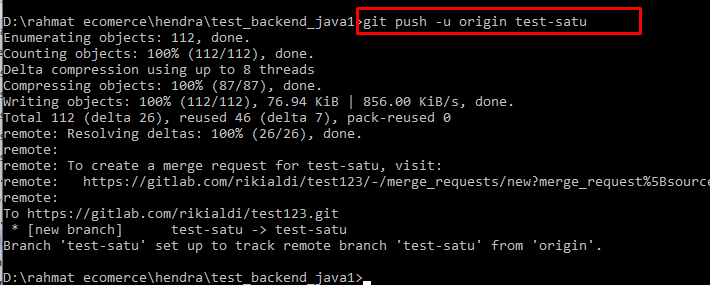
Perhatikan gambar dibawah ini :



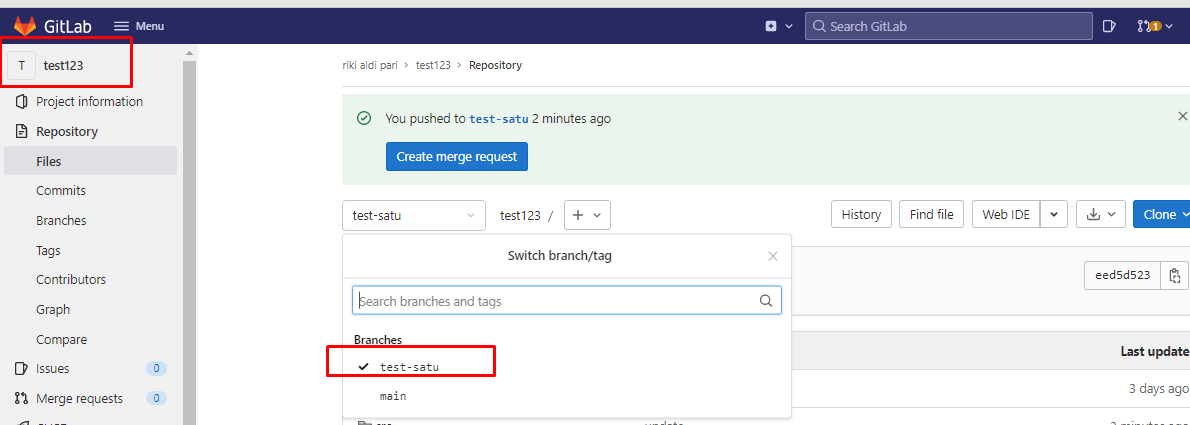
Perintah **push** semua file :

“git push -u origin test-satu”

Perhatikan gambar dibawah ini :



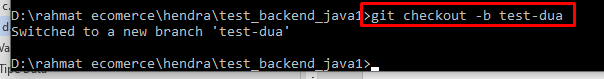
Project berhasil ditambahkan pada gitlab anda. Perhatikan gambar dibawah ini:



Perintah **clone(create branch baru dari branch yang ada) branch** test-satu menjadi test-dua:

“git checkout -b test-dua”

Perhatikan gambar dibawah ini :

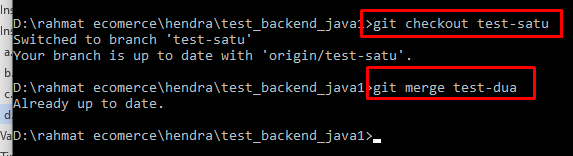


Perintah **merge branch** . misal ingin merge branch test-dua menjadi test-satu.

Step 1 : kembali ke branch test-satu : git checkout test-satu

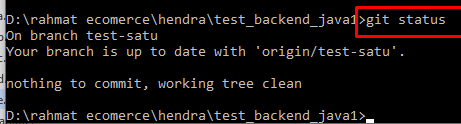
Step 2 : lakukan perintah merge : git merge test-dua

Perhatikan gambar berikut supaya lebih jelas



## Perintah melihat status perubahan file

Perintah dengan “git status”. Perhatikan gambar dibawah ini :



# Variabel

**Apa itu Variabel pada Java ?**

Variable merupakan tempat penyimpanan atau penampung nilai atau data di dalam memori. Variabel terdiri dari terdiri dari tipe data dan nama variabel. Tipe data menentukan jenis nilai atau data yang akan disimpan, sedangkan nama variabel menjadi pengenal (identifier), seperti halnya orang akan dipanggil dengan nama yang ia miliki, begitu-pun variabel.

