

**TUGAS 1**  
**WORKSHOP MOBILE FRAMEWORK**

DOKUMENTASI INSTALASI, DEFINISI, KONSEP DASAR  
PENGENALAN DAN *WORKFLOW FLUTTER*



Disusun Oleh

E41212429    Novianto Hadi Raharjo

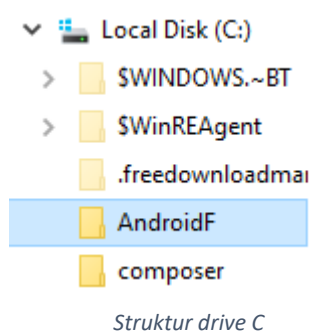
**POLITEKNIK NEGERI JEMBER**  
**JURUSAN TEKNOLOGI INFORMASI**  
**PROGRAM STUDI PLJ - TEKNIK INFORMATIKA**

# Instalasi Perangkat Lunak

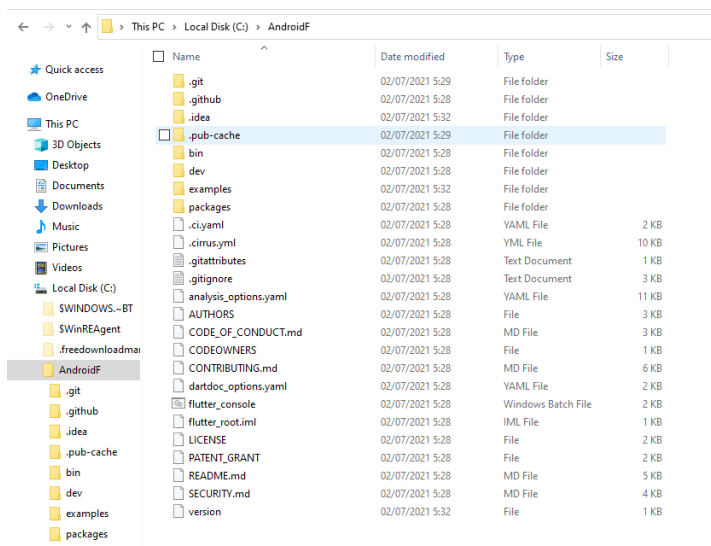
Membuat Aplikasi **Android** bisa dilakukan dengan menggunakan berbagai macam cara, salah satunya dengan menggunakan **Flutter**. Sebelum membuat aplikasi **Android** dengan **Flutter**, kita harus mempersiapkan perangkat lunak yang dibutuhkan. Setidaknya ada 4 perangkat lunak yang harus terinstall jika kita akan membuat aplikasi **android** dengan **Flutter**.

## 1. Flutter

**Flutter** merupakan suatu framework open-source yang dikembangkan oleh Google yang digunakan untuk membangun antarmuka pada aplikasi Android dan iOS. Untuk Menggunakan Flutter kita harus mengunduhnya dari laman [https://storage.googleapis.com/flutter\\_infra\\_release/releases/stable/windows/flutter\\_windows\\_2.2.3-stable.zip](https://storage.googleapis.com/flutter_infra_release/releases/stable/windows/flutter_windows_2.2.3-stable.zip) . Sembari menunggu hasil unduhan, kita siapkan folder untuk master flutter yang akan kita gunakan. Buat folder pada drive C dengan nama AndroidF, sehingga susunan drive akan terlihat seperti berikut ini.



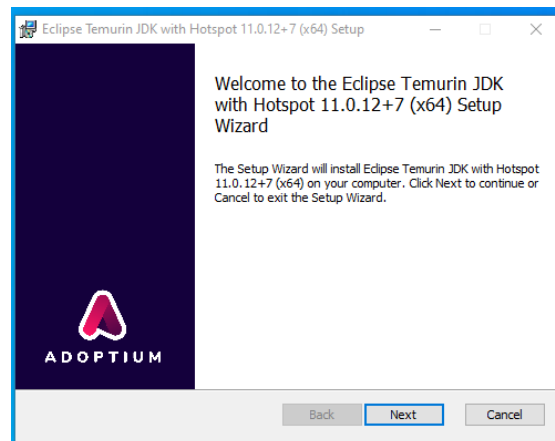
Setelah unduhan selesai, pindahkan unduhan flutter kedalam folder AndroidF yang sudah kita buat, kemudian ekstrak file unduhan. Buat sehingga akan terlihat seperti ini.



Struktur Folder AndroidF

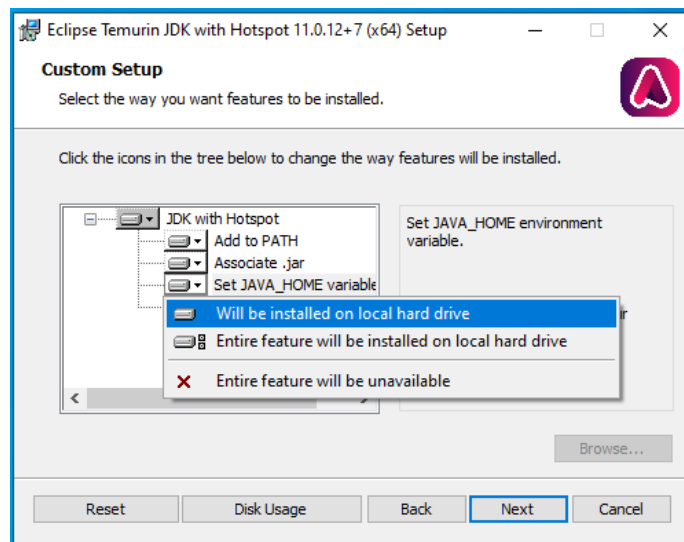
## 2. JDK

Java Development Kit(JDK) merupakan paket aplikasi yang berisi **JVM (Java Virtual Machine) + JRE (Java Runtime Environment) +** berbagai aplikasi untuk proses pembuatan kode program Java. **Kenapa flutter butuh JDK?** Sebenarnya flutter tidak membutuhkan JDK, karena Bahasa pemrograman yang digunakan flutter adalah Dart. **Lantas kenapa harus install JDK?** karena kita membutuhkan Android SDK sedangkan Android SDK membutuhkan JDK (**cmii**). Kalau memang butuh, berarti kita harus mengunduhnya dari laman <https://adoptopenjdk.net/>. Setelah unduhan selesai, jangan lupa kopinya diruput dulu sebelum melanjutkan instalasi. Instalasi JDK yang kita download tadi cukup mudah, tinggal klik 2 kali pada file JDK, kemudian aplikasi akan muncul pada layer seperti gambar berikut ini:



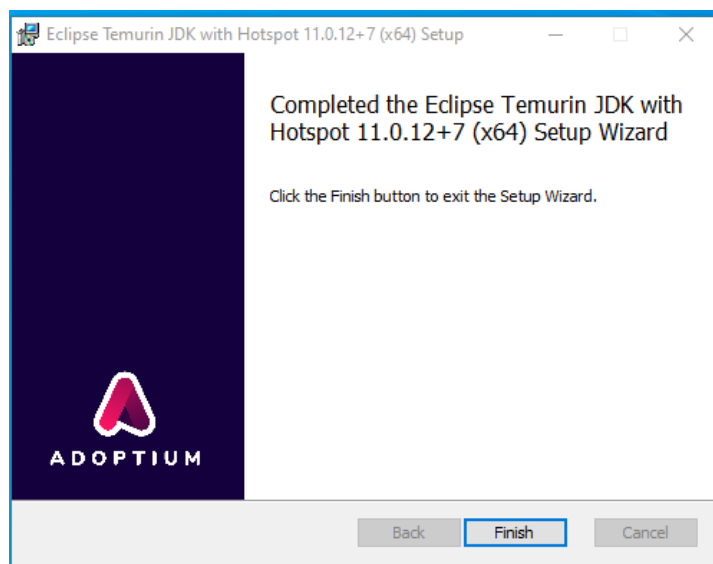
*tampilan awal installer JDK*

klik Next pada Installer, kemudian akan muncul pilihan seperti dibawah ini. **Jangan Lupa** untuk mengaktifkan **sub features Set JAVA\_HOME**, sehingga akan terlihat seperti ini.



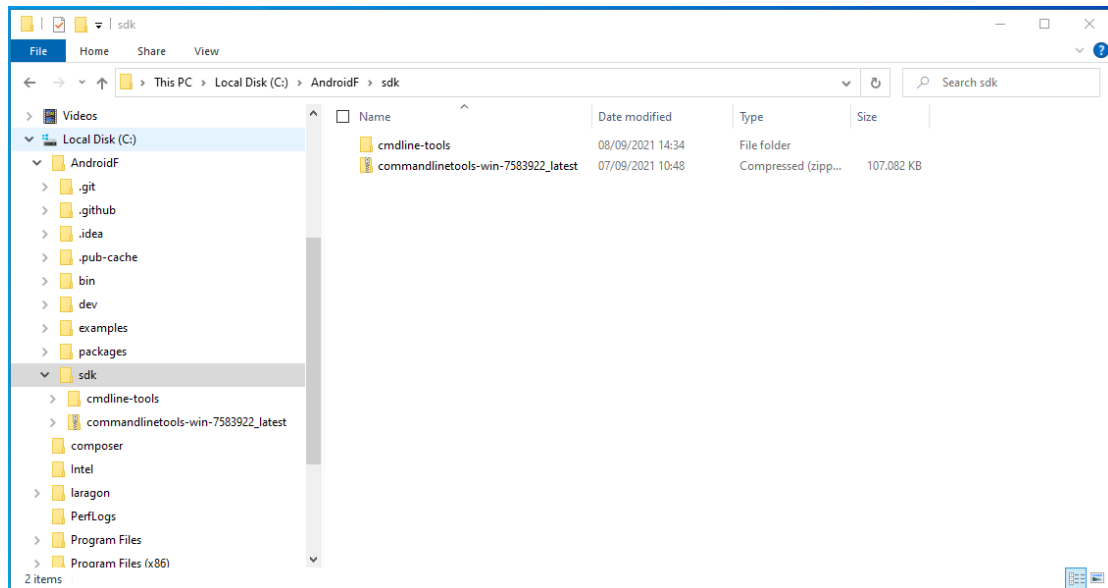
*subfeatures Set JAVA\_HOME*

Setelah selesai melakukan pengaturan **JAVA\_HOME**, klik tombol next kemudian akan muncul halaman seperti berikut, setelah itu klik finish dan Selamat, proses instalasi JDK sudah selesai.



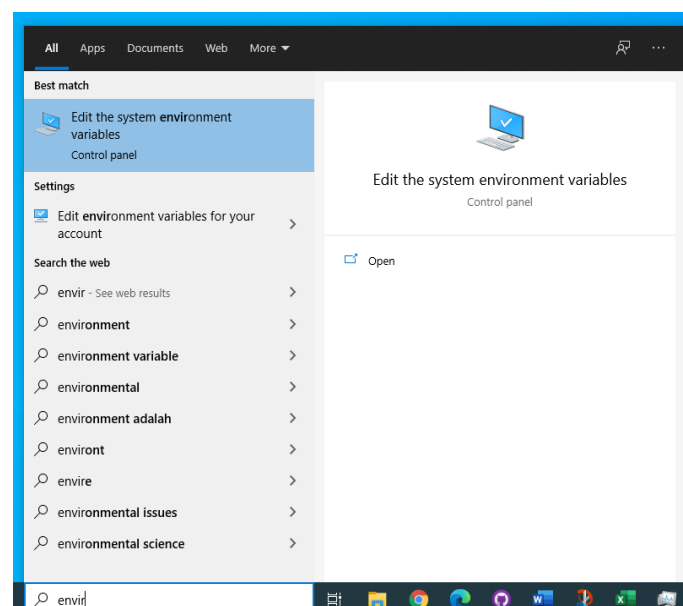
### 3. Android SDK

Android Software Development Kit (SDK) merupakan *kit* yang bisa digunakan oleh para *developer* untuk mengembangkan aplikasi berbasis Android. Di dalamnya, terdapat beberapa *tools* seperti *debugger*, *software libraries*, *emulator*, dokumentasi, *sample code* dan tutorial. Android SDK bisa diunduh melalui laman [https://dl.google.com/android/repository/commandlinetools-win-7583922\\_latest.zip](https://dl.google.com/android/repository/commandlinetools-win-7583922_latest.zip), setelah unduhan selesai buat folder **sdk** didalam folder **Androidf**, kemudian ekstrak unduhan sehingga struktur folder akan seperti ini.

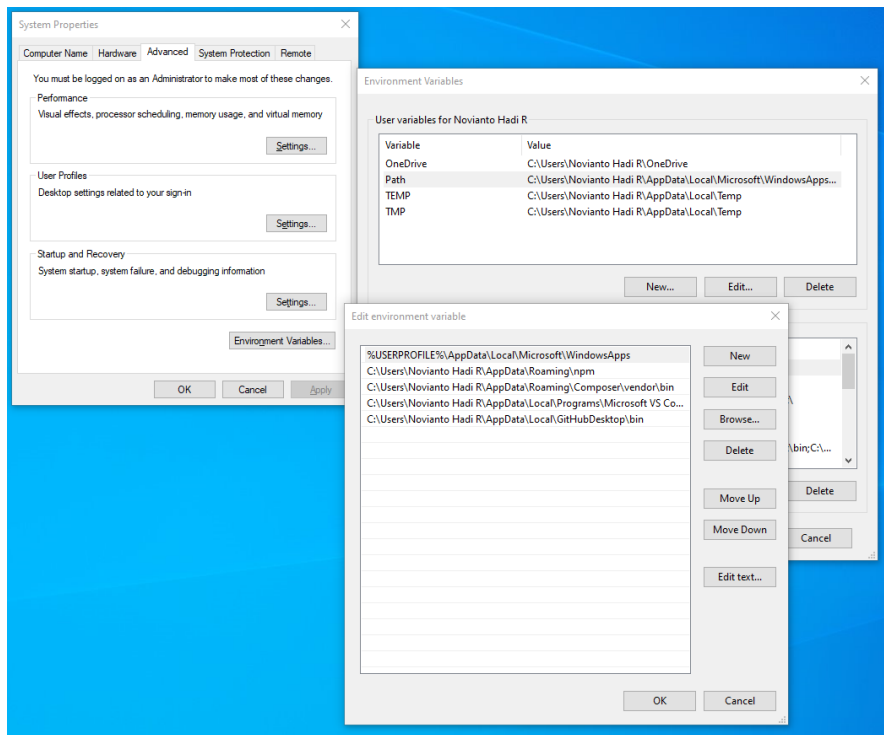


struktur folder sdk

Langkah selanjutnya adalah pengaturan **path** pada **windows environment variable**. Ketikkan environment pada windows search seperti berikut.

