

OBJECT-ORIENTED PROGRAMMING

MODUL 1 INTRODUCTION TO JAVA DEVELOPMENT AND GIT

BASIC THEORY
2022/2023

NETWORK LABORATORY TEAM





Modul 1 - Basic Java 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

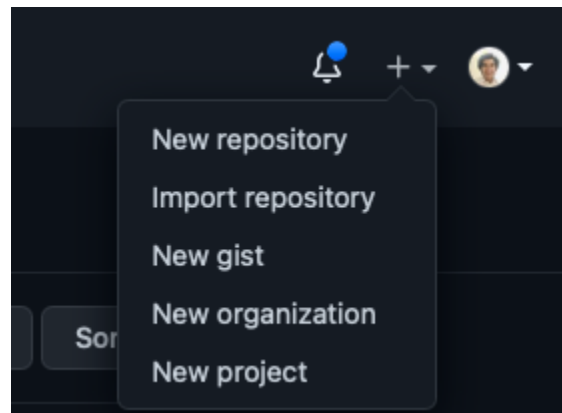
Perintah	Fungsi
git init	Menginisiasi direktori menjadi direktori Git
git config --global user.name “Fadhlan Harits”	Memberikan nama pengguna Git
git config --global user.email “harits@netlab.id”	Memberikan nama email pengguna Git
git clone [url]	Mengambil repository dari repository remote dan disimpan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



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Owner * **Repository name ***


 fadhlanhrts / OOPisFun 

Great repository names are short and memorable. Need inspiration? How about **glowing-happiness?**

Description (optional)

Learning Java and Github

☐  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☒  **Private**
You choose who can see and commit to this repository.

Referensi:

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

C. 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

- Tipe data Characters
Character atau char merupakan tipe data primitif yang berfungsi untuk menyimpan satu karakter. Tipe ini harus menggunakan single quote, seperti 'A' atau 'b'. Tipe data ini dapat dibandingkan nilainya (berdasarkan kode ASCII) dengan operasi >, <, >=, <=, ==, dan !=.

Berikut merupakan beberapa method dari Char.

<code>isDigit(ch)</code>	Returns true if the specified character is a digit.
<code>isLetter(ch)</code>	Returns true if the specified character is a letter.
<code>isLetterOfDigit(ch)</code>	Returns true if the specified character is a letter or digit.
<code>isLowerCase(ch)</code>	Returns true if the specified character is a lowercase letter.
<code>isUpperCase(ch)</code>	Returns true if the specified character is an uppercase letter.
<code>toLowerCase(ch)</code>	Returns the lowercase of the specified character.
<code>toUpperCase(ch)</code>	Returns the uppercase of the specified character.

(Referensi: Introduction to Java Programming, Comprehensive Version, 10th Edition. Daniel Y. Lang)



- Tipe data String

String merupakan tipe data non-primitif yang dapat menyimpan kumpulan karakter. Nilai dari String diapit oleh double quote (“”), misalnya “OOP”. Tipe data ini tidak bisa dibandingkan menggunakan operasi pembandingan matematis seperti pada tipe Char. namun, tipe data ini menyediakan beberapa method untuk melakukan perbandingan seperti berikut ini.

TABLE 4.8 Comparison Methods for String Objects

Method	Description
<code>equals(s1)</code>	Returns true if this string is equal to string s1.
<code>equalsIgnoreCase(s1)</code>	Returns true if this string is equal to string s1; it is case insensitive.
<code>compareTo(s1)</code>	Returns an integer greater than 0, equal to 0, or less than 0 to indicate whether this string is greater than, equal to, or less than s1.
<code>compareToIgnoreCase(s1)</code>	Same as <code>compareTo</code> except that the comparison is case insensitive.
<code>startsWith(prefix)</code>	Returns true if this string starts with the specified prefix.
<code>endsWith(suffix)</code>	Returns true if this string ends with the specified suffix.

Referensi: Introduction to Java Programming, Comprehensive Version, 10th Edition. Daniel Y. Lang

- 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 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";
```


**D. 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

E. Conditional statement

Java memiliki conditional statement seperti berikut:

- Gunakan *if* untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang ditentukan benar
- Gunakan *else* untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang sama salah
- Gunakan *else if* untuk menentukan kondisi baru yang akan diuji, jika kondisi pertama salah
- Gunakan *switch* untuk menentukan banyak blok kode alternatif yang akan dieksekusi

Contoh:

```
int time = 22;
if (time < 10) {
    System.out.println("Good morning.");
} else if (time < 20) {
    System.out.println("Good day.");
} else {
    System.out.println("Good evening.");
}
// Output "Good evening."
```



Selain if-else dan switch, java juga mendukung ternary operator yang cara kerjanya mirip dengan if-else condition. Berikut contohnya:

```
int num = 2;

boolean even = num % 2 == 0 ? true : false;
System.out.println(even); // output = true
```

Contoh di atas serupa dengan menggunakan if-else seperti berikut.

```
int num = 2;
boolean even;
if (num % 2 == 0) {
    even = true;
} else {
    even = false;
}
System.out.println(even); //output = true
```

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