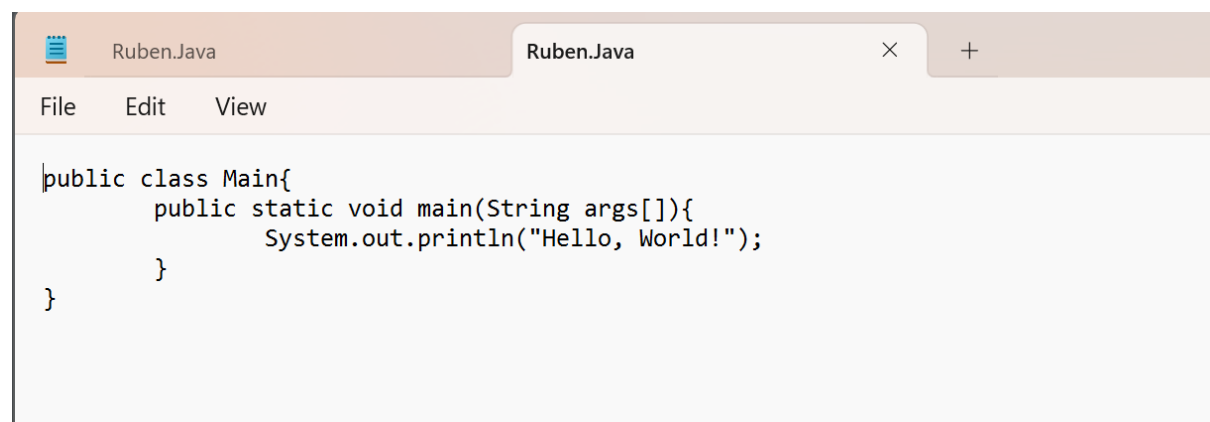


Nama : Ruben Kristanto  
NPM : 2306214624  
Matkul : Praktikum OOP  
Modul : 1 Tugas Pendahuluan

### Part 1 : screenshot version java

```
C:\Users\ruben>java -version
java version "17.0.12" 2024-07-16 LTS
Java(TM) SE Runtime Environment (build 17.0.12+8-LTS-286)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.12+8-LTS-286, mixed mode, sharing)
```

### Program Hello World dalam bentuk java

A screenshot of a code editor window titled 'Ruben.Java'. The editor has a menu bar with 'File', 'Edit', and 'View'. The code inside is a Java class named 'Main' with a 'main' method that prints 'Hello, World!'.

```
public class Main{
    public static void main(String args[]){
        System.out.println("Hello, World!");
    }
}
```

### Kompilasi dengan javac di terminal

```
C:\Users\ruben\Documents>javac Ruben.java

C:\Users\ruben\Documents>dir
Volume in drive C is Acer
Volume Serial Number is 1CF7-747F

Directory of C:\Users\ruben\Documents

01/09/2024  16:14    <DIR>          .
28/08/2024  17:21    <DIR>          ..
06/05/2024  15:38    <DIR>          Custom Office Templates
01/09/2024  16:14                417 Ruben.class
01/09/2024  16:13                109 Ruben.java
22/04/2024  13:17       1.732 Shortcut to Documents (OneDrive - Personal).lnk
               3 File(s)                2.258 bytes
               3 Dir(s)  816.781.320.192 bytes free

C:\Users\ruben\Documents>
```

