

ACARA 1 : INTRODUCTION DAN WORKFLOW PADA GIT, FLUTTER & DART

Pokok Bahasan	: Intro dan Pengenalan Git
Acara Praktikum/Pertemuan	: Minggu 1 / 1
Tempat	: Daring / Luring Politeknik Negeri Jember
Alokasi Waktu	: 100 menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami konsep git
2. Mahasiswa mampu menerapkan git

b. Indikator

Kemampuan mahasiswa dalam memahami konsep dasar git

c. Dasar Teori

Pengenalan Git

Instalasi Git di Windows memang tidak seperti di Linux yang ketik perintah langsung terinstal. Kita harus men-download dulu, kemudian melakukan ritual *next* → *next* → *finish*. Tapi dalam ritual tersebut, ada pilihan yang harus diperhatikan agar perintah git dapat dikenali di CMD.

Terdapat tiga kondisi file dalam Git.

1) Modified

Modified adalah kondisi dimana revisi atau perubahan sudah dilakukan, tetapi belum ditandai dan belum disimpan di *version control*. Contohnya pada gambar di atas, ada tiga file HTML yang dalam kondisi *modified*.

2) Staged

Staged adalah kondisi dimana revisi sudah ditandai, tetapi belum disimpan di *version control*. Untuk mengubah kondisi file dari *modified* ke *staged* gunakan perintah `git add nama_file`. Contoh:

```
git add index.html
```

3) Committed

Committed adalah kondisi dimana revisi sudah disimpan di version control. perintah untuk mengubah kondisi file dari *staged* ke *committed* adalah git commit.

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Pembagian Kelompok Tugas Project

Mahasiswa memperhatikan penjelasan dosen pengampu mata kuliah Workshop Aplikasi Mobile berbasis Android.

1. Mahasiswa menentukan kelompok dengan 4-5 anggota.
2. Membuat resume secara individu tentang konsep dasar dari pengenalan dan workflow git pada kertas folio.
3. Diskusikan dengan kelompok hasil resume yang telah dikerjakan secara individu, dan salin kedalam PPT.
4. Presentasikan hasil membuat resume di depan kelas setiap kelompok secara bergantian.

f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan (definisi, konsep dasar dan pengenalan git).
2. Dokumentasi tugas berupa file ppt.

g. Kesimpulan

Mahasiswa memahami mengenai definisi git, konsep dasar pengenalan dan workflow git.

h. Rubrik Penilaian

No	Indikator	Skor*			
1	Ketepatan waktu dan ketepatan dalam menjelaskan dari tugas ditunjang dengan bukti referensi	1	2	3	④
2	Ketepatan waktu dan ketepatan dalam menjelaskan dari tugas	1	2	③	4
3	Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas	1	②	3	4
4	Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas	①	2	3	4
Jumlah skor					

ACARA 2 : PENGENALAN DAN WORKFLOW PADA GIT, FLUTTER & DART

Pokok Bahasan	: Instalasi dan Workflow pada Git
Acara Praktikum/Pertemuan	: Minggu 1 / 2
Tempat	: Daring / Luring Politeknik Negeri Jember
Alokasi Waktu	: 100 menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu instalasi Git pada masing-masing PC
2. Mahasiswa mampu melakukan konfigurasi

b. Indikator

Keberhasilan mahasiswa dalam melakukan tahap-tahap instalasi git, sampai dengan mampu melakukan konfigurasi awal untuk akun git setiap individu.

c. Dasar Teori

-

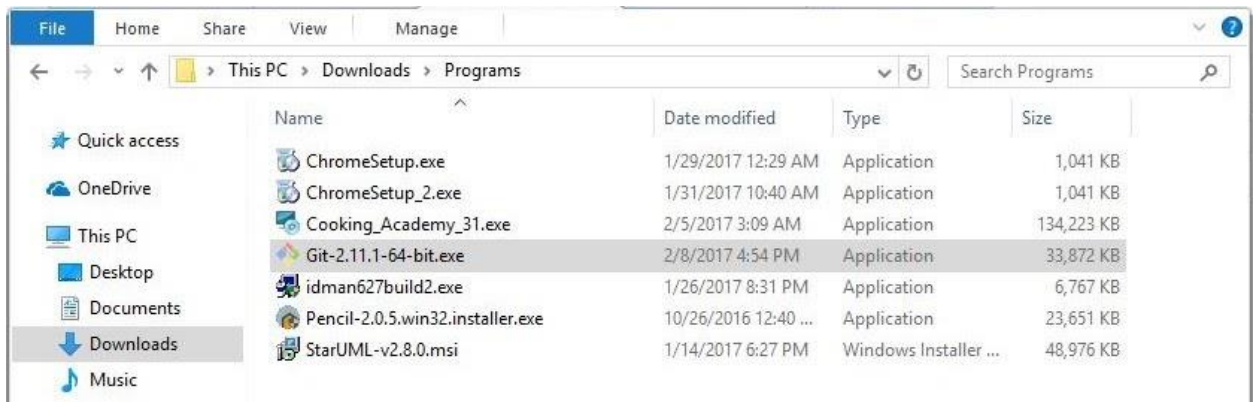
d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Instalasi Git

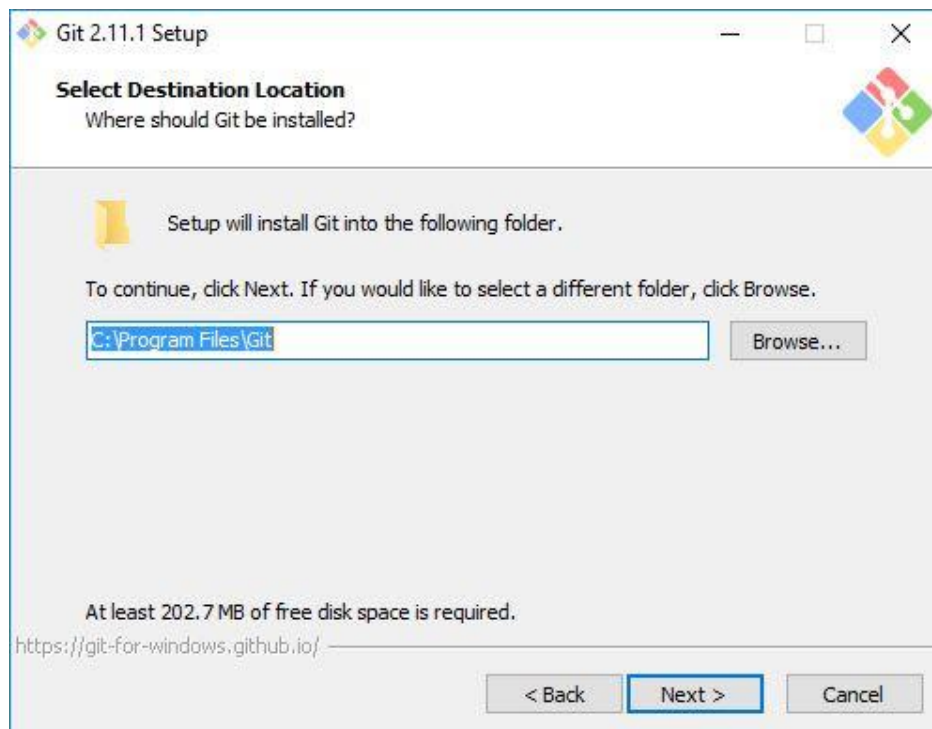
1. Download Git
Silahkan buka website resminya Git (git-scm.com).
2. Kemudian unduh Git sesuai dengan arsitektur komputer yang digunakan. Kalau menggunakan 64bit, unduh yang 64bit. Begitu juga kalau menggunakan 32bit.
3. Setelah file ter-unduh, silahkan klik 2x file instaler Git yang sudah diunduh.



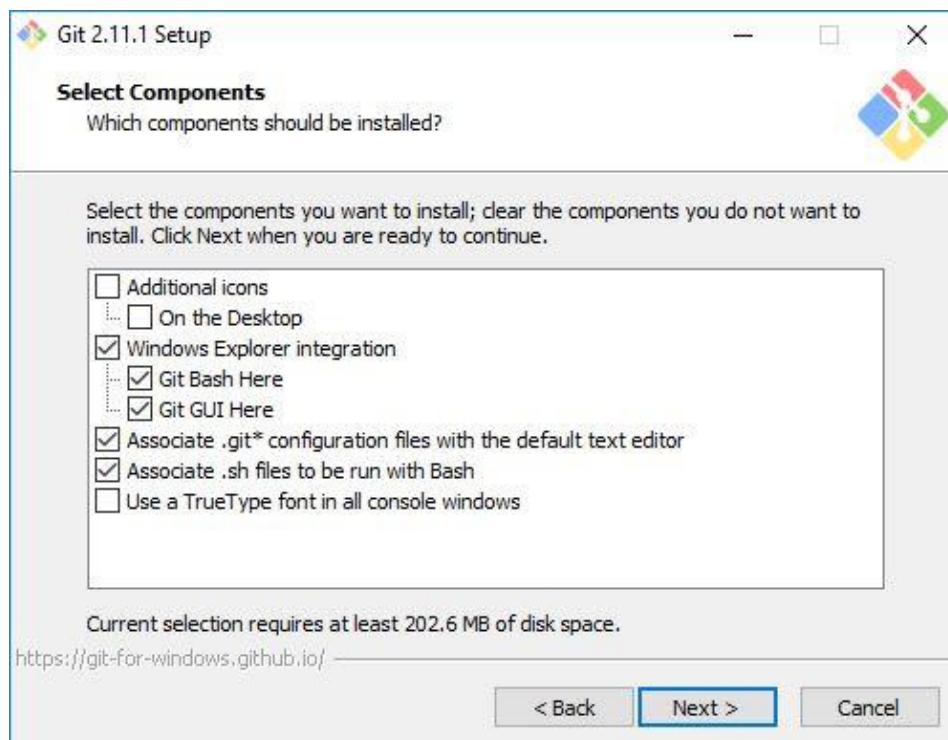
Maka akan muncul informasi lisensi Git, klik *Next >* untuk melanjutkan.



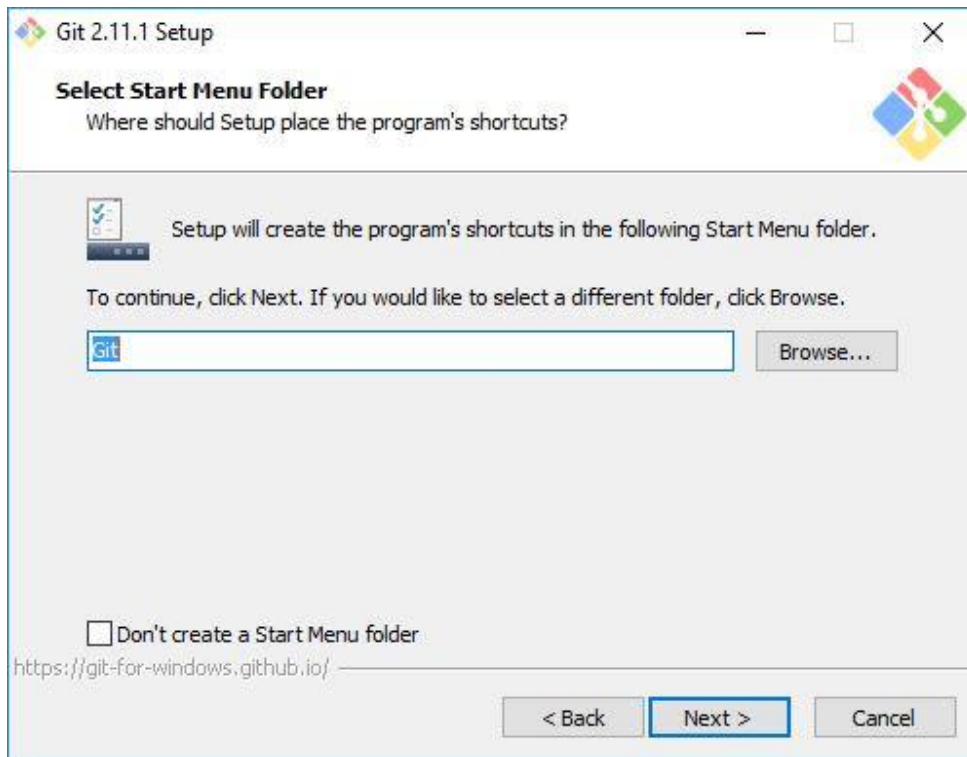
4. Selanjutnya menentukan lokasi instalasi. Biarkan saja apa adanya, kemudian klik *Next >*.



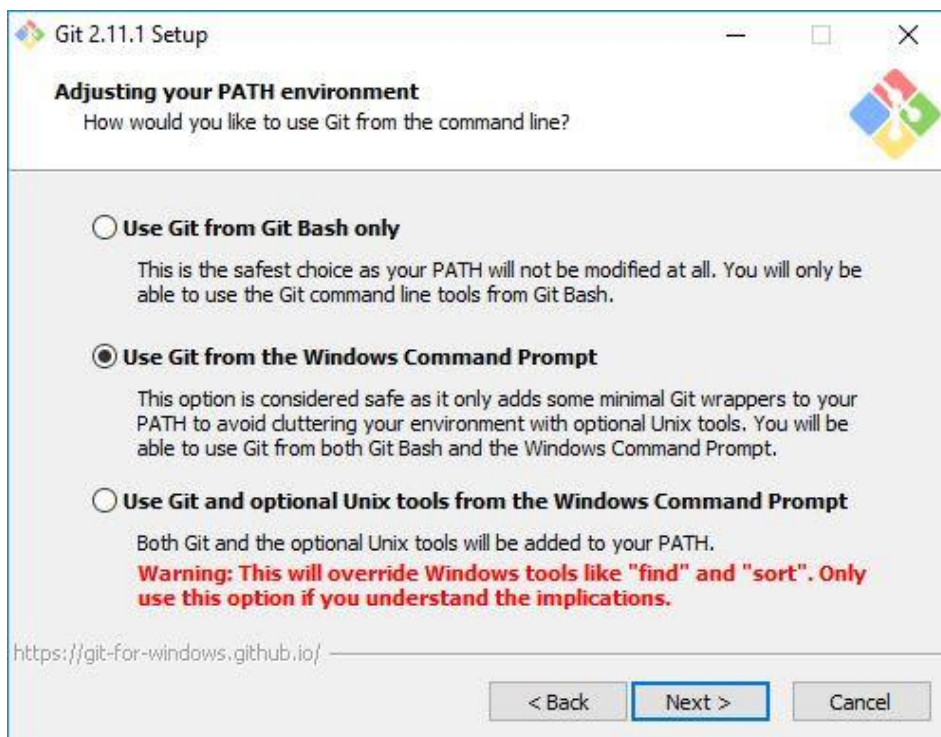
5. Selanjutnya pemilihan komponen, biarkan saja seperti ini kemudian klik *Next >*.



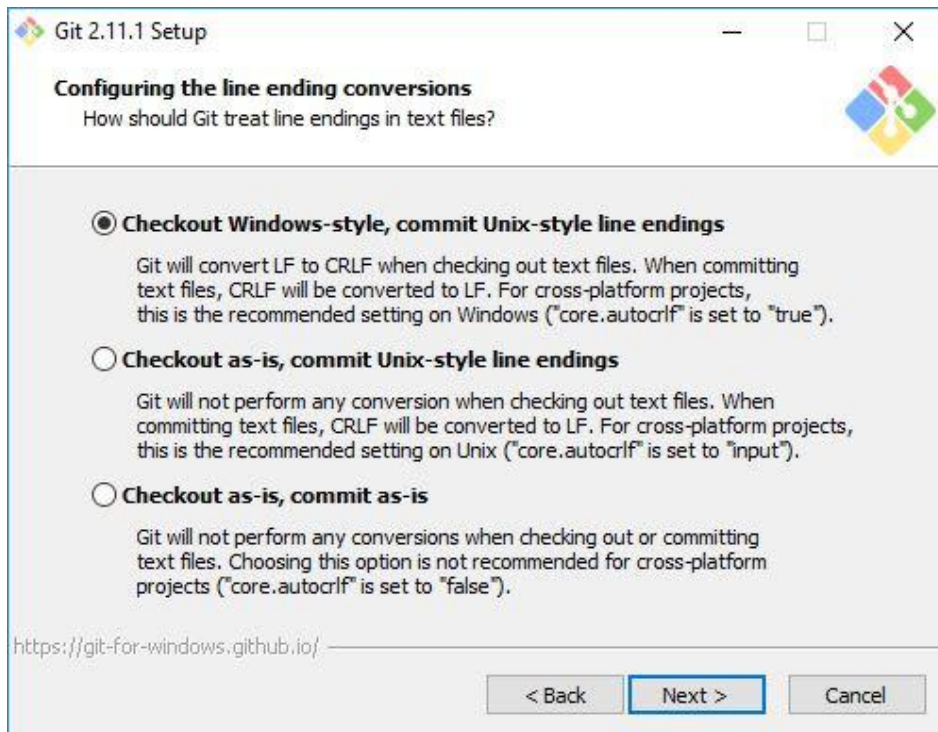
6. Selanjutnya pemilihan direktori start menu, klik *Next >*.



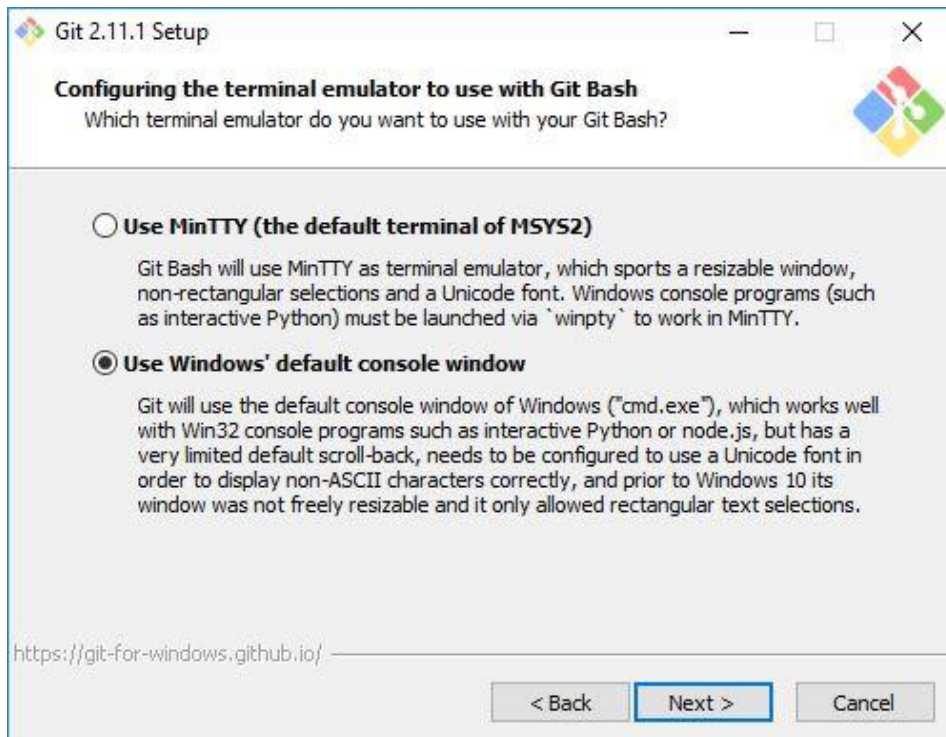
7. Selanjutnya pengaturan *PATH Environment*. Pilih yang tengah agar perintah git dapat dikenali di *Command Prompt* (CMD). Setelah itu klik *Next >*.



