

Modul 0

Object Oriented Programming

2022



MODUL 0

JAVA INTRODUCTION &
TUTORIAL GITHUB

ASISTEN PRAKTIKUM



**Abiyoga Dhaniswara
ABII**



**Ahmad Syahid Danu
ASDW**



**Amilia Agata
MILY**



**Dharu Widhyanugrah
DINO**



**Erina Zahira Nurhasan
RYIN**



**Fadli Kurniawan
YORU**



**Farel Baihaky
YNWA**



**Fazar Arya Suwandi
RJEP**



**Gede Dipta Narayana
GING**



**Hudzaifah Afif
GGEZ**



**Indira Agustia Garini
DIRA**



**Luh Komang Devi
DEVS**



**Maulana Rakha R
MARK**



**Metta Triana Asri
JMET**



**Mohamad Fahrudin
LFTG**



**M. Nanda Nugraha
NANZ**



**M. Haris Sitompul
RTSU**



**Nabila Melsyana
LALA**



**Rafidah Jasmine L
MINT**



**Windy Kurniawan
WNDD**



**Yoga Raditya N
GOYS**

Contents

Contents	I
Peraturan Praktikum	II
Introduction Java	1
Instalasi Java Development Kit di Visual Studio Code	1
Proyek Java di Visual Studio Code	2
Install Git	5
Membuat Repository Github	10
Commit File ke Github	13

Peraturan Praktikum

1. Setiap peserta praktikum harus datang tepat waktu sesuai dengan jadwal. Toleransi keterlambatan hadir 15 menit. Jika melebihi batas waktu tersebut dan tidak dapat memberikan bukti alasan keterlambatan maka praktikan tidak diperkenankan mengikuti praktikum.
2. Perizinan Praktikum :
 - a. Izin berkaitan dengan Sakit atau Kemalangan , maka praktikan dapat memberikan surat perizinan kepada pihak Komisi Disiplin maksimal 3 hari setelah jadwal (shift) praktikum.
 - b. Izin lomba atau penugasan institusi tidak berlaku apabila tidak terdapat bukti dispensasi dari Igracias. NB : screenshot dispensasi dari igracias wajib dilampirkan.
3. Seragam Praktikum :
 - a. Mahasiswa wajib menggunakan celana bahan hitam (bukan chino atau jeans) pada saat praktikum.
 - b. Mahasiswi wajib menggunakan rok hitam/biru gelap panjang tidak ketat pada saat praktikum.
 - c. Dresscode praktikum (Mengikuti Peraturan Telkom)
 - I. Senin : Menggunakan kemeja merah telkom atau kemeja putih polos
 - II. Selasa s/d Rabu : Menggunakan kemeja putih polos
 - III. Kamis s/d Sabtu : Menggunakan kemeja formal berkerah (bukan kerah sanghai dan bukan polo)
 - IV. Jumat : Menggunakan kemeja/ baju batik bukan outer.
 - d. Jika terdapat kendala dalam baju seragam maka diperbolehkan mengganti menggunakan kemeja putih.
 - e. Membuka sepatu saat memasuki ruangan lab.
4. Peraturan Pengerjaan
 - a. Studi Kasus dikerjakan secara individu.
 - b. Jawaban tidak boleh sama dengan setiap individu.
 - c. Hasil pengerjaan di push ke repository github tiap individu.
 - d. Submit hasil pengerjaan berupa file format PDF dan link github ke LMS tiap individu.
 - e. Format Pengumpulan Hasil Pengerjaan
 - I. Format nama repository :
 - OOP-KODEASISTEN-NAMAPENDEK-NIM
 - Contoh : OOP-MILY-AMILIA-1202213133
 - II. Format nama file Jurnal PDF:
 - OOP_KODEASISTEN_NAMAPENDEK_NIM_SSMODULX
 - Contoh : OOP_MILY_AMILIA_1202204087_SSMODULI

III. Format nama file Tugas Pendahuluan PDF:

- OOP_KODEASISTEN_NAMAPENDEK_NIM_SSTPX
- Contoh : OOP_MILY_AMILIA_1202204087_SSTP1

IV. Format nama folder per modul github

- Jurnal:
 - MODULX_NAMAPENDEK
 - Contoh: MODUL1_AMILIA
- Tugas Pendahuluan:
 - TPMODULX_NAMAPENDEK
 - Contoh: TPMODUL1_AMILIA

f. Salah penamaan pada file pengerjaan nilai modul akan dipotong sebesar 10%

g. Terlambat mengumpulkan file pengerjaan nilai modul akan dipotong sebesar 20%

h. Segala alat komunikasi dikumpulkan di loker yang tersedia dalam ruangan.

i. Jika ada perangkat praktikum yang bermasalah dapat menghubungi asprak yang bertugas.

j. Segala bentuk kecurangan dan plagiarisme akan diproses ke komisi disiplin dan nilai akhir modul menjadi 0

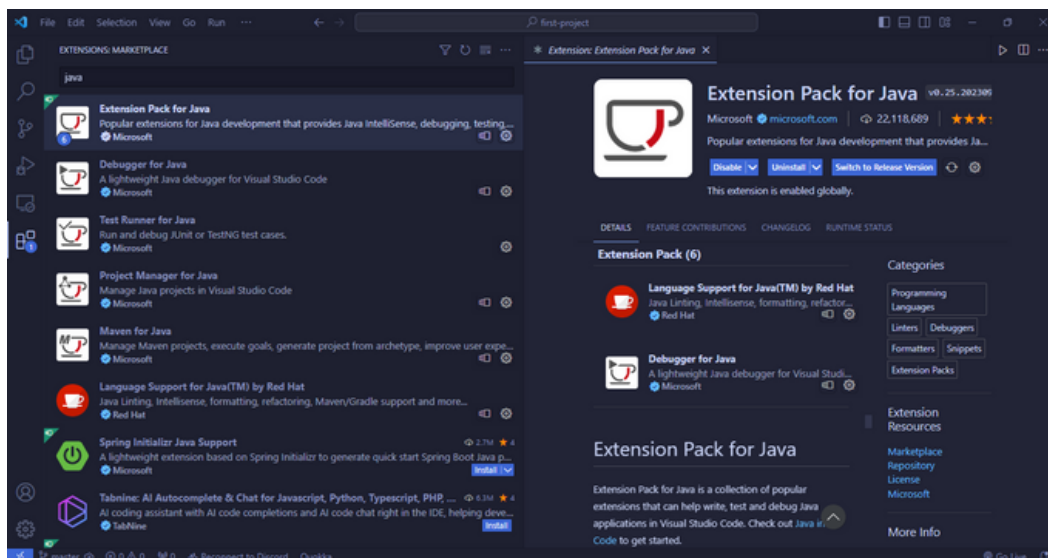
MODUL 0 – INTRODUCTION JAVA

Java merupakan salah satu bahasa pemrograman yang bersifat *open source* serta multiplatform yang berfokus pada pemrograman berorientasi objek. Hal ini karena bahasa pemrograman ini menggunakan sistem *syntax* atau kode pemrograman yang akan diubah menjadi kode *numerik (bytecode)* platform oleh *Java Virtual Machine (JVM)*, sehingga memungkinkan aplikasi Java dapat dijalankan di berbagai perangkat.

A. INSTALASI JAVA DEVELOPMENT KIT (JDK) DI VISUAL STUDIO CODE

Berikut adalah langkah-langkah instalasi JDK pada Visual Studio Code:

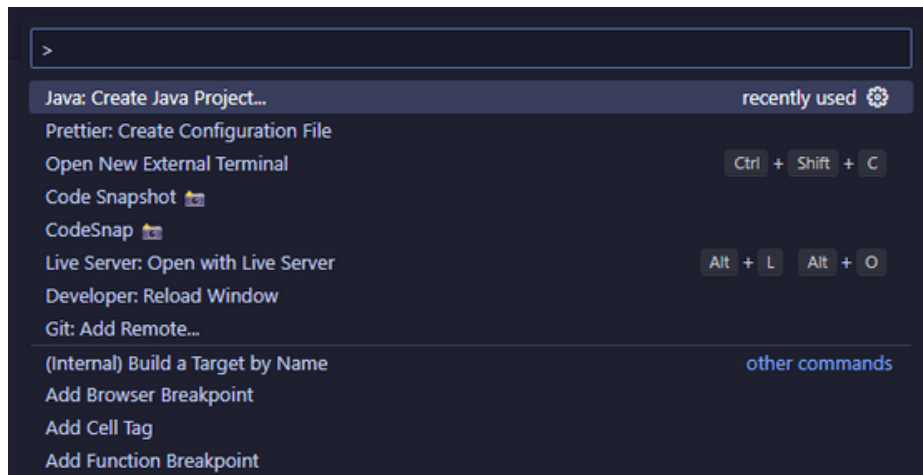
1. Install JDK pada link berikut <https://code.visualstudio.com/docs/java/java-tutorial>.
2. Setelah proses instalasi selesai, buka Visual Studio Code untuk memeriksa apakah ekstensi Java sudah terpasang atau belum.
3. Jika ekstensi Debugger for Java dan Extension Pack for Java telah terpasang, maka Visual Studio Code siap untuk menjalankan program Java.



