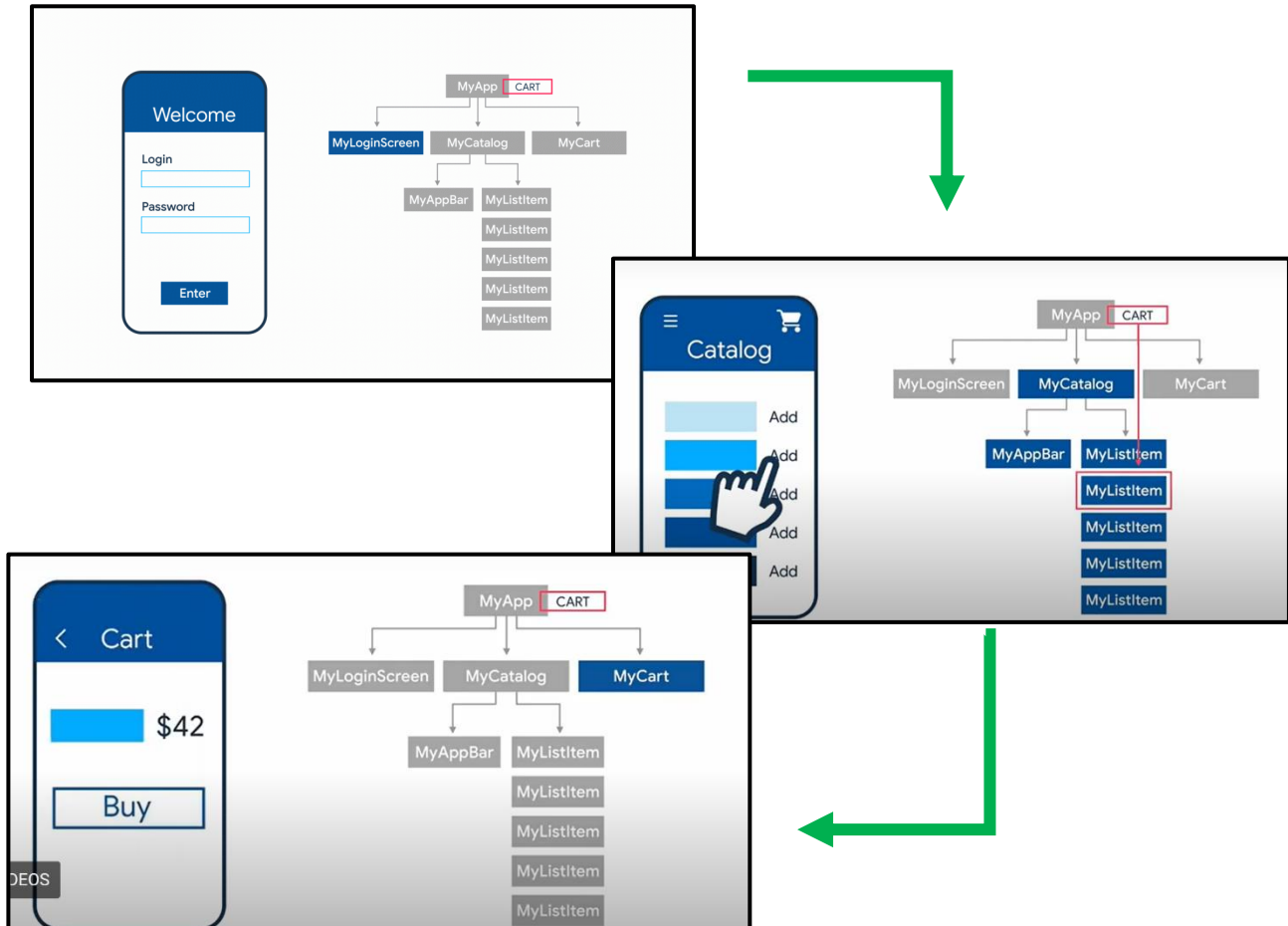


## State management

**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 memasukkan state yang kemungkinan bisa berubah sewaktu waktu, lalu widget yang subscribe (listen) dengan provider yang kita buat akan berubah sesuai dengan state yang berubah.

