Pemrograman Mobile dengan Flutter

Dasar-Dasar Pemrograman Flutter

Team Pemrograman Mobile

Rules

Tepat Waktu

Komunikasi via sms atau wa
Monday-Friday(07.00 - 19.00 WIB)

STOP Plagiarism

Tidak berbuat curang
Discount 50%

Rules

- ➢ **Ketika bicara menyalakan video dan mic**
- ➢ **Merespon ketika dipanggil**
- ➢ **Menggunakan pakaian sesuai aturan JTI**
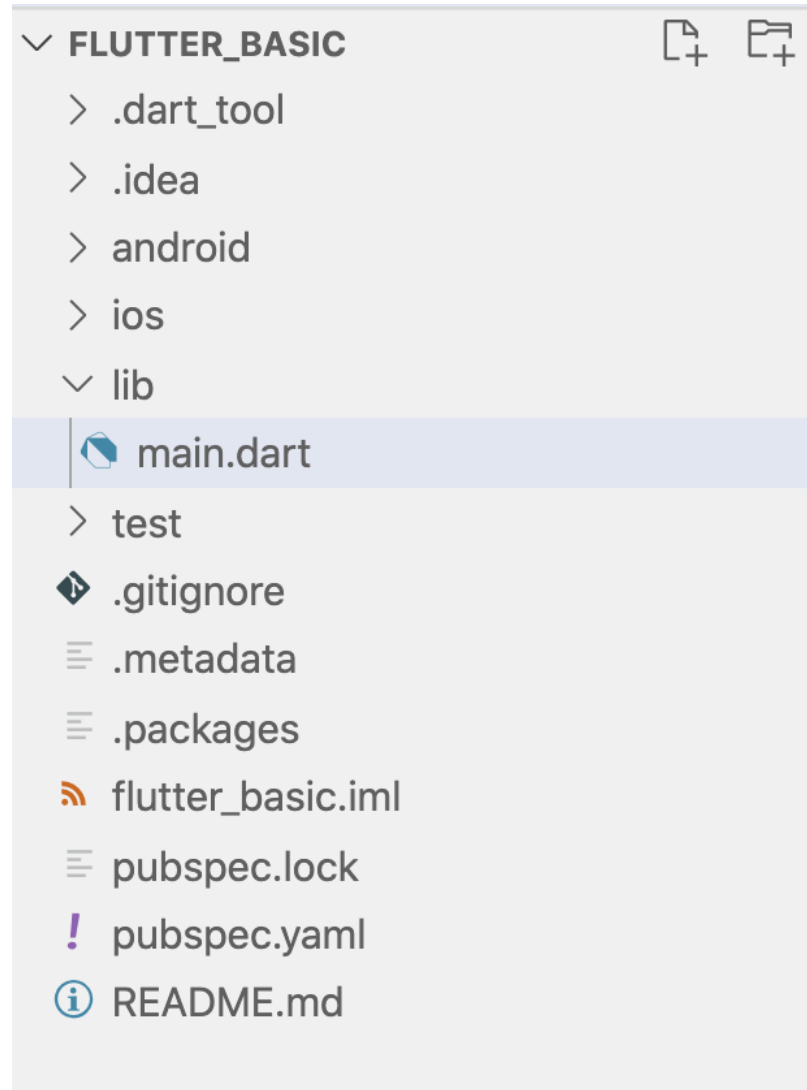- ➢ **Buat catatan**
- ➢ **Cek kuota**

