

DOKUMEN PETUNJUK INSTALASI FRAMEWORK FLUTTER DALAM SISTEM OPERASI WINDOWS



Oleh:

Ihsan Fajari

8 Agustus 2018

Daftar Isi

| | |
|--|---|
| 1. Latar Belakang | 3 |
| 2. Persiapan Sebelum Instalasi..... | 4 |
| 2.1. Persyaratan Sistem Flutter | 4 |
| 2.2. Unduh SDK Flutter..... | 5 |
| 2.3. <i>Update path</i> dan ANDROID_HOME | 5 |
| 2.3.1. <i>Update path</i> | 5 |
| 2.3.2. ANDROID_HOME | 5 |
| 2.4. Jalankan ‘flutter doctor’ di <i>console</i> | 6 |
| 2.5. Mempersiapkan Android Studio..... | 6 |
| 2.5.1. Instalasi | 6 |
| 2.5.2. Mempersiapkan Perangkat Android | 6 |
| 2.5.3. Mempersiapkan <i>Emulator</i> Android | 7 |
| 2.6. <i>Instalasi</i> Gradle..... | 9 |
| 3. Persiapan <i>Editor</i> Android Studio | 9 |
| 4. Mencoba Menjalankan <i>Project</i> Flutter. | 9 |

1. Latar Belakang

Sebuah *framework* memang selayaknya memiliki petunjuk instalasinya. Pembuatan petunjuk instalasi ini bermaksud untuk memudahkan *user* dalam proses instalasi dengan memberi petunjuk yang bertahap. Semua ini bertujuan untuk menghindari dari segala jenis kemungkinan yang dapat menyebabkan *error* yang membuat *framework* tidak bisa dipakai dengan selayaknya bahkan tidak bisa dipakai sama sekali.

Dalam *Framework* Flutter yang masih tahap *Beta* versi 0.6.0, dia sudah memiliki petunjuk instalasinya sendiri. Untuk *framework* yang tahap pengembangannya masih *Beta*, petunjuknya sudah cukup jelas dan mudah bagi *user* yang ingin memakainya. Sayangnya dalam petunjuk instalasinya tersebut, masih ada beberapa kasus tertentu (spesifik pada beberapa perangkat dengan ketentuan sistem operasi sendiri) yang belum tertangani. *User* mau tidak mau perlu eksplorasi sendiri menyelesaikan masalah tersebut. Dampaknya dapat menurunkan *value* dari penggunaan *framework* ini sendiri sehingga membuat *user* mengurungkan niatnya untuk menggunakan *framework* ini karena tidak bisa menyelesaikan proses instalasinya sendiri.

Maka dari itu disini saya bermaksud untuk membantu menangani permasalahan yang muncul diluar dari dokumen instalasi ini <https://flutter.io/get-started/> dengan dibuatkan dokumen. Isinya berupa informasi tambahan perihal instalasi *Framework* Flutter berdasarkan pengalaman yang saya peroleh saat instalasi *framework* ini.

2. Persiapan Sebelum Instalasi

Ada beberapa tahapan dan *requirement* sebelum menggunakan Flutter.

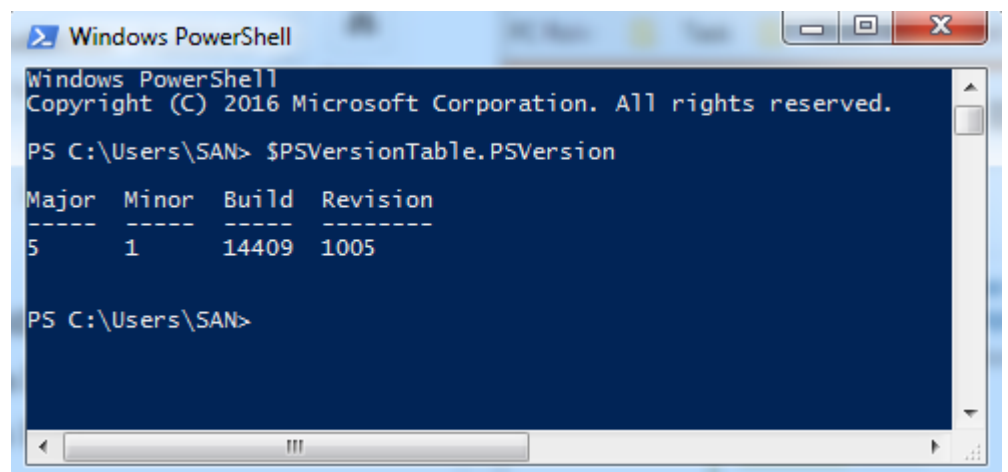
2.1. Persyaratan Sistem Flutter

Untuk menjalankannya, terdapat persyaratan minimum yang harus dipenuhi, yakni :

