

Modul Workshop Flutter

Make Portal News

Flutter adalah sebuah framework aplikasi mobil sumber terbuka yang **diciptakan oleh Google**. Flutter digunakan dalam pengembangan aplikasi untuk sistem operasi Android dan iOS. Sebelum pembuatan aplikasi dengan flutter, sebaiknya paham mengenai **istilah – istilah yang sering dibahas dalam proses development**, diantaranya:

A. Widget dan Property

Semua yang berhubungan dengan komponen Flutter disebut dengan Widget. Setiap Widget terdiri dari beberapa property. Nilai yang digunakan dalam sebuah property dapat berupa sebuah angka, string, maupun sebuah widget.

```
Container(  
  child: Text('Ini text widget'),  
  height: 30,  
), // Container
```

Pada gambar tersebut merupakan salah satu contoh untuk pembahasan widget dan property dimana **Container** dan **Text** merupakan sebuah Widget. Sedangkan **child** dan **height** merupakan sebuah Property dari Widget **Container**.

B. Stateless Widget vs Stateful Widget

Stateless Widget merupakan **widget yang statis** atau tidak dapat berubah-ubah. Sedangkan **Stateful Widget** merupakan **widget yang dinamis** atau dapat berubah-ubah.

C. Layout Widget

Layout Widget yang sering digunakan yaitu Row Widget dan Column Widget. Berikut ilustrasi hasil perbedaan antara kedua widget tersebut.

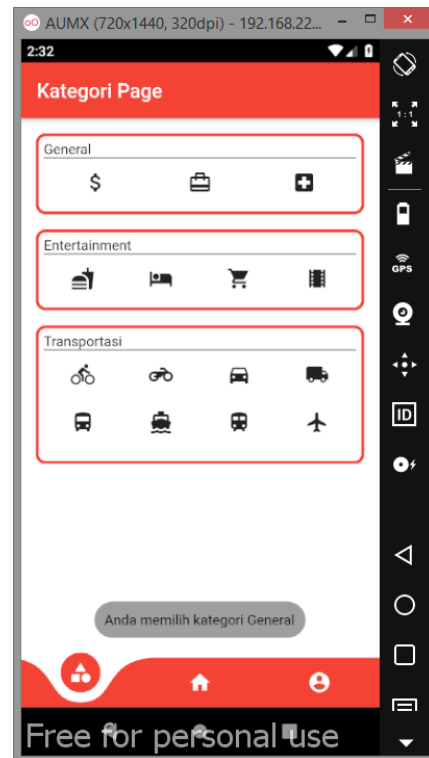
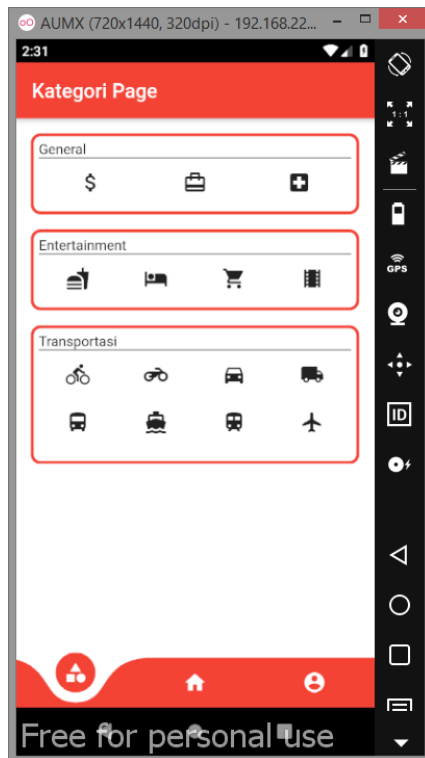


KONSEP APLIKASI

Aplikasi yang akan kita buat memiliki 3 tampilan utama diantaranya tampilan Kategori, Home dan Profil:

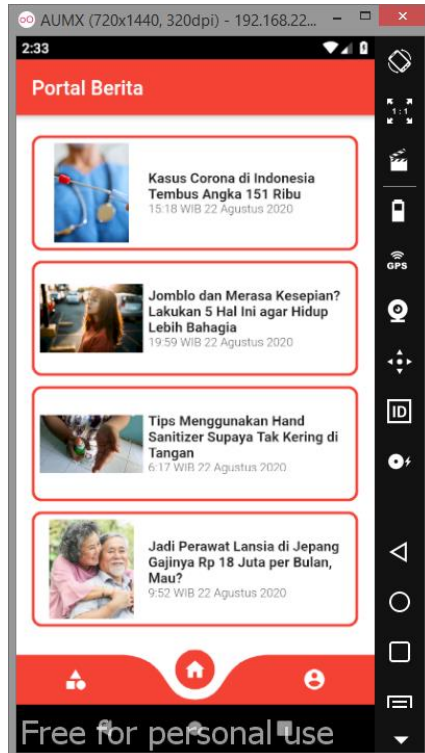
A. Halaman Kategori

Terdiri dari 3 kategori dimana tiap kategori nya terdiri dari beberapa icon seperti gambar sebelah kiri. Tiap icon sebuah button, dan jika icon di klik maka akan muncul toast seperti gambar sebelah kanan.



