

Color\_bloc.dart

import 'dart:async';

import 'package:flutter/material.dart';

enum **ColorEvent** {

to\_amber,

to\_light\_blue

} *// enum untuk membuat jenis-jenis color event*

class **ColorBloc** {

**Color** \_color = **Colors**.amber; *// color pertama x amber*

*// \_color private gakbisa di akses dari luar*

**StreamController**<**ColorEvent**> \_eventController = *// event = Output*

**StreamController**<**ColorEvent**>();

**StreamSink**<**ColorEvent**> get **eventSink** => \_eventController.sink;

**StreamController**<**Color**> \_stateController =

**StreamController**<**Color**>(); *// state = Input*

**StreamSink**<**Color**> get **\_stateSink** => \_stateController.sink;

**Stream**<**Color**> get **stateStream** => \_stateController.stream;

*// ketika UI akan memberikan event masuk lewat sink, maka stream akan masuk ke mapEvent,*

*// berdasarkan color event dia akan mengubah state.*

*// berfungsi sebagai prosesor.*

void **\_mapEventToState**(**ColorEvent** colorEvent) {

if (colorEvent == **ColorEvent**.to\_amber)

\_color = **Colors**.amber;

else

\_color = **Colors**.lightBlue;

\_stateSink.**add**(\_color);

}

*// untuk menghubungkan abis dari streamcontroller lanjut ke \_mapEventState*

**ColorBloc**() {

\_eventController.stream.**listen**(\_mapEventToState);

}

void **dispose**() {

*// dispose untuk langsung menutup streamcontroller biar tidak makan memori*

\_eventController.**close**();

\_stateController.**close**();

}

}

Main.dart

import 'package:bloc\_latihan/color\_bloc.dart';

import 'package:flutter/material.dart';

void **main**() {

**runApp**(**MyApp**());

}

class **MyApp** extends **StatefulWidget** {

@override

**\_MyAppState** **createState**() => **\_MyAppState**();

}

class **\_MyAppState** extends **State**<**MyApp**> {

**ColorBloc** bloc = **ColorBloc**();

@override

void **dispose**() {

bloc.**dispose**();

}

@override

**Widget** **build**(**BuildContext** context) {

return **MaterialApp**(

home: **Scaffold**(

appBar: **AppBar**(

title: **Text**("BLoC Tanpa Library"),

),

floatingActionButton: **Row**(

mainAxisAlignment: **MainAxisAlignment**.end,

children: [

**FloatingActionButton**(

onPressed: () {

bloc.eventSink.**add**(**ColorEvent**.to\_amber);

},

backgroundColor: **Colors**.amber,

),

**SizedBox**(width: 10),

**FloatingActionButton**(

onPressed: () {

bloc.eventSink.**add**(**ColorEvent**.to\_light\_blue);

},

backgroundColor: **Colors**.lightBlue)

],

),

body: **Center**(

child: **StreamBuilder**(

*// untuk membuild widget setiap kali dia dapatkan dari stream*

stream: bloc.stateStream,

initialData: **Colors**.amber,

builder: (context, snapshot) {

return **AnimatedContainer**(

duration: **Duration**(milliseconds: 500),

width: 100,

height: 100,

color: snapshot.data,

);

},

))));

}

}

