

## BAB 3. APLIKASI DENGAN STATEFULL WIDGET

### 3.1 Desain Aplikasi

Pada praktikum kali ini anda akan belajar mengenai cara pembuatan aplikasi sederhana yang memiliki statefull widget. Untuk pembuatan aplikasi ini dimulai dengan membuat sebuah desain yang dapat dimulai dengan membuat wireframe atau desain utuh di perangkat lunak seperti adobe xd atau sketch dan yang lain. Untuk menyederhanakan proses desain anda sudah disediakan template sederhana di starter code dengan menggunakan template material design. Berikut ini desain mockup aplikasi yang dibuat.



### 3.2 Konversi Desain Ke StatelessWidget

Untuk mempermudah proses konversi dari desain ke stateless widget perhatikanlah item item yang ada pada desain tersebut, apa saja widget yang terlibat dan apa saja widget layout yang digunakan.

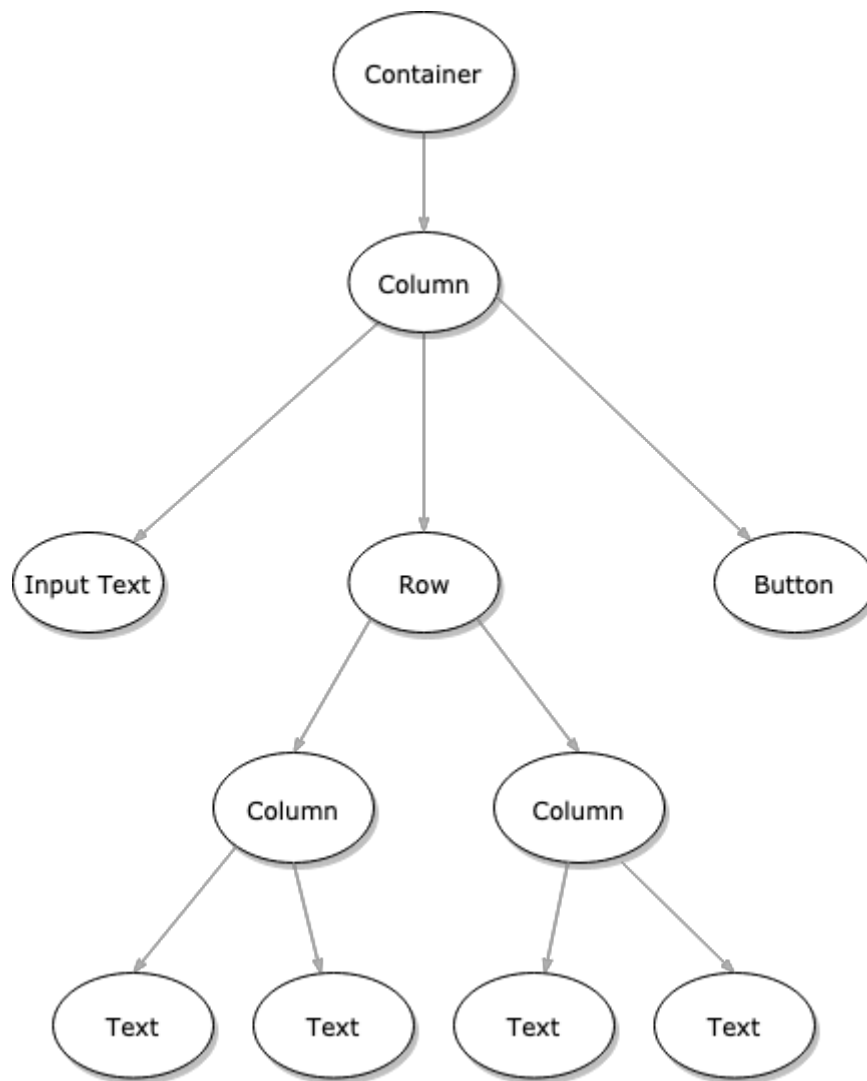
Sekilas dari UI desain yang ada dapat anda temukan bahwa terdapat satu input text, empat text, dan satu buah button. Secara lebih dalam perhatikan karakteristik item-item

tersebut seperti hint pada input, ukuran text yang berbeda, warna text pada tombol dan ukuran tombol yang memenuhi lebar layar.