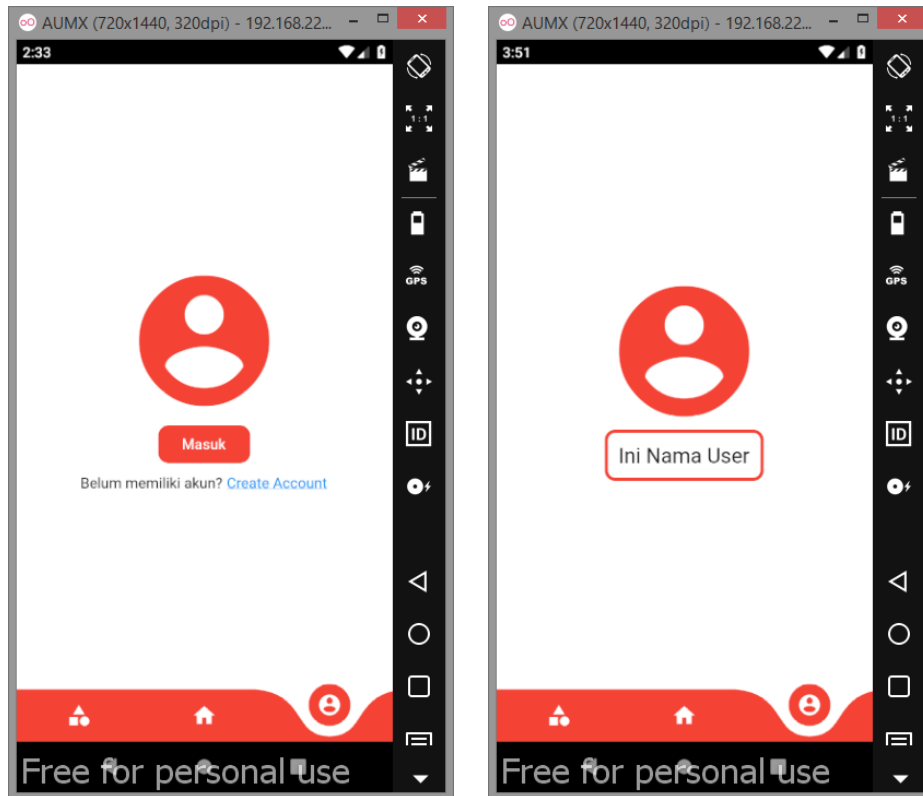
B. Halaman Home

Terdiri dari list data berita yang dapat di scroll secara vertical (seperti gambar sebelah kiri). Setiap data jika di klik maka akan pindah ke halaman detail sesuai dengan berita yang dipilih.

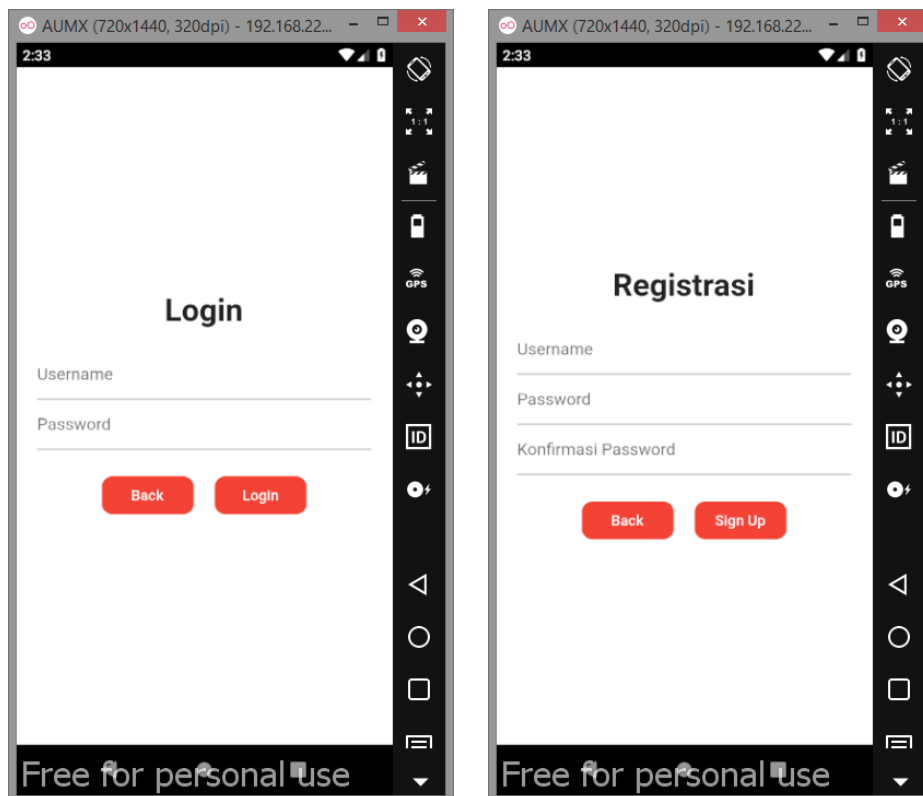


C. Halaman Profil

Ada 2 jenis tampilan diantaranya jika user belum login maka tampilan nya seperti gambar sebelah kiri, dan jika user sudah login maka tampilan nya seperti gambar sebelah kanan.

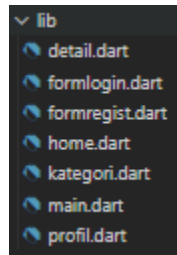


Saat user belum login, ada dua opsi yang dapat dipilih yaitu login dan regist. Untuk tampilan login (gambar sebelah kiri), dan tampilan regist (gambar sebelah kanan).

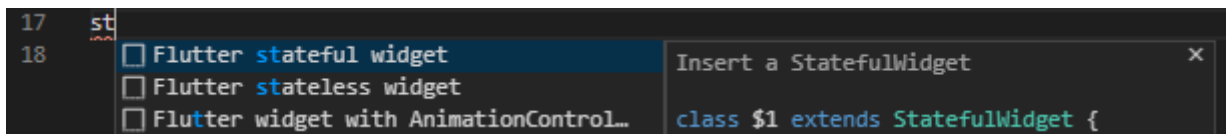


LET'S CODE

1. Buka folder project dalam Visual Studio Code, buatlah beberapa file diantaranya detail.dart , formlogin.dart , formregist.dart , home.dart , kategori.dart , dan profil.dart didalam folder lib.

