



PEMROGRAMAN BERORIENTASI OBJEK

DASAR
TEORI

MODUL 1 - INTRODUCTION TO JAVA DEVELOPMENT AND GIT

2021/2022

PHOTO BY [HTTPS://UNSPLASH.COM/@WEAREPROCREATOR](https://unsplash.com/@WEAREPROCREATOR)

Electrical Engineering Department 2nd Floor
Faculty of Engineering, University of Indonesia



Modul 1 - Introduction to Java Development and Git

A. Java

Java adalah bahasa pemrograman yang populer, dibuat pada tahun 1995. Java dimiliki oleh Oracle, dan lebih dari 3 miliar perangkat menjalankan Java. Java banyak digunakan untuk:

- Aplikasi seluler (khususnya aplikasi Android)
- Aplikasi desktop
- Aplikasi web
- Server web dan server aplikasi
- game
- Koneksi basis data

Mengapa menggunakan Java:

- Java bekerja pada platform yang berbeda (Windows, Mac, Linux, Raspberry Pi, dll.)
- Ini adalah salah satu bahasa pemrograman paling populer di dunia
- Mudah dipelajari dan mudah digunakan
- Ini open-source dan gratis
- Ini aman, cepat, dan kuat
- Ini memiliki dukungan komunitas yang sangat besar (puluhan juta pengembang)
- Java adalah bahasa berorientasi objek yang memberikan struktur yang jelas untuk program dan memungkinkan kode untuk digunakan kembali, menurunkan biaya pengembangan
- Karena Java dekat dengan C++ dan C#, memudahkan programmer untuk beralih ke Java atau sebaliknya

B. Git dan GitHub

- Git

Git merupakan distributed version control system (VCS) yaitu sistem untuk



melakukan pemantauan atau kontrol terhadap program atau script. Git pertama kali diperkenalkan oleh Linus Torvalds pada tahun 2005 dan masih digunakan oleh developer hingga saat ini, dokumentasi lengkap penggunaan Git dapat dilihat pada <https://git-scm.com/doc>. Instalasi Git pada laptop/komputer dapat dilakukan melalui <https://git-scm.com/downloads> dan memilih sistem operasi pada komputer anda, setelah melakukan instalasi Git anda dapat memastikan instalasi telah berhasil dengan membuka command prompt/terminal dan memasukkan perintah

“git --version” (output dapat berbeda berdasarkan versi yang anda gunakan).

```
[~] git --version  
git version 2.30.1
```

Setelah Git berhasil terpasang pada komputer maka anda sudah dapat menggunakan Git, berikut merupakan beberapa perintah pada Git

Perintah	Fungsi
git init	Menginisiasi direktori menjadi direktori Git
git config --global user.name “Fadhlan Harits”	Memberikan nama pengguna Git
git config --global user.email “harits@netlab.id”	Memberikan nama email pengguna Git
git clone [url]	Mengambil repository dari repository remote dan disimpan pada komputer
git add [file]	Menambahkan file untuk ditrack pada commit berikutnya

Referensi:

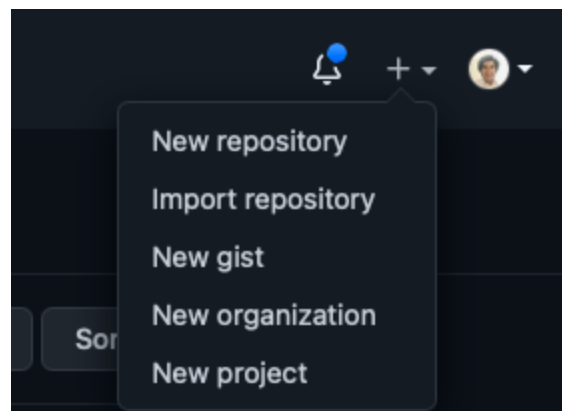
- <https://www.atlassian.com/git/tutorials/what-is-git>
- <https://education.github.com/git-cheat-sheet-education.pdf>



- GitHub

GitHub merupakan platform remote untuk menyimpan source code yang telah dibuat pada komputer/laptop serta dapat digunakan sebagai platform kolaborasi antar developer, halaman utama GitHub dapat dilihat pada <https://github.com>.

Penggunaan platform GitHub akan memerlukan sebuah akun sehingga jika anda belum memiliki akun anda dapat memilih Sign Up pada halaman GitHub, setelah anda memiliki akun maka anda dapat membuat repository untuk menyimpan program. Untuk membuat repository anda dapat memilih tombol + pada bagian kanan atas dan memilih New Repository



Selanjutnya anda akan diminta untuk melakukan konfigurasi pada repository, terdapat beberapa bagian yang perlu diisi seperti owner, nama repository, dan sifat repository (public/private) anda juga dapat memberikan konfigurasi tambahan seperti deskripsi repository serta file Readme.