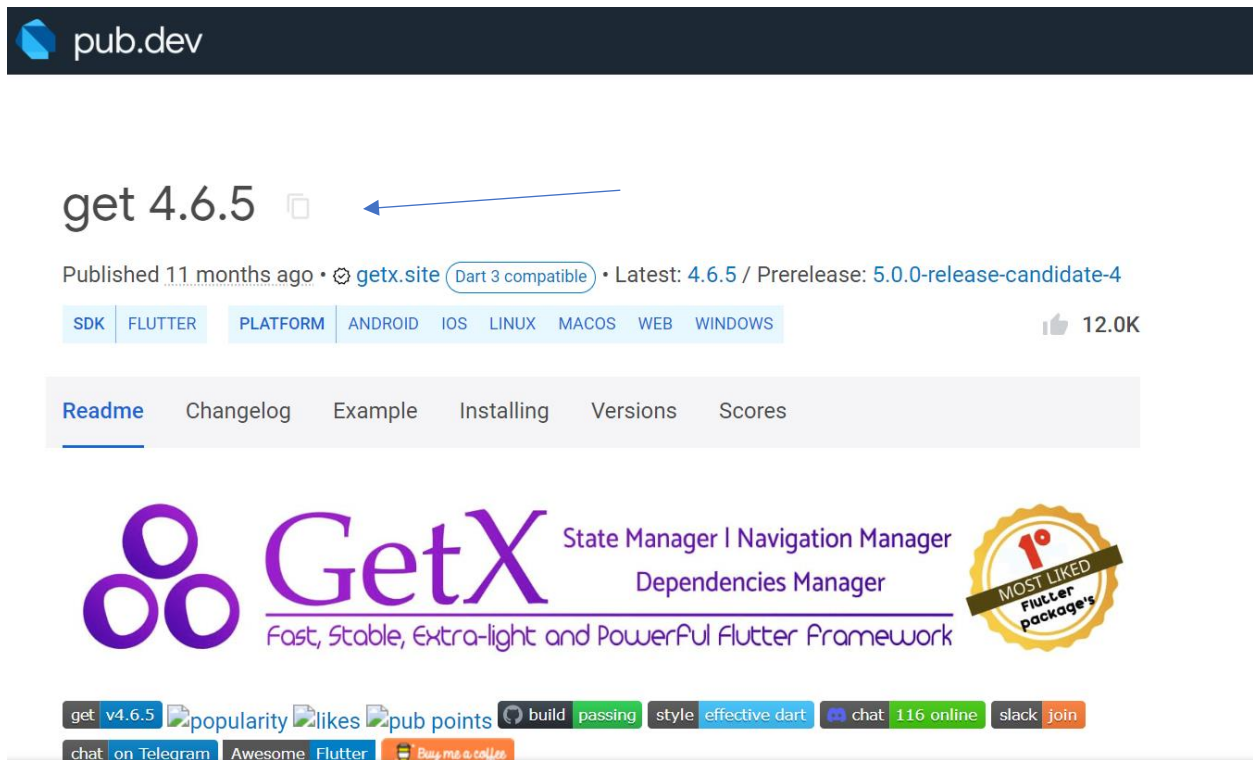
Setiap bagian dalam flutter terbuat dalam bentuk widget. Terdapat dua widget yang memiliki perlakuan state yang berbeda, yaitu *stateful* dan *stateless* widget.

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.

1. Pada perkuliahan kali ini kita akan menggunakan State Management Flutter dengan GetX1. kunjungi <https://pub.dev/> lalu cari package **getX**



Salin nama package dan masukkan ke file pubspec.yaml di projek flutter

```
dependencies:
  flutter:
    sdk: "flutter"

# The following adds the Cupertino Icons font to your application.
# Use with the CupertinoIcons class for iOS style icons.
cupertino_icons: ^1.0.2
material: ^1.0.0+2
english_words: ^4.0.0
get: ^4.6.5
google_fonts: ^4.0.4

dev_dependencies:
```

```
flutter_test:  
  sdk: "flutter"
```

