

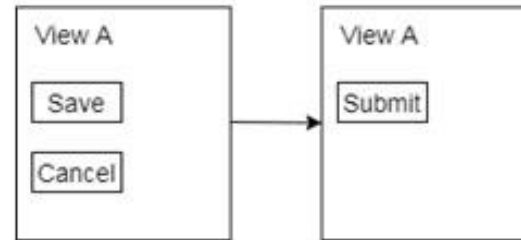


# MOBILE APPLICATION

PERTEMUAN 14

# FLUTTER STATE MANAGEMENT

- ▶ Sebuah cara untuk berbagi status aplikasi antar layar, antar aplikasi
- ▶ Flutter menggunakan cara yang berbeda untuk mengambil dan memroses data



## Procedural Approach

```
a = getComponent(viewA);  
a.removeChildren();  
Button btn = new Button();  
btn.setText('Submit');  
a.add(btn);
```

## Declarative Approach

```
return ViewA(  
  child: Button(  
    text: 'Submit'  
  );  
);
```

# Lanjutan

- ▶ Flutter bersifat deklaratif. Ini berarti Flutter membangun antarmuka penggunaannya untuk mencerminkan status aplikasi saat ini:

$$\text{UI} = f(\text{state})$$

The layout  
on the screen

Your  
build  
methods

The application state

# Contoh State

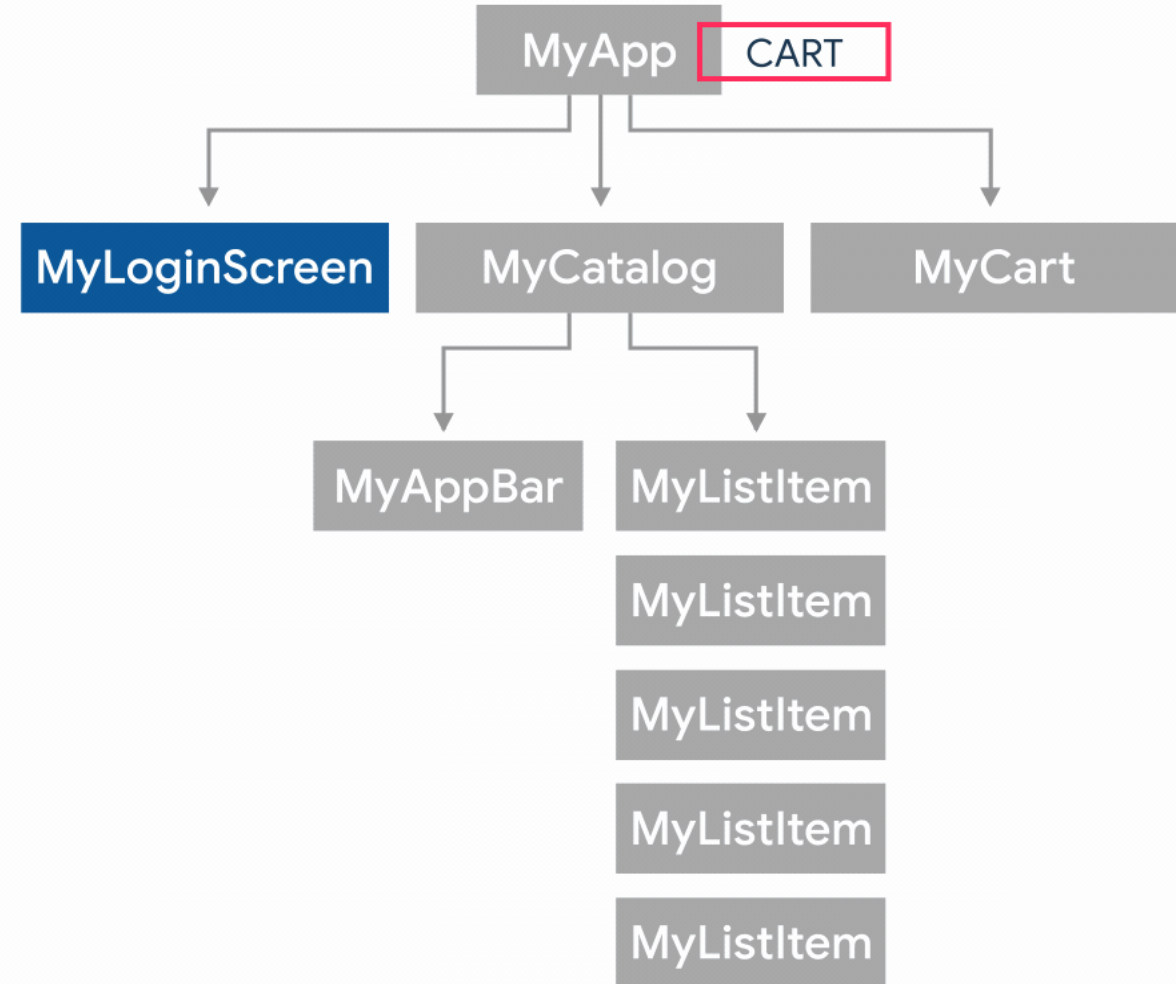
- ▶ Preferensi pengguna
- ▶ Info masuk
- ▶ Notifikasi di aplikasi jejaring sosial
- ▶ Keranjang belanja di aplikasi e-niaga
- ▶ Status artikel yang dibaca/belum dibaca di aplikasi berita