A screenshot of the GitHub 'Create a new repository' form. The form is dark-themed. At the top, it says 'Create a new repository' in white. Below that, a subtitle explains that a repository contains all project files, including revision history, and offers a link to 'Import a repository' if the user already has one elsewhere. The form has two main sections: 'Owner' and 'Repository name'. The 'Owner' section shows a dropdown menu with the user 'fadhlanhrts' selected. The 'Repository name' section shows a text input with 'OOPisFun' and a green checkmark indicating it's valid. Below these, a message suggests that great repository names are short and memorable, with an example 'glowing-happiness?'. The 'Description (optional)' section has a text input with 'Learning Java and Github'. At the bottom, there are two radio buttons for visibility: 'Public' (selected) and 'Private'. The 'Public' option is described as 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option is described as 'You choose who can see and commit to this repository.'

Referensi:

- <https://guides.github.com/activities/hello-world/>

C. Class dan Objek pada Java

- Class

Kelas adalah sekelompok objek yang memiliki properti umum. Ini adalah template atau cetak biru dari mana objek dibuat. Ini adalah entitas logis. Tidak bisa fisik. Class pada Java dapat berisikan

- Fields
- Methods
- Constructors
- Blocks
- Nested class and interface

- Object

Entitas yang memiliki keadaan dan perilaku dikenal sebagai objek misalnya, kursi, sepeda, spidol, pena, meja, mobil, dll. Itu bisa fisik atau logis (berwujud dan tidak berwujud). Objek adalah instans dari class. Tiga karakteristik objek:



- State: mewakili data (nilai) dari suatu objek.
- Behavior : merepresentasikan perilaku (fungsionalitas) suatu objek seperti deposit, withdraw, dll.
- Identitas: Identitas objek biasanya diimplementasikan melalui ID unik. Nilai ID tidak terlihat oleh pengguna eksternal. Namun, ini digunakan secara internal oleh JVM untuk mengidentifikasi setiap objek secara unik.

D. Tipe Data pada Java

- Primitive: Terdapat 8 tipe data primitif pada Java

Data Type	Size	Description
byte	1 byte	Stores whole numbers from -128 to 127
short	2 bytes	Stores whole numbers from -32,768 to 32,767
int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
double	8 bytes	Stores fractional numbers. Sufficient for storing 15 decimal digits
boolean	1 bit	Stores true or false values
char	2 bytes	Stores a single character/letter or ASCII values

referensi: https://www.w3schools.com/java/java_data_types.asp

- Non-Primitive: Disebut tipe data reference atau referensi karena pada umumnya akan merujuk pada objek tertentu (tipe data yang diambil dari suatu objek)
Contoh tipe data primitif adalah Strings, Arrays, Classes, Interface dan tipe data objek lainnya.
- Perbedaan tipe data Primitif dan Non-Primitive:
 - Tipe primitif sudah ditentukan sebelumnya (sudah ditentukan) di Java. Tipe non-primitif dibuat oleh programmer dan tidak didefinisikan oleh



Java (kecuali untuk String).

- Tipe non-primitif dapat digunakan untuk memanggil metode untuk melakukan operasi tertentu, sedangkan tipe primitif tidak bisa.
- Tipe primitif selalu memiliki nilai, sedangkan tipe non-primitif dapat bernilai nol.
- Tipe primitif dimulai dengan huruf kecil, sedangkan tipe non-primitif dimulai dengan huruf besar.
- Ukuran tipe primitif tergantung pada tipe data, sedangkan tipe non-primitif memiliki ukuran yang sama.

referensi: https://www.w3schools.com/java/java_data_types.asp

- **Type Casting:** Type casting pada java merujuk pada operasi ketika kita menetapkan nilai dari satu tipe data primitif ke tipe lain. Terdapat dua tipe data casting:

- **Widening Casting (automatically)** - converting a smaller type to a larger type size

byte -> short -> char -> int -> long -> float -> double

Contoh:

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int myInt = 9;  
  
        double myDouble = myInt; // Automatic casting  
  
        System.out.println(myInt);        // Outputs 9  
  
        System.out.println(myDouble);     // Outputs 9.0  
  
    }  
  
}
```




- Narrowing Casting (manually) - converting a larger type to a smaller size type

double -> float -> long -> int -> char -> short -> byte

```
public class Main {  
    public static void main(String[] args) {  
        double myDouble = 9.78d;  
        int myInt = (int) myDouble; // Manual casting  
  
