**Flutter Programming**

Materi:

Penggunaan perangkat mobile seperti Android dan IOS yang semakin besar belakangan ini memaksa programmer untuk dapat mendevelop program untuk kedua perangkat tersebut. Mata kuliah ini akan menggunakan Flutter untuk mendevelop sebuah program ke Android dan IOS, dengan harapan mahasiswa tidak perlu membuat program milik mereka dua kali ke masing-masing OS. Pada awal kuliah mahasiswa akan dikenalkan dengan bahasa pemrograman Dart yang merupakan dasar dari Flutter. Berikutnya, akan dibahas mengenai flutter widget, themes, dan layout untuk membuat tampilan. Setelah tampilan siap, akan diajarkan mengenai forms, gestures, dan animation supaya terdapat interaksi yang interaktif dengan program buatan. Tidak hanya itu, mahasiswa juga akan mempelajari state management, async dart, pengolahan data JSON, penggunaan SQLite database, dan pemanggilan RESTful API dengan menggunakan Http, dengan harapan program tidak hanya tampilan saja, melainkan terdapat pengolahan data dari database. Pada akhirnya mahasiswa diharapkan dapat mendevelop aplikasi mobile mereka ke Android dan IOS.

Pertemuan:

1. **Perkenalan Flutter:** perkembangan Flutter & Dart; Arsitektur Flutter; InstalasiFlutter SDK; Instalasi IDE (Android Studio/Visual Studio Code); membuat aplikasisederhana dengan Flutter. Bahasa Pemrograman Dart: Variabel dan Tipe Data:Variabel; Tipe Data; Tipe Nullable dan Non-nulllable; Operator tipe data. ControlFlow dan Function: If statement; Switch statement; For dan while loop; Assertions;Anonymous Function; Parameter optional; Nested function; menggunakan typedefs. 5 sep 2022
2. **Classes:** Libraries dan visibility; Constructors; const; Getters dan setters; Operatoroverload; Cloning objects. Inheritance dan Exceptions: Inheritance; Method extension;Object class; Exceptions. Generics dan Collections: Generic types; Collections.Asynchronous Programming: Futures; Streams; Isolates. Prinsip pemrograman denganDart: Prinsip SOLID; Injeksi Dependency. 12 sep 2022
3. **Dasar dari Flutter:** Struktur dan tool; Widget dan State; Rebuilds dan Optimasi;Arsitektur. Membangun UI di Flutter: Material; Cupertino; Membangun Layout.

19 sep 2022

1. **Widgets:** Card2; rendering widgets; tipe-tipe widget. Scrollable Widgets: ListView;setting ExploreScreen; FutureBuilder; nested ListViews; GridView. 22 sep 2022
2. **Manajemen State:** Memperbarui UI; Passing state dengan Provider; Alternatif untuksetState: BLoC pattern. 26 sep 2022,
3. **Rute dan Navigasi:** Dasar dari navigasi dan route; Passing data antara halaman danwidget. Pengenalan Navigator: Navigator 1.0; Navigator 2.0; Navigation danunidirectional data flow; UI Flow; menggunakan app router. 28 sep 2022
4. **Menyimpan Data dengan SQLite:** menambahkan database ke project;menambahkan SQLite repository. Lokalisasi dan internasionalisasi; Internasionalisasimanual; Internasionalisasi menggunakan intl. Animasi: Animasi implisit; Libraryanimasi; Animasi custom. 25 oct 2022
5. **Bekerja dengan JSON dan format lainnya:** Parsing JSON; Parsing XML;Serialization. Shared Preferences: penyimpanan data dan plugin shared\_preferences. 3 oct 2022, 11 oct 2022
6. **Networking:** membuat request HTTP; bekerja dengan data; panggilan ke REST API(advanced). Menggunakan Chopper Library: request dan response; encoding dandecoding JSON; menggunakan interceptors; Generate Chopper file; Logging requestdan response. 12 oct 2022 , 17 oct 2022, 19 oct 2022
7. **Assets, images, dan multimedia:** Assets dan images; Bekerja dengan images; Grafikvector yang scalable; Audio dan Video dengan Flutter. Forms dan gestures: Forms danvalidasi; Gestures.
8. **Interaksi dengan device:** mengambil gambar; pemrograman dengan sensor;geolocation; platform-specific package.
9. **Widgets showcase:** Material; Cupertino; community Widget. 5 oct 2022
10. **Menggunakan Firebase dengan Flutter:** Instalasi; menggunakan Firestore sebagaibackend; monetisasi aplikasi anda dengan AdMob; Flutter ML Kit; push notificationdengan FCM; authentication dengan Firebase.
11. **Testing dan profiling aplikasi:** Testing aplikasi Flutter; testing performances. Builddan release app.

**Day 1 :**

<https://dartpad.dev/>?

1. **Basic fungsi**

void main() {

int maxNumber = 35;

double pi = 3.14;

String myName = "dart";

bool isDartCool = isDartCoolFunction();

print("hello word $maxNumber");

print("pi number $pi");

print("My name $myName");

print("hello $isDartCool");

if (isDartCool){

print("hello $isDartCool");

}else{

print("but i think dart is realy cool");

}

for (int i = 0; i < 5; i++) {

print('hello ${i + 1}');

}

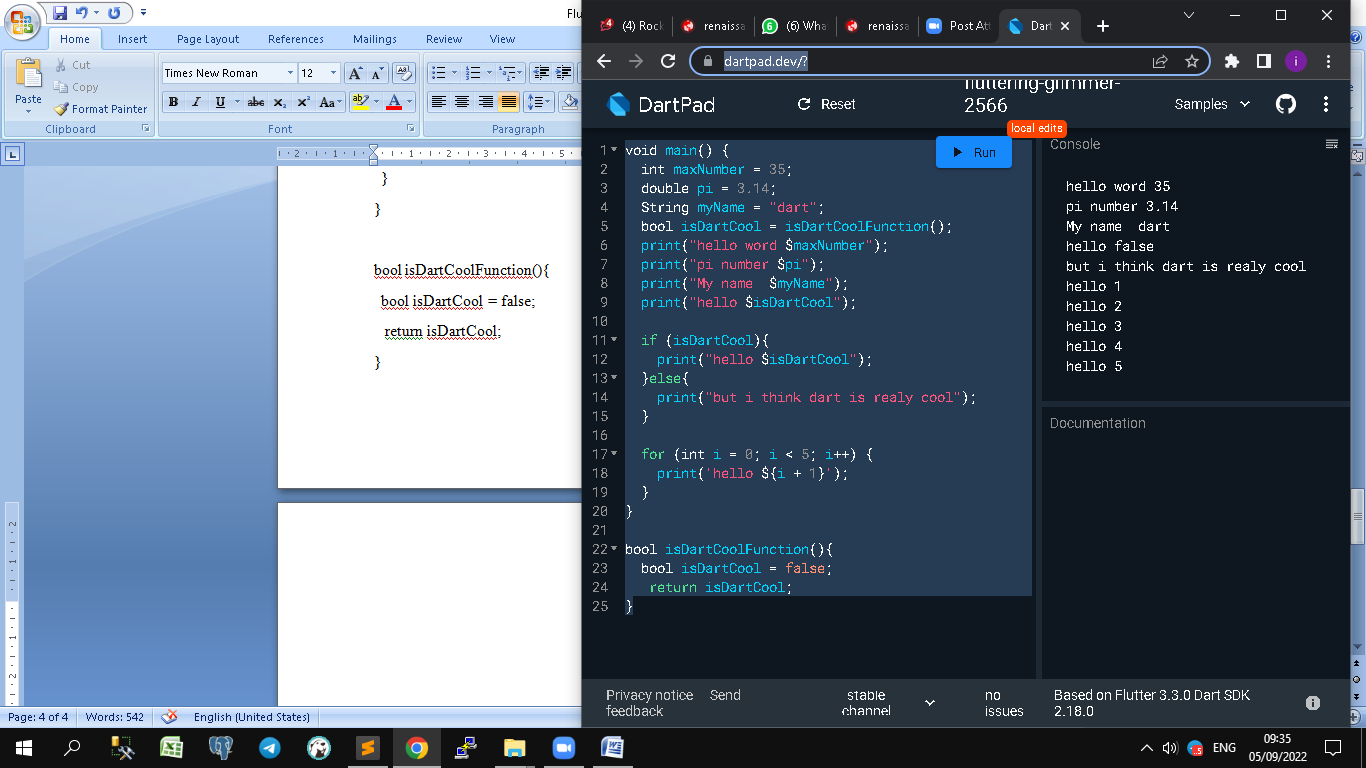
}

bool isDartCoolFunction(){

bool isDartCool = false;

return isDartCool;

}



2 type :

1. Function = non blocking
2. Oop

<https://dart.dev/guides/language/effective-dart#style>

**2. Basic konversi**

double kurs=0;

List<double> dataKurs=[10000,14000,3000];

Map dataKursByName = {'USD':'14000', 'SGD':'10000','JPY':'3500'};

void main() {

const num pi=3.14;

double standartKurs=14500;

//final num kurs;

var dana = 750;

if (dana>50){

// kurs=standartKurs ??13000; // ?? kalau null == 13000

kurs=standartKurs;

var hasilKonversi = konversiUang(dana,1);

pesan(errorCode:404,isi:'Pastikan dana diisi');

print('Hasil konversi $hasilKonversi');

pesan(isi:'Konversi sedang dilakukan',errorCode:400);

pesan(isi:'Pastikan dana diisi');

var status = hasilKonversi>100000 ? "hati-hati di jalan":"Hasil konversinya lumayan";

pesan(isi:status);

}else{

pesan(isi:"Tidak boleh konversi", errorCode:200);

}

}

// hierarki hasil = double , dari double \* integer

// double konversiUang(num dana, int indexKurs) => dana \* dataKurs[indexKurs];

double konversiUang(num dana, int indexKurs) => dana \* double.parse(dataKursByName["USD"]);

//optional parameter menggunakan [ ]

//optional parameter bisa ada isinya, bisa tidak ada

// null safety menyebabkan optional parameter harus diberi karakter ? atau berikan nilai default

// void pesan([String? isi, int? errorCode=0]) => print('Info: $isi. error code $errorCode');

// kayak map, asal ada namanya

// required wajib ada

void pesan({required String? isi, int? errorCode=0}) => print('Info: $isi. error code $errorCode');

**DAY 2**

**Program 1**

**class**

class Item{

//encapsulation -> data harus private atau protected

String \_nama="";

int \_stock=0;

int \_harga=0;

Item.Create1(this.\_nama,this.\_stock,this.\_harga);

Item.Create2(String n, int s, int h){

nama=n;

\_stock = s;

\_harga = h;

}

set nama(String value){

if (value.length>=5 && value.length<=100){

\_nama = value.toUpperCase();

}else{

print("Nama Item panjangnya 5-100");

\_nama="NONAME";

}

}

String get name => \_nama.substring(0,5);

void naikHarga(int kenaikan){

\_harga+=kenaikan;

}

double diskonHarga(double diskon){

return \_harga\*(1-diskon);

}

}

void main() {

var anItem = Item.Create1("Indomie",100,3500);

anItem.naikHarga(150); // 3650

print(anItem.diskonHarga(0.1));

anItem.nama="Coca";

print(anItem.\_nama);

var anItem2 = Item.Create2("Chitato",50,6500);

print(anItem2.nama);

}

**Program 2**

**inheritance**

// inheritance

class Snack extends Item{

// null safety check

// late = inisialisasi variable dulu , inputannya nanti telat

// misal dalam constructor

late String expireDate;

Snack(super.nama, super.stock, super.harga, String ex) {

expireDate = ex;

}

// super : call parent

// Snack(String n, int s, int h, String ex) : super(n,s,h){

// this.expireDate = ex;

// }

}

// abstract == item butuh structure aja

// interface == isi fungsi / contoh seperti mesin atm

abstract class Item{

//encapsulation -> data harus private atau protected

String \_nama="";

int \_stock=0;

int \_harga=0;

//name constructor NamaClass.NamaContructor

Item(this.\_nama,this.\_stock,this.\_harga);

Item.Create2(String n, int s, int h){

nama = n;

\_stock = s;

\_harga = h;

}

set nama(String value){

if (value.length>=5 && value.length<=100){

\_nama = value.toUpperCase();

}else{

print("Nama Item panjangnya 5-100");

\_nama="NONAME";

}

}

String get nama => \_nama.substring(0,5);

void naikHarga(int kenaikan){

\_harga+=kenaikan;

}

double diskonHarga(double diskon){

return \_harga\*(1-diskon);

}

}

void main() {

var anItem = Snack("Indomie",100,3500,'10/10/2022');

anItem.naikHarga(150); // 3650

print(anItem.diskonHarga(0.1));

// anItem.nama="Coca Cola";

print(anItem.nama);

print(anItem.expireDate);

// cascade operator untuk penggabungan perintah perintah

// var anItem2 = Item.Create2("Chitato",50,6500)

// ..nama='Chitato Ayam Panggang'..naikHarga(750);

// print(anItem2.nama);

}

**Program 3**

**OOP**

**abstract / interface**

class Perishable{ // interface biasanya able

void sync(){}

}

class Recyclable{

void execute(){}

}

enum Kategori{

kering,basah;

@override

String toString() => "Ini jenis makanan $name";

}

// inheritance

class Snack extends Item{

late String \_expireDate;

late var \_kategori;

Snack(super.nama, super.stock, super.harga, String ex){

\_expireDate=ex;

\_kategori=Kategori.basah;

}

// override == memastikan bahwa fungsi yang dipanggil induk

@override

void execute(){

print("execute versi snack");

}

}

// interface/implements== isi fungsi / contoh seperti mesin atm

class Item implements Perishable,Recyclable{

//encapsulation -> data harus private atau protected

String \_nama="";

int \_stock=0;

int \_harga=0;

//name constructor NamaClass.NamaContructor

Item(this.\_nama,this.\_stock,this.\_harga);

@override

void sync(){

//TODO IMPLEMENT HERE

}

@override

void execute(){

print("execute versi item");

}

String get nama => \_nama.substring(0,5);

void naikHarga(int kenaikan){

\_harga+=kenaikan;

}

double diskonHarga(double diskon){

return \_harga\*(1-diskon);

}

}

void main() {

var anItem = Snack("Indomie",100,3500,'10/10/2022');

anItem.naikHarga(150); // 3650

print(anItem.diskonHarga(0.1));

// anItem.nama="Coca Cola";

print(anItem.nama);

// print(anItem.expireDate);

anItem.execute();

}

**Day 3**

<https://dartpad.dev/>?

