

PRAKTIKUM MCS BAB 1

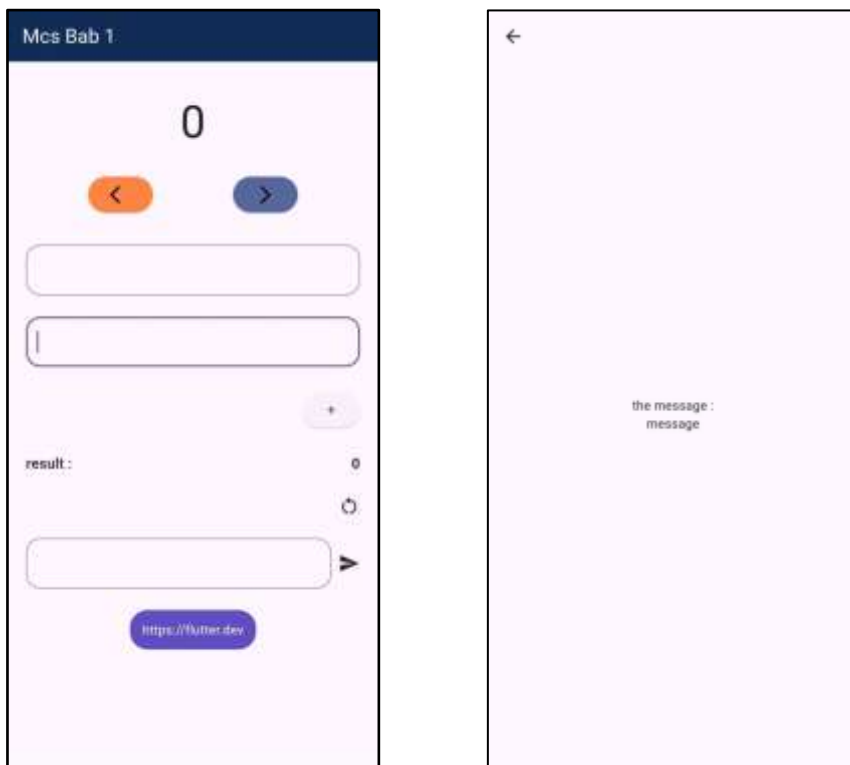
Widget-widget dasar pada Flutter

PENDAHULUAN

Pada praktikum bab 1 akan diperkenalkan cara membuat suatu halaman aplikasi yang berisikan widget-widget dasar yang ada pada flutter.

PRAKTIKUM BAB 1

Tampilan aplikasi yang akan dibangun



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

Buka folder lib dan buat 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 aplikasi menggunakan flutter fungsi main() menjadi kode yang paling utama agar aplikasi dapat berjalan. Dari fungsi main memanggil suatu class yang akan mengembalikan MaterialApp(). MaterialApp() menjadi widget paling dasar untuk membentuk suatu aplikasi. Untuk dapat menggunakan MaterialApp() maka perlu import 'package:flutter/material.dart';. MaterialApp() memiliki beberapa argumen yang dapat diisi. Seperti title: untuk label dari sebuah aplikasi, debugShowCheckedModeBanner: yang biasanya diisi false apabila aplikasi sudah siap untuk rilis agar banner aplikasi di pojok kanan menghilang, theme: untuk mengatur tema aplikasi dan home: untuk mengatur halaman mana yang ditampilkan pertama saat aplikasi dibuka.

Buat file bernama home_page.dart. 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(0xffff8343),
              ),
            ),

```

```

        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 nantinya akan dipanggil untuk argumen home: yang ada pada MaterialApp(). Terdapat perbedaan saat membuat class MyApp() dan class HomePage(), yang mana MyApp() extend dari StatelessWidget dan HomePage extend dari StatefulWidget. Berikut perbedaan 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 StatelessWidget 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)

Karena keterbatasan yang dimiliki oleh StatelessWidget dan juga peran MaterialApp() yang hanya untuk mendefinisikan aplikasi maka MyApp cukup untuk extend dari StatelessWidget.

```
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() berguna untuk mengurangi nilai dari variabel start. Sedangkan fungsi decrement berfungsi untuk menambahkan nilai dari variabel start. Namun ketika nilai start kurang dari 0 maka start akan kembali bernilai 0.

