

Install Flutter di Windows

Berikut adalah langkah-langkah untuk meng-*install* Flutter di Windows:

Langkah 1: Kunjungi Web Flutter

Silakan mengunjungi [web Flutter](#) kemudian pilih OS yang sesuai dengan perangkat kalian. Namun, dalam hal ini kita hanya akan membahas Windows, oleh karenanya bagi kalian yang menggunakan Windows silahkan untuk mengklik Windows.

Langkah 2: Cek *Requirements*

Setelah itu, kalian perlu melakukan cek persyaratan sistem sebelum men-*install* dan menjalankan Flutter. Mulai dari OS, *disk space*, hingga *tools* seperti [Windows PowerShell 5.0](#) dan [Git untuk Windows](#).

System requirements

To install and run Flutter, your development environment must meet these minimum requirements:

- **Operating Systems:** Windows 10 or later (64-bit), x86-64 based.
- **Disk Space:** 1.64 GB (does not include disk space for IDE/tools).
- **Tools:** Flutter depends on these tools being available in your environment.
 - [Windows PowerShell 5.0](#) or newer (this is pre-installed with Windows 10)
 - [Git for Windows 2.x](#), with the **Use Git from the Windows Command Prompt** option.

If Git for Windows is already installed, make sure you can run `git` commands from the command prompt or PowerShell.

Langkah 3: *Install* Flutter SDK

Kalian lakukan *restart* perangkat setelah mengunduh Git dan bersiap untuk mengunduh dan meng-*install* Flutter SDK (*Software Development Kit*). Kalian bisa [download di sini](#) untuk versi 3.7.8 (versi terbaru saat tulisan ini dibuat).

Setelah diunduh, *extract* folder **flutter** ke *path/directory* yang kalian inginkan. Ada baiknya kalian membuat folder baru yang terpisah dari *system drive* untuk menghindari masalah izin. Selanjutnya kalian klik folder **flutter**, cari dan buka file **flutter_console.bat**. Setelah itu kalian akan diarahkan menuju *command prompt* dan akan terlihat seperti gambar di bawah:

```
Flutter Console

#####  ##      ##  #####  #####  #####  #####
##      ##      ##  ##      ##      ##      ##
##      ##      ##  ##      ##      ##      ##
#####  ##      ##  ##      ##      ##      ##
##      ##      ##  ##      ##      ##      ##
##      ##      ##  ##      ##      ##      ##
##      #####  #####  ##      ##      ##      ##

WELCOME to the Flutter Console.
=====

Use the console below this message to interact with the "flutter" command.
Run "flutter doctor" to check if your system is ready to run Flutter apps.
Run "flutter create <app_name>" to create a new Flutter project.

Run "flutter help" to see all available commands.

Want to use an IDE to interact with Flutter? https://flutter.dev/ide-setup/

Want to run the "flutter" command from any Command Prompt or PowerShell window?
Add Flutter to your PATH: https://flutter.dev/setup-windows/#update-your-path

=====

D:\Development\flutter>
```

Console tersebut merupakan terminal Windows yang tersedia bagi pengembang untuk menjalankan perintah Flutter. Kalian bisa mengetikkan **“flutter”** untuk mendapatkan daftar perintah yang dapat dijalankan.

```
Flutter Console
Manage your Flutter app development.

Common commands:

flutter create <output directory>
  Create a new Flutter project in the specified directory.

flutter run [options]
  Run your Flutter application on an attached device or in an emulator.

Usage: flutter <command> [arguments]

Global options:
-h, --help                Print this usage information.
-v, --verbose              Noisy logging, including all shell commands executed.
                           If used with "--help", shows hidden options.
                           If used with "flutter doctor", shows additional
                           diagnostic information. (Use "-vv" to force verbose logging in those cases.)
-d, --device-id            Target device id or name (prefixes allowed).
--version                  Reports the version of this tool.
--suppress-analytics       Suppress analytics reporting when this command runs.

Available commands:
analyze                    Analyze the project's Dart code.
assemble                  Assemble and build Flutter resources.
attach                    Attach to a running app.
bash-completion            Output command line shell completion setup scripts.
build                     Build an executable app or install bundle.
channel                   List or switch Flutter channels.
clean                     Delete the build/ and .dart_tool/ directories.
config                    Configure Flutter settings.
create                    Create a new Flutter project.
devices                   List all connected devices.
doctor                    Show information about the installed tooling.
downgrade                 Downgrade Flutter to the last active version for the
current channel.
drive                     Run integration tests for the project on an attached
device or emulator.
emulators                 List, launch and create emulators.
format                    Format one or more Dart files.
gen-l10n                  Generate localizations for the current project.
install                   Install a Flutter app on an attached device.
logs                      Show log output for running Flutter apps.
precache                  Populate the Flutter tool's cache of binary artifacts.
pub                       Commands for managing Flutter packages.
```

Langkah 4: Perbarui *Path*

Selanjutnya kalian perlu memperbarui *path*. Caranya, di menu **start** ketik “env” dan kalian akan menemukan **Edit environment variables for your account**. Cara lainnya kalian bisa mengaksesnya melalui **Control Panel > System and Security > System > Advanced System Settings > Environment Variables....** Setelah itu, klik!

Di bagian **User variables** periksa apakah ada **Path**:

- Jika ada entrinya, tambahkan *path* lengkap ke flutter\bin menggunakan ; sebagai pemisah dari *values* yang ada.
- Jika entri tidak ada, buat variabel pengguna baru bernama Path dengan path lengkap ke flutter\bin sebagai *value*.

Sekarang, kalian perlu memeriksa apakah *framework* Flutter sudah bisa diakses secara global. Kalian bisa membuka terminal, katakanlah *Command Prompt* dan ketika **flutter**.

Apakah kalian menemukan daftar perintah yang didapatkan sebelumnya saat membuka file **.bat**?

Jika iya, tandanya kalian sudah berhasil memasang Flutter di sistem, tetapi jika tidak, mungkin kalian harus melakukan prosesnya kembali.

```
Command Prompt
C:\Users\Soumyadeep Das>flutter
Manage your Flutter app development.

Common commands:

flutter create <output directory>
    Create a new Flutter project in the specified directory.

flutter run [options]
    Run your Flutter application on an attached device or in an emulator.

Usage: flutter <command> [arguments]

Global options:
-h, --help                Print this usage information.
-v, --verbose              Noisy logging, including all shell commands executed
                           .
                           If used with "--help", shows hidden options. If used
                           with "flutter doctor", shows additional diagnostic i
nformation. (Use "-vv" to force verbose logging in those cases.)
-d, --device-id            Target device id or name (prefixes allowed).
    --version              Reports the version of this tool.
    --suppress-analytics   Suppress analytics reporting when this command runs.

Available commands:
analyze                    Analyze the project's Dart code.
assemble                  Assemble and build Flutter resources.
attach                    Attach to a running app.
bash-completion            Output command line shell completion setup scripts.
build                     Build an executable app or install bundle.
channel                   List or switch Flutter channels.
clean                     Delete the build/ and .dart_tool/ directories.
config                    Configure Flutter settings.
create                    Create a new Flutter project.
devices                   List all connected devices.
doctor                    Show information about the installed tooling.
downgrade                 Downgrade Flutter to the last active version for the current
channel.
drive                     Run integration tests for the project on an attached device
or emulator.
emulators                 List, launch and create emulators.
format                    Format one or more Dart files.
gen-l10n                  Generate localizations for the current project.
install                   Install a Flutter app on an attached device.
logs                      Show log output for running Flutter apps.
precache                  Populate the Flutter tool's cache of binary artifacts.
pub                       Commands for managing Flutter packages.
run                       Run your Flutter app on an attached device.
screenshot                Take a screenshot from a connected device.
```


Langkah 5: Jalankan Flutter Doctor

Selanjutnya, kalian perlu memeriksa apakah ada sesuatu yang hilang atau kurang dan perlu di-*install*. Untuk melakukannya, buka *Command Prompt* dan ketik “**Flutter Doctor**” untuk melihat *requirement* yang dibutuhkan.

[~] Android toolchain - develop for Android devices

- Android SDK at D:\Android\sdk

X Android SDK is missing command line tools; download from <https://goo.gl/XxQghQ>

- Try re-installing or updating your Android SDK, visit <https://docs.flutter.dev/setup/#android-setup> for detailed instructions.

Berdasarkan contoh di atas, Android SDK merupakan *tools* yang harus kita *install* selanjutnya.

Tapi, memungkinkan pula kalian mendapatkan hasil seperti di bawah ini:

Doctor summary (to see all details, run flutter doctor -v):

[~] Flutter (Channel stable, v1.2.1, on Microsoft Windows [Version 10.0.17134.706], locale en-US)

[~] Android toolchain - develop for Android devices (Android SDK version 28.0.3)

[~] Android Studio (version 3.2)

[~] VS Code, 64-bit edition (version 1.29.1)

[!] Connected device

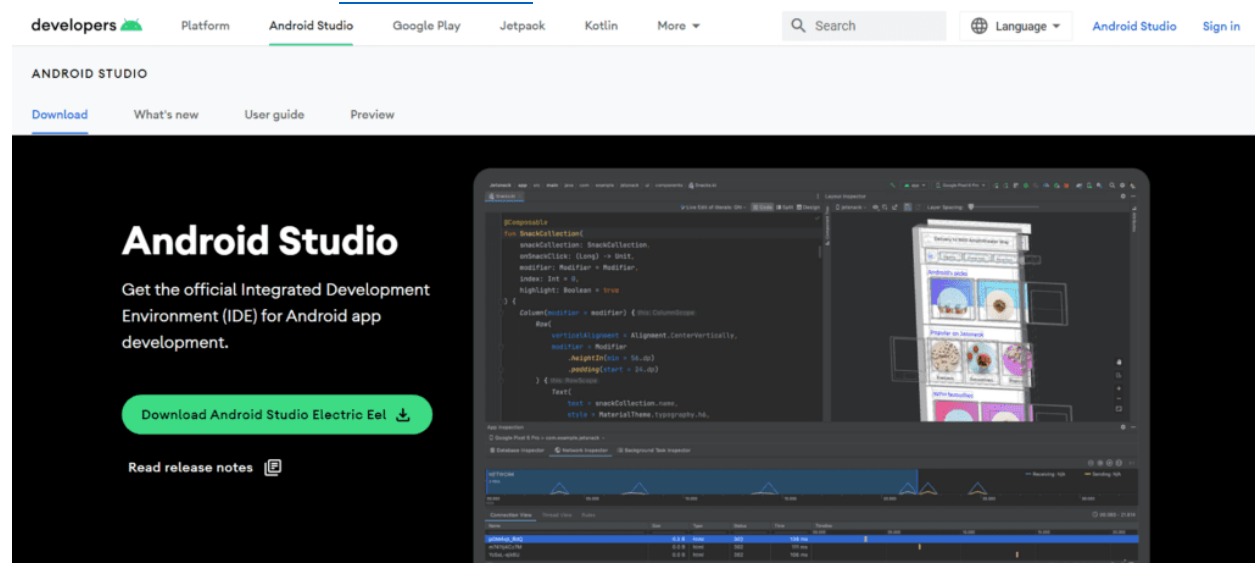
! No devices available

! Doctor found issues in 1 category.

