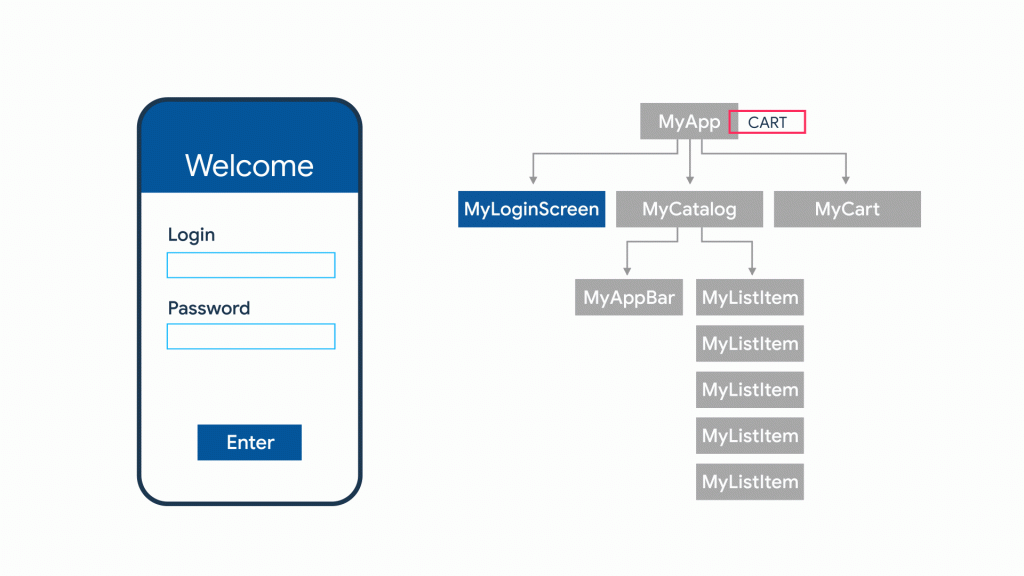
Apa itu State Management?

**State management** adalah sebuah cara untuk mengatur data / state kita bekerja, bisa juga untuk memisahkan antara logic dan view, dimana logic tersebut juga bisa *reusable*.



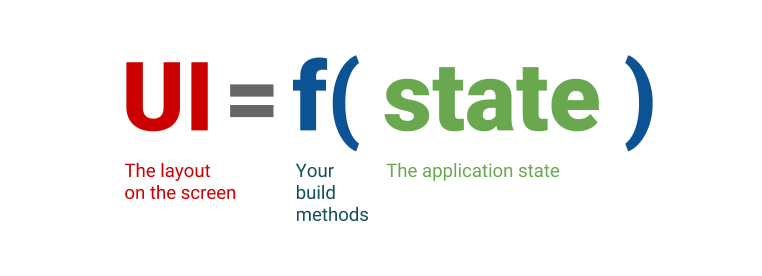
Cara kerja state management seperti *provide and listen*, artinya adalah kamu bisa memasukan state yang kemungkinan bisa berubah sewaktu waktu, lalu widget yang subscribe (listen) dengan provider yang kita buat akan berubah sesuai dengan state yang berubah.

Setiap bagian dalam flutter terbuat dalam bentuk widget. Terdapat dua widget yang memiliki perlakukan state yang berbeda, yaitu stateful dan stateless widget.

Apa perbedaan keduanya?

* **Stateless** merupakan widget yang dimuat secara statis dimana seluruh konfigurasi yang dimuat didalamnya telah diinisialisasikan sejak awal widget itu dimuat. Artinya, state jenis ini tidak dapat diubah dan tidak akan pernah berubah. Stateless biasa digunakan hanya untuk tampilan saja seperti *button*, item, box container, dan lain-lain.
* **Stateful** merupakan suatu widget yang sifatnya dinamis atau dapat berubah-ubah, kebalikan dari stateless widget. Pada stateful, kamu dapat menggunakan property initState yang berfungsi untuk menginisialisasi state yang akan pertama kali dijalankan. Stateful widget dapat mengubah atau mengupdate tampilan, menambah widget lainnya, mengubah nilai variabel, icon, warna, dan masih banyak lagi.