**1. Multiple inheritance / mixin**

// mixin == multiple inheritance

// pemanggilan otomatis sesuai urutan

mixin Flyable{

void fly(){

print('Animal is flying');

}

}

mixin Walkable{

void walk(){

print('Animal is walking');

}

}

mixin Swimmable{

void swin(){

print("Animal is swimming");

}

}

class Mammal{}

Class Bird {}

class Cat extends Mammal with Walkable{

}

class Duck extends Bird with Walkable, Flyable, Swimmable{

}

void main() {

for (int i = 0; i < 5; i++) {

print('hello ${i + 1}');

}

}

**2. Extension Methods**

//Extension Method

// mau nya urutan tidak berubah

// dart otomatis berubah

// maka dibuat method

// selection sort

extension Sorting on List<int>{

List<int> sortAsc(){

var list = [...this]; // create clone dari array

//menggunakan spread operator

// list of int yang akan disorting

var length = this.length;

for (int i = 0;i<length-1;i++){

int min = i;

for (int j=i+1;j<length;j++){

if (list[j]<list[min]){

min=j;

}

}

int tmp=list[min];

list[min]=list[i];

list[i]=tmp;

}

return list;

}

}

void main() {

var arrayNumbers = [10,3,5,6,4];

print(arrayNumbers);

var sortedNumbers = arrayNumbers.sortAsc();

print(sortedNumbers);

print(arrayNumbers);

}

**3. Functional Programming**

//fungtional Programming

// 1. hanya digunakan tanpa angka random di dalam nya

// 2. tidak ada konsep for / while

// 3. tidak boleh mengubah nilai variabel yang sudah diinisialisasi

// boleh nya create variabel baru untuk simpan data

// 4. function bisa dijadikan variabel parameter, bisa juga di-return

// 5.(higher-order functions) == fungsi yang mengambil fungsi lain

// 6. fungsi yang dibuat dalam fungsi lain == closure

int sum(int angka1, int angka2){

// var x = 4;

// var y = x+1;

return angka1+angka2;

}

int fibonacci(n){

if(n<=0){

return 0;

}

else if (n==1){

return 1;

}

else {

return fibonacci(n-1)+fibonacci(n-2);

}

}

// fungsi lambda dapat menyimpan hasil

//Return sebuah [angka] berdasarkan input yang ditambahkan dengan angka 1

Function olahAngka(angka){

var count = 1;

//variabel count berada dalam closure

return ()=>print("angka bernilai ${angka+count++}");

}

// class bisa simpan type data apapun

abstract class Cache<T>{

T getByKey(String key);

void setByKey(String key, T value);

}

void main() {

//print(sum(2,3));

//lambda function == fungsi tanpa nama

Function jumlah=(int x, int y){

return x+y;

};

Function printLambda=() =>print('Ini adalah lambda function');

printLambda(); // lambda function

print(jumlah(2,5));

// contoh (higher-order functions)

void myHigherOrderFunction(String pesan, Function myFunction){

print(pesan);

print(myFunction(2,7));

}

myHigherOrderFunction('Hai',jumlah);

myHigherOrderFunction('Konichiwa',(num1,num2)=>num1+num2);

var arrAngka = [0,1,1,2,3,5,8,13];

arrAngka.forEach((item){

item\*=5;

print(item);

});

print(arrAngka);

// 6.fungsi lambda dapat menyimpan hasil

var hasil = olahAngka(2);

hasil();

hasil();

}

**4.Future**

//Future, dengan 3 kondisi : uncompleted ,

// completed with data, completed with error

// asynchronous == non blocking

// synchronuous == blocking / nunggu antrian

void main() async { // async

// Future uncompleted

final myFuture = Future((){

print("Creating Future");

return 15;

});

print("main() complete");

// Style 1 : untuk dapat News

getNews().then((value){ // completed with data

print("Berita terbaru $value");

})

.catchError((error){ // completed with error

print("There was an error $error");

})

.whenComplete((){

// seperti finally, masuk ke code ini ketika ada error ataupun tidak

print("End of code");

});

;

print("Getting the news");

// Style 2 : untuk dapat News

//kombinasi dengan async await , jadi bisa tambahkan

// keyword asycn di main lalu await di getNews lalu gunakan try catch

try {

var news = await getNews();

print("News : $news");

}

catch(error){

print("Sorry : $error");

}

finally{

print("Getting News completed");

}

}

Future<String> getNews(){

return Future.delayed(Duration(seconds:3),() {

var callSuccess = true; // bisa di set false untuk testing completed with error

if (callSuccess){

return "Goods news";

}

else{

throw "Failed to get news";

}

});

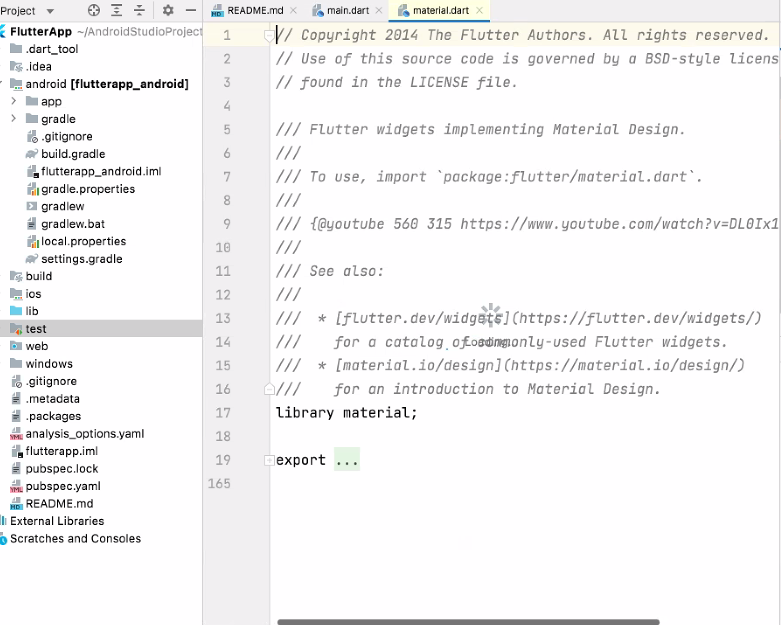
}

**6. Android Studio**

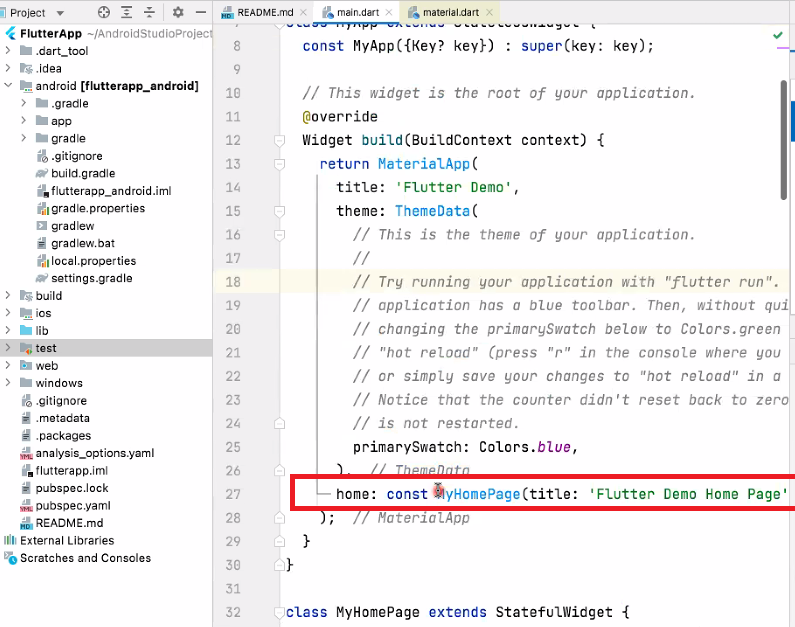
1. Buka Android studio
2. Pilih perangkat
3. Title