Dalam laporan tersebut, semua *tools* sudah terpasang namun *device* belum terhubung. Untuk mengatasi masalah tersebut, kalian bisa menghubungkan perangkat Android melalui USB atau menjalankan emulator Android.

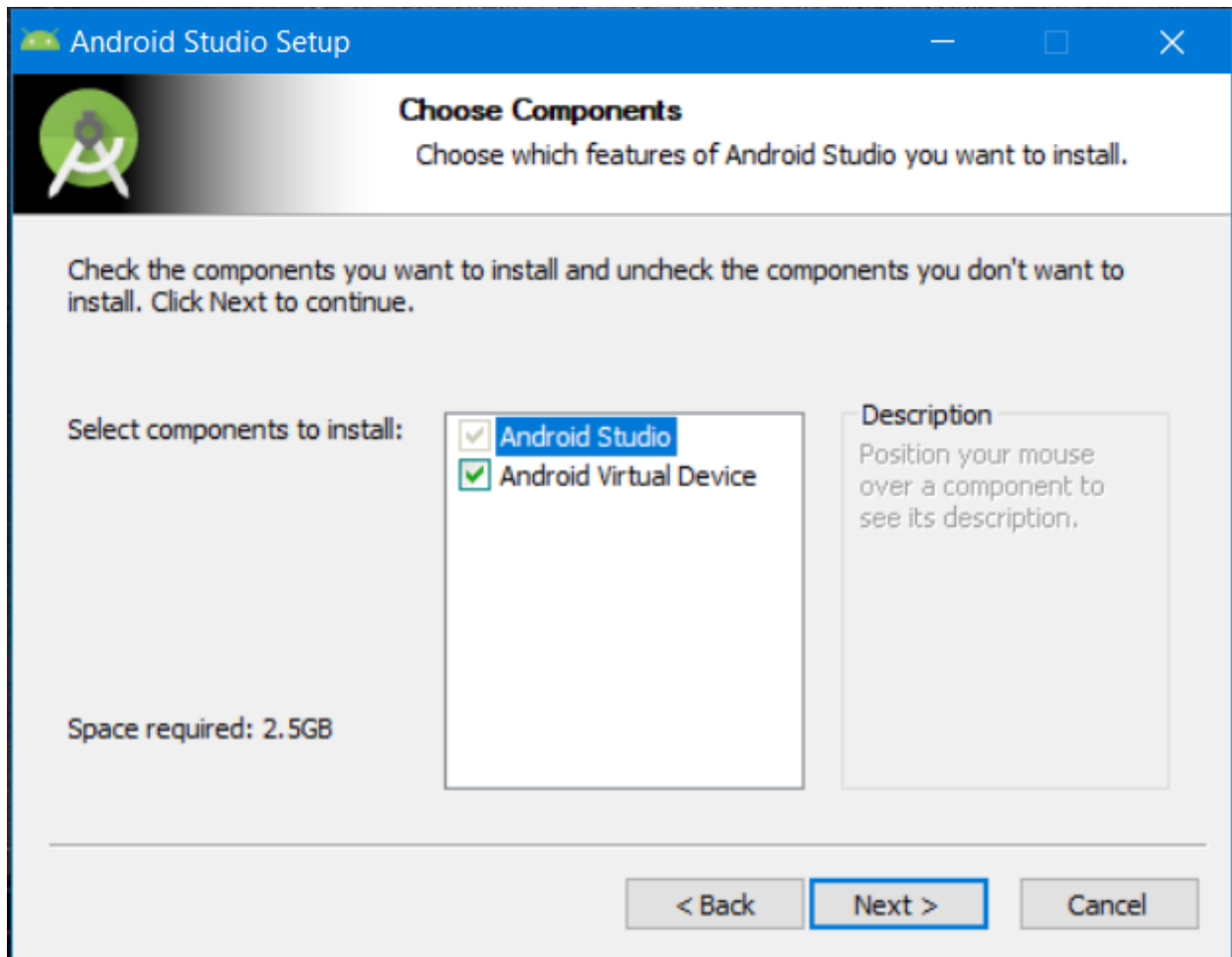
Langkah 6: Install Android Studio

Selanjutnya, kalian perlu memasang Android Tools dan emulator Android. *Tools* ini dapat kalian unduh melalui link resmi [Android Studio](#) dan klik ‘**Download Android Studio**’.

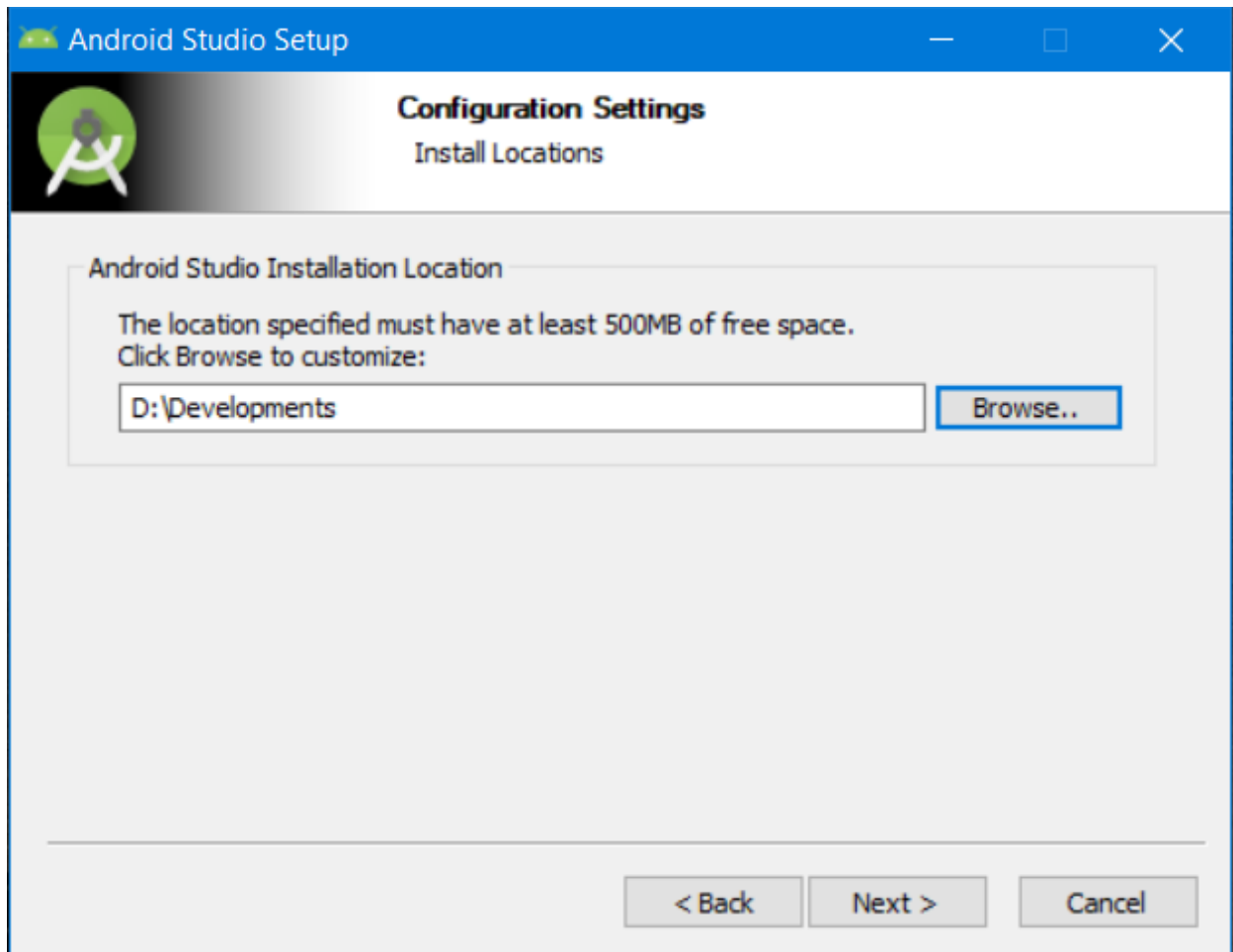


Setelah selesai mengunduh, selanjutnya lakukan proses instalasi.

Di dalam **Components**, pastikan Android Studio dan Android Virtual Devices sudah diceklis dan kemudian **next**.



Pilih direktori yang kalian inginkan untuk menyimpan hasil instalasi. Setelah dipilih, klik **next**.



Kemudian, klik **Install** dan tunggu beberapa saat hingga proses instalasi selesai. Klik **Launch Android Studio** kemudian klik **Finish**.
Di halaman awal, klik **Next > Custom > Next**.



Welcome

Android Studio

Welcome! This wizard will set up your development environment for Android Studio.
Additionally, the wizard will help port existing Android apps into Android Studio
or create a new Android application project.

