

PRAKTIKUM MCS BAB 1

Widget-widget dasar pada Flutter

PENDAHULUAN

Pada praktikum MCS bab 1, praktikan akan diajarkan mengenai *widget-widget* dasar yang ada pada *flutter* yang digunakan untuk membentuk suatu halaman aplikasi. Selain itu, praktikan juga akan langsung mengimplementasikan *widget-widget* tersebut dalam membentuk sebuah aplikasi.

- **MaterialApp()**

`MaterialApp()` merupakan *widget* paling dasar yang berfungsi sebagai fondasi yang mengatur konfigurasi umum dari aplikasi. Untuk menggunakan *widget* ini, pengguna harus melakukan *import* terhadap '**package:flutter/material.dart**'. Berikut merupakan struktur dari *widget* `MaterialApp()`:

```
MaterialApp(  
  title: ...,  
  debugShowCheckedModeBanner: ...,  
  theme: ...,  
  home: ....  
  initialRoute: ...,  
  routes: {...}  
  //dan lain lain  
);
```

- **Scaffold()**

`Scaffold()` merupakan *widget* umum yang digunakan untuk membentuk halaman pada sebuah aplikasi. *Widget* ini memiliki berbagai macam properti, seperti **appBar:**, **body:**, **floatingActionButton:**, **drawer:**, **bottomNavigationBar:** dan lain-lain. Berikut merupakan struktur dari *widget* `scaffold()`:

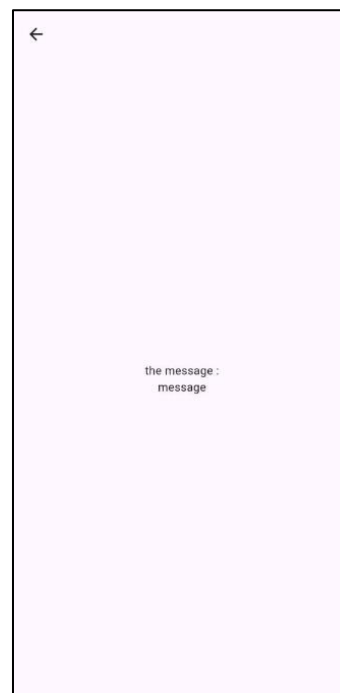
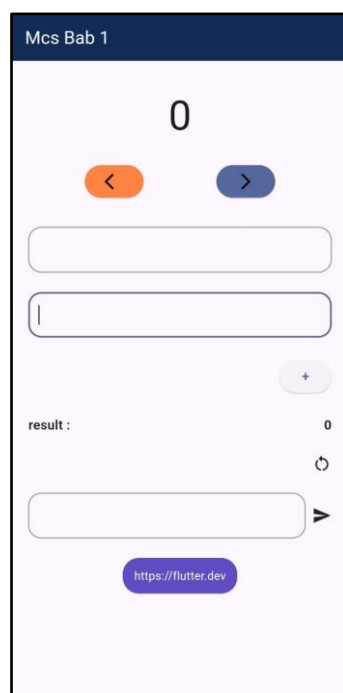
```
Scaffold(  
  appBar: ...,  
  body: ...,  
  floatingActionButton: ...,  
  drawer: ...,  
  bottomNavigationBar: ...,  
)
```

Untuk menggunakan berbagai jenis *widget* yang tersedia pada flutter, kita wajib meng*extends* *class* yang akan membentuk aplikasi dengan memilih salah satu diantara 2 *widget*, yaitu **StatelessWidget()** atau **StatefulWidget()**. Berikut beberapa perbedaan antara **StatelessWidget()** dengan **StatefulWidget()**:

- **StatelessWidget():**
 - Hanya fokus pada tampilan
 - Tidak dapat menangani perubahan tampilan. Misal terdapat angka 1 pada halaman kemudian ingin dilakukakn perubahan tertentu dengan aktivitas tertentu pada angka tersebut, dengan StatelessWdget hal tersebut tidak bisa dilakukan
 - Dibuat dengan extends dari StatelessWidget
 - Biasanya hanya digunakan untuk mengembalikan MaterialApp
- **StatefulWidget():**
 - Dapat menangani perubahan tampilan, Misal terdapat angka 1 pada halaman kemudian ingin dilakukakn perubahan tertentu dengan aktivitas tertentu pada angka tersebut, dengan StatefulWidget hal tersebut bisa dilakukan
 - Dibuat dengan extends dari StatefulWidget
 - Menggunakan 2 class(widget dan state)

PRAKTIKUM BAB 1

Tampilan aplikasi yang akan dibangun



Penjelasan cara kerja aplikasi akan diterangkan oleh Penanggung Jawab (PJ).

Buatlah *project flutter* baru pada android studio dan pilihlah tempat untuk menyimpan *project* tersebut. Setelah *project* selesai terbentuk, bukalah folder **lib** yang tersedia dan buatlah file bernama **main.dart**. Adapun isi dari file main.dart adalah sebagai berikut :

```
import 'package:flutter/material.dart';
import 'package:mcs_bab_1/home_page.dart';

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

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const HomePage(),
    );
  }
}
```

Ketika membangun sebuah aplikasi menggunakan *flutter* fungsi **main()** menjadi kode yang paling utama agar aplikasi dapat berjalan. Fungsi tersebut akan memanggil sebuah *class* yang akan mengextends *stateles widget* dan mengembalikan **MaterialApp()** yang berisikan beberapa argumen, seperti **title**: yang digunakan untuk memberikan label dari sebuah aplikasi, **debugShowCheckedModeBanner**: yang *valuenya* diset menjadi *false* untuk menghilangkan label *debug* yang berada pada bagian pojok kanan atas, **theme**: yang digunakan untuk mengatur tema aplikasi dan **home**: yang digunakan untuk mengatur halaman mana yang akan pertama kali ditampilkan pada saat aplikasi dijalankan. Untuk menggunakan *widget* tersebut, pengguna harus melakukan *import* terhadap **'package:flutter/material.dart'**

Setelah mempersiapkan fondasi awal aplikasi, langkah berikutnya adalah membuat sebuah file bernama **home_page.dart** yang berisikan beberapa baris kode yang digunakan untuk membangun halaman aplikasi. Adapun isi dari home_page.dart adalah sebagai berikut :

```

import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:mcs_bab_1/receiver.dart';
import 'package:url_launcher/url_launcher.dart';

class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  int start = 0;
  TextEditingController input1Controller = TextEditingController();
  TextEditingController input2Controller = TextEditingController();
  TextEditingController messageController = TextEditingController();
  int result = 0;
  String flutterUrl = "https://flutter.dev";

  void increment() {
    setState(() {
      start++;
    });
  }

  void decrement() {
    setState(() {
      start--;
      if(start <= 0){
        start = 0;
      }
    });
  }

  void sumOf({required int x, required int y}){
    setState(() {
      result = x + y;
    });
  }

  void resetResult() {
    setState(() {
      input1Controller.text = "";
      input2Controller.text = "";
      result = 0;
    });
  }

  @override
  void dispose() {
    input1Controller.dispose();
    input2Controller.dispose();
    messageController.dispose();
    // TODO: implement dispose
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(