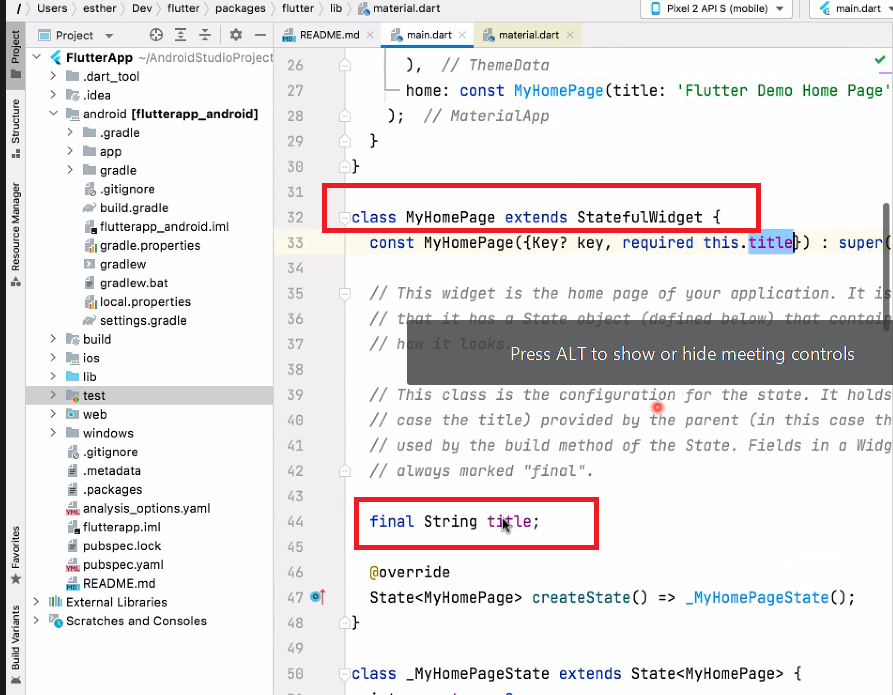
1. Material = design



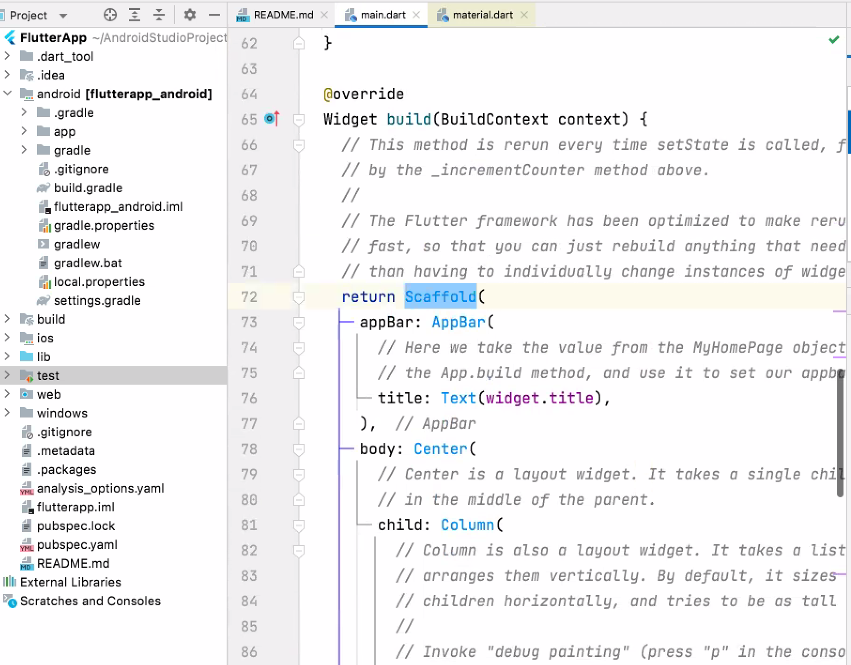
1. Home title



1. Class MyHomePage extends StatefullWidget

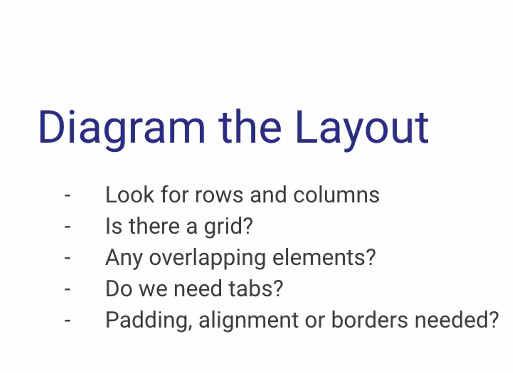


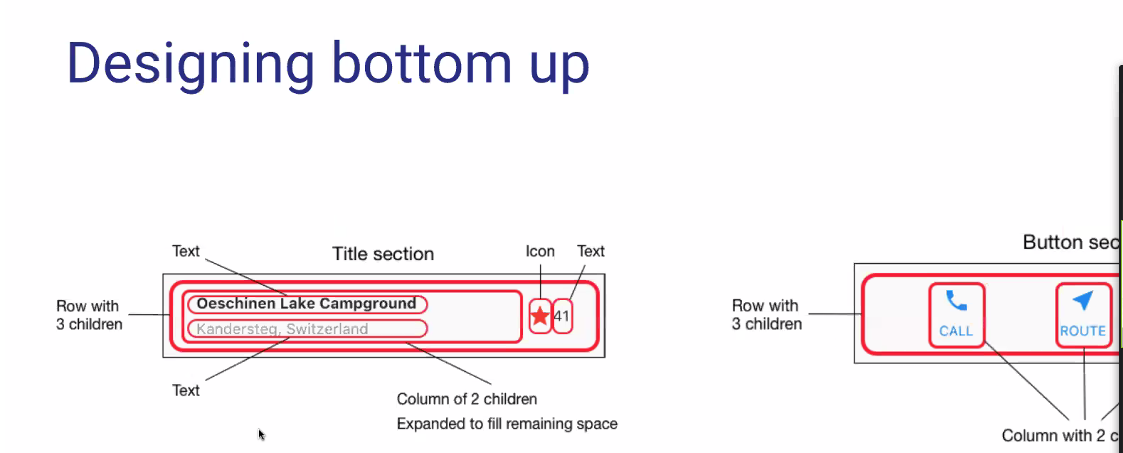
1. Scaffold : kerangka

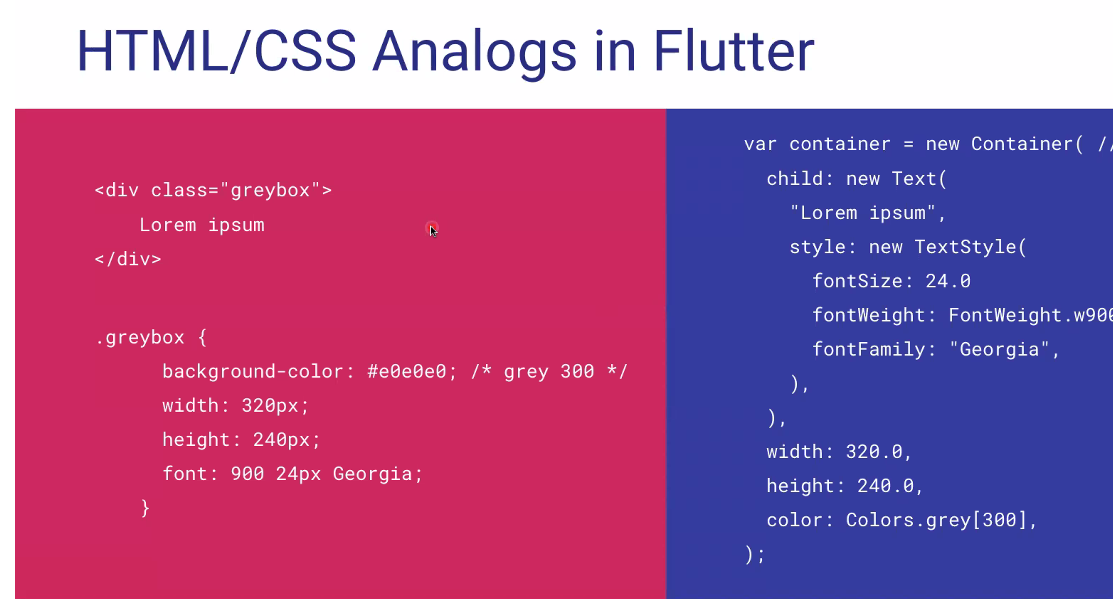


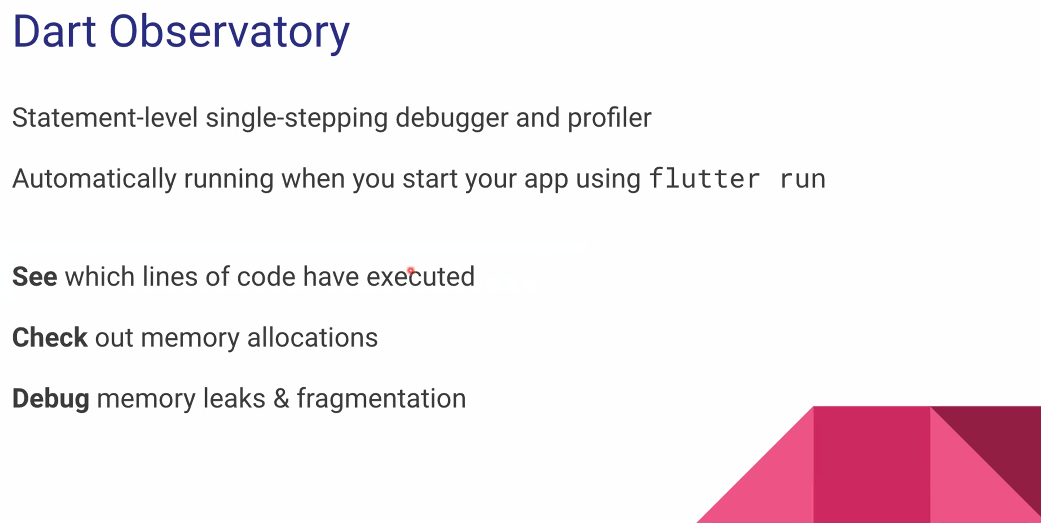
**Day 4**

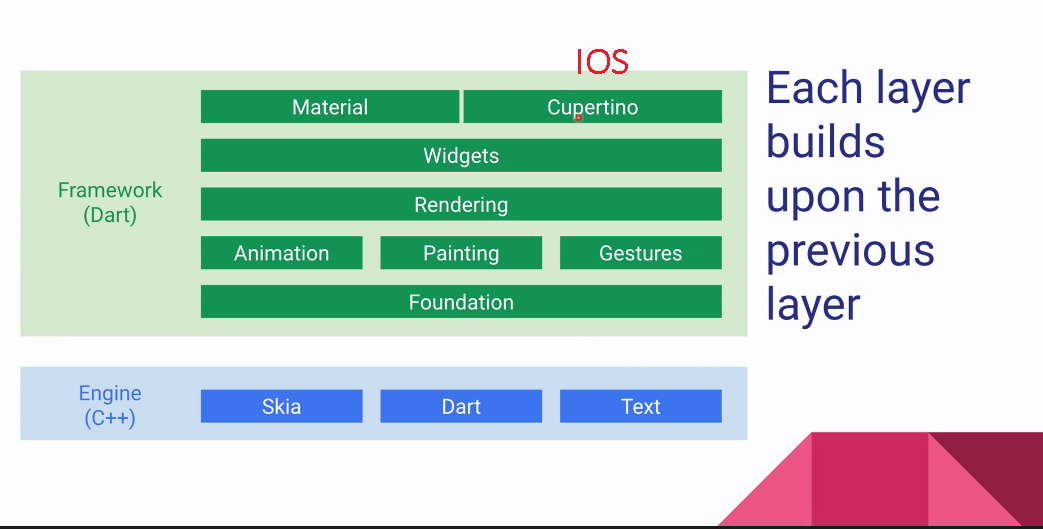
1. **Pengenalan Widgets**

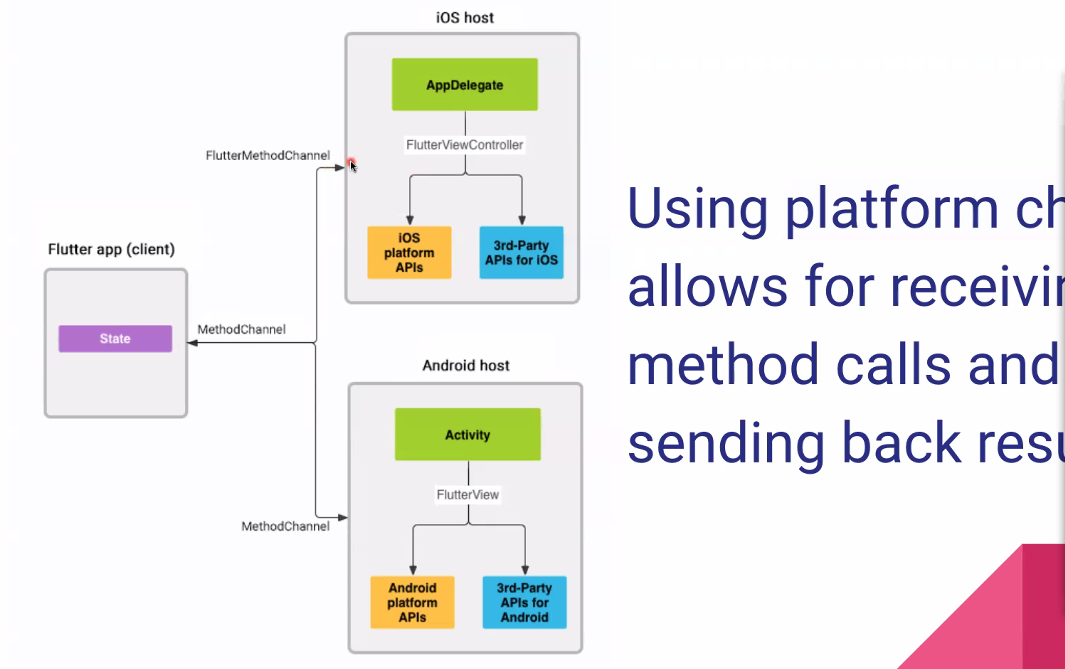
****

****

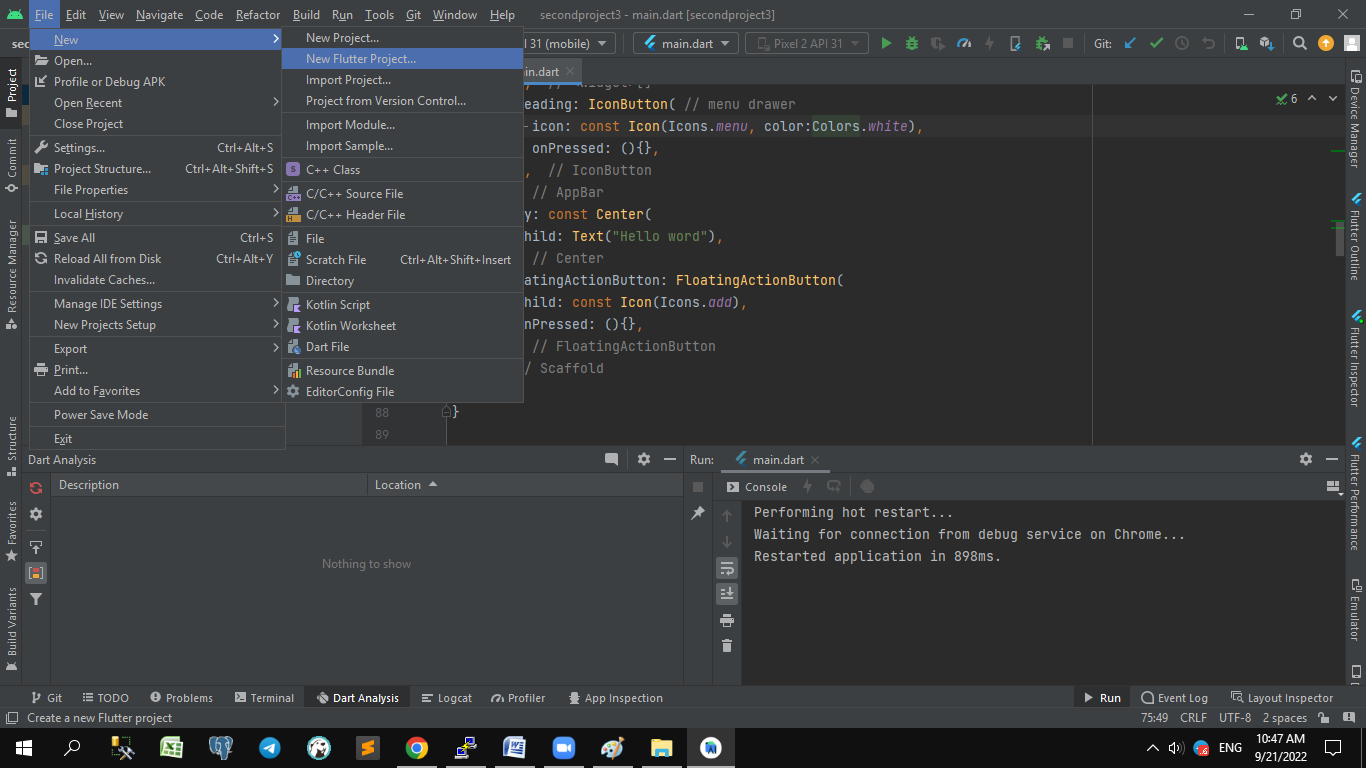
****

****

****

****

1. **Praktek di android studio**
   1. **Buat Project Flutter**

****

* 1. **Materi awal**

**Home**

// home : const Scaffold(

// body: Center(

// child: judul(

// text: "Hello word ",

// ),

// ),

// )

//

**Class Judul**

class Judul extends StatelessWidget{

final String text;

const Judul ({Key ? key, required this.text}): super (key :key );

@override

Widget build(BuildContext context){

return Text(

text,

style: const TextStyle(

fontSize: 24.0,

fontWeight: FontWeight.bold

),

);

}

}

* 1. **Materi 1 : BiggerText**

**Home**

// home : const Scaffold(

// body: Center(

// child: BiggerText(

// text: "Hello word ",

// ),

// ),

// )

//

**Class BiggerText**

// setiap ada kendala pakai ALT + ENTER

// recommendation

class BiggerText extends StatefulWidget {

final String text;

const BiggerText({Key ? key, required this.text}):super(key :key);

@override

State<StatefulWidget> createState() => \_BiggerTextState(); //

}

**Class BiggerTextState**

class \_BiggerTextState extends State<BiggerText>{

double \_textSize=16;

@override

Widget build(BuildContext context){

return Column(

// mainAxisAlignment: MainAxisAlignment.center,

mainAxisAlignment: MainAxisAlignment.spaceEvenly,

children: <Widget>[

Text(

widget.text, style: TextStyle(fontSize: \_textSize),

),

ElevatedButton(onPressed: (){

setState(() {

\_textSize=\_textSize+1;

});

}, child: const Text("Perbesar"))

],

);

}

}

* 1. **Materi 2 : FirstScreen**

home: const FirstScreen()

**Class FirstScreen**

class FirstScreen extends StatelessWidget{

const FirstScreen({Key ? key }): super(key:key);

@override

Widget build(BuildContext context){

return Scaffold(

appBar: AppBar(

title: const Text("Home Screen"),

actions: <Widget>[

IconButton(onPressed: (){}, icon: const Icon(

Icons.search,

color: Colors.white,

))

],

leading: IconButton( // menu drawer

icon: const Icon(Icons.menu, color:Colors.white),

onPressed: (){},

),

),

body: const Center(

child: Text("Hello word"),

),

floatingActionButton: FloatingActionButton(

child: const Icon(Icons.add),

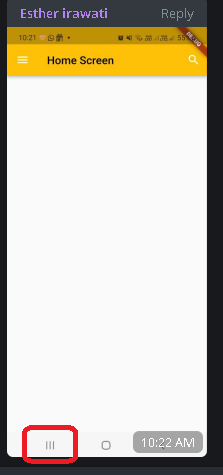
onPressed: (){},

),

);

}

}

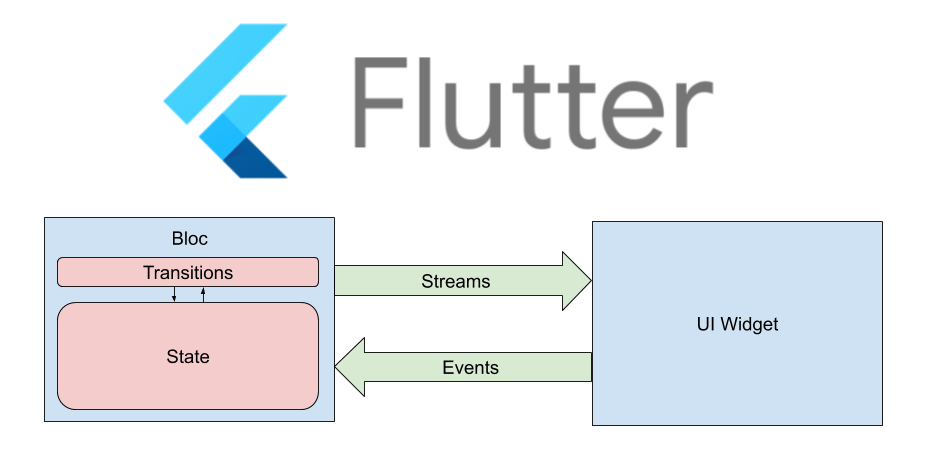


**Contoh Leading menu drawer**

**Day 5**

**1. Manajemen State**

BLoC pattern



**2.Praktek di android studio**

* **Main.dart**

import 'dart:html';

import 'package:flutter/material.dart';

import 'detail\_page.dart';

void main() {

runApp(const MyApp());

}

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

// This widget is the root of your application.