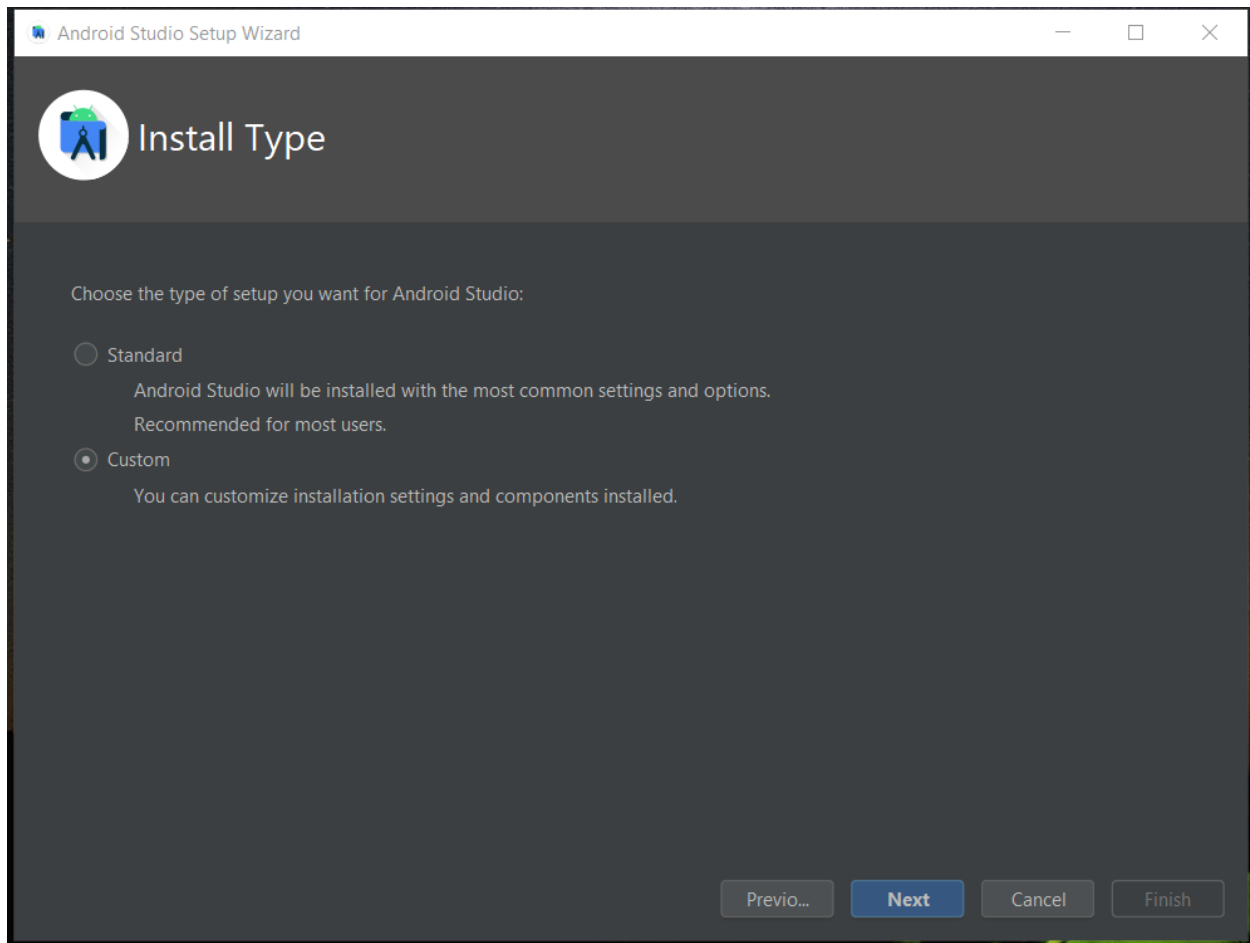


Previo...

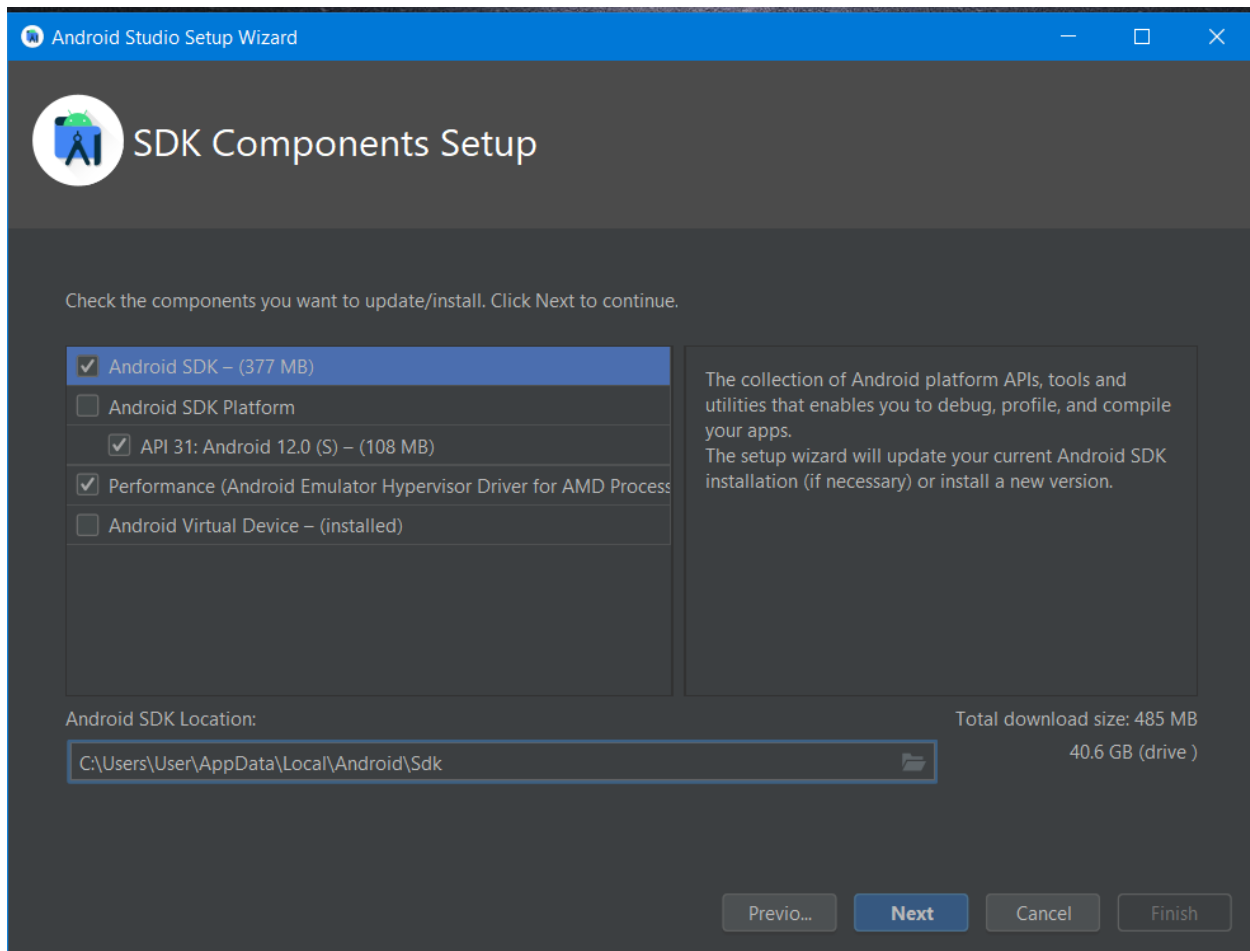
Next

Cancel

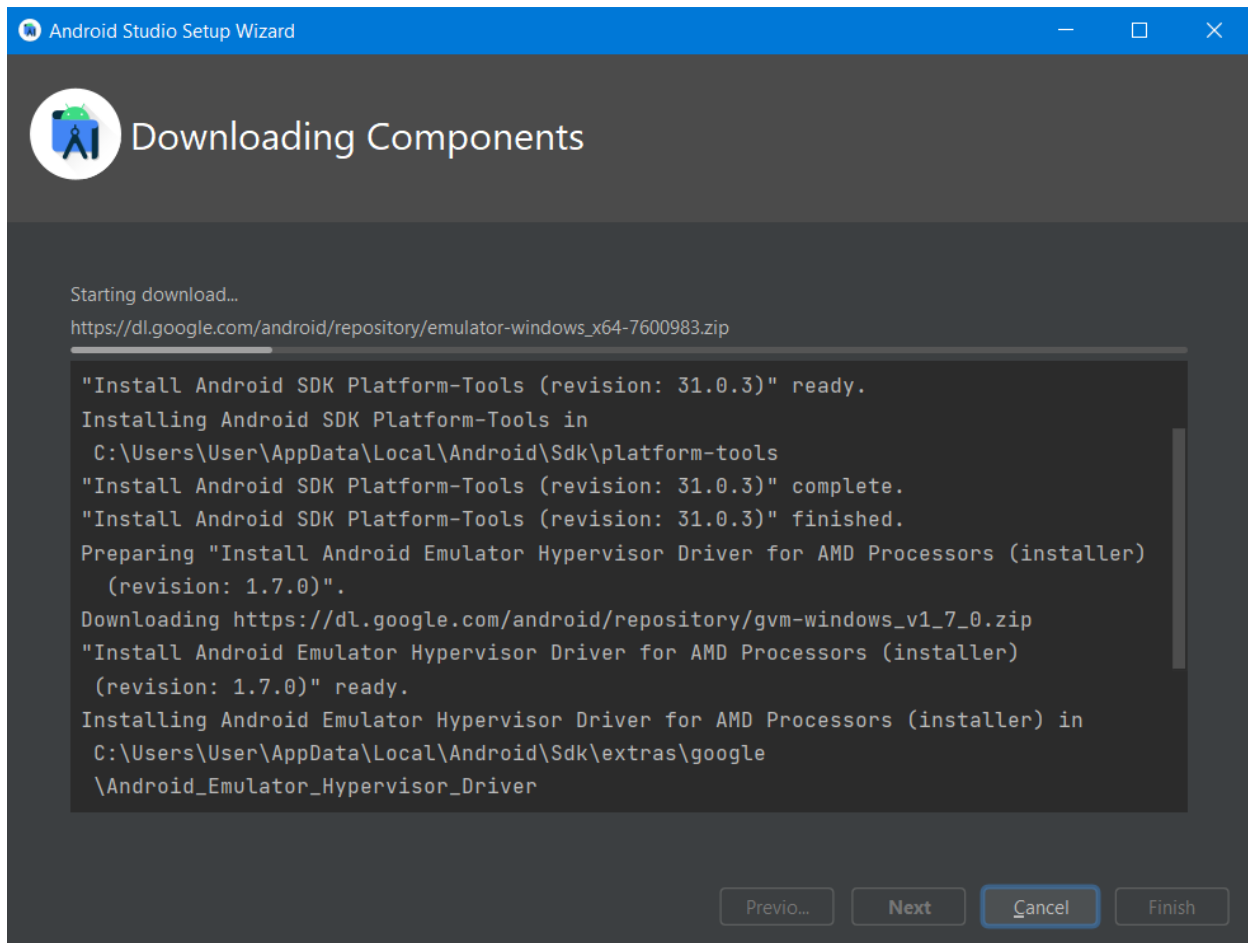
Finish



Langkah selanjutnya cukup penting dan krusial, pastikan kalian ceklis sebagaimana gambar di bawah ini:



Setelah itu, klik **Finish** dan Android Studio akan memasang semua kebutuhan kalian untuk menjalankan proyek menggunakan Flutter. Agak lama sih ya, tapi tunggu aja.



Sekarang, kita sudah siap untuk membuat proyek di Android Studio dan menjalankannya baik di perangkat Android langsung maupun emulator.

Langkah 7: Set Up Emulator Android

Ada dua cara yang bisa kalian lakukan, apakah menggunakan perangkat Android langsung atau menggunakan emulator.

Jika menggunakan Android langsung, kunjungi [official docs page](#) dan ikuti langkahnya secara tepat seperti dalam gambar di bawah. Kalian unduh [Google USB Driver](#) dan pasang sesuai dengan petunjuknya.