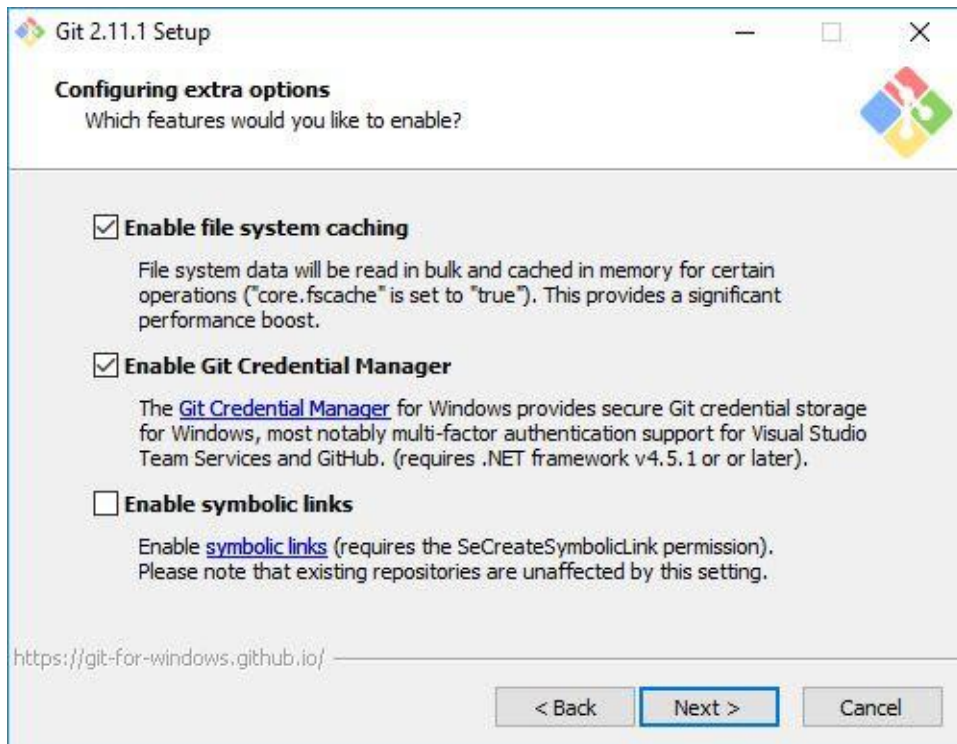
8. Selanjutnya konversi *line ending*. Biarkan saja seperti ini, kemudian klik *Next >*.



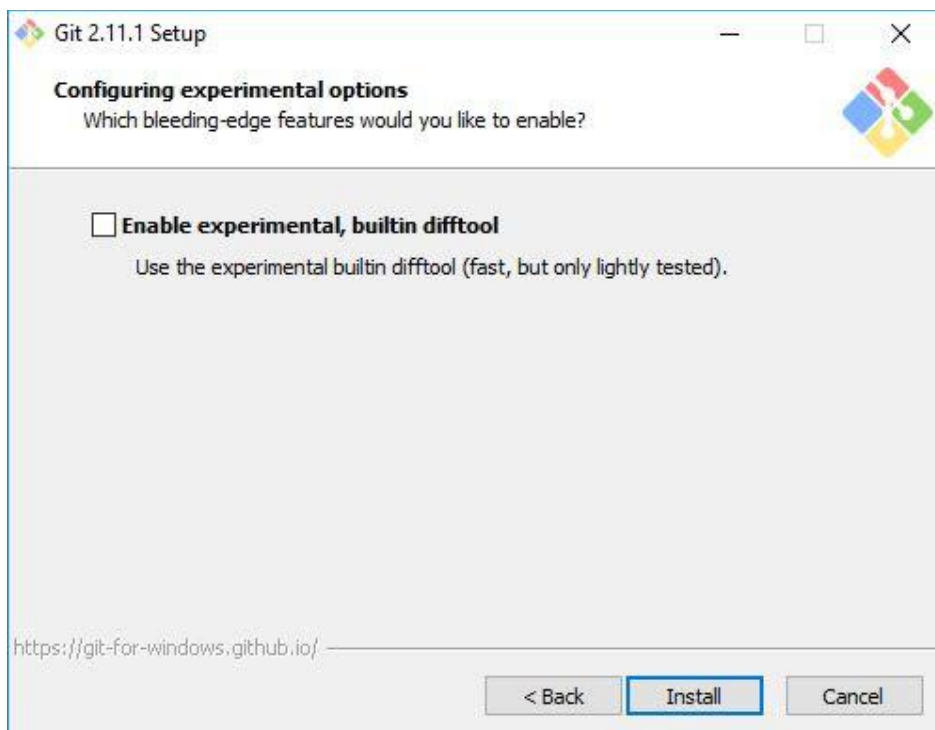
9. Selanjutnya pemilihan emulator terminal. Pilih saja yang bawah, lalu klik *Next >*.



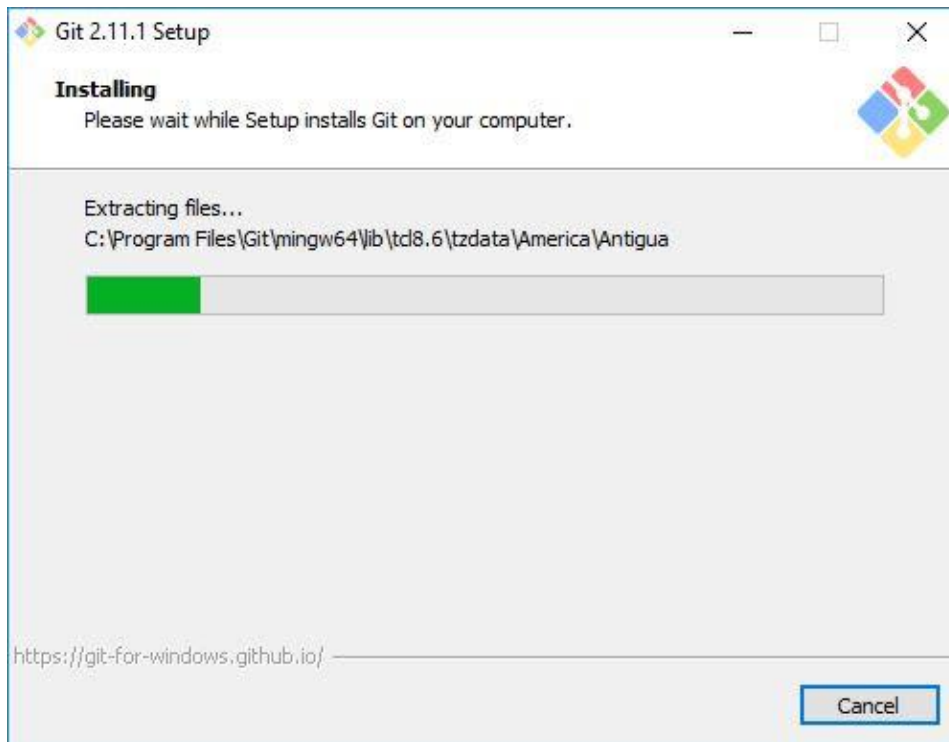
10. Selanjutnya pemilihan opsi ekstra. Klik saja *Next >*.



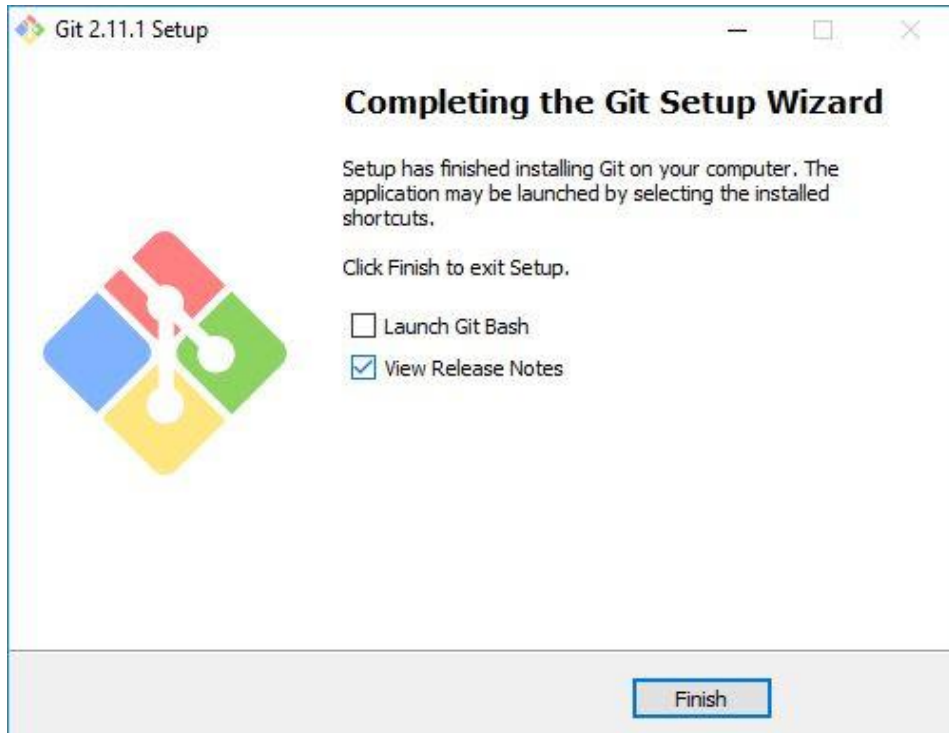
11. Selanjutnya pemilihan opsi ekspreimental, langsung saja klik *Install* untuk instalasi.



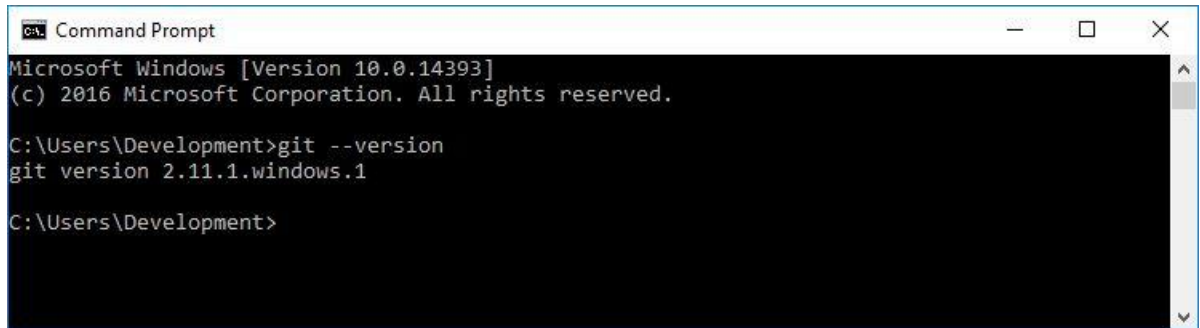
12. Tunggu beberapa saat, instalasi sedang dilakukan.



13. Setelah selesai, kita bisa langsung klik *Finish*.



14. Selamat, Git sudah terinstal di Windows. Untuk mencobanya, silahkan buka CMD atau PowerShell, kemudian ketik perintah `git --version`.



```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Development>git --version
git version 2.11.1.windows.1

C:\Users\Development>
```

Konfigurasi Awal

Ada beberapa konfigurasi yang harus dipersiapkan sebelum mulai menggunakan Git, seperti *name* dan *email*.

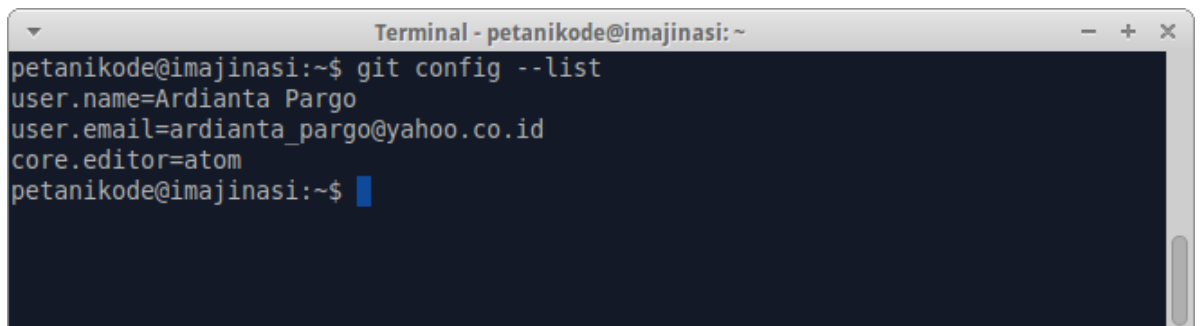
1. Silahkan lakukan konfigurasi dengan perintah berikut ini.

```
git config --global user.name "developer"
git config --global user.email developer@polije.ac.id
```

2. Kemudian periksa konfigurasinya dengan perintah:

```
git config --list
```

Apabila berhasil tampil seperti gambar berikut ini, berarti konfigurasi berhasil.



```
Terminal - petanikode@imajinasi: ~
petanikode@imajinasi:~$ git config --list
user.name=Ardianta Pargo
user.email=ardianta_pargo@yahoo.co.id
core.editor=atom
petanikode@imajinasi:~$
```

- Konfigurasi `core.editor` bersifat opsional, sedangkan *name* dan *email* wajib.

- *Username* dan *email* harus mengikuti akun Github, Gitlab, Bitbucket atau yang lainnya, agar mudah dalam mengintegrasikan.

Revisi Pertama

1. Ubah kondisi tiga file HTML tadi menjadi *staged* dengan perintah git add.

```
git add index.html
git add about.html
git add contact.html
```

atau bisa melakukannya seperti ini:

```
git add index.html about.html contact.html
```

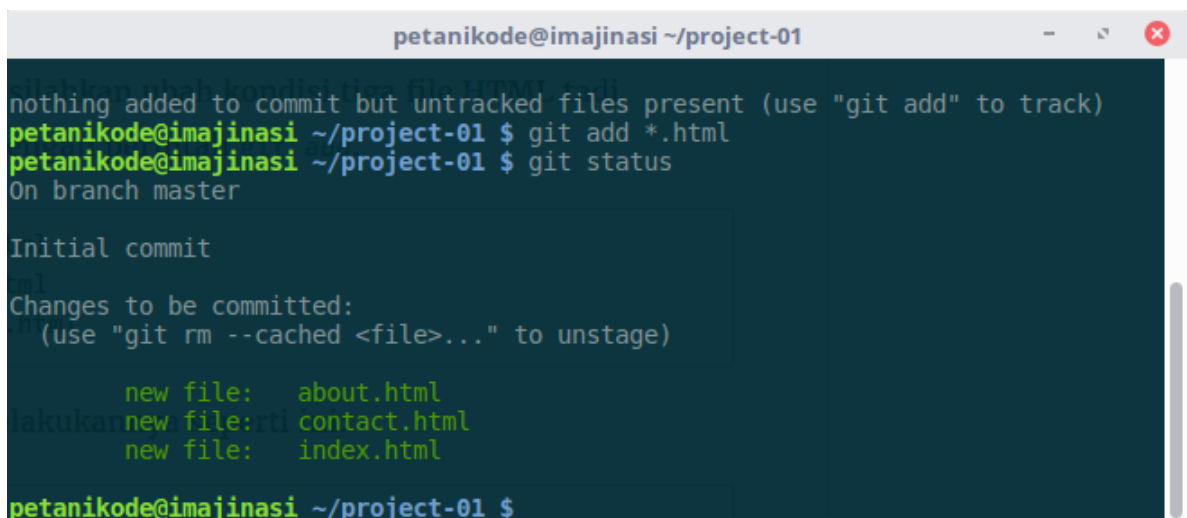
atau:

```
git add *.html
```

atau seperti ini (semua file dan direktori):

```
git add .
```

2. Selanjutnya, cobalah ketik perintah `git status` lagi. Kondisi filenya sekarang akan menjadi *staged*.



```
petanikode@imajinasi ~/project-01
nothing added to commit but untracked files present (use "git add" to track)
petanikode@imajinasi ~/project-01 $ git add *.html
petanikode@imajinasi ~/project-01 $ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)


        new file:   about.html
        new file:   contact.html
        new file:   index.html

petanikode@imajinasi ~/project-01 $
```

3. Lalu, ubah kondisi file tersebut ke *committed* agar semua perubahan disimpan oleh Git.

```
git commit -m "Commit pertama"
```

4. Setelah itu, coba cek dengan perintah git status lagi.



```
petanikode@imajinasi ~/project-01  
petanikode@imajinasi ~/project-01 $ git commit -m "commit pertama"  
[master (root-commit) cf08ca0] commit pertama  
3 files changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 about.html  
create mode 100644 contact.html  
create mode 100644 index.html  
petanikode@imajinasi ~/project-01 $ git status  
On branch master  
nothing to commit, working directory clean  
petanikode@imajinasi ~/project-01 $
```

5. Selamat, revisi pertama sudah kita buat. Selanjutnya cobalah untuk membuat revisi kedua.

Revisi Kedua

Ceritanya ada perubahan yang akan kita lakukan pada file index.html.

1. Silahkan modifikasi isi file index.html. Sebagai contoh diisikan seperti ini.

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>Belajar Git - Project 01</title>  
  </head>  
  <body>  
    <p>Hello Semua, Saya sedang belajar Git</p>  
  </body>  
</html>
```

2. Setelah itu ketik lagi perintah git status.

```
petanikode@imajinasi ~/project-01
nothing to commit, working directory clean
petanikode@imajinasi ~/project-01 $ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
petanikode@imajinasi ~/project-01 $
```

Terlihat di sana, file **index.html** sudah dimodifikasi. Kondisinya skarang berada dalam *modified*.

3. Lakukan *commit* lagi seperti revisi pertama.