- Sistem Operasi: Windows 7 SP1 atau lebih (64-bit), Windows 8, Windows 8.1, dan Windows 10.
- Partisi Kosong Pada Hardisk: Setidaknya diperlukan 400 MB hanya untuk sistem Flutternya itu sendiri, ini belum termasuk SDK dan IDE/tools (Android Studio, Emulator, dll) yang totalnya bisa hingga 10GB lebih.
- *Tools*:

- a. PowerShell 5.0 atau yang lebih baru. Untuk pengguna Windows 10, Anda tidak perlu repot – repot lagi menyiapkan *tools* ini, sistem operasinya sudah menyiadakannya sendiri.

Untuk melihat versi PowerShell Anda, klik *icon* Windows dan ketikan ‘powershell’, jika ada buka aplikasi tersebut dan masukan ‘\$PSVersionTable.PSVersion’ di dalam PowerShell tersebut, kemudian akan muncul berupa informasi versi berapa yang sedang dipakai, jika tidak ada maka PowerShell anda tidak berjalan dengan baik (*error corrupted application*) sehingga tentunya perlu download dan install versi terbaru.



```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\SAN> $PSVersionTable.PSVersion

Major Minor Build Revision
-----
5      1      14409  1005

PS C:\Users\SAN>
```

- b. Git untuk Windows: Anda cukup mendownload Github Desktop untuk memenuhi kebutuhan prasyarat *tools* ini. Untuk menguji apakah *tools* tersebut sudah terpasang dengan baik, bisa masukan *command* ‘git’ dalam Command Prompt atau PowerShell.

2.2. Unduh SDK Flutter

Unduhlah SDK terbaru versi beta yang disediakan Flutter di link <https://flutter.io/setup-windows/> di bagian ‘*Get the Flutter SDK*’. Kemudian silahkan *extract* di folder yang anda kehendaki yang tidak memiliki hak istimewa seperti C:\Program Files\ . Usai *extract*, silahkan jalankan flutter_console.bat. Dengan ini Anda sekarang sudah bisa membuka *console* Flutter. Harap update Flutter Anda dengan input ‘flutter upgrade’ di dalam *console* tersebut.

2.3. Update path dan ANDROID_HOME

2.3.1. Update path

Anda bisa melewati step *update path* jika sekiranya anda tidak perlu menjalankan Flutter dengan *console* bawaan Windows. Untuk menjalankan perintah Flutter pada *console* Windows, silahkan:

1. *update PATH Environment variable* di Control Panel > User Accounts > Change my environment variables.
2. Dibawah kalimat ‘User variables’ pilih bagian PATH kemudian klik Edit...
3. Sisipkan *directory* Flutter Anda kedalam *textbox* tersebut setelah ‘;’. Bila Anda menaruh Flutter di C:\src\flutter\bin, maka *copy directory* tersebut dan *paste* ke dalam *textbox* tersebut. Tekan OK jika sudah dan OK lagi dibagian bawah bersebelahan dengan Cancel.

2.3.2. ANDROID_HOME

Lakukan *update* pada ANDROID_HOME, hanya jika Anda pernah menginstall Android SDK *directory* lain dan saat *input command* ‘flutter doctor’ dalam command prompt, tertera informasi bahwasanya ‘.. Android SDK has been installed to a custom location, set \$ANDROID_HOME to that location.’, Secara default, Android SDK berada di C:\Users\Your.PC.Name\AppData\Local\Android\sdk1.

Untuk melakukan *update*:

1. Masuk ke *directory* Anda menyimpan Android SDK tersebut, dilanjut dengan *copy directory* tersebut.
2. Buka Control Panel > User Accounts > Change my environment variables.

3. Jika tidak ada ANDROID_HOME dibawah 'User variables for Your.PC.Name' maka tekan 'New...', kemudian di bagian 'variable name' beri nama ANDROID_HOME dan isilah *value* tersebut dengan *paste* hasil copy tadi untuk menyisipkan *directory*. Selesai dengan tekan OK, dan OK yang di bagian bawah. Jika ada ANDROID_HOME, cukup tekan 'Edit...' kemudian ganti *value* tersebut dengan *directory* hasil *copy* tadi dengan *paste*.

2.4. Jalankan 'flutter doctor' di *console*

Fungi 'flutter doctor' bertujuan untuk memeriksa apakah ada dependensi atau komponen lainnya yang belum terpasang di perangkat kita. Dia akan memberikan laporan hasil pemeriksaanya dengan menampilkan informasinya di console yang kita gunakan. Untuk komponen Dart SDK tidak perlu kita siapkan sendiri karena sudah ada sepaket dengan Flutter itu sendiri. Jika ada *error*, bisa diikuti petunjuk penyelesaiannya dibawahnya.