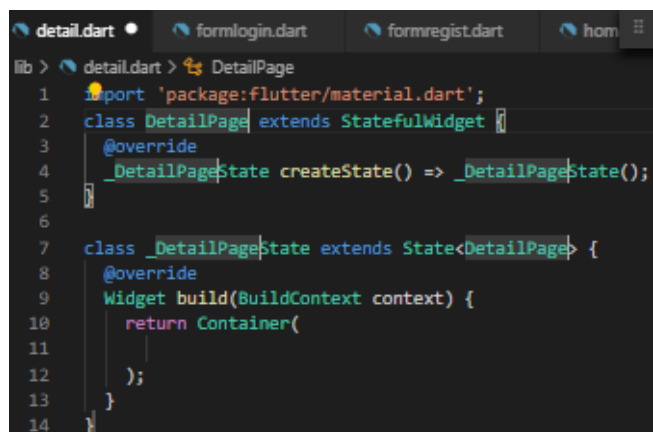
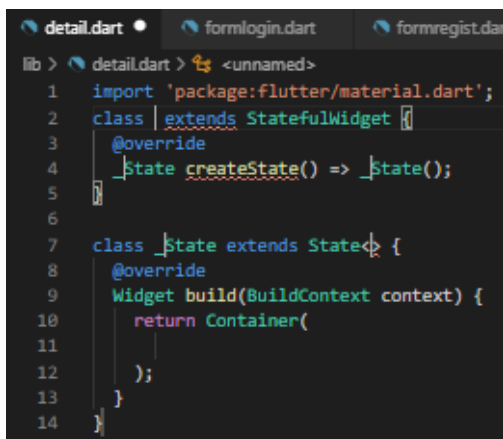


2. Pada setiap file yang telah dibuat pada langkah 1, ketikkan stf kemudian pilih Flutter stateful widget.



Berilah nama Widget pada masing-masing file tersebut sesuai keinginan contohnya seperti pada tabel dan gambar dibawah ini.

No	Nama File	Nama Class
1	detail.dart	DetailPage
2	formlogin.dart	LoginPage
3	formregist.dart	RegistPage
4	home.dart	HomePage
5	kategori.dart	KategoriPage
6	profil.dart	ProfilPage




3. Membuat **NavigationBar** (dari package curved navigation bar) dalam file **main.dart** seperti yang ditunjukkan pada gambar berikut :



Dokumentasi untuk library tersebut dapat dilihat pada link https://pub.dev/packages/curved_navigation_bar. Kemudian pilih tab installing jika ingin tau tahap penginstalan package tersebut seperti gambar berikut:

curved_navigation_bar 0.3.3

Published Jun 19, 2020 •  mobilemvp.app

FLUTTER | ANDROID | IOS | WEB

Readme | Changelog | Example | **Installing** | Versions | Scores

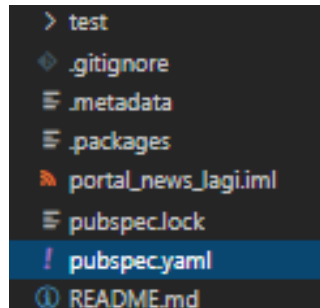
Use this package as a library

1. Depend on it

Add this to your package's pubspec.yaml file:

```
dependencies:  
  curved_navigation_bar: ^0.3.3
```

Untuk instalasi package step nya buka file pubspec.yaml pada folder root project



Kemudian tambahkan nama package kedalam dependencies yang ada pada pubspec.yaml

```
dependencies:  
  flutter:  
    sdk: flutter  
  
# The following adds the Cupertino Icons font to your application.  
# Use with the CupertinoIcons class for iOS style icons.  
cupertino_icons: ^0.1.3  
curved_navigation_bar: ^0.3.3
```

Hapus semua code yang ada pada file main.dart kemudian panggil package dan tambahkan beberapa code seperti berikut.

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

```

import 'package:curved_navigation_bar/curved_navigation_bar.dart'; // call package
import 'home.dart';
import 'kategori.dart';
import 'profil.dart';

void main() {
  runApp(MyApp(initialPage: 0));
}

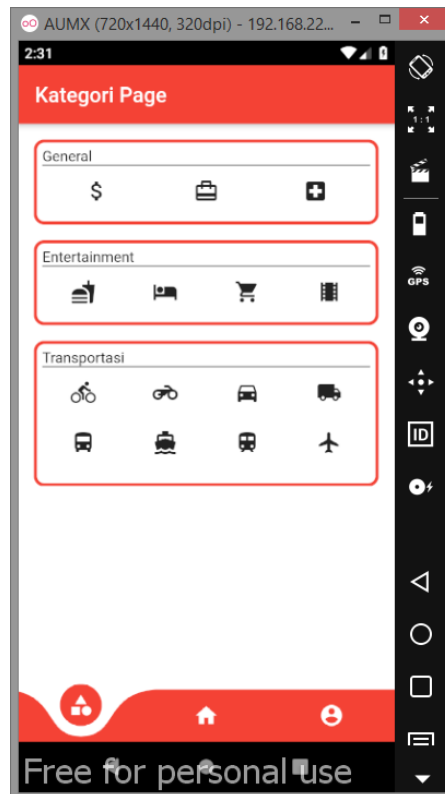
class MyApp extends StatefulWidget {
  const MyApp({this.initialPage});
  final int initialPage;
  @override
  _MyAppState createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  int _page = 0;
  final _pageOption = [KategoriPage(), HomePage(), ProfilPage()];
  @override
  void initState() {
    super.initState();
    setState(() {
      _page = widget.initialPage;
    });
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Portal News',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primarySwatch: Colors.red,
        scaffoldBackgroundColor: Colors.white,
      ),
      home: SafeArea(
        child: Scaffold(
          body: _pageOption[_page],
          bottomNavigationBar: CurvedNavigationBar(
            height: 50.0,
            color: Colors.red,
            backgroundColor: Colors.white,
            onTap: (index) {
              setState(() {
                _page = index;
              });
            },
            items: <Widget>[
              Icon(Icons.category, color: Colors.white),
              Icon(Icons.home, color: Colors.white),
              Icon(Icons.account_circle, color: Colors.white),
            ],
          ),
        ),
      ),
    );
  }
}