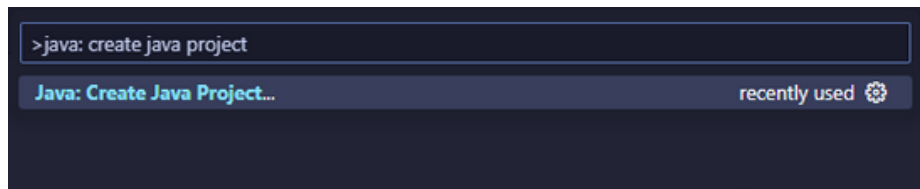
B. PROYEK JAVA DI VISUAL STUDIO CODE (VSC)

Berikut ini adalah langkah-langkah untuk membuat proyek Java dan menjalankan program di Visual Studio Code:

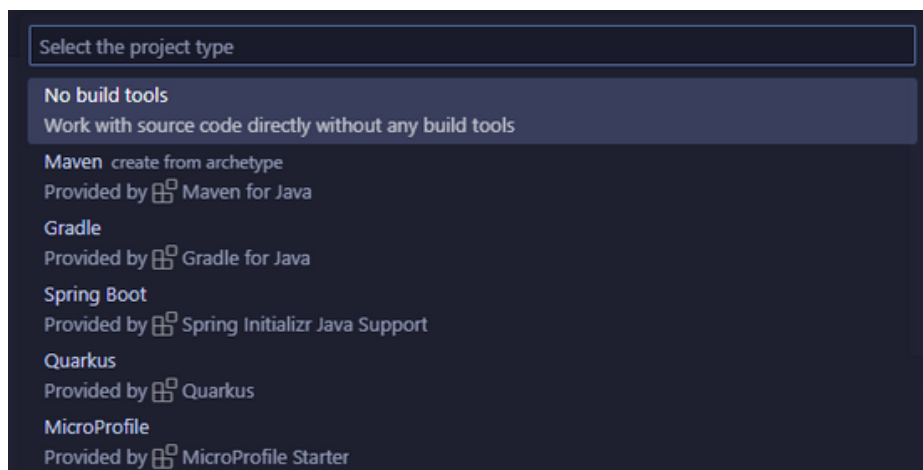
1. Buka *Command Palette* dengan menekan kombinasi **Ctrl+Shift+P** pada Windows atau **Cmd+Shift+P** pada Mac, nanti akan muncul *Command Palette* di bagian atas Visual Studio Code.



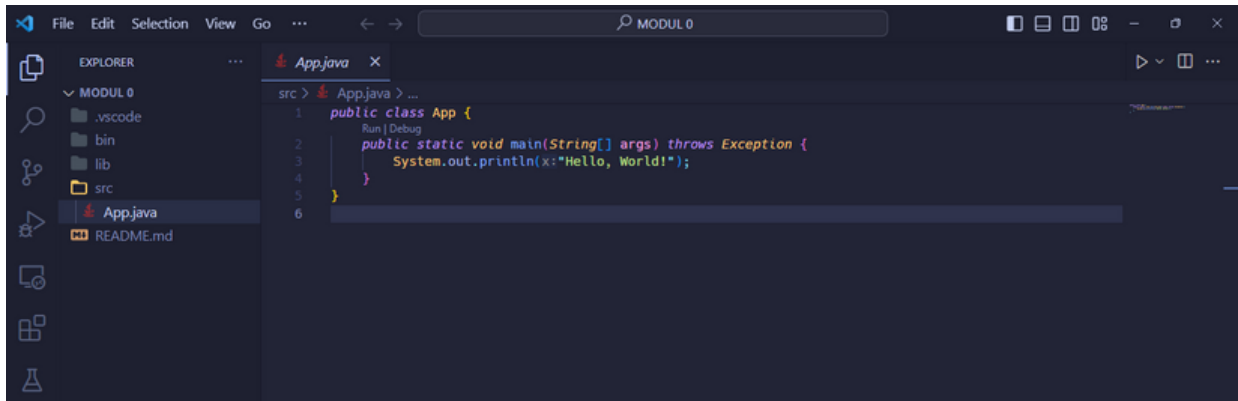
2. Kemudian masukkan "Java: Create Java Project", lalu tekan *Enter*.



4. Akan muncul beberapa jenis proyek yang dapat dibuat, untuk praktikum ini kita menggunakan jenis "No build tools"



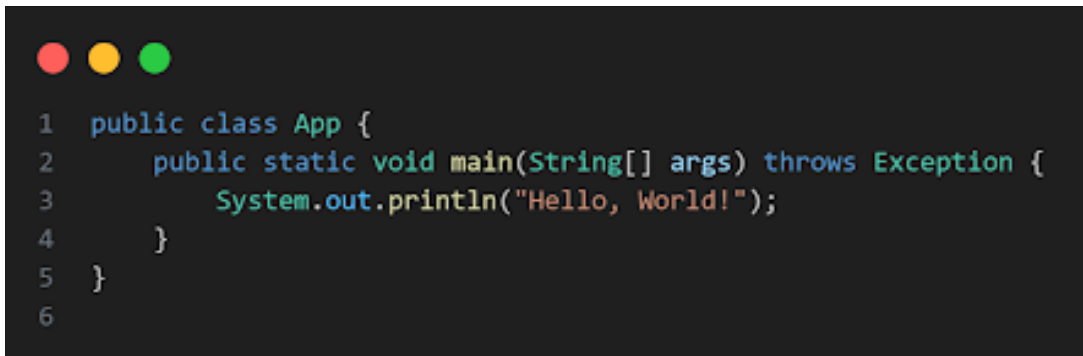
4. Pilih lokasi penyimpanan folder untuk program Java yang akan dibuat, lalu pilih “select the project location”.
5. Kemudian masukkan nama proyek Java yang akan dibuat pada *text field* yang muncul.
6. Jika sudah, VSCode akan memproses proyek yang dibuat dan otomatis membuka folder terkait. Semua kode yang dituliskan nantinya akan berada pada folder src.



7. Untuk mengeksekusi kode ini dapat mengklik tombol run yang berada pada pojok kanan atas atau dapat mengklik tulisan “Run” yang berada tepat di atas deklarasi fungsi



Dari contoh proyek Java yang telah dibuat pada contoh diatas, dapat dijelaskan bahwa:



```
1 public class App {  
2     public static void main(String[] args) throws Exception {  
3         System.out.println("Hello, World!");  
4     }  
5 }  
6
```

- “public class App” mendefinisikan sebuah **class** dengan nama “App” dalam program Java.
- “public static void main(String[] args) throws Exception” adalah metode utama (main method) yang merupakan titik awal dari eksekusi sebuah program. Setiap program Java harus memiliki metode main sebagai **titik mulai eksekusi (entry point)**. Maka dari itu, untuk melakukan running pada program java , dilakukan pada **main method**.
- Jika sebuah program Java tidak memiliki metode main dengan signature yang benar, JVM tidak akan tahu dari mana harus memulai eksekusi program, dan program tersebut tidak dapat dijalankan. Oleh karena itu, metode main dengan signature yang benar adalah komponen penting dalam setiap program Java yang ingin dijalankan.
- Dimana didalam method main terdiri dari:
 - a. “**Public**” menunjukkan bahwa *method* “main” dapat diakses dari mana saja maupun dari luar *class*.
 - b. “**Static**” menunjukkan bahwa *method* “main” bersifat statis (dapat dipanggil tanpa perlu membuat objek dari class “App”).
 - c. “**Void**” merupakan tipe data yang menunjukkan bahwa *method* “main” tidak mengembalikan sebuah nilai.
 - d. “**Main**” merupakan blok program yang akan dieksekusi pertama kali serta menjadi *entry point* dari program.
 - e. “(String[] args)” merupakan parameter *array* dari argumen baris perintah yang dapat diteruskan ke program Java.