```

```

    appBar: AppBar(
      title: const Text("Mcs Bab 1", style: TextStyle(color:
Colors.white)),
      backgroundColor: const Color(0xff102C57),
    ),
    body: ListView(
      children: [
        const SizedBox(height: 32,),

        Center(
          child: Text(start.toString(), style: const TextStyle(fontSize:
50)),
        ),

        const SizedBox(height: 20,),

        Row(
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
          children: [
            ElevatedButton(
              onPressed: () {decrement();},
              style: ElevatedButton.styleFrom(
                backgroundColor: const Color(0xffffF8343),
              ),
              child: const Icon(Icons.arrow_back_ios, color:
Colors.black),
            ),

            ElevatedButton(
              onPressed: () {increment();},
              style: ElevatedButton.styleFrom(
                backgroundColor: const Color(0xff55679C),
              ),
              child: const Icon(Icons.arrow_forward_ios, color:
Colors.black),
            ),
          ],
        ),

        const SizedBox(height: 32,),

        //input1
        Container(
          margin: const EdgeInsets.symmetric(horizontal: 20),
          child: TextFormField(
            controller: input1Controller,
            keyboardType: TextInputType.number,
            inputFormatters: <TextInputFormatter>[
              FilteringTextInputFormatter.allow(RegExp(r'[0-9]'))
            ],
            decoration: InputDecoration(
              border: OutlineInputBorder(
                borderRadius: BorderRadius.circular(16),
              ),
            ),
          ),
        ),

        const SizedBox(height: 24,),

```

```

//input2
Container(
  margin: const EdgeInsets.symmetric(horizontal: 20),
  child: TextFormField(
    controller: input2Controller,
    keyboardType: TextInputType.number,
    inputFormatters: <TextInputFormatter>[
      FilteringTextInputFormatter.allow(RegExp(r'[0-9]'))
    ],
    decoration: InputDecoration(
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(16),
      ),
    ),
  ),
),

const SizedBox(height: 24,),

//sumButton
Container(
  margin: const EdgeInsets.symmetric(horizontal: 20),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.end,
    children: [
      ElevatedButton(
        onPressed: () {
          sumOf(
            x: int.parse(input1Controller.text),
            y: int.parse(input2Controller.text),
          );
        },
        child: const Text(
          "+",
          style: TextStyle(
            fontSize: 16,
            fontWeight: FontWeight.w600,
          ),
        ),
      ),
    ],
  ),
),

const SizedBox(height: 24,),

Container(
  margin: const EdgeInsets.symmetric(horizontal: 20),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      const Text(
        "result :",
        style: TextStyle(
          fontSize: 16,
          fontWeight: FontWeight.w600,
        ),
      ),
      Text(
        "$result",

```

```

        style: const TextStyle(
          fontSize: 16,
          fontWeight: FontWeight.w700,
        ),
      ),
    ],
  ),
),

const SizedBox(height: 24,),

//reset result
Container(
  margin: const EdgeInsets.symmetric(horizontal: 20),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.end,
    children: [
      GestureDetector(
        child: const Icon(Icons.restart_alt),
        onTap: () {resetResult();},
      )
    ],
  ),
),

const SizedBox(height: 24,),

//send message
Container(
  margin: const EdgeInsets.symmetric(horizontal: 20),
  child: Row(
    children: [
      Expanded(
        child: TextFormField(
          controller: messageController,
          decoration: InputDecoration(
            border: OutlineInputBorder(
              borderRadius: BorderRadius.circular(16),
            ),
          ),
        ),
      ),
    ],
  ),

  const SizedBox(width: 8,),

  GestureDetector(
    child: const Icon(Icons.send),
    onTap: () {
      Navigator.push(
        context,
        MaterialPageRoute(builder: (context) =>
Receiver(message: messageController.text),),
      );
    },
  ),
),

const SizedBox(height: 24,),

```

```

        Center(
          child: GestureDetector(
            child: Container(
              padding: const EdgeInsets.symmetric(vertical: 12,
horizontal: 14),
              decoration: BoxDecoration(
                borderRadius: BorderRadius.circular(20),
                color: const Color(0xff604CC3)
              ),
              child: Text(flutterUrl, style: const TextStyle(color:
Colors.white)),
            ),
            onTap: () async{
              await launchUrl(Uri.parse(flutterUrl));
            },
          ),
        ),
      ],
    ),
  );
}
}

