

Modul 14

Testing, Debugger and Deployment

Testing Flutter widget

Module Overview

Melakukan testing terhadap *widget* guna memitigasi kesalahan yang mungkin terdapat dalam aplikasi Flutter.

Module Objectives

Setelah mempelajari dan mempraktikkan modul ini, mahasiswa diharapkan dapat:

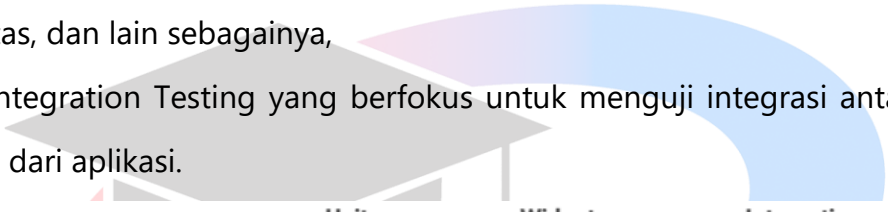
- Mengetahui apa yang dimaksud dengan *widget testing*
- Mengetahui cara melakukan *widget testing* bagaimana cara melakukannya

Setelah aplikasi Anda telah selesai dibangun maka langkah selanjutnya yang harus Anda lakukan adalah melakukan uji coba terhadap aplikasi Anda. Langkah ini penting untuk dilakukan agar Anda dapat memastikan bahwa aplikasi Anda sudah siap pakai dan minim akan kesalahan (*bug*) ketika Anda berniat untuk mempublish maupun membagikan aplikasi Anda kepada pengguna umum.

Pada umumnya terdapat beberapa jenis testing yang dapat Anda lakukan untuk menguji aplikasi Anda baik secara otomatis maupun secara manual. Namun perlu

Anda perhatikan semakin kompleks fitur yang Anda bangun maka akan semakin sulit untuk mengujinya secara manual. Untuk itu pada pertemuan ini Anda akan belajar cara melakukan testing secara otomatis dengan menuliskan sendiri kode program testing Anda. Pengujian secara otomatis dapat dibedakan menjadi beberapa jenis, yaitu:

- Unit Testing yang berfokus untuk menguji sebuah fungsi, *method*, dan *class*,
- Widget Testing yang berfokus untuk menguji *widget* termasuk fungsionalitas, visibilitas, dan lain sebagainya,
- serta Integration Testing yang berfokus untuk menguji integrasi antar bagian-bagian dari aplikasi.



	Unit	Widget	Integration
Confidence	Low	Higher	Highest
Maintenance cost	Low	Higher	Highest
Dependencies	Few	More	Most
Execution speed	Quick	Quick	Slow

Perbedaan Ketiga Jenis Testing dalam Beberapa Aspek

(Sumber: <https://docs.flutter.dev/testing>)

Bagaimana jika Anda ingin melakukan pengujian-pengujian tersebut pada aplikasi Flutter Anda? Untuk dapat melakukan hal ini Anda dapat memanfaatkan beberapa *package* yang tersedia seperti misalnya `flutter_test`, `espresso`, `mockito`, dan lain sebagainya. Saat ini Anda akan lebih berfokus untuk belajar menggunakan **`flutter_test`** yang mana *package* ini merupakan *package* yang sudah diikutsertakan secara *default* oleh Flutter ketika Anda membuat sebuah proyek baru.

```
40  ✓ dev_dependencies:
41  ✓   flutter_test:
42  ✓   | sdk: flutter
```

Selain itu Anda juga dapat melihat contoh kode program testing sederhana yang dilakukan terhadap *widget* aplikasi Anda pada file **main.dart** yang dibuat secara otomatis ketika Anda membuat sebuah proyek baru dalam folder **test/widget_test.dart**.

Widget Test

Untuk dapat melakukan widget test pada Flutter maka Anda dapat menggunakan fungsi **testWidget()** yang merupakan bagian dari package **flutter_test**. Adapun fungsi **testWidget()** secara keseluruhan terdiri atas **testWidgets(String description, WidgetTesterCallback callback { })**, di mana:

- **description** : deskripsi pengujian yang akan dilakukan,
- **callback** : menerima instance dari **WidgetTester** sehingga Anda dapat berinteraksi dengan widget serta tempat Anda menulis kode program testing aplikasi Anda.