INSTALL GIT

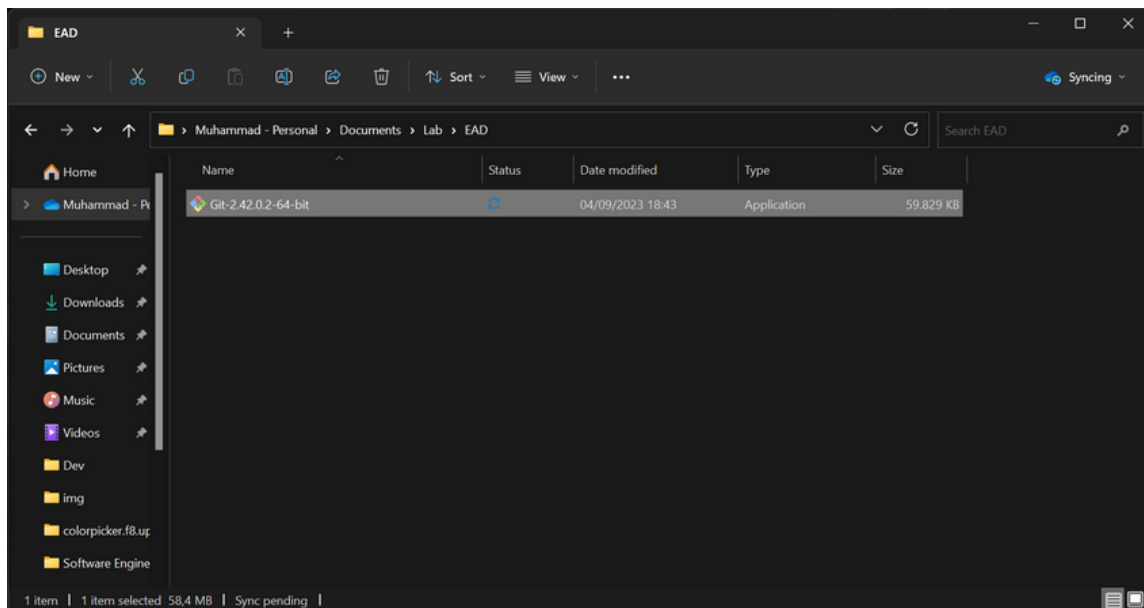
GitHub dan Git adalah dua konsep penting dalam pengembangan perangkat lunak. Git adalah sistem kontrol versi yang digunakan untuk melacak perubahan kode sumber secara lokal di komputer pengembang. Sementara itu, GitHub adalah platform berbasis web yang memanfaatkan Git sebagai dasar untuk menyediakan *hosting* proyek perangkat lunak secara *online*, kolaborasi tim, pelacakan isu, peninjauan kode, dan alat-alat lainnya. Kesimpulannya, Git berperan di komputer pengembang, sementara GitHub adalah platform untuk berbagi, berkolaborasi, dan mengelola proyek perangkat lunak secara lebih efisien dengan dukungan dari Git di belakang layarnya.

Berikut merupakan langkah-langkah untuk instalasi Git bagi yang belum melakukannya:

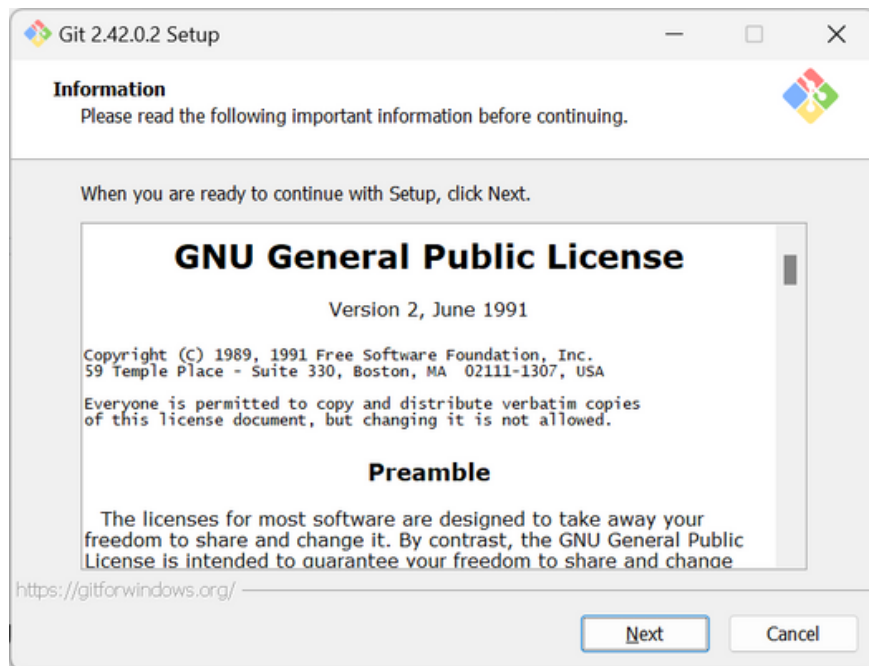
1. Unduh aplikasi Git pada link yang diberikan

<https://drive.google.com/drive/folders/1bbISAYq8i9mcHTo-Ok3pFY8j838uorlv>.

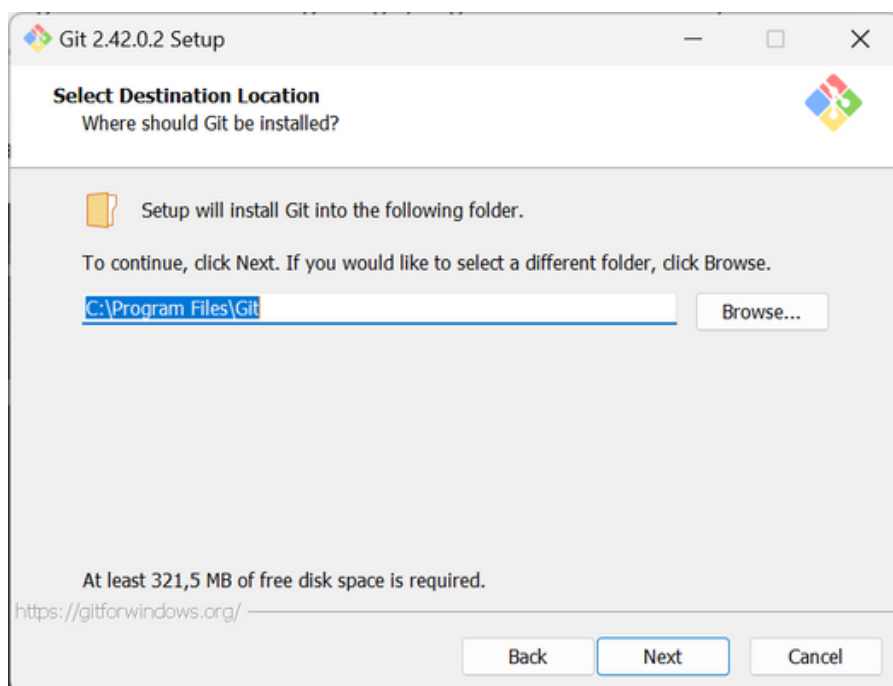
2. Buka *file* Git yang sudah di unduh.