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

// This is the theme of your application.

//

// Try running your application with "flutter run". You'll see the

// application has a blue toolbar. Then, without quitting the app, try

// changing the primarySwatch below to Colors.green and then invoke

// "hot reload" (press "r" in the console where you ran "flutter run",

// or simply save your changes to "hot reload" in a Flutter IDE).

// Notice that the counter didn't reset back to zero; the application

// is not restarted.

primarySwatch: Colors.purple,

),

home: const DetailPage(),

);

}

}

* **Buat class baru**

import 'package:flutter/material.dart';

class DetailPage extends StatelessWidget {

const DetailPage({Key? key}): super(key:key);

@override

Widget build(BuildContext context){

// TODO : implemend build

return Scaffold(

body: SafeArea (

child: SingleChildScrollView(

child: Column (

crossAxisAlignment: CrossAxisAlignment.stretch,

children: <Widget>[

SizedBox(

height:150,

child:ListView(

scrollDirection: Axis.horizontal,

children: [

Image.asset('images/1.jpg'),

],

),

),

Container(

margin: const EdgeInsets.only(top: 16.0),

child: const Text(

'Monster Truck',

textAlign: TextAlign.center,

style: TextStyle(

fontSize: 30.0,

fontWeight: FontWeight.bold

),

),

),

Container(

margin: const EdgeInsets.symmetric(vertical: 16.0),

child: Row(

mainAxisAlignment: MainAxisAlignment.spaceEvenly,

children: <Widget>[

Column(

children: const<Widget>[

Icon(Icons.calendar\_today),

SizedBox(height: 8.0),

Text('18 Juni 2022'),

],

),

Column(

children: const<Widget>[

Icon(Icons.timeline),

SizedBox(height: 8.0),

Text('20:08-22:00'),

],

),

Column(

children: const<Widget>[

Icon(Icons.favorite),

SizedBox(height: 8.0),

Text('20'),

],

),

],

),

),

Container(

margin: const EdgeInsets.only(top: 16.0),

child: const Text(

"Monster Truck",

textAlign: TextAlign.center,

style: TextStyle(

fontSize: 16.0,

),

),

),

SizedBox(

height:150,

child:ListView(

scrollDirection: Axis.horizontal,

children: [

Image.network("https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQKHlksdJWoZVLzpINNNZ8K35qjBnR5I5kKEg&usqp=CAU"),

Image.network("https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRXujXLmRodcoyhHAgIJkC4QJuxa2AREDyChA&usqp=CAU"),

Image.network("https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTO3p-vlNw8WqcL578VwC59TKJxxxF16yxWAA&usqp=CAU"),

],

),

),

],

),

),

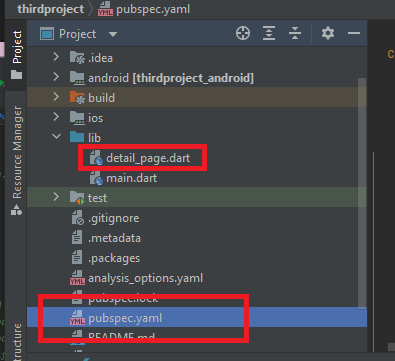
),

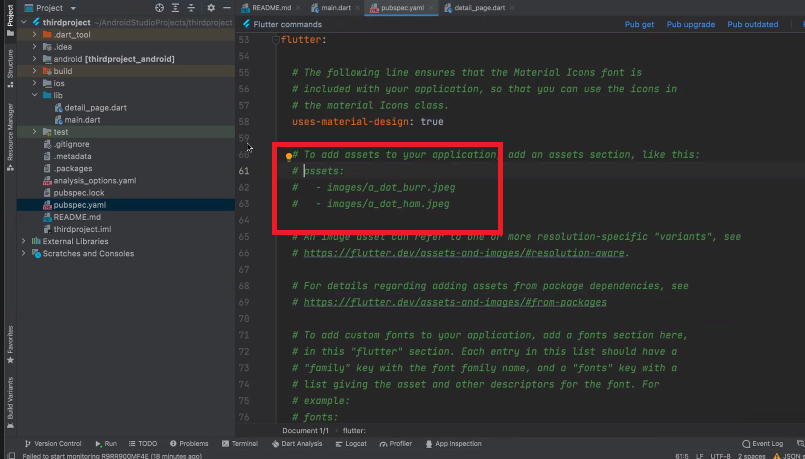
);

}

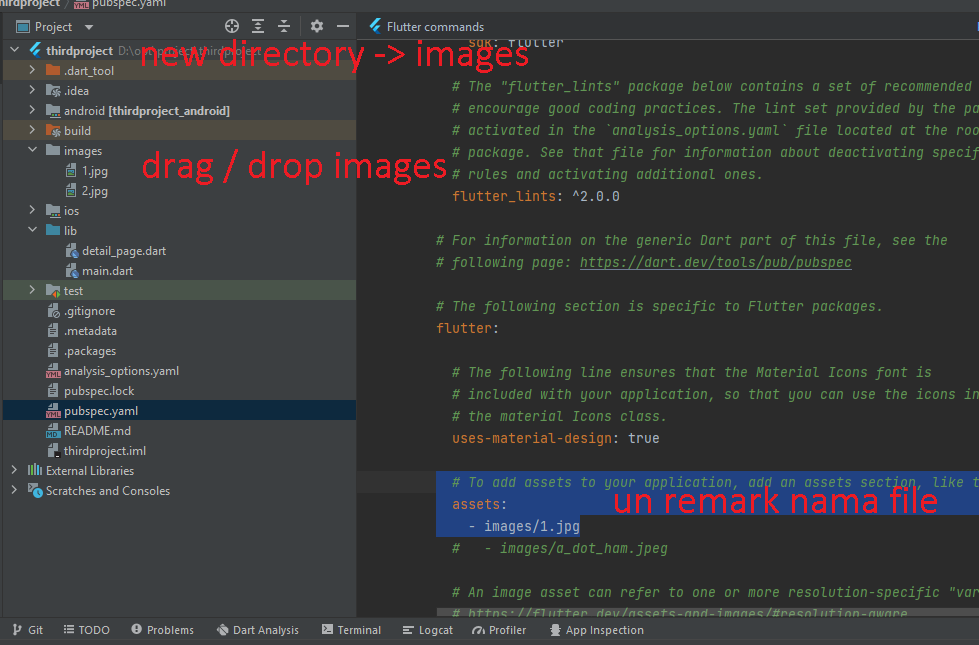
}

* **Tambah asset image**





*# To add assets to your application, add an assets section, like this:*assets:  
 - images/1.jpg



**Day 6**

**1. Rute dan Navigasi**

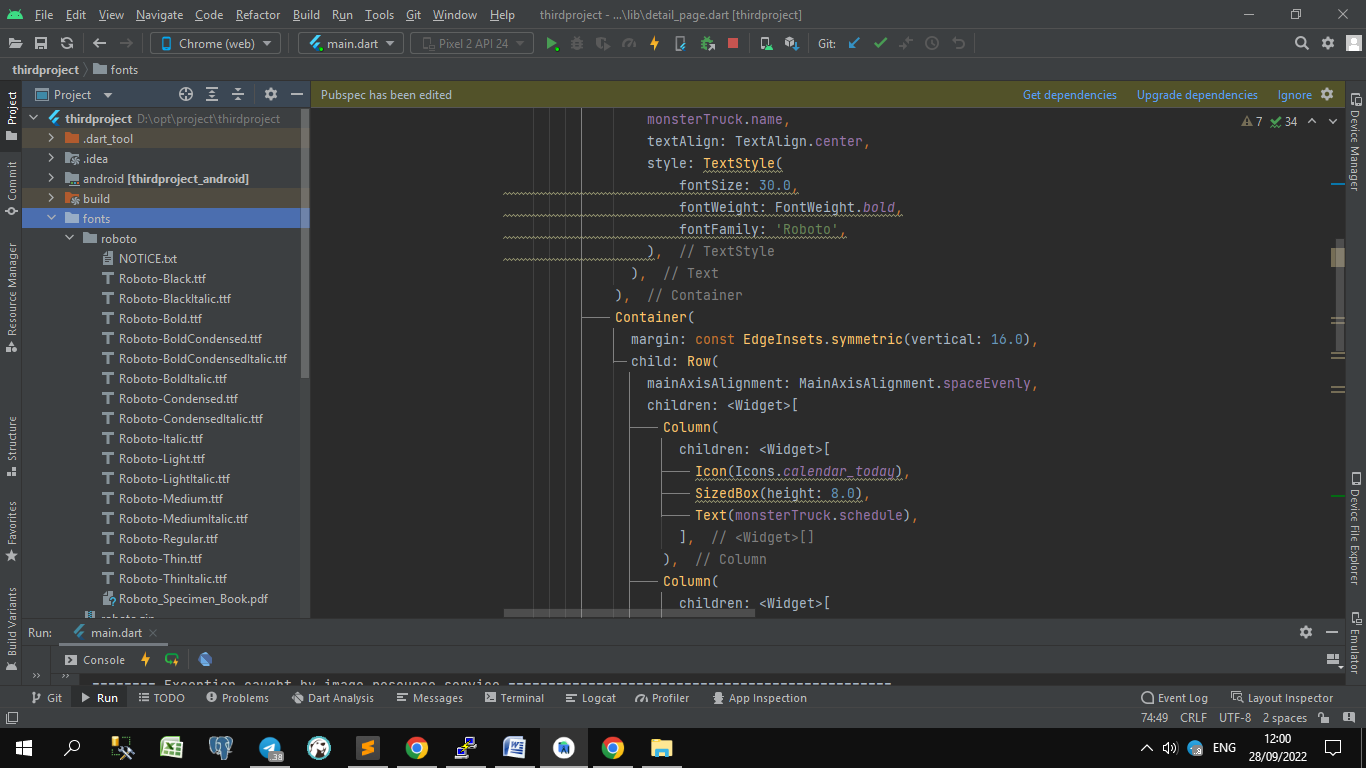
* **Tambah font**

Buat folder / klik kanan Project - new – directory – nama : fonts

Di <https://www.dafont.com/>

Download font yang sering dipakai : roboto

Taruh di folder project/fonts



Di file pubspec.yaml

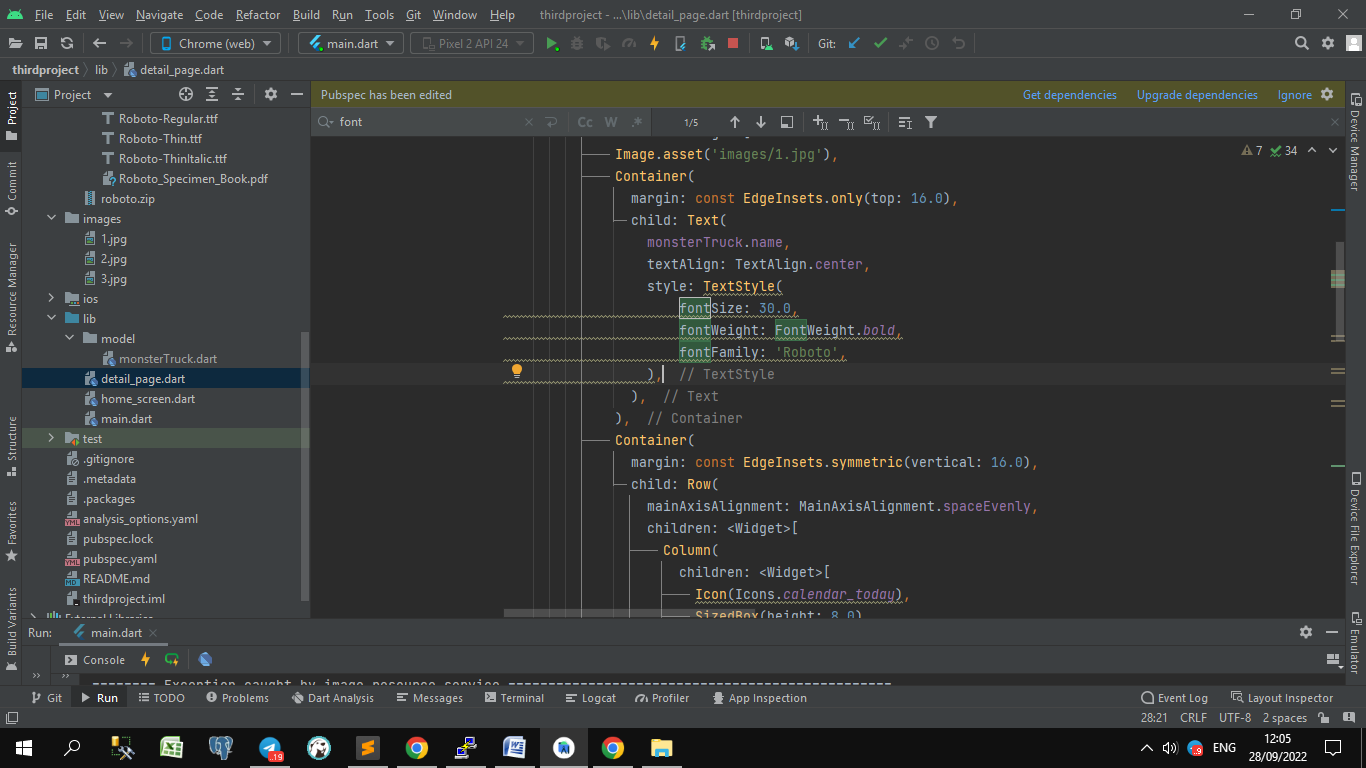
Buka fonts remark / perbaiki spasi nya 2x spasi