# Welcome

Login

Password

Enter



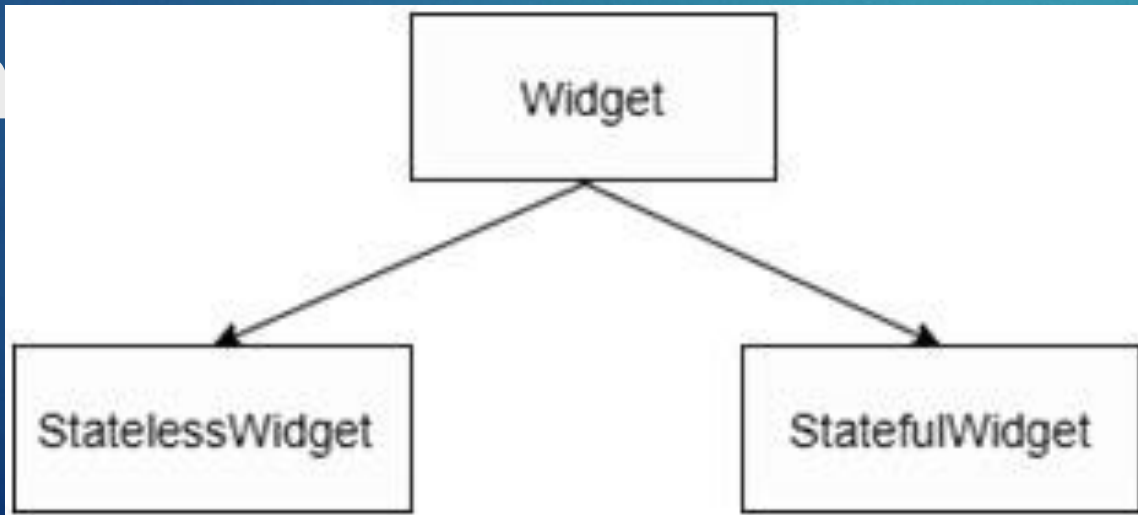
Widget stateless tidak pernah berubah

Widget stateful dapat mengubah tampilannya sebagai respons terhadap interaksi pengguna atau saat menerima data

## Jenis State Widget



Lanjutan  
Flutter memiliki hierarki kelas Widget berikut. Kita dapat memperluas kelas Widget Stateless atau kelas Widget Stateful untuk mendefinisikan widget stateless atau stateful kustom kita masing-masing.



karena variabel instansnya selalu final.  
Stateless Widget  
Semua variable bersifat final dan  
permanen isinya

Setiap upaya untuk menginisialisasi ulang  
mereka akan menimbulkan kesalahan

Widget stateless harus selalu mengganti  
metode build. Metode ini menggunakan

```
// A reusable custom Stateless widget
class MovieName extends StatelessWidget {
  final String name;
  final int count = 0;
}
```