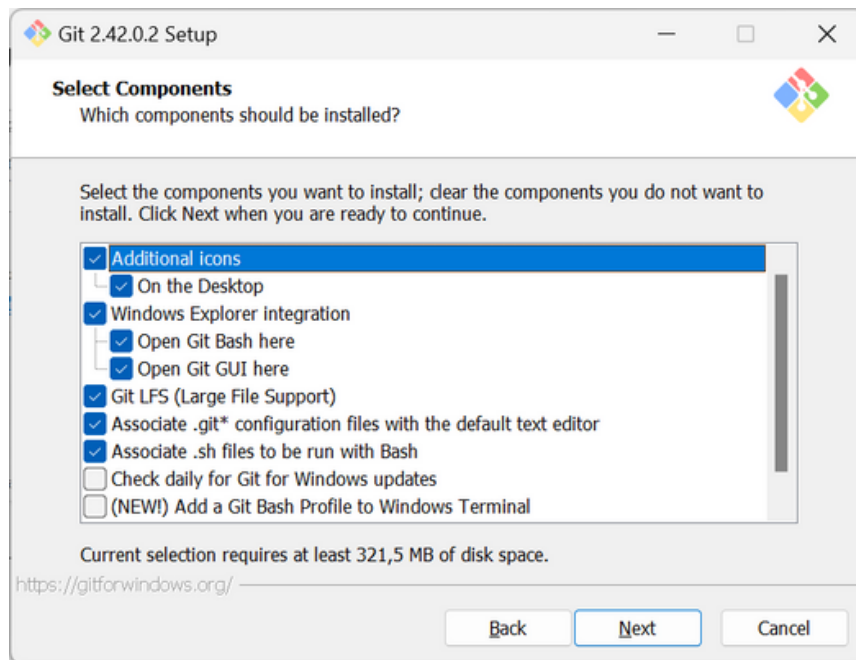
3. Pilih "next".



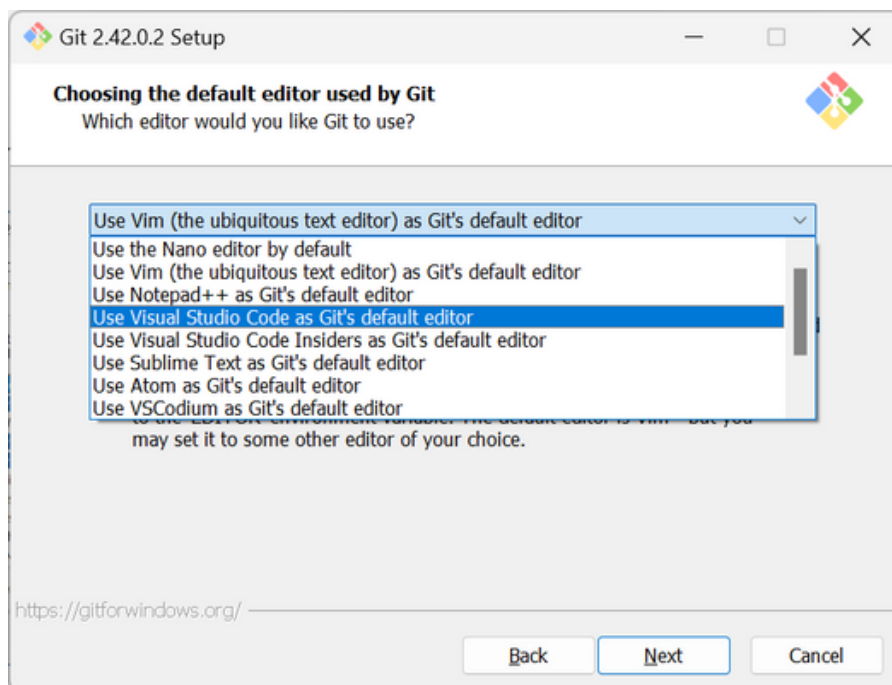
4. Pilih *file directory* yang diinginkan untuk instalasi Git, kemudian klik "next".



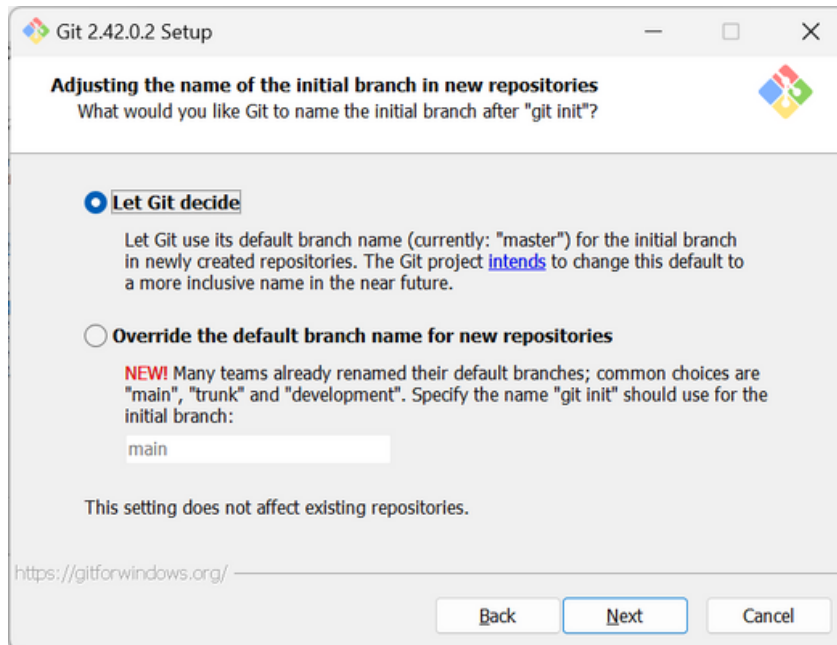
5. Pilih beberapa *components* yang ingin ditambahkan pada Git, kemudian klik “next



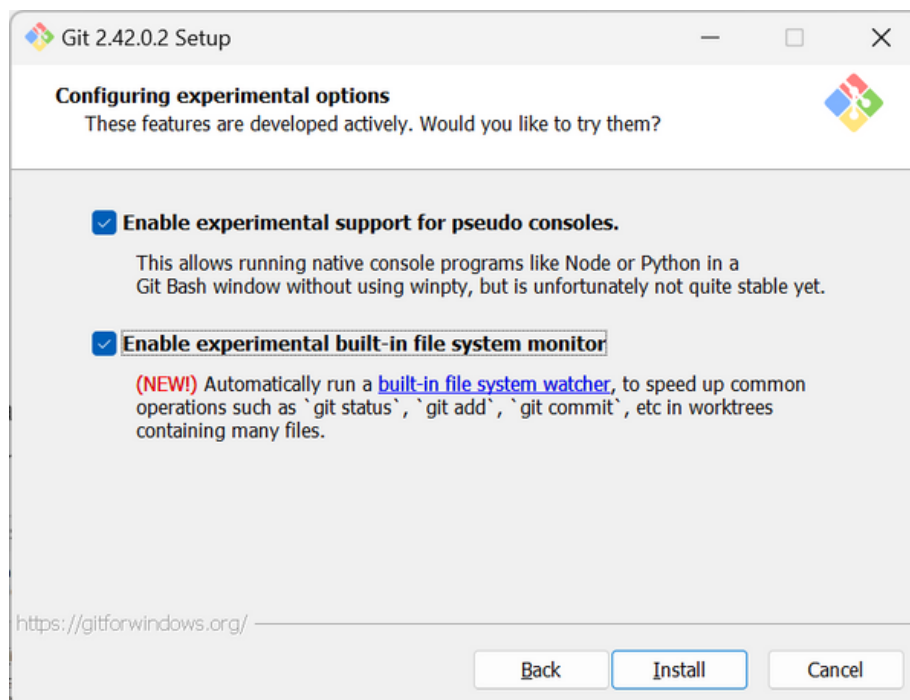
6. Pilih Visual Studio Code sebagai *default* untuk *text editor* pada Git.



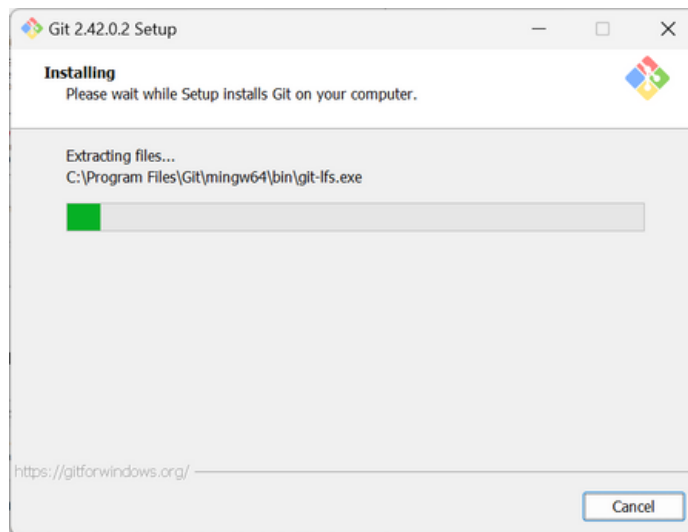
7. Click option next pada pilihan yang tersisa dan biarkan semuanya pada pengaturan default sampai di step selanjutnya.