Contoh lain



Pemakaian:



* **Buat class baru HomeScreen**

Klik kanan lib – create dart file – nama class

import 'package:flutter/material.dart';

import 'package:thirdproject/model/monsterTruck.dart';

import 'detail\_page.dart';

class HomeScreen extends StatelessWidget{

const HomeScreen({Key ? key}):super(key:key);

@override

Widget build(BuildContext context) {

// TODO : implemend build

return Scaffold(

appBar: AppBar(

title: const Text('Truck List'),

),

//<https://api.flutter.dev/flutter/widgets/ListView/ListView.builder.html>

// Creates a scrollable, linear array of widgets that are created on demand

// parameter itemBuilder , itemCount dll

body: ListView.builder(

itemBuilder: (context, index){

final MonsterTruck monsterTruck = monsterTruckList[index];

//https://api.flutter.dev/flutter/material/InkWell-class.html

//A rectangular area of a [Material](https://api.flutter.dev/flutter/material/Material-class.html) that responds to touch.

return InkWell(

onTap: (){

// apps tumpuk2 / stack

// MaterialPageRoute: routing ke halaman lain

Navigator.push(context, MaterialPageRoute(builder: (context){

return DetailPage(monsterTruck:monsterTruck);

}));

},

child:Card(

child: Row(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

// <https://api.flutter.dev/flutter/widgets/Expanded-class.html>

// Using an [Expanded](https://api.flutter.dev/flutter/widgets/Expanded-class.html) widget makes a child of a [Row](https://api.flutter.dev/flutter/widgets/Row-class.html), [Column](https://api.flutter.dev/flutter/widgets/Column-class.html),

//or [Flex](https://api.flutter.dev/flutter/widgets/Flex-class.html) expand to fill the available space along the main axis

//(e.g., horizontally for a [Row](https://api.flutter.dev/flutter/widgets/Row-class.html) or vertically for a [Column](https://api.flutter.dev/flutter/widgets/Column-class.html)). If

//multiple children are expanded, the available space is divided

//among them according to the [flex](https://api.flutter.dev/flutter/widgets/Flexible/flex.html) factor.

Expanded(

flex:1,

// child:Image.asset('images/1.jpg'),

child:Image.asset(monsterTruck.poster),

),

Expanded(

flex:2,

child:Padding(

padding: const EdgeInsets.all(8.0),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

mainAxisSize: MainAxisSize.min,

children: <Widget>[

Text(monsterTruck.name,style: const TextStyle(fontSize: 16.0),),

const SizedBox(height: 10,),

Text(monsterTruck.broadcaster),

],

),

),

),

],

),

),

);

},

itemCount: monsterTruckList.length,

),

);

}

}

* **Buat class baru untuk isian/ model mapping**

Klik kanan lib

– create directory/package – nama : model

– create dart file – nama class : monsterTruck

class MonsterTruck{

String name;

String broadcaster;

int numEpisodes;

String schedule;

String poster;

List<String> imageUrls;

MonsterTruck({

required this.name,

required this.broadcaster,

required this.numEpisodes,

required this.schedule,

required this.poster,

required this.imageUrls,});

}

var monsterTruckList = [

MonsterTruck(

name:'Monster Truck',

broadcaster:'Netflix',

numEpisodes:20,

schedule:'20.00-21.00',

poster:'images/1.jpg',

imageUrls:[

'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQKHlksdJWoZVLzpINNNZ8K35qjBnR5I5kKEg&usqp=CAU'

,'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQKHlksdJWoZVLzpINNNZ8K35qjBnR5I5kKEg&usqp=CAU'

,'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQKHlksdJWoZVLzpINNNZ8K35qjBnR5I5kKEg&usqp=CAU'

]

),

MonsterTruck(

name:'Monster Truck',

broadcaster:'Netflix',

numEpisodes:20,

schedule:'20.00-21.00',

poster:'images/2.jpg',

imageUrls:['https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQKHlksdJWoZVLzpINNNZ8K35qjBnR5I5kKEg&usqp=CAU']

),

MonsterTruck(

name:'Monster Truck',

broadcaster:'Netflix',

numEpisodes:20,

schedule:'20.00-21.00',

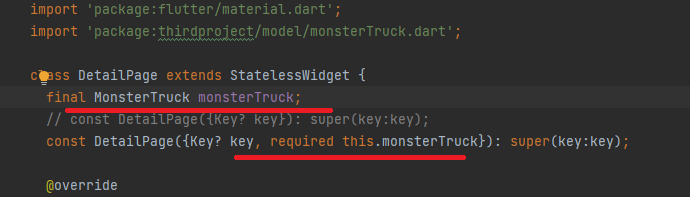
poster:'images/3.jpg',

imageUrls:['https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQKHlksdJWoZVLzpINNNZ8K35qjBnR5I5kKEg&usqp=CAU']

)

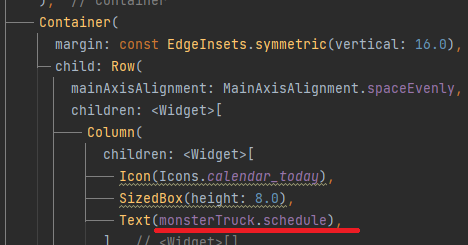
];

* **Buat akses class di constructor Class DetailPage**



* **Di Class DetailPage**

Ganti isi pakai list dalam class MonsterTruck



**Day 7**

**1. Bekerja dengan JSON dan format lainnya**

**news\_screen.dart : List news**

import 'package:flutter/material.dart';  
import 'package:thirdproject/detail\_news.dart';  
import 'package:thirdproject/model/article.dart';  
  
class NewsScreen extends StatelessWidget {  
 static const *routeName* = '/articles\_list';  
 const NewsScreen({Key? key}) : super(key: key);  
  
 @override  
 Widget build(BuildContext context) {  
 // *TODO : implemend build* return Scaffold(  
 appBar: AppBar(  
 title: const Text('News'),  
 ),  
 body: FutureBuilder<String>(  
 //DefaultAssetBundle == widget yang membaca data yang kita berikan  
 future: DefaultAssetBundle.*of*(context).loadString('assets/articles.json'),  
 builder: (context, snapshot) {  
 final List<Article> articles = parseArticles(snapshot.data);  
 return ListView.builder(  
 // shrinkWrap: true,  
 itemCount: articles.length,  
 itemBuilder: (context, index) {  
 return \_buildArticleItem(context, articles[index]);  
 });  
 },  
 ),  
 );  
 }  
}  
  
Widget \_buildArticleItem(BuildContext context, Article article) {  
 return ListTile(  
 contentPadding: const EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0),  
 leading: Image.network(  
 article.urlToImage,  
 width: 100,  
 ),  
 title: Text(article.title),  
 subtitle: Text(article.author),  
 onTap: () {  
 Navigator.*pushNamed*(context, DetailNewsScreen.*routeName*,arguments: article);  
 },  
 );  
}

**details\_news.dart : detail list news screen**

import 'package:flutter/material.dart';  
import 'package:thirdproject/model/article.dart';  
  
class DetailNewsScreen extends StatelessWidget{  
 static const *routeName* = '/articles\_detail';  
 final Article article;  
 const DetailNewsScreen({Key ? key, required this.article}):super(key:key);  
  
 @override  
 Widget build(BuildContext context) {  
 // *TODO : implemend build* return Scaffold(  
 appBar: AppBar(  
 title: const Text('News'),  
 ),  
 body: SingleChildScrollView(  
 child: Column(  
 children: [  
 Image.network(article.urlToImage),  
 Padding(  
 padding: const EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0),  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Text(article.description),  
 Divider(color: Colors.*grey*,),  
 Text(  
 article.title,  
 style: const TextStyle(  
 color: Colors.*black*,  
 fontWeight: FontWeight.*w100*,  
 fontSize: 24  
 ),  
 ),  
 const Divider(color: Colors.*grey*,),  
 Text(article.content),  
 Text('Date: ${article.publishedAt}', ),  
 const SizedBox(height: 10),  
 ElevatedButton(onPressed: (){}, child: const Text('More ...')),  
 ],  
 )  
 )  
 ],  
 ),  
 )  
 );  
 }  
}

**model/article.dart : class article – buat ambil data dari JSON**

import 'dart:convert';  
  
class Article{  
 final String author;  
 final String title;  
 final String description;  
 final String url;  
 final String urlToImage;  
 final String publishedAt;  
 final String content;  
  
 Article({  
 required this.author,  
 required this.title,  
 required this.description,  
 required this.url,  
 required this.urlToImage,  
 required this.publishedAt,  
 required this.content,  
 });  
 factory Article.fromJson(Map<String,dynamic> article)=>Article(  
 author: article['author'],  
 title: article['title'],  
 description: article['description'],  
 url: article['url'],  
 urlToImage: article['urlToImage'],  
 publishedAt: article['publishedAt'],  
 content: article['content']  
 );  
}  
  
List<Article> parseArticles(String? json){  
 if (json==null){  
 return [];  
 }  
 final List parsed=jsonDecode(json);  
 return parsed.map((json)=>Article.fromJson(json)).toList();  
}

**main.dart : class main –**

import 'dart:html';  
import 'package:flutter/material.dart';  
import 'package:thirdproject/detail\_news.dart';  
import 'package:thirdproject/model/article.dart';  
// import 'detail\_page.dart';  
import 'home\_screen.dart';  
import 'news\_screen.dart';  
  
void main() {  
 runApp(const MyApp());  
}  
  
class MyApp extends StatelessWidget {  
 const MyApp({Key? key}) : super(key: key);  
  
 // This widget is the root of your application.  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title: 'Flutter Demo',  
 theme: ThemeData(  
 primarySwatch: Colors.*purple*,  
 // dp : density pixel menyesuaikan  
 visualDensity: VisualDensity.*adaptivePlatformDensity*,  
 ),

//The name of the first route to show, if a Navigator is built.  
 initialRoute: NewsScreen.*routeName*,

// {Map routes = const {}}  
 routes: {  
 NewsScreen.*routeName*: (context)=>const NewsScreen(),  
 DetailNewsScreen.*routeName* : (context) => DetailNewsScreen(  
 article: ModalRoute.*of*(context)?.settings.arguments as Article,  
 ),  
 }  
 );  
 }  
}

**Day 8**

Bahan Materi :

Cari packages / dependencies <https://pub.dev/packages?q=webview>

Video <https://www.youtube.com/watch?v=XawP1i314WM&list=PLjxrf2q8roU23XGwz3Km7sQZFTdB996iG>

Contoh <https://gallery.flutter.dev/#/>

cookbook <https://docs.flutter.dev/cookbook>

showcase <https://flutter.dev/showcase>

design <https://material.io/design/material-studies>

Contoh PR :