Perhatikan juga bagaimana UI pada desain tersebut disusun dalam cara pandang row dan column, secara sederhana aplikasi ini terdiri dari satu kolom dan tiga baris dimana baris pertama berisi input, baris kedua berisi info suhu dan nilainya, serta baris ketiga berisi sebuah button. Khusus pada baris kedua terbagi menjadi dua buah kolom yang memiliki dua buah text.



Jika dikonversi menjadi sebuah tree layout diatas dapat direpresentasikan menjadi tree berikut ini.



### 3.2.1 Langkah Mengkonversi Desain Ke Widget

1. Buat Project flutter baru
2. Hapus starter code di main.dart sehingga menjadi seperti kode program dibawah ini

```
import 'package:flutter/material.dart';
import 'package:flutter/services.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.blue,
    visualDensity: VisualDensity.adaptivePlatformDensity,
  ),
  home: Scaffold(
    appBar: AppBar(
      title: Text("Konverter Suhu"),
    ),
    body: Container(),
  ),
);
}

```

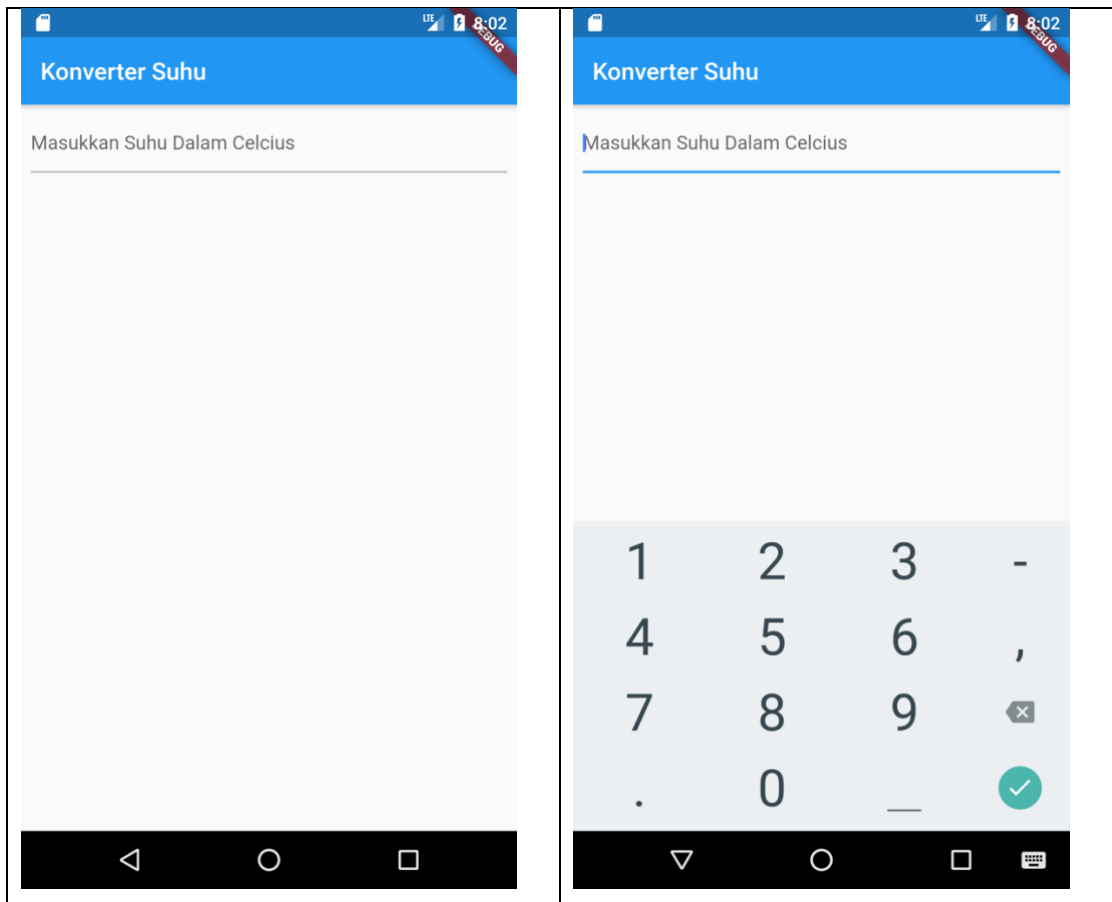
3. Mulai lah dengan menambahkan margin pada container dan memberikan child berupa sebuah InputForm pada container tersebut.

```

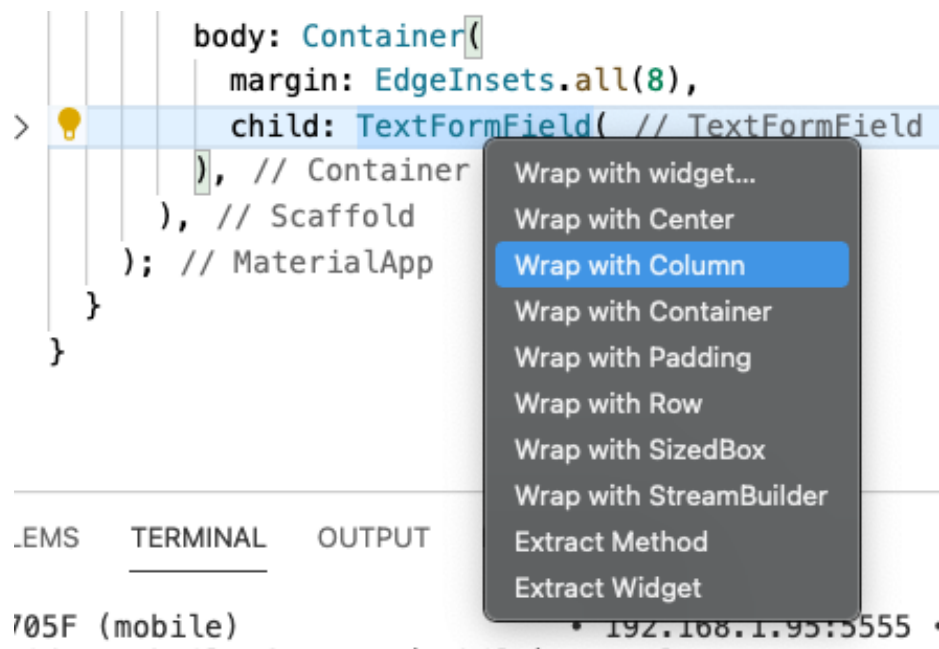
body: Container(
  margin: EdgeInsets.all(8),
  child: TextFormField(),
),