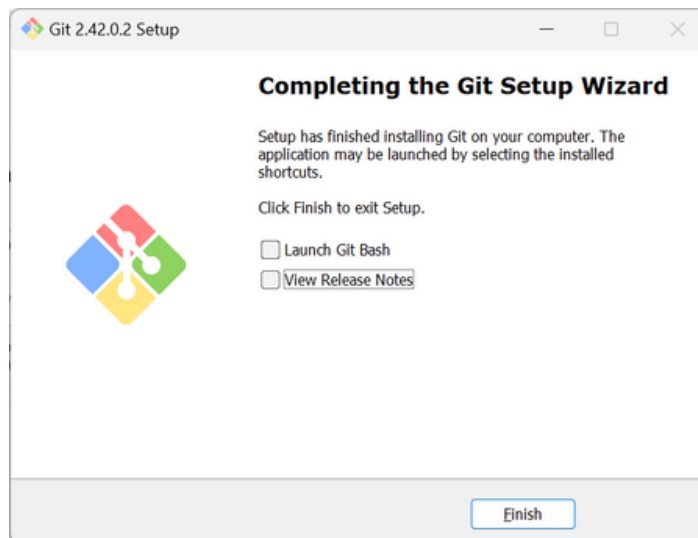
8. Pada bagian "Configuring experimental options", pilih kedua opsi berikut.



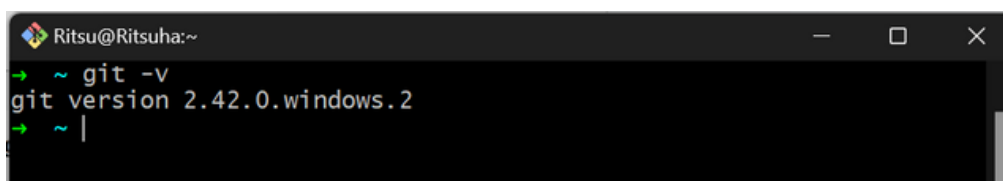
9. Tunggu proses instalasi.



10. Proses instalasi selesai.



11. Buka terminal untuk melakukan *check* apabila Git sudah terinstall dengan *command* :
Git -v atau Git -version.

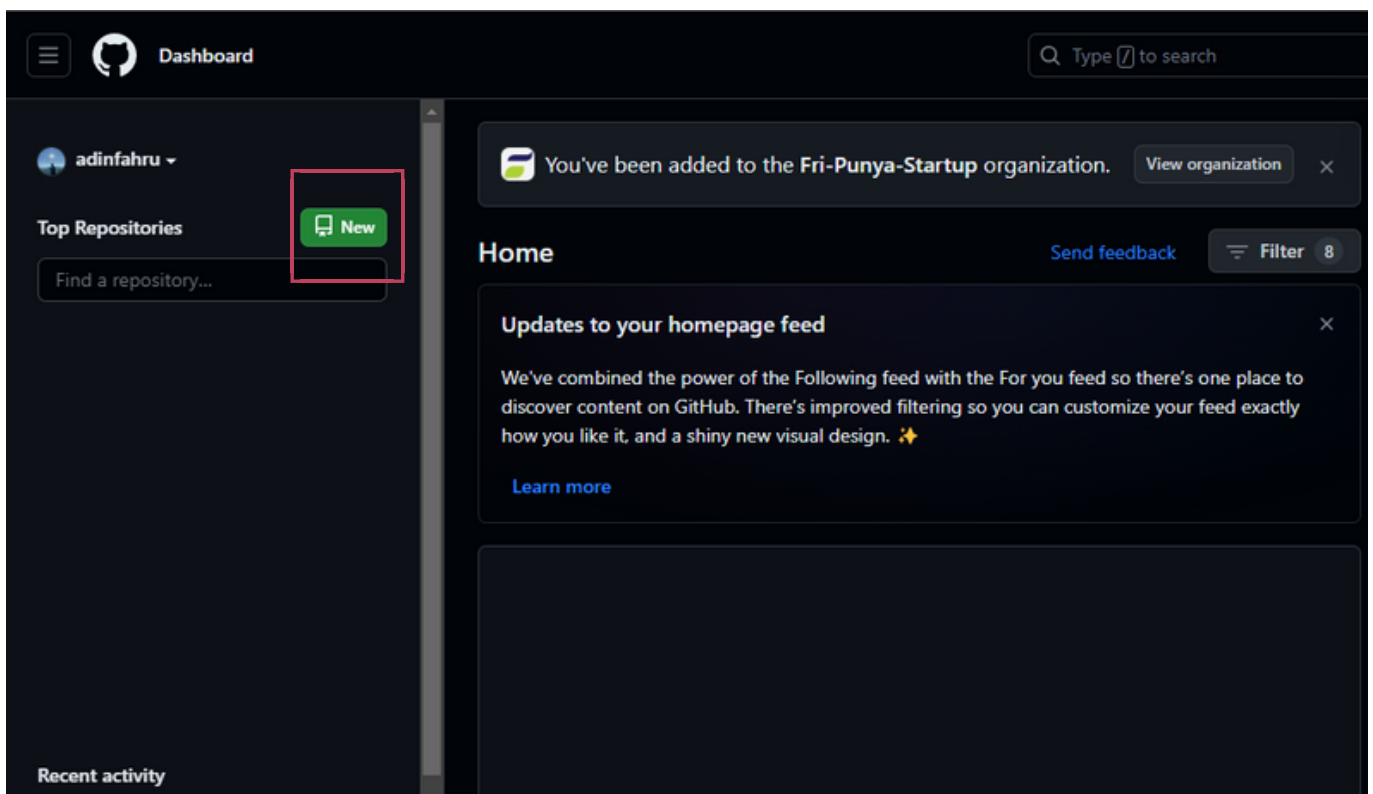


MEMBUAT REPOSITORY GITHUB

Selama praktikum OOP ini, semua tugas dan kode yang praktikan buat akan diunggah dan disimpan di *repository* GitHub yang khusus dibuat. *Repository* ini akan berfungsi sebagai wadah pusat untuk semua kegiatan praktikum, memungkinkan praktikan untuk menyimpan catatan-catatan penting yang berkaitan dengan praktikum. Para praktikan diharapkan untuk mengundang GitHub asisten praktikan ke dalam repositori GitHub.

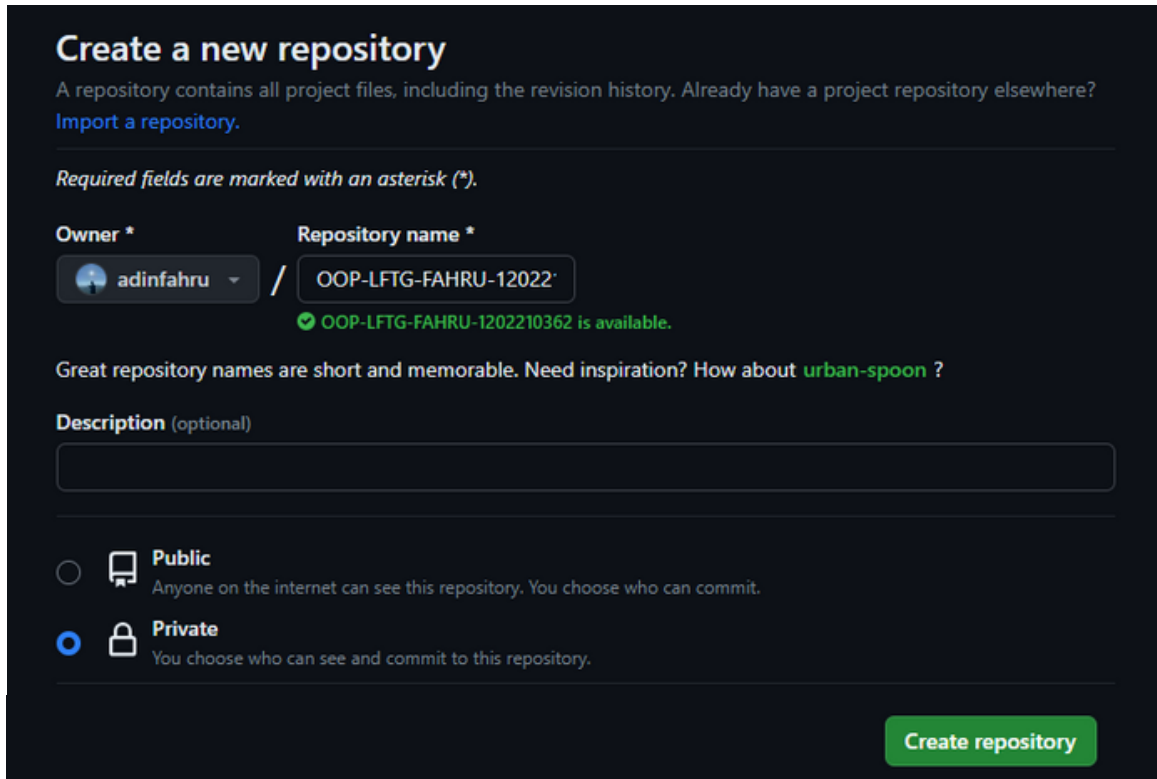
Berikut adalah langkah-langkah untuk membuat repository di GitHub :

1. Buka *website* <https://github.com/> dan lakukan *sign in*, jika belum memiliki akun bisa melakukan *sign up*.
2. Pilih menu “New” untuk membuat repository.