<https://codelabs.developers.google.com/codelabs/mdc-103-flutter/#0>

<https://codelabs.developers.google.com/codelabs/flutter-cupertino?hl=en#0>

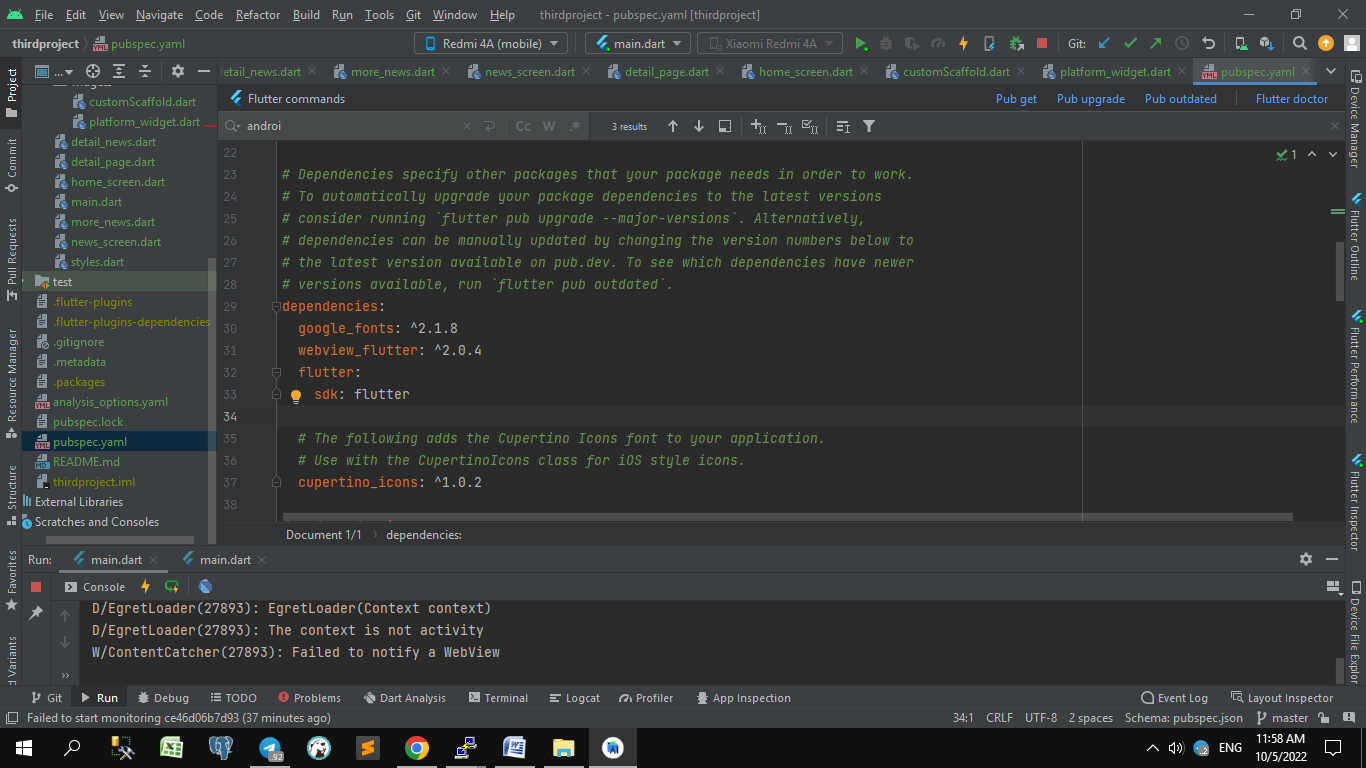
**1.Widgets showcase**

**Webview**

// menampilkan URL : dari web

// tambahkan dependencies ke pubspecs.yaml

// dependencies : webview\_flutter: ^2.0.4

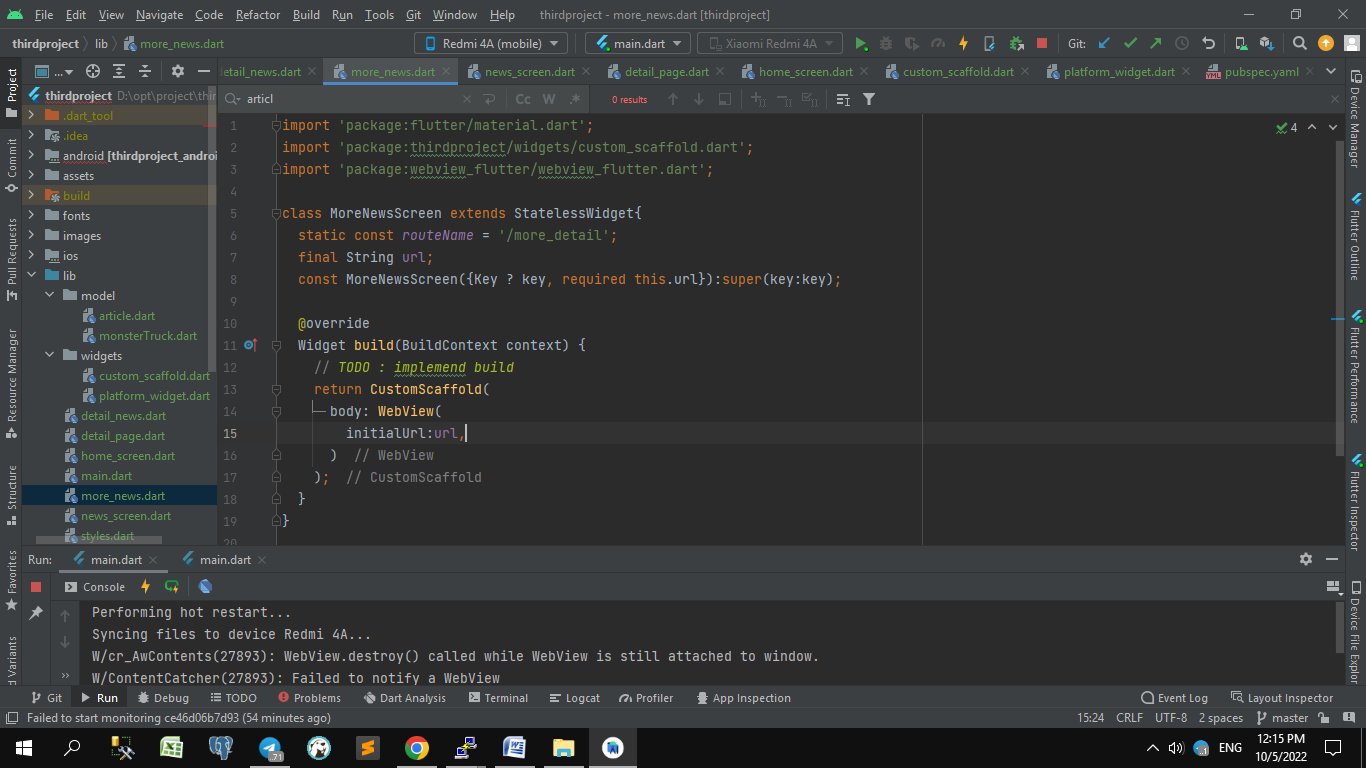


**File : more\_news.dart**

import 'package:flutter/material.dart';  
import 'package:thirdproject/widgets/customScaffold.dart';  
import 'package:webview\_flutter/webview\_flutter.dart';  
  
class MoreNewsScreen extends StatelessWidget{  
 static const *routeName* = '/more\_detail';  
 final String url;  
 const MoreNewsScreen({Key ? key, required this.url}):super(key:key);  
  
 @override  
 Widget build(BuildContext context) {  
 // *TODO : implemend build* return CustomScaffold(  
 body: WebView(  
 initialUrl:url,  
 )  
 );  
 }  
}

**Custom scaffold**

// menggantikan fungsi scaffold : u/ dimodifikasi sesuai kebutuhan



**File : custom\_scaffold.dart**

import 'package:flutter/material.dart';  
import 'package:webview\_flutter/webview\_flutter.dart';  
  
//biar poni tidak tertutup component kita

// tanpa context  
  
class CustomScaffold extends StatelessWidget{  
 final Widget body;  
 const CustomScaffold({Key ? key, required this.body}):super(key:key);  
  
 @override  
 Widget build(BuildContext context) {  
 // *TODO : implemend build* return Scaffold(  
 appBar: AppBar(  
 title: Text("More News Detail"),  
 ),  
 body: SafeArea(  
 child: Stack(  
 children: [  
 body,  
 \_buildShortAppBar(context) ,  
 ],  
 ),  
 )  
 );  
 }  
}  
  
Widget \_buildShortAppBar(BuildContext context){  
 return Card(  
 margin :const EdgeInsets.all(0),  
 shape: const BeveledRectangleBorder(  
 borderRadius: BorderRadius.only(  
 bottomRight: Radius.circular(16.0),  
 )  
 ),  
 child: Row(  
 mainAxisSize: MainAxisSize.min,  
 children: [  
 IconButton(  
 icon: Icon(Icons.*arrow\_back*),  
 onPressed: (){  
 Navigator.*pop*(context);  
 },  
 ),  
 Padding(  
 padding: const EdgeInsets.only(right:16.0),  
 child : Text(  
 'N',  
 style: Theme.*of*(context).textTheme.headline6,  
 )  
 )  
 ],  
 ),  
 );  
}

**Platform Widget**

// mendeteksi Device: android / iOS

**File : platform\_widget.dart**

import 'package:flutter/foundation.dart';  
import 'package:flutter/material.dart';  
import 'package:webview\_flutter/webview\_flutter.dart';  
  
//biar poni tidak tertutup component kita

// tanpa context  
class PlatformWidget extends StatelessWidget{  
 final WidgetBuilder androidBuilder;  
 final WidgetBuilder iosBuilder;  
 const PlatformWidget({Key ? key, required this.androidBuilder, required this.iosBuilder}):super(key:key);  
  
 @override  
 Widget build(BuildContext context) {  
 // *TODO : implemend build* switch (defaultTargetPlatform){  
 case TargetPlatform.android:  
 return androidBuilder(context);  
 case TargetPlatform.iOS:  
 return iosBuilder(context);  
 default:  
 return androidBuilder(context);  
 }  
 }  
}

//pemakaian

**File : news\_screen.dart**

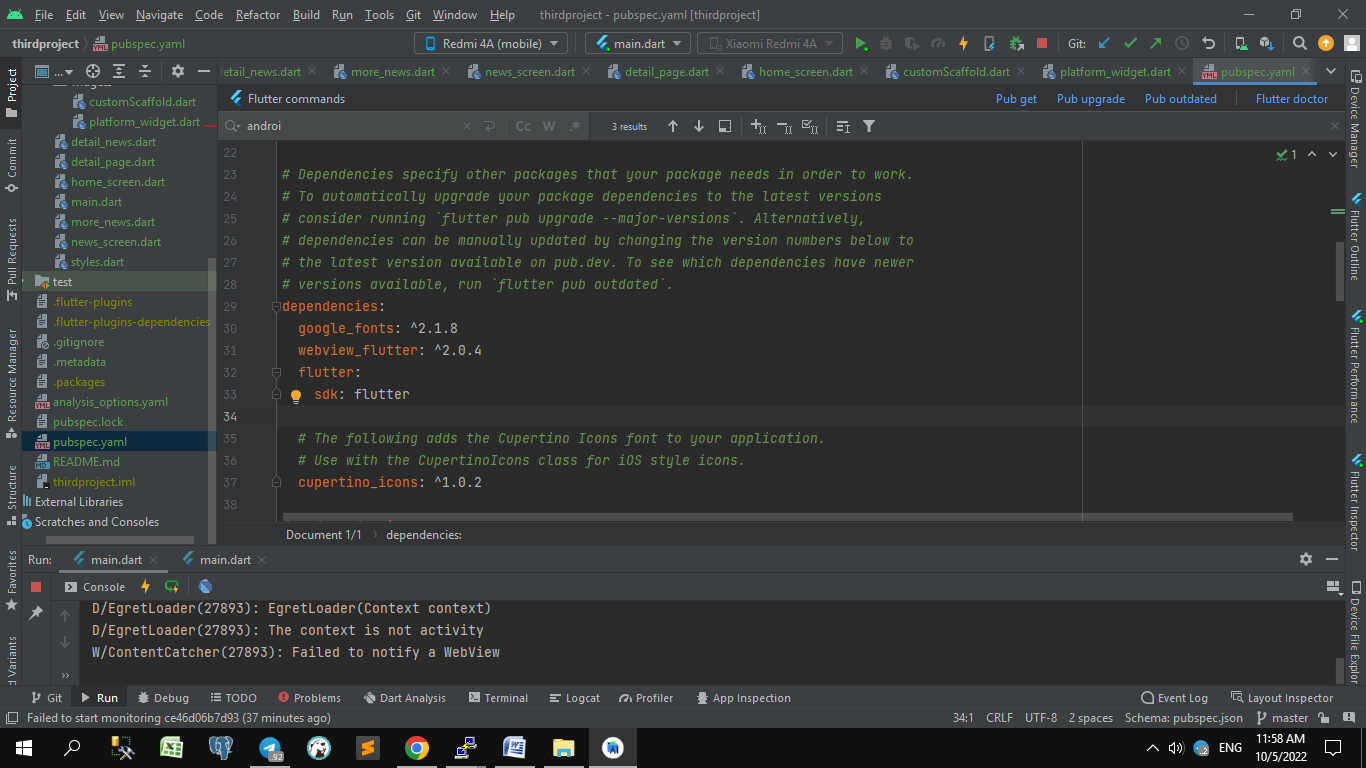
import 'package:flutter/cupertino.dart';  
import 'package:flutter/material.dart';  
import 'package:thirdproject/detail\_news.dart';  
import 'package:thirdproject/model/article.dart';  
import 'package:thirdproject/widgets/platform\_widget.dart';  
  
class NewsScreen extends StatelessWidget {  
 static const *routeName* = '/articles\_list';  
  
 const NewsScreen({Key? key}) : super(key: key);  
  
 @override  
 Widget build(BuildContext context) {  
 // *TODO : implemend build* // return \_buildAndroid(context);  
 return PlatformWidget(  
 androidBuilder: \_buildAndroid,  
 iosBuilder: \_buildIos  
 );  
 }  
}  
  
// dipisah method u/ dipanggil di android / ios  
FutureBuilder<String> buildFutureBuilder(BuildContext context) {  
 return FutureBuilder<String>(  
 //DefaultAssetBundle == widget yang membaca data yang kita berikan  
 future: DefaultAssetBundle.*of*(context).loadString('assets/articles.json'),  
 builder: (context, snapshot) {  
 final List<Article> articles = parseArticles(snapshot.data);  
 return ListView.builder(  
 // shrinkWrap: true,  
 itemCount: articles.length,  
 itemBuilder: (context, index) {  
 return \_buildArticleItem(context, articles[index]);  
 });  
 },  
 );  
}  
  
// platform android  
Widget \_buildAndroid(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: const Text('News'),  
 ),  
 body: buildFutureBuilder(context),  
 );  
}  
  
  
// platform iOS  
Widget \_buildIos(BuildContext context) {  
 return CupertinoPageScaffold(  
 navigationBar: const CupertinoNavigationBar(  
 middle: const Text('News'),  
 transitionBetweenRoutes: false,  
 ),  
 child: buildFutureBuilder(context),  
 );  
}  
  
Widget \_buildArticleItem(BuildContext context, Article article) {  
 return Material(  
 child: ListTile(  
 contentPadding:  
 const EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0),  
 leading: Image.network(  
 article.urlToImage,  
 width: 100,  
 ),  
 title: Text(article.title),  
 subtitle: Text(article.author),  
 onTap: () {  
 Navigator.*pushNamed*(context, DetailNewsScreen.*routeName*,  
 arguments: article);  
 },  
 ),  
 );  
}