```

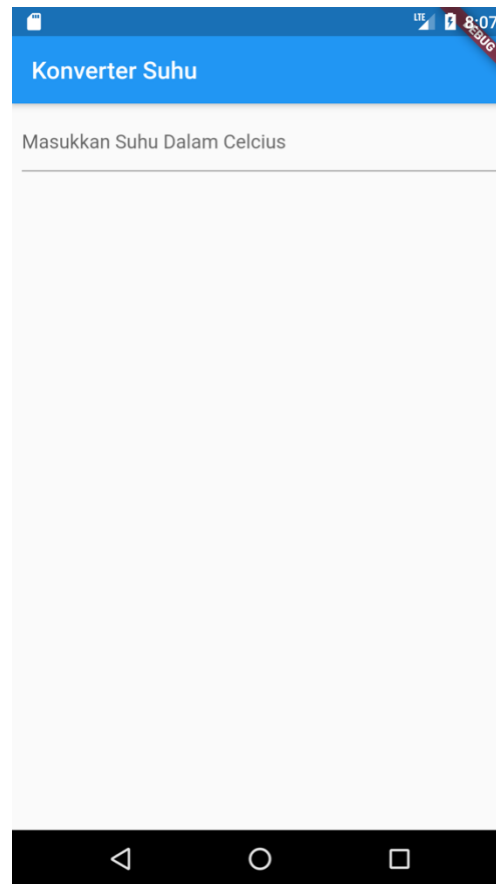
4. Mulailah dengan cara seperti ini walaupun TextFormField bukan merupakan parent widget pada widget tree yang kita butuhkan namun kita dapat melakukan wrapping dengan widget lain menggunakan shortcut Ctrl + .
5. Lanjutkan dengan melakukan modifikasi pada TextFormField carilah properties agar
  - a. Memiliki hint dengan text “Masukkan Suhu Dalam Celcius”
  - b. Memiliki validasi hanya angka.
  - c. Memiliki tampilan input keyboard khusus angka.
6. Setelah memasang properties tersebut berikut ini tampilan yang diharapkan



