# Praktikum 4

# State Management



## TUJUAN PEMBELAJARAN

1. Mahasiswa dapat mengimplementasikan

## DASAR TEORI

Dasar teori.

## TUGAS PENDAHULUAN

1. Siapkan project Tourism Place yang telah dikerjakan pada praktikum sebelumnya.
2. Check aplikasi Tourism Place dan pastikan semua fitur yang ada pada praktikum sebelumnya telah dikerjakan
3. Jalankan aplikasi Tourism Place dan cobalah semua fitur yang ada. Pastikan semua fitur dapat berjalan dengan baik.

## PERCOBAAN

### Percobaan 1 : State Management menggunakan setState()

Langkah Percobaan:

1. Buka project tempat wisata pada praktikum sebelumnya
2. Bukalah file tourism\_place.dart dan pindahkan variable tourismPlaceList kedalam class MainScreen dan Ubahlah tipe datanya menjadi List<TourismPlace>.

Text

Description automatically generated

1. Tambahkan variable baru dengan nama doneTourismPlaceList

Text

Description automatically generated

1. Buatlah file dart baru dengan nama done\_tourism\_list.dart kemudian buat class dengan nama DoneTourismList:

Text

Description automatically generated

1. Buat file baru dengan nama list\_item.dart dan buatlah class ListItem didalamnya

Text

Description automatically generated

1. Buka main\_screen.dart dan amati fungsi listItem(). Kemudian pindahkan isi dari method listItem() pada class MainScreen yang berada dalam area biru, ke dalam method build dalam class ListItem

A picture containing text

Description automatically generated Text

Description automatically generated

1. Tambahkan beberapa variable dan parameter dalam constructor class ListItem

Text

Description automatically generated

1. Jangan lupa menghapus method listItem dalam class MainScreen.
2. Tambahkan kode program berikut pada method build() dalam class ListItem:

Text

Description automatically generated

1. Buatlah class \_MainScreenState dibawah class MainScreen. Kemudian pindahkan method build milik MainScreen ke \_MainScreenState. Tambahkan override method createState pada MainScreen. Serta pindahkan List tourismPlaceList dan doneTourismPlaceList kedalam class \_MainScreenState. Variable List doneTourismPlaceList dideklarasikan dalam class \_MainScreenState karena akan digunakan oleh seluruh child dari class \_MainScreenState.

Text

Description automatically generated

1. Buatlah file dart baru dengan nama tourism\_list.dart dan buat class TourismList dan \_TourismListState. Serta pindahkan List tourismPlaceList kedalam class \_TourismListState. List tourismPlaceList dipindah karena List tersebut hanya digunakan secara lokal di class \_TourismListState sehingga tidak perlu didklarasikan dalam parent class. Setelah dua class tersebut dibuat kemudian pindahkan(cut & paste) ListView.builder yang ada pada class \_MainScreenState kedalam \_TourismListState. Berikut ini adalah isi dari class TourismList dan \_TourismListState.

Text

Description automatically generated

Text

Description automatically generated

1. Kemudian ubahlah pemanggilan ListItem dalam method build pada class \_TourismListState

Text

Description automatically generated

1. Ubahlah warna Card pada class ListItem seperti berikut:

Graphical user interface, text

Description automatically generated

1. Ubahlah MainScreen dan \_MainScreenState seperti berikut:

Text

Description automatically generated

1. Hasil running program akan menjadi seperti berikut:

Graphical user interface, text

Description automatically generated Graphical user interface, text

Description automatically generatedShape, square

Description automatically generated

Graphical user interface, application

Description automatically generatedShape, square

Description automatically generated Graphical user interface, application

Description automatically generatedA picture containing shape

Description automatically generated

### Percobaan 2: State Management Menggunakan Provider

Langkah Percobaan:

1. Tambahkan dependencies Provider dalam file pubspec.yaml

Text

Description automatically generated

1. Kemudian jalankan perintah flutter pub get pada terminal

Graphical user interface, text

Description automatically generated

1. Setelah package berhasil ditambahkan kemudian buat folder baru bernama provider dalam folder lib. Kemudian buat file baru bernama done\_tourism\_provider.dart dan class bernama DoneTourismProvider didalamnya.