```
git add index.html
git commit -m "ditambahkan isi"
```

- Dengan demikian, revisi kedua sudah disipan oleh Git.
- Argumen **-m** untuk menambahkan pesan setiap menyimpan revisi.

4. Sekarang Git sudah mencatat dua revisi yang sudah dilakukan.

Bisa diibaratkan revisi-revisi ini sebagai *checkpoint* pada Game. Apabila nanti ada kesalahan, kita bisa kembali ke *checkpoint* ini.



f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu tahap instalasi git yang di buktikan dengan screenshot pada masing-masing PC pada setiap prosesnya.
2. Dokumentasi hasil konfigurasi berupa akun git pribadi.

g. Kesimpulan

Mahasiswa berhasil instalasi Git pada masing-masing PC serta dilanjutkan melakukan konfigurasi dengan menggunakan akun individu serta melakukan revisi pertama dan kedua dengan berhasil.

h. Rubrik Penilaian

No	Indikator	Skor*			
1	Ketepatan waktu dan ketepatan dalam menjelaskan tahapan instalasi secara berurutan, serta konfigurasi	1	2	3	④
2	Ketepatan waktu dan ketepatan dalam menjelaskan tahapan instalasi, serta konfigurasi	1	2	③	4
3	Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas, serta konfigurasi	1	②	3	4
4	Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas, serta konfigurasi	①	2	3	4
Jumlah skor					

ACARA 3 : PENGENALAN DAN INSTALASI FLUTTER, FLUTTER & DART

Pokok Bahasan	: Instalasi Flutter
Acara Praktikum/Pertemuan	: Minggu 1 / 3
Tempat	: Daring / Luring Politeknik Negeri Jember
Alokasi Waktu	: 100 menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu melakukan instalasi flutter
2. Mahasiswa menjelaskan maksud setiap langkah & tahap instalasi flutter

b. Indikator

Keberhasilan mahasiswa dalam melakukan revisi pertama pada git, serta memahami setiap command git revisi pertama.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

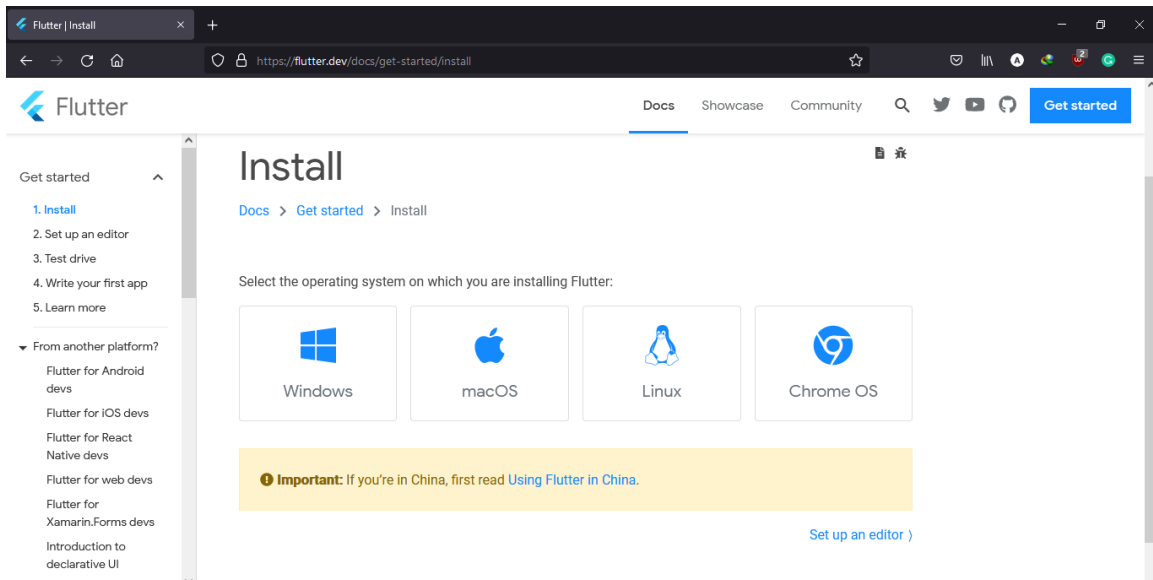
e. Prosedur Kerja

Instalasi flutter :

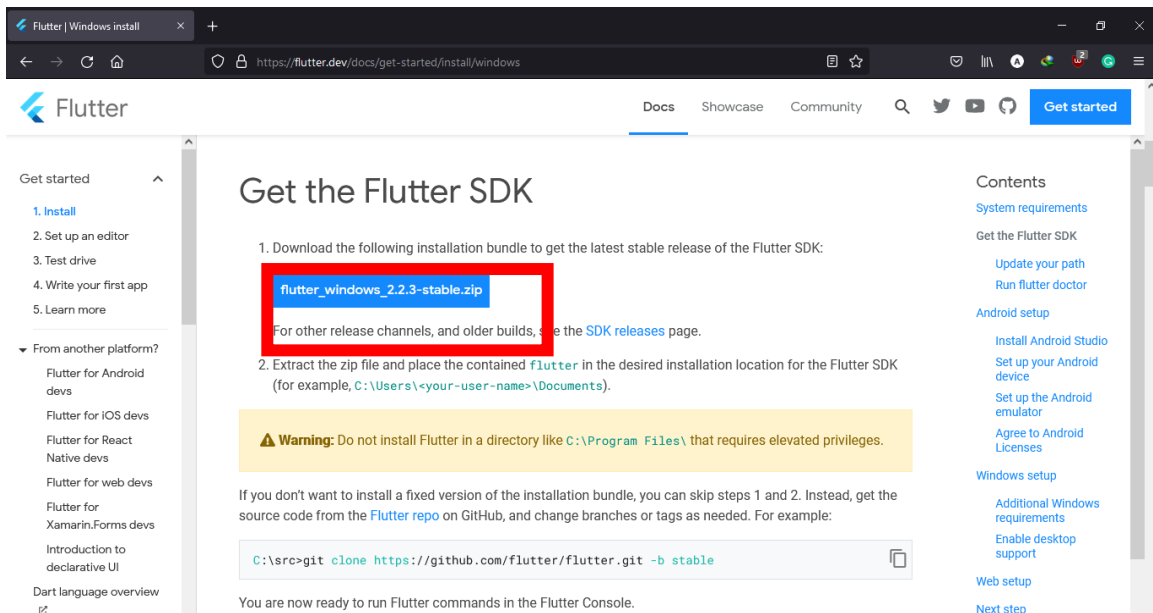
A. Tanpa Android Studio

- **SDK Flutter**

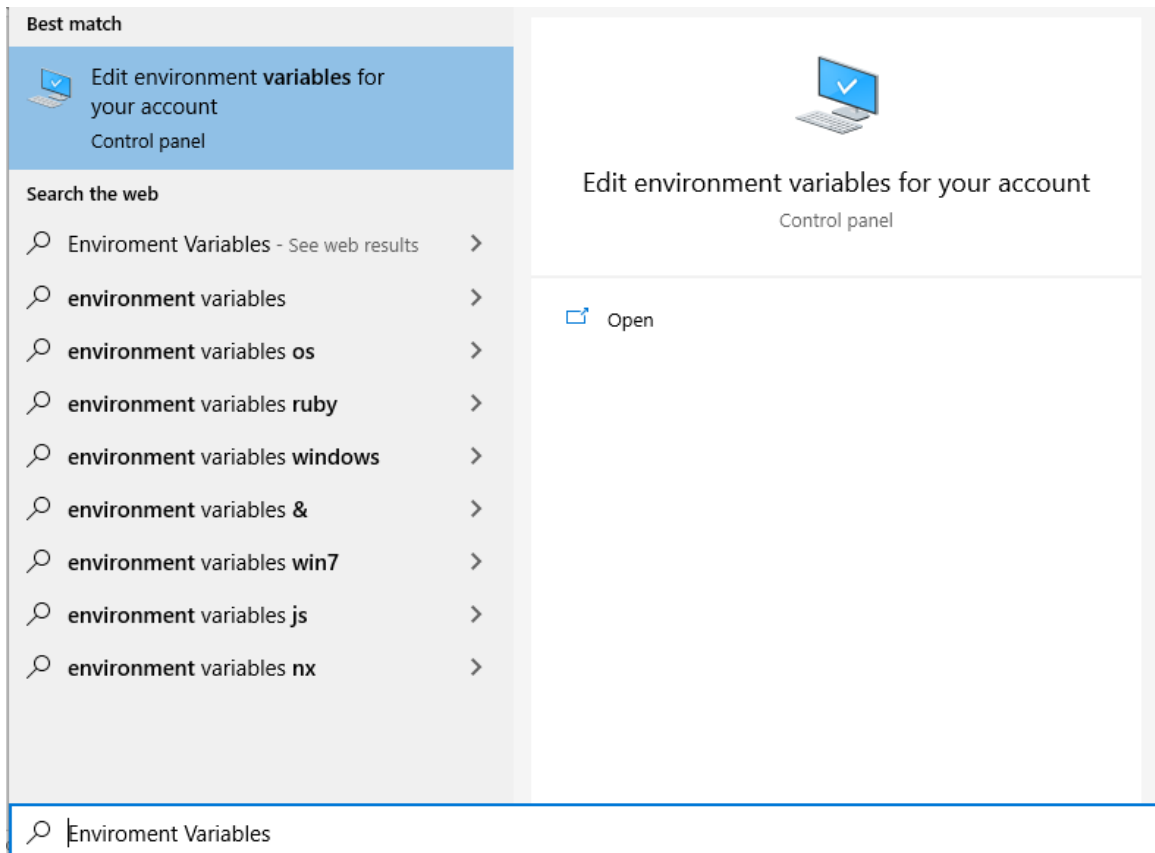
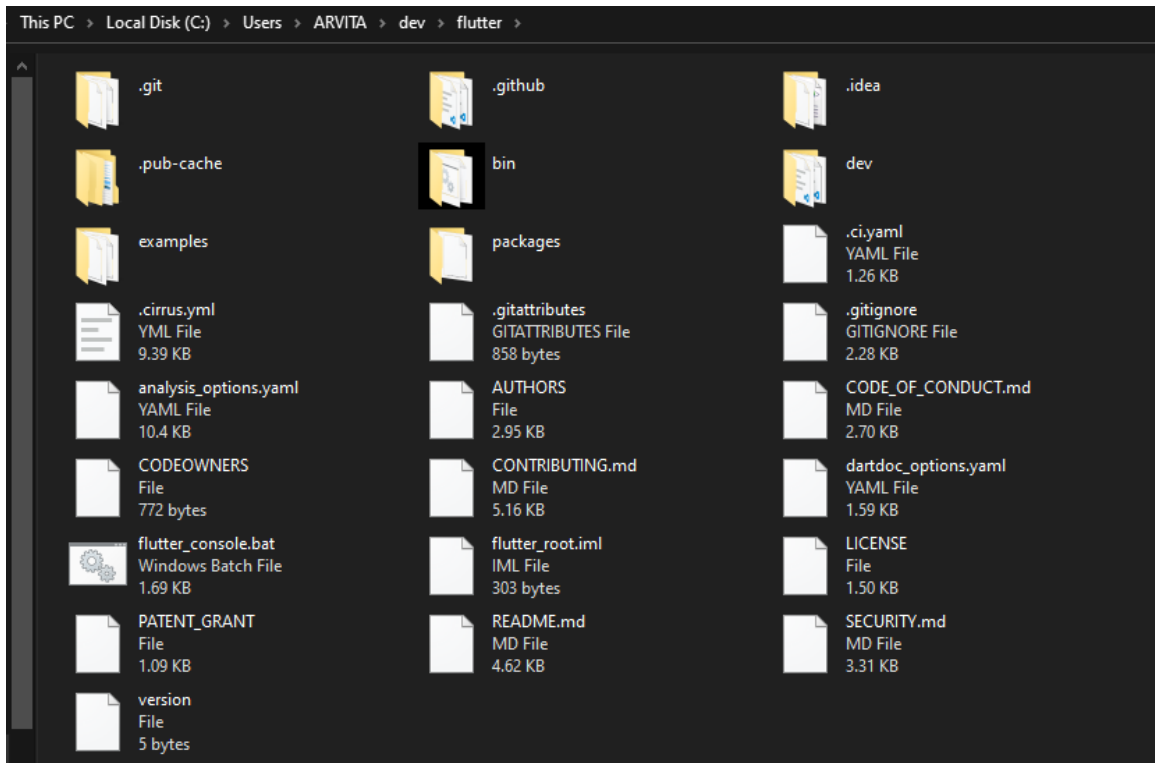
1. Download flutter di <https://flutter.dev/docs/get-started/install>



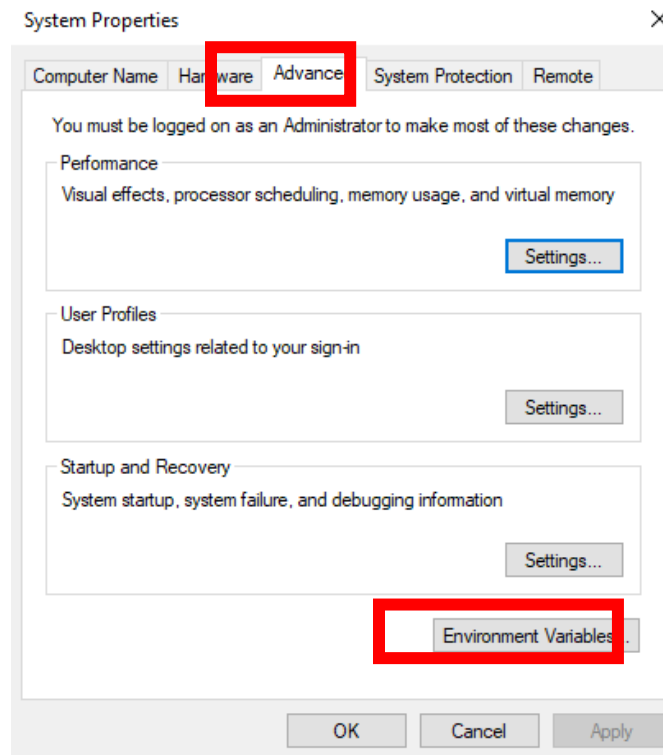
2. Pilih sesuai platform laptop, kemudian download.



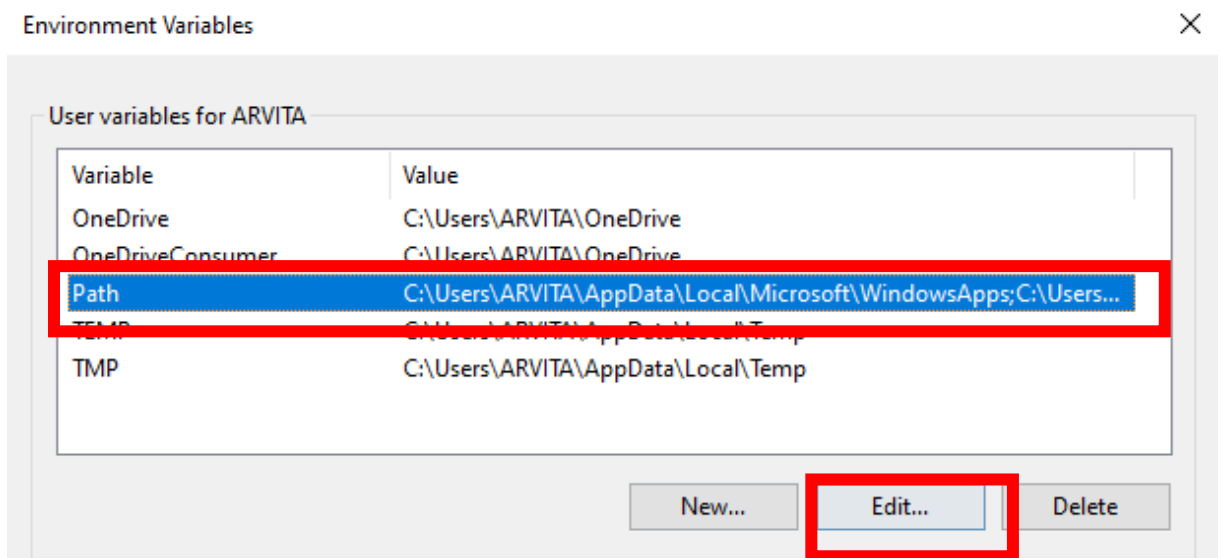
3. Pindahkan hasil download ke dalam folder yang sudah ditentukan, kemudian extract file hasil download.
4. Selanjutnya kita harus setting Environment Variable agar file exe di folder flutter\bin bisa dipanggil dari cmd. Ketik env pada search bar, lalu pilih Edit Environment Variables for Your Account



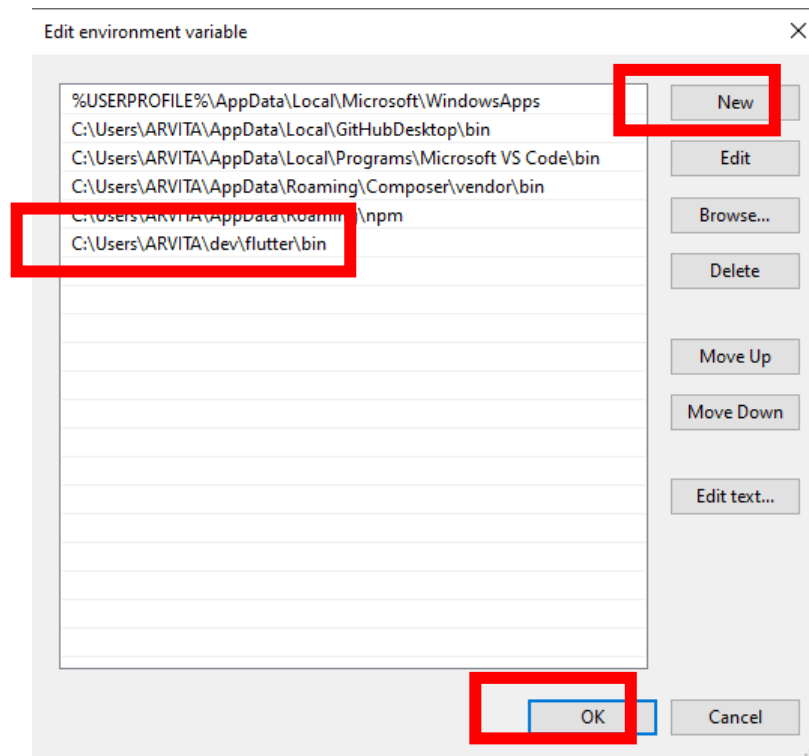
5. pada tab **Advanced**, pilih **Environment Variables**



6. Pada **user variables**, klik **Path** lalu **Edit**



7. Klik tombol **New**, tambahkan full path `C:\Users\ARVITA\dev\flutter\bin`, lalu klik **ok**



8. Klik **Ok** pada **Environment Variables**, lalu **close** dan **buka CMD**
9. Jalankan lagi perintah

```
flutter doctor
```

```
Welcome to Flutter! - https://flutter.dev

The Flutter tool uses Google Analytics to anonymously report feature usage
statistics and basic crash reports. This data is used to help improve
Flutter tools over time.