        System.out.println(myDouble);    // Outputs 9.78  
        System.out.println(myInt);       // Outputs 9  
    }  
}
```

referensi: https://www.w3schools.com/java/java_type_casting.asp

- Deklarasi variabel pada java: Terdapat beberapa tipe variabel yang berbeda pada java, seperti String, int, float dan lain lain. Terdapat beberapa cara deklarasi variabel salah satu cara deklarasi yang paling mudah dapat mengikuti contoh berikut:

type variable = value;

contohnya adalah: `String name = "John";`



E. Operator pada Java

- Aritmatik

Operator	Name	Description	Example
+	Addition	Adds together two values	$x + y$
-	Subtraction	Subtracts one value from another	$x - y$
*	Multiplication	Multiplies two values	$x * y$
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	$x \% y$
++	Increment	Increases the value of a variable by 1	$++x$
--	Decrement	Decreases the value of a variable by 1	$--x$

- Logika

Operator	Name	Description	Example
&&	Logical and	Returns true if both statements are true	$x < 5 \ \&\& \ x < 10$
	Logical or	Returns true if one of the statements is true	$x < 5 \ \ x < 4$
!	Logical not	Reverse the result, returns false if the result is true	$!(x < 5 \ \&\& \ x < 10)$



- Perbandingan

Operator	Name	Example
==	Equal to	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

referensi: https://www.w3schools.com/java/java_operators.asp

F. Method pada Java

Metode adalah blok kode yang hanya berjalan ketika dipanggil. Anda dapat meneruskan data, yang dikenal sebagai parameter, ke dalam metode. Metode digunakan untuk melakukan tindakan tertentu, dan mereka juga dikenal sebagai fungsi. Metode digunakan untuk menggunakan kembali kode: tentukan kode sekali, dan gunakan berkali-kali.

Suatu method atau metode harus dideklarasikan di dalam kelas dengan didefinisikan dengan nama metode dan dilanjutkan dengan () atau tanda kurung. Java menyediakan beberapa metode yang telah ditentukan sebelumnya, seperti `System.out.println()`, tetapi Anda juga dapat membuat metode sendiri untuk melakukan tindakan tertentu.

Contoh penggunaan method di dalam Main:

```
public class Main {  
    static void myMethod() {  
        // code to be executed  
    }  
}
```



Definisi:

- myMethod() adalah nama metodenya
- static berarti bahwa metode tersebut milik kelas Utama dan bukan objek dari kelas Utama. Anda akan mempelajari lebih lanjut tentang objek dan cara mengakses metode melalui objek nanti dalam tutorial ini.
- void berarti metode ini tidak memiliki nilai balik. Anda akan mempelajari lebih lanjut tentang nilai pengembalian nanti di bab ini

referensi: https://www.w3schools.com/java/java_methods_param.asp

G. Access Modifier pada Java

- Default

Ketika kita tidak menyebutkan pengubah akses apa pun, itu disebut *access modifier* default. Lingkup modifier ini terbatas pada paket saja. Ini berarti bahwa jika kita memiliki kelas dengan pengubah akses default dalam sebuah paket, hanya kelas-kelas yang ada dalam paket ini yang dapat mengakses kelas ini. Tidak ada kelas lain di luar paket ini yang dapat mengakses kelas ini. Demikian pula, jika kita memiliki metode default atau anggota data di kelas, itu tidak akan terlihat di kelas paket lain. Mari kita lihat contoh untuk memahami ini:

- Private

Sesuai dengan namanya, *private* atau *pribadi*. Artinya hanya bisa diakses oleh pemiliknya. Sehingga, sesuatu yang diberikan *access modifier private* hanya dapat diakses oleh *class* nya sendiri. Ada beberapa ketentuan khusus tentang penggunaan *access modifier* ini yakni:

- Data dan method *private* hanya bisa diakses di dalam *class* yang memilikinya
- *Class* dan *Interface* tidak dapat dideklarasikan sebagai *private*
- Jika sebuah *class* memiliki konstruktor *private* maka kita tidak dapat membuat *object* dari *class* tersebut di luar *class* tersebut.



- **Public**

Anggota, metode, dan kelas yang dideklarasikan publik dapat diakses dari mana saja. Pengubah ini tidak membatasi akses.

- **Protected**

Anggota dan metode data yang protected hanya dapat diakses oleh kelas-kelas dari paket yang sama dan subkelas yang ada dalam paket apa pun. Anda juga dapat mengatakan bahwa pengubah akses yang dilindungi mirip dengan pengubah akses default dengan satu pengecualian bahwa ia memiliki visibilitas di sub kelas. Kelas tidak dapat dinyatakan terlindungi. Access modifier ini umumnya digunakan dalam hubungan parent child.

	Class	Package	Subclass (same package)	Subclass (diff package)	Outside Class
public	Yes	Yes	Yes	Yes	Yes
protected	Yes	Yes	Yes	Yes	No
default	Yes	Yes	Yes	No	No
private	Yes	No	No	No	No

Referensi:

- <https://beginnersbook.com/2013/05/java-access-modifiers/>