1. Rancanglah tampilan aplikasi berikut yang akan Anda gunakan untuk melakukan pengujian.

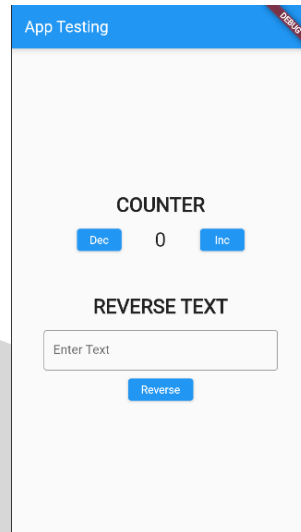
```
1  import 'package:flutter/material.dart';
2  import 'package:flutter/src/widgets/container.dart';
3  import 'package:flutter/src/widgets/framework.dart';
4
5  class MainScreen extends StatefulWidget {
6    const MainScreen({super.key});
7
8    @override
9    State<MainScreen> createState() => _MainScreenState();
10 }
11
12 class _MainScreenState extends State<MainScreen> {
13   TextEditingController textfield = TextEditingController();
14   int value = 0;
```

```

16  @override
17  Widget build(BuildContext context) {
18    return Scaffold(
19      appBar: AppBar(title: Text("App Testing")),
20      body: Container(
21        margin: EdgeInsets.all(40),
22        child: Column(
23          mainAxisAlignment: MainAxisAlignment.center,
24          crossAxisAlignment: CrossAxisAlignment.center,
25          children: [
26            Padding(
27              padding: EdgeInsets.all(15),
28              child: Text("COUNTER", style: TextStyle(fontWeight: FontWeight.bold, fontSize: 25)),
29            ),
30            Row(
31              mainAxisAlignment: MainAxisAlignment.spaceEvenly,
32              children: [
33                ElevatedButton(
34                  onPressed: () {
35                    setState(() {
36                      value--;
37                    });
38                  },
39                  child: Text("Dec", key: Key("btnDec")), // ElevatedButton
40                ),
41                ElevatedButton(
42                  onPressed: () {
43                    setState(() {
44                      value++;
45                    });
46                  },
47                  child: Text("Inc", key: Key("btnInc")), // ElevatedButton
48                ),
49              ], // Row
50            ),
51            Padding(padding: EdgeInsets.only(top: 20, bottom: 20)),
52            Padding(
53              padding: EdgeInsets.all(15),
54              child: Text("REVERSE TEXT", style: TextStyle(fontWeight: FontWeight.bold, fontSize: 25)),
55            ), // Padding
56            TextField(
57              controller: textfield,
58              decoration: InputDecoration(
59                hintText: "Enter Text", border: OutlineInputBorder(), // InputDecoration
60              ), // TextField
61            ),
62            Padding(
63              padding: EdgeInsets.only(top: 10),
64              child: ElevatedButton(
65                onPressed: () {
66                  setState(() {
67                    textfield.text = String.fromCharCode(
68                      textfield.text.runes.toList().reversed); // String.fromCharCode
69                  });
70                },
71                child: Text("Reverse", key: Key("btnReverse")), // ElevatedButton // Padding
72              ),
73            ),
74          ], // Column // Container // Scaffold
75        ),
76      ),
77    );
78  }
79

```

Jika kode program tersebut dijalankan maka tampilan aplikasi yang muncul adalah sebagai berikut ini:



2. Pada **test/widget_test.dart**, tambahkan skenario pengujian untuk mengecek apakah teks pada tampilan yang Anda buat sudah sesuai atau tidak.

```

14   testWidgets('Test Text Widget', (WidgetTester tester) async {
15     await tester.pumpWidget(const MyApp());
16     expect(find.text('COUNTER'), findsOneWidget);
17     expect(find.text('Dec'), findsOneWidget);
18     expect(find.text('Kurang'), findsNothing);
19     expect(find.text('0'), findsOneWidget);
20     expect(find.text('Inc'), findsOneWidget);
21     expect(find.text('Tambah'), findsNothing);
22     expect(find.text('REVERSE TEXT'), findsOneWidget);
23     expect(find.text('Reverse'), findsOneWidget);
24   });

