

MODUL

PEMROGRAMAN BERBASIS PLATFORM



PERTEMUAN - 1

PENDAHULUAN DAN INSTALASI

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ATMA JAYA YOGYAKARTA





Daftar Gambar

| | |
|--|----|
| Gambar 1 SDK Flutter | 6 |
| Gambar 2 Pastikan File yang diekstrak berada pada Foler C:\src..... | 7 |
| Gambar 3 Copy Path Bin untuk melakukan setting Environment | 7 |
| Gambar 4. Paste path flutter bin yang sudah dicopy sebelumnya | 9 |
| Gambar 5 Jalankan perintah "flutter doctor"..... | 10 |
| Gambar 6 Pilih Bagian Community, Lalu tunggu hingga download selesai..... | 11 |
| Gambar 7. Tunggu hingga proses Instalasi berhasil | 11 |
| Gambar 8 Centang pada bagian Desktop development with C++ Lalu install..... | 12 |
| Gambar 9 Tunggu hingga proses selesai..... | 12 |
| Gambar 10 Sudah tidak ada issue dan flutter siap digunakan..... | 13 |
| Gambar 11 Pilih instalasi berdasarkan OS yang digunakan | 14 |
| Gambar 12 Pilih bagian "I accept the agreement": | 15 |
| Gambar 13 Klik Install dan tunggu hingga proses selesai..... | 15 |
| Gambar 14 Tampilan Awal dari Vs. Code..... | 16 |
| Gambar 15 Cara menambahkan ekstension Flutter pada Vs. Code..... | 17 |
| Gambar 16 Halaman Utama Github | 21 |
| Gambar 17 Pilih Windows | 22 |
| Gambar 18 Jika sudah tampil seperti ini, maka akun git sudah berhasil dibuat dan dapat digunakan..... | 25 |
| Gambar 19 Folder Yang akan dipush pada github..... | 26 |
| Gambar 20 Pilih Git Base Here | 26 |
| Gambar 21 Lalu Masukkan nama Repository yang diinginkan, untuk jenis dari repository nya tergantung kebutuhan, boleh di public atau di private, lalu klik "Create repository"..... | 28 |
| Gambar 22 Code baris 1 dan 2 setelah dimasukkan pada git base | 29 |
| Gambar 23 Git add. dan git commit | 30 |
| Gambar 24 Code pada baris ke 3 | 30 |
| Gambar 25 Code sudah berhasil di push pada github menggunakan Git..... | 31 |
| Gambar 26 Pilih bagian merah sebelah kanan, lalu klik pada "Publish to Github" .. | 32 |
| Gambar 27 Repository telah dibuat dan code sudah berhasil di push | 33 |
| Gambar 28 Pada message commit telah berubah dan code juga tentu telah berubah | 34 |
| Gambar 29 Pilih bagian Flutter: New Project..... | 35 |
| Gambar 30 Pilih yang empty..... | 36 |
| Gambar 31 Tunggu hingga proses selesai..... | 36 |
| Gambar 32 Tampilan setelah emulator dihidupkan | 38 |
| Gambar 33 Tunggu hingga proses selesai..... | 39 |
| Gambar 34 Gambar ketika aplikasi pertama dirun | 40 |



| | |
|---|----|
| Gambar 35 Ketika ditambah sebuah header | 40 |
| Gambar 36 Pilih bagian yang diberi tanda..... | 41 |
| Gambar 37 Setelah mencentang lalu klik tombol download. | 42 |
| Gambar 38 Tunggu hingga proses selesai. | 42 |
| Gambar 39 Tampilan awal dari Andorid Studio..... | 43 |
| Gambar 40 Pilih bagian New Project lalu pilih pada No Activity..... | 43 |
| Gambar 41 Tampilan pada bagian membuat aplikasi baru..... | 44 |
| Gambar 42 Tunggu hingga proses selesai. | 45 |
| Gambar 43 tampilan awal dari project..... | 45 |
| Gambar 44 Klik Pada Device Manager, lalu Create Device. | 46 |
| Gambar 45 Minimal menggunakan Oreo 26..... | 47 |



TUJUAN

Setelah menyelesaikan modul ini, praktikan diharapkan mampu :

1. Melakukan Instalasi Flutter
2. Memahai Cara Instalasi Vs. Code dan cara penggunannya
3. Memahami Cara Instalasi dan Penggunaan Git dan Github.
4. Memahami Cara Pembuatan Project Baru Flutter.

DASAR TEORI

A. Pengenalan Flutter

Flutter adalah sebuah framework open-source yang dikembangkan oleh google untuk membangun antarmuka pengguna (UI) yang kaya, indah dan responsive secara konsisten di berbagai platform, termasuk iOS, Android, Web, Desktop, dan lainnya. Flutter menggunakan bahasa pemrograman Dart, yang juga dikembangkan oleh Google, sebagai bahasa utama untuk mengembangkan aplikasi.

Berikut beberapa Komponen Utama dan fitur yang membangun dasar dari Flutter:

1. Dart : Flutter menggunakan Dart sebagai sebagai bahasa pemogramannya, dimana dart merupakan bahasa pemrograman yang efisien, dioptimalkan dengan UI, dan mudah dipelajar. Dart memungkinkan developer untuk mengembangkan aplikasi Flutter dengan mudah dan efisien.
2. Widget : Flutter dibangun dengan konsep “widget”. Widget adalah element dasar dari pembangunan UI dalam Flutter, dan mereka digunakan untuk membangun antar muka secara hierarkis. Flutter memiliki 2 jenis widget, yaitu “StatelessWidget” yang bersifat Statis dan tidak berubah serta “StatfullWidget” yang memiliki keadaan (State) dan berubah seiring waktu.
3. Hot Reload : Fitur ini memungkinkan pengembang untuk melihat perubahan yang mereka buat secara langsung dalam aplikasi



saat mengedit code. Hal ini dapat mempercepat proses pengembangan dan memungkinkan pengembang dapat melakukan eksperiment dan iterasi dengan waktu yang cepat.

4. Plugin,Paket : Flutter memiliki ekosistem yang kaya dengan berbagai plugin dan paket yang dapat diintregasikan dengan mudah ke dalam aplikasi. Ini memungkinkan pengembang untuk memperluas fungsionalitas aplikasi dengan cepat, seperti integrasi dengan database, API, kamera, GPS dan lainnya.



B. INSTALASI FLUTTER DI VS. CODE

Instalasi Flutter memerlukan beberapa syarat dan ketentuan, diantaranya:

1. Memiliki Sistem Operasi Windows 10 atau lebih (64 bit), berbasis x86-64.
2. Memiliki Space penyimpanan minimal 1,64GB (Tidak termasuk untuk IDE/tools).
3. Tools: Windows PowerShell dan Git (<https://git-scm.com/download/win>). **Tutorial instalasi Git ada pada halaman 18 (Bagian E) modul ini.**

Setelah syarat sudah dipenuhi, maka dapat dapat melakukan cara berikut:

- a. Mendownload SDK Flutter. Anda dapat mendownload nya melalui link <https://docs.flutter.dev/get-started/install/windows>. Klik pada tombol berwarna hijau dan tunggu hingga download selesai.

1. Download the following installation bundle to get the latest stable release of the Flutter SDK:

[flutter_windows_3.10.5-stable.zip](#)

Gambar 1 SDK Flutter

- b. Pada Partisi C, buat folder bernama src, Lalu ekstrak File Zip yang sudah di-download pada folder tersebut lalu tunggu hingga selesai.

* Warning: **Do not install Flutter** to a path that **contains special characters or spaces**. Pastikan lokasi dan folder tidak mengandung karakter spesial dan tidak memiliki spasi.

- Contoh benar: C:\user\src**flutterMain**.
- Contoh salah: C:\src**flutter main**.

*Warning: **Do not install Flutter** in a directory like **C:\ Program Files ** that requires elevated privileges.



- Tidak boleh mengistal Flutter pada folder/direktori yang memerlukan izin administrator.

| This PC > Windows-SSD (C:) > src | | | |
|----------------------------------|--------------------|-------------|------|
| Name | Date modified | Type | Size |
| flutter | 6/14/2023 10:44 PM | File folder | |

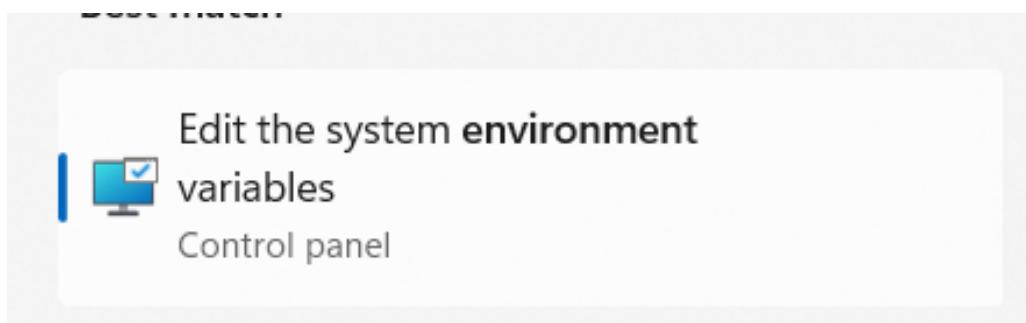
Gambar 2 Pastikan File yang diekstrak berada pada Foler C:\src

- Lalu masuk pada Folder tersebut, Lalu cari folder Bernama bin.
Lalu copy path folder tersebut.

C:\src\flutter\bin

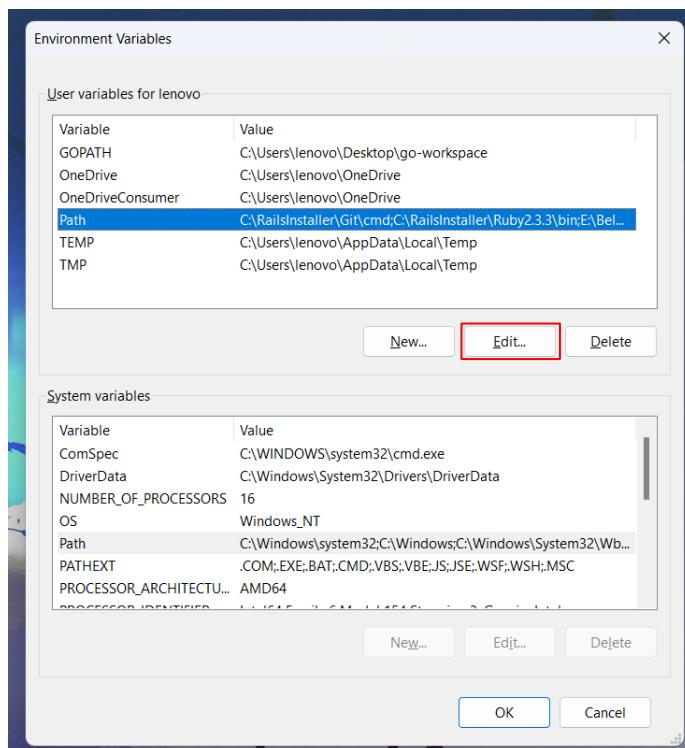
Gambar 3 Copy Path Bin untuk melakukan setting Environment.

- Lalu Update path Environment. Clik “**Windows**” pada keyboard atau klik tombol start, lalu ketik Environment.



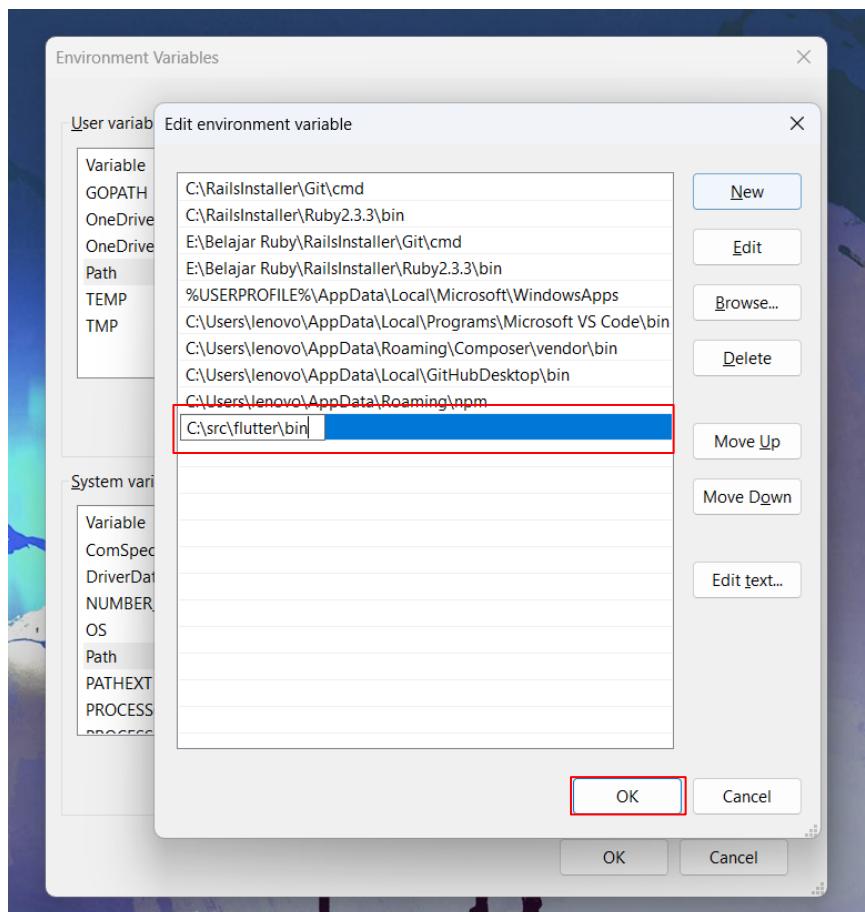


e. Akan muncul tampilan seperti ini, Pilih bagian path, lalu klik Edit.





- f. Lalu akan muncul tampilan seperti ini, Klik tombol new, lalu copy path yang telah disalin sebelumnya (Lokasi Bin pada folder Flutter) Lalu klik Ok dan keluar dari Environment.



Gambar 4. Paste path flutter bin yang sudah dicopy sebelumnya

- g. Flutter sudah berhasil di install. Namun ada beberapa syarat lagi yang harus dipenuhi. Untuk melihat syarat lainnya, bisa membuka command Prompt. Lalu ketik "**Flutter doctor**". Flutter Doctor akan melakukan pengecekan apa saya yang masih belum terpenuhi untuk menjalankan flutter.



```
C:\Users\lenovo>flutter doctor
Found an existing Pub cache at C:\Users\lenovo\AppData\Local\Pub\Cache.
It can be repaired by running 'dart pub cache repair'.
It can be reset by running 'dart pub cache clear'.
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.10.5, on Microsoft Windows [Version 10.0.22621.1848], locale en-US)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[!] Android toolchain - develop for Android devices (the doctor check crashed)
  X Due to an error, the doctor check did not complete. If the error message below is not helpful, please let us know
    about this issue at https://github.com/flutter/flutter/issues.
  X Exception: Android toolchain - develop for Android devices exceeded maximum allowed duration of 0:04:30.000000
[✓] Chrome - develop for the web
[X] Visual Studio - develop for Windows
  X Visual Studio not installed; this is necessary for Windows development.
    Download at https://visualstudio.microsoft.com/downloads/.
    Please install the "Desktop development with C++" workload, including all of its default components
[✓] Android Studio (version 2022.1)
[✓] VS Code (version 1.79.2)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 2 categories.
```

Gambar 5 Jalankan perintah "flutter doctor".

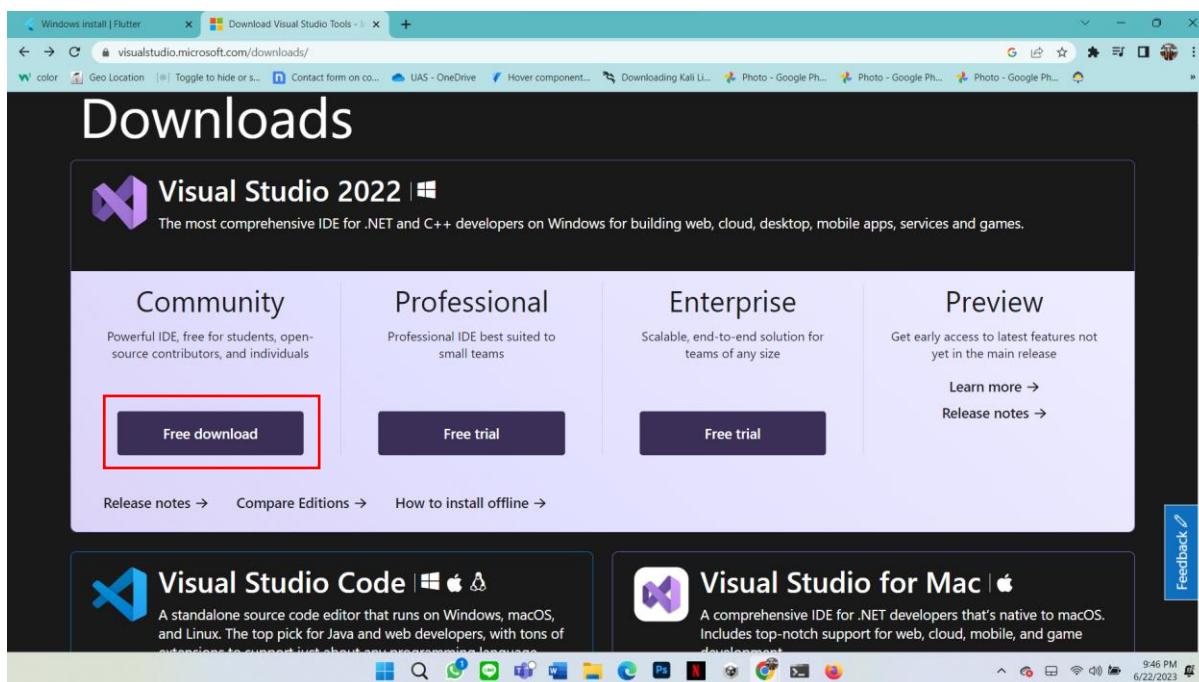
Pada device yang saya gunakan, saya harus melakukan beberapa hal seperti melakukan instalasi Visual Studio (Berbeda dengan Visual Studio Code), dan Android Tool Chain. Jika anda belum melakukan instalasi Code Editor (Visual Studio Code), maka pada VS Code juga akan memiliki tanda yang sama. Jadi perlu melakukan instalasi apa yang diperlukan. Namun pada dasarnya Instalasi Flutter sudah berhasil namun belum dapat digunakan seutuhnya.



