# Menggunakan Packages

Package Dependencies

Dalam pengembangan suatu aplikasi, kita tidak akan lepas dari *package/library* (selanjutnya akan disebut *package*). *Package* di sini merupakan sebuah kode yang dibuat untuk menyelesaikan suatu masalah. Contohnya ketika aplikasi yang kita buat membutuhkan fitur kalender sementara fitur tersebut tidak di-support oleh Flutter. Alih-alih membuat fitur kalender dari nol, kita dapat menggunakan *package* yang telah dibuat oleh developer lain. Waktu pembuatan fitur menjadi lebih singkat!

*Package dependencies* merupakan sekumpulan *package* yang dibutuhkan dalam pengembangan aplikasi. *Package* tersebut akan diatur oleh *package manager*. Setiap bahasa pemrograman memiliki *package manager*-nya masing-masing, contohnya NodeJS memiliki npm atau yarn, Java dengan maven atau gradle, PHP dengan composer. Begitu pula dengan Flutter yang ditulis dengan bahasa dart memiliki *package manager* bernama **pub**.

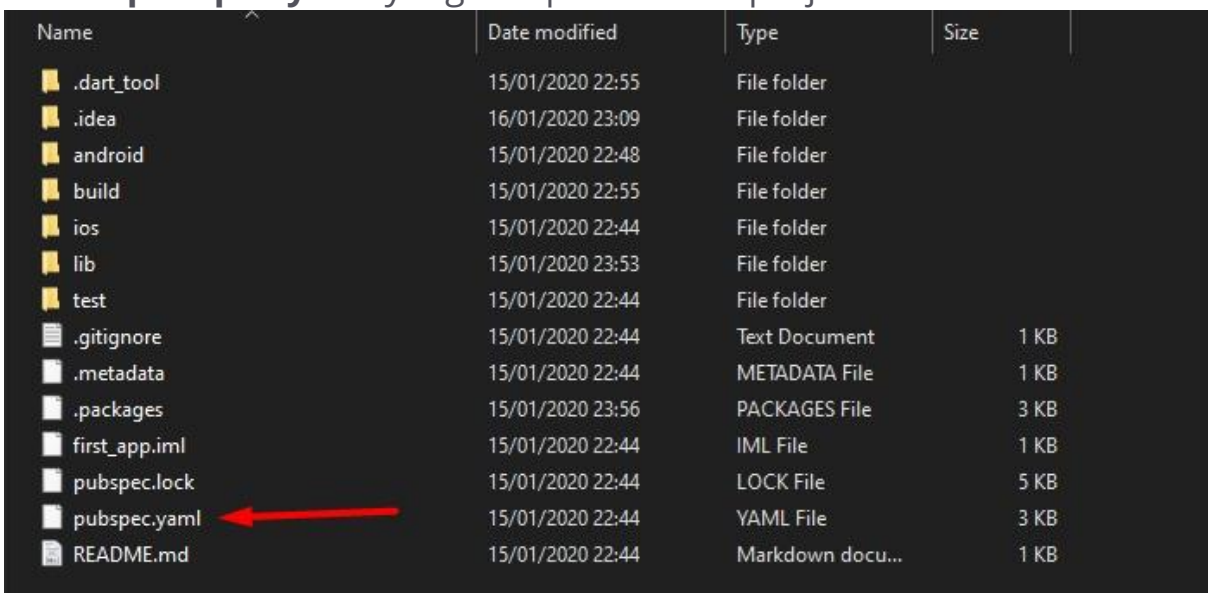
Kali ini kita akan membahas mengenai *package manager pub*, bagaimana menggunakan pub, dan di mana mencari package yang dapat digunakan untuk aplikasi kita.

Dart Pub

Seperti yang telah kita singgung sebelumnya Pub merupakan sebuah *package manager*. Pub memiliki tugas untuk mengatur *package* apa saja yang dibutuhkan dalam pengembangan aplikasi. Pada *package manager* kita dapat mengatur versi *package* yang ingin kita gunakan. Pengaturan versi sangat penting karena ketika versi flutter/dart yang digunakan tidak cocok dengan package yang kita butuhkan akan berpengaruh pada jalannya aplikasi yang kita buat. Oleh karena itu, kita harus memastikan versi yang kompatibel dengan versi Flutter yang terinstal.

Lalu bagaimana kita menggunakan pub pada project Flutter kita? Untuk mengatur *package-package* yang akan kita gunakan, cukup buka

berkas **pubspec.yaml** yang ada pada folder project.




Ketika membuka berkas **pubspec.yaml** kita akan melihat begitu banyak pengaturan tapi tidak perlu khawatir karena yang kita bahas hanya mengenai *package dependencies*-nya saja.