Flutter tool analytics are not sent on the very first run. To disable
reporting, type 'flutter config --no-analytics'. To display the current
setting, type 'flutter config'. If you opt out of analytics, an opt-out
event will be sent, and then no further information will be sent by the
Flutter tool.

By downloading the Flutter SDK, you agree to the Google Terms of Service.
Note: The Google Privacy Policy describes how data is handled in this
service.

Moreover, Flutter includes the Dart SDK, which may send usage metrics and
crash reports to Google.

Read about data we send with crash reports:
https://flutter.dev/docs/reference/crash-reporting

[✓] Flutter (Channel stable, 2.2.3, on Microsoft Windows [Version 10.0.19042.1165], locale en-US)
[✓] Android toolchain - develop for Android devices (Android SDK version 30.0.3)
[✓] Chrome - develop for the web
[✓] Android Studio (version 4.1.0)
[✓] VS Code (version 1.59.1)
[✓] Connected device (2 available)

• No issues found!
```

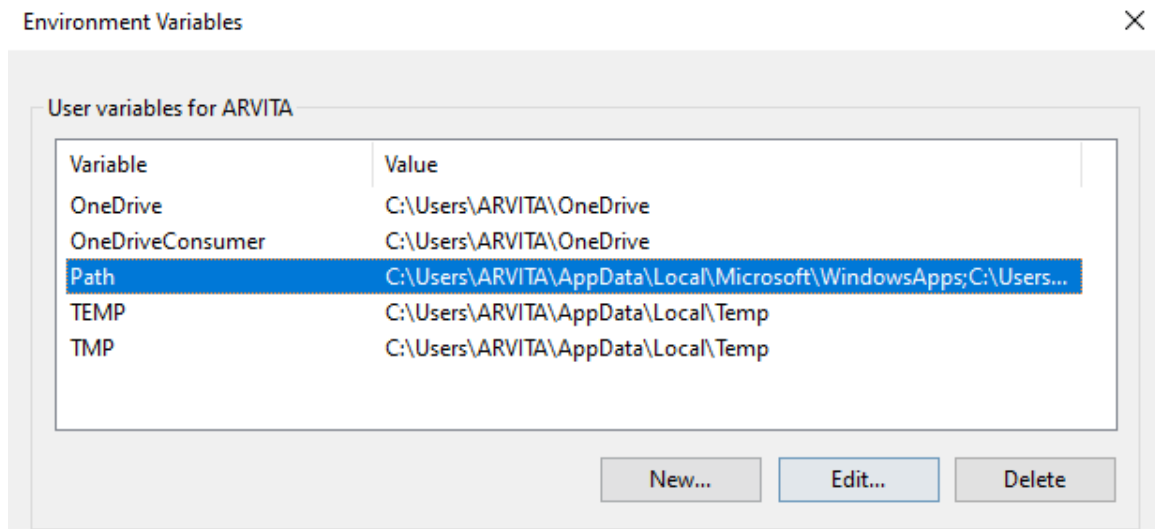
- JDK dan Android SDK

Jika belum menginstal Android JDK atau tidak menginstall Android studio maka lakukan langkah berikut :

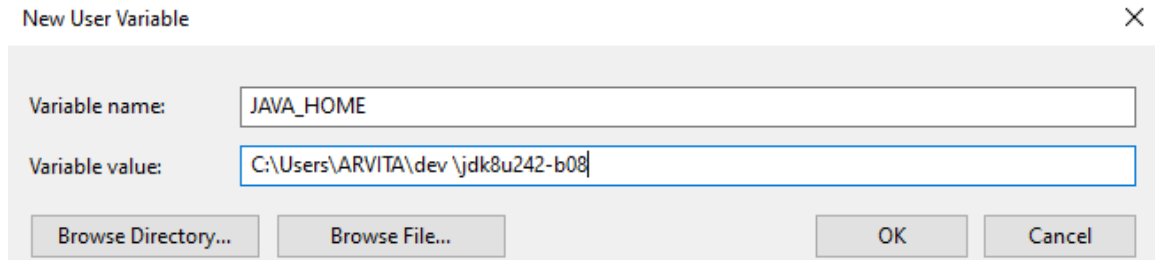
1. Download OPEN JDK di : <https://www.oracle.com/java/technologies/javase-jdk13-downloads.html>. Pilih yang [Windows 32](#) atau [Windows 64](#) bit
2. Ekstrak di folder tadi, sehingga struktur folder sekarang jadi seperti ini

androidSDK
flutter
jdk8u242-b08

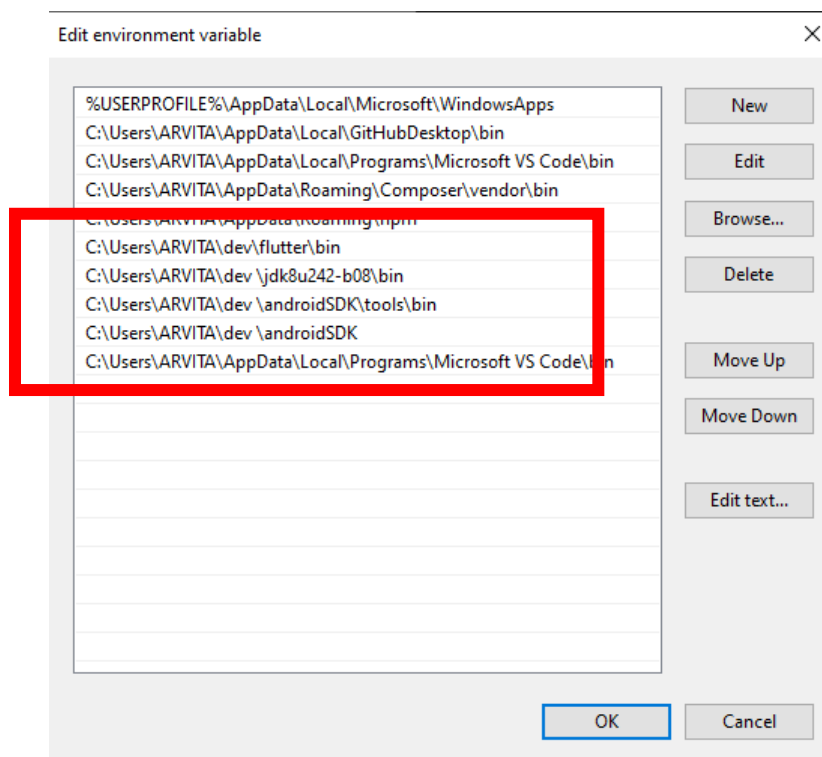
2. Tambahkan Environment Variable untuk JDK. Sesuaikan path JDK. Kalau saya di C:\Users\ARVITA\dev \jdk8u242-b08. Pada Environment Variable, klik New



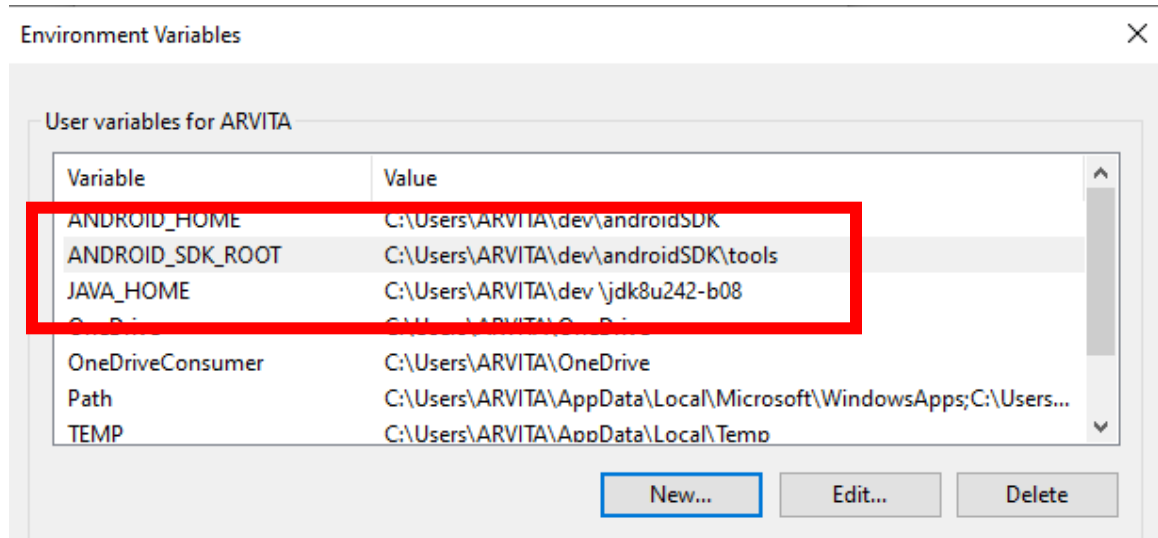
3. Tambahkan Environment Variable untuk JAVA_HOME, lalu tekan OK, OK juga pada environment variable



4. Silakan download Android SDK di <https://developer.android.com/studio/#command-tools>
5. Saya ekstrak di C:\Users\ARVITA\dev juga agar terkumpul dengan flutter SDK. Nama folder SDK nya silahkan sesuaikan, saya jadikan androidSDK
6. Tambahkan PATH environment variable AndroidSDK\tools\bin



7. Pastikan pada Environment Variable seperti dibawah ini



8. Untuk test konfigurasi, ketik perintah ini pada CMD:

```
sdkmanager
```

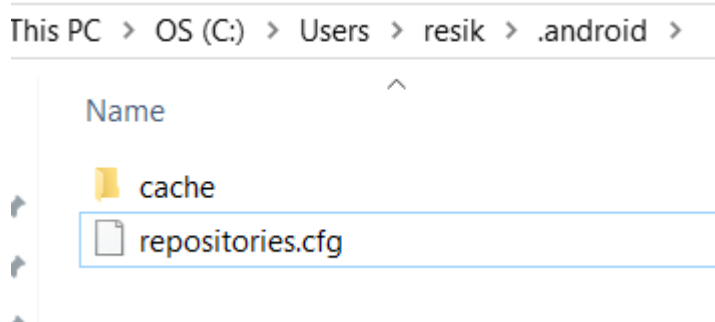
9. Jika error seperti itu (repositories.cfg could not be loaded), silakan buat file repositories.cfg folder yang diminta. Contohnya seperti berikut:

```

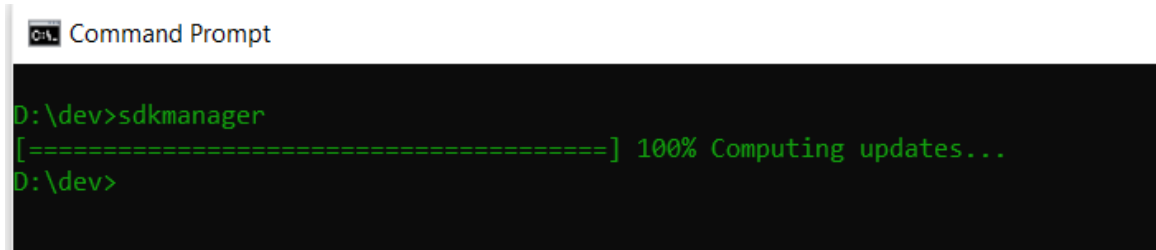
C:\> Command Prompt

D:\dev> sdkmanager
Warning: File C:\Users\resik\.android\repositories.cfg could not be loaded.
[=====] 100% Computing updates...
D:\dev>

```



10. Setelah itu ketik ulang sdk manager pada CMD, maka akan muncul seperti ini :



```
Command Prompt

D:\dev>sdkmanager
[=====] 100% Computing updates...
D:\dev>
```

11. Tentukan versi Android. Pada CMD ketikkan perintah

```
sdkmanager "build-tools;29.0.3" "platform-tools" "platforms;android-29" "system-images;android-29;default;x86_64"
```

12. Lanjutkan instalasi SDK, tinggal Accept-accept.

```
sdkmanager --update
```

13. Sebelum mulai menggunakan Flutter, kamu harus menyetujui lisensi dari android. Jalankan ini di CMD

```
flutter doctor --android-licenses
```

14. Silakan accept sampai ada tulisan **All SDK package licenses accepted.**

15. Kamu bisa Emulator Android untuk menjalankan aplikasi Androidmu. Sebagai alternatif, bisa gunakan HP atau emulator lain seperti GenyMotion, Bluestacks dan sebagainya.

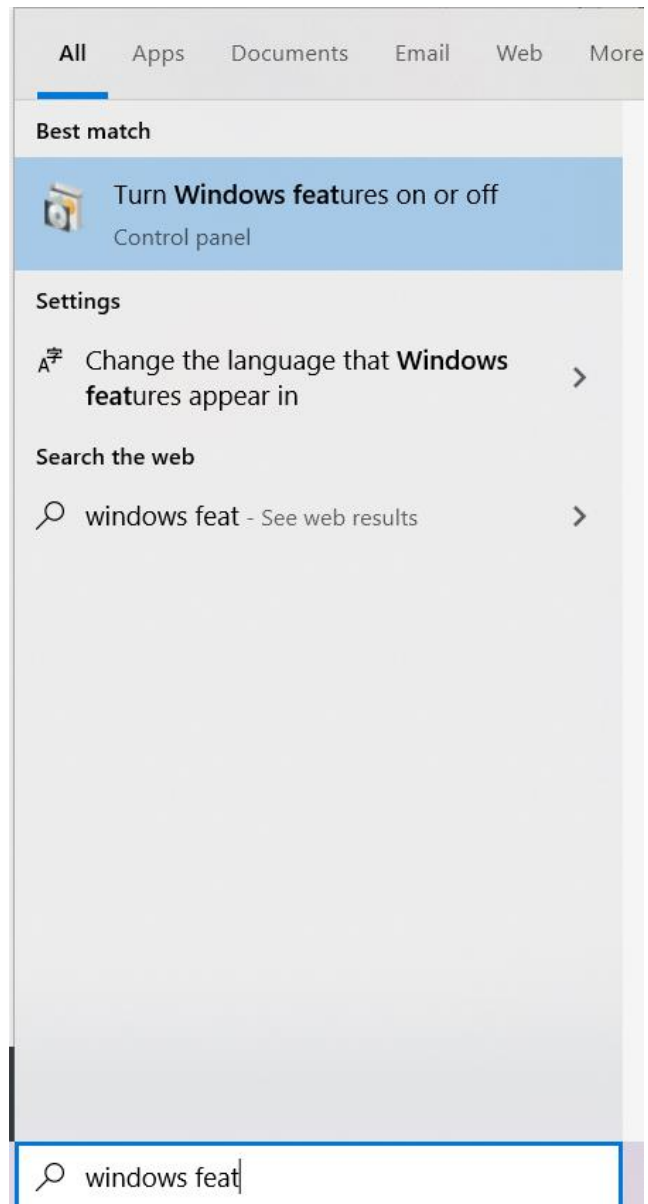
16. silakan install Android Emulator melalui SDK Manager. ketik

```
sdkmanager "emulator"
```

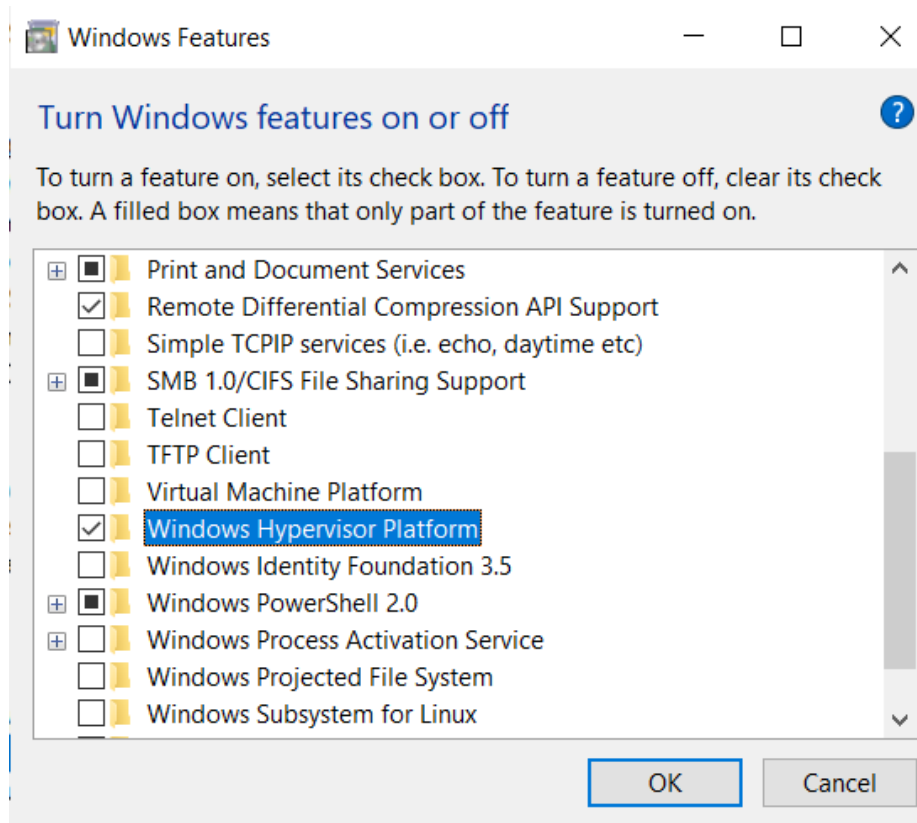
17. Selanjutnya create AVD