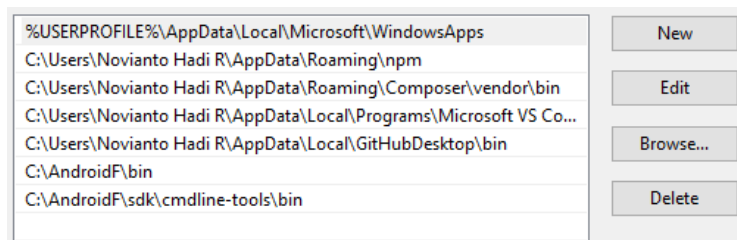


Edit Environment Variables



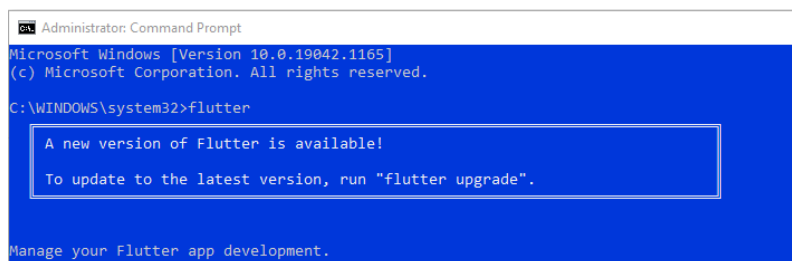
*Advanced Environment Variables*

Tambahkan path direktori dari **flutter** dan **sdk** kedalam path environment variables, disini path direktori flutter => `C:\AndroidF\bin` dan sdk => `C:\AndroidF\sdk\cmdline-tools\bin` setelah path ditambahkan, maka environment variables akan terlihat seperti berikut:



*setelah path direktori ditambahkan kedalam environment variable*

Setelah path ditambahkan, sekarang saatnya kita melakukan cek apakah path tadi sudah berhasil mengarah ke **flutter** dan **sdk**. Cara mengeceknya bisa dengan menggunakan command prompt (cmd) dengan privilege administrator kemudian ketikkan perintah **flutter** jika berhasil maka tampilan akan sebagai berikut.



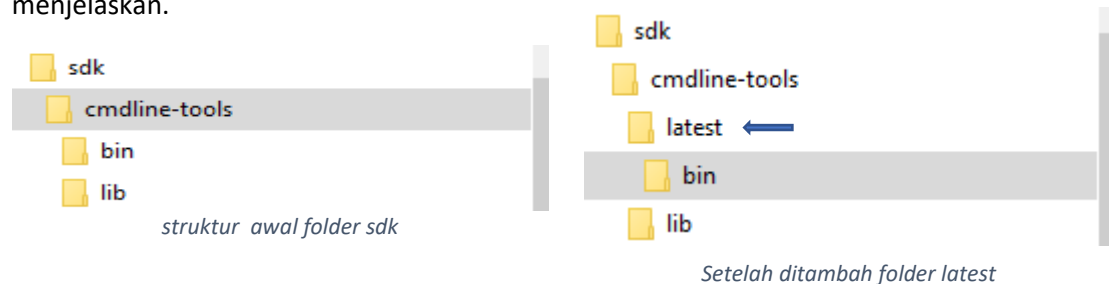
*Cek Flutter di Command Prompt*

Setelah **flutter** berhasil, sekarang kita lakukan cek untuk **sdk**-nya dengan mengetikkan perintah **sdksmanager** pada command prompt. Kemudian saya mendapatkan tampilan seperti ini.

```
C:\Windows\System32>sdkmanager
Error: Could not determine SDK root.
Error: Either specify it explicitly with --sdk_root= or move this package into its expected location: <sdk>\cmdline-tools\latest\
C:\Windows\System32>
```

*sdkmanager error*

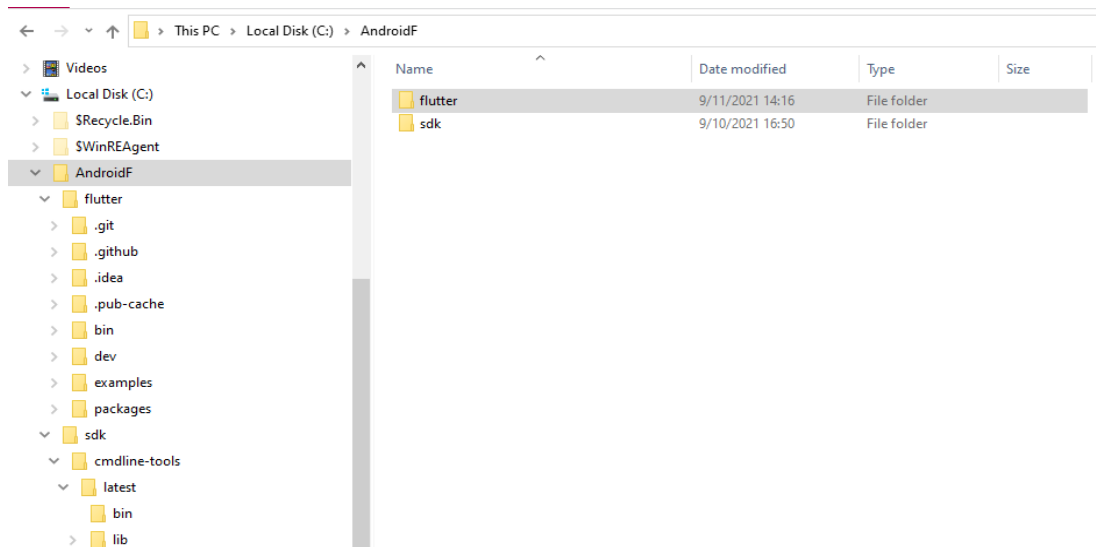
Pada tahap ini, saya mendapatkan error untuk pengaturan **sdk**. Setelah mencari di forum forum pemrograman, akhirnya yang dicari ketemu, pengaturan yang saya lakukan sudah benar. tapi kenapa masih error? Karena untuk versi yang terbaru **folder bin** yang ada di **sdk** harus berada dibawah **folder latest**, akhirnya saya buat **folder latest** dan memindahkan **folder bin** kedalam **folder latest**. Maksudnya apa?? Mungkin gambar dibawah ini bisa lebih menjelaskan.