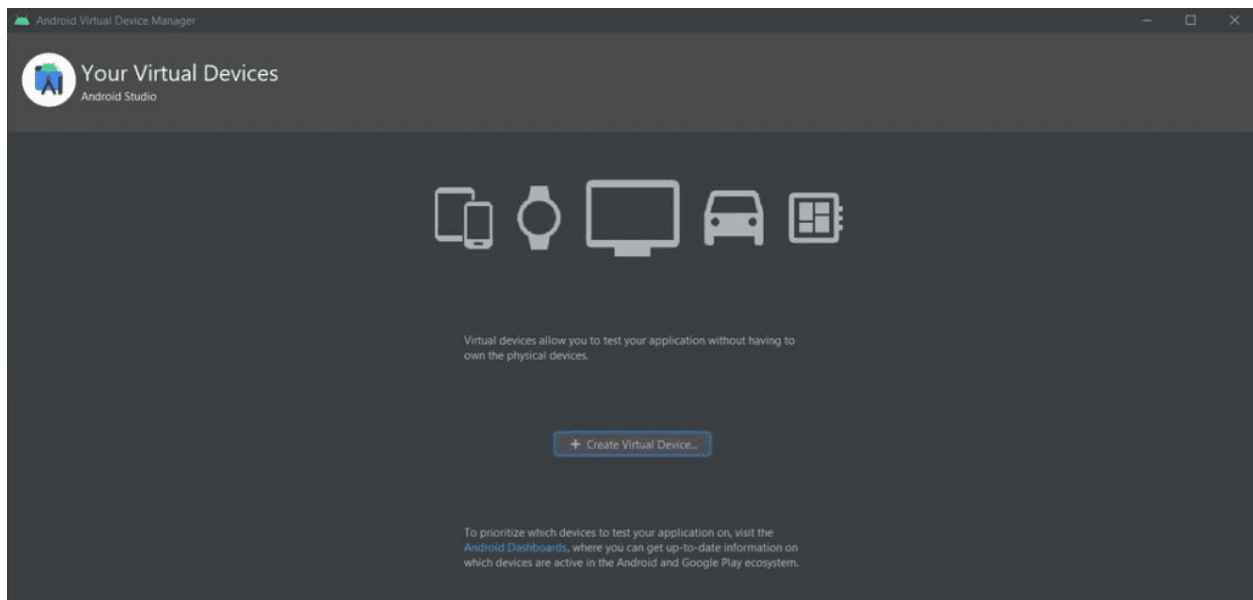
Set up your Android device

To prepare to run and test your Flutter app on an Android device, you need an Android device running Android 4.1 (API level 16) or higher.

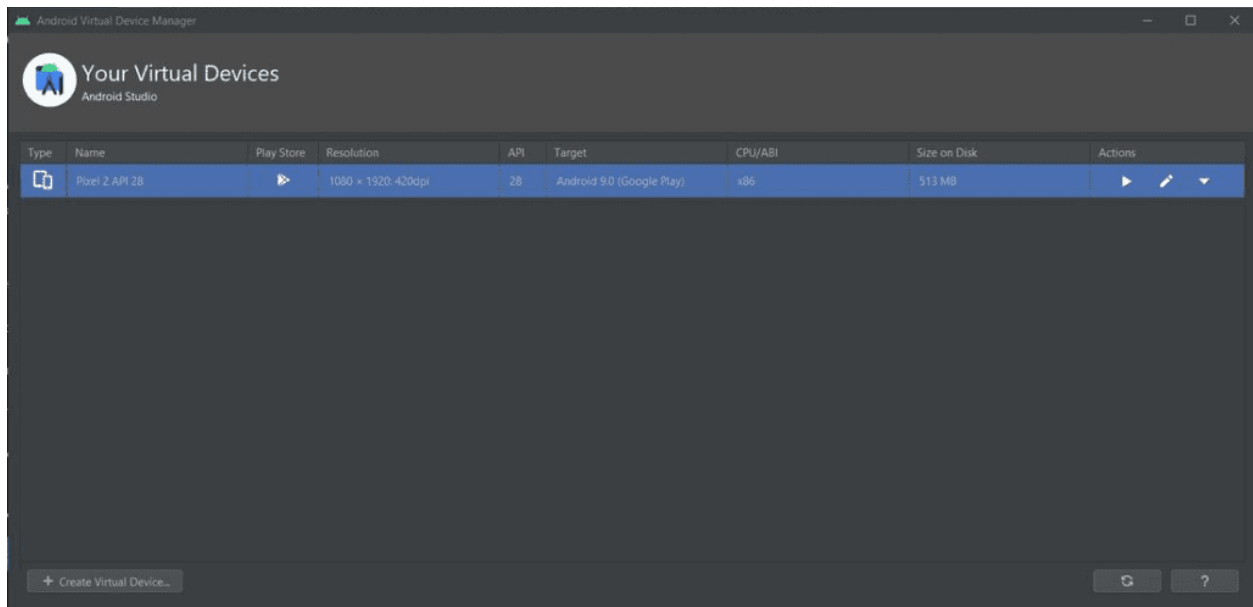
1. Enable **Developer options** and **USB debugging** on your device. Detailed instructions are available in the [Android documentation](#).
2. Windows-only: Install the [Google USB Driver](#).
3. Using a USB cable, plug your phone into your computer. If prompted on your device, authorize your computer to access your device.
4. In the terminal, run the `flutter devices` command to verify that Flutter recognizes your connected Android device. By default, Flutter uses the version of the Android SDK where your `adb` tool is based. If you want Flutter to use a different installation of the Android SDK, you must set the `ANDROID_ID_SDK_ROOT` environment variable to that installation directory.

Jika menggunakan emulator, langkah yang harus dilakukan adalah:

- Buka **Android Studio**.
- Di menu bagian paling atas, klik **Tools > SDK Manager**.
- Verifikasi bahwa kalian sudah memiliki SDK versi terbaru.
- Di bagian **SDK Tools**, jangan lupa untuk periksa Google USB Driver untuk dihubungkan ke perangkat Android. Setelah itu, klik **Apply** dan **Ok** untuk memulai proses instalasi.
- Untuk melihat emulator pertama kalian, buka Android Studio dan klik **Tools > AVD Manager**.



- Klik **Create Virtual Device ...** , pilih perangkat dan dimensi ukurannya sesuai keinginan kalian.



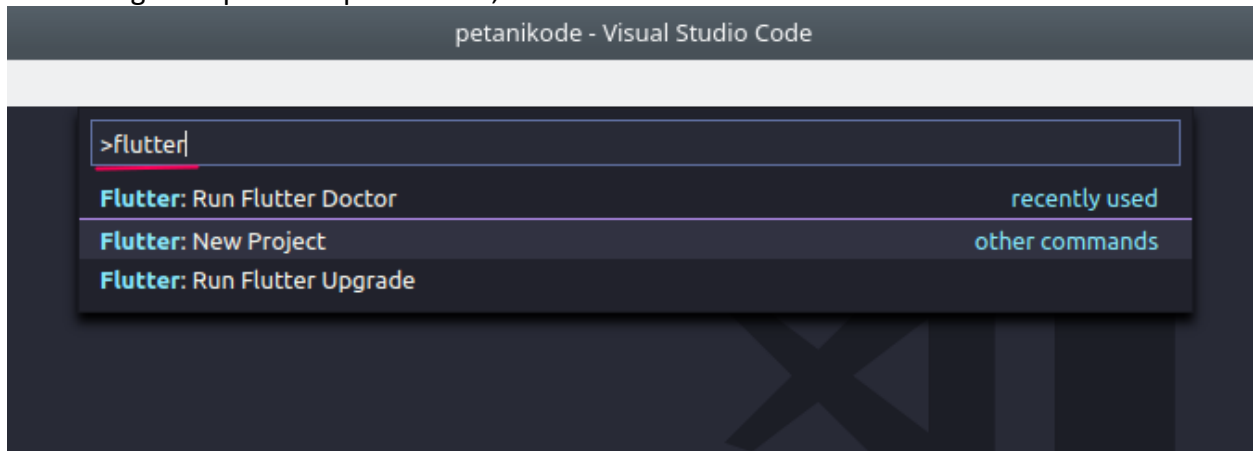
Akhirnya, kalian telah menyelesaikan cara *install* Flutter dan memiliki *framework* Flutter yang dapat dijalankan begitupula dengan perangkat atau emulatoarnya. Selamat mencoba!

FLUTTER

Setelah menginstal extension, kita akan bisa menggunakan beberapa perintah seperti:

1. Membuat Project Baru
2. Menjalankan perintah **flutter doctor**
3. Menjalankan Update

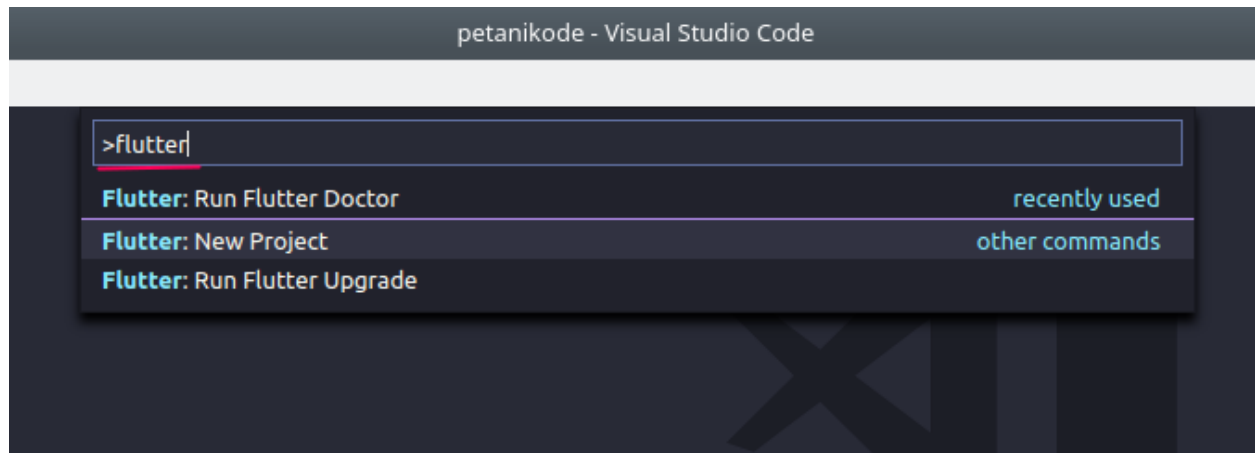
Cara mengakses perintah-perintah ini, tekan **Ctrl+Shift+P** lalu ketik **flutter**.



2. Membuat Project Flutter di VS Code

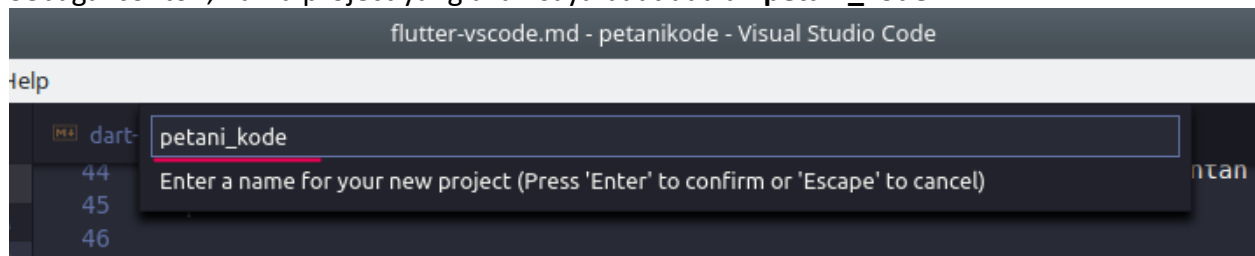
Cara membuat project Flutter di VS Code:

Tekan tombol **Ctrl+Shift+P**, lalu pilih **New Project**.

