

StatelessWidget dan StatefulWidget

Seperti yang kita tahu jantung dari aplikasi Flutter adalah widget. Sebagian besar yang ada pada Flutter merupakan widget. Membuat tombol, menampilkan gambar, text, dan membuat tampilan berada di tengah pada Flutter semuanya menggunakan widget. Kita juga dapat membuat widget sendiri untuk dapat digunakan lain waktu ataupun dibagikan kepada Flutter developer lain (dalam bentuk packages).

Widget pada Flutter memiliki dua jenis, yaitu StatelessWidget dan StatefulWidget. Sebagai developer Flutter, kita harus memahami betul kedua jenis widget tersebut, maka pada bagian ini kita akan mempelajari apa itu StatelessWidget dan StatefulWidget.

Apa itu State?

Sebelum membahas kedua jenis widget tersebut, kita harus berkenalan terlebih dahulu dengan istilah *State*. Kenapa demikian? Widget kita akan terus berurusan dengan State. Lalu apa itu State?

Untuk teman-teman dengan background *frontend web developer*, tentu tak akan asing dengan istilah State ini, terutama menggunakan framework ReactJS. Tetapi untuk Anda tanpa *background* tersebut tidak perlu risau. State tidaklah sulit untuk dimengerti. Jadi State adalah data yang ada pada suatu widget. Widget menyimpan data yang nantinya dapat berubah sesuai interaksi pengguna.

Karena Flutter menggunakan paradigma OOP (Object-Oriented Programming), state biasanya menjadi suatu properti dari sebuah class. Contohnya sebagai berikut:

```
1. class ContohWidget extends StatelessWidget{  
2.     final String _judul  
3.     ...  
4. }
```

Variabel `_judul` di atas merupakan contoh pendeklarasian state pada suatu widget.

StatelessWidget

Setelah mengenal apa itu state, maka yang pertama kita akan bahas adalah StatelessWidget. StatelessWidget adalah widget yang nilai *state*-nya tidak dapat berubah (*immutable*) maka widget tersebut lebih bersifat statis dan memiliki interaksi yang terbatas.

Sekarang kita akan membuat sebuah Widget sederhana:

```
1. class Heading extends StatelessWidget {  
2.  
3.   final String text;  
4.  
5.   Heading({this.text});  
6.  
7.   @override  
8.   Widget build(BuildContext context){  
9.     return Text(  
10.       text,  
11.       style: TextStyle(  
12.         fontSize: 24.0,  
13.         fontWeight: FontWeight.bold,  
14.       ),  
15.     );  
16.   }  
17. }
```

Widget di atas merupakan sebuah widget untuk membuat Heading (sebuah text yang digunakan untuk judul). Kita akan panggil widget yang telah diubah ke kode project pertama kita.

```
1. import 'package:flutter/material.dart';  
2.  
3. void main() => runApp(MyApp());  
4.  
5. class MyApp extends StatelessWidget {  
6.   @override  
7.   Widget build(BuildContext context) {  
8.     return MaterialApp(  
9.       title: 'Flutter Demo',  
10.      theme: ThemeData(  
11.        primarySwatch: Colors.blue,  
12.      ),  
13.      home: Scaffold(  
14.        body: Center(  
15.          child: Text("Hello world!"),  
16.        ),  
17.      ),  
18.    );  
19.  }
```

```

17.     ),
18.   );
19. }
20. }
21.
22. class Heading extends StatelessWidget {
23.   final String text;
24.
25.   Heading({this.text});
26.
27.   @override
28.   Widget build(BuildContext context){
29.     return Text(
30.       text,
31.       style: TextStyle(
32.         fontSize: 24.0,
33.         fontWeight: FontWeight.bold,
34.       ),
35.     );
36.   }
37. }

