# FLUTTER NOTE

# State Stuff’s

Stateless vs Stateful :

**Stateless** - > aplikasi yang tidak membutuhkan perubahan state / keadaan

**Stateful** -> Aplikasi yang membutuhkan perubahan state

**State Management** -> tidak akan merender ulang bagian headbar

dapat menggunakan stateless rasa stateful, karena kita bisa mengubah

widget mana saja yang mau diubah

Stateless Widget -> kondisi page yang tidak memerlukan perubahan

A screenshot of a computer

Description automatically generated with low confidence

Stateful Widget -> kondisi page yang memerlukan perubahan/event

(increment atau ada button lain)

A screenshot of a computer

Description automatically generated with low confidence

# Styling Text

Styling :

A picture containing text, screenshot, font, software

Description automatically generated

# Column , Row and Stack

## Column

*body*: Column(

*mainAxisAlignment*: *MainAxisAlignment*.center,

*crossAxisAlignment*: *CrossAxisAlignment*.start,

*children*: [

              Container(

*width*: 200,

*height*: 50,

*color*: Colors.green,

              ),

              Container(

*width*: 50,

*height*: 50,

*color*: Colors.blue,

              ),

              Container(

*width*: 300,

*height*: 50,

*color*: Colors.red,

              ),

            ],

          )),

MainAxisAlignment :  
A screenshot of a computer

Description automatically generated

Result :

A colorful rectangles on a white background

Description automatically generated with low confidence

## Row

*body*: Row(

*mainAxisAlignment*: *MainAxisAlignment*.center,

*crossAxisAlignment*: *CrossAxisAlignment*.start,

*children*: [

              Container(

*height*: 200,

*width*: 50,

*color*: Colors.green,

              ),

              Container(

*height*: 50,

*width*: 50,

*color*: Colors.blue,

              ),

              Container(

*height*: 300,

*width*: 50,

*color*: Colors.red,

              ),

            ],

          )),

Result :

A red and green rectangles

Description automatically generated with medium confidence

## Stack

*body*: Stack(

*children*: [

              Container(

*height*: 400,

*width*: 400,

*color*: Colors.green,

              ),

              Container(

*height*: 300,

*width*: 300,

*color*: Colors.blue,

              ),

              Container(

*height*: 200,

*width*: 200,

*color*: Colors.red,

              ),

            ],

          )),

# Assign List Variable to children

class MyApp extends StatelessWidget {

  List<Widget> myList = [

    Container(

*height*: 300,

*width*: 300,

*color*: Colors.red,

    ),

    Container(

*height*: 300,

*width*: 300,

*color*: Colors.green,

    ),

    Container(

*height*: 300,

*width*: 300,

*color*: Colors.blue,

    ),

    Container(

*height*: 300,

*width*: 300,

*color*: Colors.yellow,

    ),

  ];

*@*override

  Widget build(BuildContext *context*) {

    return MaterialApp(

*debugShowCheckedModeBanner*: false,

*home*: Scaffold(

*appBar*: AppBar(

*title*: Text(

          "Column",

        )),

*body*: ListView(

          // scrollDirection: Axis.horizontal, untuk scroll ke horizontal

          // heightnya akan langsung jadi full kalo

*children*: myList,

        ),

      ),

    );

  }

}

# ListView

*body*: ListView(

          // scrollDirection: Axis.horizontal, untuk scroll ke horizontal

          // heightnya akan langsung jadi full kalo

*children*: [

            Container(

*height*: 300,

*width*: 300,

*color*: Colors.red,

            ),

            Container(

*height*: 300,

*width*: 300,

*color*: Colors.green,

            ),

            Container(

*height*: 300,

*width*: 300,

*color*: Colors.blue,

            ),

            Container(

*height*: 300,

*width*: 300,

*color*: Colors.yellow,

            ),

          ],

        ),

Result :  
A red blue and green rectangle

Description automatically generated with low confidence

\*bisa di scroll kebawah

## List.Generate()

class MyApp extends StatelessWidget {

  final List<Color> myColor = [

    Colors.red,

    Colors.green,

    Colors.blue,

    Colors.amber

  ];

  final List<Widget> myList = List.generate(

    100,

    (*index*) => Text(*index*.toString()),

  );

*@*override

  Widget build(BuildContext *context*) {

    return MaterialApp(

*debugShowCheckedModeBanner*: false,

*home*: Scaffold(

*appBar*: AppBar(

*title*: Text(

            "Column",

          ),

        ),

*body*: ListView(

*children*: myList,

        ),

      ),

    );

  }

}

A screenshot of a computer

Description automatically generated with low confidence

## ListView.Builder

class MyApp extends StatelessWidget {

  List<Color> myColor = [Colors.red, Colors.green, Colors.blue, Colors.amber];

