

Bab 6

Praktikum

Pemrograman Layout & Gestures



Flutter

Dosen : Bambang Sugiarto, ST, MT

**Program Studi S1 Teknik Informatika
Fakultas Teknik
Universitas Sangga Buana YPKP Bandung**

*Dirangkum dari berbagai sumber referensi
(hanya untuk penggunaan internal/tidak untuk dipublikasikan)*

Tujuan Praktikum

- Mahasiswa mengetahui pemrograman layout dan gestures pada Flutter.
- Mahasiswa mampu membuat pemrograman layout dan gestures pada Flutter.
- Mahasiswa mampu membuat implementasi pemrograman layout dan gestures dengan membuat aplikasi mobile sederhana pada Flutter dengan mencoba untuk mengatur layout dengan Row & Column widget.



Pendahuluan

- Sesuai dengan materi yang disampaikan pada kuliah teori tentang kosep dasar layout dan gesture pada flutter, maka pada praktikum kita kali ini akan mencoba untuk mengaplikasikan layout dan gesture pada pemrograman Flutter.
- Praktikum kita yang pertama adalah mengaplikasikan multiple child layout widget yang
 - Sedangkan pada praktikum yang kedua kita akan mencoba mengaplikasikan GestureDetector dengan berbagai macam listener, dimana pengguna akan memberikan gesture yang berbeda-beda.



Praktikum Multiple Child Layout Widges

<https://daengweb.id/belajar-flutter-basic-2-layout-menggunakan-widget/>

- Pada praktikum kali ini, kita akan mempraktekan pemrograman layout dengan menggunakan multiple child widget.
- Kita akan mencoba untuk mengatur layout dengan Row & Column widget.
- Row() & Column() dapat digunakan untuk mengatur posisi dari *widget* yang akan ditampilkan.
- **Row** sendiri berarti **baris**, maka apapun yang diapit oleh *class* ini akan menggunakan posisi baris atau berderet kesamping (Horizontal).
 - Sedang **Column** berarti kolom, maka *widget* yang berada didalamnya akan tersusun kebawah (Vertical).



- Untuk memahami penggunaan dari kedua class ini, kita akan menggunakan schema sebagai berikut :
 - Kolom pertama terdapat dua buah objek, yakni **Icon** dan **Text**. Dimana kedua objek ini berderet secara *horizontal*.
 - Kolom kedua terdapat sebuah **Card**, dimana didalam *card* memiliki dua buah objek yang berderet secara *vertical*
- Langkah-langkah praktikum :
 1. Buka file main.dart
 2. Modifikasi program pada main.dart tersebut menjadi :



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

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

class HomePage extends StatelessWidget {
  build(context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          backgroundColor: Colors.red[800],
          leading: Icon(Icons.home),
          title: Text('DW Flutter Basic')
        ),