2. buat file di **lib/main.dart** lalu ganti isinya dengan script di bawah ini.

```
import 'home.dart';  
import 'package:flutter/material.dart';  
  
void main() {  
  runApp(const MyApp());  
}  
  
class MyApp extends StatelessWidget {  
  const MyApp({Key? key}) : super(key: key);  
  
  // This widget is the root of your application.  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Flutter Demo',  
      debugShowCheckedModeBanner: false,  
      home: Scaffold(  
        appBar: AppBar(  
          title: Text("List Product"),  
        ),  
        body: Home(),  
      ),  
    );  
  }  
}
```

3. Buat file **home.dart** di dalam folder **lib** lalu ikuti scriptnya seperti di bawah ini. Penjelasan singkatnya sudah saya tulis di dalam script di bawah.

```
import 'controller.dart';  
import 'total_page.dart';  
import 'package:flutter/material.dart';  
import 'package:get/get.dart';  
  
//saat menggunakan state management kita tidak perlu menggunakan stateful lagi  
class Home extends StatelessWidget {  
  //inisialisai  
  final Controller c = Get.put(Controller());
```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Padding(
      padding: const EdgeInsets.all(15),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Row(
            mainAxisAlignment: MainAxisAlignment.spaceAround,
            children: [
              Container(
                width: 80,
                height: 80,
                decoration: BoxDecoration(
                  borderRadius: BorderRadius.circular(15),
                  image: DecorationImage(
                    image: NetworkImage(
                      'https://cdn.pixabay.com/photo/2011/03/16/16/01/tomatoes-5356\_\_340.jpg',
                    ),
                    fit: BoxFit.cover,
                  ),
                ),
              ),
            ],
          ),
          Row(
            children: [
              GestureDetector(
                onTap: () => {
                  //saat ditap maka variabel tomat akan bertambah
                  //fungsi addTomato() bisa dilihat di file
controller.dart

                  c.addTomato(),
                },
                child: Container(
                  width: 45,
                  height: 45,
                  decoration: BoxDecoration(
                    borderRadius: BorderRadius.circular(15),
                    color: Colors.black12,
                  ),
                  child: Icon(Icons.add),
                ),
              ),
            ],
          ),
          SizedBox(

```

```

        width: 10,
      ),
      //variabel tomato akan secara reaktif bertambah atau
      berkurang saat tombol + atau - ditap
      //perlu dibungkus dengan Obx() agar variabel di dalamnya
      dapat berubah secara reaktif
      Obx(
        () => Text(
          "${c.tomato.toString()}",
          style: TextStyle(
            fontSize: 15,
          ),
        ),
      ),
      SizedBox(
        width: 10,
      ),
      GestureDetector(
        onTap: () => {
          //saat ditap maka variabel tomat akan berkurang
          //fungsi removeTomato() bisa dilihat di file

```

controller.dart

```

        c.removeTomato(),
      },
      child: Container(
        width: 45,
        height: 45,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(15),
          color: Colors.black12,
        ),
        child: Icon(Icons.remove),
      ),
    ],
  ),
],
),
SizedBox(
  height: 20,
),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceAround,
  children: [
    Container(

```

```

        width: 80,
        height: 80,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(15),
          image: DecorationImage(
            image: NetworkImage(
              'https://cdn.pixabay.com/photo/2018/10/03/21/57/cabbage
-3722498_960_720.jpg',
            ),
          fit: BoxFit.cover,
        ),
      ),
    ),
  ),
  Row(
    children: [
      GestureDetector(
        //saat ditap maka variabel cabbage akan bertambah
        //fungsi addCabbage() bisa dilihat di file
        onTap: () => c.addCabbage(),
        child: Container(
          width: 45,
          height: 45,
          decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(15),
            color: Colors.black12,
          ),
          child: Icon(Icons.add),
        ),
      ),
      SizedBox(
        width: 10,
      ),
      //variabel cabbage akan secara reaktif bertambah atau
      berkurang saat tombol + atau - ditap
      //perlu dibungkus dengan Obx() agar variabel di dalamnya
      dapat berubah secara reaktif
      Obx(
        () => Text(
          "${c.cabbage.toString()}",
          style: TextStyle(
            fontSize: 15,
          ),
        ),
      ),
    ],
  ),

```

controller.dart