```
avdmanager create avd - name Nexus6P - tag 11 - package "system-images;android-29;default;x86_64"
```

18. Aktifkan Windows Hypervisor Platform

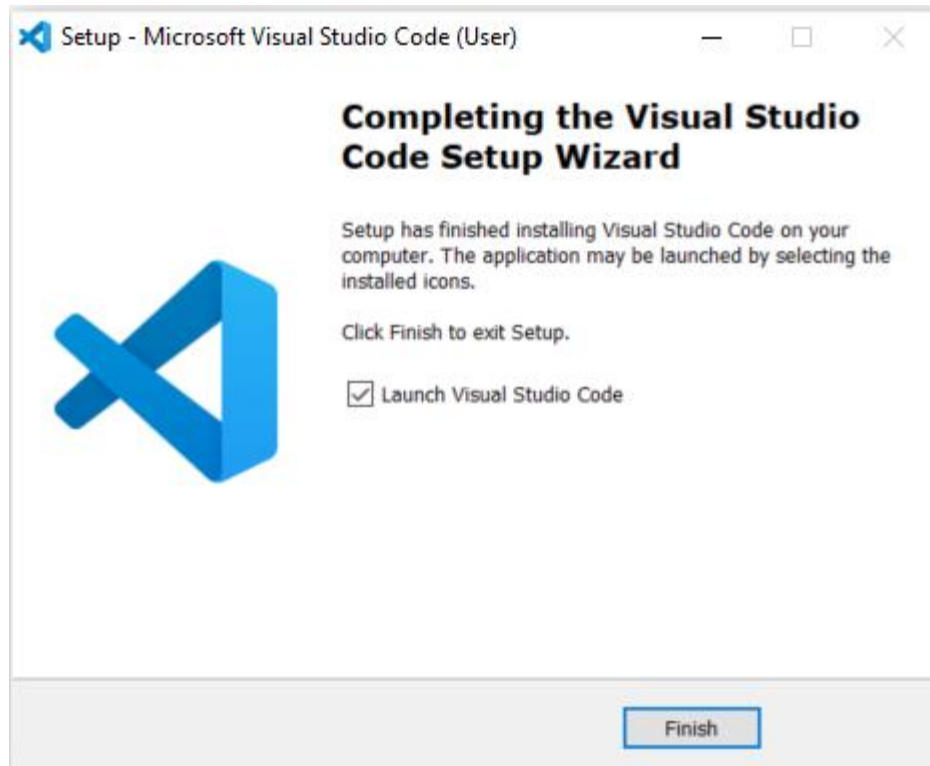


19. Aktifkan Windows Hypervisor Platform, lalu OK dan Restart Komputer

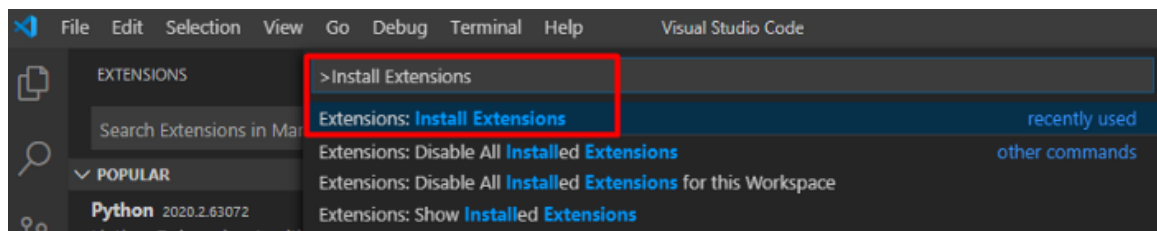


- **Visual Studio Code**

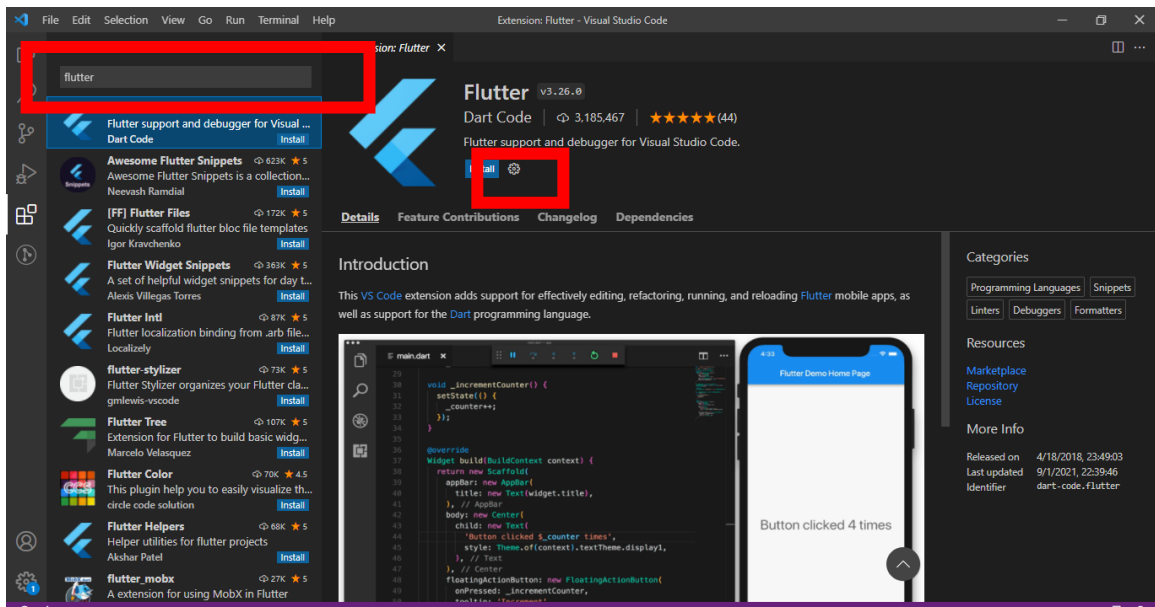
1. Silakan download Visual Studio code di <https://code.visualstudio.com/download>. Lalu lakukan instalasi sampai muncul seperti ini:



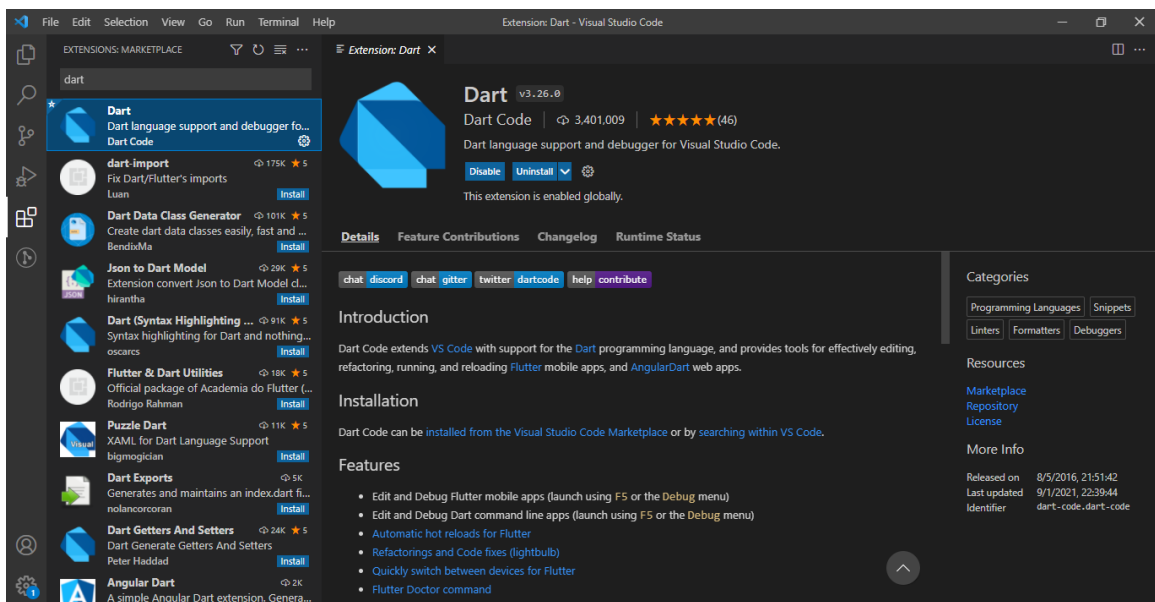
2. Selanjutnya lakukan instalasi extension Flutter dan Dart pada visual studio. Klik View-> Command Pallet -> Ketik Install Extensions lalu enter



3. Search Flutter lalu Install

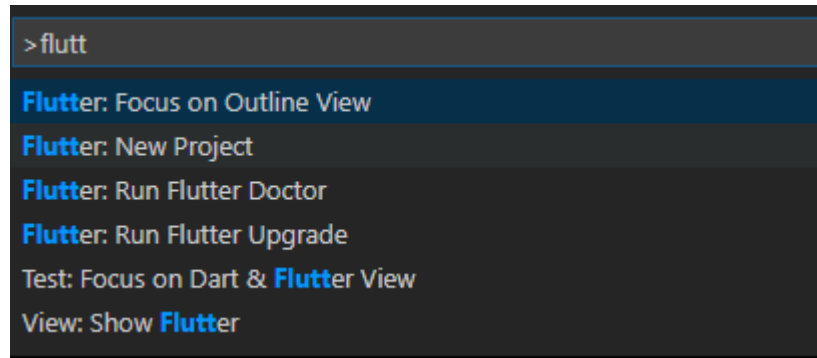


4. Lakukan hal sama untuk extensi Dart (jika belum terinstall)

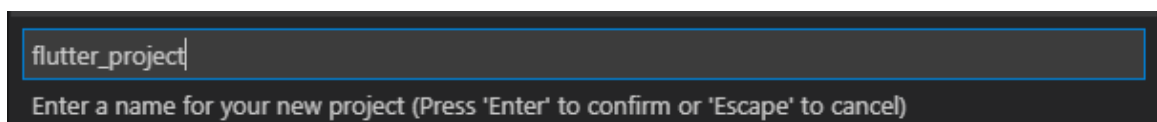


- Uji Coba Project Dengan Flutter

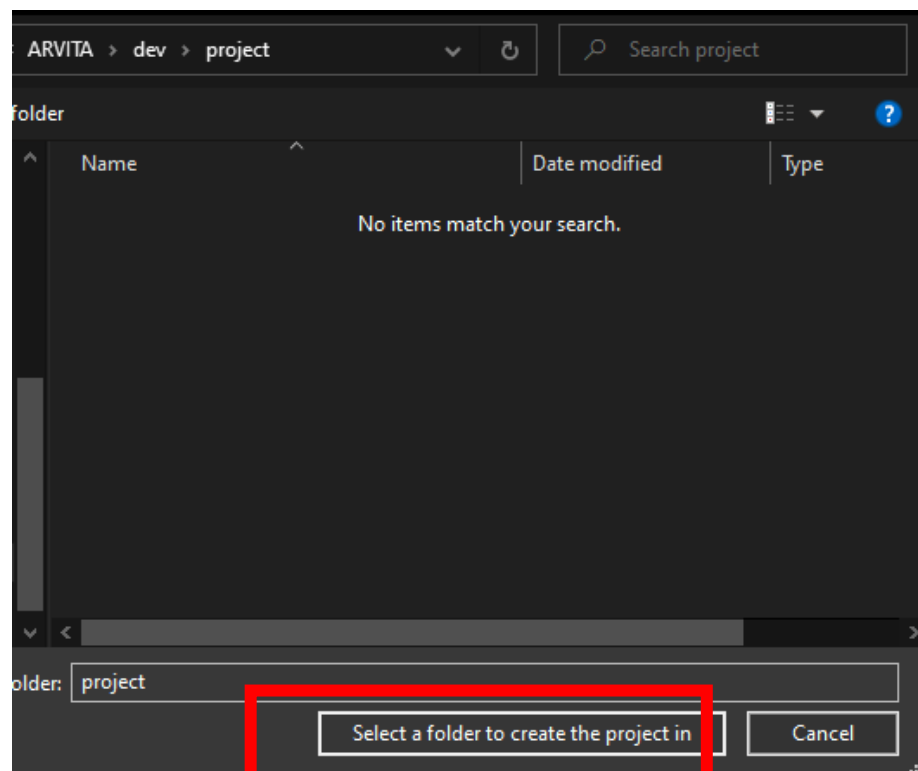
1. Pada Visual Studio Code, tekan F1, lalu ketik Flutter: New Project

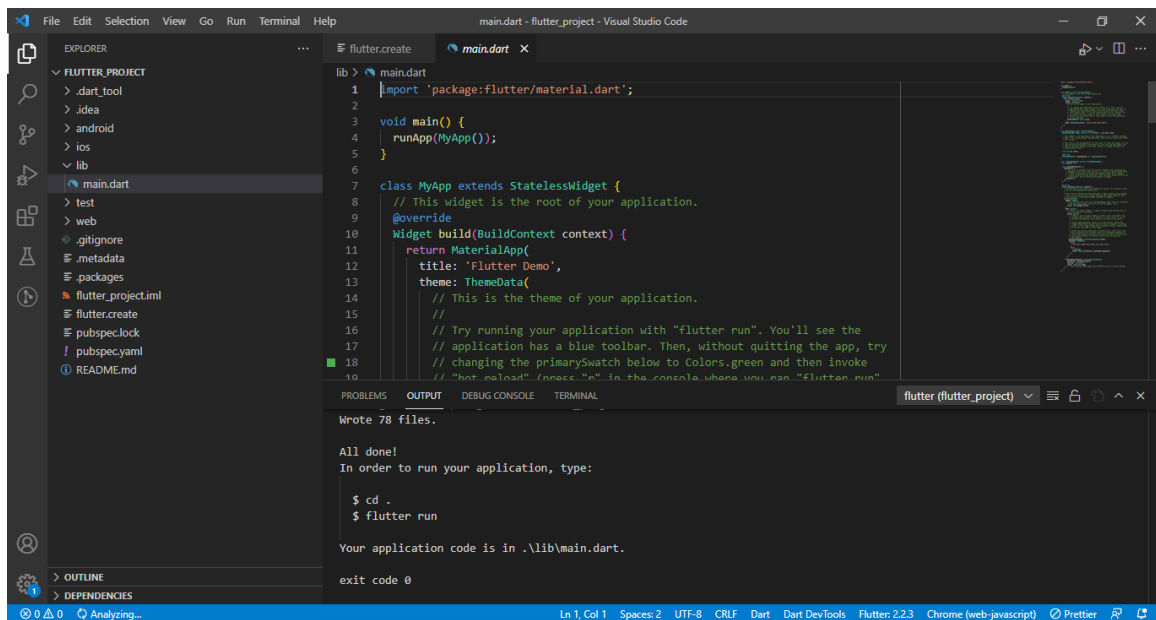


2. Contohnya nama projectnya adalah flutter_project

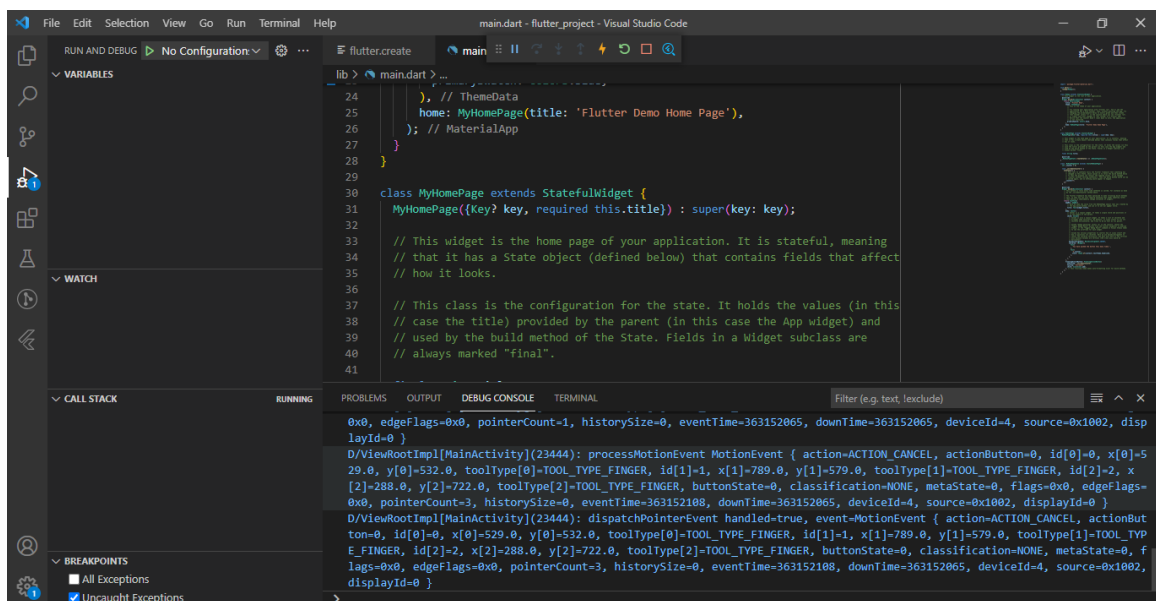


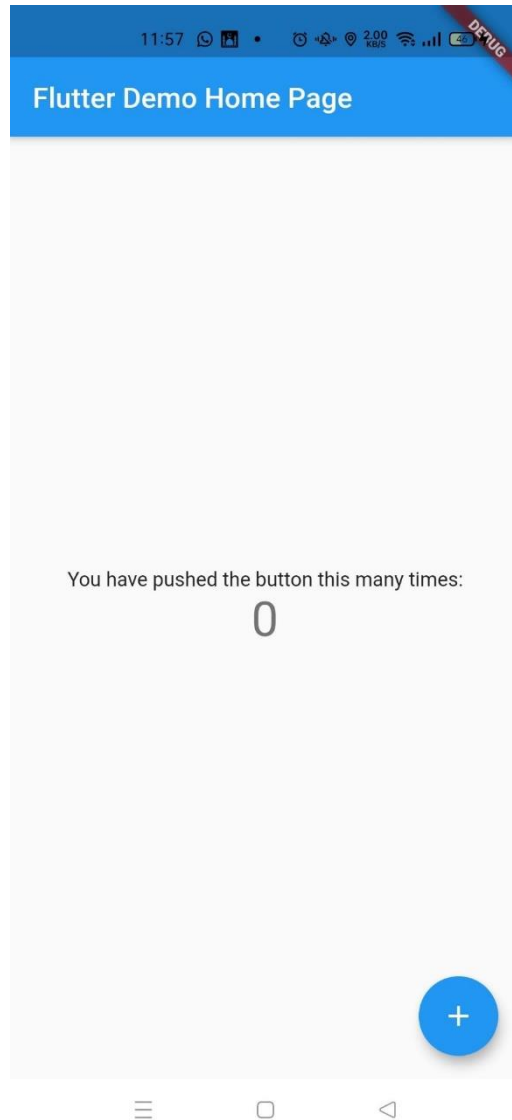
3. Pilih tempat menyimpan, silahkan sesuaikan dengan kebutuhan





4. Tekan F5, lalu pilih emulator untuk menjalankan

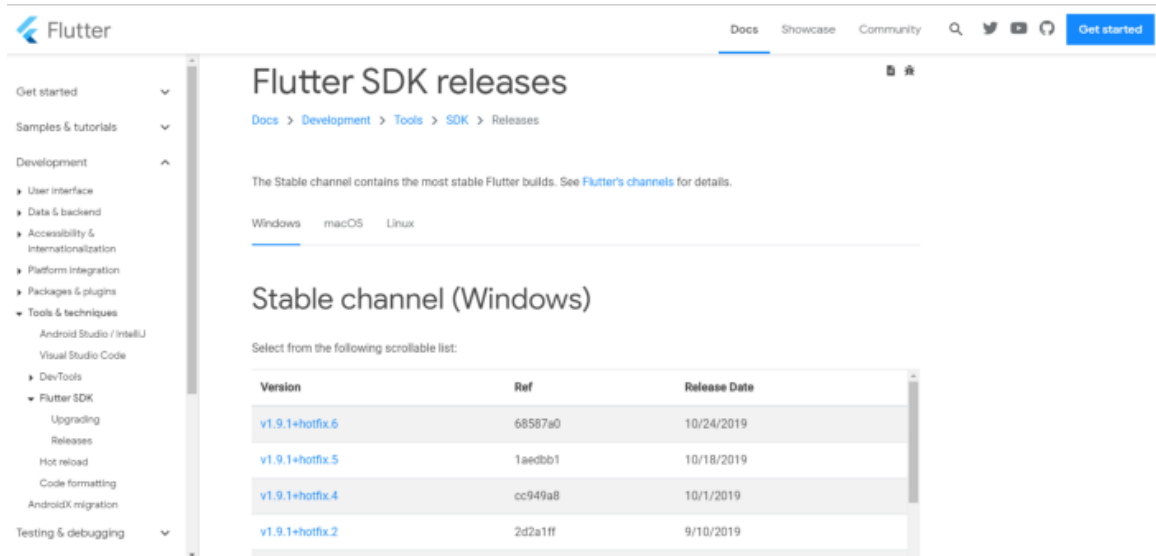




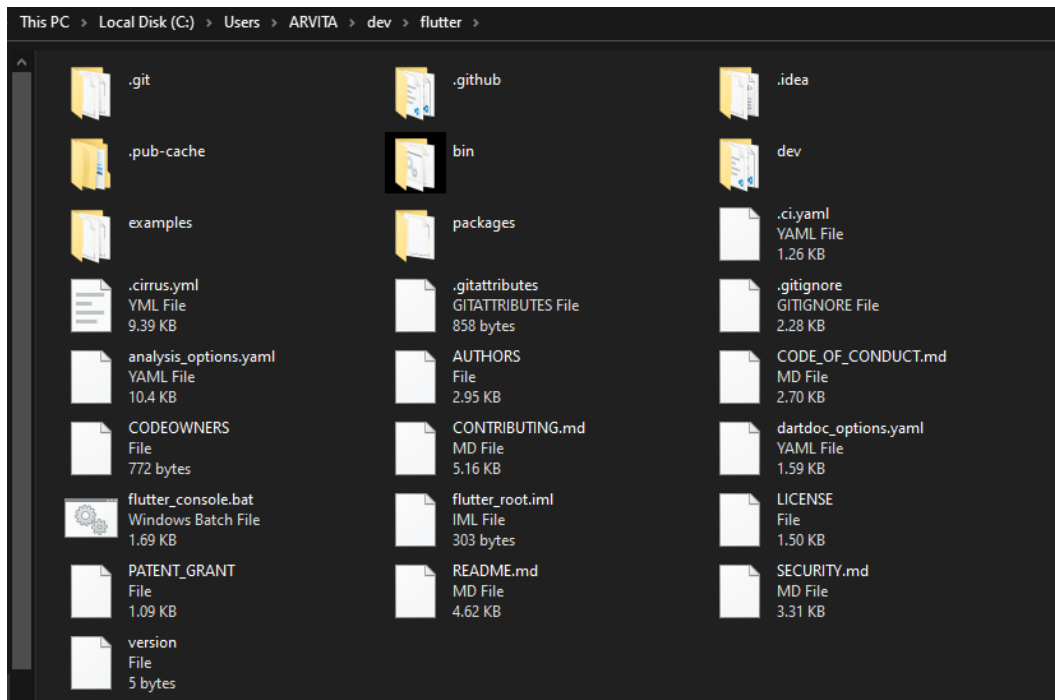
B. Menggunakan Android Studio

1. Unduh Flutter SDK <https://flutter.dev/docs/development/tools/sdk/releases>

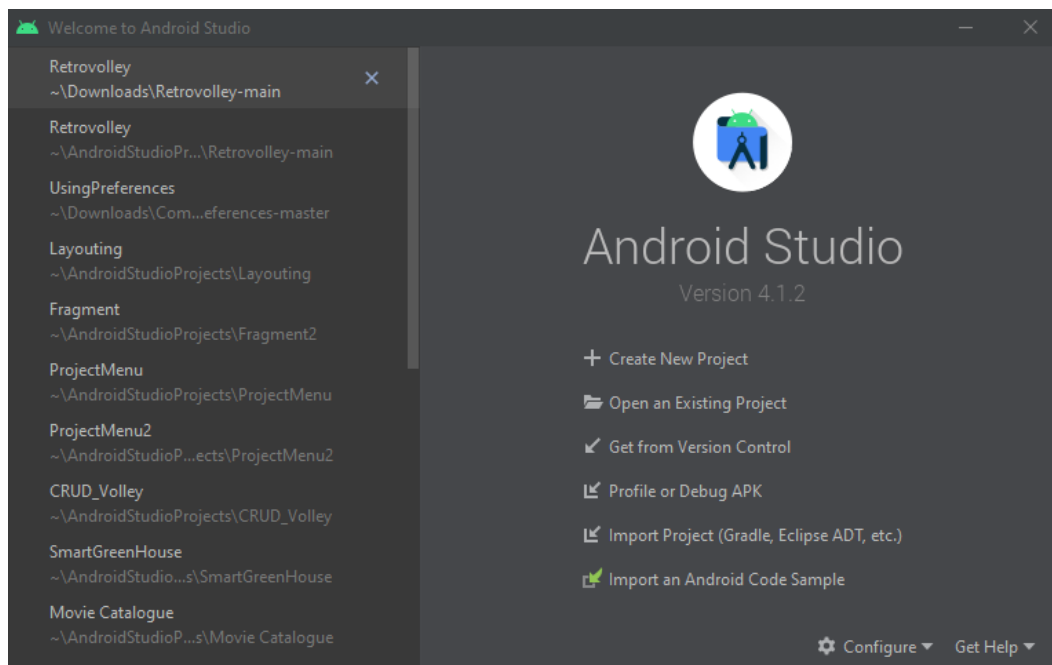
Untuk pemula, untuk memilih Flutter SDK versi terbaru (paling atas) dengan tipe *Stable Channel*.



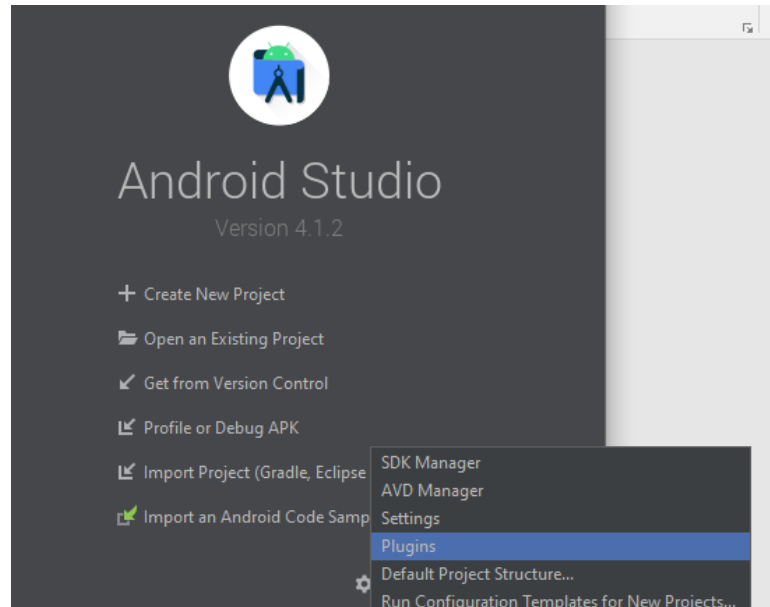
2. Ekstrak berkas ZIP/RAR Flutter SDK yang sebelumnya telah anda unduh, ke dalam direktori folder yang anda inginkan.
3. **Simpanlah di dalam direktori yang mudah diakses.** Contoh: C:\Users\ARVITA\dev\flutter



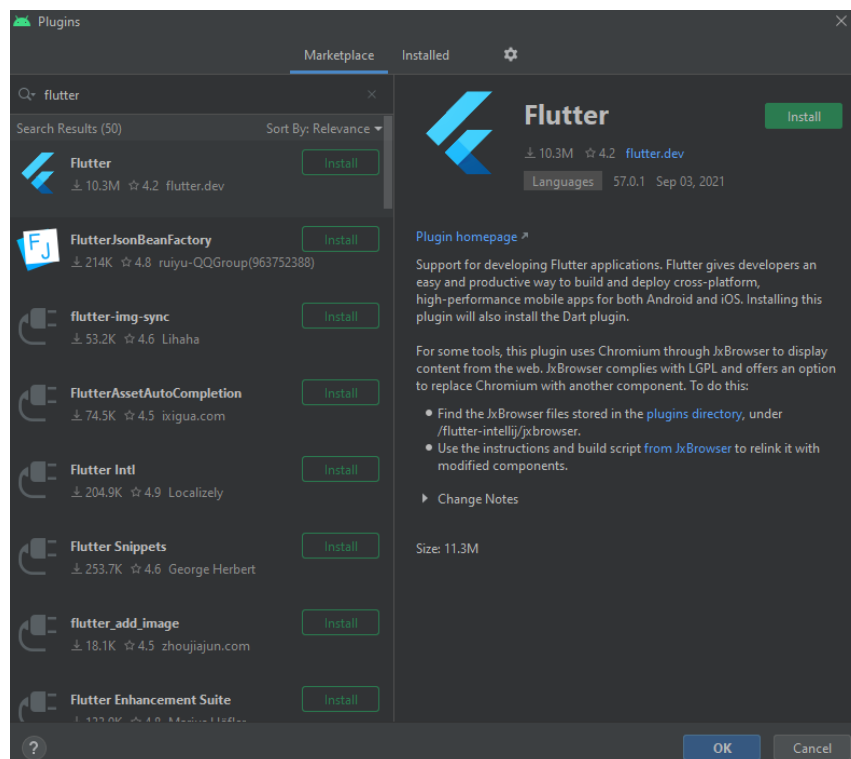
4. Buka *Android Studio* yang telah terpasang pada komputer anda, lalu tunggu hingga halaman awal *Android Studio* ditampilkan.



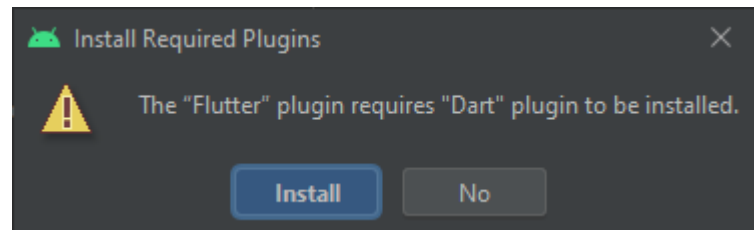
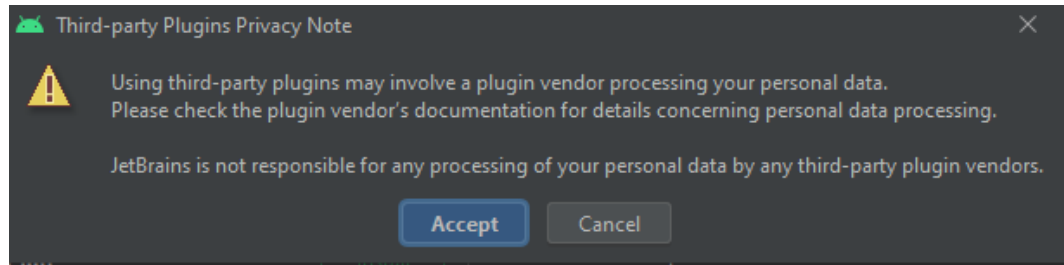
5. Buka *Configure* -> *Plugins*



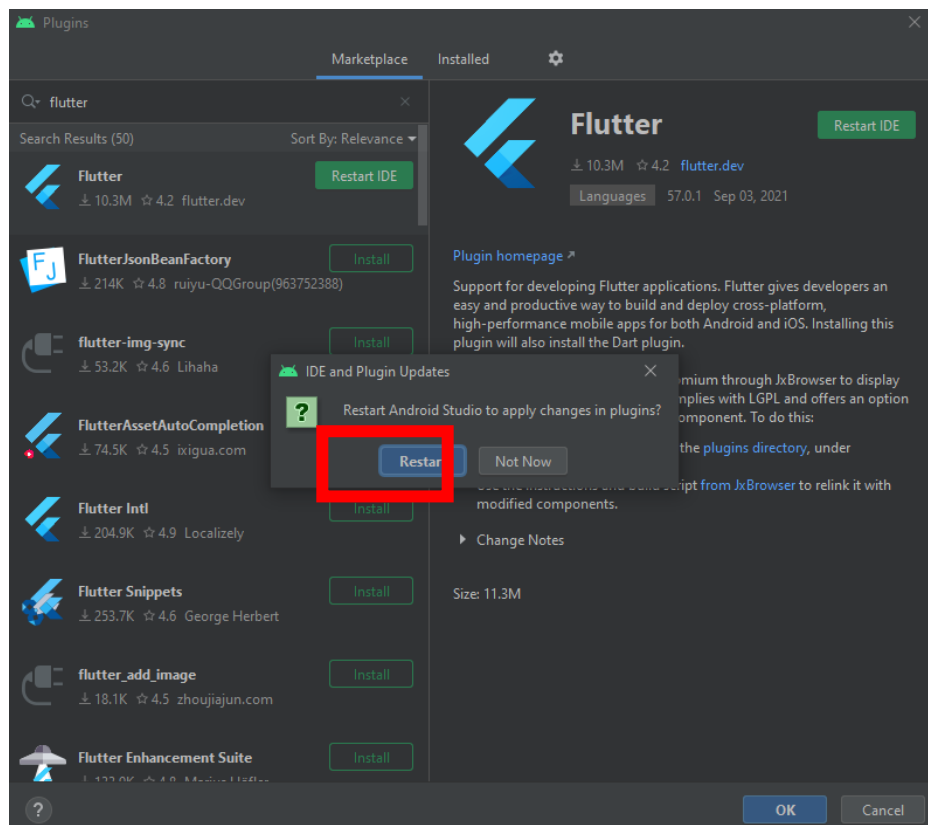
