

Hot reload

Fitur **hot reload** pada Flutter membantu Anda untuk melakukan percobaan, membangun UI, menambahkan fitur, dan memperbaiki bug dengan cepat dan mudah. Hot reload berfungsi dengan menyuntikkan berkas kode yang diperbarui ke dalam **Dart Virtual Machine** (VM) yang sedang berjalan. Setelah VM memperbarui class dengan versi field dan fungsi yang baru, *framework* Flutter secara otomatis membangun kembali *widget*. Hal ini memungkinkan Anda dengan cepat melihat efek dari perubahan Anda.

Lebih lanjut, berikut ini cara melakukan hot reload pada aplikasi Flutter:

1. Jalankan aplikasi Flutter. Hanya aplikasi Flutter dalam **mode debug** yang dapat melakukan **hot reload**.
2. Ubah beberapa kode kecil dalam salah satu file Dart di *project* Anda. Jika perubahan yang terjadi banyak atau perubahan berada di dalam *initState* atau *main*, Anda harus melakukan *hot restart*.
3. Jika Anda menggunakan IDE/editor yang didukung Flutter (Android Studio/IntelliJ/Visual Studio Code), pilih Save/Save All (cmd+s/ctrl+s) atau klik tombol **Hot Reload** pada toolbar di setiap IDE.
4. Jika Anda menjalankan aplikasi di *command line* menggunakan **flutter run**, masukkan perintah **r** di jendela terminal.

Setelah operasi hot reload berhasil, Anda akan melihat pesan di konsol seperti ini:

```
Performing hot reload...
```

```
Reloaded 1 of 448 libraries in 978ms.
```

Hot reload menyebabkan semua *widget* yang ada dibangun kembali. Hanya kode yang terlibat dalam *rebuilding widget* yang dieksekusi ulang secara otomatis. Seperti contoh fungsi *main* dan *initState* tidak akan dijalankan kembali jika menggunakan **hot reload**.

Apa bedanya *hot reload*, *hot restart*, dan *full restart*?

- **Hot reload** memuat perubahan kode ke dalam VM dan menjalankan kembali metode build yang ada di dalam widget. Ini akan membangun kembali widget-widget yang ada, dan mempertahankan status terakhir aplikasi.
- **Hot restart** memuat perubahan kode ke dalam VM dan mengulang kembali aplikasi dari awal dan akan kehilangan state aplikasi (kembali ke nilai awal).

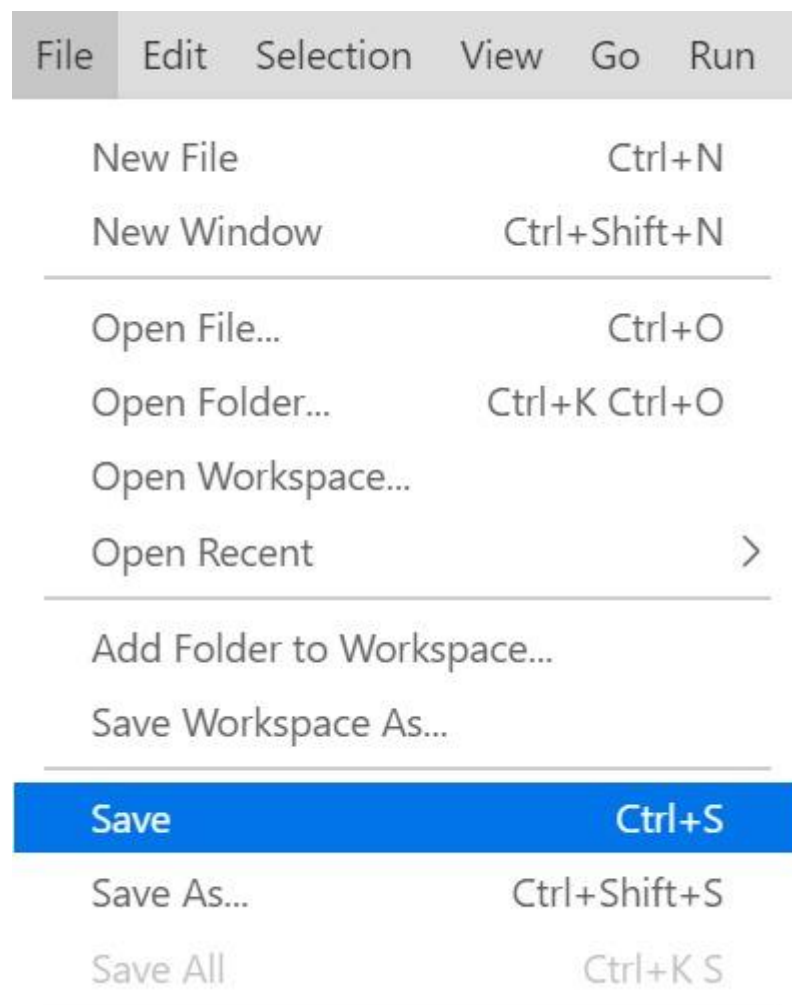
- **Full restart** mengkompilasi kode dari awal, tidak ada pintasan keyboard untuk ini, Anda harus menghentikan dan menjalankan kembali project (klik stop kemudian run kembali).

Flutter web saat ini hanya mendukung **hot restart**.

VISUAL STUDIO CODE

Flutter menawarkan siklus pengembangan cepat dengan *Hot Reload*. Fitur tersebut memiliki kemampuan memuat ulang kode aplikasi yang sedang berjalan tanpa memulai ulang atau kehilangan status aplikasi. Caranya, Buat perubahan ke sumber aplikasi, beritahu IDE atau alat baris perintah Anda bahwa Anda ingin melakukan Hot Reload, dan lihat perubahan di simulator, emulator, atau perangkat Anda.