```

4. Pembuatan halaman kategori pada file kategori.dart.



- a. Membuat AppBar

Ubah Widget Container menjadi Scaffold dan gunakan property appBar didalam Widget tersebut seperti code berikut:

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

class KategoriPage extends StatefulWidget {
  @override
  _KategoriPageState createState() => _KategoriPageState();
}

class _KategoriPageState extends State<KategoriPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Kategori Page')),
    );
  }
}
```

- b. Pembuatan body dalam halaman kategori seperti berikut



Berdasarkan tampilan tersebut dapat diketahui bahwa terdapat beberapa tombol icon yang memiliki respon yang berbeda – beda sehingga dianjurkan membuat sebuah fungsi untuk menyimpan widget tersebut dengan parameter icon.

```
import 'package:flutter/material.dart';
class KategoriPage extends StatefulWidget {
  @override
  _KategoriPageState createState() => _KategoriPageState();
}
class _KategoriPageState extends State<KategoriPage> {
  iconTag(Icon icon) {
    return IconButton(
      icon: icon,
      onPressed: () {},
    );
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Kategori Page')),
    );
  }
}
```

Body yang ada pada halaman kategori memiliki jarak tepi dengan widget luar (margin), sehingga dapat menggunakan Widget Container dengan property margin serta child nya berisi Widget Column (karena terdiri dari beberapa Container yang berjajar secara vertical) dan Widget SizedBox (untuk menyediakan space kosong).

```
body: Container(
  margin: EdgeInsets.all(15),
  child: Column(
    children: [
      Container(...),
      SizedBox(height: 15),
      Container(...),
      SizedBox(height: 15),
      Container(...),
    ],
  ),
),
```

Di setiap Widget Container tersebut memiliki jarak tepi ke dalam (property padding), dan memiliki dekorasi border berwarna merah serta tepi melengkung (property decoration).


```
Container(
  padding: EdgeInsets.all(5),
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(10),
    border: Border.all(color: Colors.red, width: 3),
  ),
  child: ....
),
```

Sehingga didapatkan seperti gambar berikut:



Kemudian tambahkan property child ke setiap Widget Container tersebut dengan berisi Widget Column dan menggunakan property children didalamnya.

```
child: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    Text('General'),
    Container(height: 1.5, color: Colors.grey),
    Row(
      mainAxisAlignment: MainAxisAlignment.spaceAround,
      children: [
        iconTag(Icon(Icons.attach_money)),
        iconTag(Icon(Icons.card_travel)),
        iconTag(Icon(Icons.local_hospital)),
      ],
    ),
  ],
),
```

Sehingga didapatkan hasil seperti berikut:



Sesuai dengan konsep, kondisi yang digunakan yaitu jika icon di klik maka akan muncul sebuah toast yang muncul pada bagian bawah layar. Widget toast tersebut dapat menggunakan package `fluttertoast` (<https://pub.dev/packages/fluttertoast>). Maka yang perlu dimodifikasi yaitu fungsi `iconTag` yang telah dibuat sebelumnya menjadi seperti berikut (tag yang digunakan untuk parameter akan menyesuaikan dengan icon yang dipilih) :

```
iconTag(Icon icon, String tag) {
  return IconButton(
    icon: icon,
    onPressed: () {
      FToast(context).showToast(
        toastDuration: Duration(milliseconds: 2000),
        child: Container(
          margin: EdgeInsets.only(bottom: 20),
          padding: EdgeInsets.all(10),

```

```

        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(25.0),
          color: Colors.grey,
        ),
        child: Text('Anda memilih kategori ' + tag),
      ),
    ),
  ],
);
}

```

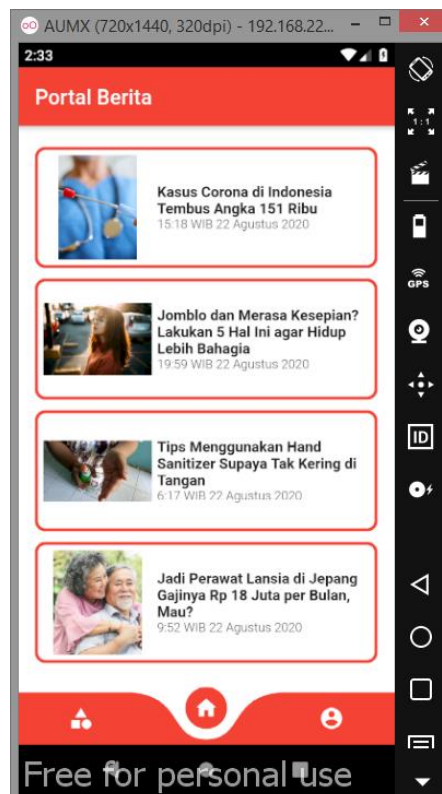
Dikarenakan ada tambahan parameter pada fungsi iconTag maka saat memanggil fungsi tersebut juga perlu diperbaiki menjadi :

