

Pemrograman User Interface dengan Flutter

Aplikasi Sederhana dengan Text, Column dan Row

Fadjar Fathurrahman

1 Tujuan

- Mampu mengenal dan menggunakan baris perintah flutter
- Mampu membuat dan membangun project Flutter
- Mampu membuat program sederhana berbasis StatelessWidget

2 Baris perintah flutter

Dalam praktikum ini kita akan menggunakan baris perintah flutter untuk membuat dan membangun project Flutter.

Beberapa perintah yang sering digunakan:

- `flutter create nama_project`: membuat project baru dengan nama `nama_project`
- `flutter run -d web`: melakukan testing aplikasi pada web server
- `flutter run -d chrome`: melakukan testing aplikasi pada web server dan menggunakan browser berbasis Chromium (seperti Google Chrome) untuk menjalankan aplikasi.
- `flutter run`: menjalankan testing aplikasi pada default target, biasanya adalah Android atau iOS.

3 Struktur aplikasi Flutter

Buat project baru dengan flutter. Misalkan nama project ini adalah `proj_01`:

```
flutter create proj_01
```

Direktori baru dengan nama `proj_01` akan dibuat oleh flutter beserta file-file lainnya yang diperlukan. Struktur dari direktori `proj_01` kurang lebih adalah sebagai berikut.

```
proj_01/  
├─ android  
├─ ios  
├─ lib  
│   └─ main.dart  
├─ proj_01.iml  
├─ pubspec.lock  
├─ pubspec.yaml  
├─ README.md  
└─ test
```

```
| └─ widget_test.dart
|   web
```

Kode program Flutter (dalam bahasa pemrograman Dart) akan berada dalam subdirektori lib.

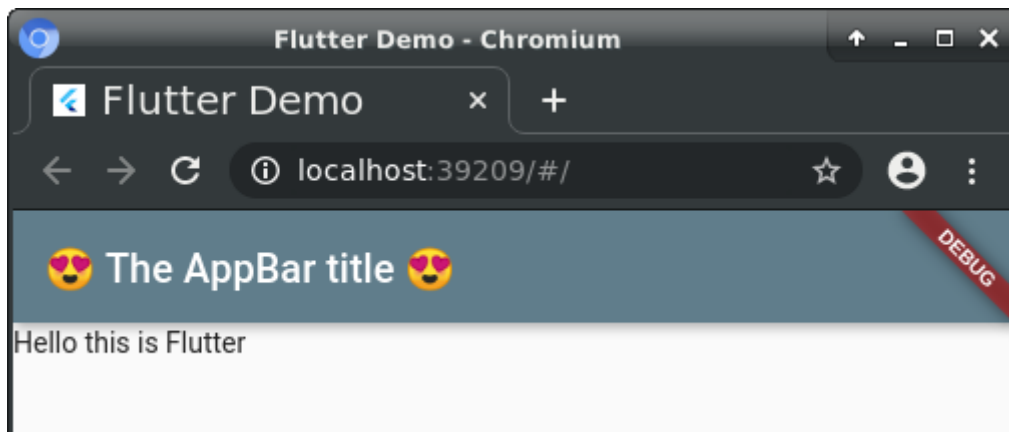
Buka file lib/main.dart, hapus semua kode yang ada didalamnya dan ganti dengan kode berikut.

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

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Judul Aplikasi',
      theme: ThemeData(
        primarySwatch: Colors.blueGrey,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      home: Scaffold(
        appBar: AppBar(title: Text(' The AppBar title ')),
        body: Text("Hello this is Flutter"),
      )
    );
  }
}
```

Tampilan aplikasi ketika dijalankan pada web browser kurang lebih seperti berikut.



Untuk membuat aplikasi Flutter, kita perlu meng-import beberapa pustaka standard pada Flutter. Baris berikut ini akan mengimport elemen-elemen user interface (Material Design, yang dikembangkan oleh Google) yang digunakan oleh Flutter:

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

Seperti pada aplikasi Dart yang lain, fungsi pertama yang akan dijalankan oleh Flutter adalah fungsi main.