**Deklarasi Variabel**

|  |
| --- |
| typedata variable = value; |

**Contoh program sbb :**

|  |
| --- |
| String namaVariabel = “nama”; |

Sesuai dengan namanya, isi variabel bisa berubah sepanjang kode program. Sebagai contoh, jika saya membuat program menghitung luas persegi, maka bisa menyiapkan variabel panjang dan lebar yang diisi angka 10 dan 12. Kemudian nanti isi variabel panjang dan lebar ini bisa ditukar dengan angka 20, 50, atau nilai lain.

Variabel juga bisa dipakai untuk menampung nilai inputan, misalnya kita ingin panjang dan lebar ini diisi oleh user (pengguna aplikasi). Terkait tentang cara input data ke dalam kode program Java akan kita bahas dalam tutorial terpisah.

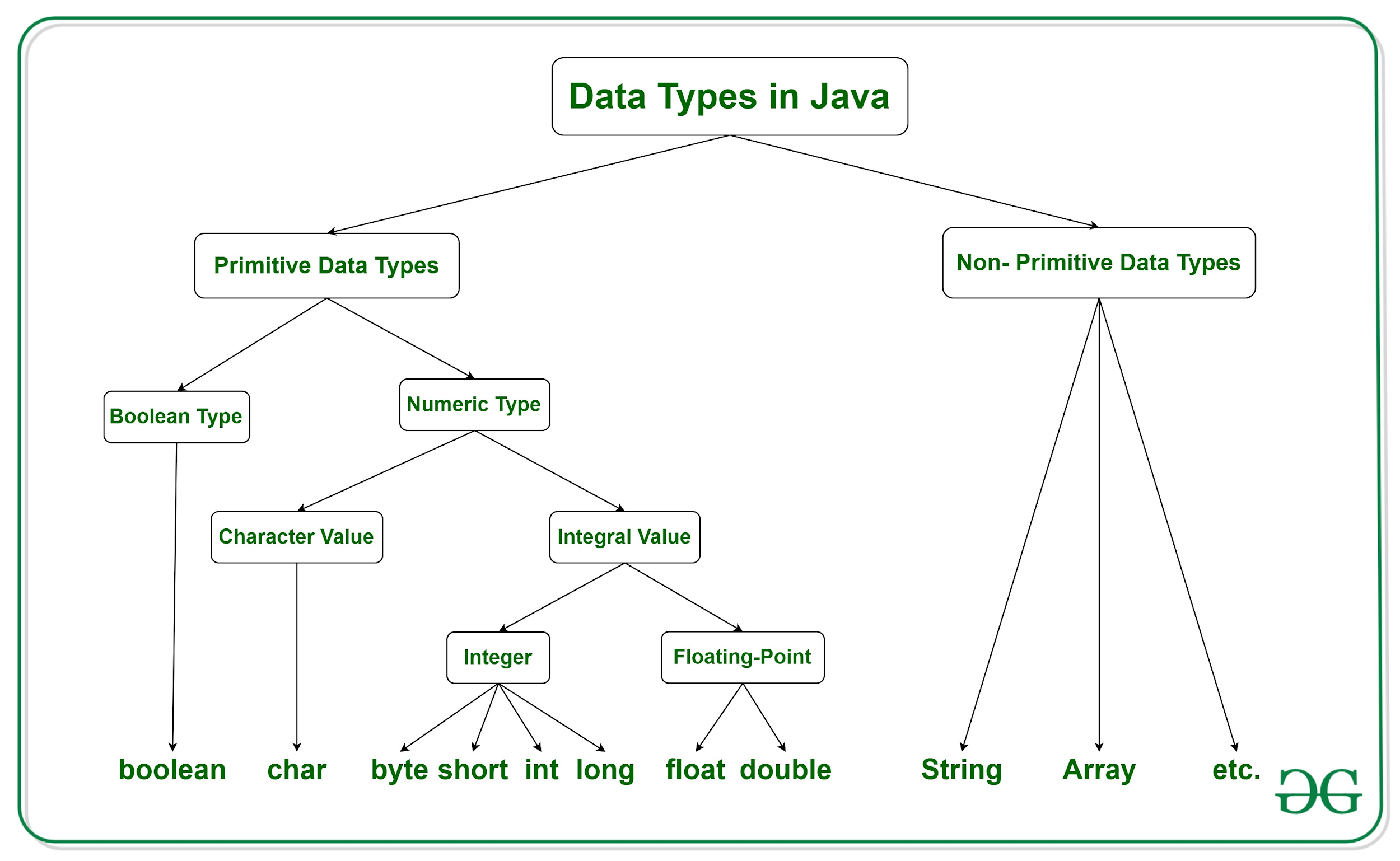
**Aturan Penamaan Variabel dalam Bahasa Java**