**Ganti Theme / Material.io**

// ganti font / color dll

// tambahkan dependencies ke pubspecs.yaml

// dependencies : google\_fonts: ^2.1.8



**File : styles.dart**

import 'package:flutter/material.dart';  
import 'package:google\_fonts/google\_fonts.dart';  
  
const Color primaryColor = Color(0xFFFFFFFF);  
const Color secondaryColor = Color(0xFF6B38FB);  
  
final TextTheme myTextTheme = TextTheme(  
headline1: GoogleFonts.*merriweather*(  
fontSize: 92, fontWeight: FontWeight.*w300*, letterSpacing: -1.5),  
headline2: GoogleFonts.*merriweather*(  
fontSize: 57, fontWeight: FontWeight.*w300*, letterSpacing: -0.5),  
headline3:  
GoogleFonts.*merriweather*(fontSize: 46, fontWeight: FontWeight.*w400*),  
headline4: GoogleFonts.*merriweather*(  
fontSize: 32, fontWeight: FontWeight.*w400*, letterSpacing: 0.25),  
headline5:  
GoogleFonts.*merriweather*(fontSize: 23, fontWeight: FontWeight.*w400*),  
headline6: GoogleFonts.*merriweather*(  
fontSize: 19, fontWeight: FontWeight.*w500*, letterSpacing: 0.15),  
 subtitle1: GoogleFonts.*merriweather*(  
 fontSize: 15, fontWeight: FontWeight.*w400*, letterSpacing: 0.15),  
 subtitle2: GoogleFonts.*merriweather*(  
 fontSize: 13, fontWeight: FontWeight.*w500*, letterSpacing: 0.1),  
 bodyText1: GoogleFonts.*libreFranklin*(  
 fontSize: 16, fontWeight: FontWeight.*w400*, letterSpacing: 0.5),  
 bodyText2: GoogleFonts.*libreFranklin*(  
 fontSize: 14, fontWeight: FontWeight.*w400*, letterSpacing: 0.25),  
 button: GoogleFonts.*libreFranklin*(  
 fontSize: 14, fontWeight: FontWeight.*w500*, letterSpacing: 1.25),  
 caption: GoogleFonts.*libreFranklin*(  
 fontSize: 12, fontWeight: FontWeight.*w400*, letterSpacing: 0.4),  
 overline: GoogleFonts.*libreFranklin*(  
 fontSize: 10, fontWeight: FontWeight.*w400*, letterSpacing: 1.5),  
);

//pemakaian

**File : main.dart**

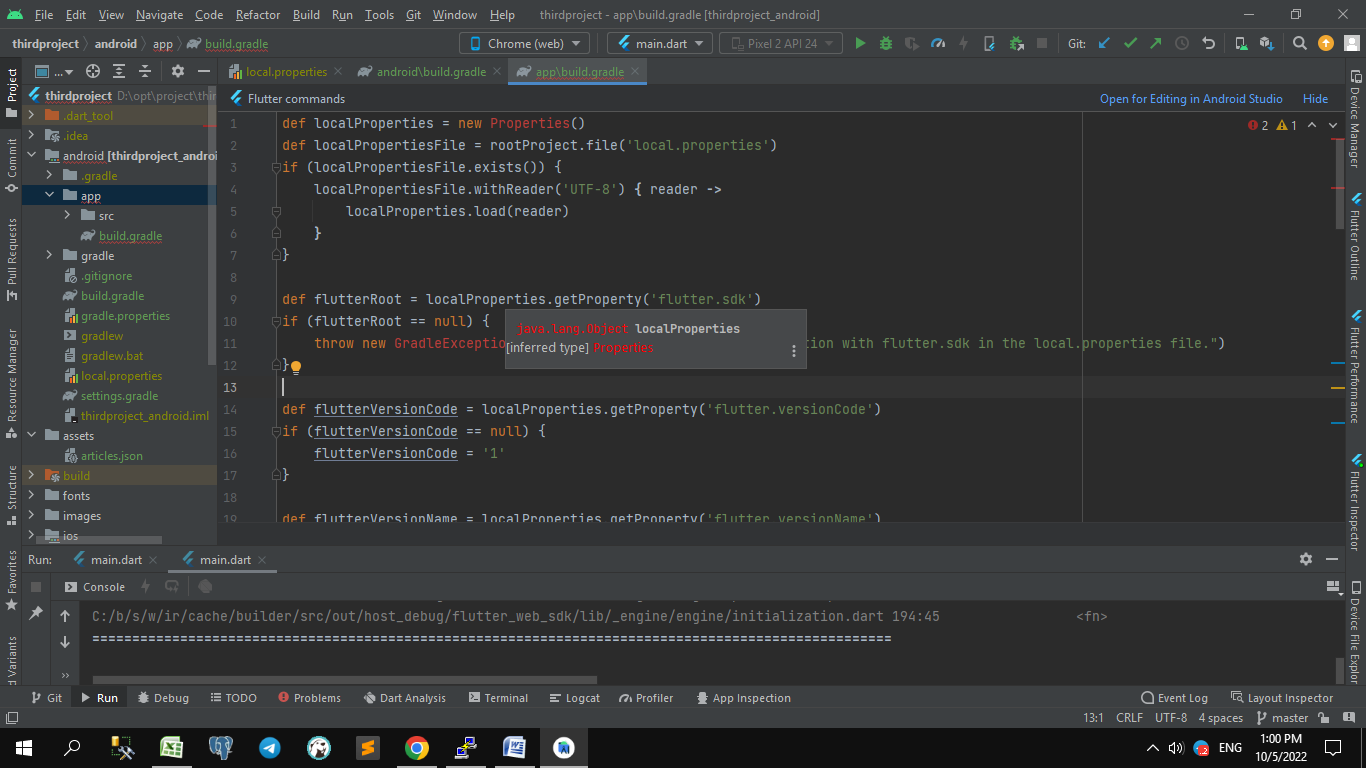
// ganti default theme jadi styles.dart

import 'package:flutter/material.dart';  
import 'package:thirdproject/detail\_news.dart';  
import 'package:thirdproject/more\_news.dart';  
import 'package:thirdproject/news\_screen.dart';  
import 'package:thirdproject/model/article.dart';  
import 'package:thirdproject/styles.dart';  
  
void main() {  
 runApp(const MyApp());  
}  
class MyApp extends StatelessWidget {  
 const MyApp({Key? key}) : super(key: key);  
  
 // This widget is the root of your application.  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title: 'Flutter Demo',  
 theme: ThemeData(  
 colorScheme: Theme.*of*(context).colorScheme.copyWith(  
 primary: primaryColor,  
 onPrimary: Colors.*black*,  
 secondary:secondaryColor,  
 ),  
 // This is the theme of your application.  
 // primarySwatch: Colors.purple,  
 // dp : density pixel menyesuaikan  
 visualDensity: VisualDensity.*adaptivePlatformDensity*,  
 textTheme: myTextTheme,  
 appBarTheme: const AppBarTheme(elevation: 8),  
 elevatedButtonTheme: ElevatedButtonThemeData(  
 style: ElevatedButton.*styleFrom*(  
 primary: secondaryColor,  
 onPrimary: Colors.*white*,  
 textStyle: const TextStyle(),  
 shape: const RoundedRectangleBorder(  
 borderRadius: BorderRadius.all(Radius.circular(0))  
 )  
 )  
 )  
 ),  
 initialRoute: NewsScreen.*routeName*,  
 routes: {  
 NewsScreen.*routeName*: (context)=>const NewsScreen(),  
 DetailNewsScreen.*routeName* : (context) => DetailNewsScreen(  
 article: ModalRoute.*of*(context)?.settings.arguments as Article,  
 ),  
 MoreNewsScreen.*routeName* : (context) => MoreNewsScreen(  
 url : ModalRoute.*of*(context)?.settings.arguments as String  
 ),  
 }  
 );  
 }  
}

**Jalankan ke device android / setting version android**

Ganti di android – app – build.gradle

flutter.minSdkVersion=19  
flutter.targetSdkVersion=31  
flutter.compileSdkVersion=android-31



**File : build.gradle**

def localProperties = new Properties()  
def localPropertiesFile = rootProject.file('local.properties')  
if (localPropertiesFile.exists()) {  
 localPropertiesFile.withReader('UTF-8') **{** reader **->** localProperties.load(reader)  
 **}**}  
  
def flutterRoot = localProperties.getProperty('flutter.sdk')  
if (flutterRoot == null) {  
 throw new GradleException("Flutter SDK not found. Define location with flutter.sdk in the local.properties file.")  
}  
  
def flutterVersionCode = localProperties.getProperty('flutter.versionCode')  
if (flutterVersionCode == null) {  
 flutterVersionCode = '1'  
}  
  
def flutterVersionName = localProperties.getProperty('flutter.versionName')  
if (flutterVersionName == null) {  
 flutterVersionName = '1.0'  
}  
  
apply plugin: 'com.android.application'  
apply plugin: 'kotlin-android'  
apply from: "$flutterRoot/packages/flutter\_tools/gradle/flutter.gradle"  
  
android **{** compileSdkVersion 32 //localProperties.getProperty('flutter.compileSdkVersion') //flutter.compileSdkVersion  
 ndkVersion flutter.ndkVersion  
  
 compileOptions **{** sourceCompatibility JavaVersion.VERSION\_1\_8  
 targetCompatibility JavaVersion.VERSION\_1\_8  
 **}** kotlinOptions **{** jvmTarget = '1.8'  
 **}** sourceSets **{** main.java.srcDirs += 'src/main/kotlin'  
 **}** defaultConfig **{** // *TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).* applicationId "id.co.suryabuanagroup.thirdproject"  
 // You can update the following values to match your application needs.  
 // For more information, see: https://docs.flutter.dev/deployment/android#reviewing-the-build-configuration.  
 minSdkVersion 19 //localProperties.getProperty('flutter.minSdkVersion')//flutter.minSdkVersion  
 targetSdkVersion 32 //localProperties.getProperty('flutter.targetSdkVersion')//flutter.targetSdkVersion  
 versionCode flutterVersionCode.toInteger()  
 versionName flutterVersionName  
 **}** buildTypes **{** release **{** // *TODO: Add your own signing config for the release build.* // Signing with the debug keys for now, so `flutter run --release` works.  
 signingConfig signingConfigs.debug  
 **}  
 }  
}**flutter **{** source '../..'  
**}**dependencies **{** implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin\_version"  
**}**

**Day 9**

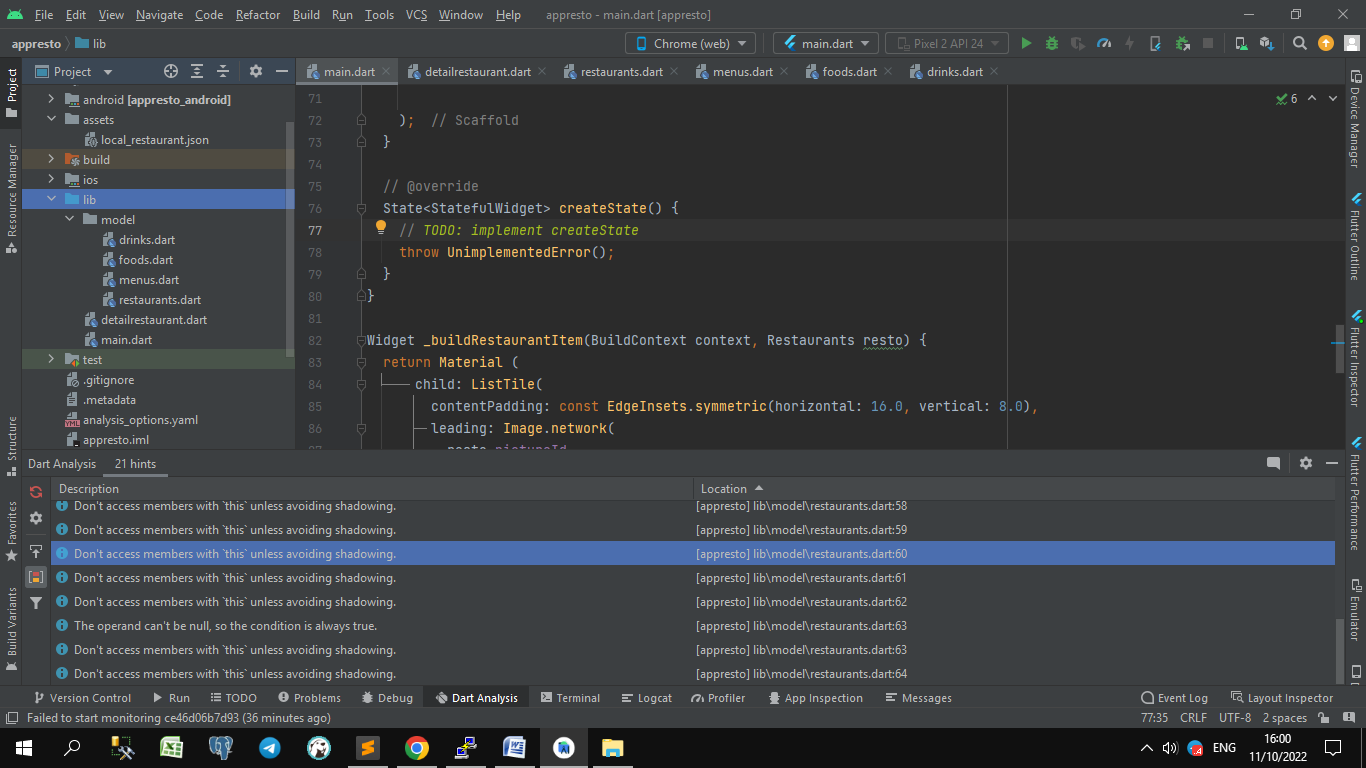
Tools Convert json to dart class : <https://javiercbk.github.io/json_to_dart/>

**1.Networking**

**File :pubspec.yaml**

*# To add assets to your application, add an assets section, like this:*assets:  
 - assets/local\_restaurant.json

Buat model dari file json : pakai tools : <https://javiercbk.github.io/json_to_dart/>