3. Tambahkan nama *repository* dengan format:

OOP-KODE ASISTEN-NAMA PENDEK PRAKTIKAN-NIM dan pilih *option private*.



Create a new repository

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

Required fields are marked with an asterisk (*).

Owner * **Repository name ***

adinfahru / OOP-LFTG-FAHRU-12022

✓ OOP-LFTG-FAHRU-1202210362 is available.

Great repository names are short and memorable. Need inspiration? How about [urban-spoon](#) ?

Description (optional)

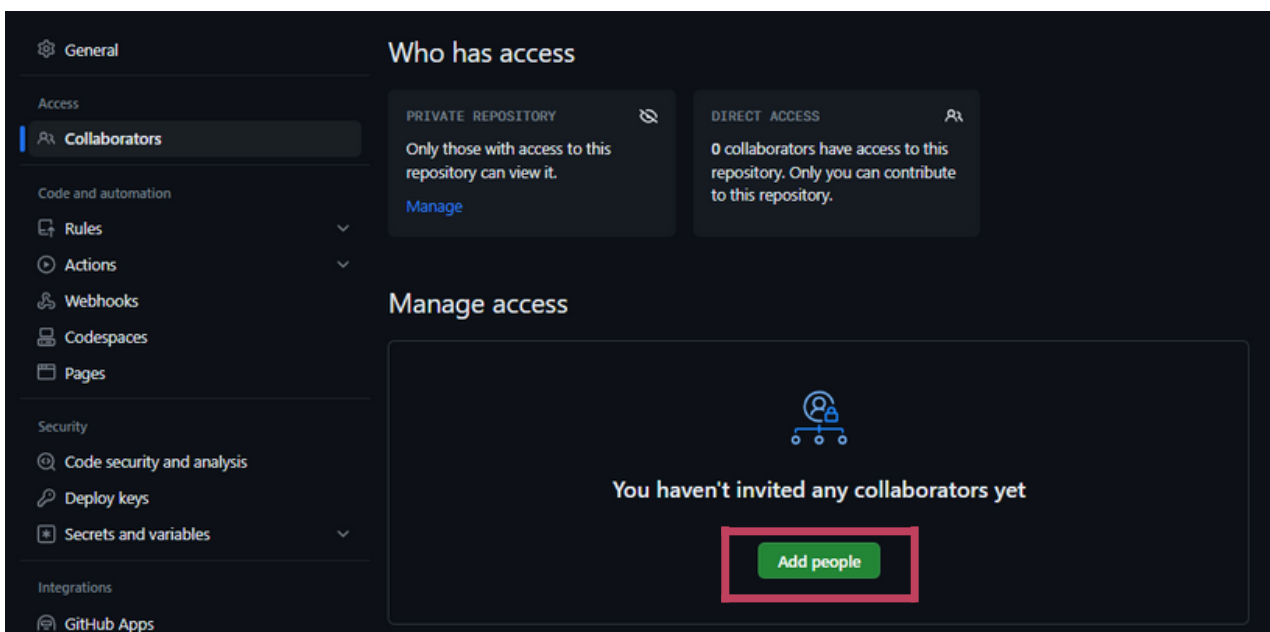
☐ **Public**
Anyone on the internet can see this repository. You choose who can commit.

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

[Create repository](#)

4. Pada menu *settings* dan submenu *collaborators*, tambahkan *collaborators* yaitu akun GitHub asisten praktikum.

Untuk github asisten praktikum bisa dilihat melalui : [Username github](#)



General

Access

Collaborators

Who has access

PRIVATE REPOSITORY

Only those with access to this repository can view it.

[Manage](#)

DIRECT ACCESS

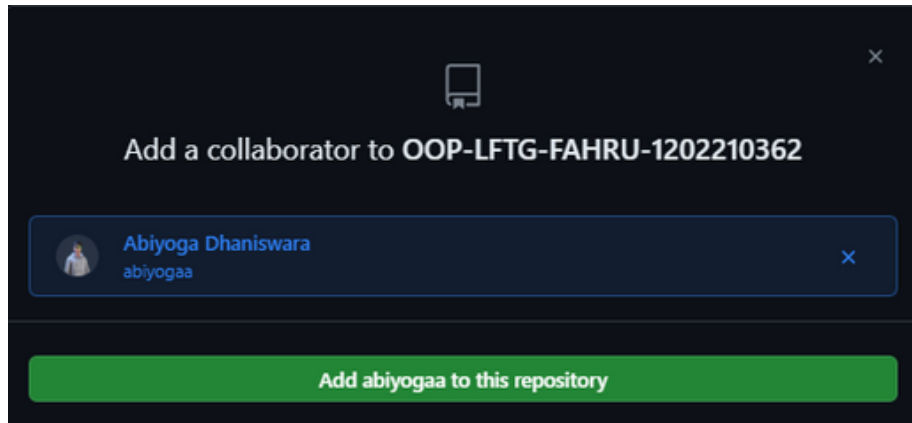
0 collaborators have access to this repository. Only you can contribute to this repository.

Manage access

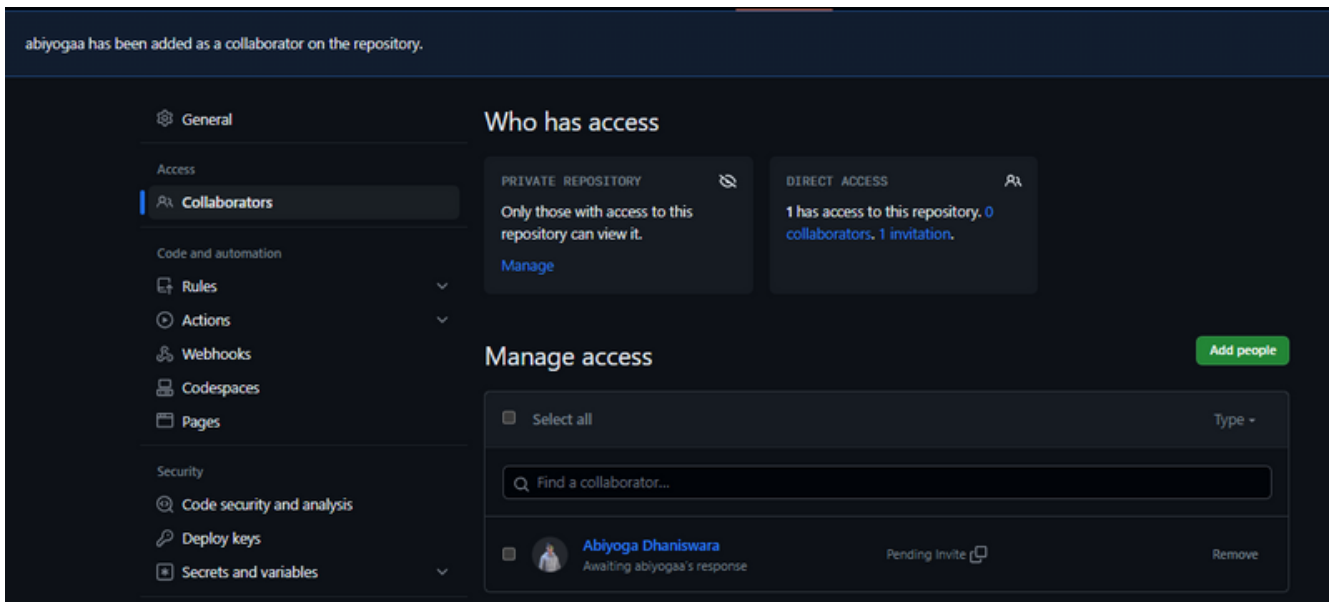
You haven't invited any collaborators yet