2.5. Mempersiapkan Android Studio

2.5.1. Instalasi

Silahkan untuk unduh dan *install* Android Studio. Saat selesai proses instalasi, lanjut dengan membuka 'Android Studio Setup Wizard', kemudian akan download dan install Android SDK terbaru, Platform-Tools dan Build-Tools.

2.5.2. Mempersiapkan Perangkat Android

Anda bisa melewati tahap ini jika Anda akan menguji aplikasi yang anda buat dengan menggunakan *emulator* dari Android Studio.

Dengan *configuration* yang akan dilakukan sekarang, Anda akan dapat menggunakan *handphone* Android Anda untuk menguji aplikasinya secara langsung. Berikut tahapannya:

1. Buka 'Settings' di *handphone* Android anda.
2. Khusus untuk Android 8.0, Anda perlu menekan lagi bagian 'System'.
3. *Scroll* ke bagian bawah hingga menemukan 'About phone', tekan menu tersebut.
4. *Scroll* ke bagian bawah hingga menemukan 'Build Number', tekan sebanyak 7 kali.
5. Akan muncul *dialog-box* ke layar anda, menginformasikan bahwa *handphone* anda sudah bisa untuk *development* karena telah berubah menjadi *developer mode*.

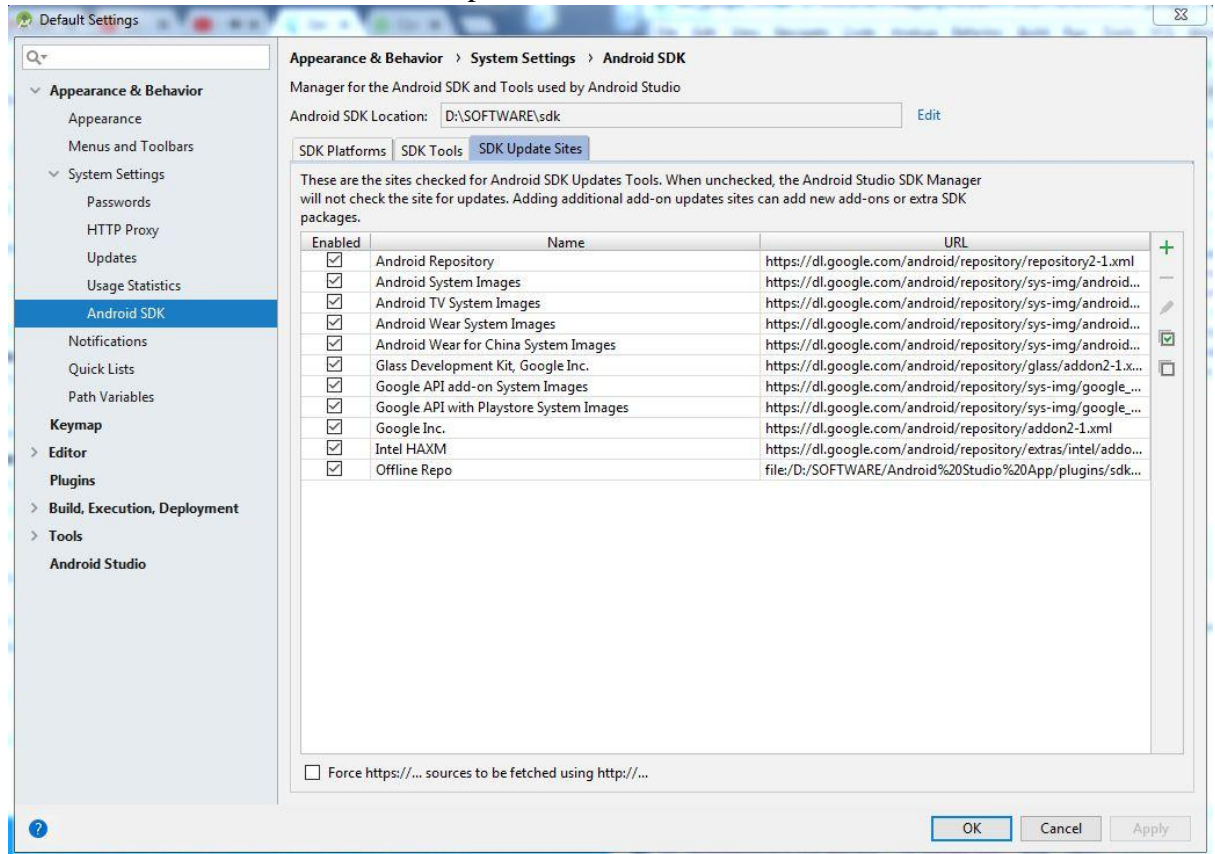
6. Kembali ke layar sebelumnya untuk masuk ke menu ‘Developer options’.
7. Kemudian silahkan *enable Developer options* dan *USB debugging* Anda.
8. Sekarang *handphone* Anda sudah bisa digunakan untuk pengujian aplikasi yang sedang Anda bangun.