*@*override

  Widget build(BuildContext *context*) {

    return MaterialApp(

*debugShowCheckedModeBanner*: false,

*home*: Scaffold(

*appBar*: AppBar(

*title*: Text(

          "Column",

        )),

*body*: ListView.builder(

*itemCount*: myColor.length, // kalo indexnya banyak pake length aja biar langsung dibaca semua

*itemBuilder*: (*context*, *index*) {

            return Container(

*height*: 300,

*width*: 300,

*color*: myColor[*index*],

            );

          },

        ),

      ),

    );

  }

}

Result :

A red blue and green rectangle

Description automatically generated with low confidence

## ListView.Separated

class MyApp extends StatelessWidget {

  List<Color> myColor = [Colors.red, Colors.green, Colors.blue, Colors.amber];

*@*override

  Widget build(BuildContext *context*) {

    return MaterialApp(

*debugShowCheckedModeBanner*: false,

*home*: Scaffold(

*appBar*: AppBar(

*title*: Text(

          "Column",

        )),

*body*: ListView.separated(

*separatorBuilder*: (*context*, *index*) {

            return Container(

*height*: 10,

            );

          },

*itemCount*: myColor.length,

*itemBuilder*: (*context*, *index*) {

            return Container(

*height*: 300,

*width*: 300,

*color*: myColor[*index*],

            );

          },

        ),

      ),

    );

  }

}

A screenshot of a computer

Description automatically generated with low confidence

Ada separator antara container

## List.Divider

class MyApp extends StatelessWidget {

  List<Color> myColor = [Colors.red, Colors.green, Colors.blue, Colors.amber];

*@*override

  Widget build(BuildContext *context*) {

    return MaterialApp(

*debugShowCheckedModeBanner*: false,

*home*: Scaffold(

*appBar*: AppBar(

*title*: Text(

          "Column",

        )),

*body*: ListView.separated(

*separatorBuilder*: (*context*, *index*) {

            return Divider(

*color*: Colors.black,

            );

          },

*itemCount*: myColor.length,

*itemBuilder*: (*context*, *index*) {

            return Container(

*height*: 300,

*width*: 300,

*color*: myColor[*index*],

            );

          },

        ),

      ),

    );

  }

}

A screenshot of a computer

Description automatically generated with low confidence A screenshot of a cell phone

Description automatically generated with medium confidence

\*Perbedaannya dengan separator adalah, divider akan ngasih warna di tengah”nya

# List Tile

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

*@*override

  Widget build(BuildContext *context*) {

    return MaterialApp(

*debugShowCheckedModeBanner*: false,

*home*: Scaffold(

*appBar*: AppBar(

*title*: Text("List Tile"),

        ),

*body*: ListView(

*children*: [

            ListTile(

*title*: Text("Ragil Jaya"),

*subtitle*: Text("Asoy Enyoy"),

*leading*: CircleAvatar(),

*trailing*: Text("10:00PM"),

            ),

            Divider(),

            ListTile(

*title*: Text("Ragil Jaya"),

*subtitle*: Text("Asoy Enyoy"),

*leading*: CircleAvatar(),

*trailing*: Text("10:00PM"),

            ),

          ],

        ),

      ),

    );

  }

}

Output :  
A screen shot of a phone

Description automatically generated with low confidence

# Image Widget

Buka pubspec.yaml , terus uncomment bagian assets:

 # To add assets to your application, add an assets section, like this:

  assets:

    - images/testimage.png

  #   - images/a\_dot\_ham.jpeg

## Asset

class MyApp extends StatelessWidget {

  const MyApp({*super*.*key*});

*@*override

  Widget build(BuildContext *context*) {

    return MaterialApp(

*home*: Scaffold(

*appBar*: AppBar(

*title*: Text("Image Widget"),

        ),

*body*: Center(

*child*: Container(

*width*: 350,

*height*: 500,

*color*: Colors.amber,

*child*: Image(

*image*: AssetImage("images/testimage.png"),

            ),

          ),

        ),

      ),

    );

  }

}

Result :

A picture containing text, screenshot, font, logo

Description automatically generated

## Network

class MyApp extends StatelessWidget {

  const MyApp({*super*.*key*});

*@*override

  Widget build(BuildContext *context*) {

    return MaterialApp(

*home*: Scaffold(

*appBar*: AppBar(

*title*: Text("Image Widget"),

        ),

*body*: Center(

*child*: Container(

*width*: 350,

*height*: 500,

*color*: Colors.amber,

*child*: Image(

*image*: NetworkImage("https://picsum.photos/200/300"),

            ),

          ),

        ),

      ),

    );

  }

}

Result :

A screen shot of a cell phone

Description automatically generated with low confidence