```

*class **HomePage*** merupakan *class* yang akan dipanggil pada argumen *home*: yang ada pada *widget* `MaterialApp()` nantinya akan dipanggil untuk argumen *home*: yang ada pada `MaterialApp()`. Terdapat sedikit perbedaan saat membuat *class* `MyApp()` dan *class* `HomePage()`, yang mana *class* `MyApp()` *mengextends* `StatelessWidget` dan *class* `HomePage()` *mengextends* `StatefulWidget`. Karena peran dari `MaterialApp()` hanya digunakan untuk mendefinisikan hal umum dari aplikasi, maka *class* `MyApp` cukup *mengextends* `StatelessWidget` saja.

```

int start = 0;
TextEditingController input1Controller = TextEditingController();
TextEditingController input2Controller = TextEditingController();
TextEditingController messageController = TextEditingController();
int result = 0;
String flutterUrl = "https://flutter.dev";

```

variabel ini nantinya akan dibutuhkan saat membangun halaman

```

void increment() {
  setState(() {
    start++;
  });
}
void decrement() {
  setState(() {
    start--;
    if(start <= 0) {
      start = 0;
    }
  });
}

```


Fungsi ***decrement()*** berisikan beberapa baris kode yang berguna untuk mengurangi nilai dari variabel *start* dan terdapat pengkondisian pada fungsi tersebut yang mana ketika nilai dari *start* kurang dari 0, maka aplikasi hanya akan menampilkan angka 0 ke layar. Sedangkan fungsi ***increment()*** digunakan untuk menambahkan nilai dari variabel *start*.

```
void sumOf({required int x, required int y}){
  setState(() {
    result = x + y;
  });
}

void resetResult(){
  setState(() {
    input1Controller.text = "";
    input2Controller.text = "";
    result = 0;
  });
}
```

Fungsi dari ***sumOf()*** nantinya digunakan untuk menjumlahkan *input1Controller* dan *input2Controller* dan hasilnya akan disimpan ke dalam variabel *result*. Sedangkan fungsi ***resetResult()*** berisikan baris program yang berfungsi untuk mengosongkan kembali nilai dari variabel *input1Controller*, *input2Controller* dan *result*.

```
@override
void dispose() {
  input1Controller.dispose();
  input2Controller.dispose();
  messageController.dispose();
  // TODO: implement dispose
  super.dispose();
}
```

Fungsi ***dispose()*** berguna untuk menghindari kebocoran memori pada aplikasi. Adapun ***setState({})*** digunakan untuk mengubah tampilan aplikasi apabila terdapat perubahan yang mengharuskan tampilan pada aplikasi berubah.

```
return Scaffold(
  appBar: ,
  body: ,
);
```

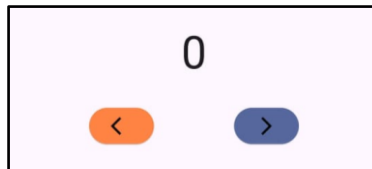
Widget **scaffold** digunakan untuk membentuk halaman aplikasi. Widget ini tidak hanya memiliki properti **appBar:** dan **body:** saja. Namun, terdapat beberapa properti lain yang dapat digunakan, seperti **backgroundColor:**, **floatingActionButton:**, **drawer:** dan lain-lain. Pada bab ini, properti Scaffold yang digunakan hanyalah properti *appbar* dan *body* saja. Isi dari

properti `body`: merupakan *widget* `ListView` yang memungkinkan pengguna untuk melakukan *scroll* terhadap aplikasi yang berisikan *widget-widget*.