```

child: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    Text('General'),
    Container(height: 1.5, color: Colors.grey),
    Row(
      mainAxisAlignment: MainAxisAlignment.spaceAround,
      children: [
        iconTag(Icon(Icons.attach_money), 'General'),
        iconTag(Icon(Icons.card_travel), 'General'),
        iconTag(Icon(Icons.local_hospital), 'General'),
      ],
    ),
  ],
),
),

```

5. Pembuatan halaman home pada file home.dart.



- a. Membuat AppBar
Ubah Widget Container menjadi Scaffold dan gunakan property appBar didalam Widget tersebut seperti code berikut:

```
// FILE home.dart

import 'package:flutter/material.dart';

class HomePage extends StatefulWidget {
  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Portal Berita')),
    );
  }
}
```

- b. Membuat variabel linkAPI (link yang menyimpan data – data secara online, untuk workshop kali ini menggunakan linkAPI <http://apinews.lokerprogrammer.com/>) dan membuat fungsi containerBerita untuk menyimpan widget container yang berisi data berita. Setiap container tersebut dapat di klik sehingga perlu dibungkus dengan Widget MaterialButton.

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

final String linkAPI = 'http://apinews.lokerprogrammer.com/';

class HomePage extends StatefulWidget {
  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  @override
  containerBerita(String datetime, String judul, String content, String image) {
    return MaterialButton(
      child: Container(...),
      splashColor: Colors.white,
      highlightColor: Colors.white,
      onPressed: () {
        Navigator.push(
          context, MaterialPageRoute(builder: (context) => DetailPage()));
      },
    );
  }
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Portal Berita')),
    );
  }
}
```

Di setiap Widget Container tersebut memiliki jarak tepi ke dalam (property padding), memiliki jarak tepi ke luar bagian bawah (property margin), dan memiliki dekorasi border berwarna merah serta tepi melengkung (property decoration).

```
child: Container(
  padding: EdgeInsets.all(5),
  margin: EdgeInsets.only(bottom: 10),
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(10),
    border: Border.all(color: Colors.red, width: 3),
```

```
),
  child: ....
),
```

Di setiap Widget Container tersebut berisi 2 widget berjajar secara horizontal sehingga menggunakan Widget Row dengan property children. Widget Row tersebut memiliki pembagian ukuran rasio dari kedua isi tersebut yaitu 1 : 2 sehingga dapat menggunakan Widget Flexible.

```
child: Row(
  children: [
    Flexible(
      flex: 1,
      fit: FlexFit.tight,
      child: Image.network(image, height: 100),
    ),
    Flexible(
      flex: 2,
      fit: FlexFit.tight,
      child: Container(...),
    ),
  ],
),
```

Didalam container tersebut berisi 2 teks yang berjajar secara vertical, sehingga menggunakan Widget Column dengan property children berupa 2 Widget Text.

```
child: Container(
  margin: EdgeInsets.all(5),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text(judul, style: TextStyle(fontWeight: FontWeight.bold)),
      Text(
        datetime,
        style:
          TextStyle(fontWeight: FontWeight.w300, fontSize: 12),
      ),
    ],
  ),
),
```

c. Pembuatan body pada halaman home.

```
return Scaffold(
  appBar: AppBar(title: Text('Portal Berita')),
  body: Container(
    margin: EdgeInsets.fromLTRB(0, 20, 0, 20),
    child: ....
  ),
);
```

Body pada halaman home terdiri dari list data – data yang dapat di scroll sehingga dapat menggunakan Widget ListView, dengan jumlah data menyesuaikan dengan data yang ada (dinamis), maka widget yang digunakan yaitu Widget ListView.builder .

```
child: ListView.builder(
  itemCount: 7,
  itemBuilder: (context, i) {
    return containerBerita('datetime', 'judul' + i.toString(),
      'content', linkAPI + 'img/corona.jpg');
  },
),
```

),

- d. List data berita pada halaman home menyesuaikan dengan data dari linkAPI, sehingga membutuhkan package http (dokumentasi dapat dilihat pada link <https://pub.dev/packages/http>).

Data yang dibaca dari linkAPI tersebut berupa JSON, sehingga perlu mencantumkan beberapa library. Serta buatlah fungsi **Future<List> getData() async** untuk mengambil data JSON tersebut.

```
import 'package:flutter/material.dart';
import 'detail.dart';
import 'dart:async';
import 'dart:convert';
import 'package:http/http.dart' as http;
final String linkAPI = 'http://apinews.lokerprogrammer.com/';

class HomePage extends StatefulWidget {
  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  Future<List> getData() async {
    final response = await http.get(linkAPI);
    return json.decode(response.body);
  }
}
```

Bungkus code utama body kedalam property builder pada Widget FutureBuilder. Gunakan property future pada Widget FutureBuilder yang berisi fungsi getData.

```
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: Text('Portal Berita')),
    body: FutureBuilder<List>(
      future: getData(),
      builder: (context, snapshot) {
        if (!snapshot.hasData) {
          return Center(child: CircularProgressIndicator());
        } else {
          return Container(
            margin: EdgeInsets.fromLTRB(0, 20, 0, 20),
            child: ....
          );
        }
      },
    ),
  );
}
```

Dan ubah data parameter dalam containerBerita menyesuaikan dengan data dari linkAPI seperti berikut:

```
child: ListView.builder(
  itemCount: snapshot.data.length,
  itemBuilder: (context, i) {
    return containerBerita(
      snapshot.data[i]['waktu'] +
        ' ' +
        snapshot.data[i]['tanggal'],
      snapshot.data[i]['judul'],
      snapshot.data[i]['content'],
      linkAPI + 'img/' + snapshot.data[i]['image'],
    );
  },
),
```