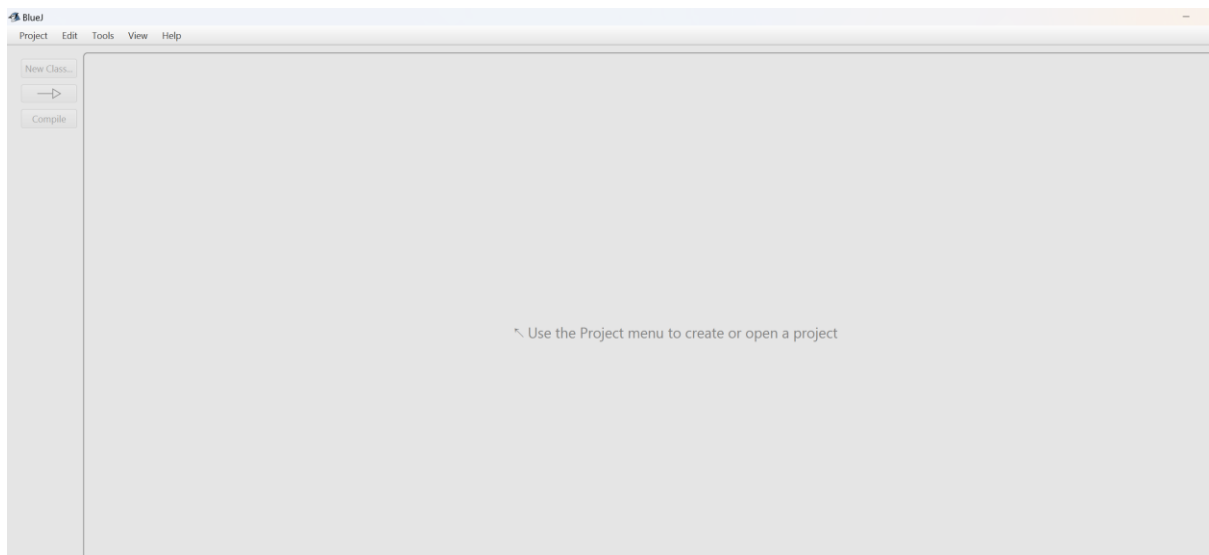
Hasil running codenya

```
C:\Users\ruben\Documents>Java Ruben
Hello, World!

C:\Users\ruben\Documents>|
```

Part 2:

Tampilan awal bluej setelah terinstall



Tampilan version git setelah diinstall

```
C:\Users\ruben\Documents>git --version
git version 2.43.0.windows.1

C:\Users\ruben\Documents>|
```

Part 3, Teori:

1. Tipe data primitive adalah tipe data yang hanya bisa menyimpan satu jenis nilai saja seperti float, integer, double, char dan sebagainya. sedangkan tipe data non primitive adalah tipe data yang bisa menyimpan lebih dari 1 nilai atau value dan dapat diubah sewaktu waktu seperti array, String, class, dan lain-lain.

3 perbedaan dari tipe data primitive dan non primitive :

- Tipe data primitive harus memiliki nilai sedangkan non primitive bisa bernilai null
- Tipe data primitive hanya bisa menyimpan 1 value sedangkan non primitive bisa lebih dari 1 nilai
- Primitif penamaan dimulai dengan huruf kecil sedangkan non primitive penamaan dimulai dengan huruf kapital.

Referensi :

[1] "MODUL 1: Class, Enumeration, Interface | LEARN.NETLAB," *Netlabdte.com*, 2024.  
<https://learn.netlabdte.com/docs/OOP/BasicJava> Diakses : Minggu, 1 September 2024

2. Class adalah blueprint atau struktur untuk membuat suatu objek, misal ada class Bernama baju, dan dari kelas baju kita bisa membuat subclass atau child class berupa jenis jenis baju seperti polo, t-shirt, kemeja, jas, dan sebagainya yang Dimana semuanya memiliki atributnya masing-masing tetapi tetap saling berhubungan, semua child class memiliki atribut atau sifat/nilai dari parent classnya.  
Objek adalah sesuatu yang dibuat untuk merepresentasikan objek dunia nyata berdasarkan atribut atau sifat yang dimiliki oleh class yang mendasari objek tersebut.

Misalnya sebuah class memiliki class Barang

Dengan atribut jenis dan harga, kita dapat membuat objek atau simpelnya variable yang berisikan atribut atau nilai "jenis" dan "harga" tersebut.

3 tipe access modifier :

- Public, Dimana semua class, method, dan field dapat diakses oleh siapapun dan dimanapun.
- Private, Dimana semua method dan field yang ada pada class tersebut hanya dapat diakses di dalam class tempat method atau field itu dibuat.
- Protected, Dimana semua class, method dan field hanya dapat diakses oleh class yang berada dalam package yang sama atau child class dalam package yang sama atau berbeda.

3 tipe class :

- Final, mirip dengan constant menandakan bahwa class tersebut dan isinya tidak akan berubah selama program tersebut berjalan.
- Static class hanya dapat diakses oleh class lain yang juga static biasanya berupa nested class, digunakan untuk penggunaan memori yang lebih efisien.
- Abstract class, adalah class yang dapat diisi method dan constructor abstract atau tidak ada isinya maupun method dan constructor yang sudah diisi, abstract class adalah class yang dapat dibayangkan atau digunakan sebagai perpanjangan sebuah class yang isinya bisa fleksibel(override) maupun tidak.