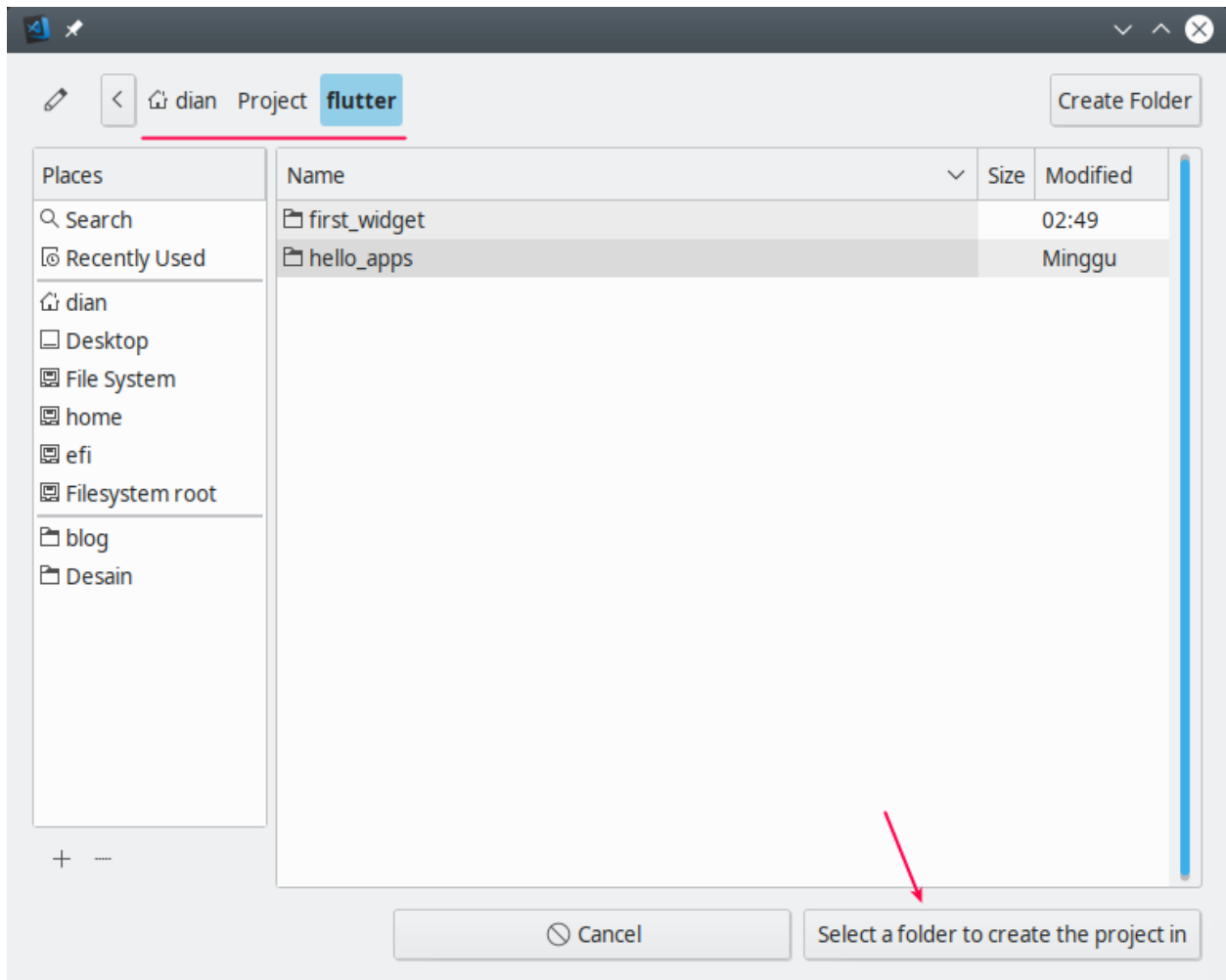


Kemudian kita, akan diminta menentukan nama project-nya. Isi namanya dengan huruf kecil dan pisah dengan garis bawah (*underscore*) jika namanya lebih dari dua suku kata.

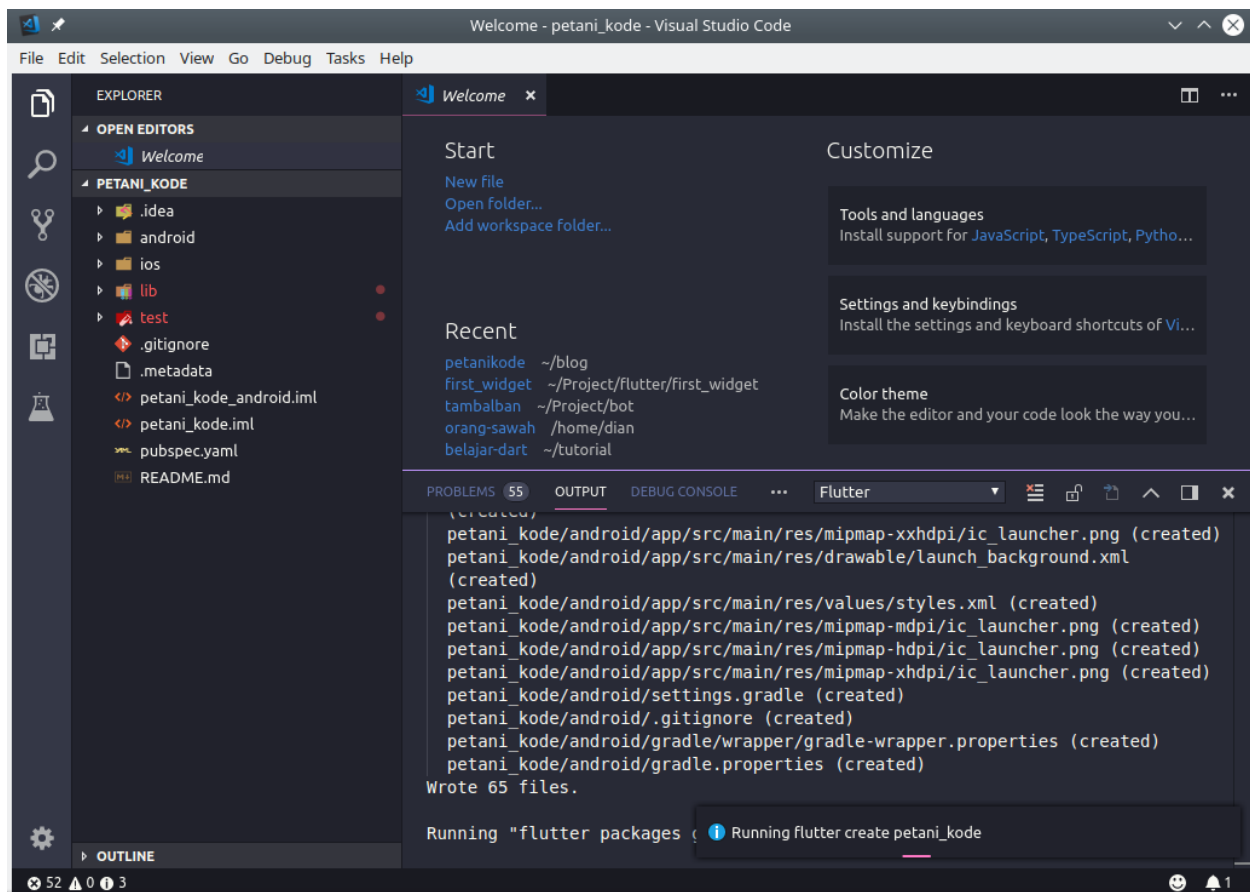
Sebagai contoh, nama project yang akan saya buat adalah **petani_kode**.



Kemudian kita akan diminta untuk menentukan lokasi penyimpanan project. Kita bisa menyimpan di tempat yang diinginkan.



Tunggulah proses pembuatannya sampai selesai...



3. Emulator untuk Flutter

Ada tiga macam emulator yang bisa digunakan untuk menjalankan aplikasi Flutter:

1. Emulator bawaan Android Studio
2. Smartphone fisik (HP sendiri)
3. Genymotion

Silakan pilih mau pakai yang mana. Saya sendiri pernah mencoba menggunakan Emulator Android Studio dan Smartphone fisik.

Menurut saya, yang paling enak menggunakan hp sendiri. Karena dapat mengurangi resource komputer dan mudah digunakan, tinggal colok dengan kabel data dan Flutter akan langsung mendeteksinya.

Tapi ada kekurangannya, yaitu: API yang bisa dipakai akan bergantung dari versi OS HP itu saja. Misal:

Kamu ingin mencoba menggunakan API yang ada di level 26 (Oreo), sedangkan OS smartphone-mu masih level 19 (Kitkat); ya... tidak akan bisa.

Bagaimana caranya agar bisa?

Kita bisa menggunakan emulator bawaan Android Studio atau Genymotion.

Karena dengan emulator ini, kita bisa menginstall OS versi berapapun yang kita inginkan.