),

6. Pembuatan halaman detail pada file detail.dart



Data pada halaman detail didapatkan dari data berita yang telah dipilih pada halaman home, sehingga untuk menerima data tersebut maka halaman ini perlu constructor (method khusus yang akan dieksekusi pada saat pembuatan objek).

```
// FILE detail.dart
import 'package:flutter/material.dart';
class DetailPage extends StatefulWidget {
  const DetailPage({this.judul, this.content, this.datetime, this.image});
  final String judul;
  final String content;
  final String datetime;
  final String image;
  @override
  _DetailPageState createState() => _DetailPageState();
}
```

Sehingga saat pemanggilan Class DetailPage juga tambahkan parameter – parameter tersebut.

```
// FILE home.dart
....
containerBerita(String datetime, String judul, String content, String image) {
  return MaterialButton(
    ....
    onPressed: () {
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => DetailPage(
            judul: judul, content: content, datetime: datetime, image: image,
```

```

    ),
  ),
);
},
);
}
....

```

Pada halaman detail terdapat button lingkaran di pojok kanan bawah (menggunakan property `floatingActionButton`).

```

// FILE detail.dart
import 'package:flutter/material.dart';
class DetailPage extends StatefulWidget {
  const DetailPage({this.judul, this.content, this.datetime, this.image});
  final String judul;
  final String content;
  final String datetime;
  final String image;
  @override
  _DetailPageState createState() => _DetailPageState();
}

class _DetailPageState extends State<DetailPage> {
  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        floatingActionButton: FloatingActionButton(
          child: Icon(Icons.home),
          onPressed: () {
            Navigator.pop(context);
          },
        ),
        body: ....
      ),
    );
  }
}

```

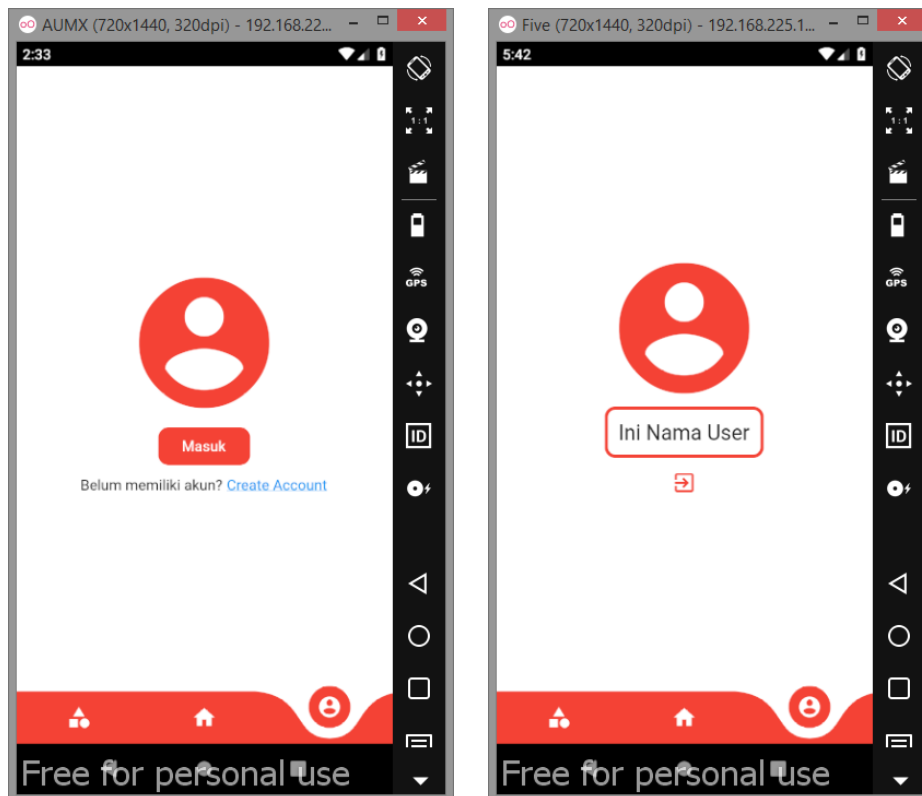
Body pada halaman detail dapat di scroll secara vertical, sehingga menggunakan Widget `ListView`.

```

body: Container(
  padding: EdgeInsets.fromLTRB(20, 20, 20, 80),
  child: ListView(
    children: [
      Text(
        widget.judul,
        style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
        textAlign: TextAlign.center,
      ),
      SizedBox(height: 20),
      Text(widget.datetime,
        style: TextStyle(fontWeight: FontWeight.w300)),
      SizedBox(height: 20),
      Image.network(widget.image, height: 200),
      SizedBox(height: 20),
      Text(
        widget.content,
        textAlign: TextAlign.justify,
      ),
    ],
  ),
),

```

7. Pembuatan halaman profil



Pada halaman profil terdapat 2 jenis tampilan tergantung kondisi belum / sudah login. Kondisi tersebut dapat disimpan menggunakan package Shared Preferences (https://pub.dev/packages/shared_preferences), dan import juga file untuk formulir login dan formulir regist. Buat fungsi initState yang akan menjalankan fungsi startLaunch pada saat pertama kali class ProfilPage dipanggil.

```
import 'package:flutter/material.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'dart:async';
import 'formlogin.dart';
import 'formregist.dart';
import 'main.dart';

class ProfilPage extends StatefulWidget {
  @override
  _ProfilPageState createState() => _ProfilPageState();
}

class _ProfilPageState extends State<ProfilPage> {
  var content;
  SharedPreferences prefs;

  @override
  void initState() {
    super.initState();
    startLaunch();
  }
  startLaunch(){...}
  logout(){...}
  belumLogin(){...}
  sudahLogin(){...}
  @override
  Widget build(BuildContext context) {return Center(child: content);}
```



```
}
```