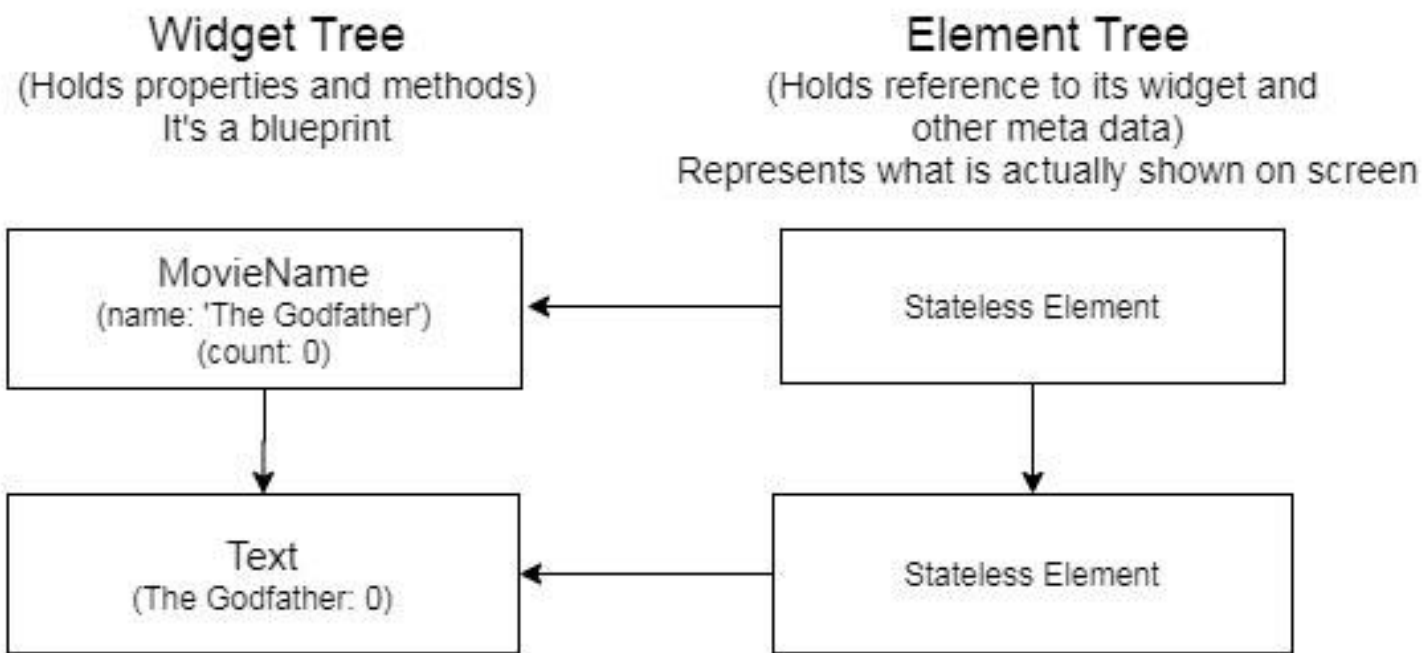


UI.

Prosesan Stateless  
Setiap kelas widget memiliki kelas Elemen yang sesuai dan metode untuk membuat instance-nya

Misalnya, kelas StatelessWidget memiliki kelas StatelessElement yang sesuai dan metode createElement() untuk membuat instance-nya. Elemen inilah yang sebenarnya mewakili apa yang ditampilkan di layar perangkat.

# Penjelasan



dapat  
ditampilkan di  
layar.

Widget  
MovieName  
memiliki Text  
sebagai anaknya. Jadi  
elemen  
memanggil  
metode `build()`



# Stateful : Apa itu State?

STATE ADALAH NILAI DARI SEMUA VARIABEL YANG DIBUTUHKAN UNTUK MEMBUAT ANTARMUKA PENGGUNA.

Stateless vs Stateful

**Widget Stateful** bekerja dengan dua kelas:

Subkelas StatefulWidget yang mendefinisikan widget. Hal ini tidak dapat diubah.

Subkelas Status untuk menyimpan status (data) untuk widgetnya dan metode build() untuk membuat turunan untuk widgetnya.