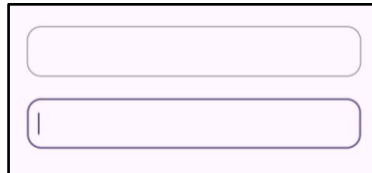
```
appBar: AppBar(  
  title: const Text("Mcs Bab 1", style: TextStyle(color: Colors.white),),  
  backgroundColor: const Color(0xff102C57),  
),
```

Aplikasi pada bab ini memiliki sebuah `appBar` yang bertuliskan “Mcs Bab 1” dan memiliki warna dari `appBar` tersebut adalah `0xff102C57`. Warna dengan *code hexa* dapat dicari di berbagai website salah satunya <https://colorhunt.co>.



```
body: ListView(  
  children: [  
    const SizedBox(height: 32,),  
    Center(  
      child: Text(start.toString(), style: const TextStyle(fontSize: 50),),  
    ),  
    const SizedBox(height: 20,),  
    Row(  
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
      children: [  
        ElevatedButton(  
          onPressed: () {decrement();},  
          style: ElevatedButton.styleFrom(  
            backgroundColor: const Color(0xffffF8343),  
          ),  
          child: const Icon(Icons.arrow_back_ios, color: Colors.black,),  
        ),  
        ElevatedButton(  
          onPressed: () {increment();},  
          style: ElevatedButton.styleFrom(  
            backgroundColor: const Color(0xff55679C),  
          ),  
          child: const Icon(Icons.arrow_forward_ios, color: Colors.black,),  
        ),  
      ],  
    ),  
    const SizedBox(height: 32,),  
    //...  
  ],  
),
```

*Widget **Center()*** digunakan untuk membuat *widget* turunannya berada di tengah. *Widget* yang ada di dalam *Center* adalah *widget **Text()*** yang diambil *valuenya* dari variabel *start*. Variabel *start* memiliki nilai awal yang telah didefinisikan sebesar 0, maka yang akan ditampilkan di halaman aplikasi adalah nilai 0. *Widget **Row()*** dapat memiliki turunan yang banyak dan mengarah secara horizontal. Pada bab ini, penggunaan *widget **Row()*** akan membungkus 2 *widget* lainnya bernama ***ElevatedButton()***. *Widget* tersebut nantinya akan digunakan untuk mengubah angka yang tertera pada bagian atas. Untuk mengurangi angka tersebut, pada *ElevatedButton* pertama diberikan *Icon* panah ke kiri dengan warna menggunakan *code hexa* 0xffFF8343 dan pada bagian *onPressed* diberikan fungsi *decrement()*. Sedangkan untuk *ElevatedButton* kedua digunakan untuk menambahkan angka dengan *Icon* panah ke kanan dengan warna menggunakan *code hexa* 0xff55679C dan pemanggilan fungsi *increment()* pada properti *onPressed*.



```
body: ListView(
  children: [
    //...

    const SizedBox(height: 32,),

    //input1
    Container(
      margin: const EdgeInsets.symmetric(horizontal: 20),
      child: TextFormField(
        controller: input1Controller,
        keyboardType: TextInputType.number,
        inputFormatters: <TextInputFormatter>[
          FilteringTextInputFormatter.allow(RegExp(r'[0-9]'))
        ],
        decoration: InputDecoration(
          border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(16),
          ),
        ),
      ),
    ),

    const SizedBox(height: 24,),

    //input2
    Container(
      margin: const EdgeInsets.symmetric(horizontal: 20),
      child: TextFormField(
        controller: input2Controller,
```

```

        keyboardType: TextInputType.number,
        inputFormatters: <TextInputFormatter>[
          FilteringTextInputFormatter.allow(RegExp(r'[0-9]'))
        ],
        decoration: InputDecoration(
          border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(16),
          ),
        ),
      ),
    ),
  ),
  //...
],
),