* Variabel bisa terdiri dari huruf, angka dan karakter underscore / garis bawah ( \_ ).
* Karakter pertama dari variabel hanya boleh berupa huruf dan underscore ( \_ ), tidak bisa berupa angka.
* Variabel harus selain dari *keyword*. Sebagai contoh, kita tidak bisa memakai kata **int** sebagai nama variabel, karena **int** merupakan *keyword* untuk menandakan tipe data **integer**.
* Nama variabel sebaiknya ditulis menggunakan gaya penulisan **camelCase**, dimana setiap kata juga diawali dengan huruf besar, kecuali kata pertama dan tanpa spasi. Contoh: **panjang**, **panjangSegitiga**, atau **luasSegitiga**.

# Tipe Data

**Apa itu Tipe Data ?**

Tipe data adalah jenis data yang tersimpan dalam variabel.



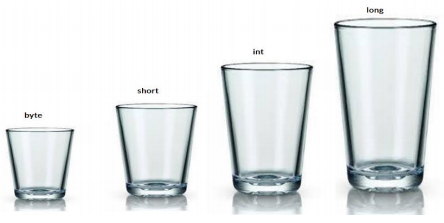
Gambar Jenis Type Data

## Tipe Data Primitif

Tipe data primitif dapat digunakan untuk menyimpan satu nilai dalam satu waktu ke dalam sebuah variable. Tipe data primitif tersebut diantaranya sbb :

* byte
* short
* int
* long
* float
* double
* char
* boolean

byte, short, int and long merupakan tipe data primitif yang dikelompokkan kedalam tipe data integer, maksudnya byte, short, int and long variable dapat menampung data dalam bilangan integer, masing masing memiliki jangkauan yang berbeda. Seluruh tipe data integer dapat menampung bilangan integer positif dan negatif.



Gambar Asumsi Panjang Type Data

1. **byte**

Ukuran untuk tipe data byte ini adalah 8 bit. Jangkauan nilai ini adalah antara -128 sampai +127. Variable bertipe data byte di deklarasikan dengan keyword byte.

|  |
| --- |
| byte num1 = 120; |

1. **Short**

Ukuran untuk tipe data short ini adalah 16 bit. Jangkauan nilai ini adalah antara -32768 sampai +32767. Variable bertipe data short di deklarasikan dengan keyword short.

|  |
| --- |
| short num2 = 10000; |

1. **Int**

Ukuran untuk tipe data int ini adalah 32 bit. Jangkauan nilai ini adalah antara –2,147,483,648 to +2,147,483,647. Variable bertipe data int di deklarasikan dengan keyword int.

|  |
| --- |
| int num3 = 123; |

1. **Long**

Ukuran untuk tipe data long ini adalah 64 bit. Jangkauan dari tipe data ini cocok untuk mendeklarasikan nilai-nilai yang besar. Variable bertipe data int di deklarasikan dengan keyword long.

|  |
| --- |
| long num4 = 1234567891; |

Tabel Jangkauan Tipe Data Integer:

|  |  |  |
| --- | --- | --- |
| **Tipe Data** | **Ukuran (bit)** | **Range** |
| Byte | 8 | -128 s.d. 127 |
| Short | 16 | -32768 s.d. 32767 |
| Int | 32 | -2147483648 s.d. 2147483647 |
| long | 64 | -9223372036854775808  s.d. 9223372036854775807 |
|  |  |  |

Untuk nilai float dan double merupakan tipe data floating dengan kapasitas penyimpanan yang berbeda.