C. INSTALASI VISUAL STUDIO

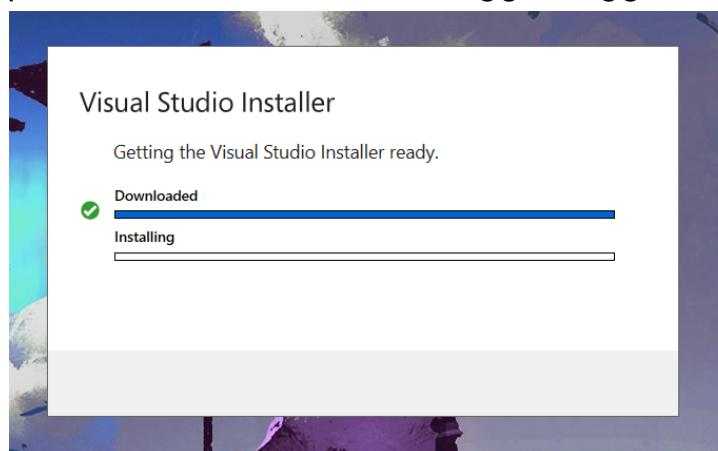
Seperti yang sudah diperlihatkan sebelumnya, Pada point Visual Studio masih terdapat ketidaklengkapan. Jadi kita perlu melakukan instalasi Visual Studio. Berikut Langkah-langkah dalam melakukan instalasi Visual Studio untuk kebutuhan Flutter.

1. Visual Studio dapat didownload pada link <https://visualstudio.microsoft.com/downloads/>. Lalu akan ada 3 pilihan, pilih bagian Community.



Gambar 6 Pilih Bagian Community, Lalu tunggu hingga download selesai.

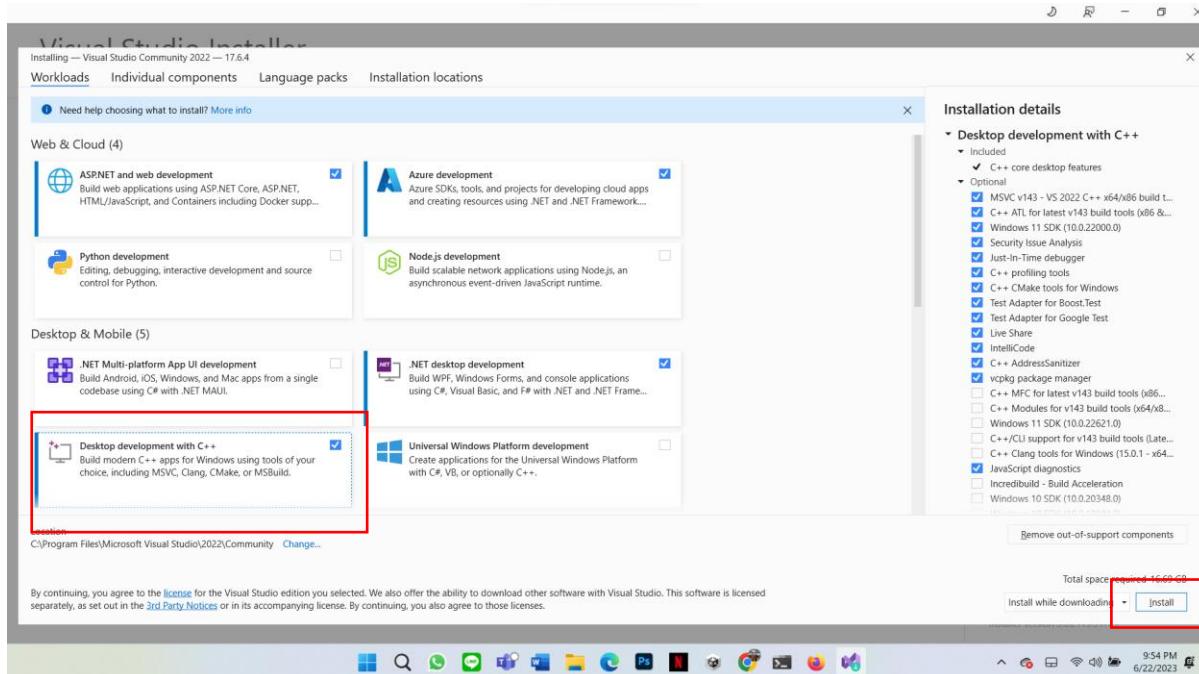
2. Lalu run Apk yang sudah berhasil di download tersebut. Lalu akan ada pop up dan klik "**Continue**" dan tunggu hingga selesai.



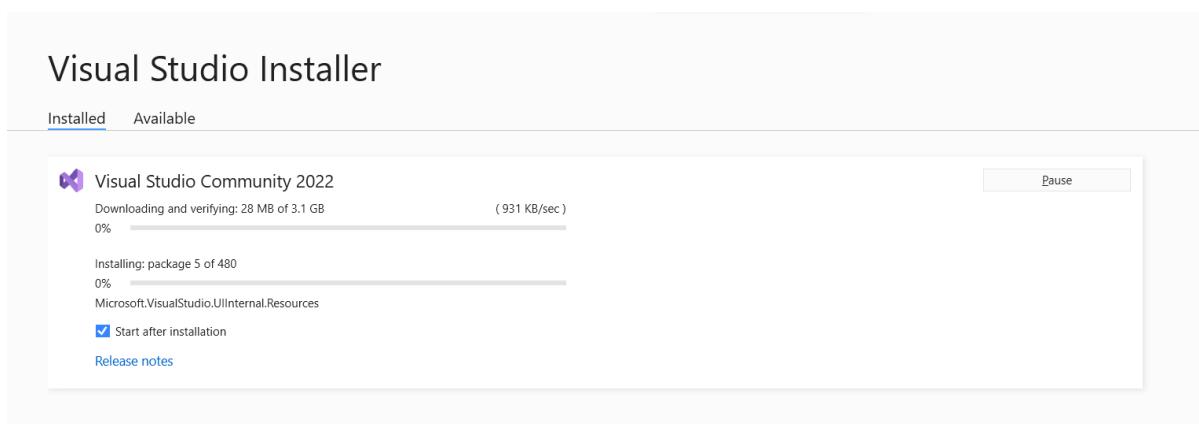
Gambar 7. Tunggu hingga proses Instalasi berhasil



3. Centang pada bagian Dekstop & Mobile -> Dekstop development With C++. Lalu Klik Install dan tunggu hingga proses selesai. Pastikan ketika proses install laptop tidak mati dan penyimpanan cukup.



Gambar 8 Centang pada bagian Desktop development with C++ Lalu install.



Gambar 9 Tunggu hingga Proses selesai

Tunggu hingga proses instalasi selesai. Jika sudah selesai, maka component pada visual studio yang dibutuhkan pada flutter sudah benar dan dapat dilakukan pengecekan Kembali dengan **flutter doctor**. Setelah Visual Studio diinstal (Componen Dekstop



development with C++), Maka issue pada flutter akan hilang dan flutter siap digunakan.

```
C:\Users\lenovo>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.10.5, on Microsoft Windows [Version 10.0.22621.1848], locale en-US)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 33.0.0)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop for Windows (Visual Studio Community 2022 17.6.4)
[✓] Android Studio (version 2022.1)
[✓] VS Code (version 1.79.2)
[✓] Connected device (3 available)
[✓] Network resources

• No issues found!

C:\Users\lenovo>
```

Gambar 10 Sudah tidak ada issue dan flutter siap digunakan.

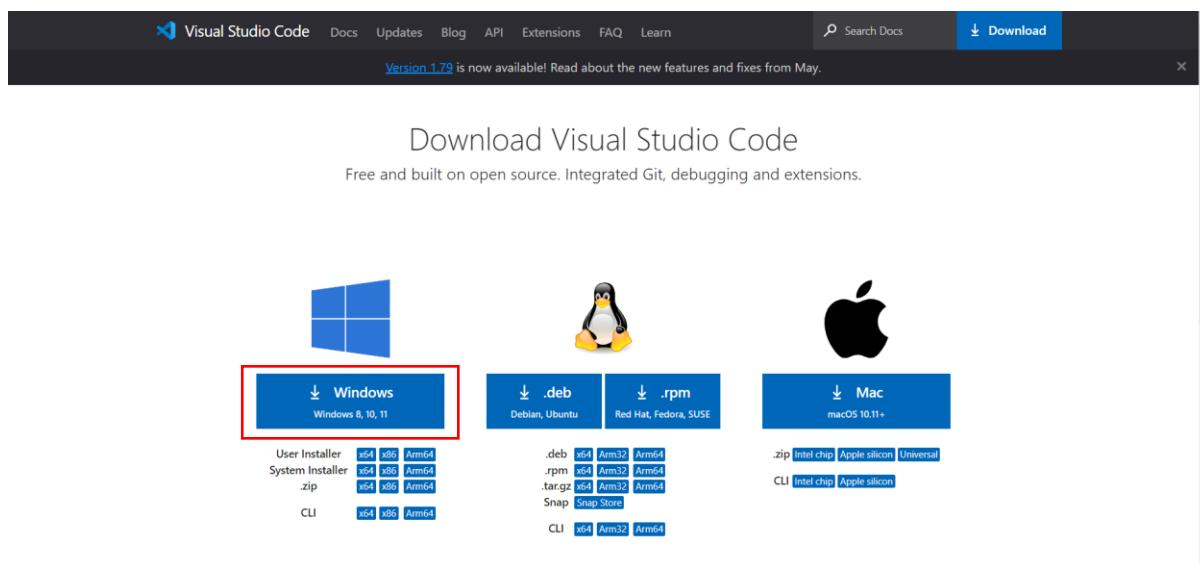


D. INSTALASI CODE EDITOR (VISUAL STUDIO CODE)

Code editor salah salah satu komponen penting ketika ingin melakukan coding. Code editor adalah sebuah *software* yang digunakan untuk menulis, mengedit, dan menformat pada sebuah komputer. Ada beberapa jenis code editor seperti Visual Studio Code, Sublime Text, Atom, Notepad++, dan PyCharm. Namun kita hanya akan menggunakan Visual Studio Code pada Mata Kuliah Pemograman Berbasis Platform (PBP) kali ini.

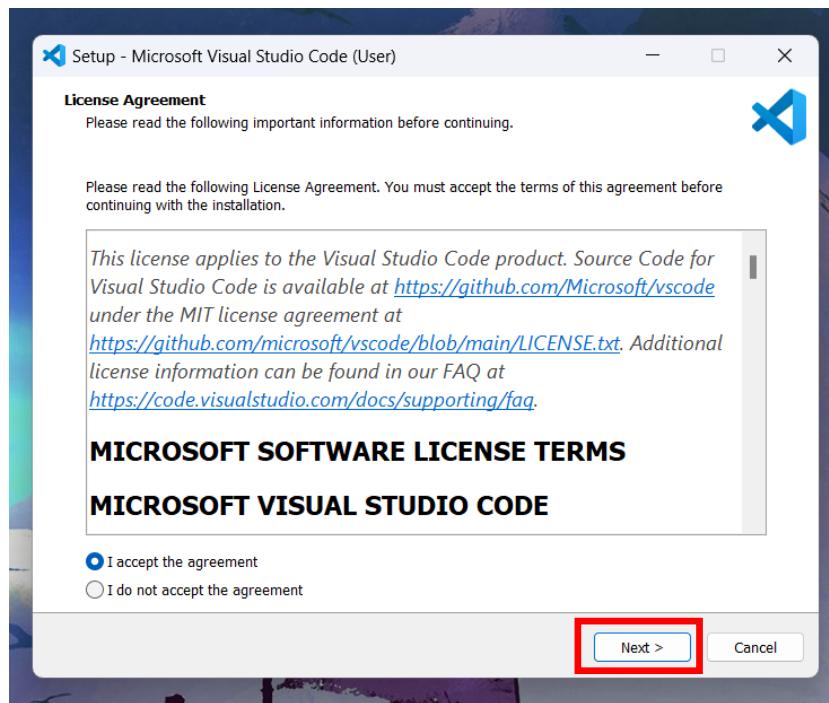
Berikut Langkah-langkah dalam melakukan instalasi Visual Studio Code:

1. Masuk pada laman <https://code.visualstudio.com/Download> dan pilih kategori berdasarkan OS yang digunakan. Karena saya menggunakan Windows, maka pilih windows. Lalu tunggu hingga download selesai.



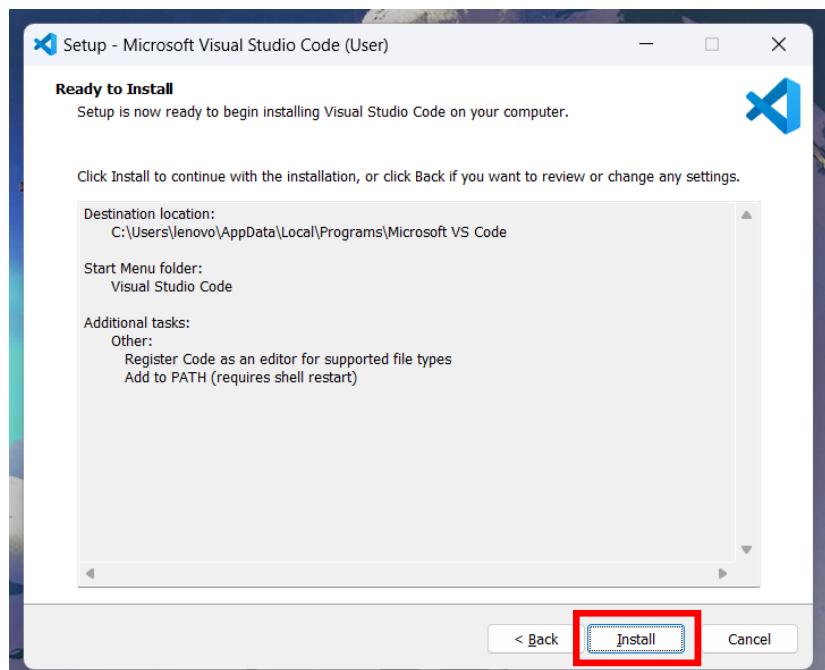
Gambar 11 Pilih instaler berdasarkan OS yang digunakan

2. Setelah selesai, Klik 2 kali pada hasil download untuk melakukan instalasi Vs. Code.
3. Pada bagian License Agreement, klik accept the agreement lalu next.



Gambar 12 Pilih bagian "I accept the agreement".

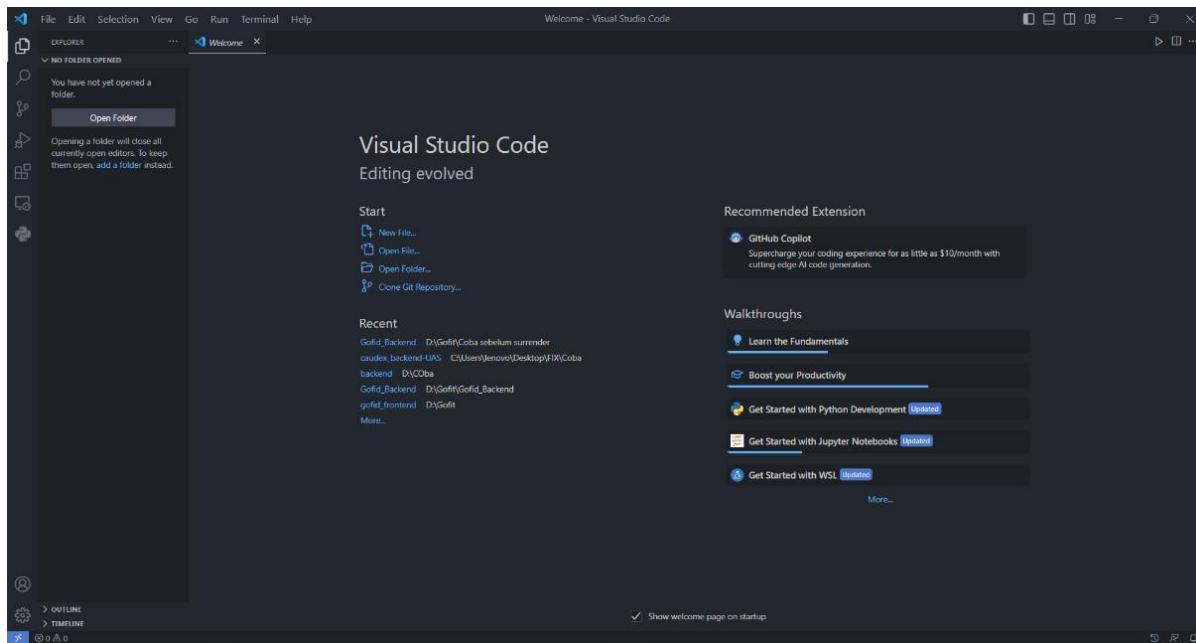
4. Pada bagian "Select Destination Location" tidak perlu diubah, langsung next saja.