```

Keterangan:

- expect : mengecek kesesuaian widget dengan yang seharusnya
- find : mencari widget yang akan Anda uji berdasarkan kriteria tertentu
- findsOneWidget : memastikan kriteria yang Anda cari setidaknya terdapat pada satu dari keseluruhan widget
- findsNothing : memastikan kriteria yang Anda cari tidak terdapat pada keseluruhan widget

3. Jalankan kode program test yang sudah Anda buat dengan menekan tombol F5. Jika seluruh kriteria test telah dipenuhi maka Anda dapat melihat hasilnya pada bagian **Console**.

PROBLEMS 29 OUTPUT DEBUG CONSOLE TERMINAL

```

Exited (1)
Connecting to VM Service at http://127.0.0.1:60888/Z_-IS6r004w=/ws
✓ Test Text Widget
Exited

```

4. Selanjutnya tambahkan pengujian untuk mengecek fungsionalitas pada bagian Counter. Di sini Anda akan menguji Counter dengan melakukan klik pada tombol Inc dan Dec serta mengecek apakah hasil nilainya berubah atau tidak.

```

25 testWidgets('Test Counter', (WidgetTester tester) async {
26   await tester.pumpWidget(const MyApp());
27   expect(find.text('0'), findsOneWidget);
28   await tester.tap(find.byKey(Key("btnInc")));
29   await tester.pump();
30   expect(find.text('1'), findsOneWidget);
31   await tester.tap(find.widgetWithText(ElevatedButton, 'Dec'));
32   await tester.tap(find.widgetWithText(ElevatedButton, 'Dec'));
33   await tester.pump();
34   expect(find.text('1'), findsNothing);
35   expect(find.text('-1'), findsOneWidget);
36 });

```

Keterangan:

- find.byKey : mencari widget berdasarkan key yang Anda gunakan
 - find.widgetWithText : mencari widget yang memiliki child Text sesuai kriteria
 - tester.tap : melakukan uji coba dengan aksi klik terhadap widget
5. Terakhir Anda di sini akan menambahkan pengujian untuk mengecek apakah bagian Reverse Text berfungsi sebagaimana mestinya.

```
37 testWidgets('Test Reverse Text', (WidgetTester tester) async {  
38   await tester.pumpWidget(const MyApp());  
39   expect(find.byType(TextField), findsOneWidget);  
40   await tester.enterText(find.byType(TextField), 'Back End Flutter');  
41   await tester.tap(find.widgetWithText(ElevatedButton, 'Reverse'));  
42   await tester.pump();  
43   expect(find.text('rettulF dnE kcaB'), findsOneWidget);  
44 });
```

Keterangan:

- find.byType : mencari widget berdasarkan tipe widget
- tester.enterText : melakukan uji coba dengan aksi input terhadap widget

Jika keseluruhan test telah ditambahkan, coba jalankan ulang dan lihat apakah testing yang Anda lakukan telah berhasil atau tidak. Jika semua kriteria testing telah dipenuhi maka pada console Anda akan melihat pesan berikut.



PROBLEMS 29 OUTPUT DEBUG CONSOLE TERMINAL

Connecting to VM Service at http://127.0.0.1:60653/1Pw1rJDt1Jk=/ws
✓ Test Text Widget
✓ Test Counter
✓ Test Reverse Text
Exited

Lalu bagaimana jika seandainya terdapat kriteria testing yang tidak terpenuhi? Nantinya Anda dapat melihat kriteria yang tidak terpenuhi tersebut sehingga hal ini akan mempermudah Anda untuk mengecek kode program yang mungkin "bermasalah".

Latihan

1. Kembangkan contoh yang sudah Anda buat sebelumnya dengan menambahkan minimal 5 widget yang belum ada pada contoh serta lakukan widget testing

kepada widget-widget yang baru Anda tambahkan dengan skenario yang Anda tentukan sendiri.



UNIVERSITAS
MIKROSKIL