[Add people](#)

5. Masukkan *username* GitHub asisten praktikum.

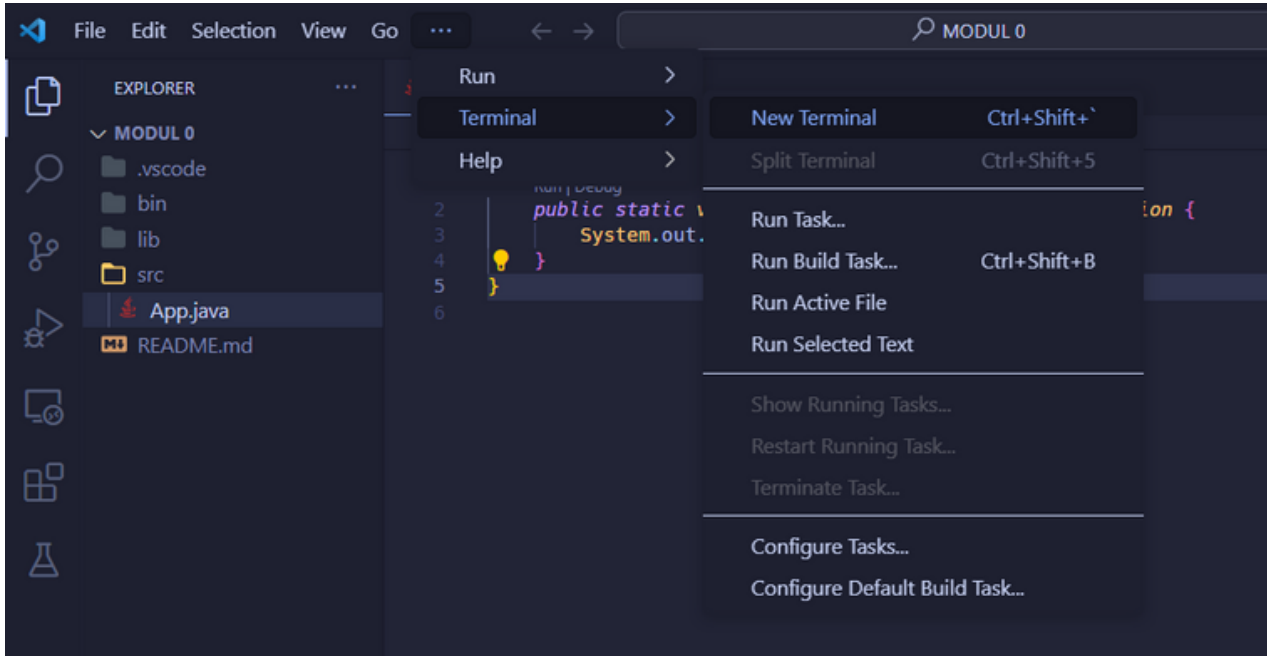


6. Asisten praktikum berhasil ditambahkan.

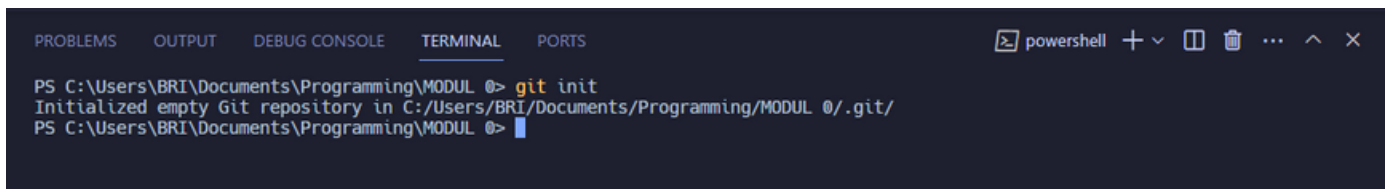


COMMIT FILE KE GITHUB

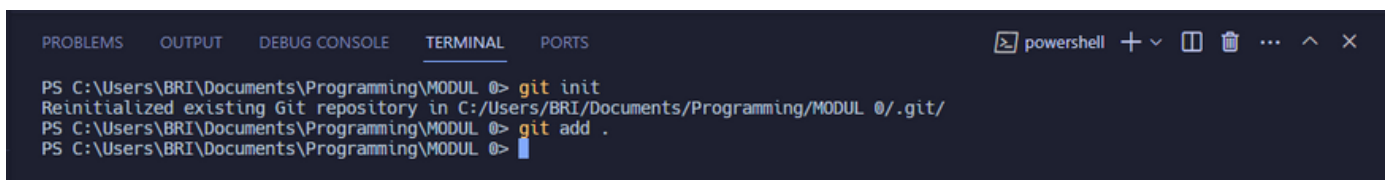
1. Buka *file* java, dan buka terminal pada *text editor*.



2. Lakukan *command* "*Git init*" untuk menginisialisasi (membuat) repository Git baru di dalam direktori atau folder tertentu.



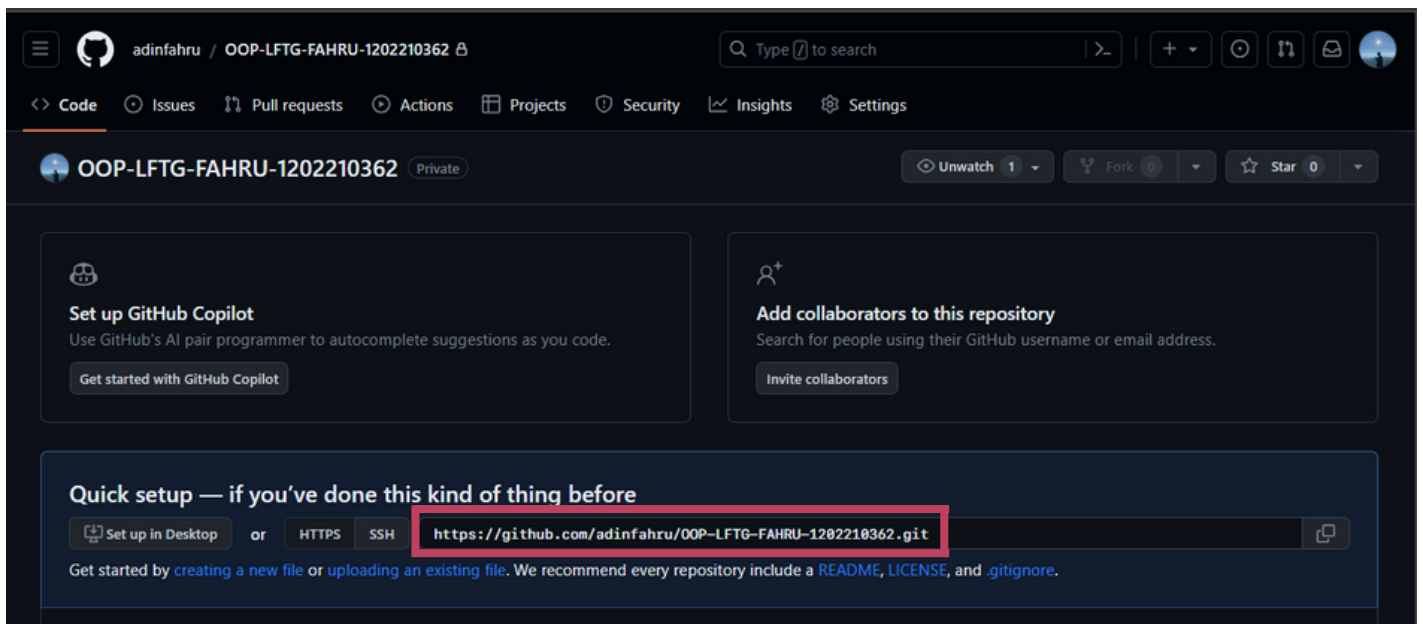
3. Lakukan *command* "*Git add .*" untuk menambahkan perubahan ke daftar perubahan (*staging area*).