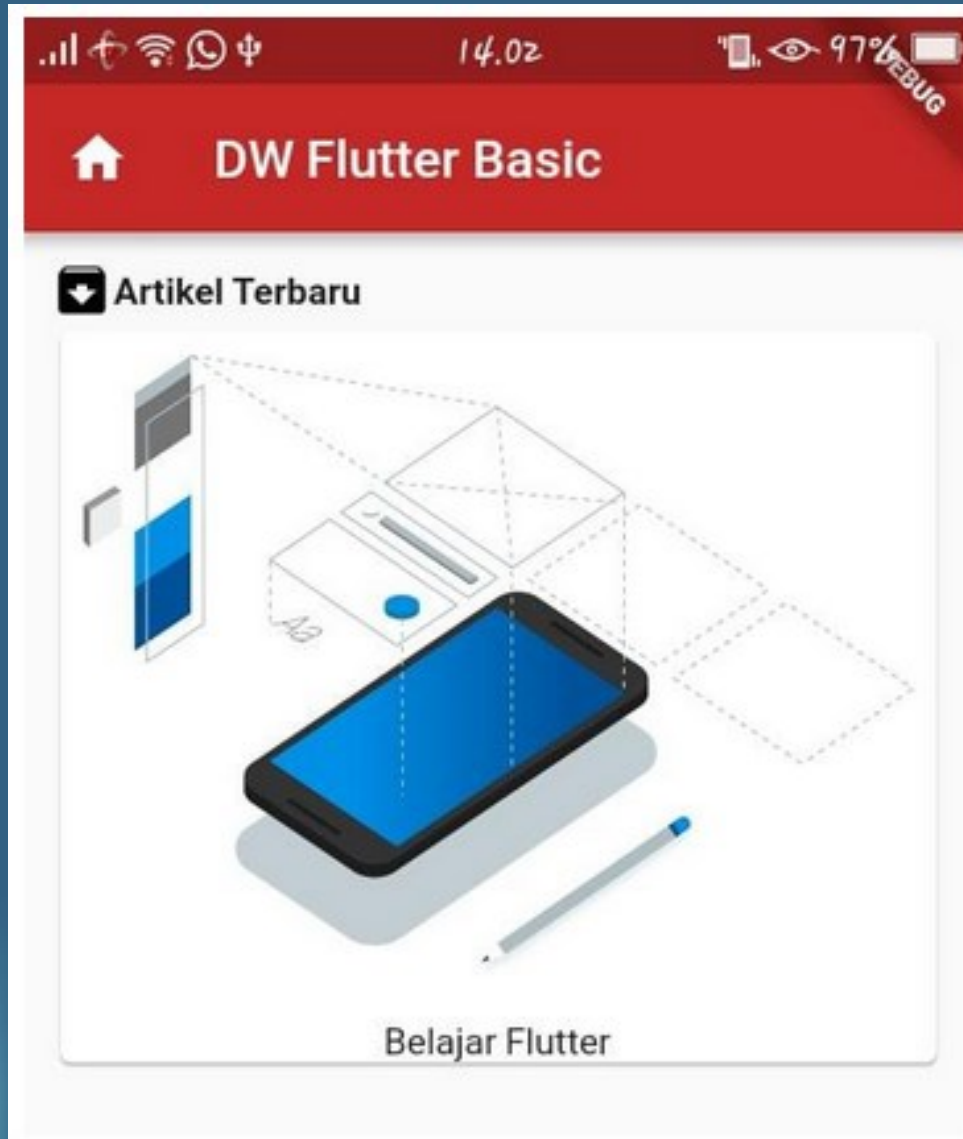
        //YANG DIMODIFIKASI
        body: Container(
          margin: EdgeInsets.all(10.0), //CODE BARU UNTUK MENGATUR MARGIN
```