Gambar 13 Klik Install dan tunggu hingga proses selesai.



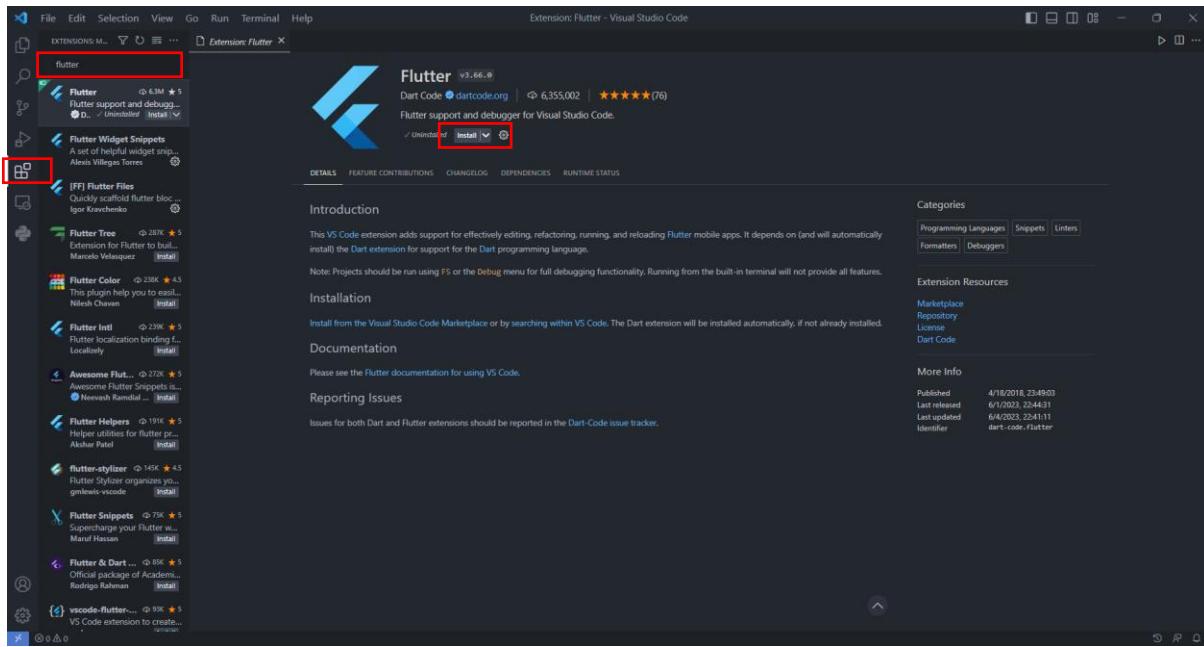
5. Next saja terus, hingga pada bagian “Ready to Install” klik Install dan tunggu hingga proses selesai. Setelah proses instalasi selesai, klik finish, dan Vs. Code akan di-run secara otomatis.



Gambar 14 Tampilan Awal dari Vs. Code

Ini adalah tampilan dari Vs. Code. Anda bisa melakukan eksplorasi sendiri. Baik dari merubah tema, memilih ekstension dan lainnya.

6. Anda Bisa menambahkan ekstension khusus flutter dengan cara masuk pada bagian Extension (Ctrl + Shift + X), lalu search flutter dan klik install. Setelah itu reload Vs. Code dengan cara tutup lalu buka Kembali, dan ekstension sudah dapat digunakan.



Gambar 15 Cara menambahkan ekstension Flutter pada Vs. Code.



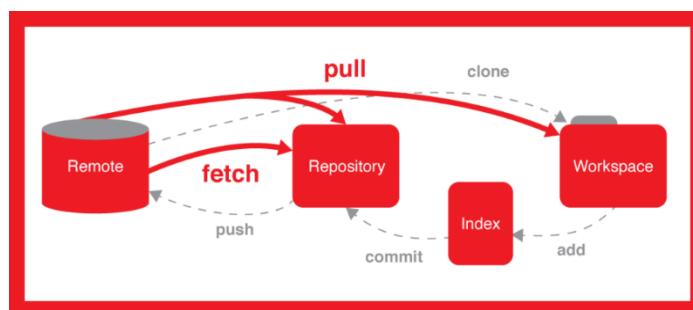
E. INSTALASI GIT DAN GITHUB

Version Control System adalah sistem yang mengelola suatu perubahan pada file dokumen, source code, atau kumpulan informasi lainnya. VCS mencatat setiap perubahan pada file yang dikerjakan oleh seseorang. Hampir semua detail perubahan dicatat, seperti: Id Perubahan, Identitas seseorang yang melakukan perubahan, waktu perubahan, judul perubahan, dan deskripsi perubahan. Biasanya hal tersebut kita temui saat kita membuat dokumen, di dalam Microsoft Word kita bisa menemukan hal yang serupa dengan Version Control System.

Selain sebagai system yang mengelola riwayat perubahan, system ini juga berguna untuk mempermudah pengembangan sebuah program secara kolaboratif atau berkelompok. Jadi dengan system VCS kita bisa mengerjakan satu berkas secara bersamaan dengan beberapa orang, tanpa menunggu satu persatu selesai dan bergantian mengerjakan.

Dan yang terakhir, Version Control System juga memungkinkan kita untuk kembali ke perubahan sebelumnya. Pada intinya, version control system bisa melakukan tugas seperti melihat riwayat perubahan, mengelola setiap perubahan (commit, push, checkout, pull), menuntaskan bagian-bagian berkas secara langsung tanpa menunggu penggerjaan yang lain selesai, dan memungkinkan kita kembali ke perubahan sebelumnya. Git termasuk dalam jenis Version Control System, dan Github adalah layanan Version Control System berbasis git secara online.

1. Git



Git adalah sebuah perangkat lunak berbasis Version Control System, dengan git kita mampu mengelola sebuah proyek atau berkas secara terdistribusi, karena perubahan yang tercatat oleh git disimpan didalam database



git dan database git tersebut tidak hanya berada pada satu tempat, namun terdapat di berbagai tempat. Artinya bahwa semua orang yang berkontribusi pada proyek tersebut bisa menyimpan database perubahannya, sehingga akan memudahkan dalam mengelola proyek baik online maupun offline.

Dengan git kita bisa melakukan penggabungan beberapa berkas atau bagian pada berkas kedalam satu proyek atau berkas final, hal ini sangat mempermudah developer ketika melakukan pengembangan aplikasi secara berkolaborasi.

2. Github

Jika dengan git kita hanya mampu mengelola setiap perubahan secara lokal, maka dengan github kita bisa melakukannya secara online. Github adalah layanan git berbasis cloud, selain itu dengan github kita dapat dengan mudah bekerja dengan git tanpa melakukannya dengan baris perintah (CLI), karena github merupakan layanan git berbasis GUI002E.

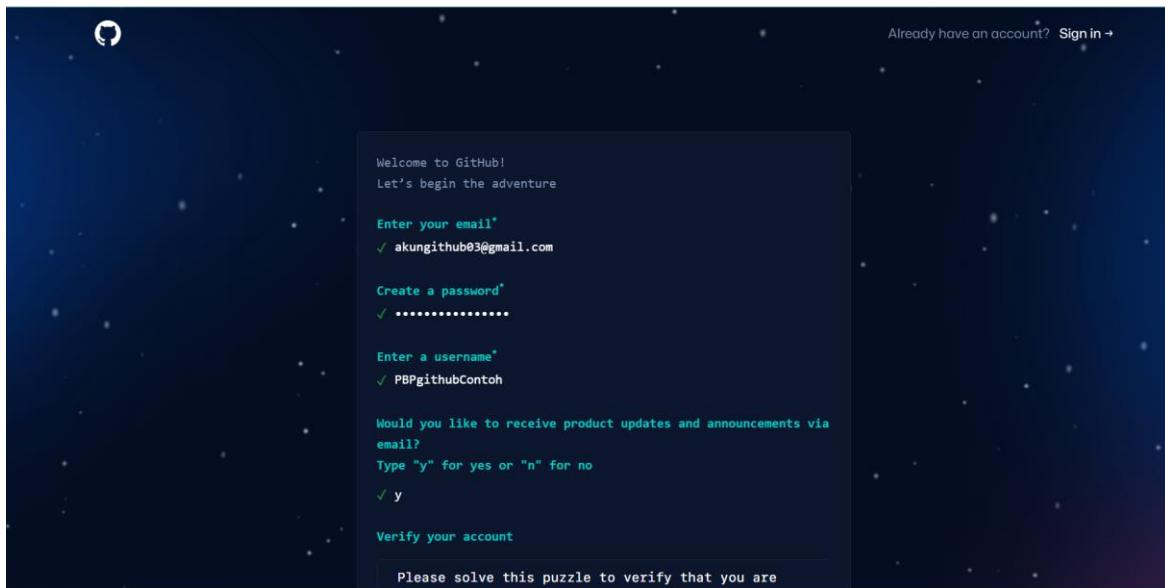
3. Membuat Akun Github

Berikut Langkah-langkah dalam membuat akun github

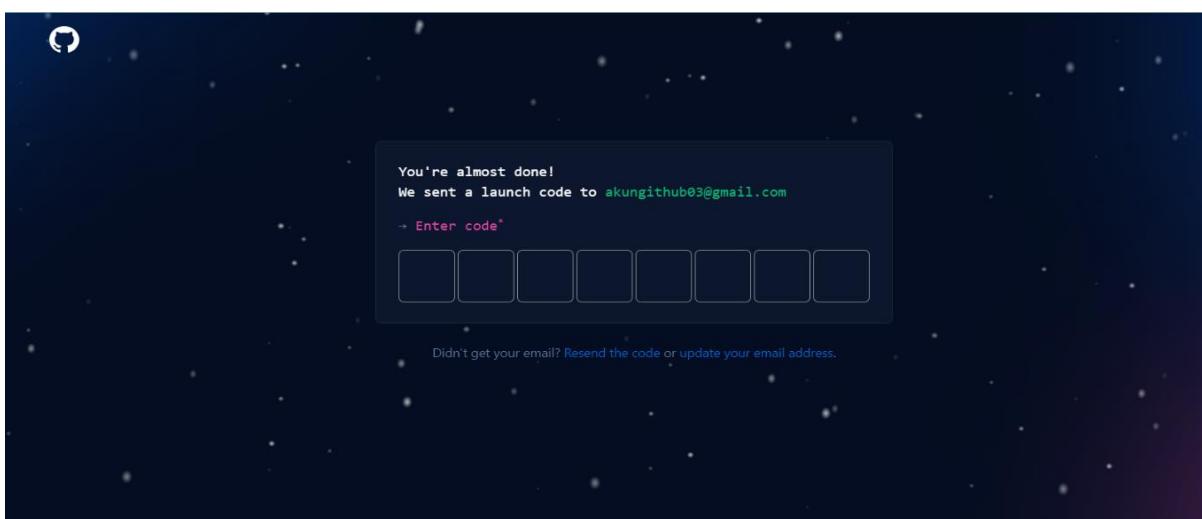
- a. Masuk pada situs resmi github <https://github.com/>
- b. Pilih "Sign Up" pada bagian kanan atas



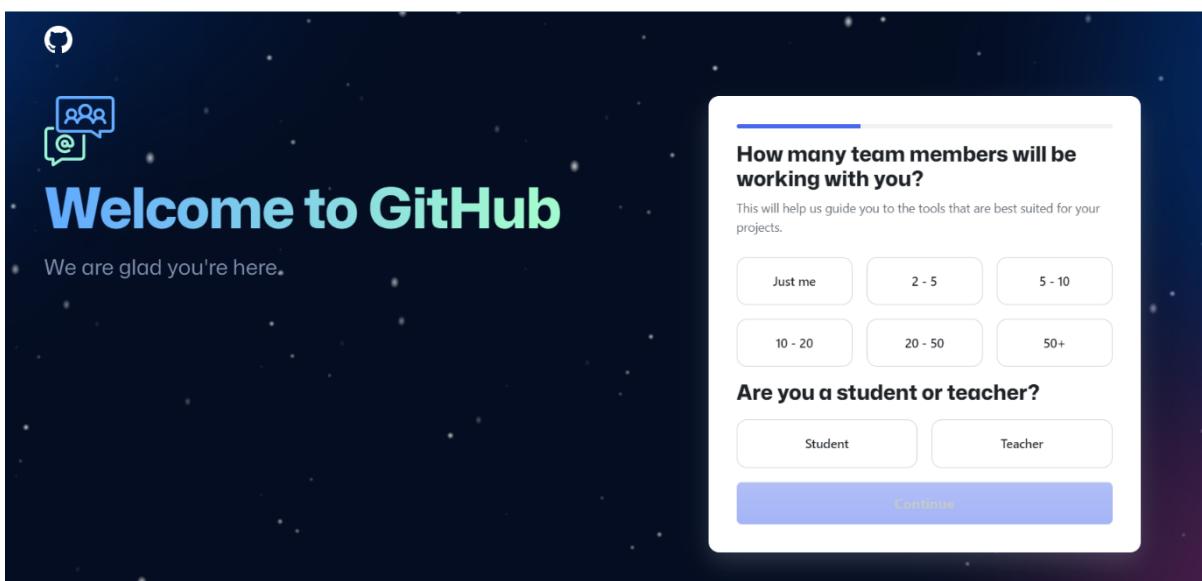
- c. Lengkapi form pendaftaran yang dibutuhkan seperti email, password, dan username.

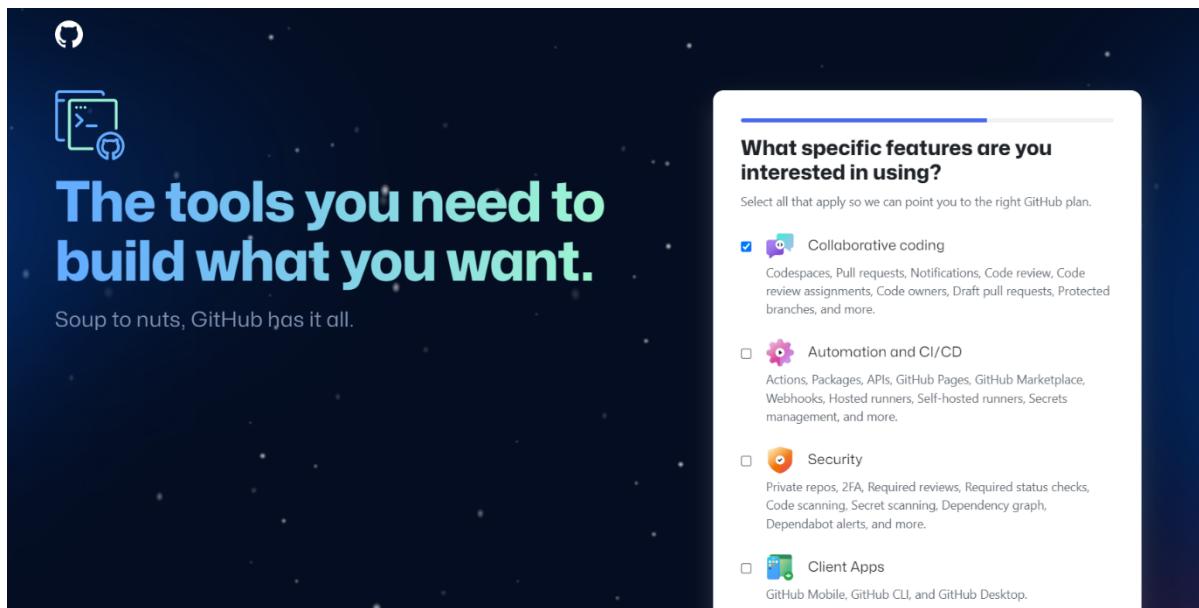


- d. Masukkan Code verifikasi yang telah dikirim pada email yang telah didaftarkan.

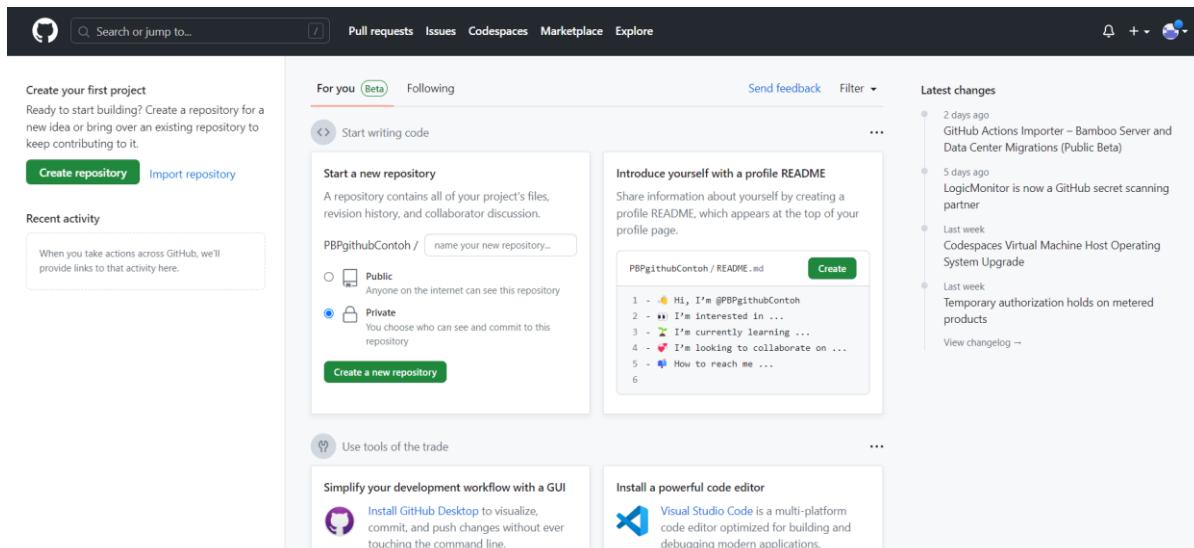


- e. Lengkapi beberapa pertanyaan seperti berikut.