1. **Float**

Ukuran untuk tipe data float ini adalah 32 bit. Jangkauan nilai ini adalah antara –1.4e–045 sampai 3.4e+038. Variable bertipe data float di deklarasikan dengan keyword float. float merupakan single precision. Variables cocok untuk merepresentasikan nilai dollars dan nilai cents.

|  |
| --- |
| float num5 = 10000f; |

1. **Double**

Ukuran untuk tipe data double ini adalah 64 bit. Jangkauan nilai ini adalah antara 4.9e–324 sampai 1.8e+308. Variable bertipe data float di deklarasikan dengan keyword float. float merupakan single precision. Variables cocok untuk merepresentasikan nilai ke akuratan nya lebih tinggi.

|  |
| --- |
| double num6 = 0.122; |

Tabel Jangkauan Tipe Data floating point:

|  |  |  |
| --- | --- | --- |
| **Tipe Data** | **Ukuran (bit)** | **Range** |
| float | 32 | 1.4e-045 to 3.4e+038 |
| double | 64 | 4.9e-324 to 1.8e+308 |
|  |  |  |
|  |  |  |
|  |  |  |

1. **Char**

tipe data char ini dapat mengisi variabel dengan satu huruf dalam satu waktu, char di java berbeda dengan char pada bahasa c/c++. Pada Bahasa c/c++ ukurannya 8 bits sedangkan di java 16 bits dalam bentuk Unicode Range nya 0 sampai 65535 dan tidak ada negative char. Variable dengan tipe data ini di deklarasikan dengan keyword char.

|  |
| --- |
| char num7 = '1';  char num8 = '%'; |

1. **Boolean**

variable dengan tipe data boolean digunakan untuk menyimpan data logika. Dimana nilai yang disimpan berupa nilai true atau false. Tipe data boolean dideklarasikan dengan keyword boolean.

|  |
| --- |
| boolean num9 = false; |

## Tipe Data Non Primitif (Reference Tipe Data)

Pada tipe data non primitif dariabel selalu menyimpan data berupa alamat data. Selalu di sebut dengan tipe data reference karena merifer ke lokasi memori di mana data di simpan. Tipe data primitif ini diantaranya class, array, enum dll.

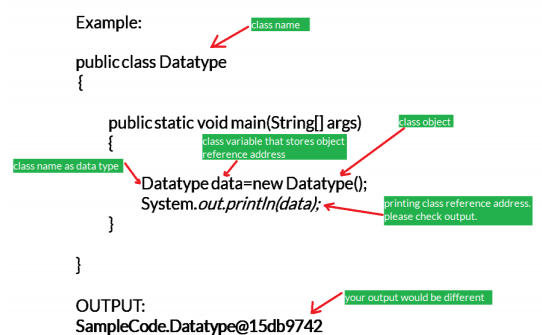
1. **Class**

Class adalah sebuah tipe data non primitif. Digunakan untuk menyimpan alamat rujukan dari sebuah objek. Tipe data class di deklarasikan menggunakan class name.

Class di bagi kedalam 2 bagian

* Class yang di defenisikan oleh user (UDF)
* Class yang sudah di siapkan olej jdk.