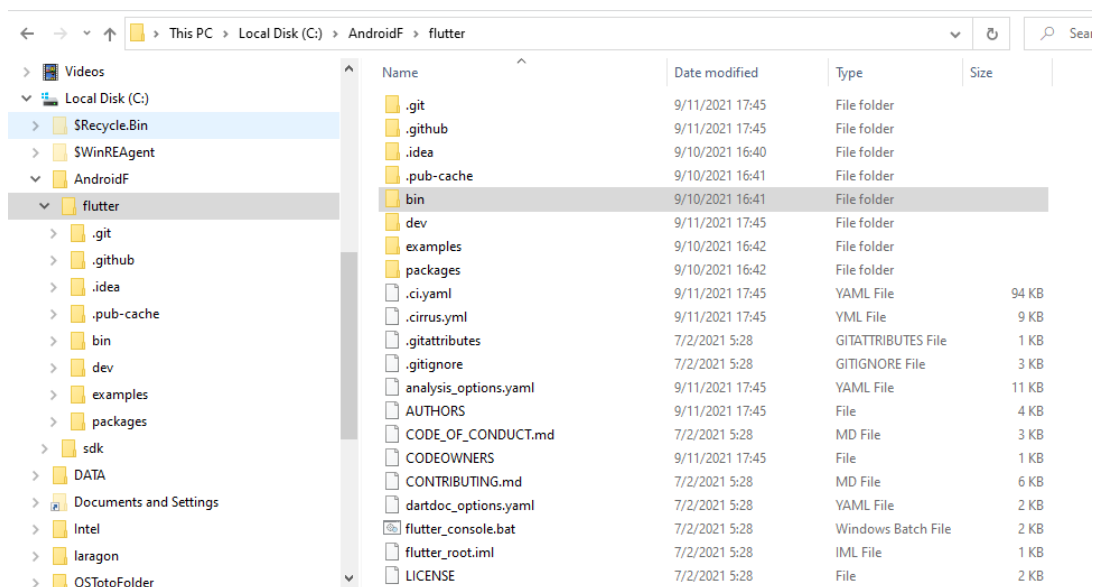
Setelah merubah struktur folder, alhamdulillah akhirnya sdkmanager juga belum bisa. Mulai agak puyeng nih, tapi jiwa pengembara semakin membara. Setelah menit berganti hari, belum juga masalah tersolusikan. Sudah banyak cara dicoba sampai sempat mau menyerah, dan ingin install Android Studio, tapi siapa yang bersungguh sungguh pasti akan ada jalan.

Akhirnya sdkmanager berhasil setelah pencarian solusi yang cukup lama, lantas bagaimana solusinya? Sabar dong, saya tak minum dulu. Oke, saya sudah minum dan sekarang waktunya membahas masalah sdkmanager, solusinya adalah re-strukturisasi total folder instalasi. Maksudnya bagaimana? Diatas ada beberapa gambar tentang strukturisasi folder yang saya gunakan, saya hapus semua dan saya buat baru. Capek memang, tapi akhirnya paham cara install flutter di windows 10 dengan visual studio code tanpa android studio.

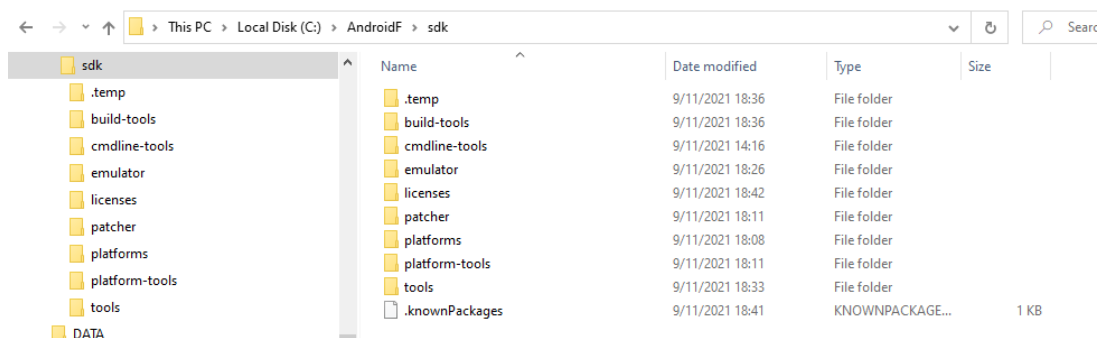
Lantas struktur folder yang baru seperti apa? Untuk folder utama tetap tidak berubah, saya menggunakan folder **AndroidF** yang berada di path **C:\AndroidF**, didalam folder **AndroidF** saya buat 2 folder baru, **folder flutter** dan **folder sdk**, sehingga path **flutter** yang baru seperti ini **C:\AndroidF\flutter\bin** dan path **sdk** yang baru seperti ini **C:\AndroidF\sdk\cmdline-tools\latest\bin**. Untuk memperjelas, berikut gambar dari struktur folder yang baru.