6. Setelah muncul jendela **Plugins**, maka pilihlah tab **Marketplace**, kemudian lakukan pencarian dengan kata kunci: “**flutter**”. Terakhir tekan tombol **Install**



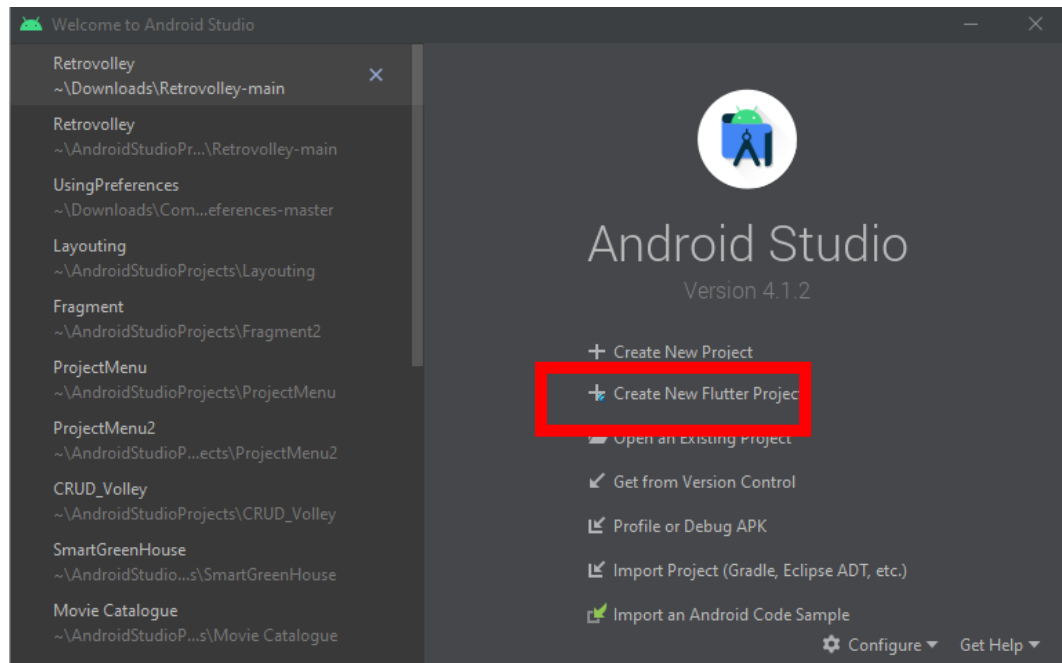
7. Jika anda mendapat *notification* berikut ini, pilih saja “Yes”



8. Tunggu proses pemasangan *plugin* hingga selesai.
9. Setelah itu pilihlah tombol "***Restart IDE***".

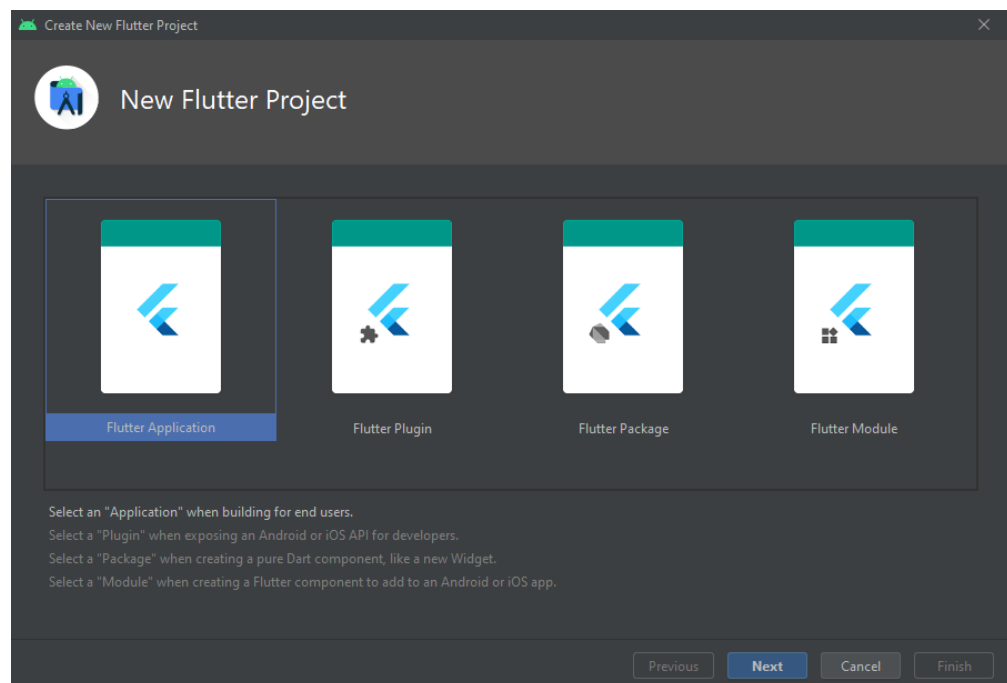


10. Tunggu proses mulai ulang *Android Studio*, hingga muncul kembali halaman awalnya seperti ini, dan lihatlah perbedaannya.

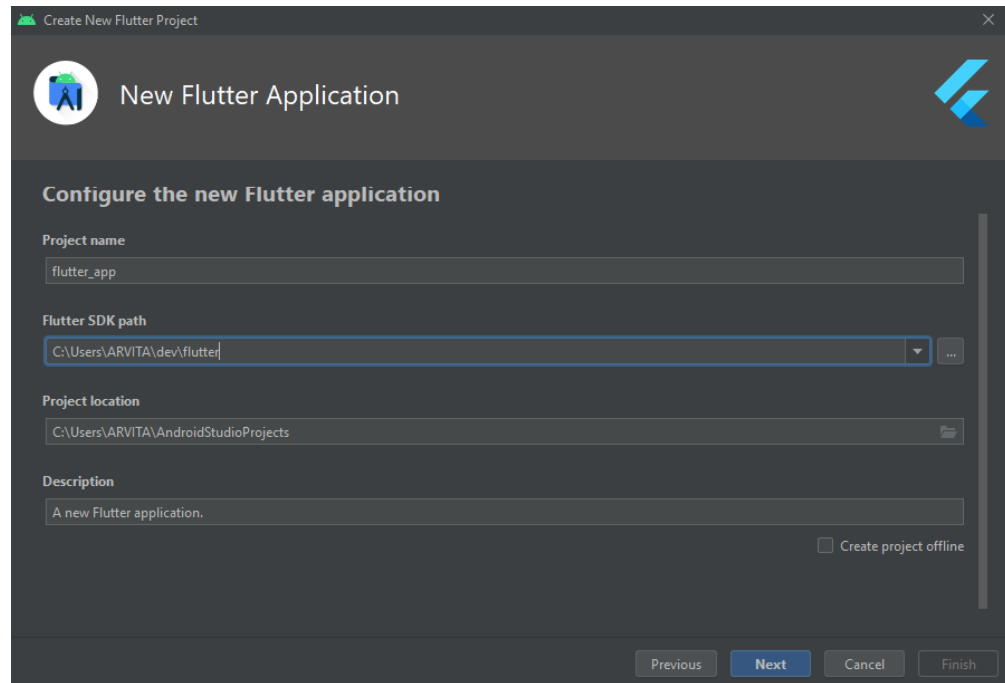


11. Pilihlah “*Create new flutter project*” di halaman awal *Android Studio* anda.

12. Kemudian, pilihlah “*Flutter Application*” lalu klik “*Next*”.



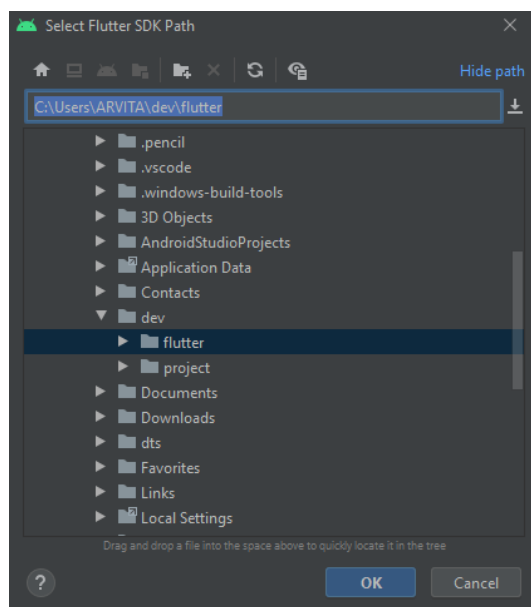
13. lalu akan muncul jendela berikut ini:



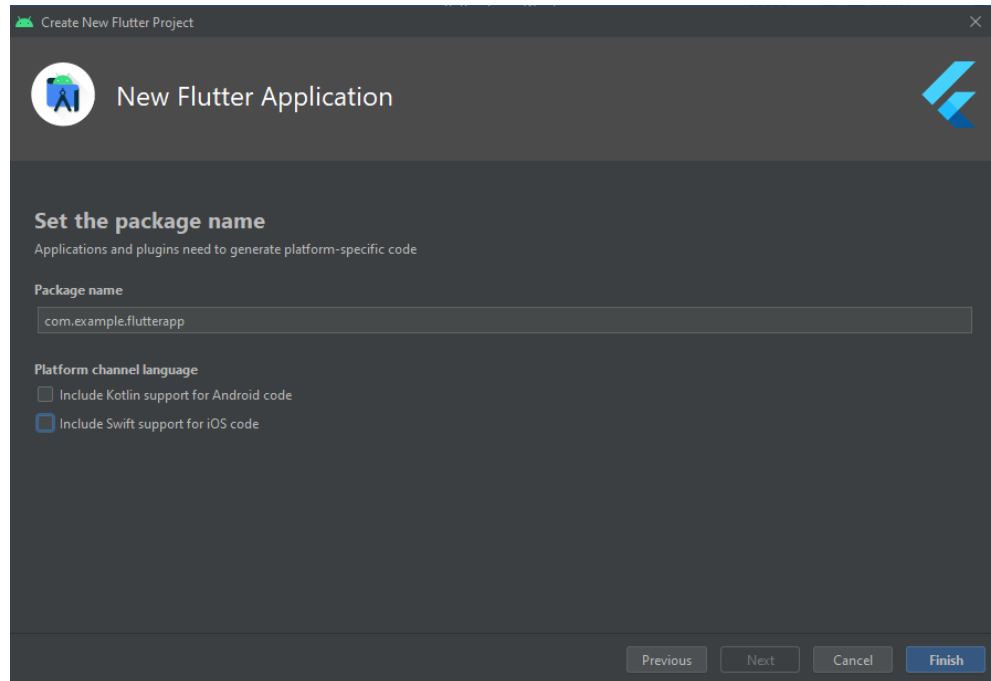
14. Silahkan beri *project name* sesuai dengan keinginan anda. Lakukan penamaan dengan menggunakan **huruf kecil dan tanpa spasi**.

15. Kemudian pada bagian *flutter SDK path*, tentukanlah direktori penyimpanan folder Flutter SDK yang telah kita unduh pada **langkah no. 1–2** di atas.

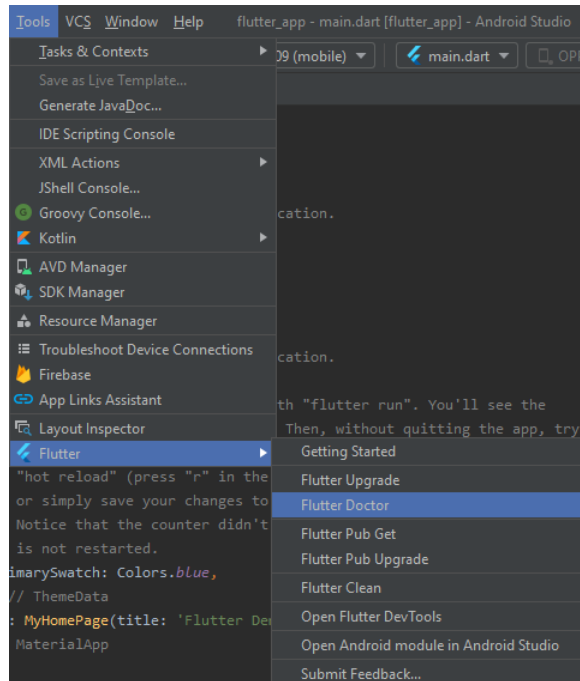
16. Tekan tombol diatas untuk menentukan direktori Flutter SDK



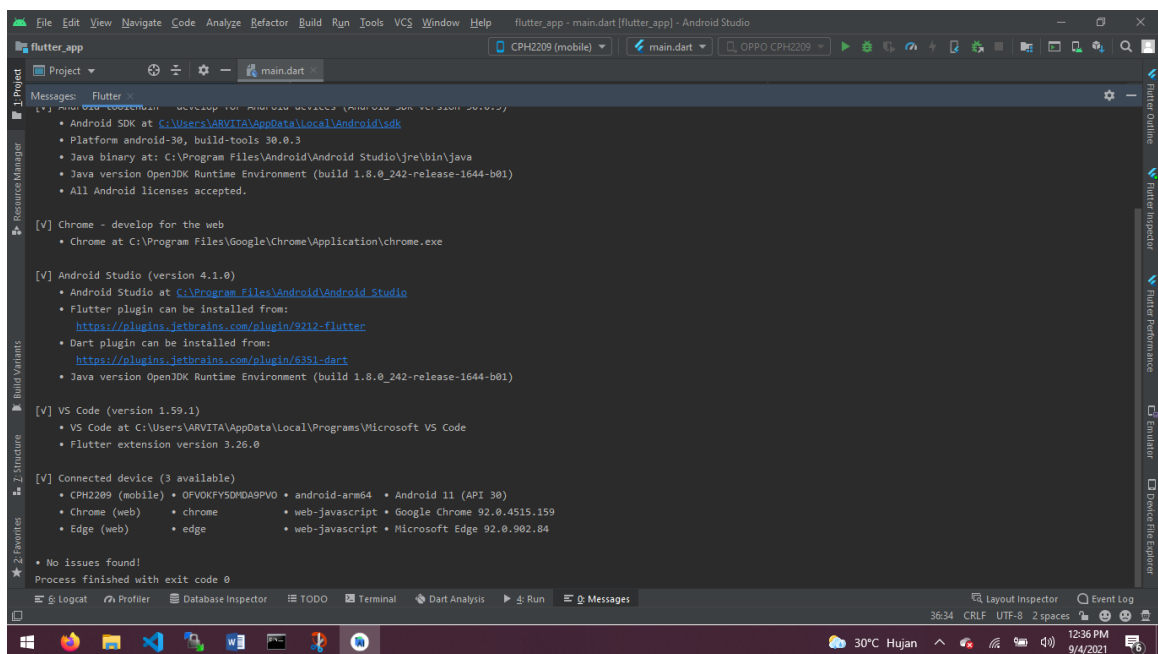
17. Untuk bagian *project location*, silahkan anda tentukan tempat untuk menyimpan project anda.
Direkomendasikan agar nama folder tidak mengandung spasi.
18. Jika semua konfigurasi telah selesai, silahkan klik “*Next*”.
19. Kemudian pada jendela selanjutnya, anda harus menentukan domain aplikasi, untuk keperluan penamaan direktori *package* pada aplikasi anda. Untuk bagian ***AndroidX*** sebaiknya **di ceklis** dan untuk ***Platform channel language*** tidak perlu di ceklis.



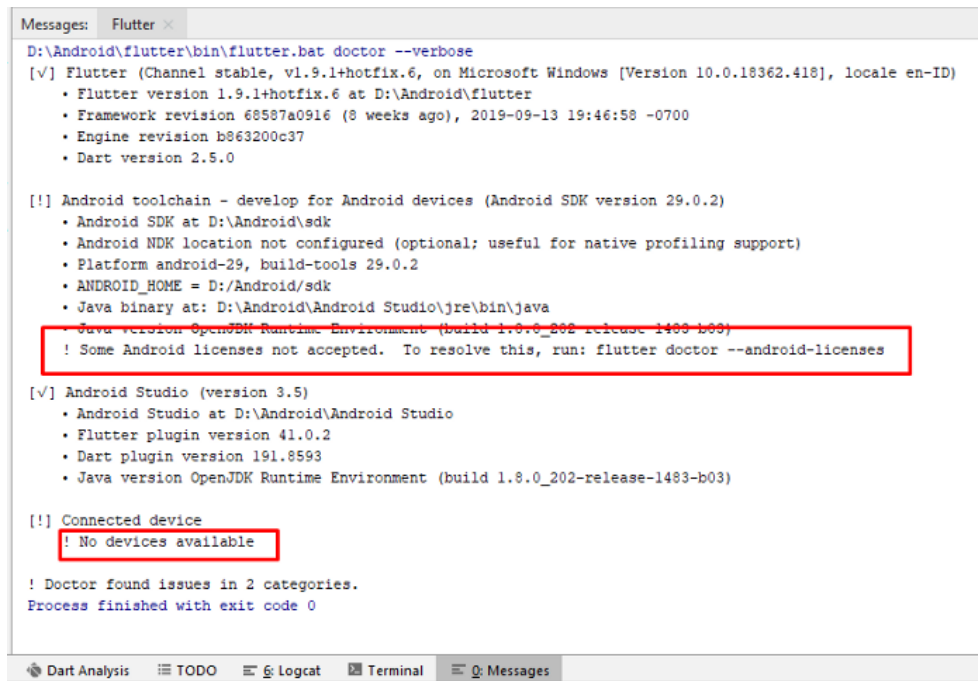
20. Jika telah selesai semua, klik “*Finish*”.
21. Tunggu proses *building* hingga selesai.
22. Project telah berhasil dibuat
23. Di dalam Android Studio, bukalah menu **Tools -> Flutter -> Flutter Doctor** seperti contoh gambar berikut:



24. Lihatlah hasil keterangan yang muncul dalam tab **Messages**, seperti gambar berikut ini:



25. Jika ada issue seperti berikut :