```

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 di variabel result. Sedangkan fungsi resetResult() 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: ,
);

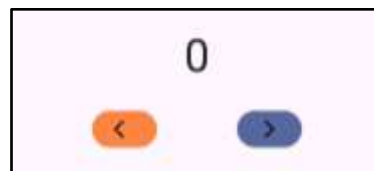
```


Scaffold digunakan untuk membuat halaman aplikasi, di dalam Scaffold tidak hanya memiliki properti appBar dan Body saja namun terdapat yang lain juga seperti backgroundColor:, floatingActionButton:, drawer: dan lain-lain. Di dalam Scaffold pada bab ini hanya akan menggunakan appBar dan body saja. Isi dari properti body: yaitu ListView, suatu widget yang memungkinkan pengguna untuk scroll berisikan widget-widget.



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

aplikasi di bab ini memiliki appBar yang di dalamnya bertuliskan “Mcs Bab 1” dan warna dari appBar itu sendiri 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(0xffFF8343),  
          ),  
          child: const Icon(Icons.arrow_back_ios, color: Colors.black),  
        ),  
  
        ElevatedButton(  
          onPressed: () {increment();},  
          style: ElevatedButton.styleFrom(  
            backgroundColor: const Color(0xff102C57),  
          ),  
          child: const Icon(Icons.arrow_forward_ios, color: Colors.white),  
        ),  
      ],  
    ),  
  ],  
)
```

```

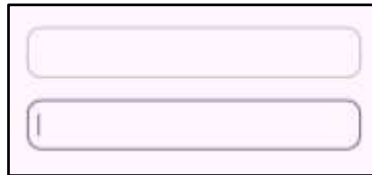
        backgroundColor: const Color(0xff55679C),
      ),
      child: const Icon(Icons.arrow_forward_ios, color: Colors.black,),
    ),
  ],
),

const SizedBox(height: 32,),

//...
],
),

```

Widget Center untuk membuat widget turunannya berada di tengah. Widget di dalam Center yaitu Text yang diambil dari variabel start. Variabel start memiliki nilai 0 maka yang akan ditampilkan di halaman aplikasi adalah 0. Widget Row() dapat memiliki turunan yang banyak dan mengarah horizontal. Di dalam Row pada kali ini terdapat 2 widget yang bernama ElevatedButton. Tombol diatur untuk mengubah nilai dari start. Tombol kiri untuk decrement() dan tombol kanan untuk increment.



```

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),
      ),
    ),
  ),
),
//...
],
),

```

Selanjutnya di bawahnya ada 2 form untuk input angka lalu dijumlahkan, form diatur agar memiliki border dan juga border diatur agar hanya angka saja yang dapat masuk. 2 form tersebut memiliki controller yang berbeda agar nilai yang masuk tidak sama. Controller diambil dari variabel `inputController1` dan `inputController2`. `inputController1` digunakan untuk menyimpan apa yang diisi di form pertama dan `inputController2` digunakan untuk menyimpan apa yang diisi di form kedua.



```

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,),

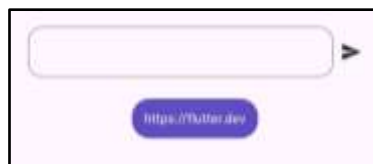
Container(
  margin: const EdgeInsets.symmetric(horizontal: 20),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      const Text(
        "result :",
        style: TextStyle(
          fontSize: 16,
          fontWeight: FontWeight.w600,
        ),
      ),
   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

//...
],
),

```

sumOf() dipanggil pada sebuah button yang berbentuk + untuk menjumlahkan angka yang masuk ke dalam input1Controller dan input2Controller. sumOf() memiliki konstruktor dengan argumen x dan y bertipe integer sehingga input1Controller dan input2Controller harus dikonversikan ke dalam bentuk integer. Hasil ditampung di variabel result dan lalu widget Text yang berisikan variabel result menampilkan hasil penjumlahannya. Dibawahnya terdapat icon bernama restart_alt berbentuk melingkar yang dibungkus oleh GestureDetector. GestureDetector berguna agar setiap widget yang diturunkannya dapat memiliki sebuah aksi ketika diklik. 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)),
          );
        },
      ),
    ],
  ),
),
```