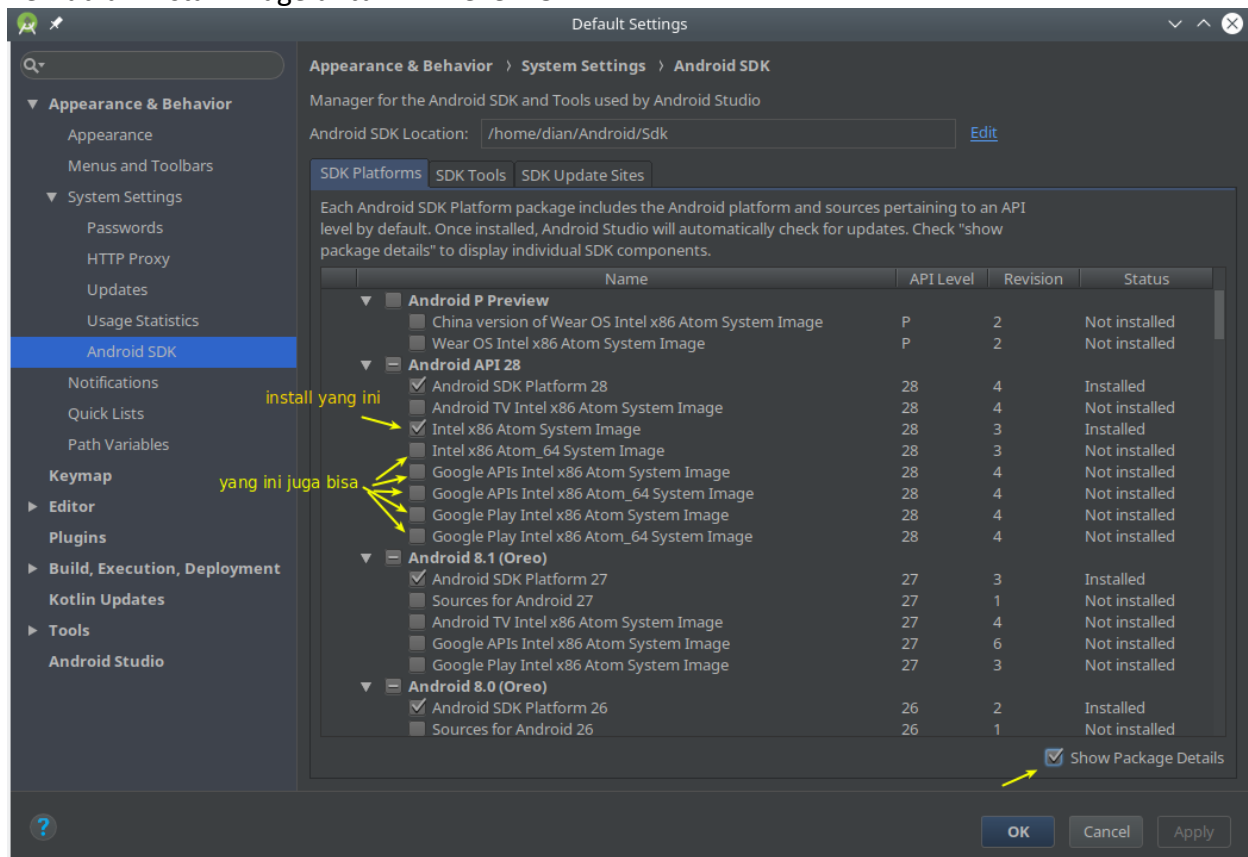
Nah, buat yang ingin menggunakan emulator android studio, saya merekomendasikan menginstal image API level 28.

Silakan ikuti langkah-langkah berikut untuk menginstalnya...

Buka Android Studio, Klik ikon Android SDK Manager:

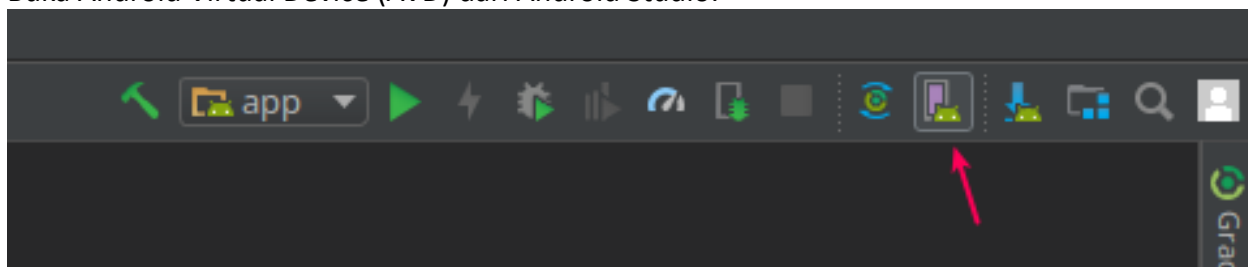


Kemudian install image untuk API Level 28:

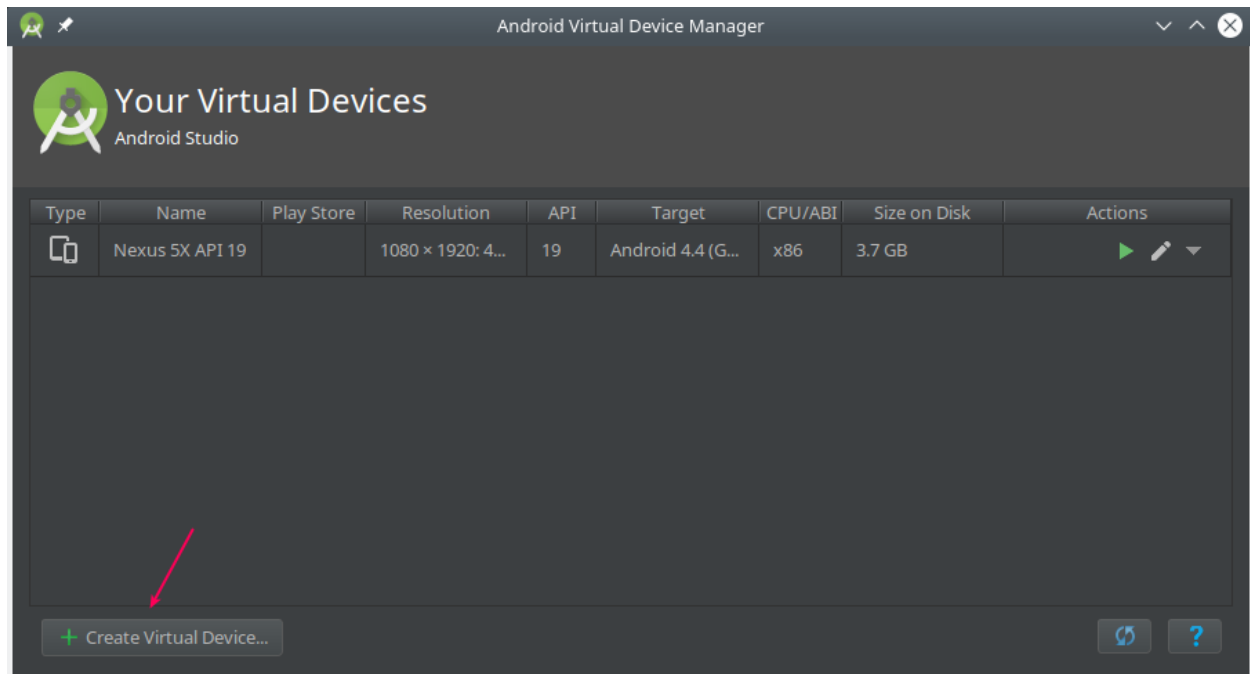


Setelah terinstal, buatlah emulator baru dengan image tersebut.

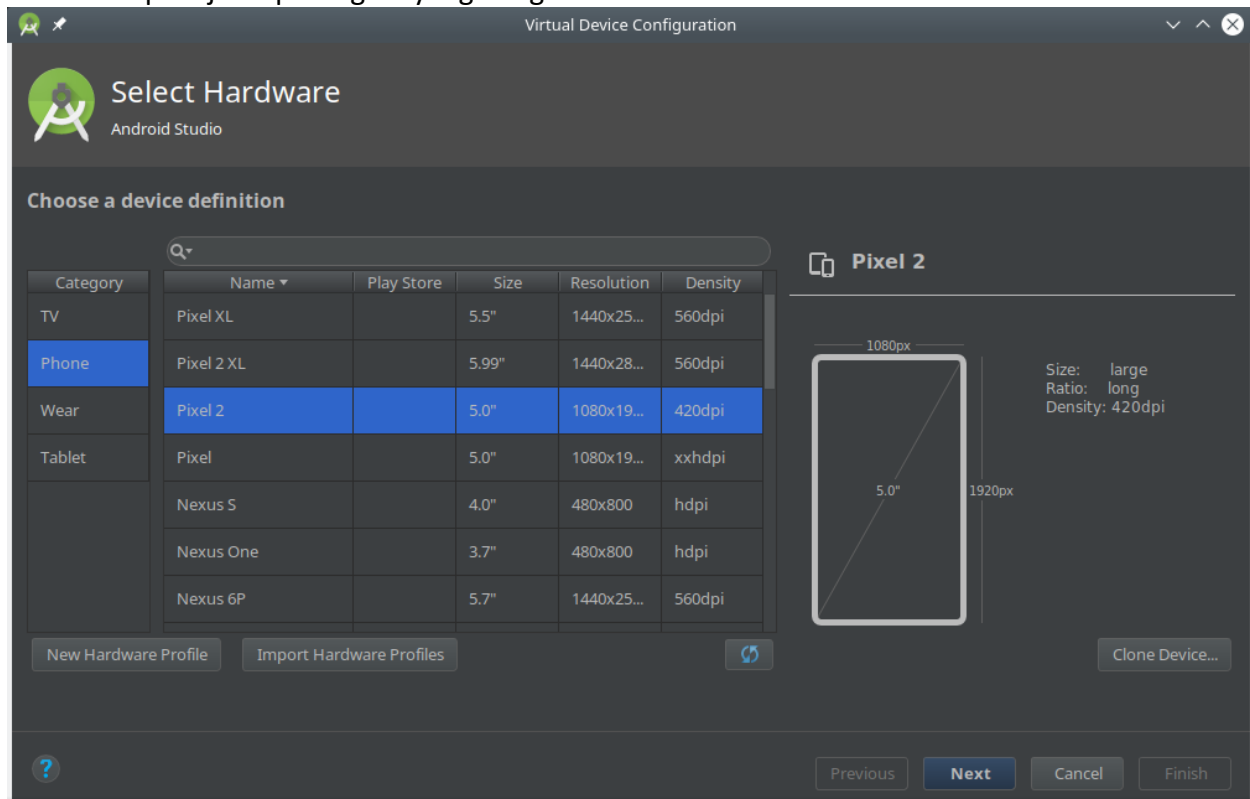
Buka Android Virtual Device (AVD) dari Android Studio:



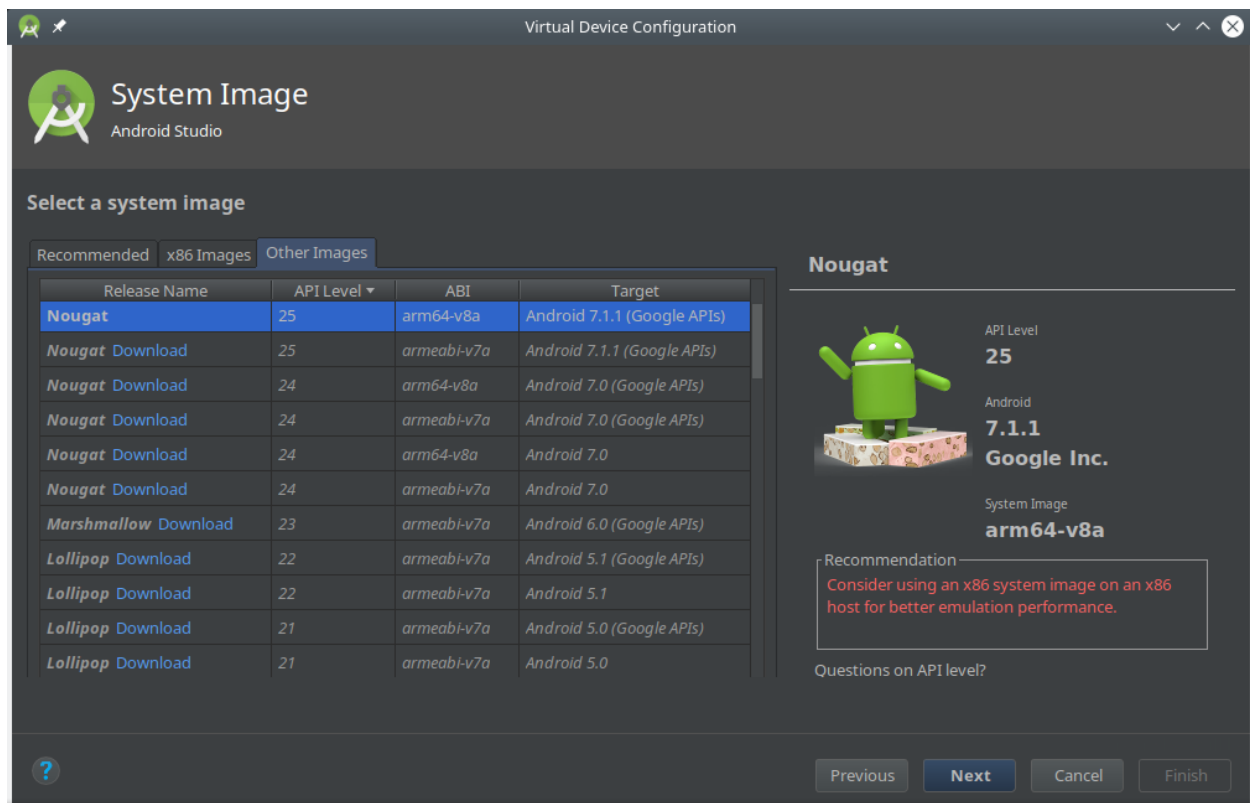
Klik **Create New Virtual Device**:



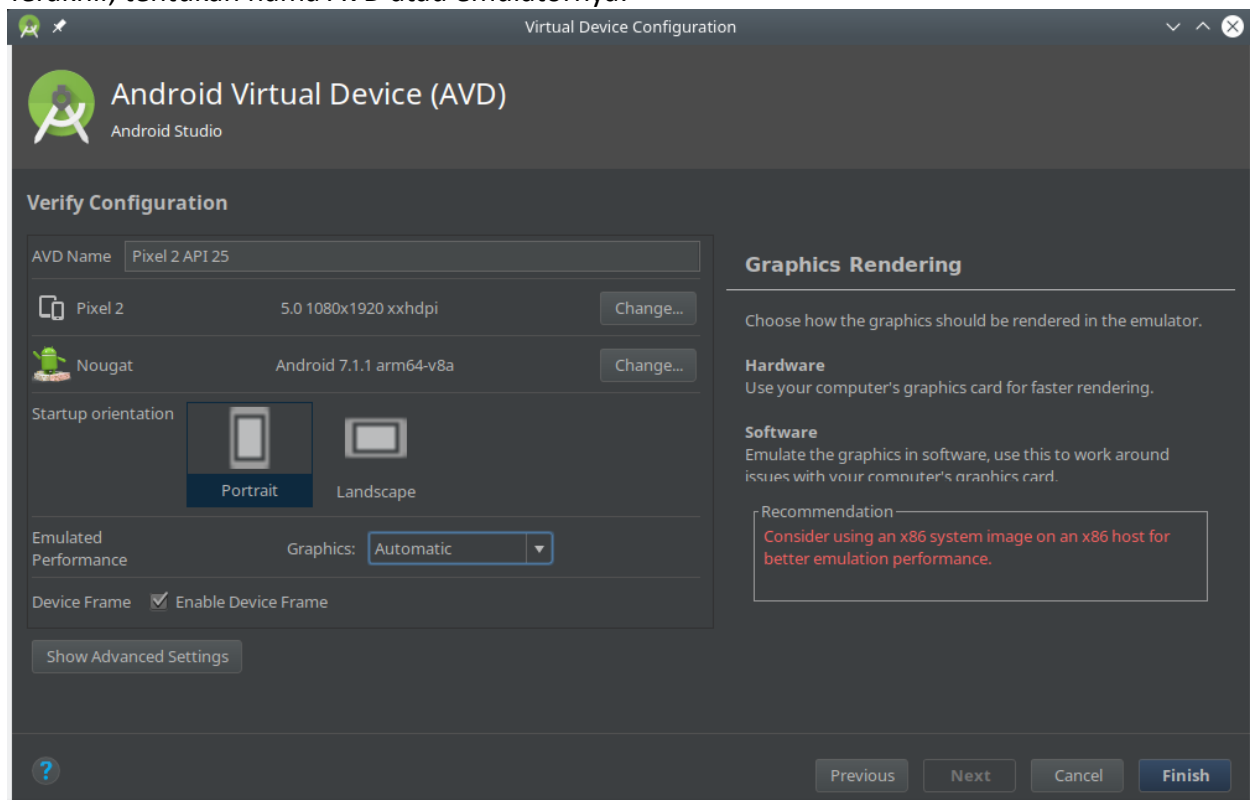
Kemudian pilih jenis perangkat yang diinginkan:



Berikutnya pilih system image yang akan digunakan:



Terakhir, tentukan nama AVD atau emulatorenya:



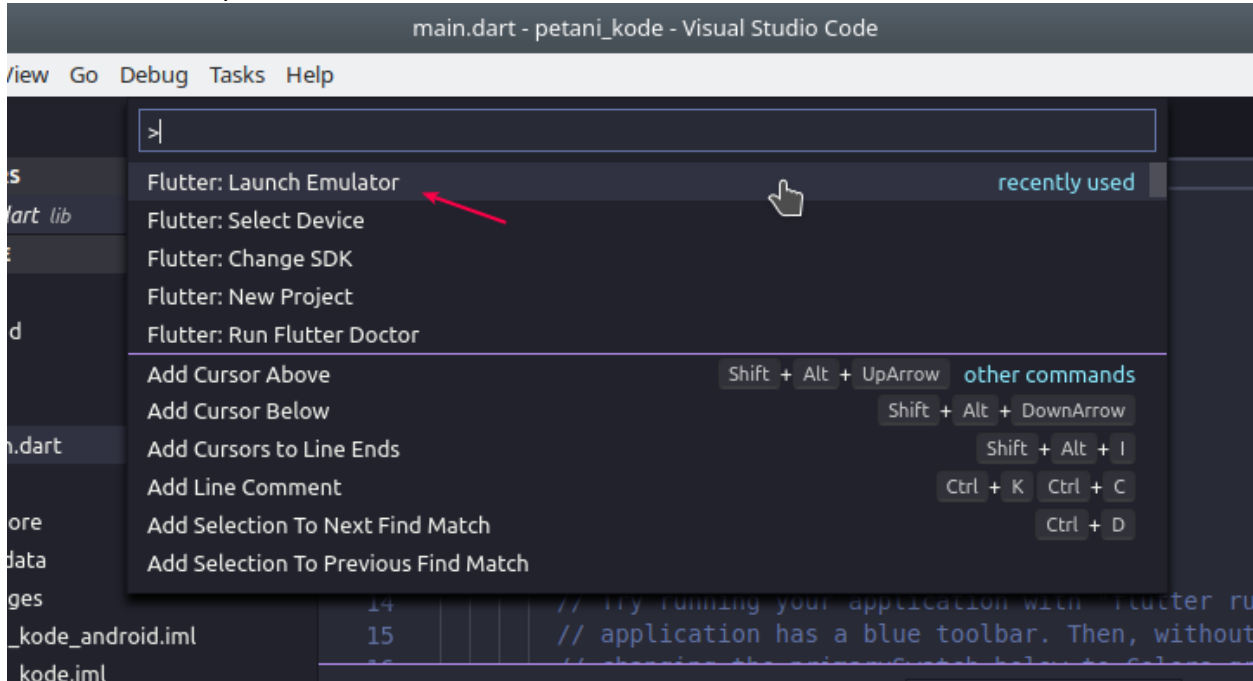
Dengan demikian...

...kita sudah bisa menggunakan android emulator untuk menjalankan aplikasi Flutter.

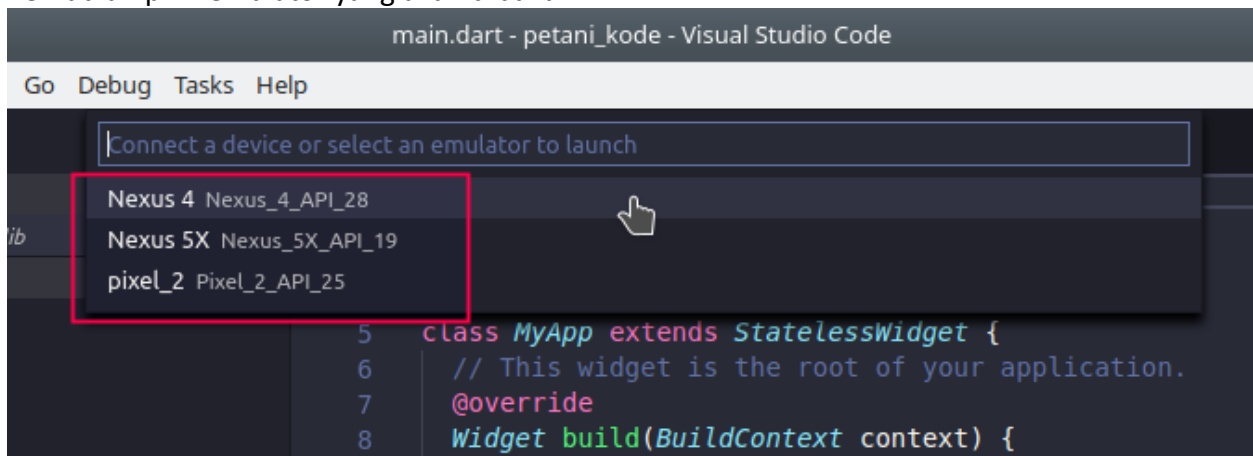
Kita bisa menutup Android Studio, karena sudah tidak dibutuhkan. Kecuali, kalau ingin ngoding di Android Studio.