```

Kita coba ubah widget **Text** yang menampilkan "Hello world!" dengan widget **Heading** yang kita buat.

```

1. import 'package:flutter/material.dart';
2.
3. void main() => runApp(MyApp());
4.
5. class MyApp extends StatelessWidget {
6.   @override
7.   Widget build(BuildContext context) {
8.     return MaterialApp(
9.       title: 'Flutter Demo',
10.      theme: ThemeData(
11.        primarySwatch: Colors.blue,
12.      ),
13.      home: Scaffold(
14.        body: Center(
15.          child: Heading( // mengubah widget Text
16.            text:"Hello world!",
17.          ),
18.        ),
19.      ),
20.    );
21.  }
22. }
23.
24. class Heading extends StatelessWidget {
25.   final String text;
26.
27.   Heading({this.text});
28.

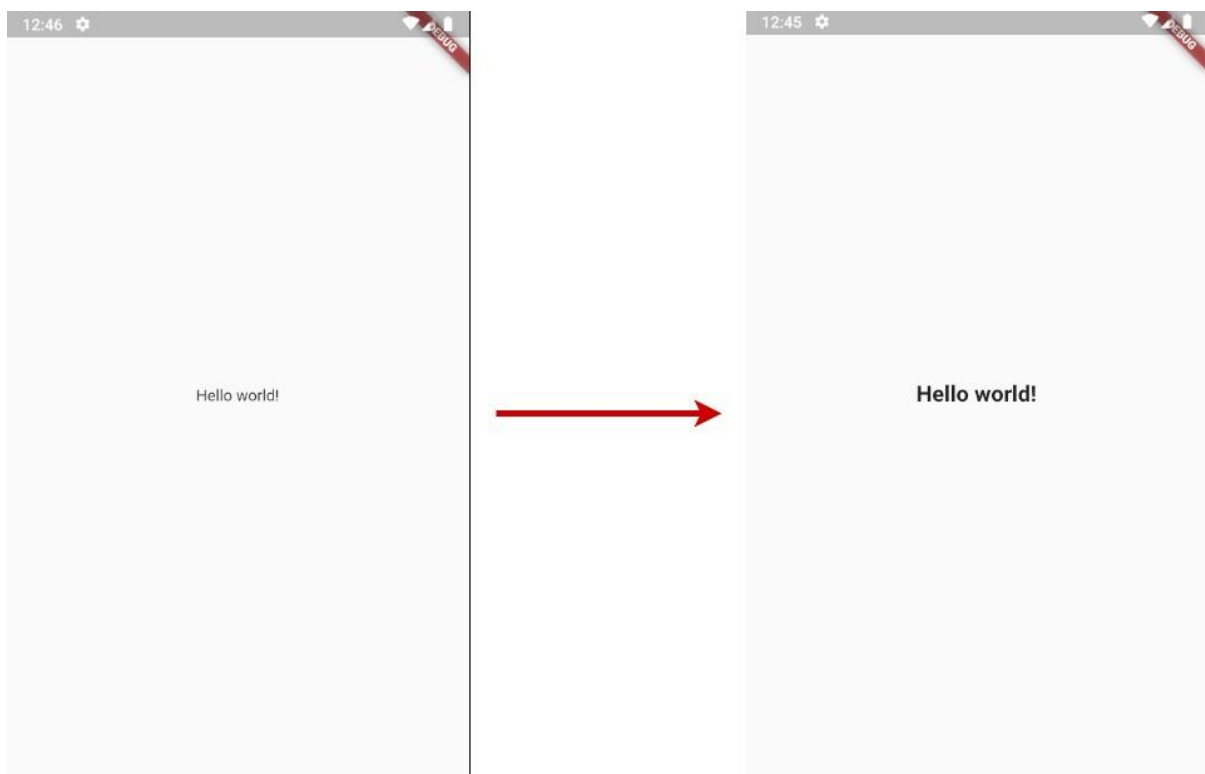
```

```

29.  @override
30.  Widget build(BuildContext context){
31.    return Text(
32.      text,
33.      style: TextStyle(
34.        fontSize: 24.0,
35.        fontWeight: FontWeight.bold,
36.      ),
37.    );
38.  }
39. }

```

Maka ketika kita ubah `Text` dengan widget `Heading`, hasilnya akan berubah. Tulisan "Hello world!" jadi lebih besar.



Sesuai definisi `StatelessWidget`, *state*-nya tidak dapat berubah (*immutable*), maka *state* yang ada di dalam kelas tersebut harus dibuat `final`. Nilainya hanya dapat diisi melalui *constructor class*-nya.

```

1. final String text; // state text bersifat final
2.
3. Heading({this.text}); // lalu state text masuk ke constructor

```

StatefulWidget

Kebalikan dari StatelessWidget, StatefulWidget ialah widget yang *state*-nya dapat berubah-ubah nilainya, yang berarti StatefulWidget bersifat dinamis dan memiliki interaksi yang tak terbatas.