*Struktur Folder Baru*



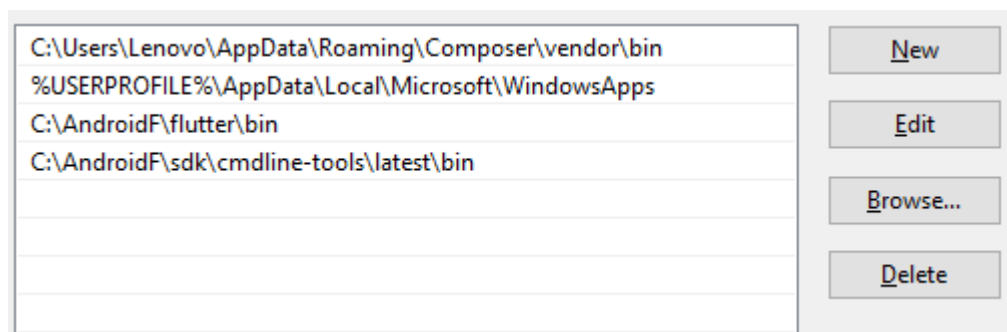
*Struktur Didalam Folder Flutter*



*Struktur didalam Folder sdk*

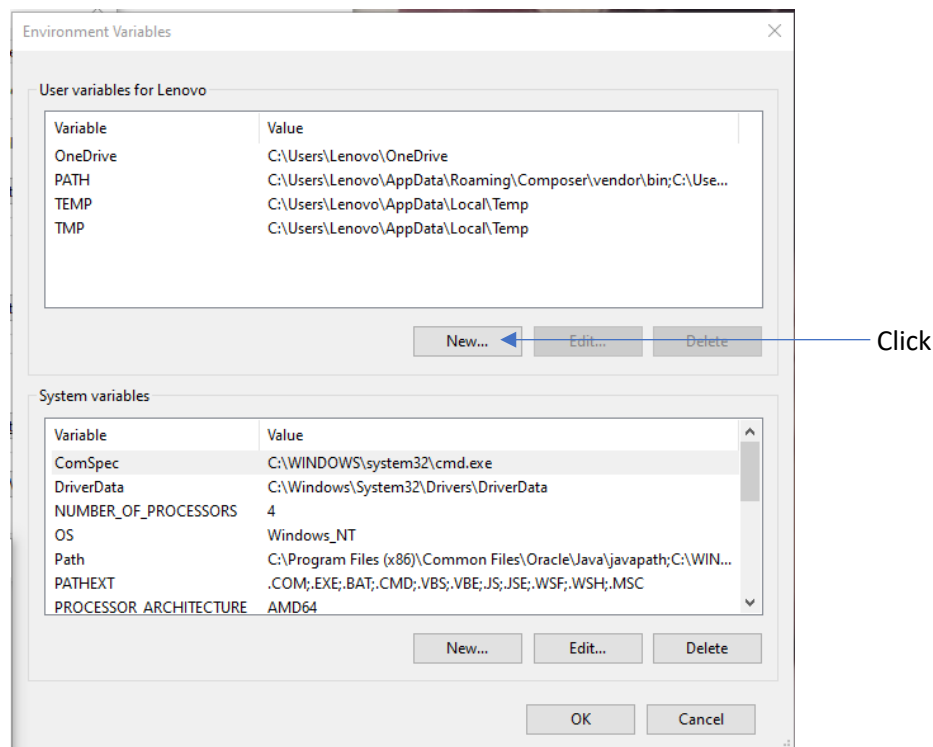
Struktur folder sdk diatas setelah melakukan unduhan platform, build-tools dan licenses, sebelumnya hanya ada folder **cmdline-tools** saja.

untuk pengaturan path di environment variable menjadi seperti dibawah ini:



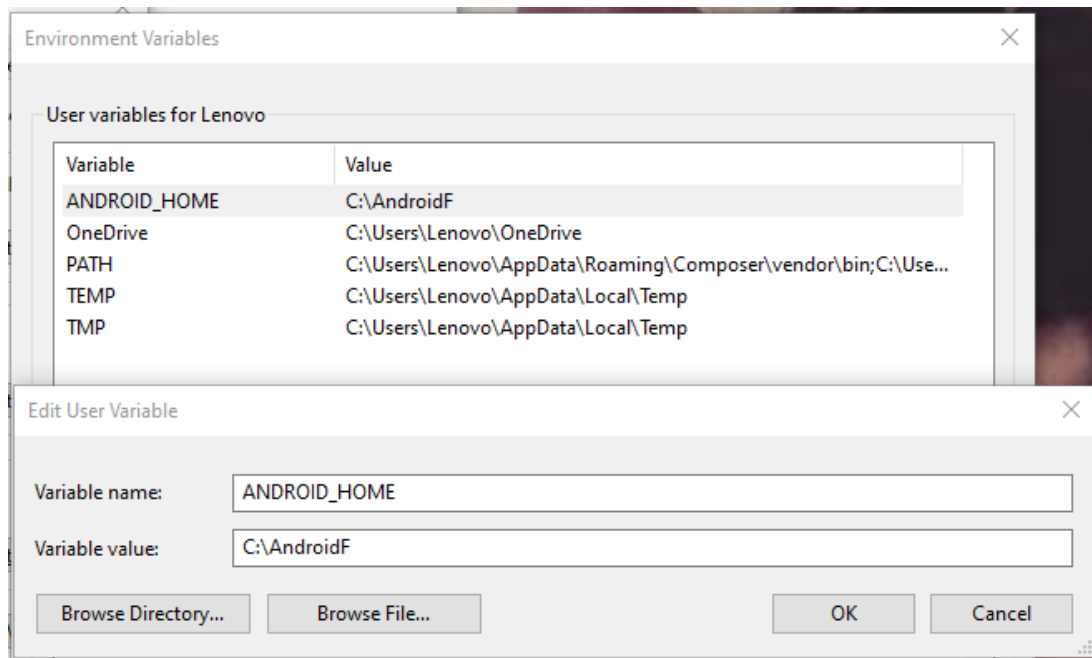
*path environment variable yang baru*

Kemudian tambahkan environment variable **ANDROID\_HOME** dengan klik tombol New...



Kemudian akan muncul pop up seperti dibawah ini, isikan Variable name dengan **ANDROID\_HOME** dan Variable value diisi dengan path root folder dari **flutter**, disini root folder dari **flutter** adalah folder **AndroidF**, jadi pathnya adalah **C:\AndroidF**, sehingga akan seperti berikut.





Menambahkan ANDROID HOME

Sebenarnya kenapa sih harus menambahkan variabel **ANDROID\_HOME**? Disini variabel **ANDROID\_HOME** digunakan oleh **Android** atau **flutter** untuk mencari lokasi dari **sdk** yang akan digunakan, karena pengaturan default **sdk** berada di dalam variabel **%ANDROID\_HOME%\sdk**, Maksudnya? Ribet ya, hm..iya lokasi default dari **sdk** itu berada di variabel **%ANDROID\_HOME%\sdk**, disini bisa diartikan menjadi **C:\AndroidF\sdk** kok bisa? Ya bisa lah, kan variabel **ANDROID\_HOME** berisi path **C:\AndroidF**, jadi tinggal tambahkan **\sdk**. Kok agak mumet ya? Berarti sudah agak paham. Tak kasih satu contoh lagi, buka tekan windows + r (run), kemudian ketikkan **%ANDROID\_HOME%**, Tara...., sampai sini paham?

Selanjutnya, apa gunanya pengaturan **PATH**? Namanya juga **PATH** yang berarti jalur. Pengaturan **PATH** bisa diartikan dengan pengaturan atau pembuatan jalur menuju file yang bisa dieksekusi atau bahasa kerennya executable file, biasanya berada didalam folder bin (biasanya). Ketika kita sudah setting **path flutter** kita bisa mengeksekusi file **flutter.bat** di command prompt tanpa mengetikkan **path** dari file **flutter dot bat**. Kok yo mbulet se, maksudnya begini, coba buka command prompt kemudian ketik **c:\AndroidF\flutter\bin\flutter** kemudian enter, terus ketik **flutter** kemudian enter, Sama kan?, kok bisa? Karena kita sudah melakukan pengaturan **PATH** menuju file **flutter.bat**, jadi Ketika di cmd kita tidak perlu mengetikkan pathnya lagi, cukup panggil nama file executable-nya saja. Kalau belum paham, ya sudahlah..