7. Wrap TextFormField kedalam sebuah kolom dengan menekan Ctrl + .



8. Berikut ini tampilan program setelah di wrap ke column



9. Langkah selanjutnya adalah mengisi Row untuk Text Info Suhu dalam Kelvin dan Reamur.
10. Langkah selanjutnya adalah mengisi sebuah button yang memiliki warna biru text putih dan even on click null.
11. Lakukan lah langkah 9 dan 10 secara mandiri sehingga layout yang dibentuk menjadi seperti pada gambar desain. Berikut ini resource yang dapat anda gunakan untuk menyelesaikan langkah 9 dan 10.
  - a. <https://flutter.dev/docs/cookbook/forms/text-input>
  - b. <https://stackoverflow.com/questions/50014342/button-width-match-parent>
  - c. <https://medium.com/flutter-community/beginners-guide-to-text-styling-in-flutter-3939085d6607>
  - d. <https://medium.com/@yuvrajpandey24/working-with-raised-button-in-flutter-6f5c0f71aab3>
  - e. <https://daengweb.id/mengenal-widjet-flutter-2-container-stack-positioned>
  - f. <https://medium.com/jlouage/container-de5b0d3ad184>
  - g. <https://stackoverflow.com/questions/53850149/flutter-crossaxisalignment-vs-mainaxisalignment>

### 3.3 Membuat Aplikasi Interaktif

Setelah berhasil membuat layout selanjutnya adalah membuat aplikasi yang mampu melakukan konversi suhu dari input berupa derajat celcius ke tiga derajat lain yaitu kelvin, reamur dan fahrenheit.

#### 3.3.1 Convert Ke Statefull Widget

Langkah pertama adalah dengan mengkonversi widget ke StatefullWidget. Hal ini dapat dilakukan dengan menekan tombol Ctrl + . pada widget yang ingin di konversi. Berikut ini kode program sebelum dikonversi.

```
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.blue,  
        visualDensity: VisualDensity.adaptivePlatformDensity,  
      ),  
    );  
  }  
}
```

Setelah dilakukan konversi kode program berubah memiliki state seperti dibawah ini :

```
class MyApp extends StatefulWidget {  
  @override  
  _MyAppState createState() => _MyAppState();  
}  
  
class _MyAppState extends State<MyApp> {  
  // text controller
```

#### 3.3.2 Membuat State

Untuk membuat state perhatikan beberapa best practice berikut ini mengenai elemen mana dari UI yang akan dimasukkan kedalam state.

1. Apakah ada elemen pada UI yang berubah ?
2. Apakah yang menyebabkan elemen tersebut berubah ?

Jika diperhatikan dari UI yang disediakan maka elemen yang dapat berubah adalah input dari user, dan output dari proses konversi suhu. Sedangkan yang menjadi trigger perubahan adalah ketika tombol konversi ditekan. Pada pengembangan aplikasi menggunakan flutter

elemen yang berubah ini akan dimasukkan kedalam state dan elemen yang mentrigger perubahan akan disebut event.

Untuk membuat state tambahkanlah kode program yang menurut anda perlu ditambahkan berikut ini contoh kode program pada state dari widget MyApp yang sekarang berubah menjadi statefull widget.

```
class _MyAppState extends State<MyApp> {  
  //variabel berubah  
  double _inputUser = 0;  
  double _kelvin = 0;  
  //tambahkan variabel lain yang dibutuhkan
```

Berikut ini beberapa url yang dapat membantu proses pembuatan variabel state

1. <https://www.w3adda.com/dart-tutorial/dart-variables>

### 3.3.3 Mengambil Data Input Suhu

Selain memiliki variabel input tentu saja input ini harus diambil dari TextFormField yang menadi cara user menginputkan angka untuk di konversi, carilah cara menambahkan controller terhadap TextFormField ini. Berikut ini beberapa url yang dapat membantu proses tersebut.

1. <https://flutter.dev/docs/cookbook/forms/retrieve-input>
2. <https://stackoverflow.com/questions/61538657/how-to-get-value-from-textformfield-on-flutter>
3. <https://medium.com/flutter-developer-indonesia/retrieve-value-from-text-field-to-widget-text-e12a9f162307>

### 3.3.4 Membuat Event

Langkah selanjutnya adalah membuat event dimana event perhitungan konversi dan update hasil konversi dilakukan saat tombol Konversi ditekan. Untuk menyambungkan proses klik button ke perubahan state ada beberapa langkah yang harus anda lakukan.