Referensi :

[1] "Types of Classes in Java," *GeeksforGeeks*, Mar. 05, 2022.  
<https://www.geeksforgeeks.org/types-of-classes-in-java/>

Diakses : Minggu 1 September 2024

[2] [1] "MODUL 1: Class, Enumeration, Interface | LEARN.NETLAB," *Netlabdte.com*, 2024. <https://learn.netlabdte.com/docs/OOP/BasicJava> Diakses : Minggu, 1 September 2024

3. Constructor adalah method yang digunakan umumnya untuk membuat objek baru Sedangkan method adalah function pada umumnya, keduanya dapat dipanggil menggunakan parameter maupun tidak. Constructor adalah method yang pertama dipanggil Ketika sebuah objek ingin mengakses sebuah class.

Referensi :

[1] "MODUL 1: Class, Enumeration, Interface | LEARN.NETLAB," *Netlabdte.com*, 2024. <https://learn.netlabdte.com/docs/OOP/BasicJava> Diakses : Minggu, 1 September 2024

4. interface dapat dibayangkan sebagai blueprint untuk sebuah class yang hanya memiliki method kosong, lalu bagaimana cara menggunakan atau mengisi method ini dapat diisi menurut class yang mengimplementasikan interface tersebut.

Berikut perbedaan abstract class dan interface :

- abstract class menggunakan "extends" sedangkan interface menggunakan "implements"
- abstract class dapat berisi method dan constructor biasa yang memiliki isi, sedangkan interface hanya bisa diisi oleh abstract method
- sebuah class dapat menggunakan atau mengimplentasikan beberapa interface tetapi hanya satu abstract class.

Referensi :

[1] "Difference between Abstract class and Interface - Javatpoint," *www.javatpoint.com*, 2021. <https://www.javatpoint.com/difference-between-abstract-class-and-interface#:~:text=Interface%20supports%20multiple%20inheritance.> (accessed Sep. 01, 2024).

[2] "MODUL 1: Class, Enumeration, Interface | LEARN.NETLAB," *Netlabdte.com*, 2024. <https://learn.netlabdte.com/docs/OOP/BasicJava> Diakses : Minggu, 1 September 2024

- enum adalah tipe class yang digunakan Ketika kita ingin sebuah class memiliki isi yang konstan dan tidak dapat dirubah sama sekali, kita hanya menggunakannya untuk mengakses isi dari class itu, contohnya adalah nama-nama hari, bulan, tipe mobil dan sebagainya.

Referensi :

[1] "MODUL 1: Class, Enumeration, Interface | LEARN.NETLAB," *Netlabdte.com*, 2024.  
<https://learn.netlabdte.com/docs/OOP/BasicJava> Diakses : Minggu, 1 September 2024

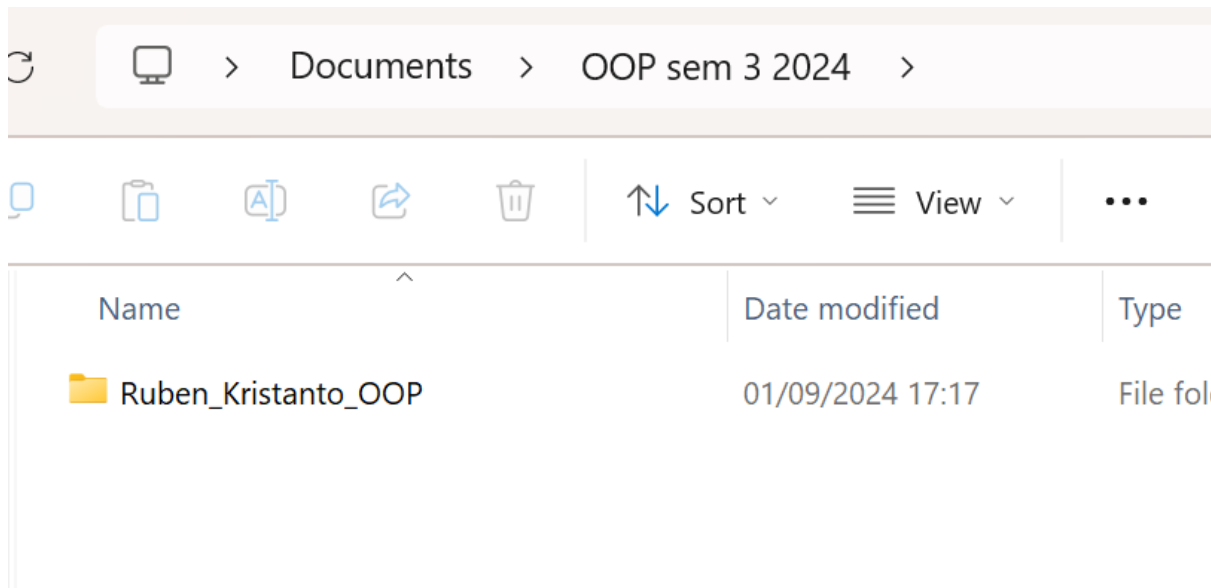
Part 4 :