```

Kemudian di bawah 2 *ElevatedButton*, terdapat 2 widget ***TextFormField()*** yang berfungsi untuk menerima nilai *input* angka yang dimasukkan oleh pengguna lalu dijumlahkan. Kedua *widget* tersebut nantinya akan memiliki sebuah border dengan radius setiap sudutnya sebesar 16 dan diatur agar pengguna hanya dapat memasukkan nilai berupa angka saja. Pada masing-masing *TextFormField()* terdapat properti *controller* yang digunakan sebagai penanda agar ketika pengguna memasukkan nilai pada 1 *controller*, inputan tersebut hanya masuk ke *controller* tersebut. Pada *TextFormField()* pertama, nilai pada properti *controller* diambil dari variabel ***input1Controller*** dan pada *TextFormField()* kedua, nilai dari properti *controller* diambil dari variabel ***input2Controller***.



```

body: ListView(
  children: [
    //...

    //sumButton
    Container(
      margin: const EdgeInsets.symmetric(horizontal: 20),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.end,
        children: [
          ElevatedButton(
            onPressed: () {
              sumOf(
                x: int.parse(input1Controller.text),
                y: int.parse(input2Controller.text),
              );
            },
          ),
        ],
      ),
    ),
  ],
)

```

```

        child: const Text(
          "+",
          style: TextStyle(
            fontSize: 16,
            fontWeight: FontWeight.w600,
          ),
        ),
      ],
    ),
  ),
),
),

const SizedBox(height: 24,),

Container(
  margin: const EdgeInsets.symmetric(horizontal: 20),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      const Text(
        "result :",
        style: TextStyle(
          fontSize: 16,
          fontWeight: FontWeight.w600,
        ),
      ),

      Text(
        "$result",
        style: const TextStyle(
          fontSize: 16,
          fontWeight: FontWeight.w700,
        ),
      ),
    ],
  ),
),

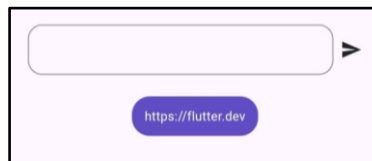
const SizedBox(height: 24,),

//reset result
Container(
  margin: const EdgeInsets.symmetric(horizontal: 20),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.end,
    children: [
      GestureDetector(
        child: const Icon(Icons.restart_alt),
        onTap: () {resetResult();},
      ),
    ],
  ),
),

//...
],
),

```

Fungsi `sumOf()` akan bekerja ketika pengguna menekan *button* dengan *icon* “+” yang terletak di bawah 2 `TextFormField()`. Fungsi `sumOf()` berfungsi untuk menjumlahkan angka hasil *input* yang tersimpan ke dalam `input1Controller` dan `input2Controller`. Fungsi `sumOf()` memiliki 2 konstruktor, yakni argumen `x` dan `y` yang bertipe data *integer* sehingga `input1Controller` dan `input2Controller` harus dikonversikan terlebih dahulu ke dalam bentuk *integer*. Hasil penjumlahan akan ditampilkan di dalam variabel `result` dan *widget* `Text()` yang berisikan variabel `result` akan menampilkan hasil penjumlahannya. Di bawahnya terdapat *icon* bernama `restart_alt` yang berbentuk melingkar yang dibungkus oleh **`GestureDetector()`**. *Widget* ini berguna untuk membuat setiap *widget* yang diturunkannya dapat memiliki sebuah aksi ketika pengguna berinteraksi. *Icon* tersebut berfungsi untuk mengosongkan kembali `input1Controller`, `input2Controller` dan `result`.



```
body: ListView(
  children: [
    //...

    //send message
    Container(
      margin: const EdgeInsets.symmetric(horizontal: 20),
      child: Row(
        children: [
          Expanded(
            child: TextFormField(
              controller: messageController,
              decoration: InputDecoration(
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(16),
                ),
              ),
            ),
          ),
        ],
      ),

      const SizedBox(width: 8),

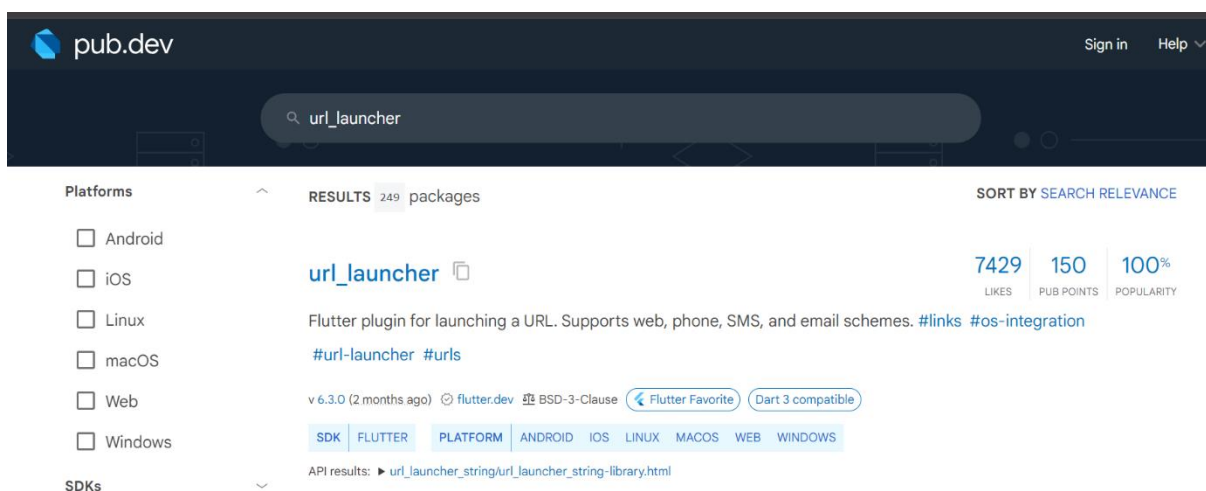
      GestureDetector(
        child: const Icon(Icons.send),
        onTap: () {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => Receiver(message:
messageController.text)),
          );
        },
      ),
    ],
  ),
),
```