1. Buka `lib/main.dart`.
2. Ubah string '*You have **pushed** the button this many times*' menjadi '*You have **clicked** the button this many times*'.
Penting: Jangan hentikan aplikasi Anda. Biarkan aplikasi Anda berjalan.
3. Simpan perubahan baris perintah dengan cara: **Pilih File > Save All**, atau klik Hot Reload (tombol petir kuning).



Anda akan langsung melihat string yang sudah diperbarui di aplikasi yang sedang berjalan.

Kasus spesial

Bagian berikutnya menjelaskan situasi umum di mana kode yang dimodifikasi tidak akan berjalan lagi setelah hot reload. Dalam beberapa kasus, perubahan kecil pada kode Dart akan memungkinkan Anda untuk terus menggunakan hot reload untuk aplikasi Anda. Dalam beberapa kasus, kita perlu melakukan hot restart atau full restart:

- **Aplikasi mati atau keluar**

Hot reload akan berhenti jika aplikasi terhenti atau keluar, misalnya jika aplikasi terlalu lama berada di *background*.

- **Compilation errors**

Ketika perubahan kode mellihatkan kesalahan kompilasi, hot reload selalu menghasilkan pesan kesalahan yang seperti di bawah ini:

Hot reload was

rejected: '/Users/obiwan/Library/Developer/CoreSimulator/Devices/AC94F0FF-16F7-46C8-B4BF-218B73C547AC/data/Containers/Data/Application/4F72B076-42AD-44A4-A7CF-57D9F93E895E/tmp/ios_testWIDYdS/ios_test/lib/main.dart': warning: line 16 pos 38: unbalanced '{' opens here

Widget build(BuildContext context) {

^

'/Users/obiwan/Library/Developer/CoreSimulator/Devices/AC94F0FF-16F7-46C8-B4BF-218B73C547AC/data/Containers/Data/Application/4F72B076-42AD-44A4-A7CF-57D9F93E895E/tmp/ios_testWIDYdS/ios_test/lib/main.dart': error: line 33 pos 5: unbalanced ')'

};

^

Dalam situasi ini cukup perbaiki kesalahan pada baris kode Dart yang ditentukan untuk tetap menggunakan hot reload.

- **CupertinoTabView's builder**

Hot reload tidak akan berjalan ketika ada perubahan pada builder di CupertinoTabView. Untuk lebih lanjutnya silahkan baca <https://github.com/flutter/flutter/issues/43574>.

- **Tipe enum**

Hot reload tidak akan bekerja ketika tipe enum diubah menjadi kelas reguler atau sebaliknya.

- - *// Contoh sebelum:*
 - `enum Level { beginner, intermediate, advanced }`
 -
 -
 - *// Contoh sesudah:*
 - `class Level {`
 - `Level(this.id);`
 - `int id;`
 - `}`

- **Perubahan font**

Hot reload mendukung sebagian besar perubahan aset. Namun, jika Anda mengubah font, Anda harus menggunakan hot restart.

- **Generic types**

- Hot reload tidak akan bekerja ketika pendeklarasian tipe generik dimodifikasi.
- **// Contoh sebelum:**
- `class A<X> {`
- `X ax;`
- `}`
-
-
- **// Contoh sesudah:**
- `class A<X, Y> {`
- `X ax;`
- `Y ay;`
- `}`

- **Native code**

Jika Anda mengubah kode native tiap platform (seperti Java, Kotlin, Swift, atau Objective-C), Anda harus melakukan full restart (stop dan run kembali project) untuk melihat perubahan.

- **State sebelumnya dikombinasikan dengan kode baru**

Fitur hot reload pada Flutter kadang-kadang digambarkan sebagai stateful hot reload yang mempertahankan status aplikasi Anda. Ini memungkinkan Anda untuk melihat efek perubahan terbaru saja tanpa membuang kondisi saat ini. Misalnya, jika aplikasi Anda mengharuskan pengguna untuk login, Anda dapat memodifikasi dan melakukan hot reload beberapa halaman di bawah hierarki navigasi, tanpa memasukkan kembali kredensial login Anda. Saat disimpan, maka hasil output akan sesuai dengan yang kita inginkan atau kita ubah.

Jika perubahan kode memengaruhi kondisi aplikasi Anda, seperti data atau fungsi yang seharusnya berjalan dengan baik tiba-tiba berhenti berfungsi. Maka disarankan untuk menggunakan Hot Restart.

Seperti Kode di bawah ini :

```
class MyWidget extends StatelessWidget {
  Widget build(BuildContext context) {
    return GestureDetector(onTap: () => print('T'));
  }
}
```

Diubah menjadi:

```
class MyWidget extends StatefulWidget {
  @override
  State createState() => MyWidgetState();
}

class MyWidgetState extends State { /*...*/ }
```

Kemudian tekan tombol hot reload, maka console akan menampilkan output seperti berikut:

MyWidget is not a subtype of StatelessWidget

Dalam situasi ini, full restart diperlukan untuk melihat aplikasi yang diperbarui.

Selamat Anda telah menyelesaikan modul pertama ini. Semoga tetap semangat ya. Apalagi kini Anda telah membuat project dan menjalankan aplikasi Flutter pertama Anda. Wow! Pada modul selanjutnya Anda telah siap untuk membuat aplikasi Anda sendiri. Sudah siap? Jaga semangat Anda dan sampai jumpa di modul berikutnya ya!