Coba kita fokus pada kode yang ada pada **pubspec.yaml** berikut:

```
1. dependencies:
2.   flutter:
3.     sdk: flutter
4.
5.   # The following adds the Cupertino Icons font to your application.
6.   # Use with the CupertinoIcons class for iOS style icons.
7.   cupertino_icons: ^0.1.2
8.
9. dev_dependencies:
10.    flutter_test:
11.      sdk: flutter
```

Kode di atas merupakan *package-package* yang digunakan pada *project* Flutter kita. Jika kita perhatikan, terdapat 2 jenis *dependency* yaitu dependencies dan dev_dependencies. Fungsi dev_dependencies digunakan untuk package-package yang berkaitan ketika proses pengembangan aplikasi Flutter, contohnya seperti flutter_test yang digunakan untuk *testing*. Package di dalam dev_dependencies tidak akan disertakan ketika aplikasi dirilis pada *play store* atau *app store*. Fungsi *dependencies* digunakan untuk *package-package* yang langsung berkaitan dengan fitur aplikasi Flutter, contohnya seperti cupertino_icons yang digunakan untuk mendapatkan

ikon-ikon cupertino (icon untuk iOS) dan contoh lainnya seperti cloud_firestore yang merupakan *package* untuk firebase firestore.

Sekarang kita akan fokus pada *dependencies*. Untuk mendaftarkan *package* yang dibutuhkan kita cukup menulis seperti di bawah ini pada bagian *dependencies*:

```
1. nama_package: versi
```

nama_package merupakan nama package yang kita butuhkan, lalu disambung dengan versinya. Penulisan versi bisa langsung seperti contoh 0.1.2, atau kita menambahkan simbol caret (^) seperti ^0.1.2 . Simbol caret (^) artinya: gunakan versi *patch* terbaru dari versi yang telah ditentukan. Jika versi nya ^0.1.2 artinya kita akan gunakan versi minimal 0.1.2 dan maksimal versi terbaru. Karena itu, jika versi *package* tersebut sekarang sudah *update*, maka *package* yang digunakan merupakan versi terbaru.

**Catatan**: Hanya pada versi patch atau pada angka terakhir yaitu angka 2 jika pada contoh cupertino_icons: ^0.1.2. Atau kita juga bisa gunakan versi minimal dan maksimal seperti contoh '>=0.1.2 <2.0.0' yang artinya kita akan menggunakan versi terbaru yang ada pada saat ini yang masih berada di dalam range tersebut yaitu lebih besar sama dengan 0.1.2 dan lebih kecil dari 2.0.0.

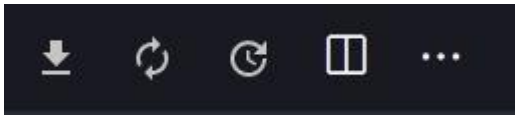
Okay sebagai contoh kita akan menambahkan sebuah package provider yang nantinya akan kita gunakan.

```
1. dependencies:
2.   flutter:
3.     sdk: flutter
4.
5.   cupertino_icons: ^0.1.2
6.   provider: ^4.0.1
```