- f. Jika berhasil mendaftar, maka akan masuk pada halaman utama dashboard dari github seperti berikut



Gambar 16 Halaman Utama Github

Setelah sampai pada halaman ini, maka akun github anda sudah berhasil didaftar dan sudah dapat digunakan. Anda dapat mempelajari lebih lanjut dengan mencari referensi pada media lain seperti google, youtube dan lainnya.

2. Instalasi Git

Dalam instalasi git, diperlukan beberapa persyaratan system yang harus dipenuhi, seperti

- Perangkat dengan OS minimal Windows 7 atau setara



b. Web browser (Firefox, Chrome, Edge, dll)

Berikut Langkah-langkah dalam membuat akun github

- Unduh git pada <https://git-scm.com/>

- Pilih OS sesuai yang digunakan, Karena saya sedang menggunakan windows, maka saya pilih windows.

The screenshot shows the official Git website at <https://git-scm.com/>. The main navigation bar includes links for About, Documentation, Downloads (selected), and Community. The Downloads section features three download options: macOS, Windows (highlighted with an orange border), and Linux/Unix. To the right, there's a large image of a computer monitor displaying the latest source release version 2.40.1. Below the monitor, sections for GUI Clients and Logos are shown, along with a link to 'View Logos →'. At the bottom, there's a section titled 'Git via Git' with instructions and a command-line example: `git clone https://github.com/git/git`.

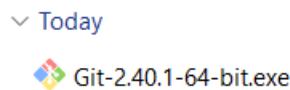
Gambar 17 Pilih Windows

- Lalu Pilih sesuai dengan arsitektur perangkat yang digunakan. Jika menggunakan 64 bit, pilih 64, jika menggunakan 32, pilih 32. Lalu tunggu hingga download selesai.

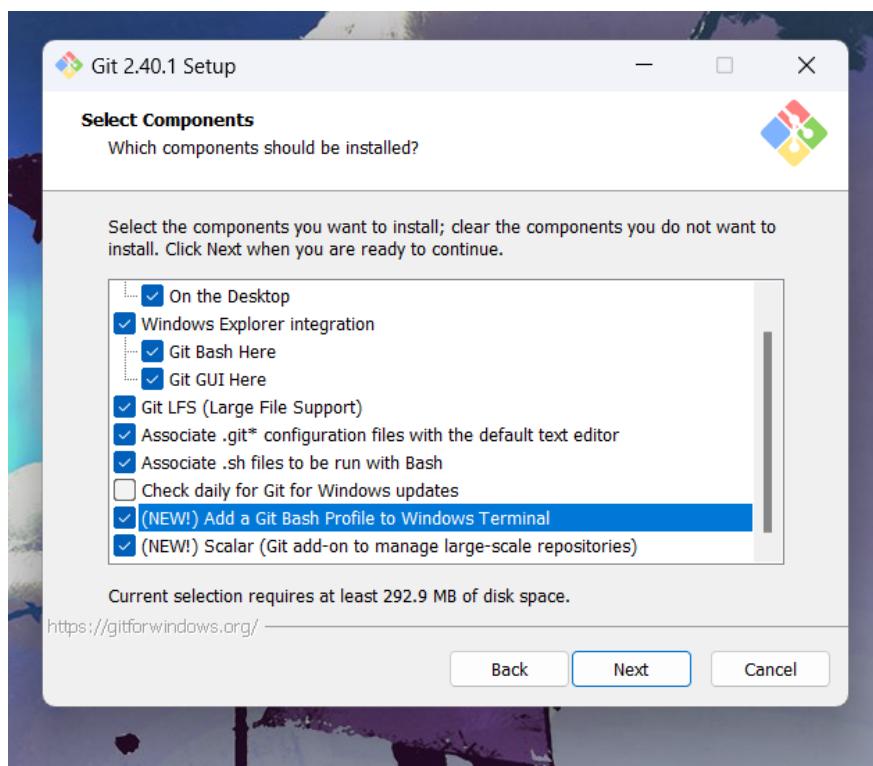
This screenshot shows the 'Download for Windows' page from the Git website. It features a large button labeled 'Click here to download' for the latest 64-bit version. Below the button, there are links for other Windows downloads like the Standalone Installer and 32-bit versions. A 'Using winget tool' section provides instructions for installing via the Windows Package Manager. At the bottom, a 'Now What?' section tells users to start using Git after download.



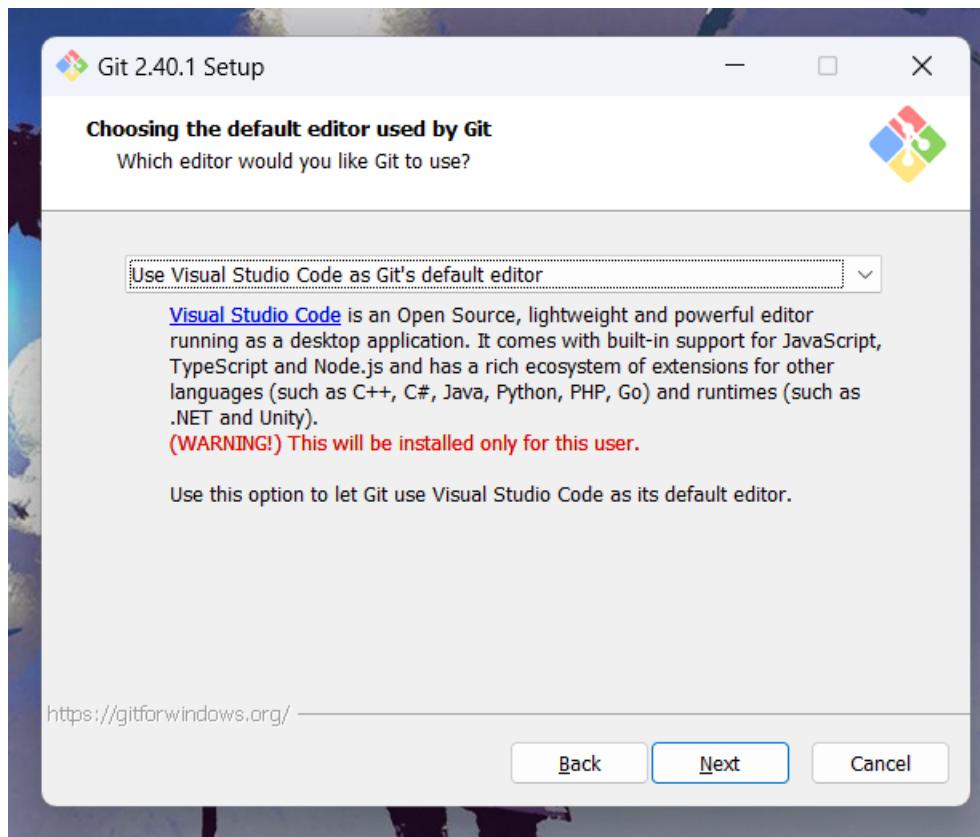
- d. Setelah selesai mengunduh, klik 2x pada file Git yang telah terunduh.



- e. Pada Lisensi GNU, Next saja
f. Pada saat menentukan lokasi instalasi, usahakan tidak memindahkan lokasi. Langsung next saja.
g. Pada pemilihan komponen, Atur sesuai dengan gambar dibawah, lalu next.



- h. Pilih editor teks sebagai default untuk menggunakan git (Karena hanya menggunakan Vs.Code , maka pilih VS.code saja).



- i. Untuk selanjutkan, Next Saja karena kita hanya menggunakan VSCode untuk upload repository. Tidak usah diubah settingannya dan biarkan saja default.
- j. Untuk finally, klik install saja jika sudah sampai di final setting, tunggu hingga selesai. Jika sudah selesai, maka gitu sudah berhasil di install.
- k. Jika sudah selesai melalukan instalasi, maka bisa ke Command Prompt atau sejenisnya. Ketik git --version untuk mengecek apakah instalasi sudah berhasil atau tidak. Jika berhasil, maka akan muncul seperti ini.

```
C:\WINDOWS\system32\cmd. x + ^

Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lenovo>git --version
git version 2.40.1.windows.1

C:\Users\lenovo>
```



- I. Karena di PC/Laptop, github tidak mengenali device anda, maka anda harus melakukan setting agar device anda gunakan untuk version control system. Oleh karena itu, anda dapat melakukan cara berikut pada Command Prompt atau terminal
 1. git config --global user.name <nama username Github anda>
 2. git config --global user.email <email yang didaftarkan pada github>
 3. git config -list (untuk melihat apakah sudah terdaftar atau tidak username dan email anda)

Ket: Hapus tanda <>
- m. Hasilnya akan seperti ini

```
C:\Users\lenovo>git config --global user.name PBPgithubContoh
C:\Users\lenovo>git config --global user.email akungithub03@gmail.com
C:\Users\lenovo>git config --list
user.name=PBPgithubContoh
user.email=akungithub03@gmail.com
core.autocrlf=false
core.editor="C:\Users\lenovo\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait
C:\Users\lenovo>
```

Gambar 18 Jika sudah tampil seperti ini, maka akun git sudah berhasil dibuat dan dapat digunakan



F. BEKERJA DENGAN GIT DAN GITHUB

Persiapan:

1. Device yang sudah terinstall git
2. Telah memiliki akun Github
3. Memiliki Koneksi Internet
4. Vs. Code
5. Program atau code yang akan di push pada github

Langkah-Langkah:

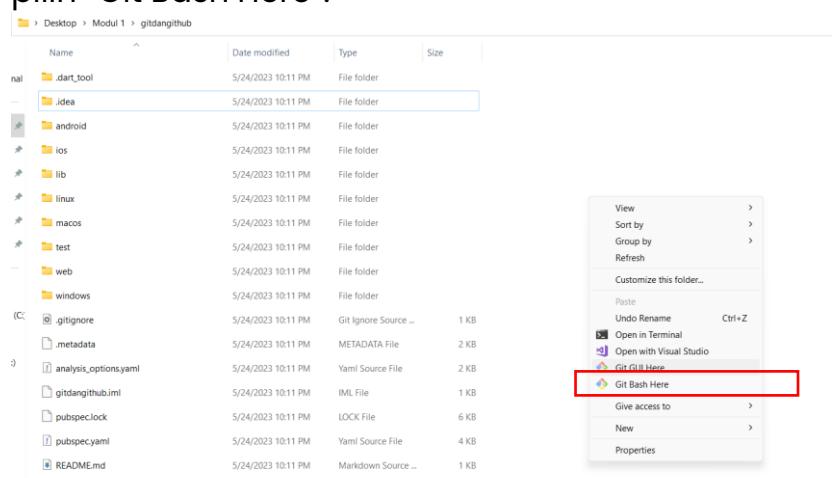
1. Siapkan folder yang akan dipush pada github



gitdangithub

Gambar 19 Folder Yang akan dipush pada github

2. Masuk pada file tersebut, lalu klik kanan di dalam file tersebut dan pilih "Git Bash Here".



Gambar 20 Pilih Git Base Here

Setelah itu maka akan tampil sebuah command prompt seperti berikut ini



| Name | Date modified | Type | Size |
|-----------------------|--------------------|----------------------|------|
| .dart_tool | 5/24/2023 10:11 PM | File folder | |
| .idea | 5/24/2023 10:11 PM | File folder | |
| android | 5/24/2023 10:11 PM | File folder | |
| ios | 5/24/2023 10:11 PM | File folder | |
| lib | 5/24/2023 10:11 PM | File folder | |
| linux | 5/24/2023 10:11 PM | File folder | |
| macos | 5/24/2023 10:11 PM | File folder | |
| test | 5/24/2023 10:11 PM | File folder | |
| web | 5/24/2023 10:11 PM | File folder | |
| windows | 5/24/2023 10:11 PM | File folder | |
| .gitignore | 5/24/2023 10:11 PM | Text File | 1 KB |
| .metadata | 5/24/2023 10:11 PM | Text File | 1 KB |
| analysis_options.yaml | 5/24/2023 10:11 PM | Yaml Source File | 4 KB |
| gitdangithub.iml | 5/24/2023 10:11 PM | Text File | 1 KB |
| pubspec.lock | 5/24/2023 10:11 PM | LOCK File | 6 KB |
| pubspec.yaml | 5/24/2023 10:11 PM | Yaml Source File | 4 KB |
| README.md | 5/24/2023 10:11 PM | Markdown Source File | 1 KB |

3. Ketikkan pada command prompt

-git init

-git status

```
lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub
$ git init
Initialized empty Git repository in C:/Users/lenovo/Desktop/Modul 1/gitdangithub
/.git

lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub (master)
$ git status
On branch master

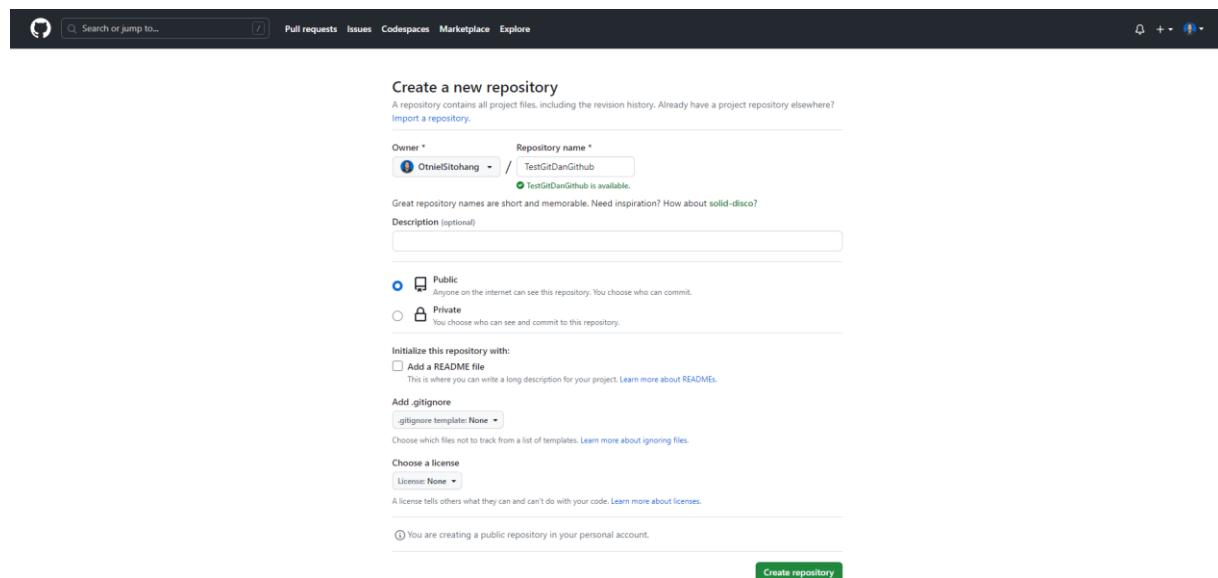
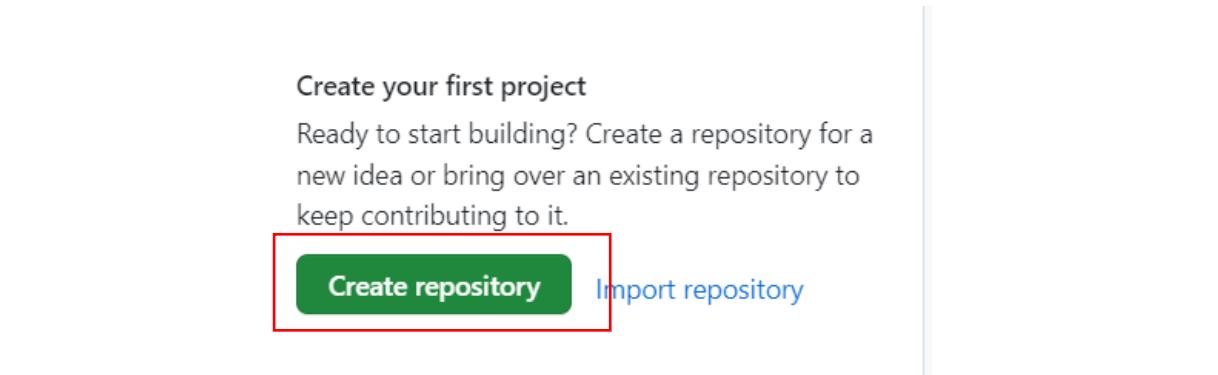
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    .metadata
    README.md
    analysis_options.yaml
    android/
    ios/
    lib/
    linux/
    macos/
    pubspec.lock
    pubspec.yaml
    test/
    web/
    windows/

nothing added to commit but untracked files present (use "git add" to track)

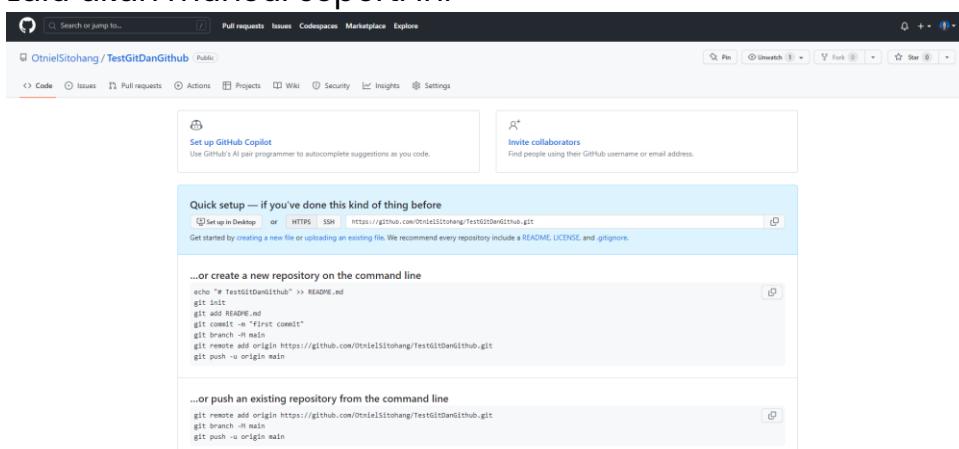
lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub (master)
$
```

4. Lalu buat repository baru pada github



Gambar 21 Lalu Masukkan nama Repository yang diinginkan, untuk jenis dari repository nya tergantung kebutuhan, boleh di public atau di private, lalu klik "Create repository".