```

child: Column(
  children: <Widget> [
    Row(
      children: <Widget>[
        Icon(Icons.archive),
        Text('Artikel Terbaru', style: new TextStyle(fontWeight: FontWeight.bold))
      ]
    ),
    Card(
      child: Column(
        children: <Widget>[
          Image.network('https://flutter.io/images/homepage/header-illustration.png'),
          Text('Belajar Flutter')
        ]
      ),
    ),
  ],
);
}

```

- Hasilnya adalah sebagai berikut :



Penjelasan program :

- Child dari Container terdapat *class* Column(), dimana children dari Column() terbagi menjadi dua bagian, yakni: Row() dan Card().
- Dapat disimpulkan, bahwa kedua objek tersebut akan membentuk kolom (vertical), sehingga apapun yang berada didalam Row() akan diletakkan diatas dan apapun yang berada didalam Card() akan diletakkan setelah objek dari Row di-render.
 - Row() juga memiliki *property* children, dimana didalamnya terdapat dua buah objek, yakni: Icon & Text. Maka Icon dan Text ini akan di-render secara horizontal.
 - Card() memiliki *property* child yang menggunakan *class* Column().
 - children dari *Column* terdapat dua buah objek yakni: Image dan Text.



Praktikum Pemrograman Gestures

<https://www.codingtive.com/2019/07/tutorial-flutter-menggunakan-gesture-detector-dan-penjelasan-nya.html/>

- Seperti sudah dijelaskan pada kuliah teori, GestureDetector merupakan widget yang digunakan untuk mendeteksi gesture
- Contohnya ketika pengguna tap, press, drag, dan lain-lain.
- GestureDetector sangat berguna ketika kita ingin membuat suatu widget yang tidak memiliki listener, contohnya jika kita ingin menambahkan listener klik pada widget seperti gambar, text atau icon.
 - Pada praktikum kali ini, kita akan menggunakan GestureDetector dengan berbagai macam listener, dimana pengguna akan memberikan gesture yang berbeda-beda pada widget yang menggunakan GestureDetector, kemudian hasilnya dapat ditampilkan pada title AppBar.



- Langkah-langkah praktikum :
 1. Buka file main.dart
 2. Modifikasi program pada main.dart tersebut menjadi :

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

void main() {
  runApp(MaterialApp(
    title: "Gesture Detector",
    debugShowCheckedModeBanner: false,
    home: MyApp(),
  ));
}

class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => _MyAppState();
}
```



```
class _MyAppState extends State<MyApp> {

  String _title;

  @override
  void initState() {
    super.initState();
    _title = "Gesture Detector";
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        title: Text(_title),
      ),
      body: ListView(
        children: <Widget>[

          ],
        ),
      );
  }
}
```



3. Kita sudah membuat kode dasarnya, kemudian buat fungsi di dalam class `_MyAppState` untuk mengubah state title seperti berikut :


```
void _changeTitle(String title) {  
    setState(() {  
        _title = title;  
    });  
}
```

4. Selanjutnya kita akan membuat fungsi untuk menampilkan Text di dalam Container yang nantinya akan kita tambahkan GestureDetector.

5. Sehingga kita akan membuat fungsi berikut di dalam class `_MyAppState` dengan kode program sebagai berikut :



```
Widget _buildItem(String title) {  
  return Container(  
    margin: EdgeInsets.all(8.0),  
    padding: EdgeInsets.all(8.0),  
    color: Colors.blue,  
    child: Center(  
      child: Text(  
        title,  
        style: TextStyle(  
          color: Colors.white,  
          fontWeight: FontWeight.bold,  
        ),  
      ),  
    ),  
  );  
}
```

- 
6. Sekarang saatnya menambahkan GestureDetector didalam children dari widget ListView sehingga kode programnya sebagai berikut :

```
body: ListView(  
  children: <Widget>[  
    GestureDetector(  
      onTap: () {  
        _changeTitle("onTap Clicked");  
      },  
      child: _buildItem("Click with tap"),  
    ),  
    GestureDetector(  
      onDoubleTap: () {  
        _changeTitle("onDoubleTap Clicked");  
      },  
      child: _buildItem("Click with double tap"),  
    ),  
    GestureDetector(  
      onLongPress: () {  
        _changeTitle("onLongPress Clicked");  
      },  
      child: _buildItem("Click with long press"),  
    ),  
  ],  
)
```



```
GestureDetector(  
  onVerticalDragDown: (dragDetails) {  
    _changeTitle("Vertical Drag Clicked");  
  },  
  child: _buildItem("Click with vertical drag"),  
) ,  
GestureDetector(  
  onHorizontalDragEnd: (dragDetails) {  
    _changeTitle("Horizontal Drag Clicked");  
  },  
  child: _buildItem("Click with horizontal drag"),  
) ,  
GestureDetector(  
  onScaleEnd: (scaleDetails) {  
    _changeTitle("Scale Clicked");  
  },  
  child: _buildItem("Click with scale left"),  
) ,  
],  
) ,
```



7. Silahkan untuk di run, maka kita sudah membuat widget dengan listener yang berbeda-beda seperti tap, double tap, long press.
8. Hasilnya dapat dilihat sebagai berikut :