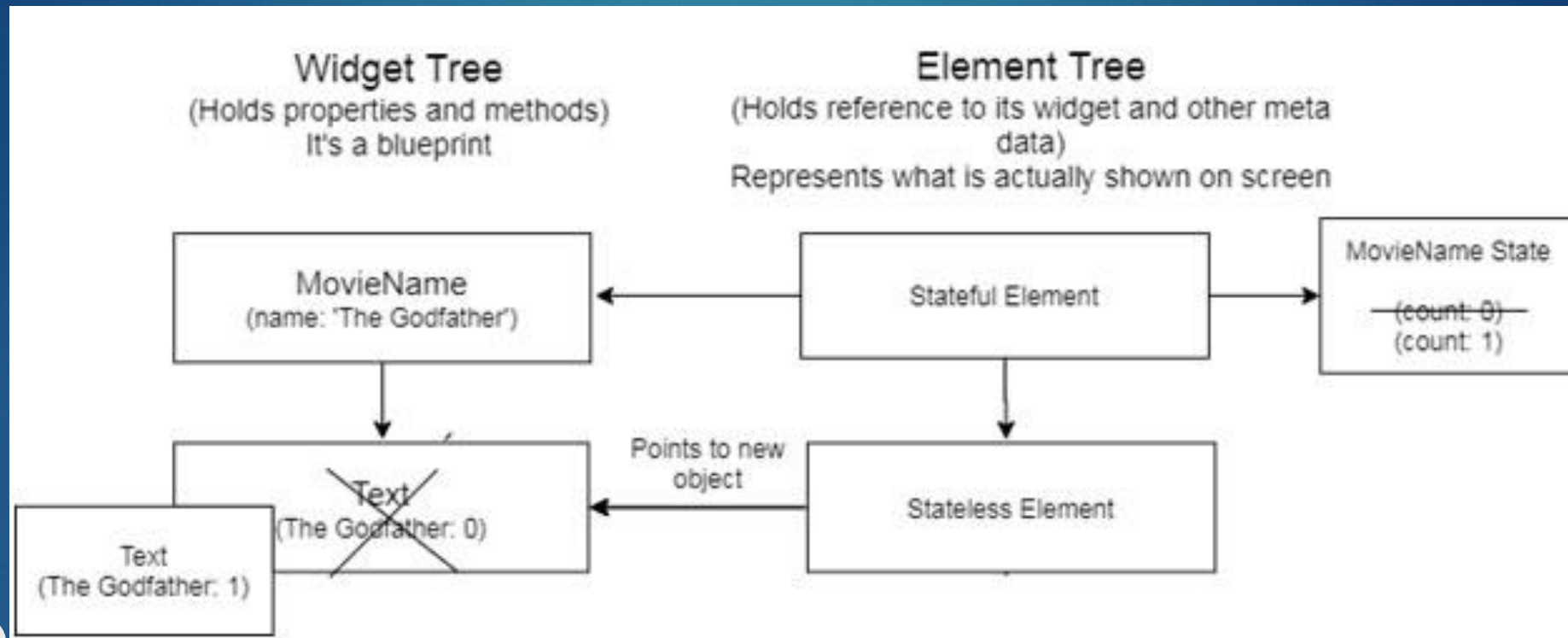


# Contoh

```
class MovieName extends StatefulWidget {  
  final String name;  
  MovieName({this.name});  
  _MovieNameState createState() => _MovieNameState();  
}  
  
class _MovieNameState extends State<MovieName> {  
  int count = 0;  
  
  @override  
  Widget build(BuildContext context) {  
    return GestureDetector(  
      child: Text(  
        '${widget.name}: $count',  
      ),  
      onTap: () {  
        setState(() {  
          count++;  
        });  
      },  
    );  
  }  
}
```

Bagian awal kode adalah deklarasi untuk stateful widget. Sedangkan item item widget dan perubahan data dilakukan di

# Pemrosesan Stateful



Mirip dengan stateless sebelumnya,  
namun diganti dengan Stateful Element  
yang mengendalikan objek Text dengan



ini saat melakukan pagination.

State: Lokal dan Global

Global: Nyatakan yang ingin kita gunakan di berbagai bagian aplikasi kita, mis. pengaturan pengguna di aplikasi Whatsapp atau daftar makanan favorit di aplikasi makanan.

Mengelola Status dengan setState dan StatefulWidget bisa menjadi kompleks dengan mengangkat State dan Callbacks, tetapi untuk aplikasi yang kompleks, jika callbacks perlu melintasi seluruh struktur atau melewati beberapa level ke bawah, maka state tersebut dapat segera menjadi kompleks.

pemeliharaan karena State Anda  
terserbit di semua tempat

Pengembang akhirnya mencampur UI  
dan logika bisnis, yang bukan merupakan  
prinsip pengkodean yang baik.