```

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));
    },
  ),

  //...
),

```

Di bagian kode ini sebelumnya kita harus membuat file bernama receiver.dart berisikan class Receiver() untuk membentuk halaman yang akan menerima dan menampilkan tulisan yang kita kasih. Isi dari receiver.dart adalah sebagai berikut :

```

import 'package:flutter/material.dart';

class Receiver extends StatefulWidget {
  String message;

  Receiver({
    super.key,
    required this.message,
  });

  @override
  State<Receiver> createState() => _ReceiverState();
}

class _ReceiverState extends State<Receiver> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            const Text("the message :"),
            Text(widget.message),
          ],
        ),
      ),
    );
  }
}

```

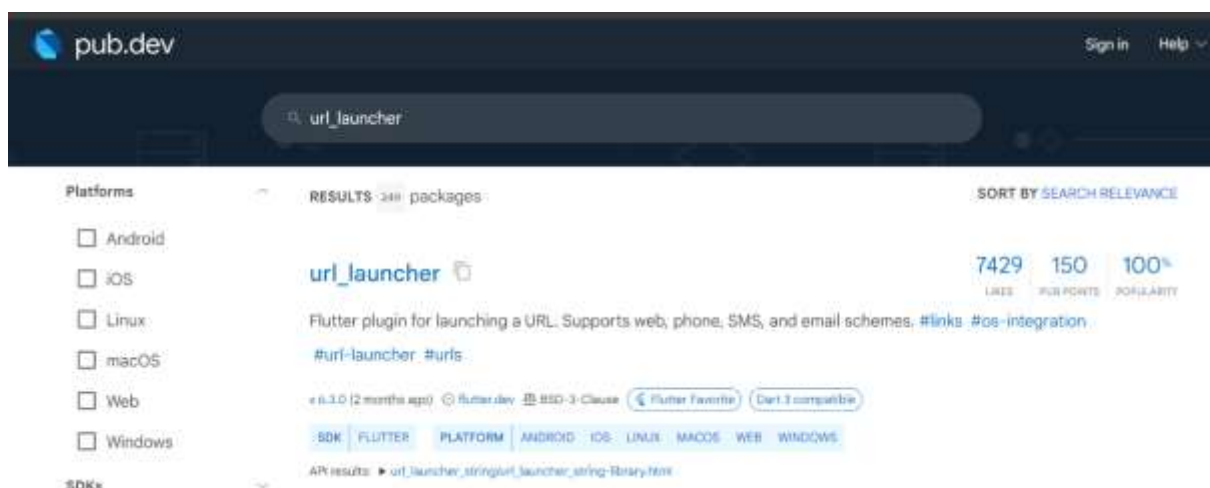
```
}
}
```



Terdapat konstruktor pada class Receiver() yang parameternya adalah variabel message. Hal tersebut digunakan untuk menampung input yang dimasukkan di class sebelumnya. Hasil ditampilkan di tengah-tengah halaman menggunakan Center().

Balik lagi ke halaman sebelumnya dimana terdapat form dan icon berbentuk send dalam 1 Row(). Isi dari form memiliki controller messageController untuk menampung input dan icon send dibungkus GestureDetector() agar dapat dilakukan suatu aksi. Aksi pada sebuah icon ini yaitu menuju halaman yang tadi sudah dibuat pada class Receiver() serta mengirimkan isi dari messageController.

Code bagian terakhir membuat sebuah button yang apabila diklik maka akan menuju halaman <https://flutter.dev>. Agar sebuah button dapat menuju <https://flutter.dev> maka diperlukan package bernama url_launcher: agar dapat membuka link yang ingin dituju. Cara memasangnya yaitu dengan cara buka file bernama pubspec.yaml dan tambahkan url_launcher: di bagian dependencies:



```
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

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

LAPORAN AKHIR

1. Berikan kesimpulan pada Bab 1!