5. Lalu akan muncul seperti ini



6. Lalu masuk Kembali pada git bash, setelah melakukan init dan status serta membuat repository, lalu masukkan code pada baris 1 dan 2 yang muncul pada github secara berurutan pada git base.



...or push an existing repository from the command line

```
git remote add origin https://github.com/OtnielSitohang/TestGitDanGithub.git  
git branch -M main  
git push -u origin main
```

```
lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub  
$ git init  
Initialized empty Git repository in C:/Users/lenovo/Desktop/Modul 1/gitdangithub  
.git/  
  
lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub (master)  
$ git status  
On branch master  
  
No commits yet  
  
Untracked files:  
(use "git add <file>..." to include in what will be committed)  
.gitignore  
.metadata  
README.md  
analysis_options.yaml  
android/  
ios/  
lib/  
linux/  
macos/  
pubspec.lock  
pubspec.yaml  
test/  
web/  
windows/  
  
nothing added to commit but untracked files present (use "git add" to track)  
  
lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub (master)  
$ git remote add origin https://github.com/OtnielSitohang/TestGitDanGithub.git  
  
lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub (master)  
$ git branch -M main  
  
lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub (main)  
$ |
```

Gambar 22 Code baris 1 dan 2 setelah dimasukkan pada git base

7. Setelah itu, masukkan code

- git add .
- git commit -m "Kata commit yang ingin dimasukkan"
- code baris ke 3



```
lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub (main)
$ git add .

lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub (main)
$ git commit -m "first commit"
[main (root-commit) 230a4ad] first commit
128 files changed, 4873 insertions(+)
create mode 100644 .gitignore
create mode 100644 .metadata
create mode 100644 README.md
create mode 100644 analysis_options.yaml
create mode 100644 android/.gitignore
create mode 100644 android/app/build.gradle
create mode 100644 android/app/src/debug/AndroidManifest.xml
create mode 100644 android/app/src/main/AndroidManifest.xml
create mode 100644 android/app/src/main/kotlin/com/example/gitdangithub/MainActivity.kt
create mode 100644 android/app/src/main/res/drawable-v21/launch_background.xml
create mode 100644 android/app/src/main/res/drawable/launch_background.xml
create mode 100644 android/app/src/main/res/mipmap-hdpi/ic_launcher.png
create mode 100644 android/app/src/main/res/mipmap-mdpi/ic_launcher.png
create mode 100644 android/app/src/main/res/mipmap-xhdpi/ic_launcher.png
create mode 100644 android/app/src/main/res/mipmap-xxhdpi/ic_launcher.png
create mode 100644 android/app/src/main/res/values-night/styles.xml
create mode 100644 android/app/src/main/res/values/styles.xml
create mode 100644 android/app/src/profile/AndroidManifest.xml
create mode 100644 android/build.gradle
```

Gambar 23 Git add. dan git commit

```
lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub (main)
$ git push -u origin main
Enumerating objects: 177, done.
Counting objects: 100% (177/177), done.
Delta compression using up to 16 threads
Compressing objects: 100% (146/146), done.
Writing objects: 100% (177/177), 266.92 KiB | 8.61 MiB/s, done.
Total 177 (delta 20), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (20/20), done.
To https://github.com/OtnielSitohang/TestGitDanGithub.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```



```
lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub (main)
$ |
```

Gambar 24 Code pada baris ke 3

Code sudah berhasil di push pada repository yang sudah dibuat. Lalu pada repository pada github direfresh dan tampilannya akan seperti ini dan push sudah berhasil.



The screenshot shows a GitHub repository named 'OtnielSitohang/TestGitDanGitHub'. The repository has 1 branch and 1 commit. The commit message is 'PPBgithubContoh first commit'. The commit was made 2 minutes ago by 'OtnielSitohang' with 1 commit. The commit details show a list of files: android, ios, lib, linux, macos, test, web, windows, .gitignore, .metadata, README.md, analysis_options.yaml, pubspec.lock, and pubspec.yaml. All files are listed as 'first commit' and were pushed 2 minutes ago. The repository has 0 stars, 1 watching, and 0 forks. It also includes sections for Releases, Packages, and Languages (C++ 43.4%, CMake 35.4%, Dart 11.4%, HTML 3.5%, Swift 3.3%, C 2.7%, Other 0.3%).

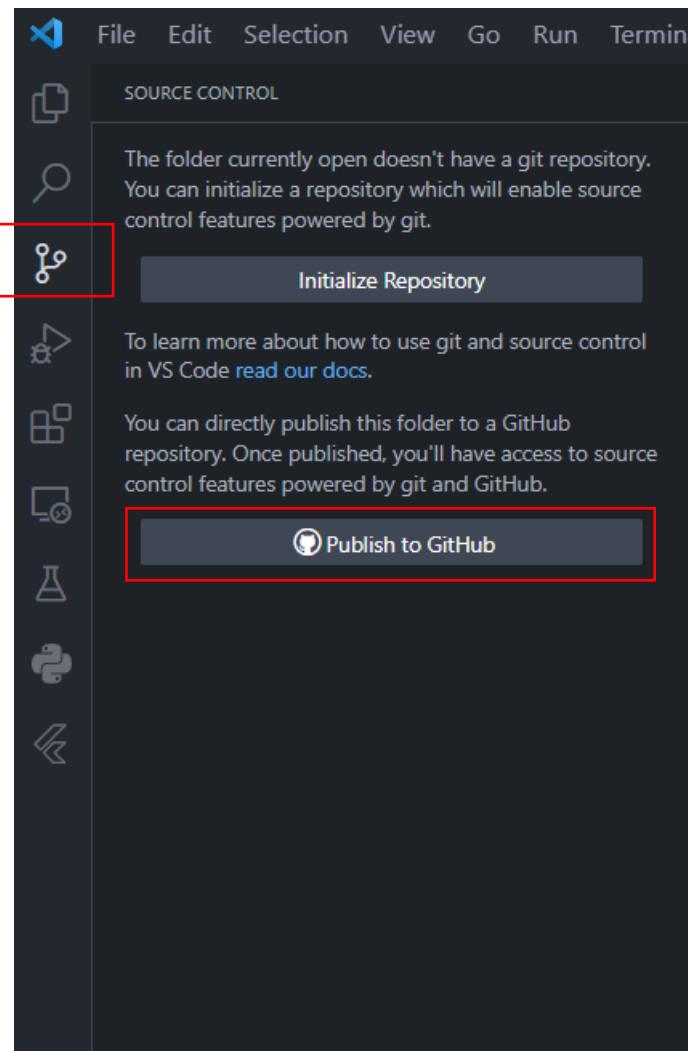
Gambar 25 Code sudah berhasil di push pada github menggunakan Git

G. PUSH CODE DENGAN CODE EDITOR (VS. CODE)

1. Buka folder yang ingin dipush pada vs code

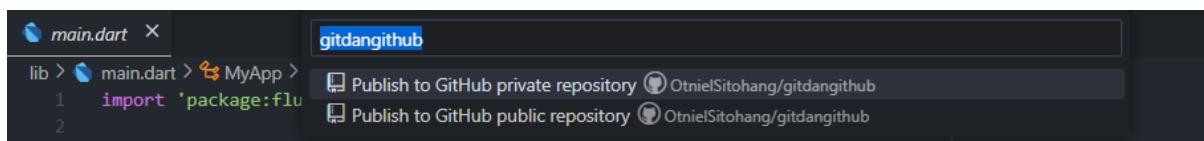
The screenshot shows the Visual Studio Code interface with a Flutter project named 'GITDANGITHUB'. The 'main.dart' file is open in the editor. The code defines a 'MyApp' widget that extends 'StatelessWidget'. It imports 'package:flutter/material.dart' and runs the application with 'runApp(const MyApp())'. The 'MyApp' class sets the theme to 'ThemeData'. The 'main' function is annotated with 'Run|Debug Profile'. The project structure in the Explorer sidebar includes 'lib' (containing 'main.dart'), 'android', 'ios', 'test', 'web', 'windows', '.gitignore', '.metadata', 'analysis_options.yaml', 'gitdangithub.iml', 'pubspec.lock', 'pubspec.yaml', and 'README.md'. The status bar at the bottom shows 'Ln 16 Col 50 Spaces: 2 UTF-8 CRLF (1) Dart Windows (Windows-10) ✓ Spell ⌘ Preferences'.

2. Masuk pada bagian push to github. Jika baru pertama kali membuka folder dan belum pernah di push atau bukan folder clone maka akan tampil tampilan seperti ini.

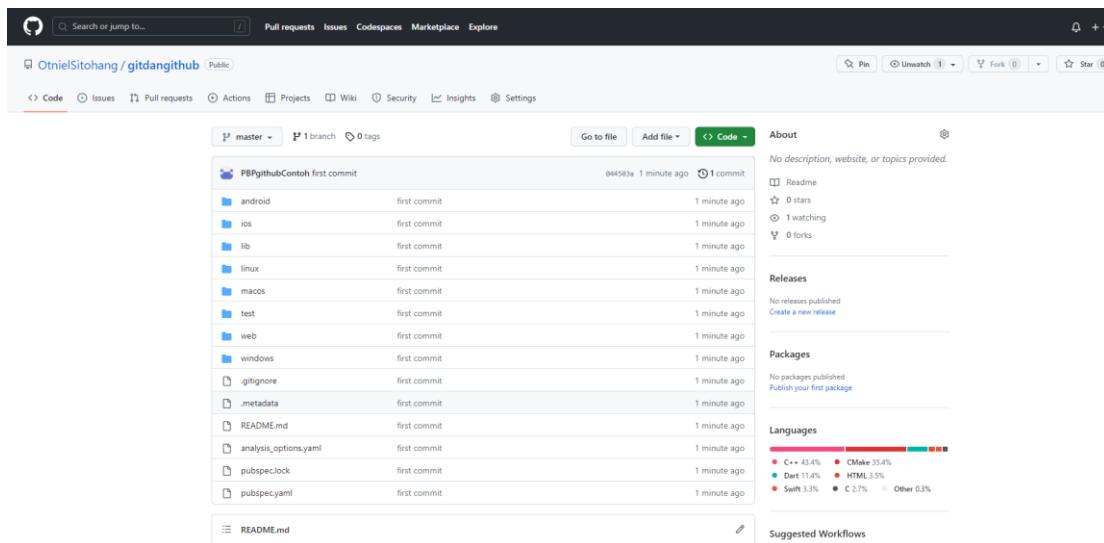


Gambar 26 Pilih bagian merah sebelah kanan, lalu klik pada "Publish to Github"

3. Setelah itu, akan muncul tampilan seperti ini. Bagian berwarna biru adalah nama repo akan akan otomatis di buat nantinya. Anda dapat memilih jenis repository yang akan dibuat (Berupa public maupun private) sesuai dengan kebutuhan anda.

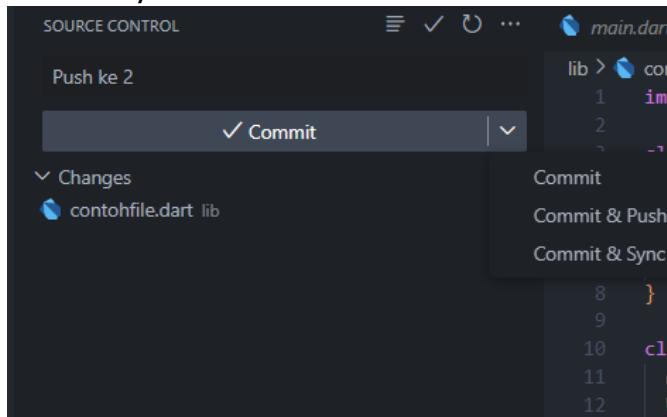


4. Maka repository akan otomatis di buat dan code berhasil dipush pada github. Untuk penamaan repository tidak boleh ada spasi dan harus menggunakan huruf kecil.



Gambar 27 Repository telah dibuat dan code sudah berhasil di push

5. Jika terdapat perubahan atau penambahan code, anda dapat melalukan hal yang sama, namun pada source control tidak perlu memasukkan nama repository karena repository telah ada. Anda hanya perlu memasukkan message commit lalu commit dan sync.



Sync berfungsi untuk melakukan sinkronisasi jika ada orang lain yang melakukan push sebelum anda. Jadi code perlu disinkronisasikan agar tidak ada code yang bertabrakan. Setelah itu maka pada repository juga akan berubah



| PBPgithubContoh Push ke 2 | | |
|---------------------------|--------------|-----------------------|
| | | 5154ec8 now 2 commits |
| android | first commit | 5 minutes ago |
| ios | first commit | 5 minutes ago |
| lib | Push ke 2 | now |
| linux | first commit | 5 minutes ago |
| macos | first commit | 5 minutes ago |
| test | first commit | 5 minutes ago |
| web | first commit | 5 minutes ago |
| windows | first commit | 5 minutes ago |
| .gitignore | first commit | 5 minutes ago |
| .metadata | first commit | 5 minutes ago |
| README.md | first commit | 5 minutes ago |
| analysis_options.yaml | first commit | 5 minutes ago |
| pubspec.lock | first commit | 5 minutes ago |
| pubspec.yaml | first commit | 5 minutes ago |

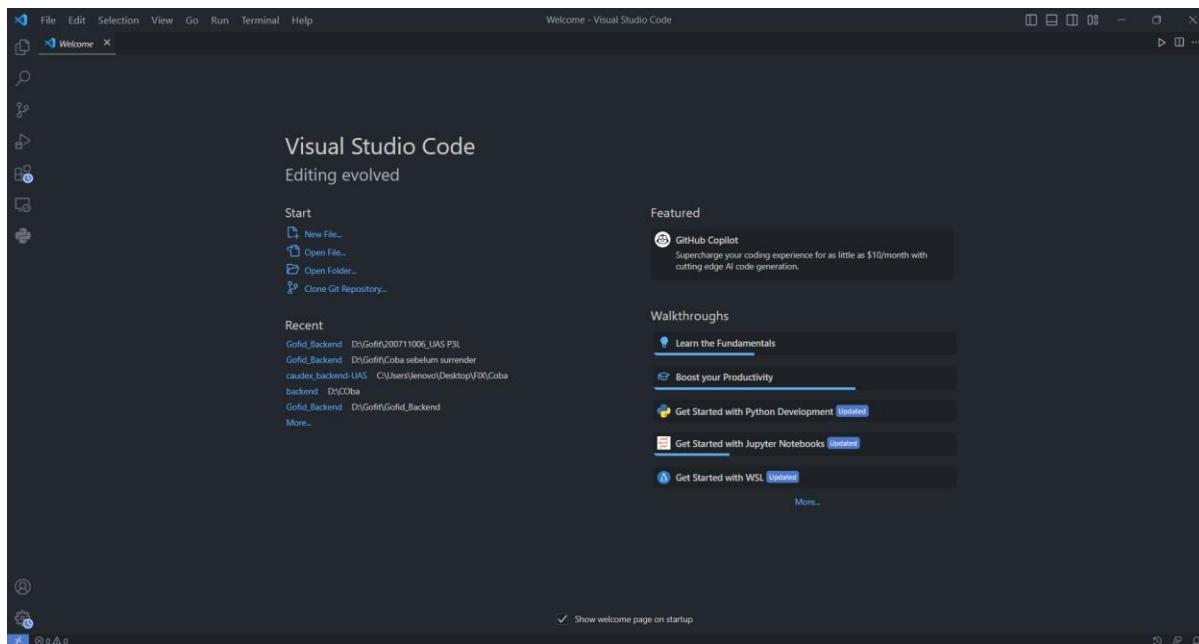
Gambar 28 Pada message commit telah berubah dan code juga tentu telah berubah



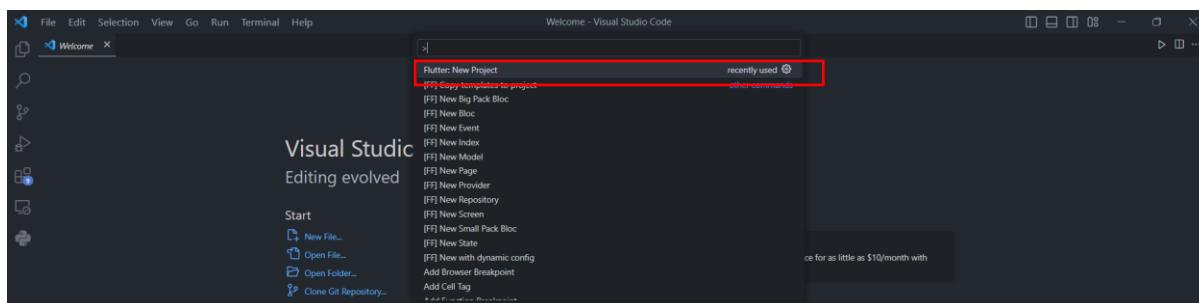
H. MEMBUAT PROGRAM HELLO WORDL SEDERHANA

Berikut Langkah-langkah dalam membuat program baru menggunakan framework flutter

1. Buka Aplikasi Visual Studio Code

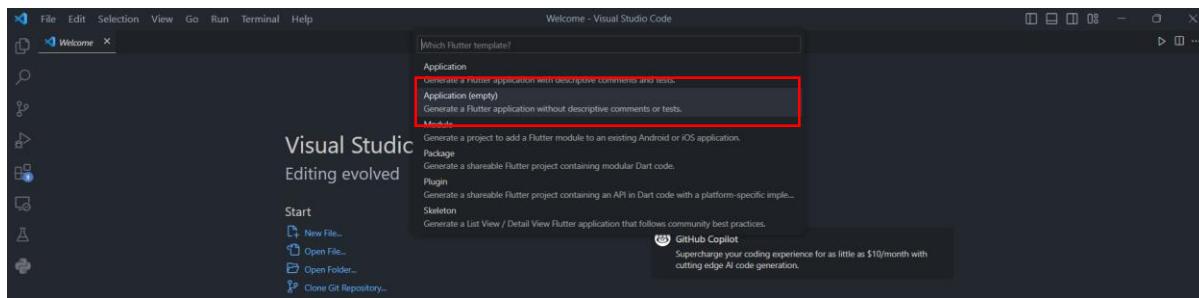


2. Open Command Pallate (**Ctrl + Shift + P**) dengan menggunakan shortcut pada windows atau **Cmd + Shift + P** pada MacOS.
3. Lalu cari Flutter: New Project and enter.



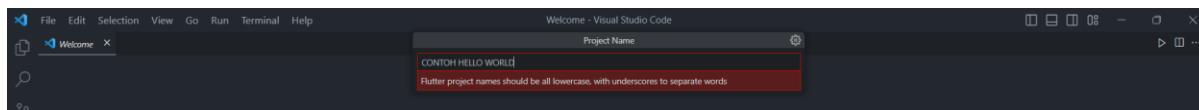
Gambar 29 Pilih bagian Flutter: New Project

4. Lalu pada bagian ini, saya sarankan memilih Application (Empty) agar membuat aplikasi kosongan, jika memilih yang diatas, maka aplikasi yang diberikan sudah berisi sebuah aplikasi sederhana, karena kita ingin membuat hello world, maka pilih application empty.

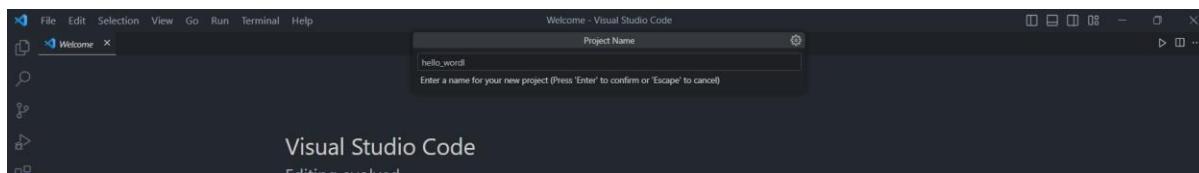


Gambar 30 Pilih yang empty

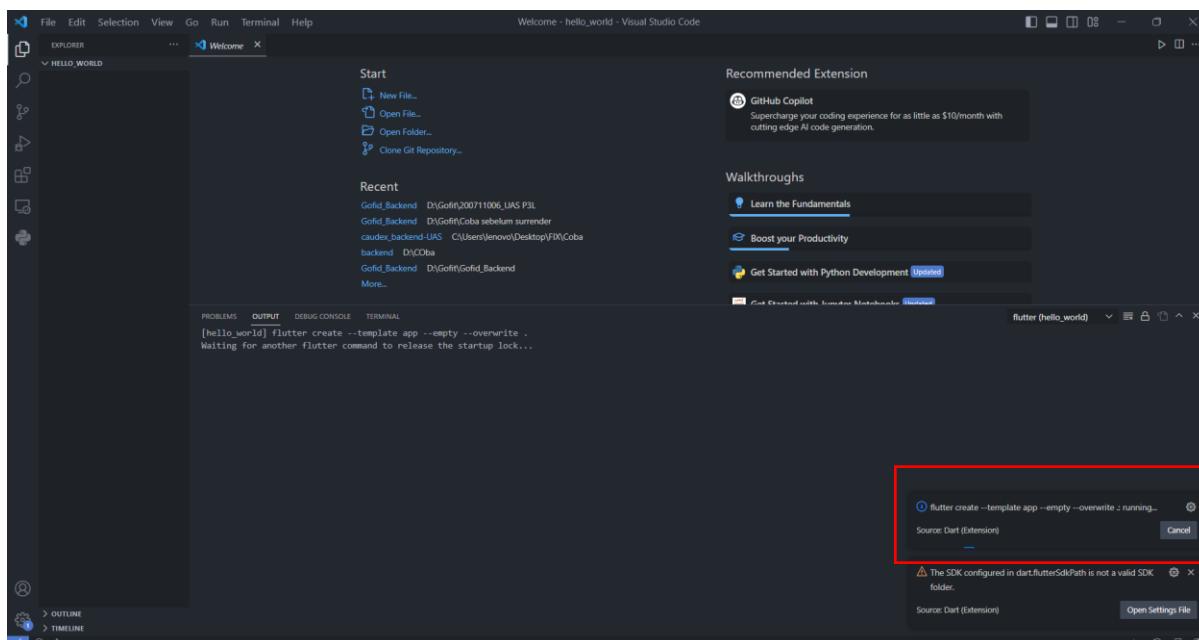
5. Setelah itu anda bebas menyimpan file dimana pun yang anda mau. Kemudian anda menamakan file tersebut. Namun penamaan file tidak boleh mengandung huruf kapital, karena akan error seperti dibawah ini



Berikan penamaan file yang benar seperti gambar dibawah ini, lalu enter.



6. Tunggu hingga proses selesai, mungkin memerlukan beberapa waktu.



Gambar 31 Tunggu hingga proses selesai.



7. Berikut tampilan jika proses pembuatan sudah selesai, dan code sudah bisa gunakan, namun masih kosongan.