4. Lakukan *command* `Git commit -m "(message)"` untuk membuat *commit* yang akan merekam perubahan-perubahan tersebut dalam *history repository*.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\BRI\Documents\Programming\MODUL 0> git init
Reinitialized existing Git repository in C:/Users/BRI/Documents/Programming/MODUL 0/.git/
PS C:\Users\BRI\Documents\Programming\MODUL 0> git add .
PS C:\Users\BRI\Documents\Programming\MODUL 0> git commit -m "modul 0 commit"
```

5. Lakukan *command* `"Git remote add origin (url repository)"` untuk menambahkan *repository remote* sebagai "origin" ke dalam *repository lokal* Anda.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Reinitialized existing Git repository in C:/Users/BRI/Documents/Programming/MODUL 0/.git/
PS C:\Users\BRI\Documents\Programming\MODUL 0> git add .
PS C:\Users\BRI\Documents\Programming\MODUL 0> git commit -m "modul 0 commit"
[master (root-commit) b52b939] modul 0 commit
4 files changed, 30 insertions(+)
create mode 100644 .vscode/settings.json
create mode 100644 README.md
create mode 100644 bin/App.class
create mode 100644 src/App.java
PS C:\Users\BRI\Documents\Programming\MODUL 0> git remote add origin https://github.com/adinfahru/OOP-LFTG-FAHRU-1202210362.git
```

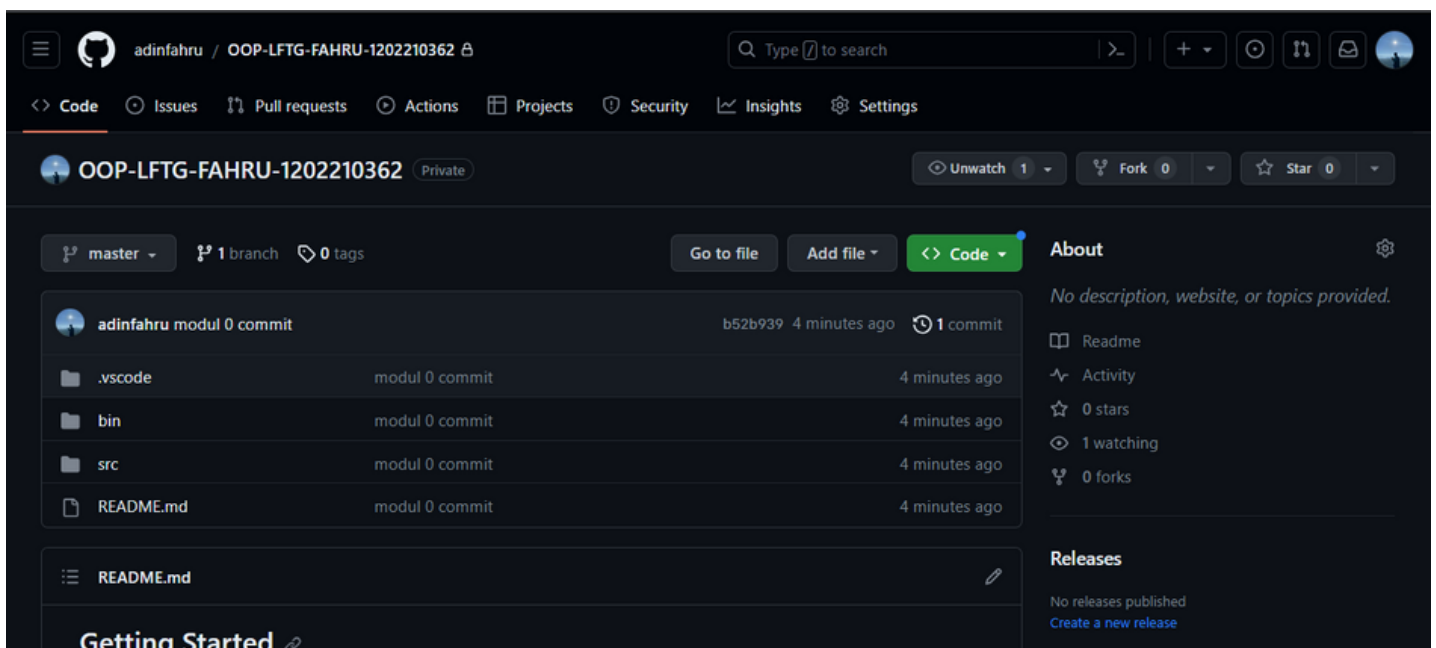
6. Lakukan *command* "Git push origin master" untuk melakukan *push* ke dalam *repository* *origin* untuk *branch* *master*.

```

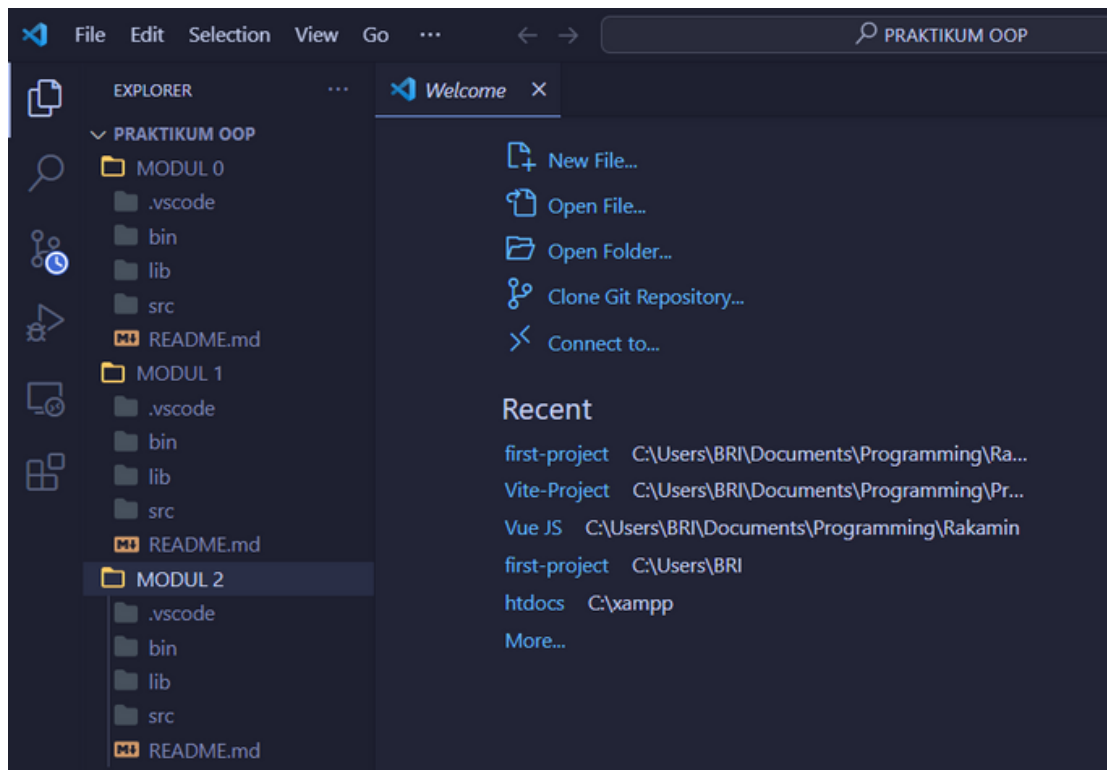
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\BRI\Documents\Programming\MODUL 0> git add .
PS C:\Users\BRI\Documents\Programming\MODUL 0> git commit -m "modul 0 commit"
[master (root-commit) b52b939] modul 0 commit
4 files changed, 30 insertions(+)
create mode 100644 .vscode/settings.json
create mode 100644 README.md
create mode 100644 bin/App.class
create mode 100644 src/App.java
PS C:\Users\BRI\Documents\Programming\MODUL 0> git remote add origin https://github.com/adinfahru/OOP-LFTG-FAHRU-1202210362.git
PS C:\Users\BRI\Documents\Programming\MODUL 0> git push origin master

```

7. File berhasil di *push* ke *repository*.



Note: Folder pengerjaan semua modul terdapat di dalam 1 Folder Praktikum OOP dan *file* java setiap modul terdapat pada folder yang berbeda.



**"If You're Not A Good Today,
Don't Worry. There Are
Other Days To Be Useful."**

find us on :

 @eadlaboratory

 @XXXXXXXXXXXXXX