```
Messages: Flutter x
D:\Android\flutter\bin\flutter.bat doctor --verbose
[✓] Flutter (Channel stable, v1.9.1+hotfix.6, on Microsoft Windows [Version 10.0.18362.418], locale en-ID)
    • Flutter version 1.9.1+hotfix.6 at D:\Android\flutter
    • Framework revision 68587a0916 (8 weeks ago), 2019-09-13 19:46:58 -0700
    • Engine revision b963200c37
    • Dart version 2.5.0

[!] Android toolchain - develop for Android devices (Android SDK version 29.0.2)
    • Android SDK at D:\Android\sdk
    • Android NDK location not configured (optional; useful for native profiling support)
    • Platform android-29, build-tools 29.0.2
    • ANDROID_HOME = D:/Android/sdk
    • Java binary at: D:\Android\Android Studio\jre\bin\java
    • Java version OpenJDK Runtime Environment (build 1.8.0_202-release-1483-b03)
    ! Some Android licenses not accepted. To resolve this, run: flutter doctor --android-licenses

[✓] Android Studio (version 3.5)
    • Android Studio at D:\Android\Android Studio
    • Flutter plugin version 41.0.2
    • Dart plugin version 191.8593
    • Java version OpenJDK Runtime Environment (build 1.8.0_202-release-1483-b03)

[!] Connected device
    ! No devices available

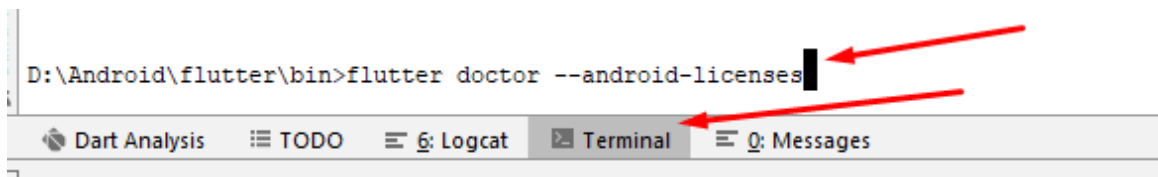
! Doctor found issues in 2 categories.
Process finished with exit code 0
```

Penyebab *issues* tersebut adalah:

Some Android licenses not accepted.... : diharuskan menyetujui lisensi (syarat & ketentuan) dari Android terlebih dahulu.

No devices available : Tidak ada perangkat ponsel Android yang terhubung ke komputer.

26. Untuk mengatasi *issues* tersebut, bukalah tab **Terminal**, kemudian masuklah ke folder “**bin**” dalam **direktori penyimpanan Flutter SDK** anda pada tahap nomor 1–2.



```
D:\Android\flutter\bin>flutter doctor --android-licenses
```

27. Setelah masuk folder “**bin**”, ketiklah perintah android-licenses seperti pada contoh dalam gambar diatas!

28. Berikan **input: y** untuk menyetujui license

```
Terminal: Local +
D:\Android\flutter\bin>flutter doctor --android-licenses
Warning: File C:\Users\TIJUNIDA\.android\repositories.cfg could not be loaded.
1 of 7 SDK package license not accepted. 100% Computing updates...
Review license that has not been accepted (y/N)? y
1/1: License android-sdk-arm-dbt-license: input: y
-----
Terms and Conditions
This is the Android Software Development Kit License Agreement
1. Introduction

January 16, 2019 input: y
-----
Accept? (y,N): y
All SDK package licenses accepted
```

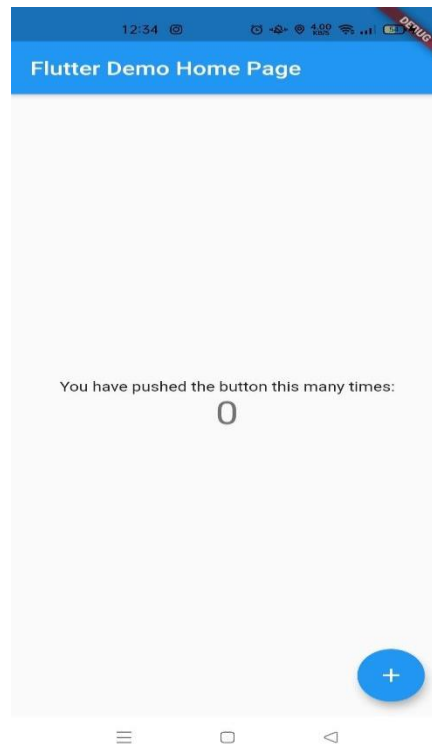
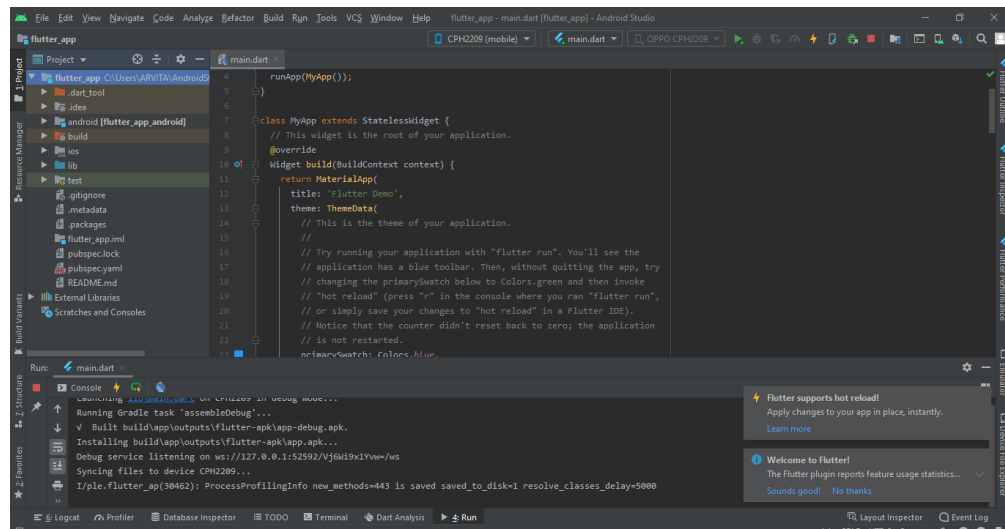
29. Kemudian, ketik perintah **flutter doctor** lagi untuk mengecek apakah *issues* sudah diperbaiki atau belum.