```
main.dart - hello_world - Visual Studio Code

File Edit Selection View Go Run Terminal Help
EXPLORER lib > main.dart > ...
lib > main.dart > ...
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MainApp());
5 }
6
7 class MainApp extends StatelessWidget {
8   const MainApp({super.key});
9
10 @override
11 Widget build(BuildContext context) {
12   return const MaterialApp(
13     home: Scaffold(
14       body: Center(
15         child: Text("Hello World!"),
16       ),
17     ), // Scaffold
18   ); // MaterialApp
19 }
20

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
flutter (hello_world) ... flutter run
$ cd .
$ flutter run
Your empty application code is in ./lib/main.dart.
exit code 0

OUTLINE TIMELINE DEPENDENCIES
Analyzing... 1, Col 1  Spaces 2  UTF-8  CR/LF  ⚡ Dart  Windows (Windows-x64)  Spell  Prettier
```

8. Untuk menjalankan program, anda harus memiliki sebuah emulator. Emulator bisa dibuat menggunakan Software Android Studio atau menggunakan emulator hp masing-masing (Harus android). Pertama anda harus memilih device yang ingin anda gunakan sebagai emulator dengan cara Ctrl + Shift + P, lalu pilih "Flutter: Select Device".

```
main.dart - hello_world - Visual Studio Code

File Edit Selection View Go Run Terminal Help
RUN AND DEBUG lib > main.dart > ...
Run and Debug
To customize Run and Debug configurations.
Create a launch.json file.
Show all automatic debug configurations.

Flutter: Select Device recently used
Flutter new project
Flutter copy templates to project
Flutter new Big Pack Bloc
Flutter new Bloc
Flutter new Event
Flutter new Event
Flutter new Index
Flutter new Model
Flutter new Page
Flutter new Provider
Flutter new Repository
Flutter new Screen
Flutter new Small Pack Bloc
Flutter new State
Flutter new dynamic config
Flutter new with dynamic config
Add Browser Breakpoint
other commands
```

9. Maka akan tampil daftar devide yang available untuk digunakan.

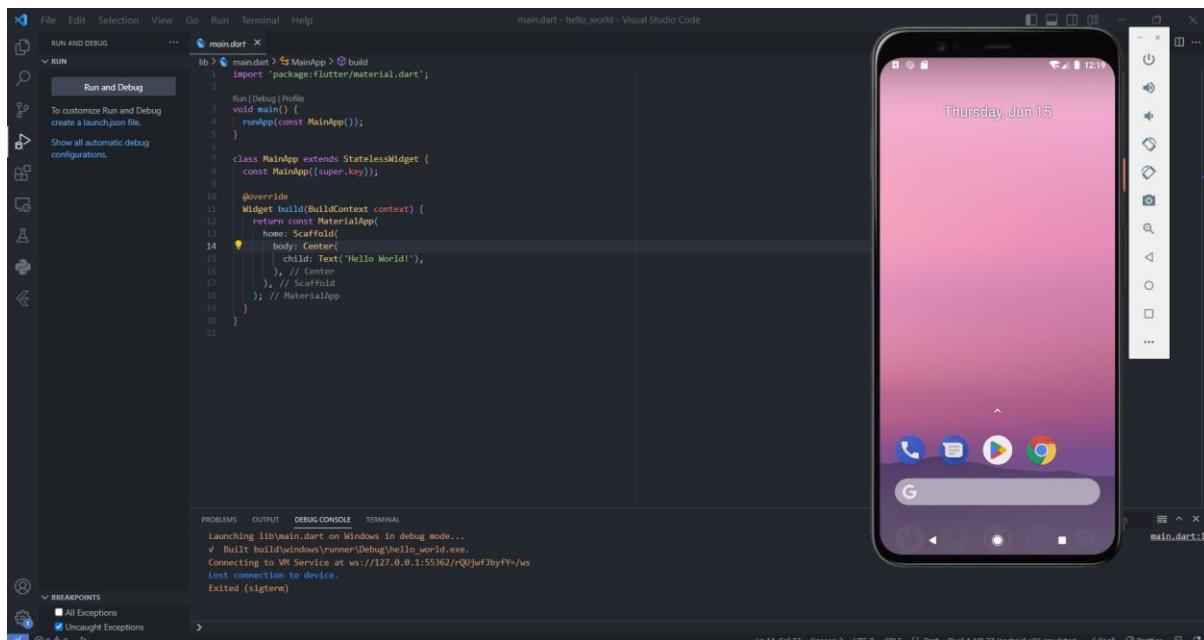
```
main.dart - hello_world - Visual Studio Code

File Edit Selection View Go Run Terminal Help
RUN AND DEBUG lib > main.dart > ...
Run and Debug
To customize Run and Debug configurations.
Create a launch.json file.
Show all automatic debug configurations.

Select a device to use
Available Devices
Windows desktop
Chrome web
Edge web
M2010K7BNY mobile
Start Pixel 4 API 27 mobile emulator
Start Pixel 4 API 27 mobile emulator (cold boot)
Create Android emulator
Offline Emulators
```

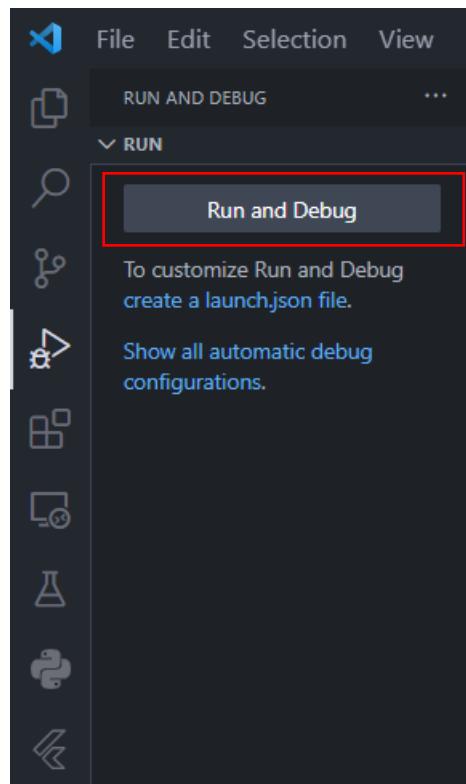


Pada saat ini, saya memiliki 1 emulator dari android studio dan 1 emulator menggunakan device smartphone. Anda bisa memilih yang mana saja. Namun jika ingin menggunakan smartphone anda harus mengaktifkan mode pengembang serta mengizinkan debugging menggunakan USB. Saya akan menggunakan emulator dari Android Studio dengan cara pilih emulator tersebut lalu enter dan tunggu hingga proses selesai.

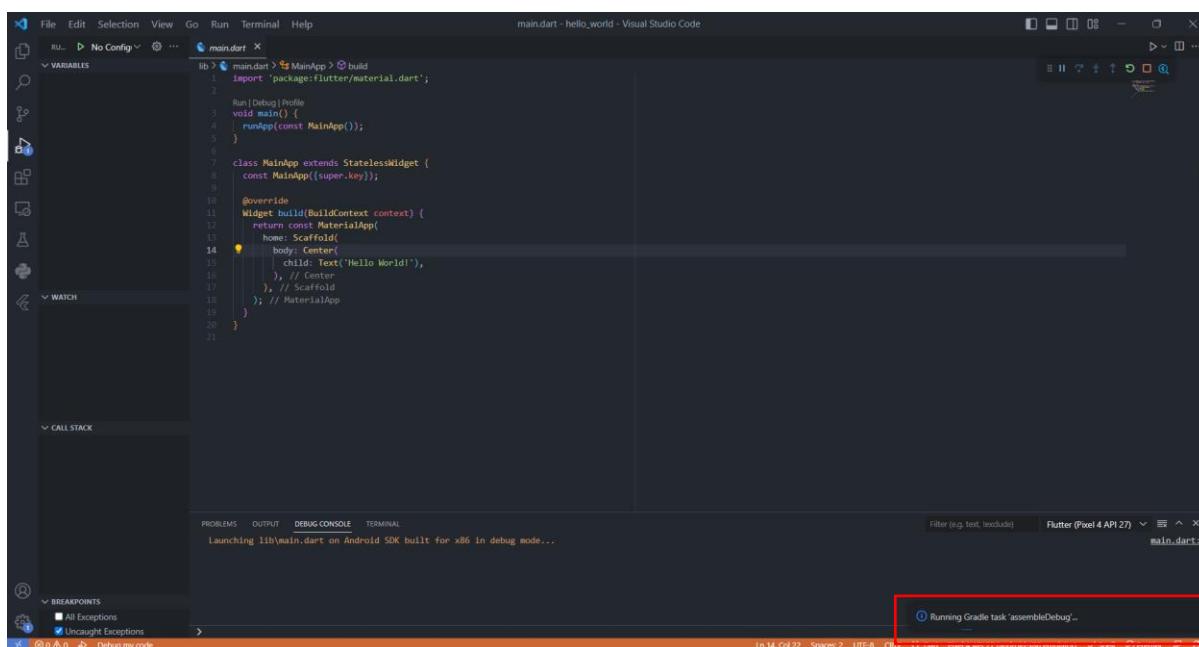


Gambar 32 Tampilan setelah emulator dihidupkan

10. Kemudian untuk menjalankan program, anda perlu masuk pada bagian “Run and Debugging” atau “Ctrl + Shift + D” pada keyboard, lalu klik pada bagian “Run and Debug”.



Anda bisa juga menggunakan cara lain yaitu dengan F5 pada keyboard, maka otomatis akan start debugging atau Ctrl + F5 untuk start without debugging. Dan tunggu hingga proses hingga selesai.



Gambar 33 Tunggu hingga proses selesai.

11. Berikut tampilan jika program sudah berhasil di run.



The screenshot shows the Visual Studio Code interface with the main.dart file open. The code defines a simple StatelessWidget named MainApp that contains a Scaffold with a Centered 'Hello World!' text. Below the code editor is the terminal window showing the build process for an Android APK. To the right is an Android emulator window displaying the application's output.