1. Buat Fungsi di dalam state MyAppState
2. Panggil Fungsi ini ketika tombol konversi di tekan
3. Tambahkan Set State di fungsi tersebut dan ubah lah state dari variabel yang anda gunakan sebagai dasar perhitungan konversi.
4. Tambahkan kode program untuk melakukan konversi suhu dan set state variabel yang anda gunakan untuk kelvin, reamur dan fahrenheit.



Berikut ini url yang dapat anda gunakan untuk membantu proses ini.

1. <https://rumusrumus.com/rumus-konversi-suhu/>
2. <https://stackoverflow.com/questions/50758704/how-to-convert-to-double>
3. Project Hello World Flutter cek bagaimana function increment di panggil dan bagaimana state di set.

### **3.3.5 Menampilkan Output Result**

Untuk menampilkan output anda hanya perlu menyambungkan variabel yang berubah pada state ke widget Text yang menampilkan hasil. Berikut ini beberapa url yang dapat membantu anda menyelesaikan proses ini.

1. <https://stackoverflow.com/questions/28419255/how-do-you-round-a-double-in-dart-to-a-given-degree-of-precision-after-the-decim>
2. Project Hello World Flutter cek bagaimana variabel counter dihubungkan ke Text Widget.

## **3.4 Membagi Ke Widget Yang Lebih Kecil**

Selamat jika anda sampai pada langkah ini berarti aplikasi yang anda buat sudah dapat melakukan konversi dan selesai secara fitur, namun untuk proses development yang baik widget yang ada di main.dart harus dipotong potong menjadi widget yang lebih sederhana sehingga mudah untuk di maintain dan di perbaiki.

Lakukan lah ekstraksi widget ke file yang berbeda dengan cara Ctrl + . pada widget yang ingin anda potong dan pilih menu Extract Widget, berikanlah nama class yang sesuai dan pindahkan ke file baru.

Berikut ini nama nama widget dan ekstraksi yang dapat dicontoh :

1. Input : Berisi Widget TextFormField dan controller nya
2. Result : Berisi widget hasil konversi yang memiliki variabel nama dan hasil.
3. Convert : Berisi button dan reference ke function yang digunakan untuk melakukan setState().

Berikut ini resource yang dapat anda pelajari untuk melakukan konversi widget :

1. <https://medium.com/flutter-community/flutter-visual-studio-code-shortcuts-for-fast-and-efficient-development-7235bc6c3b7d>

### **3.4.1 Send Parameter ke Child**

Sebuah widget dapat mengirim paramter ke child widgetnya melalui constructor yang dimiliki oleh widget tersebut parameter ini dapat menyesuaikan dengan kebutuhan yang

dimiliki oleh widget tersebut. Berikut ini contoh pembuatan widget dan constructor dari widget yang dibuat.

```
Input(etInput: etInput),
```

```
class Input extends StatelessWidget {  
  const Input({  
    Key key,  
    @required this.etInput,  
  }) : super(key: key);  
  
  final TextEditingController etInput;  
  
  @override  
  Widget build(BuildContext context) {  
    return TextFormField(  
      decoration: InputDecoration(hintText: "Masukkan Suhu Dalam Celcius"),  
      inputFormatters: [FilteringTextInputFormatter.digitsOnly],  
      controller: etInput,  
      keyboardType: TextInputType.number,  
    );  
  }  
}
```

### 3.4.2 Send Event Handler Ke Child

Selain mengirimkan parameter ke child widget parent juga dapat mengirim fungsi turun ke child widget dimana fungsi ini akan dipanggil lagi oleh parent widget yang memiliki state. Berikut ini contoh kode program untuk menurunkan fungsi ke child widget.

```
Convert(konvertHandler: _konversiSuhu),
```

```
class Convert extends StatelessWidget {  
  final Function konvertHandler;  
  
  Convert({Key key, @required this.konvertHandler});  
  
  @override  
  Widget build(BuildContext context) {  
    return Container(  

```

```
        width: double.infinity,  
        height: 50,  
        child: RaisedButton(  
          onPressed: konvertHandler,  
          color: Colors.blueAccent,  
          textColor: Colors.white,  
          child: Text("Konversi Suhu"),  
        ),  
      );  
    }  
  }
```

### 3.5 Praktikum 1

1. Selesaikan Langkah Praktikum Di Atas dan Submit ke Repository Github