Outline

➢ Struktur Project Flutter

➢ Hot Reload dan Hot Restart

➢ Pengenalan Flutter Widget

➢ Praktikum

  ➢ Membuat Flutter UI
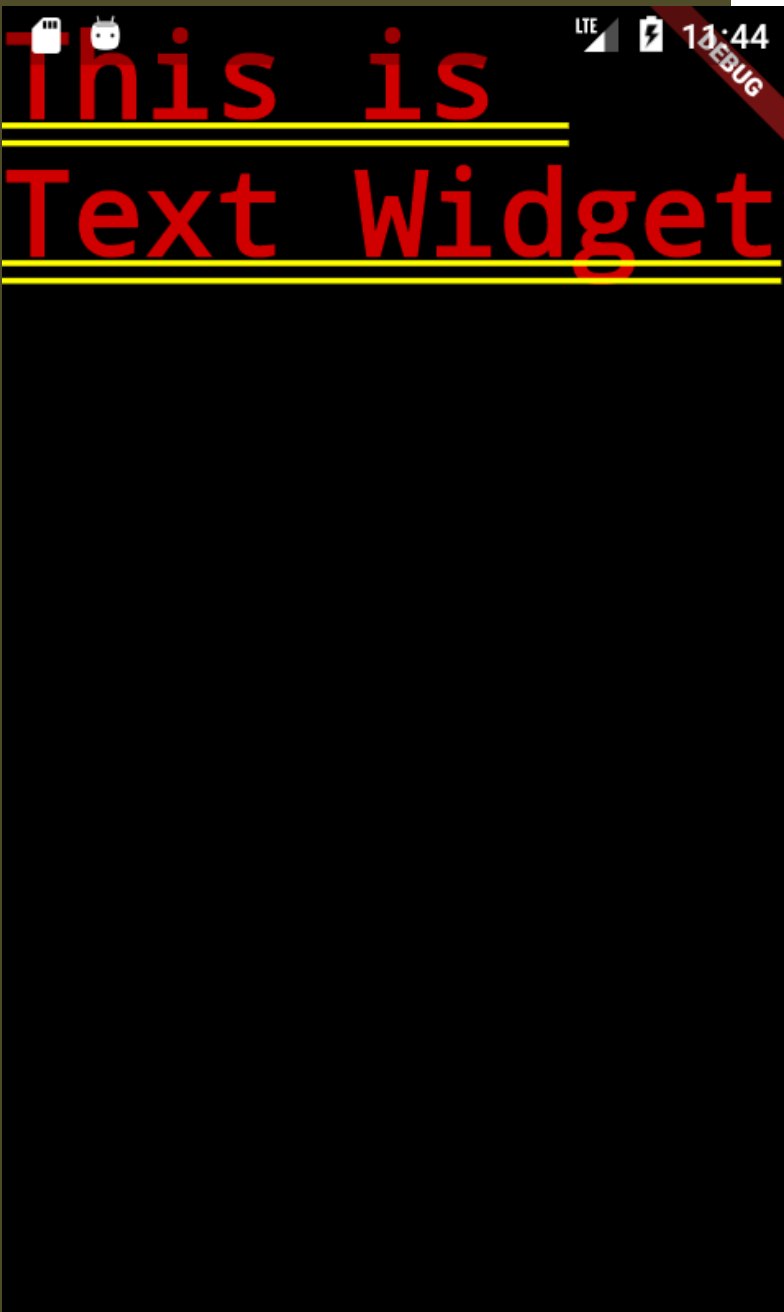
Capaian Pembelajaran Minggu 1

- Mengenal Widget Pada Flutter

- Mengenal Widget Tree

- Membuat UI pada Flutter

Struktur Project Flutter

Hot Reload dan Hot Restart

➢ Hot reload mencompile source code yang baru ditambahkan dan dikirimkan ke dart virtual machine diupdate. Setelah selesai update, dart virtual machine akan memperbarui UI sesuai dengan perubahan.

➢ Keunggulan hot reload adalah waktu prosenya cepat disbanding hot restart.

➢ Akan tetapi hot reload mempertahan state yang ada sehingga jika menggunakan state maka nilai dari widget tidak akan berubah.

➢ Hot restart akan mencompile ulang aplikasi dan mereset (destroy) state yang ada. Jadi hot restart akan membuild ulang widget tree sesuai dengan code yang telah diperbarui.