```
lib > main.dart
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MainApp());
5 }
6
7 class MainApp extends StatelessWidget {
8   const MainApp({super.key});
9
10  @override
11  Widget build(BuildContext context) {
12    return const MaterialApp(
13      home: Scaffold(
14        body: Center(
15          child: Text('Hello World!'),
16        ), // Center
17      ), // Scaffold
18    ); // MaterialApp
19  }
20}
21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Launching lib/main.dart on Android SDK built for x86 in debug mode...
↳ Built build/app/outputs/flutter-apk/app-debug.apk.
D/EGL_emulation(4509): egMainCurrent: 0xa5b8cf00: ver 2 0 (tinfo 0x865c8830)
I/zygote (4509): Do partial code cache collection, code=17KB, data=20KB
I/zygote (4509): After code cache collection, code=17KB, data=20KB
I/zygote (4509): Increasing code cache capacity to 128KB
Connecting to VM Service at ws://127.0.0.1:45232/f-01wgB0Kw/vs
I/zygote (4509): Do partial code cache collection, code=61KB, data=47KB
I/zygote (4509): After code cache collection, code=61KB, data=47KB
I/zygote (4509): Increasing code cache capacity to 256KB
D/EGL_emulation(4509): egMainCurrent: 0xa5b8cf00: ver 2 0 (tinfo 0x86c55dd0)

Your Flutter project is ready! Press F5 to start running.