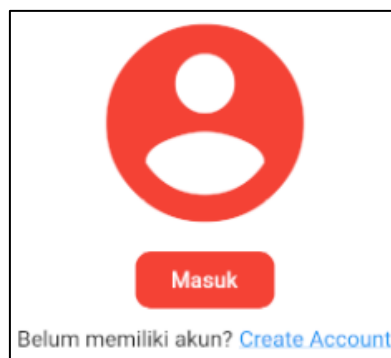
- a. Fungsi startLaunch digunakan untuk memeriksa apakah user sudah pernah login (dari preferences dengan key 'login'). Jika sudah pernah login (!= null), maka akan diarahkan ke fungsi sudahLogin. Sedangkan jika belum, maka akan diarahkan ke fungsi belumLogin. SetState digunakan isi dalam variabel content dapat terupdate berdasarkan kondisi login.

```
Future<void> startLaunch() async {  
  prefs = await SharedPreferences.getInstance();  
  var login = prefs.getString('login');  
  if (login != null) {  
    setState(() {  
      content = sudahLogin(login);  
    });  
  } else {  
    setState(() {  
      content = belumLogin();  
    });  
  }  
}
```

- b. Fungsi logOut digunakan untuk logout dengan cara menghapus preference (key 'login'), kemudian diarahkan ke tampilan profil.

```
Future<void> logOut() async {  
  print('logout');  
  prefs = await SharedPreferences.getInstance();  
  Navigator.of(context).pushReplacement(MaterialPageRoute(builder: (_) {  
    prefs.remove("login");  
    return MyApp(initialPage: 2);  
  }));  
}
```

- c. Fungsi belumLogin terdiri dari 3 widget yang sejajar secara vertical sehingga menggunakan Widget Column. Ketiga widget tersebut diantaranya sebuah icon, tombol (untuk diarahkan ke tampilan formlogin), dan Widget Row (yang berisi Widget Text dan GestureDetector).



```
belumLogin() {  
  return Column(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: [  
      Icon(Icons.account_circle, size: 150, color: Colors.red),  
      FlatButton(  
        child: Text('Masuk', style: TextStyle(color: Colors.white)),  
        color: Colors.red,  
        shape: RoundedRectangleBorder(  
          borderRadius: BorderRadius.circular(10.0),  
        ),  
      ),  
    ],  
  );  
}
```

```

        onPressed: () {
          Navigator.push(
            context, MaterialPageRoute(builder: (context) => LoginPage()));
        },
      ),
      SizedBox(height: 5),
      Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text('Belum memiliki akun? '),
          GestureDetector(
            child: Text(
              'Create Account',
              style: TextStyle(
                decoration: TextDecoration.underline, color: Colors.blue),
            ),
            onTap: () {
              Navigator.push(context,
                MaterialPageRoute(builder: (context) => RegistPage()));
            },
          ),
        ],
      ),
    ],
  ),
);
}

```

- d. Fungsi sudahLogin terdiri dari 3 widget yang sejajar secara vertical sehingga menggunakan Widget Column. Ketiga widget tersebut diantaranya sebuah icon, text, dan IconButton (untuk logout).



```

sudahLogin(String username) {
  return Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Icon(Icons.account_circle, size: 150, color: Colors.red),
      Container(
        padding: EdgeInsets.all(10),
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(10),
          border: Border.all(color: Colors.red, width: 3),
        ),
        child: Text(username, style: TextStyle(fontSize: 20)),
      ),
      IconButton(
        icon: Icon(Icons.exit_to_app),
        onPressed: () {
          logout();
        },
      ),
    ],
  );
}

```

```

    },
    color: Colors.red,
  )
],
);
}

```

e. Pembuatan halaman formRegist

Halaman ini membutuhkan linkAPI sehingga perlu di importkan file home.dart (karena terdapat variabel yang menyimpan linkAPI) dan package http karena aplikasi perlu mengirimkan data ke cloud melalui jaringan internet.

```

import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'home.dart';

class RegistPage extends StatefulWidget {
  @override
  _RegistPageState createState() => _RegistPageState();
}

class _RegistPageState extends State<RegistPage> {
  TextEditingController userController = TextEditingController();
  TextEditingController passController = TextEditingController();
  TextEditingController pass2Controller = TextEditingController();
  addDataUser(){
    ....
  }

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

```

Pada fungsi addDataUser memiliki 3 parameter. Semua isian formulir yang sudah diisi dan verifikasi password akan dijadikan sebuah kondisi untuk mengirimkan ke database menggunakan metode post. Jika tidak ada yang terima, jadi kita dapat surat lagi. Halaman ini membutuhkan linkAPI sehingga perlu di importkan file home.dart (karena terdapat variabel yang menyimpan linkAPI) dan package http karena aplikasi perlu mengirimkan data ke cloud melalui jaringan internet.