Terdapat konstruktor pada `class Receiver()` yang memiliki parameter yang diambil dari variabel `message`. Hal tersebut digunakan untuk menampung nilai *input* yang dimasukkan di `class` sebelumnya. Hasil dari *inputan* tersebut nantinya akan ditampilkan pada bagian tengah halaman aplikasi dengan menggunakan `widget Center()`.

Setelah membuat file `receiver.dart`, kita dapat melanjutkan kembali pengkodean pada file `home_page.dart` dimana terdapat `TextFormField()` dan icon berbentuk *send* dalam `1 Row()`. `Widget TextFormField` tersebut juga memiliki sebuah *controller* dengan nilai **`messageController`** untuk menampung nilai *input* dan *icon send* yang dibungkus dengan `widget GestureDetector()` agar *icon* tersebut dapat memiliki sebuah aksi. Aksi yang diberikan pada *icon* tersebut akan terjadi ketika pengguna melakukan *tap* pada *icon* tersebut dan nantinya aplikasi akan berpindah ke suatu halaman dan menampilkan pesan yang telah di*input* ke dalam `TextFormField` tersebut.

Terakhir, kita akan membuat sebuah *button* yang di dalamnya terdapat sebuah url <https://flutter.dev> yang ketika kita menekan *button* tersebut, maka kita akan menuju ke halaman dari url tersebut. Untuk membuat agar aplikasi kita dapat membuka url tersebut, perlu dilakukan penambahan terhadap sebuah *package* bernama **`url_launcher`** ke dalam file **`pubspec.yaml`** pada bagian **`dependencies`**. Kita dapat mencari berbagai *package* yang dapat digunakan pada *flutter* dengan mengunjungi website <https://pub.dev/>.




```
30 dependencies:
31   flutter:
32     sdk: flutter
33
34
35   # The following adds the Cupertino Icons
36   # Use with the CupertinoIcons class for
37   cupertino_icons: ^1.0.8
38   url_launcher: ^6.3.0
```

LAPORAN AWAL (LP)

1. Jelaskan apa itu Android!
2. Jelaskan apa itu Flutter!
3. Jelaskan apa itu MaterialApp!
4. Berikan keterangan perbedaan StatelessWidget dan StatefulWidget

LAPORAN AKHIR (LA)

1. Berikan kesimpulan pada Bab 1!