Sekarang waktunya melihat hasil dari kerja keras yang selama ini dilakukan, Cara mengeceknya bisa dengan menggunakan command prompt (cmd) dengan privilege administrator kemudian ketikkan perintah **flutter**, it's work, kemudian ketik **sdkmanager** berhasil juga, yeay...

```
C:\WINDOWS\system32>flutter
Manage your Flutter app development.
```

```
C:\Windows\System32>sdkmanager
[=====] 100% Computing updates...
```

Setelah itu kita cek dengan mengetikkan perintah *flutter doctor* dan hasilnya seperti ini.

```
Running flutter doctor...
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 2.5.0, on Microsoft Windows [Version 10.0.19042.1165], locale en-US)
[✗] Android toolchain - develop for Android devices
    X Android SDK not found at this location.
      but Android SDK not found at this location.
[✓] Chrome - develop for the web
[?] Android Studio (not installed)
[✓] Connected device (2 available)

! Doctor found issues in 2 categories.
```

Cobaan apalagi ini Tuhan..., tenang, jangan panik, panik make it stupid. Tarik napas dulu, sekarang kita ketikkan Langkah selanjutnya. Tapi itu masih error lho..Biarin!!.. sekarang kita ketik *sdkmanager "platforms;android-30"* kemudian tekan Enter, pada tahap ini, kita **WAJIB TERHUBUNG INTERNET**. Karena pada tahap ini kita melakukan download **android 11**, lha itu kok android-30, itu nama lain android 11(cmii).

```
C:\Windows\System32>sdkmanager "platforms;android-30"
License android-sdk-license: 100% Computing updates...
Terms and Conditions
This is the Android Software Development Kit License Agreement
1. Introduction
```

Setelah itu ada persetujuan, ketik *y* dan *enter*.

```
January 16, 2019
Accept? (y/N): y
[=====] 100% Unzipping... android-11/framework
```

Setelah itu kita ketikkan *sdkmanager "build-tools;30.0.0"* untuk melakukan pengunduhan build-tools versi 30.

```
C:\Windows\System32>sdkmanager "build-tools;30.0.0"
[=====] 100% Unzipping... android-11/renderscr
```

Kemudian kita ketikkan *sdkmanager --licenses*, ada sekitar enam pertanyaan yang harus kita jawab *y* semua. Kalau mau dibaca monggo, kalau saya langsung enter enter aja.

```
C:\Windows\System32>sdkmanager --licenses
6 of 7 SDK package licenses not accepted. 100% Computing updates...
Review licenses that have not been accepted (y/N)? y

August 15, 2011
Accept? (y/N): y

January 16, 2019
Accept? (y/N): y

June 2014.
Accept? (y/N): y

November 19, 2013
Accept? (y/N): y

All SDK package licenses accepted
```

Download semua kebutuhan sdk sudah, selesai, harusnya sdk sudah tidak bermasalah lagi, kita cek lagi dengan menggunakan perintah *flutter doctor*.

```
C:\Windows\System32>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 2.5.0, on Microsoft Windows [Version 10.0.19042.1165], locale en-US)
[✓] Android toolchain - develop for Android devices (Android SDK version 30.0.0)
[✓] Chrome - develop for the web
[?] Android Studio (not installed)
[✓] Connected device (2 available)

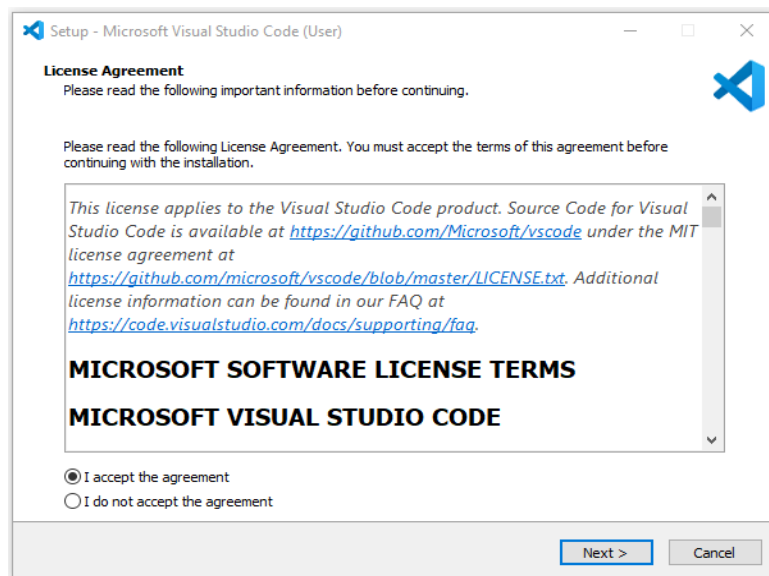
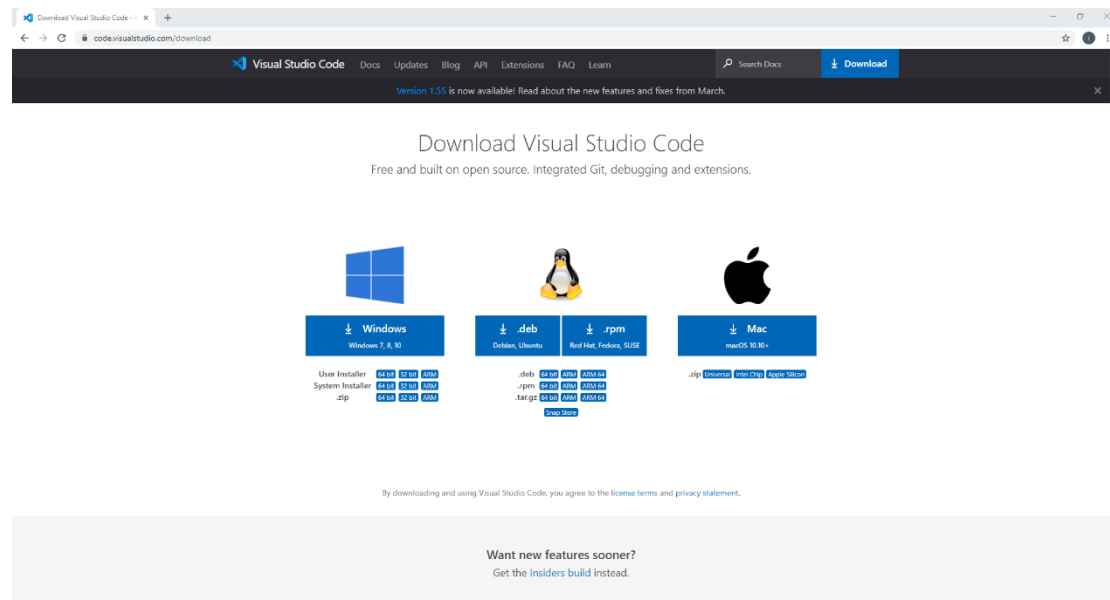
! Doctor found issues in 1 category.
```

Yeay.. Bahagia itu sederhana, seperti ini sudah bisa membuat Bahagia.

Dari keterangan flutter doctor, didapat informasi **android studio** belum terpasang, memang tidak akan terpasang, karena disini saya akan menggunakan **visual studio code**. Mumpung sedang Bahagia, yuk lanjut install VS Code.