Contoh class yang di definisikan oleh user sbb :



Gambar Contoh Class pada java

Contoh class yang sudah ada dalam jdk :

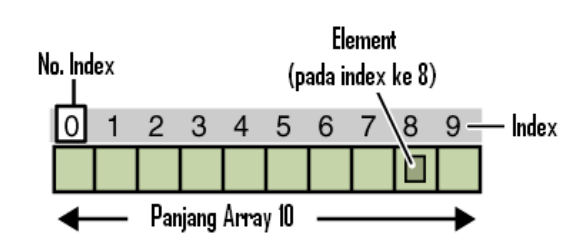
Semua class yang sudah di sediakan oleh java, selalu berawalan huruf besar, jadi berbeda dengan tipe data primitif, dimana class tipe data memiliki method yang dapat dioperasikan untuk beberapa hal. Diantaranya sbb:

* Byte
* Short
* Integer
* Long
* Float
* Double
* Character
* Boolean
* String dll

1. **Array**

Array juga termasuk tipe data non primitif. Array merupakan sebuah solusi untuk mendeklarasikan sejumlah Variabel secara tepat. Pemakaian Variabel Array akan menghemat waktu penyebutan nama Variabel. Variabel Array adalah sejumlah Variabel dengan nama yang sama.

Ilustrasi Array:



Gambar Index Array

Catatan: No. Indeks/No. Element di awali angka 0 (nol) dan diakhiri n-1 dimana n adalah panjang array.

**Deklarasi Array 1 Dimensi**

Beberapa cara mendeklarasikan sebua variabel array, yaitu:

**Deklarasi**

|  |
| --- |
| TypeData[] namaArray; contoh int[] angka; |

**Deklarasi dan Inisialisasi**

|  |
| --- |
| typeData[]namaarray = new typedata[jumlah elemen]  Contoh  int[] angka1 = new int[5]; |

**Deklarasi otomatis**

|  |
| --- |
| typeData[]namaarray = {daftar elemen}  Contoh  int[] angka1 = {1,2,3,4,5,6}; |

**Contoh program array 1 dimensi**

|  |
| --- |
| public static void main (String args[])  {  //Deklarasi array  String[] mahasiswa = {"Rini","Aldi","Acep","Putra"};  //Menampilkan isi array pada indeks ke-2  System.out.println(mahasiswa[2]);  } |

**Array Multidimensi**

Selain berupa sederetan variabel satu dimensi, kita dapat pula membuat array yang berukuran lebih dari satu dimensi atau disebut array multi-dimensi. Pada bagian ini kita mencoba mencontohkan bentuk array dua dimensi sbb:  
  
**typedata[][] namarray = new typedata [jumlah baris][jumlah kolum]**

**Contoh**

String[][] meja = new String[2][3];

Contoh program

|  |
| --- |
| public static String cetakArray(int bariske, int columke){  String[][] kontak = {  {"Lili","08111"},  {"Lala","08122"},  {"Maya","08133"}  };  return kontak[bariske][columke];  }  cetakArray (0,1); |

# Function

**Apa itu Fungsi/ *Function***

Method/fungsi pada bahasa pemrograman Java adalah kumpulan baris kode yang dikelompokkan untuk menjalankan tugas tertentu.

## Cara Membuat Fungsi di Java

Fungsi harus dibuat atau ditulis di dalam *class*.

Struktur dasarnya seperti ini:

|  |
| --- |
| *static* TypeDataKembalian namaFungsi(){  // statemen atau kode fungsi  } |

Penjelasan:

* Kata kunci static, artinya kita membuat fungsi yang dapat dipanggil tanpa harus membuat instansiasi objek.
* TypeDataKembalian adalah tipe data dari nilai yang dikembalikan setelah fungsi dieksekusi.
* namaFungsi() adalah nama fungsinya. Biasanya ditulis dengan huruf kecil di awalnya. Lalu, kalau terdapat lebih dari satu suku kata, huruf awal di kata kedua ditulis kapital.

**Contoh**

|  |
| --- |
| static void ucapSalam(){  System.out.println("Selamat Pagi");  } |

Tipe data **void** artinya kosong, fungsi tersebut tidak mengebalikan nilai apa-apa.

## Cara Memanggil/Eksekusi Fungsi

Setelah kita membuat fungsi, selanjutnya kita akan mengeksekusi fungsinya.

Fungsi dapat dipanggil dari fungsi main atau dari fungsi yang lainnya.

Contoh pemanggilan fungsi dalam dalam funsgi main:

|  |
| --- |
| public static void main(String[] args){  ucapSalam();  } |

Maka akan menghasilkan output:

|  |
| --- |
| Selamat Pagi |

## Fungsi dengan Parameter

Parameter adalah variabel yang menampung nilai untuk diproses di dalam fungsi. Parameter berperan sebagai input untuk fungsi.