Penulisan StatefulWidget sangat berbeda dengan StatelessWidget, berikut penulisannya:

```
1. class ContohStateful extends StatefulWidget {  
2.  
3.   final String parameterWidget; // ini parameter widget  
4.  
5.   ContohStateful({this.parameterWidget});  
6.  
7.   @override  
8.   _ContohStatefulState createState() => _ContohStatefulState();  
9. }  
10.  
11. class _ContohStatefulState extends State<ContohStateful>{  
12.   String _dataState; // ini state dari Widget ContohStateful  
13.  
14.   @override  
15.   Widget build(BuildContext context){  
16.     // isi sebuah widget  
17.   }  
18. }
```

Contoh di atas adalah cara penulisan StatefulWidget. Seperti yang kita lihat, penulisan StatefulWidget lebih panjang. Tetapi yang harus diperhatikan adalah properti dari setiap *class*-nya. Pada class **ContohStateful** propertinya hanya berupa parameter ketika memanggil ContohStateful, parameter tersebut tidak wajib ada. Sedangkan pada class **_ContohStatefulState**, properti **_dataState** merupakan state yang sebenarnya. Kita akan mengubah nilai **_dataState**.

Misalnya kita coba membuat contoh StatefulWidget sederhana:

```
1. class BiggerText extends StatefulWidget {  
2.   final String text;  
3.  
4.   const BiggerText({this.text});  
5.  
6.   @override  
7.   _BiggerTextState createState() => _BiggerTextState();  
8. }  
9.  
10.
```

```

11. class _BiggerTextState extends State<BiggerText> {
12.   double _textSize = 16.0;
13.
14.   @override
15.   Widget build(BuildContext context) {
16.     return Column(
17.       mainAxisAlignment: MainAxisAlignment.center,
18.       children: <Widget>[
19.         Text(widget.text, style: TextStyle(fontSize: _textSize)),
20.         ElevatedButton(
21.           child: Text("Perbesar"),
22.           onPressed: () {
23.             setState(() {
24.               _textSize = 32.0;
25.             });
26.           },
27.         ),
28.       ],
29.     );
30.   }
31. }

```

Letakkan kode di atas setelah StatelessWidget Heading yang telah kita buat sebelumnya lalu panggil StatefulWidget **BiggerText** pada MyApp.

```

1. import 'package:flutter/material.dart';
2.
3. void main() => runApp(MyApp());
4.
5. class MyApp extends StatelessWidget {
6.   @override
7.   Widget build(BuildContext context) {
8.     return MaterialApp(
9.       title: 'Flutter Demo',
10.      theme: ThemeData(
11.        primarySwatch: Colors.blue,
12.      ),
13.      home: Scaffold(
14.        body: Center(
15.          child: BiggerText(text:"Hello world!"), // Ubah widget Heading ke PerubahanText
16.        ),
17.      ),
18.    );
19.  }
20. }
21.
22. class Heading extends StatelessWidget {
23.   final String text;
24.
25.   Heading({this.text});
26.
27.   @override

```

```

28.   Widget build(BuildContext context) {
29.     return Text(
30.       text,
31.       style: TextStyle(
32.         fontSize: 24.0,
33.         fontWeight: FontWeight.bold,
34.       ),
35.     );
36.   }
37. }
38.
39. class BiggerText extends StatefulWidget {
40.   final String text;
41.
42.   const BiggerText({this.text});
43.
44.   @override
45.   _BiggerTextState createState() => _BiggerTextState();
46. }
47.
48. class _BiggerTextState extends State<BiggerText> {
49.   double _textSize = 16.0;
50.
51.   @override
52.   Widget build(BuildContext context) {
53.     return Column(
54.       mainAxisAlignment: MainAxisAlignment.center,
55.       children: <Widget>[
56.         Text(widget.text, style: TextStyle(fontSize: _textSize)),
57.         ElevatedButton(
58.           child: Text("Perbesar"),
59.           onPressed: () {
60.             setState(() {
61.               _textSize = 32.0;
62.             });
63.           },
64.         ),
65.       ],
66.     );
67.   }
68. }

```

Maka hasilnya akan seperti berikut:



Ketika tombol "Perbesar" ditekan, text "Hello world!" akan membesar karena state `_textSize` diubah nilainya. Mengubah nilai state harus dilakukan pada fungsi `setState` seperti berikut:

```
1. setState(() {  
2.   _textSize = 32.0; // ukuran text diubah menjadi 32  
3. });
```

Anda dapat memahami lebih dalam terkait Stateless dan Stateful Widget dengan membaca dokumentasi berikut ini:

- [StatelessWidget Class](#)
- [StatefulWidget Class](#)