4. Menjalankan Aplikasi Flutter

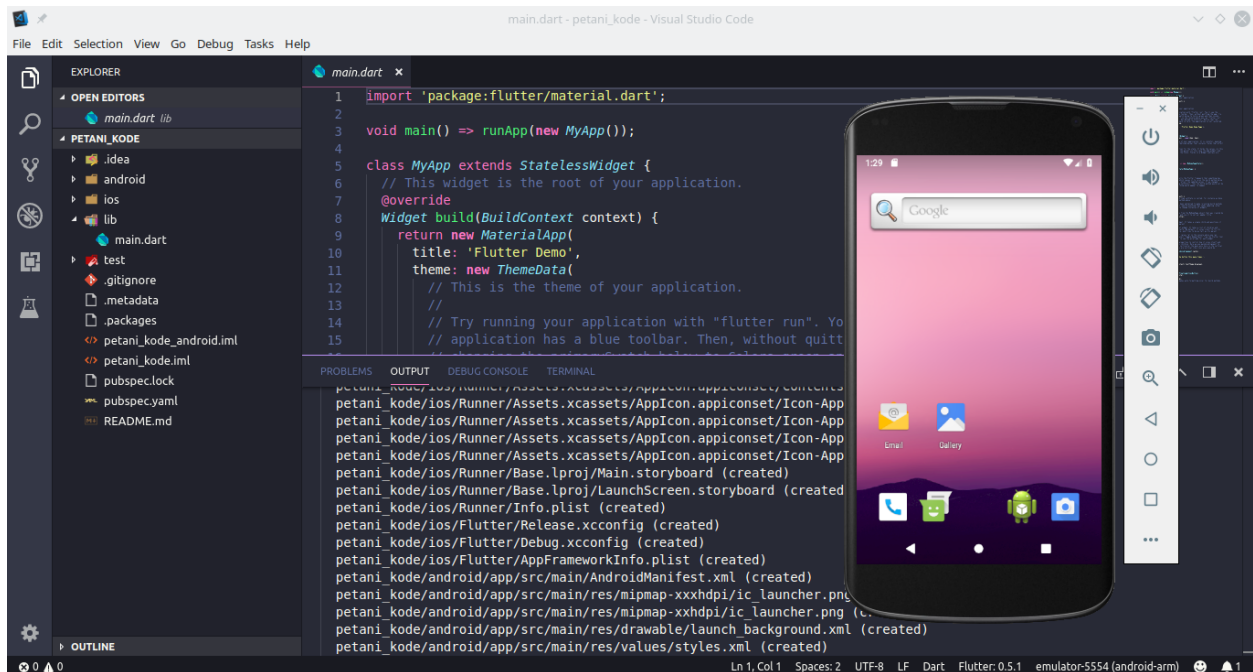
Sekarang, kembali ke project Flutter yang di VS Code. Coba tekan **Ctrl+Shift+P** lalu ketik **flutter** dan pilih **Launch Emulator**.



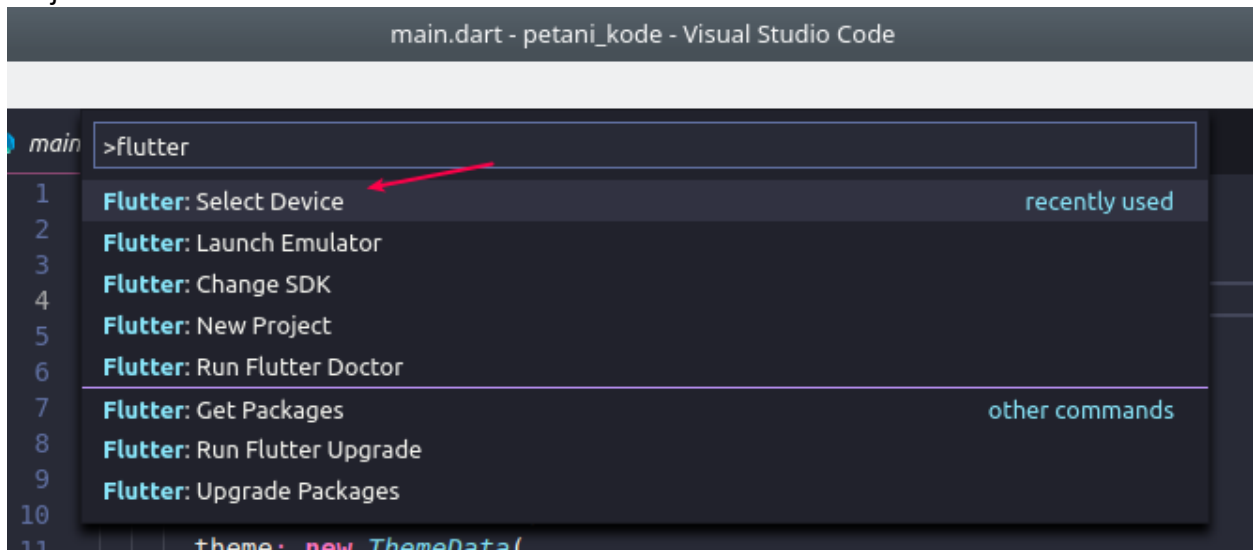
Kemudian pilih emulator yang akan dibuka.



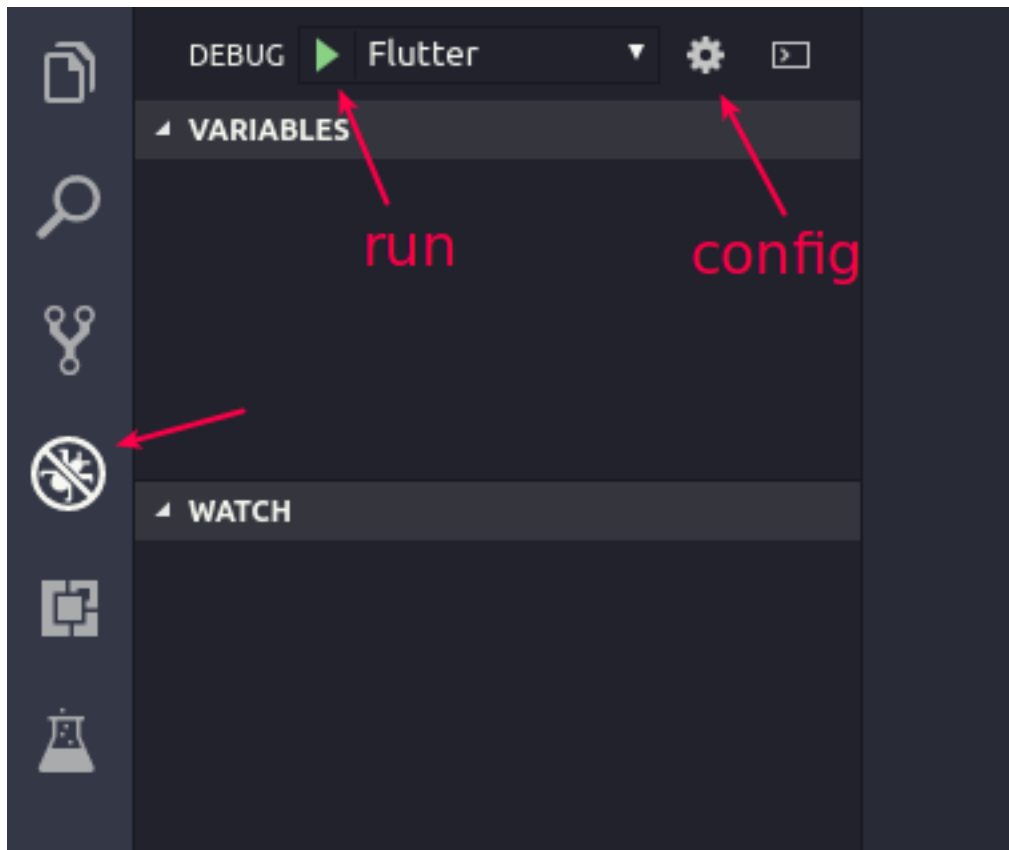
Tunggu beberapa saat sampai emulatornya terbuka seperti ini:



Sekarang, kita sudah siap menjalankan aplikasi Flutter-nya di emulator. Tekan lagi tombol **Ctrl+Shift+P**, kemudian pilih **Select Device** untuk memilih emulator yang sedang berjalan.



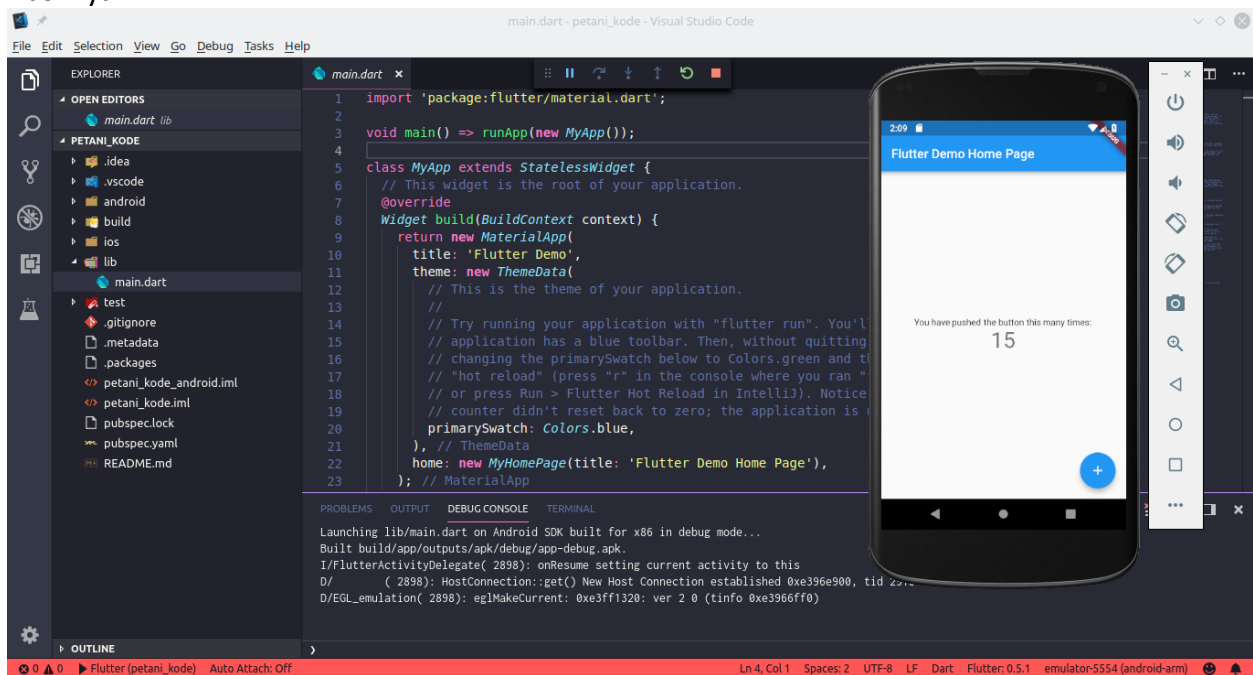
Terakhir, masuk ke menu **Debug** kemudian klik **Run**.



Jika belum bisa, coba klik **Config** terlebih dahulu, lalu coba **Run** lagi...

Tunggulah proses *build* apk-nya, biasanya untuk *run* pertama kali akan memakan waktu yang cukup lama. Karena harus men-download beberapa *dependency* untuk aplikasinya.

Hasilnya:



Selamat

Kita sudah berhasil membuat aplikasi Flutter melalui VS Code.

bb