```

        SizedBox(
          width: 10,
        ),
        GestureDetector(
          //saat ditap maka variabel cabbage akan berkurang
          //fungsi removeCabbage() bisa dilihat di file
controller.dart

          onTap: () => c.removeCabbage(),
          child: Container(
            width: 45,
            height: 45,
            decoration: BoxDecoration(
              borderRadius: BorderRadius.circular(15),
              color: Colors.black12,
            ),
            child: Icon(Icons.remove),
          ),
        ),
      ],
    ),
  ],
),
SizedBox(
  height: 25,
),
GestureDetector(
  onTap: () {
    Navigator.push(
      context,
      // TotalPage adalah halaman yang dituju
      MaterialPageRoute(
        builder: (context) => TotalPage(),
      ),
    );
  },
  child: Container(
    margin: EdgeInsets.symmetric(horizontal: 40),
    height: 50,
    width: MediaQuery.of(context).size.width,
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(15),
      color: Colors.blue,
    ),
    child: Center(
      child: Text(

```

```

        "Check Total",
        style: TextStyle(
          fontSize: 20,
          color: Colors.white,
        ),
      ),
    ),
  ),
)
],
),
),
);
}
}

```

4. Buat file **controller.dart** di dalam folder **lib** lalu ikuti scriptnya seperti di bawah ini.

```

import 'package:get/get.dart';

class Controller extends GetxController {
  //inisialisasi variabel yang dibutuhkan
  var tomato = 0.obs;
  var cabbage = 0.obs;

  //fungsi untuk menambah jumlah tomat
  addTomato() {
    return tomato.value++;
  }

  //fungsi untuk mengurangi jumlah tomat
  removeTomato() {
    if (tomato.value > 0) {
      return tomato.value--;
    }
  }

  //fungsi untuk menambah jumlah cabbage
  addCabbage() {
    return cabbage.value++;
  }
}

```



```

//fungsi untuk mengurangi jumlah cabbage
removeCabbage() {
  if (cabbage.value > 0) {
    return cabbage.value--;
  }
}

//fungsi untuk menghitung jumlah tomat dan cabbage
sumTotal() {
  return tomato.value + cabbage.value;
}
}

```

5. Buatlah file **total\_page.dart** di dalam folder **lib** lalu ikuti scriptnya seperti di bawah ini.

```

import 'package:flutter/material.dart';
import 'package:get/get.dart';

import 'controller.dart';

class TotalPage extends StatelessWidget {

  //inisialisasi
  final Controller c = Get.put(Controller());

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Total Product"),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          // ignore: prefer_const_literals_to_create_immutables
          children: [
            Text(
              "Total Product",
              style: TextStyle(fontSize: 25),
            ),
            Text(
              //mengambil hasil perhitungan dari tomat dan cabbage
              c.sumTotal().toString(),
              style: TextStyle(

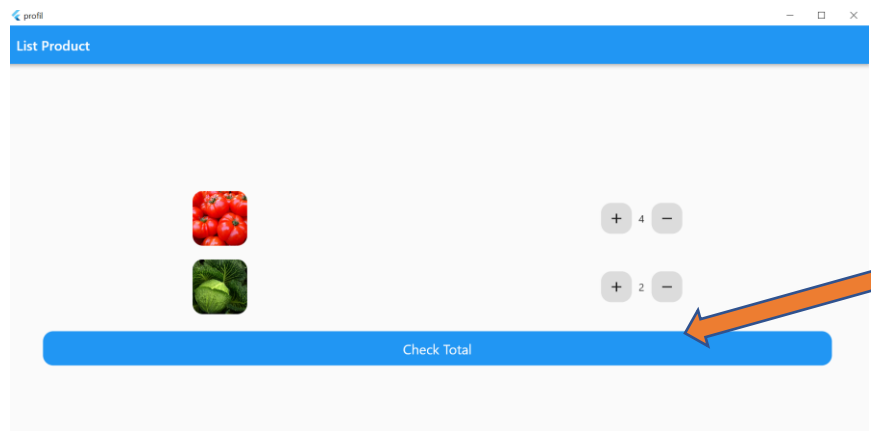
```

```

        },
      ],
    ),
  );
}
}

```

6. Jalankan projek melalui terminal dengan syntax *flutter run*



**Klik check total**