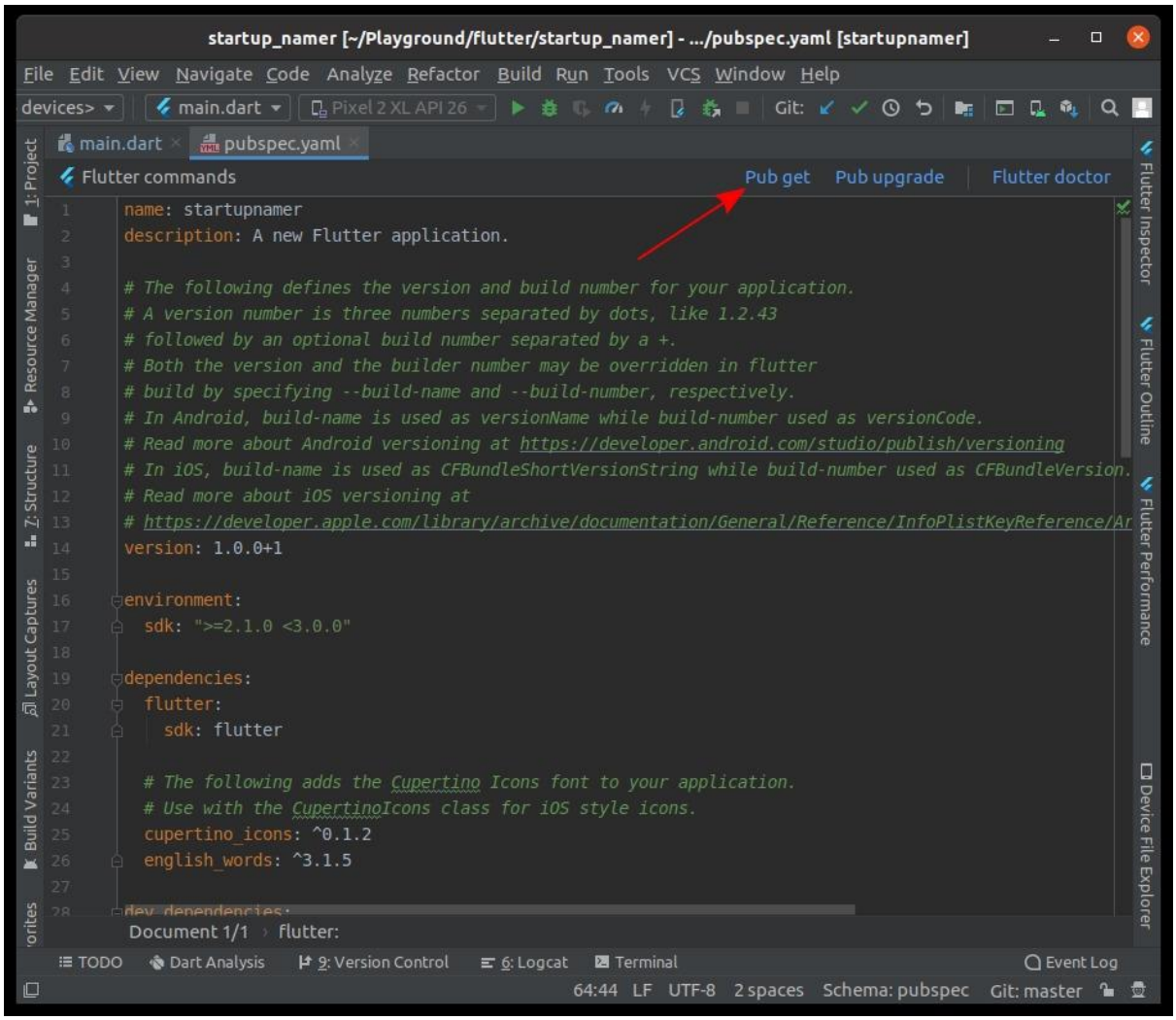
Yang perlu diperhatikan dalam menulis berkas **.yaml** adalah pada indentasinya. Indentasi atau penggunaan spasi ini sangat penting karena menunjukkan urutan dan blok kode yaml yang dibaca oleh komputer. Sebagai contoh, ketika kita menambahkan *package provider*, maka kita harus menuliskannya sejajar dengan *package* lainnya dan juga lebih menjorok ke dalam jika dibandingkan dengan *dependencies*.

Ini menunjukkan bahwa *package* seperti cupertino_icons dan provider merupakan bagian dari dependencies yang akan ditambahkan. Setiap jaraknya adalah 2 spasi, jika *dependencies* menempel pada ujung kiri, maka cupertino_icons dan provider berjarak 2 spasi dari ujung kiri.

Setelah menambahkan *package* yang dibutuhkan, kita dapat melakukan **get package** tersebut. Jika Anda menggunakan visual studio code cukup simpan berkas **pubspec.yaml**, maka nanti akan secara otomatis mensinkronisasi pubspec tersebut. Atau, bisa dengan menekan tombol seperti gambar unduh pada pojok kanan atas paling kiri.



Bila menggunakan Android Studio Anda cukup menekan tombol "**Pub get**" pada Android Studio seperti berikut:



Kita juga bisa secara manual menggunakan terminal dengan menjalankan perintah flutter pub get di dalam *project* tersebut. Setelah melakukan pub get, maka package tersebut sudah dapat digunakan.

### Tempat-tempat mencari package flutter

Flutter memiliki segudang package yang telah dibuat oleh komunitas, lantas di mana kita dapat mencari package-package untuk kita gunakan? Berikut website-website yang dapat Anda gunakan untuk mencari package.

- https://pub.dev
  Website ini merupakan *web official* untuk Anda mencari *package*.

- https://flutterawesome.com
  Berisi *package-package* yang dibuat oleh komunitas, di sini banyak sekali *package* UI keren yang dapat Anda coba.

### Private Packages

Selain menggunakan package yang ada pada pub.dev Anda juga bisa menggunakan package yang tidak dipublikasikan pada pub.dev tersebut dengan cara menggunakan url git package tersebut:

```
1. dependencies:
2.   plugin1:
3.     git:
4.       url: git://github.com/flutter/plugin1.git
```

Atau path direktori package tersebut yang tersimpan secara *offline* di komputer Anda.

```
1. dependencies:
2.   plugin1:
3.     path: ../plugin1/
```