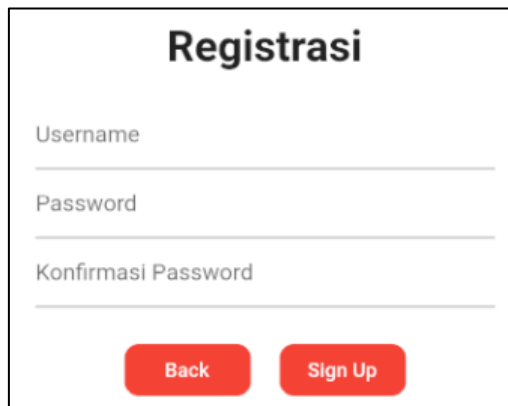
```

addDataUser(
  TextEditingController userController,
  TextEditingController passController,
  TextEditingController pass2Controller) {
  if (userController.text != '' &&
      passController.text != '' &&
      pass2Controller.text != '' &&
      passController.text == pass2Controller.text) {
    setState(() { var link = linkAPI + "regist.php";
      http.post(link, body: {
        "username": userController.text,
        "password": passController.text
      });
    });
    Navigator.pop(context);
  } else {
    userController.text = '';
    passController.text = '';
    pass2Controller.text = '';
  }
}

```

```
}
```

Dapat dilihat bahwa formregist memiliki beberapa widget yang berjajar secara vertical (sehingga menggunakan Widget Column).



The image shows a registration form titled "Registrasi". It contains three text input fields with labels "Username", "Password", and "Konfirmasi Password". Below the input fields, there are two red buttons: "Back" and "Sign Up".

```
@override
Widget build(BuildContext context) {
  return SafeArea(
    child: Scaffold(
      body: Center(
        child: Container(
          margin: EdgeInsets.all(20),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text(
                'Registrasi',
                style: TextStyle(fontWeight: FontWeight.bold, fontSize: 30),
              ),
              SizedBox(height: 20),
              TextField(
                controller: userController,
                decoration: InputDecoration(hintText: "Username"),
              ),
              TextField(
                controller: passController,
                decoration: InputDecoration(hintText: "Password"),
                obscureText: true,
              ),
              TextField(
                controller: pass2Controller,
                decoration: InputDecoration(hintText: "Konfirmasi Password"),
                obscureText: true,
              ),
              SizedBox(height: 20),
              Row(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  FlatButton(
                    child:
                      Text('Back', style: TextStyle(color: Colors.white)),
                    color: Colors.red,
                    shape: RoundedRectangleBorder(
                      borderRadius: BorderRadius.circular(10.0),
                    ),
                    onPressed: () {
                      Navigator.pop(context);
                    },
                  ),
                ],
              ),
            ],
          ),
        ),
      ),
    ),
  );
}
```

```

        SizedBox(width: 20),
        FlatButton(
          child: Text('Sign Up',
            style: TextStyle(color: Colors.white)),
          color: Colors.red,
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(10.0),
          ),
          onPressed: () {
            addDataUser(
              userController, passController, pass2Controller);
          },
        ),
      ],
    ),
  ],
),
),
),
),
);
}

```

f. Pembuatan halaman formLogin.

```

import 'package:flutter/material.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';
import 'dart:async';
import 'main.dart';
import 'home.dart';

```

```

class LoginPage extends StatefulWidget {
  @override
  _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  TextEditingController userController = TextEditingController();
  TextEditingController passController = TextEditingController();
  getData(){...}
  cekLogin(){...}

  @override
  Widget build(BuildContext context) {
    return ...;
  }
}

```

```

Future<List> getData() async {
  final response = await http.get(linkAPI + "login.php");
  return json.decode(response.body);
}

Future<void> cekLogin(List data, TextEditingController userController,
  TextEditingController passController) async {
  SharedPreferences prefs = await SharedPreferences.getInstance();
  int login = 0;
  for (var i = 0; i < data.length; i++) {
    if (data[i]['username'] == userController.text &&
      data[i]['password'] == passController.text) {
      login = 1;
    }
  }
}

```

```

if (login == 1) {
  Navigator.of(context).pushReplacement(MaterialPageRoute(builder: (_) {
    prefs.setString('login', userController.text);
    return MyApp(initialPage: 2);
  }));
}
// tidak bisa login
else {
  userController.text = '';
  passController.text = '';
  return FToast(context).showToast(
    toastDuration: Duration(milliseconds: 2000),
    child: Container(
      margin: EdgeInsets.only(bottom: 20),
      padding: EdgeInsets.all(10),
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(25.0),
        color: Colors.grey,
      ),
      child: Text('Username dan Password salah'),
    ),
  );
}
}
}

```

Login

Username

Password

Back
Login

```

@override
Widget build(BuildContext context) {
  return SafeArea(
    child: Scaffold(
      body: FutureBuilder<List>(
        future: getData(),
        builder: (context, snapshot) {
          if (snapshot.hasError) {
            print(snapshot.error);
          }
          if (!snapshot.hasData) {
            return Center(
              child: CircularProgressIndicator(),
            ); // untuk tampilan tidak ada data di db
          } else {
            return Center(
              child: Container(
                margin: EdgeInsets.all(20),
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: [
                    Text(
                      'Login',
                      style: TextStyle(
                        fontWeight: FontWeight.bold, fontSize: 30),
                    ),

```

```

    ),
    SizedBox(height: 20),
    TextField(
      controller: userController,
      decoration: InputDecoration(hintText: "Username"),
    ),
    TextField(
      controller: passController,
      decoration: InputDecoration(hintText: "Password"),
      obscureText: true,
    ),
    SizedBox(height: 20),
    Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        FlatButton(
          child: Text('Back',
            style: TextStyle(color: Colors.white)),
          color: Colors.red,
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(10.0),
          ),
          onPressed: () {
            Navigator.pop(context);
          },
        ),
        SizedBox(width: 20),
        FlatButton(
          child: Text('Login',
            style: TextStyle(color: Colors.white)),
          color: Colors.red,
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(10.0),
          ),
          onPressed: () {
            cekLogin(snapshot.data, userController.text,
              passController.text);
          },
        ),
      ],
    ),
  ],
),
);
}

```

Semangat ☺

"Do not fear failure but rather fear not trying" (Quote by Roy T. Bennett)