2.5.3. Mempersiapkan *Emulator* Android.

Anda bisa melewati tahap ini jika Anda akan menguji aplikasi yang anda buat dengan menggunakan perangkat Android yang Anda miliki.

Emulator Android sendiri adalah replika dari *operating system* Android untuk yang bisa digunakan di *desktop*, biasanya digunakan untuk testing aplikasi. Namun penggunaan *emulator* ini cukup memakan banyak RAM, walaupun sebenarnya bisa diatur ingin seberapa besar RAM yang akan dipakai untuk *emulator* tersebut. Jadi harap hati – hati jika Anda sedang membuka banyak aplikasi dan browser dengan jumlah *tab* yang sangat banyak, tentunya ini akan sangat menghambat proses penggunaan *emulator*. Disarankan untuk mengeluarkan beberapa aplikasi yang sedang tidak dipakai jika spesifikasi perangkat desktop Anda minimal. Berikut persiapan yang perlu dilakukan untuk menggunakan *emulator* tersebut :

1. Aktifkan VM Accelerator dengan membuka Android Studio > Tools > SDK Manager, buka bagian Android SDK pada sebelah kiri, kemudian masuk ke *tab* SDK Update Sites



2. Centang kotak pada Intel HAXM, tekan OK, Android Studio akan mengunduh *file* yang dibutuhkan dan menginstallnya
3. Buka AVD Manager di Android Studio > Tools > AVD Manager dan pilih Create Virtual Device.
4. Pilih ukuran layar *emulator*, kemudian Next. Jika Anda ingin memiliki ukuran *custom* pilih New Hardware Profile. Beri nama dan sesuaikan *hardware emulator* tersebut. Pilih Finish jika sudah selesai.
5. Saat menentukan *system image*, pilihlah *system image* di *tab* Recommended, pilihlah yang *support* dengan Google APIs. Pilih Next.
6. Silahkan beri nama di bagian AVD Name.
7. Ganti tipe Graphics di bagian Emulated Performance, ganti menjadi Hardware – GLES 2.0. Finish.
8. Untuk menguji *emulator* yang telah dikonfigurasi, silahkan tekan ► untuk menjalankannya.

2.6. Instalasi Gradle

Lewati tahap ini jika sudah ada Gradle di perangkat Anda.

Sebagian besar, terutama perangkat yang baru dalam *environment* Android Studio, Gradle belum tersedia dalam perangkatnya. Persiapkan Gradle dalam perangkat Anda dengan mengikut langkah – langkah berikut :

1. Download Gradle versi terbaru di <https://gradle.org/releases/>
2. Buat *directory* (contoh: C:\Gradle).
3. Pindahkan Gradle hasil download ke *directory* yang sebelumnya telah dibuat, *extract* isi Gradle tersebut di *directory* itu.
4. Tambahkan *path* Gradle.
Klik kanan MyComputer > Properties > Advanced System Settings > Environmental Variables.
5. Di bagian System Variables pilih PATH, kemudian Edit..., sisipkan *directory* tempat Anda menyimpan Gradle setelah ‘;’ (contoh : ... , C:\Gradle\gradle-4.9\bin) .
6. Tekan OK dan OK lagi untuk menyimpan pengaturan tadi.
7. Tekan OK untuk keluar dari System Properties.

3. Persiapan *Editor* Android Studio

Karena Anda menggunakan Android Studio sebagai *editor*, maka perlu install *plugin* Flutter dan Dart. Berikut yang perlu dilakukan:

1. Buka File >Settings > Plugins (sebelah kiri layar)
2. Ceklis *plugin* Dart dan Flutter
3. Tekan OK, kemudian akan muncul menu instalasi, tunggu hingga selesai,
4. Tekan Restart ketika dimintai.

4. Mencoba Menjalankan *Project* Flutter

Setup untuk penggunaan Framework Flutter sudah selesai, Anda sudah bisa memulai membuat project Flutter dan coba menjalannkanya. *Tutorial* selanjutnya bisa dibuka di <https://flutter.io/get-started/test-drive/#androidstudio>