Langkah Langkah membuat repo

The top screenshot shows the GitHub repository settings for 'RubenKristanto / Ruben-Kristanto\_OOP'. The 'Collaborators' tab is selected, showing a list of users with access. The 'Who has access' section indicates that 8 users have access to this repository. The 'Manage access' section shows a list of collaborators, including Dimas and Emir Fateen Haqqi, with their access status (Pending Invite).

The bottom screenshot shows a Windows File Explorer window titled 'Documents'. It displays a list of folders and files. The 'Custom Office Templates' folder is highlighted, and the 'OOP sem 3 2024' folder is selected.

Name	Date modified	Type	Size
Custom Office Templates	06/05/2024 15:38	File folder	
OOP sem 3 2024	01/09/2024 17:16	File folder	



*git remote add origin https://github.com/Username\_Github>Nama\_Repository.git*

A composite image showing a terminal window and a GitHub repository page. The terminal window, titled 'MINGW64: c:/Users/ruben/Documents/OOP sem 3 2024/Ruben\_Kristanto\_OOP', shows the following commands and output:

```
ruben@Ruben MINGW64 ~/Documents/OOP sem 3 2024/Ruben_Kristanto_OOP
$ git init
Initialized empty Git repository in C:/Users/ruben/Documents/OOP sem 3 2024/Ruben_Kristanto_OOP/.git/

ruben@Ruben MINGW64 ~/Documents/OOP sem 3 2024/Ruben_Kristanto_OOP (master)
$ git remote add origin https://github.com/RubenKristanto_Github/Ruben-Kristanto_OOP.git

ruben@Ruben MINGW64 ~/Documents/OOP sem 3 2024/Ruben_Kristanto_OOP (master)
$
```

The GitHub page below shows the repository 'Ruben-Kristanto\_OOP' by 'RubenKristanto'. It is a private repository with 1 branch and 0 tags. The commit history shows two commits: 'README.TXT' and 'package.bluej', both made 10 minutes ago. The README file is visible, showing a standard template with a project title and purpose.

```
ruben@Ruben MINGW64 ~/Documents/OOP sem 3 2024/Ruben_Kristanto_OOP (main)
$ git remote remove origin

ruben@Ruben MINGW64 ~/Documents/OOP sem 3 2024/Ruben_Kristanto_OOP (main)
$ git remote add origin https://github.com/RubenKristanto/Ruben-Kristanto_OOP.git

ruben@Ruben MINGW64 ~/Documents/OOP sem 3 2024/Ruben_Kristanto_OOP (main)
$ git remote -v
origin https://github.com/RubenKristanto/Ruben-Kristanto_OOP.git (fetch)
origin https://github.com/RubenKristanto/Ruben-Kristanto_OOP.git (push)

ruben@Ruben MINGW64 ~/Documents/OOP sem 3 2024/Ruben_Kristanto_OOP (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 20 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 580 bytes | 580.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/RubenKristanto/Ruben-Kristanto_OOP.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Terdapat beberapa step yang salah, saya akhirnya harus menghapus remote repo lalu meng-add dengan link atau url dari githubnya karena sebelumnya tidak ditemukan data not found.