Text Widget

```
class MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
  return MaterialApp(
  home : Text('This is Text Widget',),
  );
}
}
```

Image Widget

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

  return MaterialApp(

  home:Image(image: NetworkImage('https://flutter.github.io/assets- for-api-docs/assets/widgets/owl.jpg'),),

  );

}

}

Material Design dan iOS Cupertino

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

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Container(
        margin: EdgeInsets.only(top: 30),
        color: Colors.white,
        child: Column(
          children: <Widget>[
            AppBar(title: Text('Contoh Cupertino')),
            CupertinoButton(
              child: Text("Contoh button"),
              onPressed: () {},
            ),
            CupertinoActivityIndicator(),
```

Button Widget

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        floatingActionButton:FloatingActionButton(
          onPressed: () {
            // Add your onPressed code here!
          },
          child: Icon(Icons.thumb_up),
          backgroundColor: Colors.pink,
        ),
      ),
    );
  }
}
```
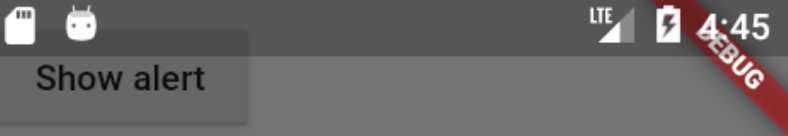
Scaffold Widget

```dart
Class MyApp extends StatelessWidget {
  int _count = 0;
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('Sample Code'),
        ),
        body: Center(
          child: Text('You have pressed the button $_count times.'),
        ),
        bottomNavigationBar: BottomAppBar(
          child: Container(
            height: 50.0,
          ),
        ),
        floatingActionButton: FloatingActionButton(
          onPressed: () => 0,
          tooltip: 'Increment Counter',
          child: Icon(Icons.add),
        ),
        floatingActionButtonLocation:
FloatingActionButtonLocation.centerDocked,
      ),
    );
  }
}
```

Dialog Widget

```
showAlertDialog(BuildContext context) {

  // set up the button
  Widget okButton = FlatButton(
    child: Text("OK"),
    onPressed: () { },
  );


  // set up the AlertDialog
  AlertDialog alert = AlertDialog(
    title: Text("My title"),
    content: Text("This is my message."),
    actions: [
      okButton,
    ],
  );


  // show the dialog
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return alert;
    },
  );
}
```
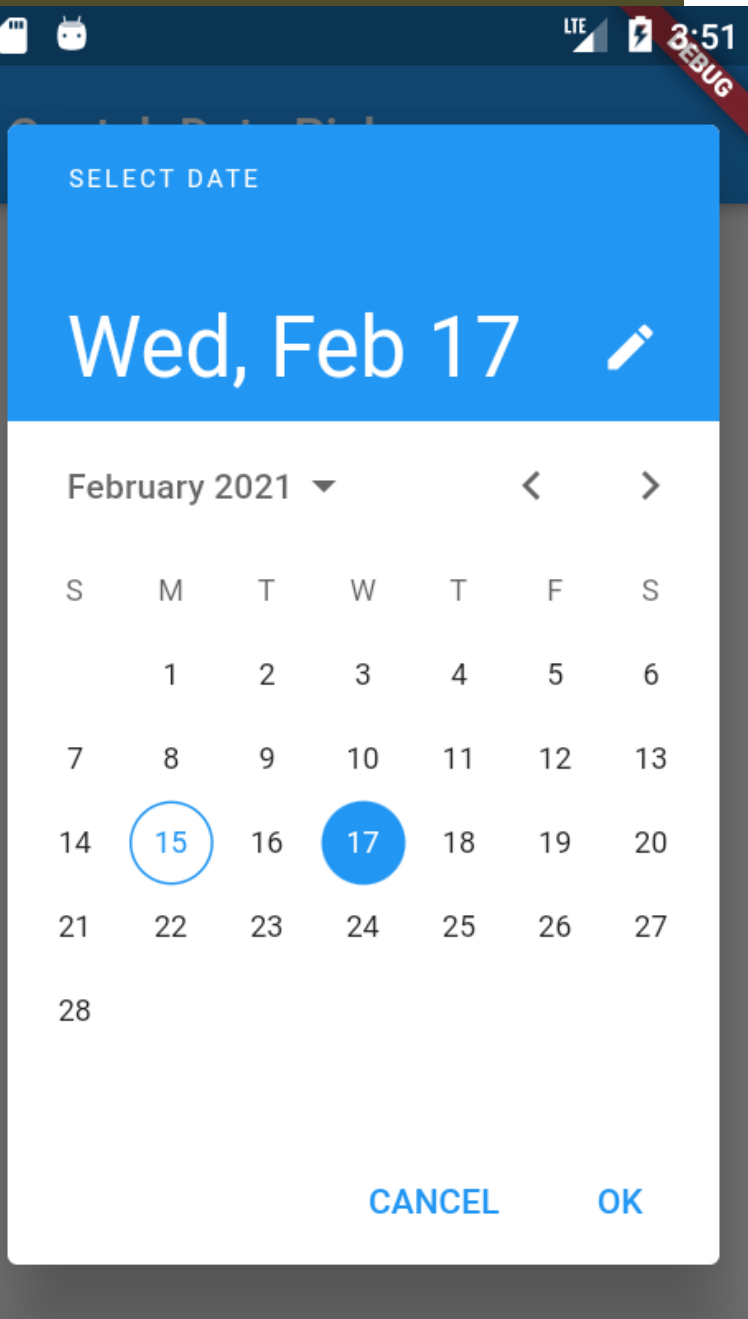
Input dan Selection Widget


```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(


      home: Scaffold(
        appBar: AppBar(
          title: Text("Contoh TextField")
        ),
        body: TextField(
            obscureText: false,
            decoration: InputDecoration(
              border: OutlineInputBorder(),

              labelText: 'Nama',
            ),
          ),
        ),
      );
  }
}
```