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated with medium confidence

1. Pada percobaan sebelumnya state yang berupa

List<TourismPlace> doneTourismPlaceList;

dideklarasikan di dalam class \_MainScreenState. Pada percobaan kali ini state disimpan dalam class DoneTourismProvider.

Text

Description automatically generated

Pada method complete() di atas setelah kita menambahkan logic untuk penambahan data atau pengurangan data pada \_doneTourismPlaceList. Jika isDone bernilai true maka data place akan ditambahkan pada \_doneTourismPlaceList. Jika isDone bernilai false maka data place akan dihapus dari \_doneTourismPlaceList.

Setelah logic tersebut kemudian dipanggil fungsi notifyListeners(). Fungsi ini akan memberitahukan kepada widget yang menggunakan state ini bahwa terjadi perubahan sehingga harus membangun ulang UI.

1. Selanjutnya tambahkan Widget ChangeNotifierProvider di atas widget yang membutuhkan data doneTourismPlaceList. Pada project kita data doneTourismPlaceList dibutuhkan oleh widget DoneTourismList dan TourismList. Maka kita widget yang harus di wrap dengan ChangeNotifierProvider adalah widget diatas DoneTourismList dan TourismList yaitu MainScreen.

Text

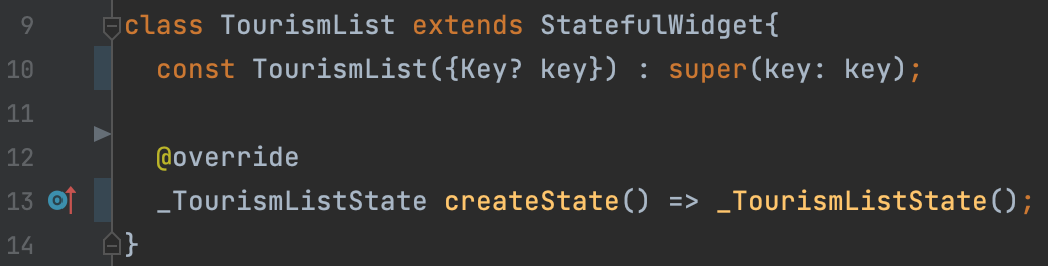
Description automatically generated

1. Karena kita sudah membuat Provider untuk doneTourismPlaceList maka variable doneTourismPlaceList di dalam MainScreen bisa kita hapus. MainScreen akan menjadi seperti berikut:

Text

Description automatically generated

1. Karena kita sudah menambahkan ChangeNotifierProvider pada MainScreen maka semua child dari MainScreen termasuk child yang berada di dalam child juga dapat menggunakan Consumer, yaitu widget untuk menggunakan provider yang telah dibuat. Berikut adalah isi dari class TourismList dan \_TourismListState yang menggunakan Consumer.



Text

Description automatically generated

Perbedaan mencolok antara penggunaan setState dengan provider adalah pada setState seluruh widget di dalam method build() akan dibangun ulang. Sementara, pada provider hanya widget yang dibungkus widget Consumer. Oleh karena itu, penting untuk meletakkan Consumer pada lingkup sekecil mungkin dan hanya pada widget yang dibutuhkan.

1. Terkadang kita perlu mengakses data dari state namun tidak perlu mengubah UI. Untuk kasus ini gunakan Provider.of untuk mendapatkan data dari state. Pada project kita class DoneTourismList adalah class yang dapat menggunakan Provider.of didalamnya. Berikut adalah isi dari class DoneTourismList:

Text

Description automatically generated

1. Aplikasi siap dijalankan.

## TUGAS

1. Ubahlah tampilan DoneTourismList menjadi seperti berikut:

Graphical user interface, application

Description automatically generated

## LAPORAN RESMI

Kumpulkan hasil Percobaan, Latihan dan Tugas di atas dan tambahkan analisa untuk tiap percobaan, latihan, dan tugas yang telah dibuat.