Struktur dasarnya seperti ini:

|  |
| --- |
| static TipeData namaFungsi(TipeData namaParameter, TipeData namaParameterLain){  // kode fungsi  } |

**Penjelasan**:

Parameter ditulis di antara tanda kurung (...);

Parameter harus diberikan tipe data;

Bila terdapat lebih dari satu parameter, maka dipisah dengan tanda koma.

Contoh fungsi yang memiliki parameter:

|  |
| --- |
| static void ucapin(String ucapan){  System.out.println(ucapan);  } |

Pada contoh tersebut, kita membuat parameter bernama ucapan dengan tipe String. Sehingga kita bisa menggunakan variabel ucapan di dalam fungsi.

Cara pemanggilan fungsi yang memiliki parameter:

|  |
| --- |
| public static void main(String[] args){  ucapin (“apa kabar”);  } |

## Fungsi yang Mengembalikan Nilai

Setelah fungsi memproses data yang diinputkan melalui parameter, selanjutnya fungsi harus mengembalikan nilai agar dapat diolah pada proses berikutnya.

Pengembalian nilai pada fungsi menggunakan kata kunci return.

Contoh:

|  |
| --- |
| static int luasPersegi(int sisi){  int luas = sisi \* sisi;  return luas;  } |

Pada contoh tersebut, kita membuat sebuah parameter bernama sisi. Kemudian fungsi akan mengembalikan nilai dengan tipe int (integer) dari variabel luas.

Contoh pemanggilanya:

|  |
| --- |
| System.out.println("Luas " + luasPersegi(5)); |

Hasil Output:

|  |
| --- |
| Luas adalah 25 |

## Pemanggilan Fungsi di Fungsi Lain

Fungsi-fungsi dapat saling memanggil untuk memproses data.

Contoh, sebuah program Kalkulator Bangun Ruang memiliki fungsi-fungsi: luasPersegi(), luasPersegiPanjang(), luasSegitiga(), luasBalok(), luasKubus() dsb.

Fungsi-fungsi tersebut dapat saling membantu, contoh fungsi luasKubus() membutuhkan fungsi luasPersegi().

Rumus:

Luas Kubus = 6 \* luasPersegi;

Luas Persegi = sisi \* sisi;

Maka programnya bisa dibuat seperti ini:

|  |
| --- |
| // membuat fungsi luasPersegi()  static int luasPersegi(int sisi){  return sisi \* sisi;  }  // membuat fungsi luasKubus()  static int luasKubus(int sisi){    // memanggil fungsi luasPersegi  return 6 \* luasPersegi(sisi);  } |

## Fungsi Static dan Non-Static

Pada contoh-contoh diatas, kita menggunakan kata kunci static sebelum membuat fungsi.

Kata kunci static akan membuat fungsi dapat dieksekusi langsung, tanpa harus membuat instansiasi objek dari class.

Contoh

|  |
| --- |
| // Fungsi non-static  void makan(String makanan){  System.out.println("Hi!");  System.out.println("Saya sedang makan " + makanan);  }    // fungsi static  static void minum(String minuman){  System.out.println("Saya sedang minum " + minuman);  } |

Cara pemanggilan

|  |
| --- |
| // fungsi main  public static void main(String[] args) {    // pemanggilan fungsi static  minum("Kopi");      // mambuat instansiasi objek saya dari class FungsiStatic  FungsiStatic saya = new FungsiStatic();  // pemanggilan fungsi non-static  saya.makan("Nasi Goreng");    } |

Apabila kita tidak membuat objek untuk memanggil fungsi *non-static*, maka akan terjadi error.