```
Terminal: Local +
D:\Android\flutter\bin>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, v1.9.1+hotfix.6, on Microsoft Windows [Version 10.0.18362.418], locale en-ID)
[✓] Android toolchain - develop for Android devices (Android SDK version 29.0.2)
[✓] Android Studio (version 3.5)
[!] Connected device
    ! No devices available

! Doctor found issues in 1 category.
```


30. Hasil flutter doctor untuk kedua kalinya

31. Running Flutter App



f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu tahap dan langkah-langkah dalam menginstallasi flutter.
2. Penjelasan pada laporan maksud dan langkah-langkah yang telah dilakukan.

g. Kesimpulan

Mahasiswa berhasil melakukan instalasi dan memahami maksud dari setiap langkah-langkah dalam instalasi flutter yang dilakukan pada prosedur kerja.

h. Rubrik Penilaian

No	Indikator	Skor*			
1	Ketepatan waktu dan ketepatan dalam menjelaskan secara rinci command git yang digunakan pada revisi pertama	1	2	3	④
2	Ketepatan waktu dan ketepatan dalam menjelaskan command git yang digunakan pada revisi pertama	1	2	③	4
3	Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas	1	②	3	4
4	Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas	①	2	3	4
Jumlah skor					

ACARA 4 : PENGENALAN DAN WORKFLOW PADA GIT, FLUTTER & DART

Pokok Bahasan	: Pengelolaan Project pada Git
Acara Praktikum/Pertemuan	: Minggu 1 / 4
Tempat	: Daring / Luring Politeknik Negeri Jember
Alokasi Waktu	: 100 menit

a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu mengelola project pada git
2. Mahasiswa menjelaskan maksud membuat repository dan pengelolaan project pada git

b. Indikator

Keberhasilan mahasiswa dalam melakukan mengelola project pada git, serta memahami setiap command git revisi berikutnya.

c. Dasar Teori

-

d. Alat dan Bahan

1. Kertas HVS
2. Spidol Besar
3. Bolpoin

e. Prosedur Kerja

Membuat project Git dan mengelolanya

Pastikan sudah install git di komputer anda. Jika belum terinstall silakan ikuti langkah-langkahnya di sini: <https://www.digitalocean.com/community/tutorials/how-to-contribute-to-open-source-getting-started-with-git>

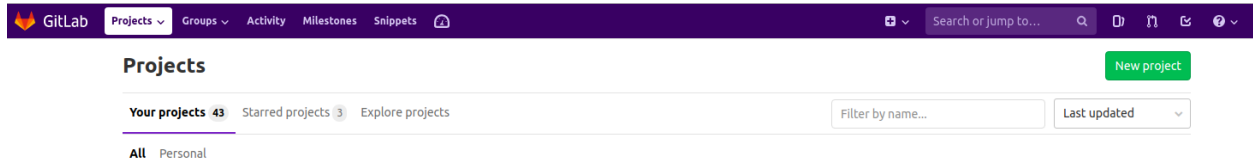
Materi Video Hari Pertama

bagi teman teman yang membutuhkan materi video silahkan akses video diatas

Membuat Repository Baru

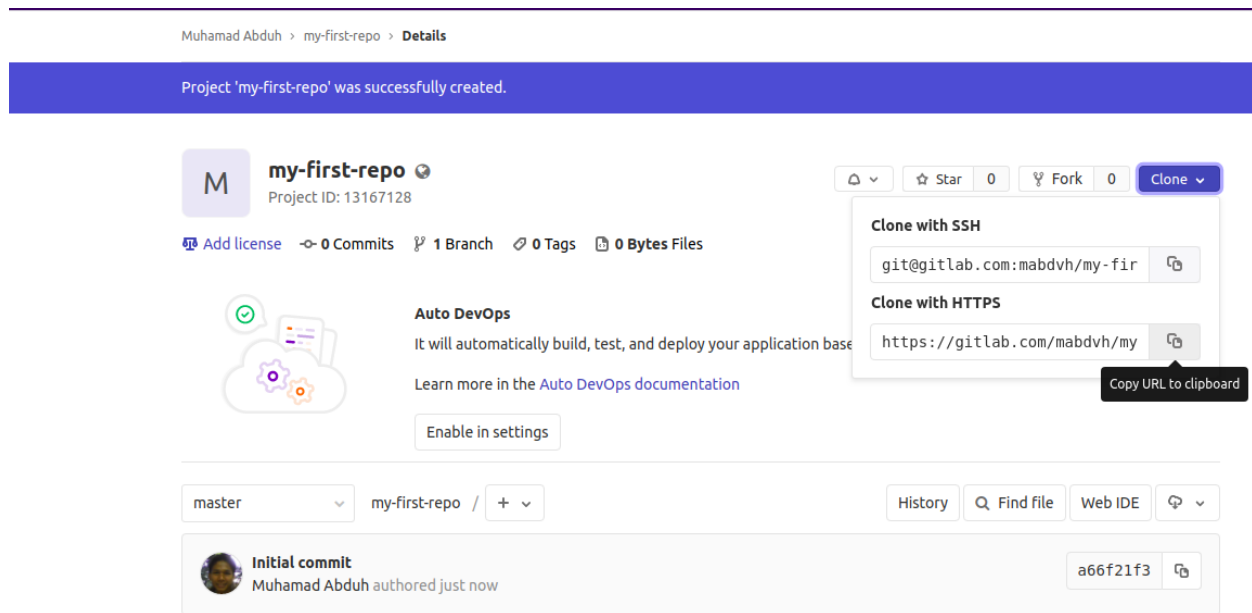
Repository atau repo di Gitlab biasa disebut dengan project. Berikut langkah-langkah untuk membuat repository baru:

1. Setelah login anda akan menuju halaman Projects yang berisi semua repository yang anda miliki. Jika Anda user baru maka belum ada repository yang tersedia. Di halaman tersebut anda akan melihat tombol New Project di sebelah kanan atas. Klik tombol tersebut.



2. Pada halaman membuat project baru, isi kolom project-name dengan nama yang Anda inginkan, misalkan: my first repo.
3. Pada kolom visibility level Anda akan melihat tiga pilihan yaitu Private, Internal, dan Public. Private artinya repo yang akan kita buat bersifat rahasia, hanya Anda dan orang yang Anda beri akses saja yang bisa membukanya. Sedangkan pilihan Internal berarti repo Anda akan dapat dilihat oleh semua user Gitlab. Pilihan “Public” berarti repo Anda akan terlihat oleh orang dari mana saja walaupun dia tidak memiliki akun gitlab. Pilih yang sesuai kebutuhan misalkan : Public.
4. Di bagian bawahnya terdapat pilihan checkbox “Initialize repository with a README”. Centang saja jika anda ingin langsung clone repository nya setelah dibuat.

5. Jika berhasil maka Anda akan diarahkan menuju halaman project atau repo yang telah dibuat. Jika sebelumnya kita centang “Initialize repository with a README” maka repository kita akan berisi file README.md.
6. Pada Bagian kanan atas terdapat tombol clone. Ketika diklik maka akan keluar dua pilihan “Clone with SSH” dan “Clone with HTTPS”. Pilih saja clone dengan HTTPS. Klik tombol salin/copy supaya langsung otomatis tercopy pada clipboard. Jika ingin clone dengan SSH maka anda harus setting dulu akun anda agar terhubung dengan komputer Anda.



7. Setelah itu menuju ke terminal atau cmd. Arahkan menuju folder dimana anda ingin menyimpan repository yang sudah Anda buat. Berikan perintah git clone <link_anda_yang_barusan_dicopy> (tanda <> hanyalah sebagai penanda template tidak usah ditulis di perintah cmd)

```
$ git clone <your_repository_link_url>
```

8. Jika sudah berhasil diclone maka akan ditemukan folder dengan nama repository di Gitlab. Cara mengeceknya dengan perintah “ls” atau “dir”.

```
$ ls
my-first-repo
```

Mengupload perubahan project dengan git push

Ketika kita sudah berhasil clone repository kita di Gitlab maka tentu kita ingin melakukan update atau penambahan code pada project kita. Agar perbaruan yang kita simpan di local (komputer) kita

dapat kita perbaharui juga di repository Gitlab maka kita bisa lakukan dengan git push. Berikut langkah-langkahnya:

1. Ketika kita sudah memberikan perubahan terhadap suatu file atau membuat folder-folder baru pada folder project kita maka cara mengecek status perbaruan adalah dengan git status

```
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
```

2. Setelah git status maka akan ditampilkan list file yang “modified” atau “untracked”. “modified” artinya adalah kita mengubah file yang sebelumnya ada sedangkan “untracked” artinya file tersebut sama sekali baru di repository. Dalam contoh di atas terdapat perubahan pada file “README.md”.

3. Untuk menaikkan perubahan (stage) yang kita buat maka berikan perintah git add

```
$ git add README.md

// Jika ada terdapat banyak file dan kita ingin naikkan semuanya maka perintahnya:
$ git add .
```

4. Sekarang perubahan tadi sudah pada tahap “staged”. jika kita cek status nya lagi maka akan didapati berbeda:

```
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   README.md
```

5. Selanjutnya adalah memberikan laporan atau commit terhadap perubahan yang kita buat dengan git commit.

```
$ git commit -m "Pesan saya: Mengubah README.md"
```

6. tanda “-m” pada perintah commit artinya kita ingin memberikan pesan atau laporan terhadap perubahan yang sudah dibuat. Dalam hal ini laporannya adalah “Pesan saya: mengubah README.md”

7. Selanjutnya adalah push untuk mengirim /upload perubahan yang ada di local computer kita menuju repository yang ada di Gitlab. Jika kita berada di branch bernama master maka perintahnya adalah:

```
$ git push origin master
```

Jika branch yang sedang kita miliki di local adalah branch lain selain master maka cara push nya adalah

```
$ git push origin <nama_branch>
```

Jika langkah tersebut sudah maka kita sudah berhasil mengupload perubahan yang kita buat di repository kita.

Git branch dan merge request (optional)

Membuat branch baru dan melakukan merge request

Clone repository

Terlebih dahulu lakukan clone terhadap repository project.

```
$ git clone https://gitlab.com/some-project  
$ cd some-project
```

Catatan: Link https di atas hanyalah contoh link dari sebuah repository. Link https tersebut bisa diperoleh di halaman gitlab pada bagian clone.

Buat branch baru

Setelah di-clone dan masuk ke folder project nya, buatlah branch baru . (myname hanyalah contoh nama branch, kita bisa mengubah sesuai nama yang diinginkan)

```
$ git branch myname  
$ git checkout myname
```

maka sekarang kita sudah berada di branch baru dan dapat mulai melakukan pekerjaan kita.

Catatan:

myname hanya contoh, silakan sesuaikan sendiri nama branch masing-masing.

Push ke branch

Setelah pekerjaan selesai, kita akan melakukan upload atau push ke branch kita sendiri.

```
$ git add .  
$ git commit -m "some message"// push ke branch  
$ git push origin myname
```

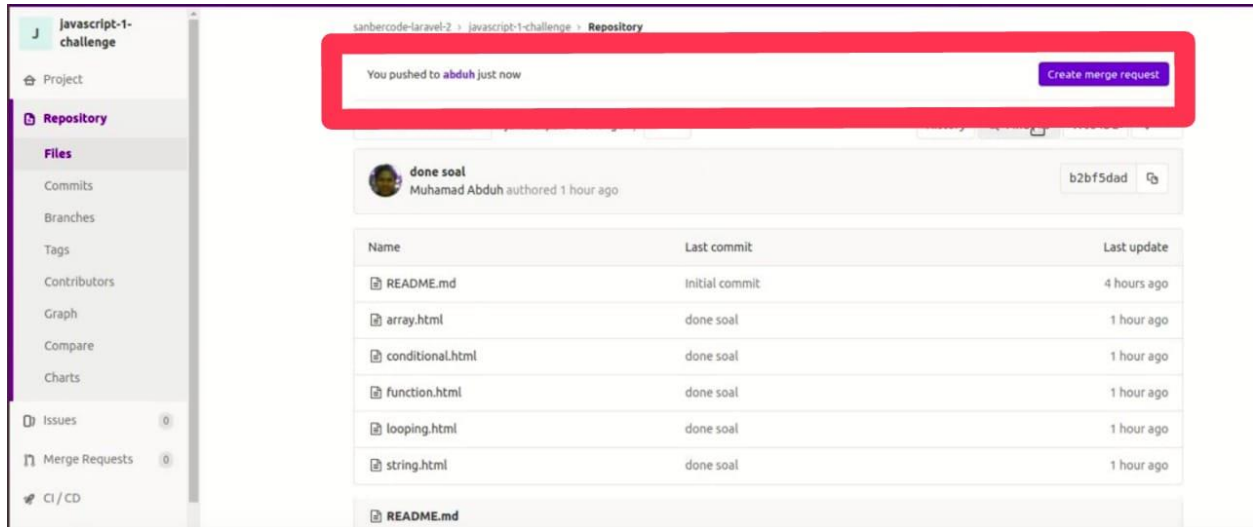
Catatan:

"some message" hanyalah contoh pesan atau komentar pada sebuah commit. Silahkan isi dengan jenis update yang dikerjakan.

Membuat merge request

Setelah melakukan push ke branch sendiri maka selanjutnya melakukan merge request.

Secara default pada halaman project akan tersedia tombol “create merge request”.



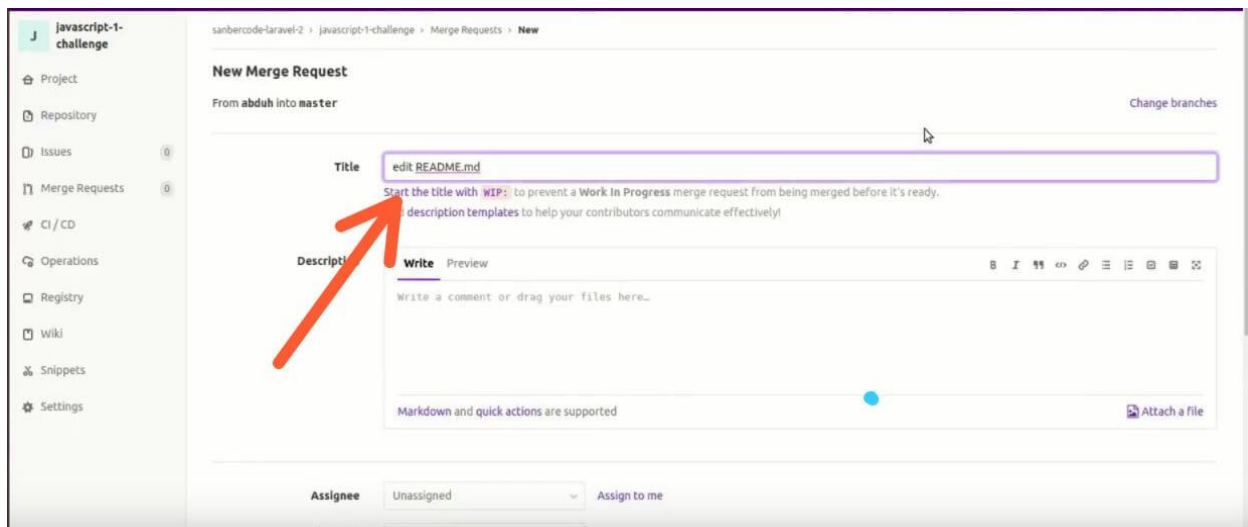
Jika tombol tersebut tersedia, klik tombol tersebut untuk menuju halaman merge request.

Pada halaman merge request isilah kolom title dengan pesan commit atau laporan progress pekerjaan misalkan : done, work in progress atau WIP, dsb.

Setelah itu, klik submit merge request maka pekerjaan kita sudah disubmit ke merge request dan akan direview oleh pengelola project.

Selengkapnya kamu bisa ikuti juga video tutorial berikut ini: tutorial merge request.

1. vscode , link download : <https://code.visualstudio.com/download>
2. Cmdr, link download: <https://cmdr.net/cmdr>
3. Git bash, link download: <https://git-scm.com/downloads>, Git bash biasanya ter-install jika kita meng-install git di pc/laptop kita.



f. Hasil dan Pembahasan

1. Dokumentasi tugas berupa laporan secara individu tahap pengelolaan project pada git yang dibuktikan dengan screenshot pada masing-masing pekerjaannya.
2. Penjelasan pada laporan maksud dari setiap tahap dan langkah pengelolaan project pada git yang dilakukan.

g. Kesimpulan

Mahasiswa berhasil melakukan pengelolaan project pada git.

h. Rubrik Penilaian

No	Indikator	Skor*			
1	Ketepatan waktu dan ketepatan dalam menjelaskan secara rinci command git yang digunakan pada revisi kedua	1	2	3	④
2	Ketepatan waktu dan ketepatan dalam menjelaskan command git yang digunakan pada revisi kedua	1	2	③	4
3	Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas	1	②	3	4
4	Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas	①	2	3	4
Jumlah skor					