Date and Time Pickers Widget

```
class _MyHomePageState extends State<MyHomePage> {
  // Variable/State untuk mengambil tanggal
  DateTime selectedDate = DateTime.now();

  // Initial SelectDate FLutter
  Future<Null> _selectDate(BuildContext context) async {
    // Initial DateTime FIinal Picked
    final DateTime picked = await showDatePicker(
        context: context,
        initialDate: selectedDate,
        firstDate: DateTime(2015, 8),
        lastDate: DateTime(2101));
    if (picked != null && picked != selectedDate)
      setState(() {
        selectedDate = picked;
      });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ),
      body: Center(
        child: Column(
          mainAxisSize: MainAxisSize.min,
          children: <Widget>[
            Text("${selectedDate.toLocal()}".split(' ')[0]),
            SizedBox(height: 20.0,),
            RaisedButton(
              onPressed: () => {
                _selectDate(context),
                print(selectedDate.day    +    selectedDate.month
selectedDate.year )
              },
```
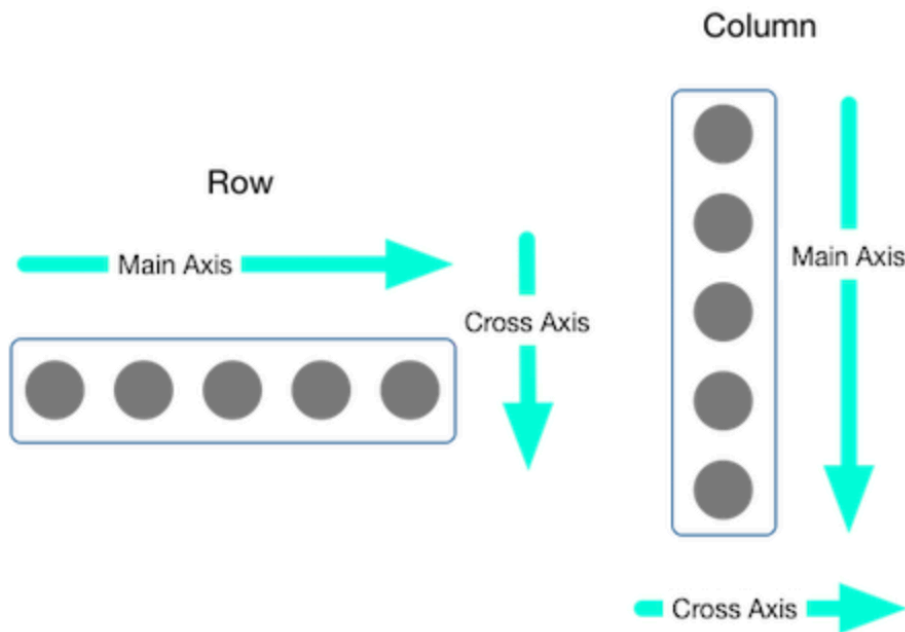
Container Widget