Gambar 34 Gambar ketika aplikasi pertama dirun

12. Anda dapat melakukan sedikit modifikasi, dimana dapat menambahkan header berupa tulisan.

The screenshot shows the Visual Studio Code interface with the main.dart file open. The code now includes an AppBar with the title 'Ini Header'. The Android emulator shows the updated application with the new header.

```
lib > main.dart > MainApp > build
1 import 'package:flutter/material.dart';
2
3 Run | Debug | Profile
4 void main() {
5   runApp(MainApp());
6 }
7
8 class MainApp extends StatelessWidget {
9   const MainApp({Key? key}) : super(key: key);
10
11  @override
12  Widget build(BuildContext context) {
13    return MaterialApp(
14      home: Scaffold(
15        appBar: AppBar(
16          title: Text('Ini Header'),
17          automaticallyImplyLeading: true,
18        ), // AppBar
19        body: Center(
20          child: Text('Hello World!'),
21        ), // Center
22      ), // Scaffold
23    ); // MaterialApp
24  }
25}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Connecting to VM Service at ws://127.0.0.1:45232/f-01wgB0Kw/vs
I/zygote (4509): Do partial code cache collection, code=61KB, data=47KB
I/zygote (4509): After code cache collection, code=61KB, data=47KB
I/zygote (4509): Increasing code cache capacity to 256KB
D/EGL_emulation(4509): egMainCurrent: 0xa5b8cf00: ver 2 0 (tinfo 0x86c55dd0)
Reloaded 1 of 665 libraries in 4.25ms (compile: 417 ms, reload: 1356 ms, reassemble: 1039 ms).
Reloaded 1 of 665 libraries in 95ms (compile: 31 ms, reload: 429 ms, reassemble: 424 ms).
Reloaded 1 of 665 libraries in 59ms (compile: 36 ms, reload: 214 ms, reassemble: 274 ms).
Reloaded 1 of 665 libraries in 49ms (compile: 30 ms, reload: 246 ms, reassemble: 148 ms).
Reloaded 1 of 665 libraries in 57ms (compile: 61 ms, reload: 124 ms, reassemble: 98 ms).
Reloaded 1 of 665 libraries in 45ms (compile: 26 ms, reload: 268 ms, reassemble: 87 ms).

Gambar 35 Ketika ditambah sebuah header

13. Program hello world sederhana telah berhasil di buat, anda dapat mempelajari lebih lanjut serta mengeksplor dari sumber-sumber lainnya seperti dokumentasi resmi flutter, youtube ataupun dari media lainnya.

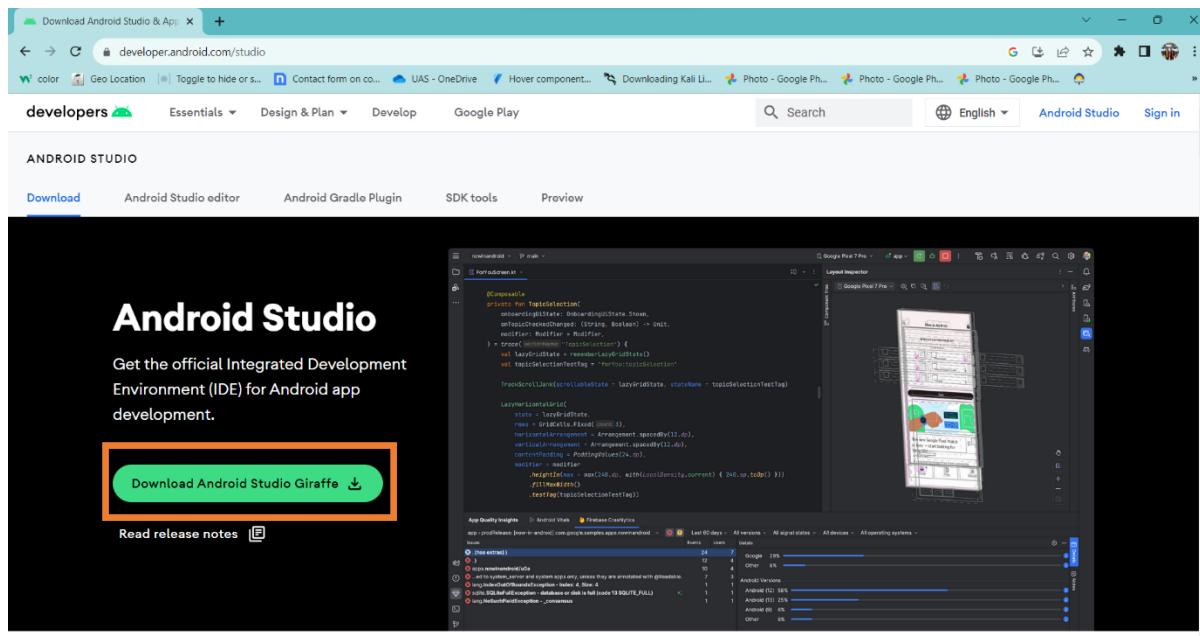


I. INSTALASI EMULATOR (ANDROID STUDIO)

Untuk running program biasanya dapat berupa 2 cara, yaitu menggunakan smartphone berupa android atau menggunakan emulator. Dalam menggunakan emulator, kita harus menginstal software Bernama "Android Studio". Android Studio merupakan sebuah tools dan IDE resmi untuk mengembangkan aplikasi android. Android Studio juga termasuk dalam code editor dan alat develop yang handal dari IntelliJ IDEA. Namun kita hanya akan menggunakan fitur emulator yang disediakan oleh Android Studio karena code editor yang digunakan adalah VS Code.

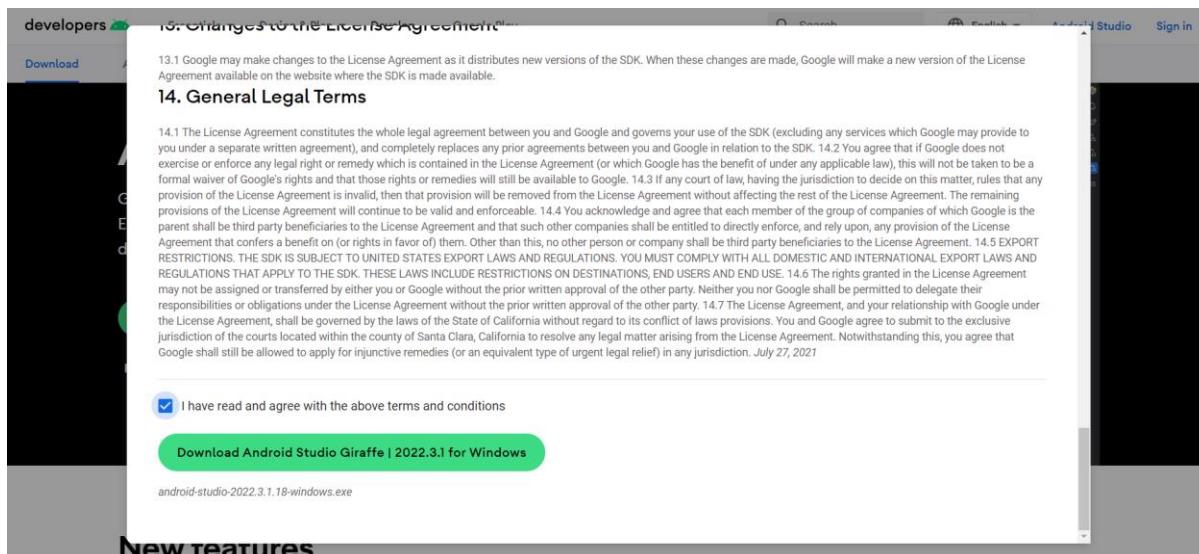
Berikut langkah-langkah dalam melakukan instalasi Android Studio:

1. Masuk pada link <https://developer.android.com/studio>, lalu pilih bagian Download Android Studio Giraffe.



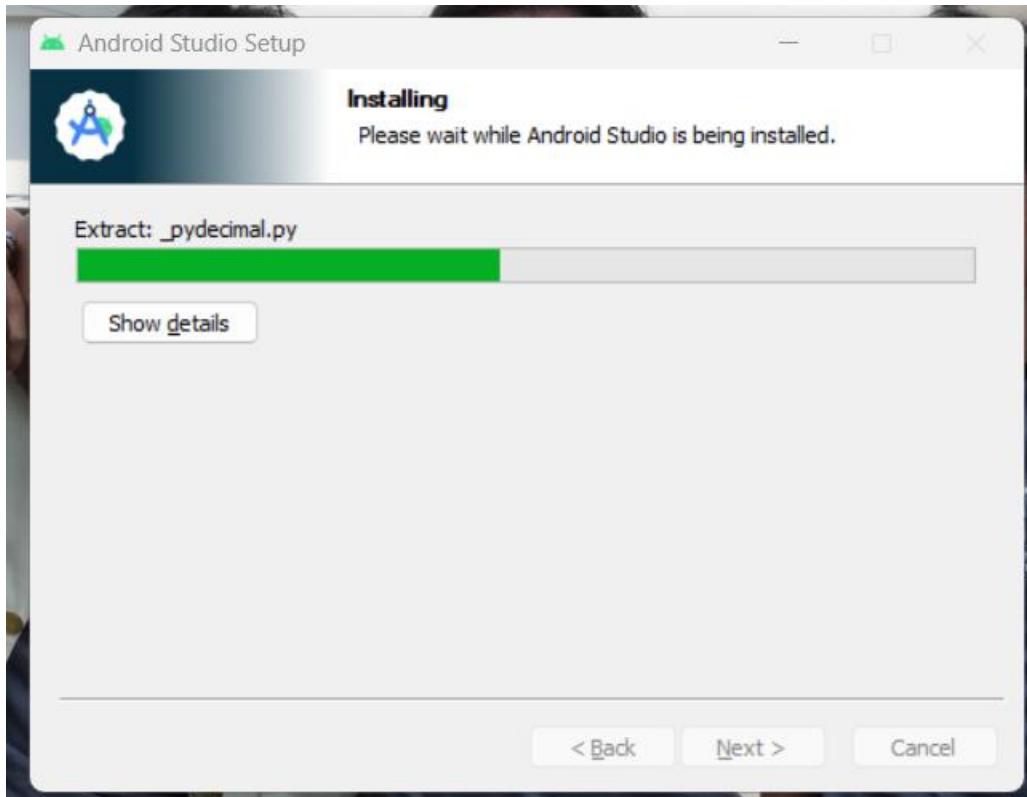
Gambar 36 Pilih bagian yang diberi tanda

2. Lalu akan muncul sebuah pop up dan scroll hingga kebawah. Kemudian centang pada bagian I have read and agree with the above terms and conditions. Kemudian tekan tombol download.



Gambar 37 Setelah mencentang lalu klik tombol download.

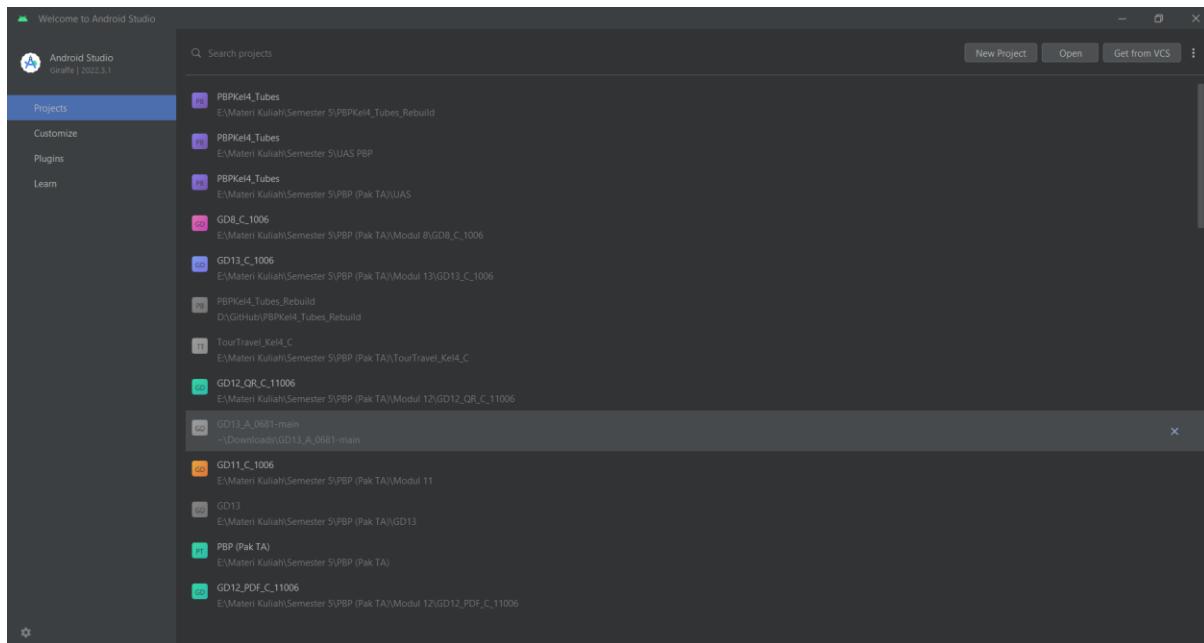
3. Tunggu hingga proses download selesai.
4. Klik 2 kali pada installer android studio. Lalu next saja terus namun pastikan lokasi file di C dan kapasitas dari C memadai dan tunggu hingga proses selesai. Kemudian Next dan Finish.



Gambar 38 Tunggu hingga proses selesai.

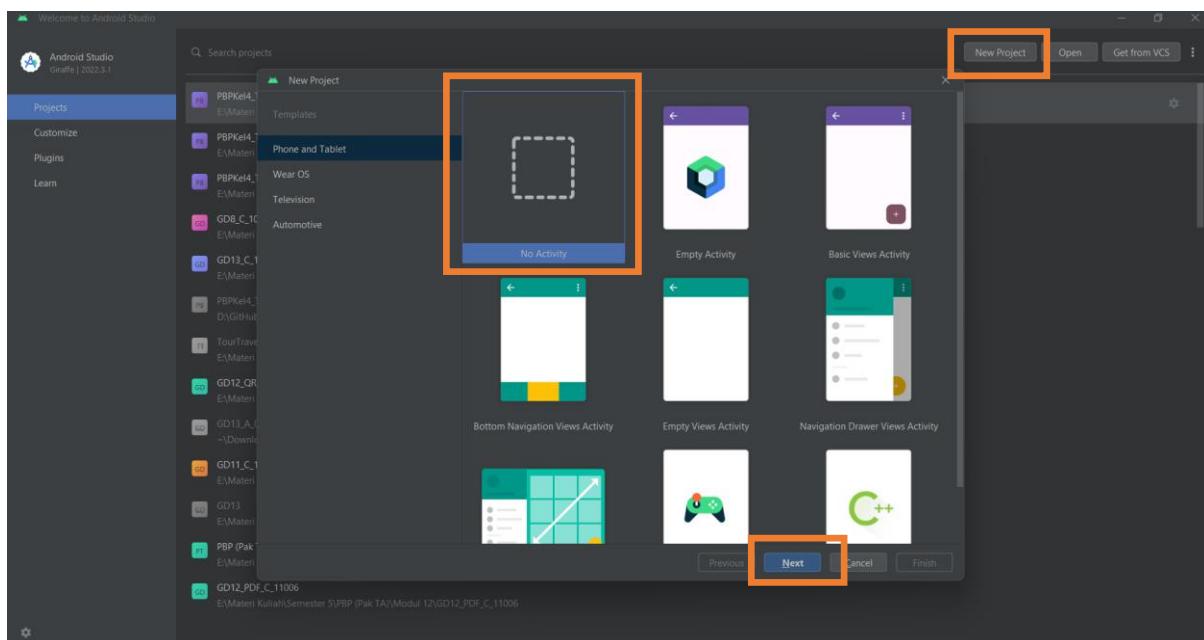


5. Ini merupakan tampilan utama dari android Studio. Dikarenakan sebelumnya sudah pernah ada project, maka tampilan awalnya adalah seperti berikut.



Gambar 39 Tampilan awal dari Andorid Studio

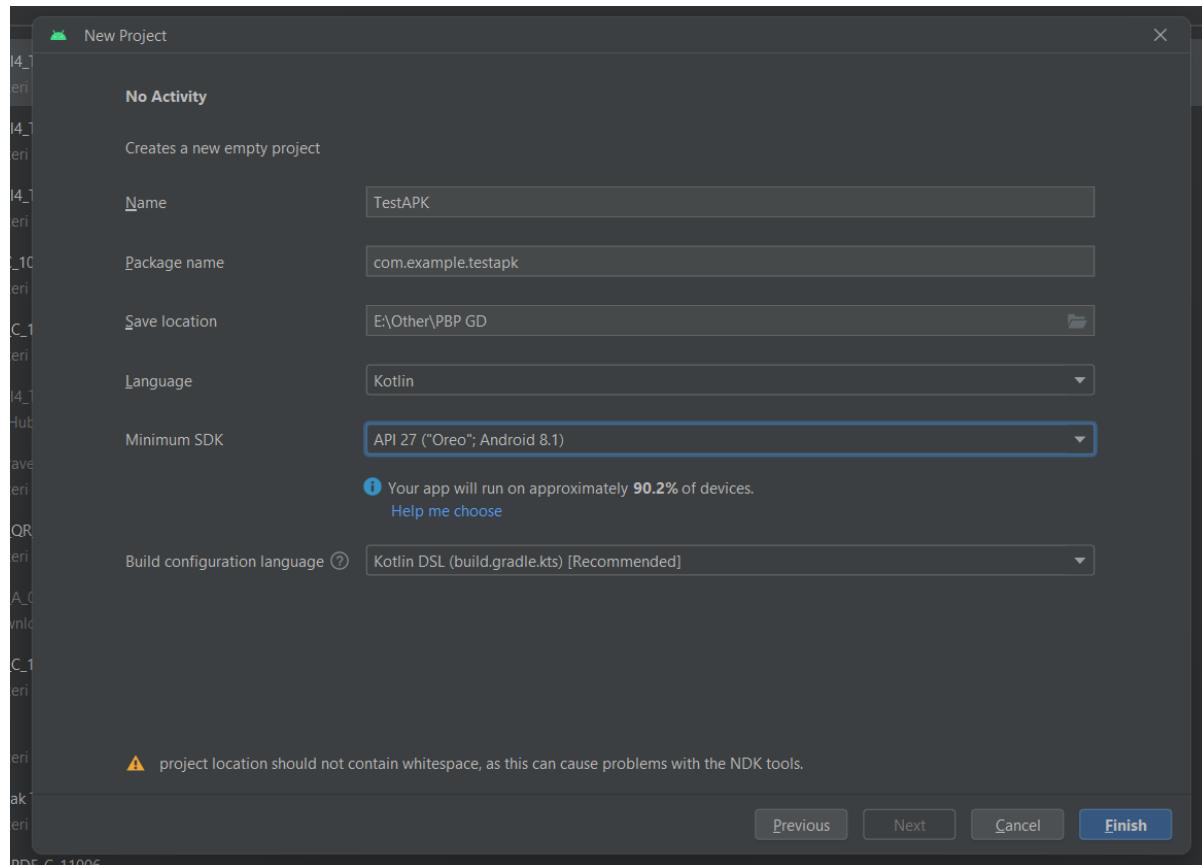
6. Untuk membuat sebuah emulator, kita harus membuat sebuah project dahulu. Klik pada bagian new project pada pojok kanan atas, kemudian pilih bagian No Activity lalu Next.



Gambar 40 Pilih bagian New Project lalu pilih pada No Activity

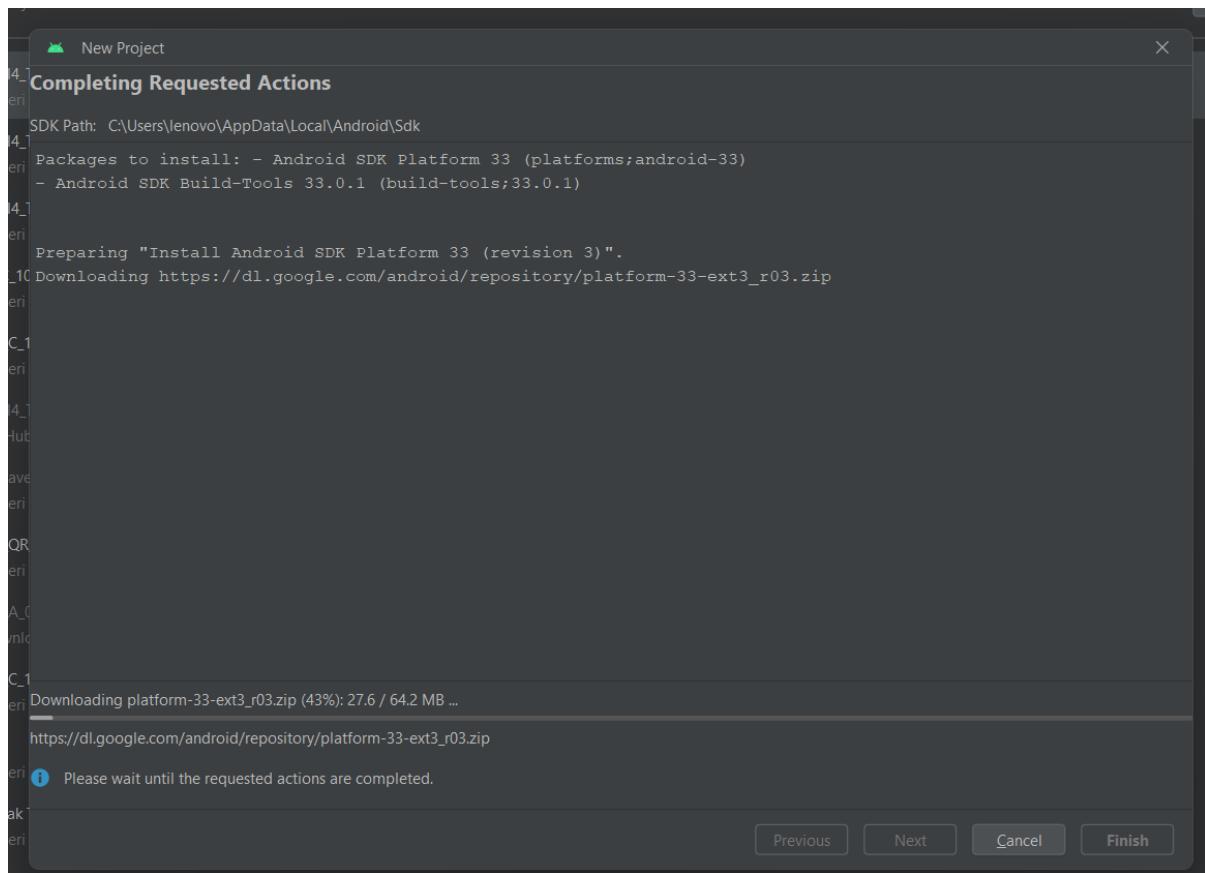


7. Pada bagian selanjutnya tidak terlalu perlu dihiraukan, karena kita hanya memerlukan emulator, jika ingin menjadikan android Studio sebagai emulator, maka yang perlu di perhatikan hanya bagian Minimum SDK yaitu API 21 (Oreo). Lalu klik finish.



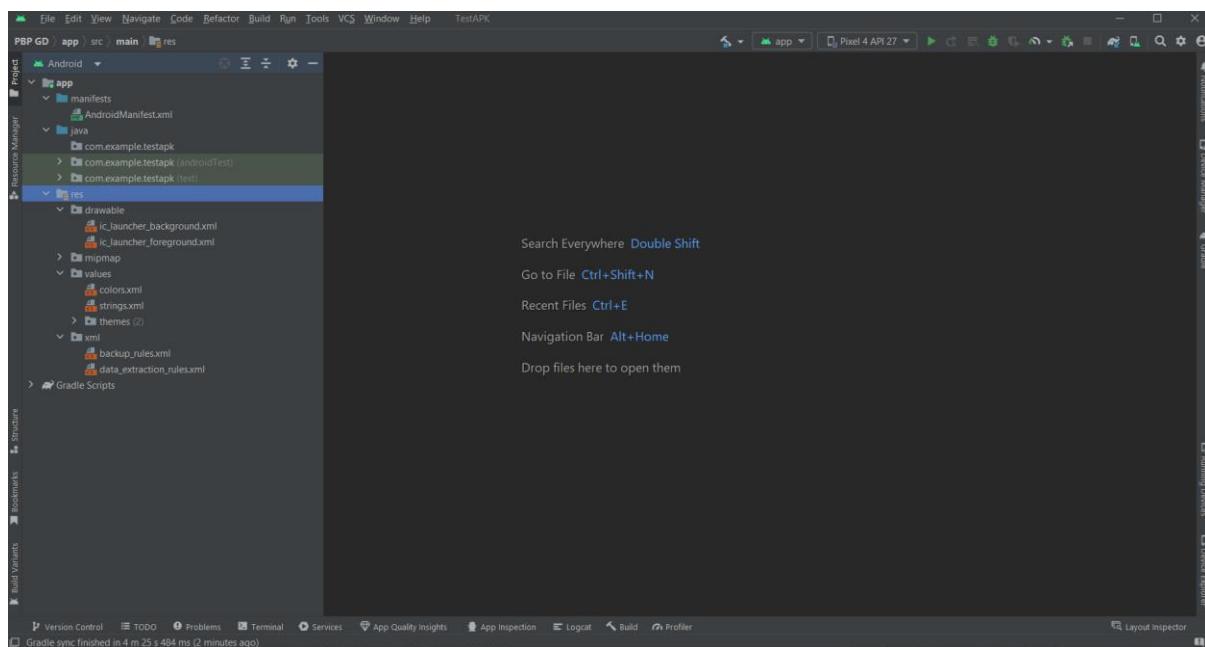
Gambar 41 Tampilan pada bagian membuat aplikasi baru

8. Setelah itu tunggu hingga proses selesai. Karena sebelumnya aplikasi baru di install, perlu melakukan download SDK dan memerlukan waktu beberapa saat. Ketika sudah selesai, klik finish.



Gambar 42 Tunggu hingga proses selesai.

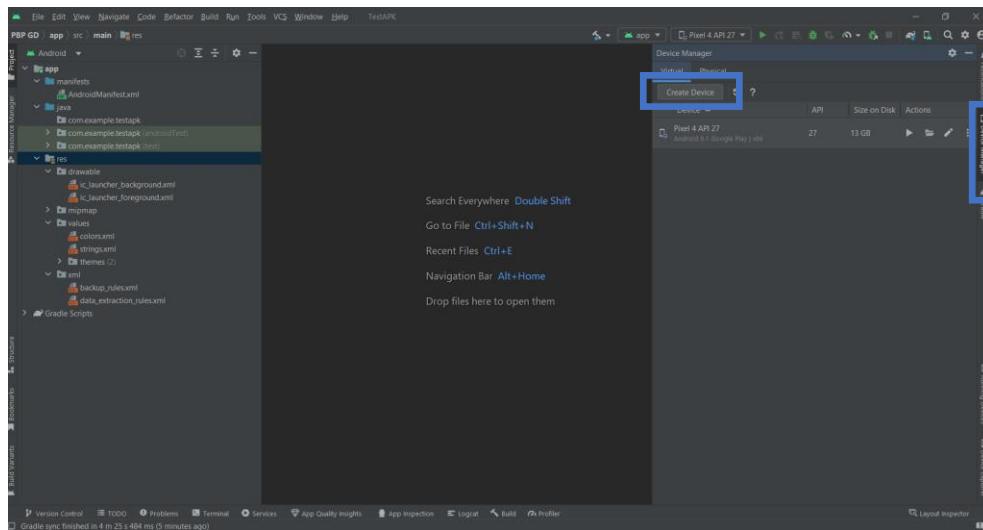
9. Ketika semua proses selesai, maka tampilan awal dari project tersebut adalah sebagai berikut.



Gambar 43 tampilan awal dari project

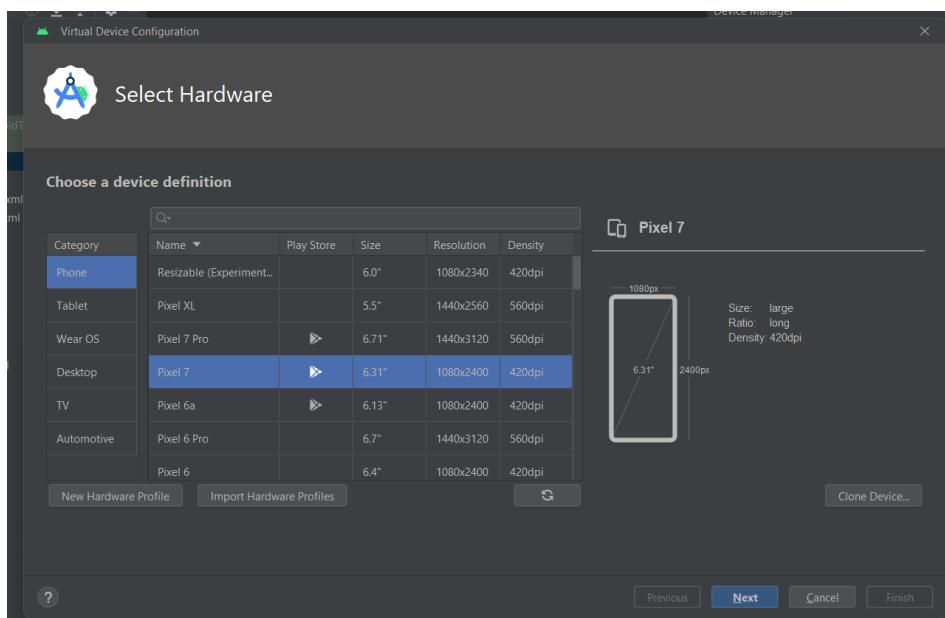


Karena yang dibutuhkan dari android studio adalah emulator nya saja, maka kita dapat membuat emulaotor. Klik pada device Manager, pada ujung kanan, kemudian “Create Device”.



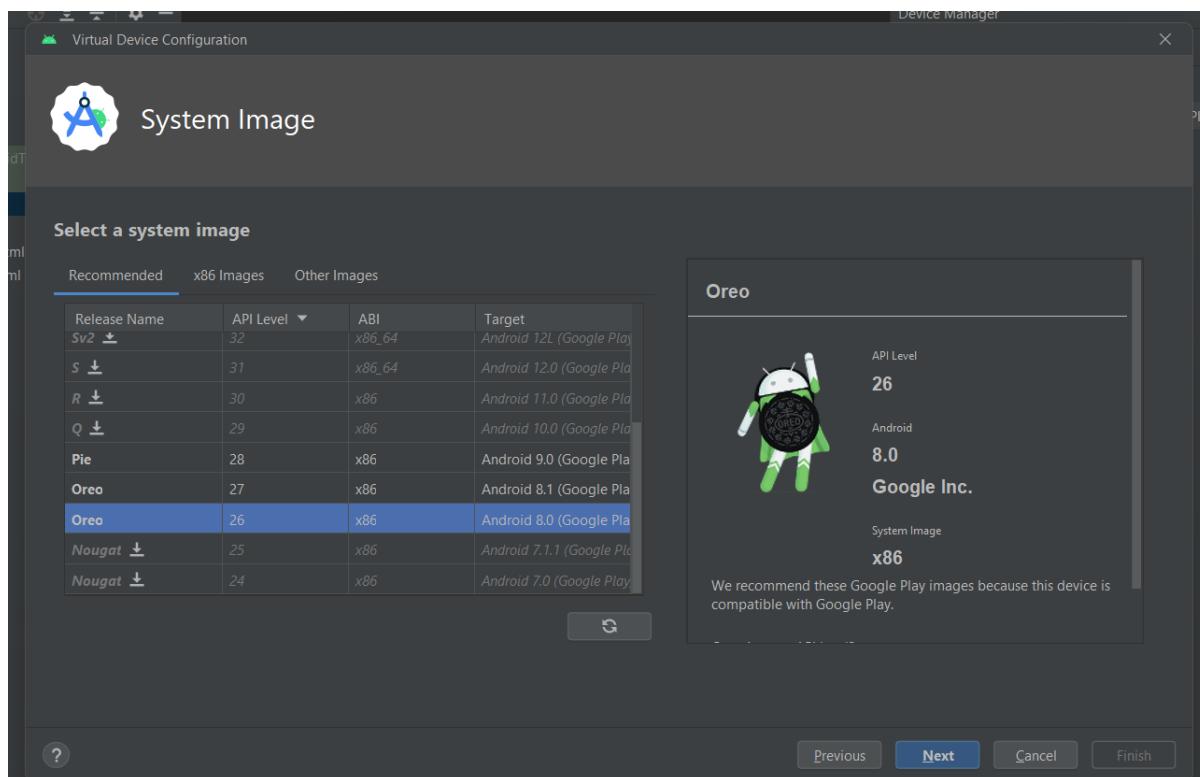
Gambar 44 Klik Pada Device Manager, lalu Create Device.

10. Pada bagian *Virtual Device Configuration*, Anda dapat menyesuaikan sesuai dengan kebutuhan. Terdapat banyak sekali jenis pilihan mulai dari Phone, Tablet, Wear OS, TV, dll. Namun sesuaikan dengan kebutuhan, karena nantinya hanya menggunakan phone, maka pilih phone saja. Anda bebas melakukan kreasti atas jenis phonanya namun saya rekomendasikan gunakan Pixel 7 dan Next.





11. Setelah itu, akan diarahkan pada bagian system image. Untuk bagian ini, usahakan menggunakan minimal oreo dengan API level 26 lalu next.



Gambar 45 Minimal menggunakan Oreo 26

12. Untuk penamaan dari emulator bersifat opsional. Anda boleh merubah namun jika tidak juga tidak menjadi masalah. Kemudian klik Finish dan tunggu hingga proses pembuatan emulator selesai. Jika sudah, emulator siap digunakan.