```
void main() {
  runApp(MyApp());
}
```

Fungsi main memanggil fungsi runApp() yang menerima input berupa Widget.

Pada Flutter, Widget dapat dianggap sebagai kelas dasar Flutter untuk menampilkan user interface. Hampir seluruh kelas atau komponen yang ada pada Flutter adalah Widget.

MyApp adalah sebuah kelas yang kita definisikan sendiri (di luar pustaka Widget). MyApp diturunkan (melalui inheritance) dari kelas StatelessWidget, yang merupakan kelas dasar Widget yang tidak memiliki keadaan atau bersifat statik atau tidak berubah (*immutable*).

Semua kelas yang mewarisi atau diturunkan dari Widget harus mengimplementasikan fungsi (atau override) build yang akan mengembalikan suatu Widget:

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    // Implementasi
  }
}
```

Kita akan banyak menggunakan pola sebagai berikut untuk top-level widget yang kita gunakan.

```
Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Judul>Nama Aplikasi',
    home: Scaffold(
      appBar: AppBar(title: Text('Judul pada AppBar')),
      body: // Widget Anda di sini,
    )
  );
}
```

Anda dapat mengganti ukuran font dan font family sebagai berikut.

```
Scaffold(
  appBar: AppBar(title: Text(' The AppBar title ')),
  body: Text("Hello this is Flutter",
    style: TextStyle(fontSize: 40, fontFamily: 'Monaco')
  ),
)
```

Ganti bagian body dari Scaffold dengan Column atau Row. Misalnya untuk Column

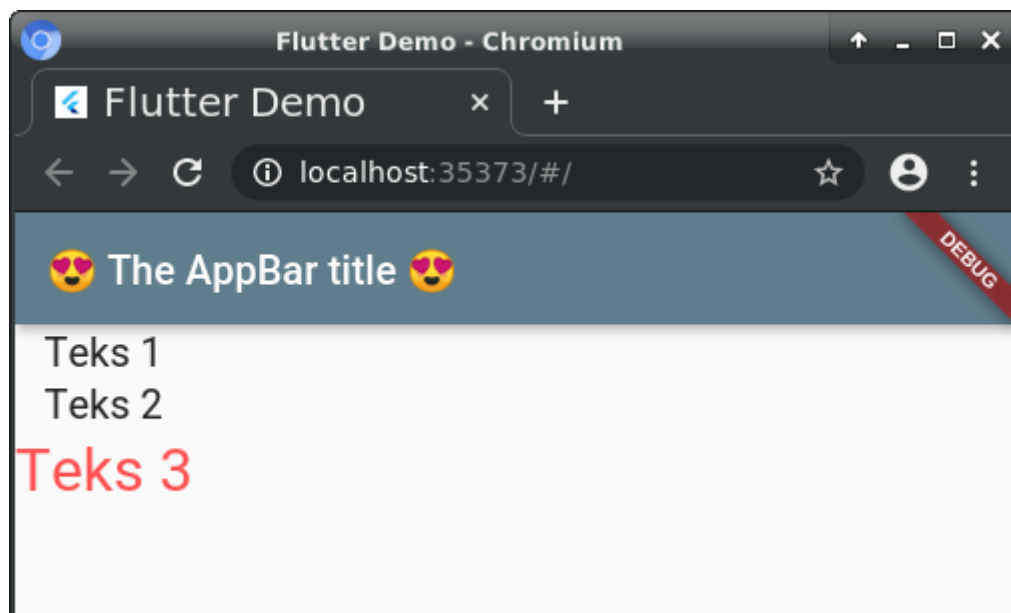
```
Column(
  children: [
    Text('Teks 1', style: TextStyle(fontSize: 20)),
    Text('Teks 2', style: TextStyle(fontSize: 20)),
    Text('Teks 3', style: TextStyle(fontSize: 30, color: Colors.redAccent),
  ]
)
```

Kode lengkap untuk file main.dart menjadi sebagai berikut.

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

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blueGrey,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      home: Scaffold(
        appBar: AppBar(title: Text(' The AppBar title ')),
        body: Column(children: [
          Text('Teks 1', style: TextStyle(fontSize: 20)),
          Text('Teks 2', style: TextStyle(fontSize: 20)),
          Text('Teks 3', style: TextStyle(fontSize: 30, color: Colors.redAccent)),
        ]),
      ),
    );
  }
}
```



Coba ganti Column menjadi Row, apa yang Anda amati?

4 Tugas

Buat program sederhana dengan Flutter yang menunjukkan penggunaan Text pada gabungan Column dan Row. Beberapa referensi:

- <https://api.flutter.dev/flutter/widgets/Text-class.html>

- <https://api.flutter.dev/flutter/widgets/Row-class.html>
- <https://api.flutter.dev/flutter/widgets/Column-class.html>