

Tugas progaming FLUTTER

Nim:535200085

Nama:blasius lawing

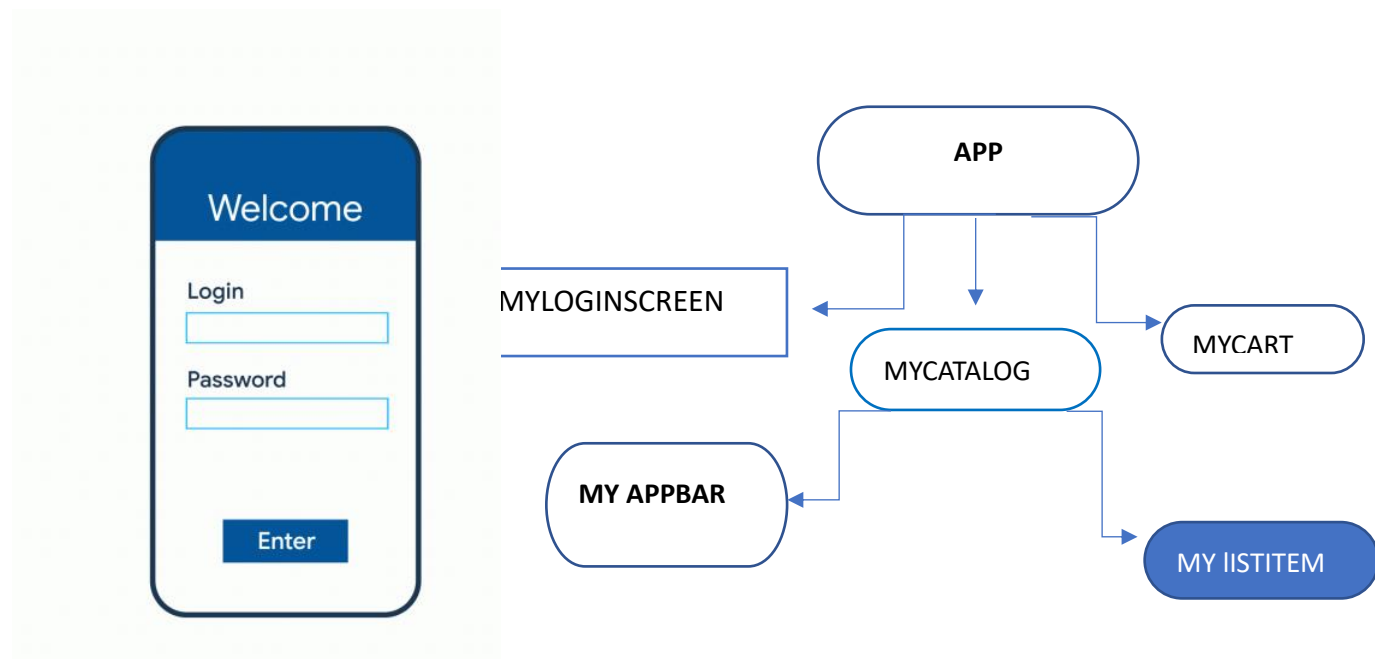
State Management – seperti yang kita ketahui Flutter adalah *Cross-Platform Mobile App SDK (Software Development Kit)* untuk membuat aplikasi Android dan iOS dalam satu codebase dengan performa tinggi. Hingga saat ini, flutter digunakan untuk mengembangkan aplikasi Web, Linux, dan MacOS.

Lalu, apa itu state? perlukah kita mengelola state ketika mengembangkan aplikasi dengan flutter? Ketika kamu menjelajahi flutter, akan ada waktu dimana kamu akan menggunakan data yang dapat diakses lebih dari satu halaman, atau kamu ingin terjadi perubahan pada *screen* (layar) ketika data berubah tanpa harus me-redraw semua komponen halaman tersebut.

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*.

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 perlakuan state yang berbeda, yaitu *stateful* dan *stateless* widget.

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.

Jenis-jenis State Management Flutter

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

1. Provider

Provider merupakan state management yang paling sederhana dan mudah digunakan. Provider menyediakan sebuah teknik mengolah state yang dapat digunakan untuk manage 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

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

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

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*. Secara umum, data akan bergerak dari BLoC ke UI, atau sebaliknya dalam bentuk *streams*.

5. GetX

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 kam

Your email address will not be published. Required fields are marked *

Comment *

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

State management memang merupakan salah satu topik pembahasan yang sangat kompleks apabila kita ingin memperdalam tentang flutter. Namun, untuk menggunakan state management, kamu dapat memanfaatkan b

- Perubahan state dilakukan dengan fungsi.
- Limited API.

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

GetX merupakan salah satu pilihan terbaik untuk state management yang memiliki performa tinggi, memiliki injection dependency yang cerdas 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.

Tags: FLUTTERMOBILEAPPSSTATEMANAGEMENT



Related Stories

3 Jenis Build Mode Flutter yang Wajib Kamu Ketahui

Debug mode

Dalam **debug mode**, aplikasi dibangun dengan informasi debugging dan pemantauan performa yang dapat ditemukan. Ini memungkinkan pengembang untuk menemukan dan memperbaiki bug dan masalah selama pengembangan aplikasi.

Debug mode juga memiliki fitur seperti pengecakan dinamis, pemantauan memori, dan tracing dinamis yang membantu dalam proses debugging. Build mode ini lebih lambat dibandingkan dengan release mode karena mengandung informasi tambahan untuk membantu proses debugging.

Profile mode

Profile mode memungkinkan pengembang untuk mengukur performa aplikasi dan menemukan bagian yang membutuhkan optimisasi. Dalam mode ini, Flutter akan memonitor dan melaporkan performa aplikasi seperti frame rate, latency, dan memori yang digunakan.

Informasi ini sangat berguna untuk memahami bagaimana aplikasi bekerja dalam kondisi produksi dan membantu pengembang untuk mengoptimalkan aplikasi kamu.

Release mode

Release mode adalah mode yang digunakan untuk membuat versi akhir dari aplikasi yang akan diterbitkan atau dipasang di app store. Dalam mode ini, kode aplikasi dikompilasi dengan lebih cepat dan dioptimalkan untuk kinerja dan ukuran file yang lebih kecil.

Informasi debugging dan pemantauan performa tidak tersedia dalam mode ini untuk menjamin kinerja optimal. Dalam release mode, semua fitur yang tidak diperlukan akan dibuang untuk membuat aplikasi lebih ringkas dan cepat.

Dengan menggunakan build mode yang sesuai pada setiap tahap pengembangan aplikasi, pengembang dapat memastikan bahwa aplikasi kamu bekerja dengan baik dan memenuhi standar yang dibutuhkan untuk diterbitkan atau dipasang di app store.