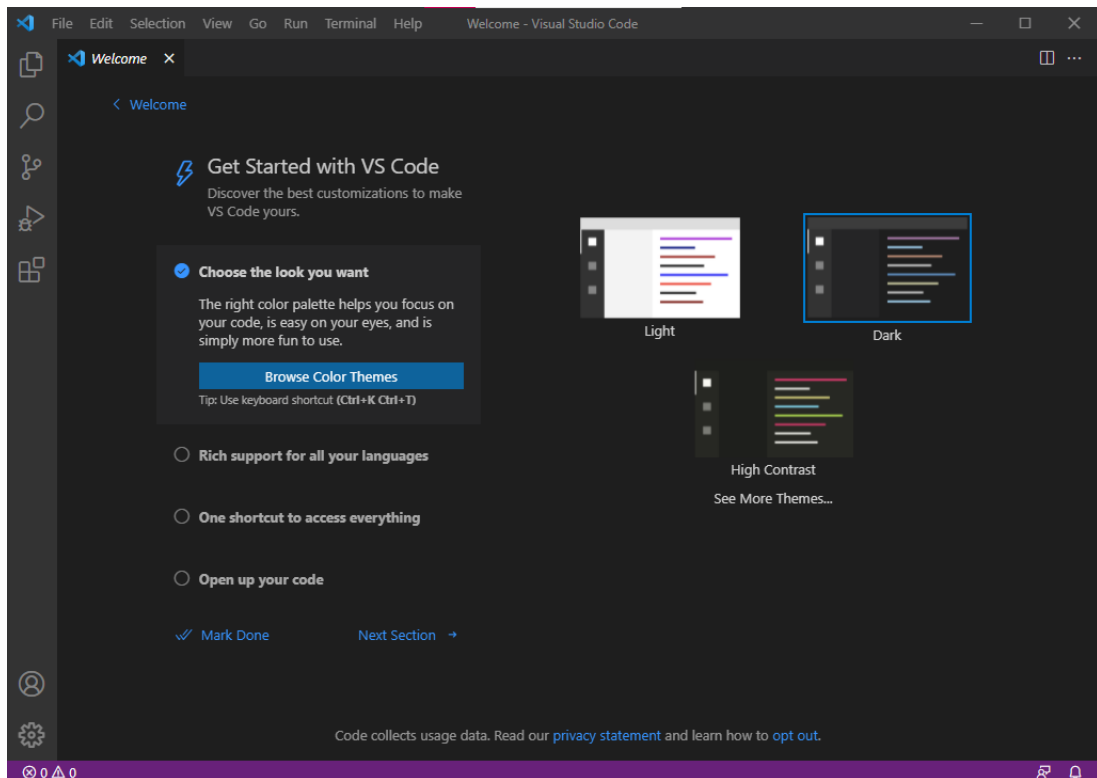
#### 4. Visual Studio Code

VS Code is my most used and favorite editor. Untuk mengunduh VSCode bisa menggunakan tautan ini <https://code.visualstudio.com/>.

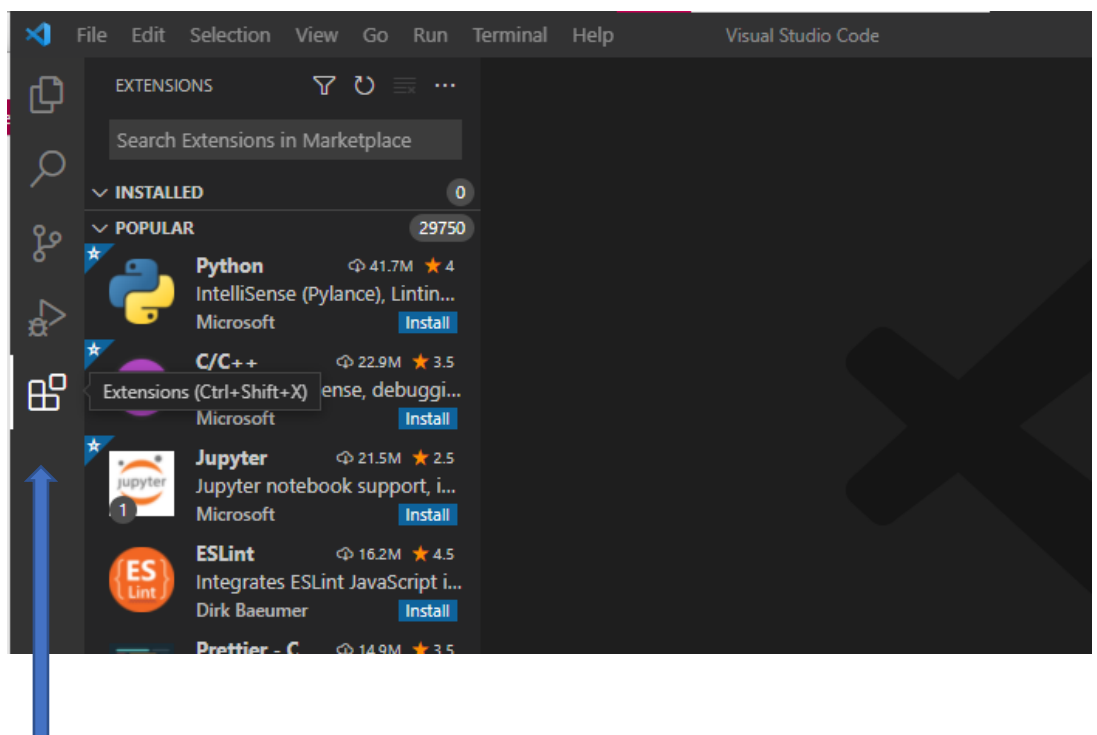


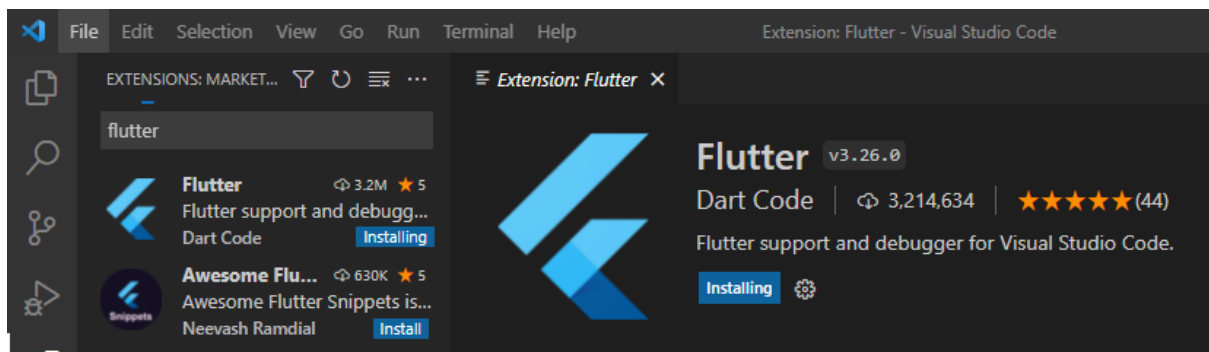
Instalasi VSCode tergolong mudah, cukup next next next next next finish, gitu aja.

Pada pertama kali menjalankan VSCode, kita diminta untuk memilih tema yang paling nyaman buat kita, disini saya memilih tema dark.



Close tab welcome dengan click tanda silang pada tab welcome, kemudian kita klik menu extension untuk menambahkan ekstensi yang diperlukan, untuk kebutuhan kali ini, kita perlu ekstensi flutter, masukkan flutter pada pencarian ekstensi, kemudian klik install.