Bersifat Deklaratif



Flutter bersifat deklaratif, artinya flutter membangun user interfacenya dengan merefleksikan setiap perubahan state. State ini lah yang menjadi trigger untuk me-redraw tampilan *user interface*. Ketika terjadi perubahan, UI akan merebuild secara otomatis dari awal.

Sumber : https://caraguna.com/pengenalan-state-management-flutter/

Jenis-jenis State Management Flutter

State manajement memang merupakan salah satu topik pembahasan yang sangat kompleks apabila kita ingin memperdalam tentang flutter. Namun, untuk menggunakan state management, kamu dapat memanfaat kan beberapa package state management berikut.

[1. Provider](https://pub.dev/packages/provider)

Provider merupakan state management yang paling sederhana dan mudah digunakan. Provider menyediakan sebuah teknik mengolah state yang dapat digunakan untuk memanage data didalam aplikasi. Manfaat menggunakan provider antara lain :

* Mengalokasikan resource menjadi lebih sederhana
* Lazy-loading
* Mengurangi *boilerplate* saat membuat kelas baru setiap saat
* Support dengan devtool, karena status aplikasi kamu akan terlihat di flutter devtool
* Peningkatan skalabilitas untuk class yang memanfaatkan mekanisme *listen* yang dibangun secara kompleks.

[2. Riverpod](https://pub.dev/packages/riverpod)

Riverpod mirip dengan provider yang *compile-safe* untuk digunakan dan teruji *(testable)*. Kamu tidak akan menemukan ProviderNotFoundException atau error dalam menangani proses dalam mengelola state. Karena, selagi kode kamu berhasil di kompilasi, maka aplikasi akan tetap dijalankan.

Riverpod mendukung *multiple providers* dengan tipe yang sama, proses asynchronous, dan mampu menambahkan provider dari file mana saja.

[3. Redux](https://pub.dev/packages/flutter_redux)

Bagi kamu yang web developer, mungkin kamu sudah tidak asing dengan yang namanya redux. Redux adalah arsitektur aliran data searah yang memudahkan pengembangan, pemeliharaan, dan pengujian aplikasi. Redux berasal dari javascript yang membuat predictable state container untuk aplikasi.

Berikut ini adalah manfaat ketika kamu menggunakan Redux :

* Sangat scalable dan terstruktur dengan baik.
* Memiliki alur yang jelas yang membuatnya menjadi mudah mengelola bahkan mengembalikan perubahan state, atau debug aplikasi.
* Perubahan state dilakukan dengan fungsi.
* Limited API.

[4. BLoC](https://pub.dev/packages/flutter_bloc)

BLoC atau Business Logic Component adalah *design pattern* yang membantu kamu untuk memisahkan presentation dengan business logic. Sehingga komponen pada project terbagi menjadi presentational component, BLoC, dan backend. Pattern ini memperbolehkan developer untuk fokus dalam mengkonversikan event menjadi state.

BLoC mengelola state dengan menggunakan pendekatan*stream* atau *reactive.* Secra umum, data akan bergerak dari BLOC ke UI, atau sebaliknya dalam bentuk *streams.*

[5. GetX](https://pub.dev/packages/get)

GetX merupakan salah satu pilihan terbaik untuk state management yang memiliki performa tinggi, memiliki injection dependency yang cerdas, serta memiliki manajemen route yang cepat dan praktis. GetX tidak akan memberatkan aplikasi kamu, meskipun memiliki fitur yang banyak, namun masing-masing fitur berada dalam container terpisah dan akan mulai dijalankan setelah dipakai. Misal, jika kamu hanya menggunakan state management, maka hanya state management lah yang akan di compile, tidak termasuk route dan lainnya.

State management ini memiliki komunitas yang besar, banyak kolaborator aktif. GetX mampu berjalan di OS Android, iOS, Web Linux, Mac, Windows, dan diserver kamu.

Sumber : https://caraguna.com/pengenalan-state-management-flutter/