Container widget berguna untuk menyimpan berbagai macam attribute dan menampung berbagai macam fungsi objek. Container merupakan **single child objek** yang artinya hanya dapat memiliki satu buah child widget, akan tetapi dalam sebuat container kita dapat menempatkan row, column, text dan container lain.

Row dan Column Widget

Column widget digunakan untuk mangatur tata letak widget secara vertikal. Sedangkan row digunakan untuk mengatur tata letak widget secara horizontal.

Stack Widget

```dart
class MyApp extends StatelessWidget {
  @override

  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: Text("Contoh Stack Widget"),
        ),
        body: Stack(
          children: <Widget>[
            Container(
              color: Colors.green,
              alignment:Alignment.bottomCenter,
              child:  Text("Satu",  style:  TextStyle(fontSize:30,color:
Colors.white)),
            ),
            Container(
              color: Colors.red,
              alignment:Alignment.bottomCenter,
              child:  Text("Dua",  style:  TextStyle(fontSize:30,color:
Colors.white)),
              height: 400.0,
              width: 300.0,
            ),
            Container(
              color: Colors.deepPurple,
              alignment:Alignment.bottomCenter,
              child:  Text("Tiga",  style:  TextStyle(fontSize:30,color:
Colors.white)),
              height: 200.0,
              width: 200.0,
            ),
          ],
```

ListView Widget

ListView widget digunanakan untuk menampilkan data dalam bentuk list dan jika datanya melebihi dari render box maka halaman tersebut dapat di scroll.

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: Text("Contoh Stack Widget"),
        ),
        body: Stack(
          children: <Widget>[
            Container(
              color: Colors.green,
              alignment:Alignment.bottomCenter,
              child:  Text("Satu",  style:  TextStyle(fontSize:30,color:
Colors.white)),
            ),
            Container(
              color: Colors.red,
              alignment:Alignment.bottomCenter,
              child:  Text("Dua",  style:  TextStyle(fontSize:30,color:
Colors.white)),
              height: 400.0,
              width: 300.0,
            ),
            Container(
              color: Colors.deepPurple,
              alignment:Alignment.bottomCenter,
              child:  Text("Tiga",  style:  TextStyle(fontSize:30,color:
Colors.white)),
              height: 200.0,
              width: 200.0,
            ),
          ],
```

GridView Widget


```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          title: Text("Contoh GridView Widget"),
        ),
        body: GridView.count(
          crossAxisCount: 2,
          children: <Widget>[
            Container(
              color: Colors.green,
              alignment:Alignment.center,
              child:  Text("1",  style:  TextStyle(fontSize:30,color:
Colors.white)),
            ),
```

```
            Container(
              color: Colors.red,
              alignment:Alignment.center,
              child:  Text("2",  style:  TextStyle(fontSize:30,color:
Colors.white)),
              height: 400.0,
              width: 300.0,
            ),
            Container(
              color: Colors.deepPurple,
              alignment:Alignment.center,
              child:  Text("3",  style:  TextStyle(fontSize:30,color:
Colors.white)),
              height: 200.0,
              width: 200.0,
            ),
```

Pertanyaan

Praktikum

Jobsheet minggu ke -2

Referensi

➢ https://flutter.dev/docs/reference/widgets

➢ https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/