# Input & Output

**Cara Mengambil Input dan Menampilkan Output ?**

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

* Input: nilai yang kita masukan ke program
* Proses: langkah demi langkah yang dilakukan untuk mengelola input menjadi sesuatu yang berguna
* Output: hasil pengolahan

Java sendiri sudah menyediakan tiga class untuk mengambil input:

* Class Scanner;
* Class BufferReader;
* dan Class Console.
* Tiga class tersebut untuk mengambil input pada program berbasis teks *(console)*. Sedangkan untuk GUI menggunakan class yang lain seperti JOptionPane dan inputbox pada form.
* Sementara untuk outputnya, Java menyediakan fungsi print(), println(), dan format().

## Mengambil Input dengan Class Scanner

*Scanner* merupakan *class* yang menyediakan fungsi-fungsi untuk mengambil input dari keyboard.

|  |
| --- |
| // mengimpor Scanner ke program  import java.util.Scanner;  public class DataKaryawan {  public static void main(String[] args) {  // deklarasi variabel  String nama, alamat;  int usia, gaji;  // membuat scanner baru  Scanner keyboard = new Scanner(System.in);    System.out.print("Nama karyawan: ");  // menggunakan scanner dan menyimpan apa yang diketik di variabel nama  nama = keyboard.nextLine();  // Tampilkan outpu lagi  System.out.print("Alamat: ");  // menggunakan scanner lagi  alamat = keyboard.nextLine();  System.out.print("Usia: ");  usia = keyboard.nextInt();  System.out.print("Gaji: ");  gaji = keyboard.nextInt();  // Menampilkan apa yang sudah simpan di variabel  System.out.println("--------------------");  System.out.println("Nama Karyawan: " + nama);  System.out.println("Alamat: " + alamat);  System.out.println("Usia: " + usia + " tahun");  System.out.println("Gaji: Rp " + gaji);  }  } |

Perlu diperhatikan, penggunakan fungsi untuk mengambil data bergantung dari tipe data yang digunakan.

Misal, tipe datanya adalah *String*, maka fungsi atau *method* yang dipakai adalah nextLine().

Begitu juga dengan tipe data lain, *Integer* menggunakan nextInt(), *Double* menggunakan nextDouble(), dsb.

## Mengambil Input dengan Class BufferReader

Class BufferReader sebenarnya tidak hanya untuk mengambil input dari keyboard saja.

Class ini juga dapat digunakan untuk membaca input dari file dan jaringan.

Class ini terletak di dalam paket java.io.

|  |
| --- |
| 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 varibel nama dengan Bufferreader  System.out.print("Inputkan nama: ");  nama = br.readLine();  // tampilkan output isi variabel nama  System.out.println("Nama kamu adalah " + nama);  }  } |

## Menampilkan Output

Ada beberapa fungsi yang sudah disediakan oleh Java:

* Fungsi System.out.print()
* Fungsi System.out.println()
* Fungsi System.out.format()

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

Lalu apa bedanya?

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

**Contoh**

|  |
| --- |
| public class PrintVsPrinln {  public static void main(String[] args) {  System.out.print("ini teks yang dicetak dengan print()");  System.out.println("sedangkan ini teks yang dicetak dengan println()");  System.out.print("pake print() lagi");  }  } |

# Daftar Pustaka

<https://www.duniailkom.com/tutorial-belajar-java-part-3-pengertian-jre-dan-jdk/>

<https://sis.binus.ac.id/2018/02/02/pengenalan-java-console/>

<https://www.duniailkom.com/tutorial-belajar-java-part-3-pengertian-jre-dan-jdk/>

<https://www.dicoding.com/blog/berkenalan-dengan-intellij-idea/>

<https://id.wikipedia.org/wiki/PostgreSQL>

<https://medium.com/@novancimol12/postman-4f181d625fe1>

<https://www.petanikode.com/git-untuk-pemula/>

<https://dedykuncoro.com/2013/03/macam-macam-tipe-data-pada-java.html>

https://www.duniailkom.com/tutorial-belajar-java-pengertian-variabel-dan-cara-penulisan-variabel-bahasa-java/

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

https://www.petanikode.com/java-prosedur-dan-fungsi/

<https://www.petanikode.com/java-input-output/>

<https://www.niagahoster.co.id/blog/java-adalah/?amp>

<https://www.kodingindonesia.com/cara-install-jdk/>

https://www.dicoding.com/blog/berkenalan-dengan-intellij-idea/