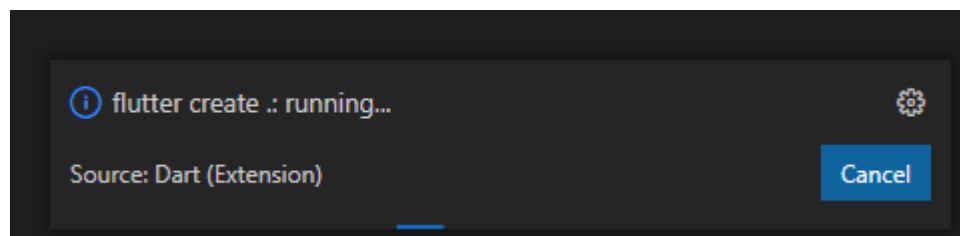
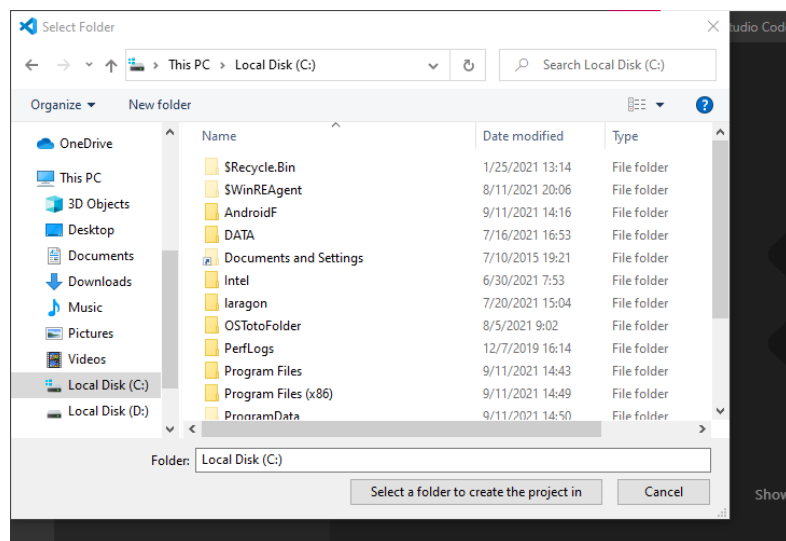


Sebelum kita membuat project, kita cek dulu dengan flutter doctor.

```
C:\Windows\System32>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 2.5.0, on Microsoft Windows [Version 10.0.19042.1165], locale en-US)
[✓] Android toolchain - develop for Android devices (Android SDK version 30.0.0)
[✓] Chrome - develop for the web
[!] Android Studio (not installed)
[✓] VS Code (version 1.60.0)
[✓] Connected device (2 available)

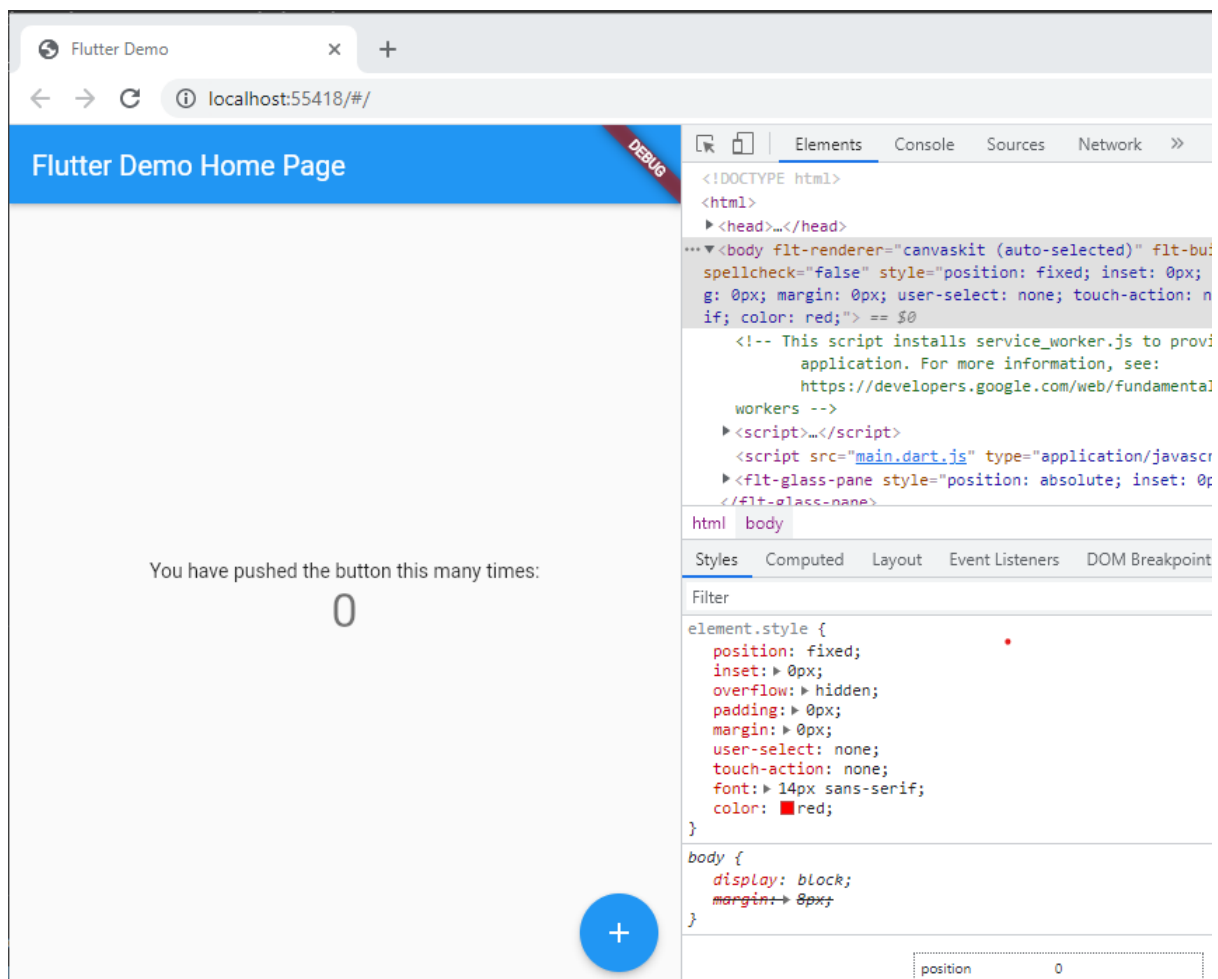
! Doctor found issues in 1 category.
```

Alhamdulillah, semua berjalan dengan baik, waktunya kita membuat project dengan flutter. Ketik **FI** kemudian cari **flut** pilih yang **New Application Project**, Pilih lokasi project dan beri nama projectnya, tunggu beberapa saat sampai project selesai dibuat.



```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({Key? key}) : super(key: key);
9
10  // This widget is the root of your application.
11  @override
12  Widget build(BuildContext context) {
13    return MaterialApp(
14      title: 'Flutter Demo',
15      theme: ThemeData(
16        // This is the theme of your application.
17        //
18        // Try running your application with "flutter run". You'll see the
19        // application has a blue toolbar. Then, without quitting the app, try
```

setelah selesai create, sekarang waktunya kita run dengan menekan tombol **F5** dan berhasil running di chrome.



Alhamdulillah selesai, Mohon maaf apabila ada kesalahan kata, semoga bermanfaat dan terima kasih.