**File :main.dart**

import 'package:flutter/material.dart';  
import 'package:appresto/model/restaurants.dart';  
import 'dart:convert';  
import 'dart:core';  
  
import 'detailrestaurant.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',  
 theme: ThemeData(  
 // This is the theme of your application.  
 //  
 // Try running your application with "flutter run". You'll see the  
 // application has a blue toolbar. Then, without quitting the app, try  
 // changing the primarySwatch below to Colors.green and then invoke  
 // "hot reload" (press "r" in the console where you ran "flutter run",  
 // or simply save your changes to "hot reload" in a Flutter IDE).  
 // Notice that the counter didn't reset back to zero; the application  
 // is not restarted.  
 primarySwatch: Colors.*blue*,  
 ),  
 initialRoute: RestaurantListPage.*routeName*,  
 routes: {  
 RestaurantListPage.*routeName*: (context) => const RestaurantListPage(),  
 DetailRestaurantPage.*routeName*: (context) => DetailRestaurantPage(  
 resto: ModalRoute.*of*(context)?.settings.arguments as Restaurants  
 ),  
 },  
 // home: const RestaurantListPage(),  
 );  
 }  
}  
  
  
class RestaurantListPage extends StatelessWidget {  
 static const *routeName*='restaurant\_list';  
 const RestaurantListPage({Key? key}): super(key:key);  
 // final String title;  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 // Here we take the value from the MyHomePage object that was created by  
 // the App.build method, and use it to set our appbar title.  
 title: const Text("Restaurant App"),  
 ),  
 body: FutureBuilder<String>(  
 //DefaultAssetBundle == widget yang membaca data yang kita berikan  
 future: DefaultAssetBundle.*of*(context).loadString('local\_restaurant.json'),  
 builder: (context, snapshot) {  
 final List<Restaurants> restaurants = parseRestaurants(snapshot.data);  
 return ListView.builder(  
 itemCount: restaurants.length,  
 itemBuilder: (context, index) {  
 return \_buildRestaurantItem(context, restaurants[index]);  
 }  
 );  
 },  
 ),  
 );  
 }  
  
 // @override  
 State<StatefulWidget> createState() {  
 // *TODO: implement createState* throw UnimplementedError();  
 }  
}  
  
Widget \_buildRestaurantItem(BuildContext context, Restaurants resto) {  
 return Material (  
 child: ListTile(  
 contentPadding: const EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0),  
 leading: Image.network(  
 resto.pictureId,  
 width: 100,  
 ),  
 title: Text(resto.name),  
 subtitle: Text(resto.description),  
 onTap: () {  
 Navigator.*pushNamed*(context, DetailRestaurantPage.*routeName*,arguments: resto);  
 },  
 )  
 );  
}  
  
List<Restaurants> parseRestaurants(String? data){  
 if (data==null){  
 return [];  
 }  
 final List parsed=jsonDecode(data)['restaurants'];  
 return parsed.map((data)=>Restaurants.fromJson(data)).toList();  
}

**File :restaurants.dart**

import 'menus.dart';  
class Restaurant {  
 List<Restaurants>? restaurants;  
 Restaurant({this.restaurants});  
  
 Restaurant.fromJson(Map<String, dynamic> json) {  
 if (json['restaurants'] != null) {  
 restaurants = <Restaurants>[];  
 json['restaurants'].forEach((v) {  
 restaurants!.add(new Restaurants.fromJson(v));  
 });  
 }  
 }  
 Map<String, dynamic> toJson() {  
 final Map<String, dynamic> data = new Map<String, dynamic>();  
 if (this.restaurants != null) {  
 data['restaurants'] = this.restaurants!.map((v) => v.toJson()).toList();  
 }  
 return data;  
 }  
}  
  
class Restaurants {  
 String id;  
 String name;  
 String description;  
 String pictureId;  
 String city;  
 double rating;  
 Menus menus;  
  
 Restaurants(  
 { this.id,  
 required this.name,  
 required this.description,  
 required this.pictureId,  
 required this.city,  
 required this.rating,  
 required this.menus});  
 factory Restaurants.fromJson(Map<String, dynamic> json) => Restaurants (  
 id : json['id'],  
 name : json['name'],  
 description : json['description'],  
 pictureId : json['pictureId'],  
 city : json['city'],  
 rating : json['rating'],  
 menus : Menus.fromJson(json['menus'])  
 );  
  
 Map<String, dynamic> toJson() {  
 final Map<String, dynamic> data = new Map<String, dynamic>();  
 data['id'] = this.id;  
 data['name'] = this.name;  
 data['description'] = this.description;  
 data['pictureId'] = this.pictureId;  
 data['city'] = this.city;  
 data['rating'] = this.rating;  
 if (this.menus != null) {  
 data['menus'] = this.menus.toJson();  
 }  
 return data;  
 }  
}

**File :menus.dart**

import 'package:appresto/model/restaurants.dart';  
  
import 'foods.dart';  
import 'drinks.dart';  
  
class Menus {  
 List<Foods> foods;  
 List<Drinks> drinks;  
  
 Menus({required this.foods, required this.drinks});  
  
 // factory constroctor == create dirinya sendiri  
 // fromJson == nama constructor  
 factory Menus.fromJson(Map<String, dynamic> json) {  
 // final = gak bisa diisi lagi kalau kosong  
 final foods = <Foods>[];  
 if (json['foods'] != null) {  
 json['foods'].forEach((v) {  
 foods.add(Foods.fromJson(v));  
 });  
 }  
 final drinks = <Drinks>[];  
 if (json['drinks'] != null) {  
 json['drinks'].forEach((v) {  
 drinks.add(Drinks.fromJson(v));  
 });  
 }  
 return Menus(foods: foods, drinks:drinks);  
 }  
  
 // toJson == nama fungsi  
 Map<String, dynamic> toJson() {  
 final Map<String, dynamic> data = Map<String, dynamic>();  
 if (foods != null) {  
 data['foods'] = foods!.map((v) => v.toJson()).toList();  
 }  
 if (drinks != null) {  
 data['drinks'] = drinks!.map((v) => v.toJson()).toList();  
 }  
 return data;  
 }  
}

**File :foods.dart**

class Foods {  
 String? name;  
  
 Foods({this.name});  
  
 Foods.fromJson(Map<String, dynamic> json) {  
 name = json['name'];  
 }  
  
 Map<String, dynamic> toJson() {  
 final Map<String, dynamic> data = Map<String, dynamic>();  
 data['name'] = name;  
 return data;  
 }  
}

**File :drinks.dart**

class Drinks {  
 String? name;  
  
 Drinks({this.name});  
  
 Drinks.fromJson(Map<String, dynamic> json) {  
 name = json['name'];  
 }  
  
 Map<String, dynamic> toJson() {  
 final Map<String, dynamic> data = Map<String, dynamic>();  
 data['name'] = this.name;  
 return data;  
 }  
}

**File :detailsrestaurant.dart**

import 'package:flutter/material.dart';  
import 'model/restaurants.dart';  
  
class DetailRestaurantPage extends StatelessWidget{  
 static const *routeName* = "/restaurant\_detail";  
 final Restaurants resto;  
 const DetailRestaurantPage({Key? key, required this.resto}) : super (key:key);  
  
 @override  
 Widget build(BuildContext context) {  
 // *TODO : implemend build* return Scaffold(  
 appBar: AppBar(  
 title: const Text('News'),  
 ),  
 body: SingleChildScrollView(  
 child: Column(  
 children: [  
 Image.network(resto.pictureId),  
 Padding(  
 padding: const EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0),  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Text(resto.description),  
 const Divider(color: Colors.*grey*,),  
 Text(  
 resto.name,  
 style: const TextStyle(  
 color: Colors.*black*,  
 fontWeight: FontWeight.*w100*,  
 fontSize: 24  
 ),  
 ),  
 const Divider(color: Colors.*grey*,),  
 Text(resto.city),  
 // Text('Date: ${resto.publishedAt}', ),  
 const SizedBox(height: 10),  
 ],  
 )  
 )  
 ],  
 ),  
 )  
 );  
 }  
}

**Day 10**

Cari icons : <https://api.flutter.dev/flutter/material/Icons-class.html>

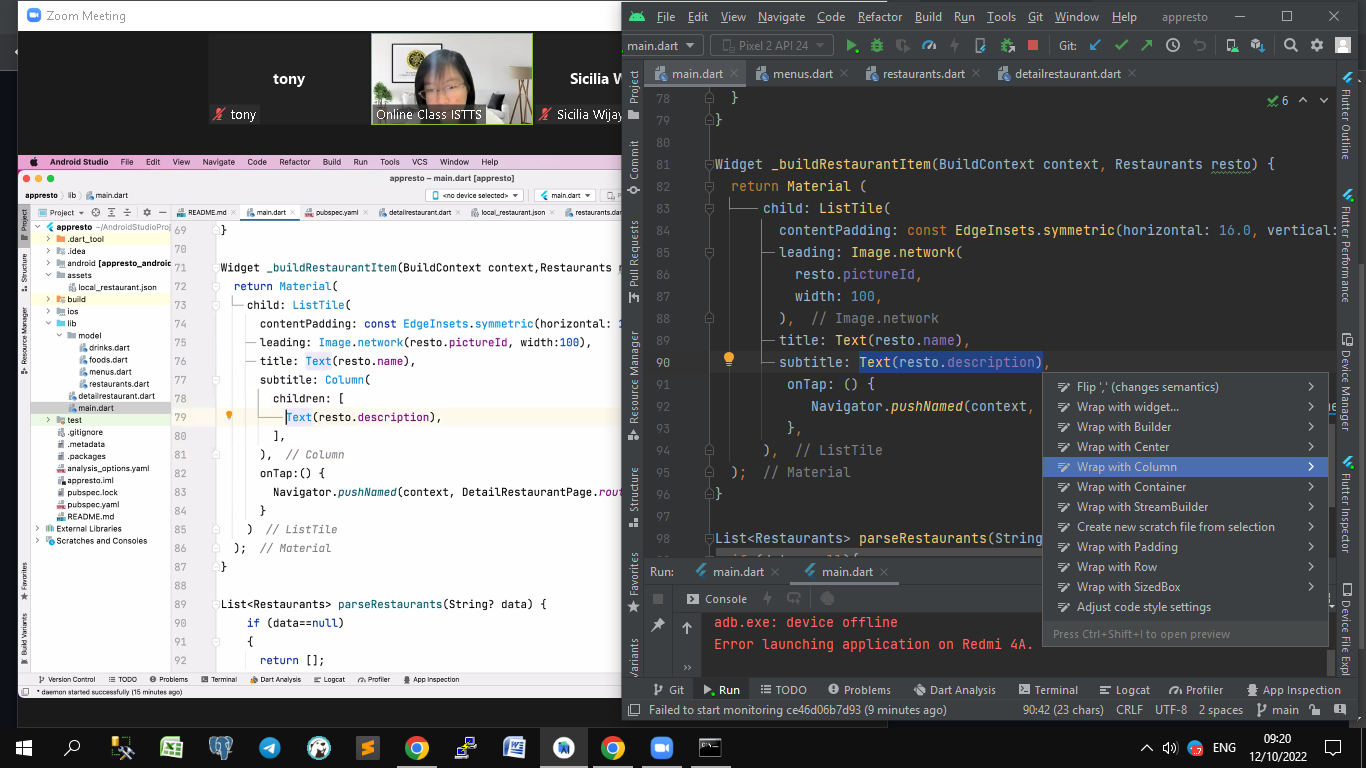
**1.Networking (lanjutan part 2)**

**Pubspec.yaml**

dependencies:  
 equatable: ^2.0.0 *# bandingan 2 object sama / tidak*

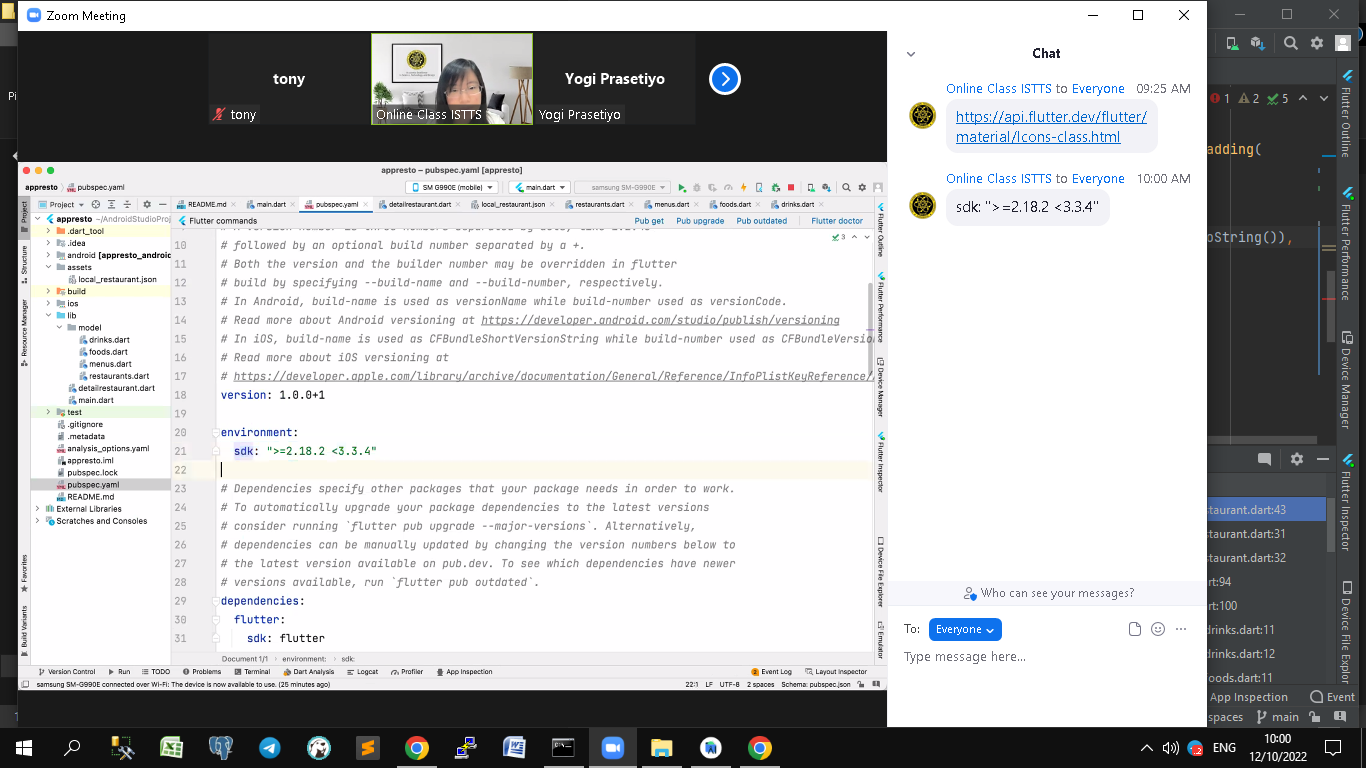
**Shortcut**   
- wrap with column => masukkan code ke Column

right click / alt + enter



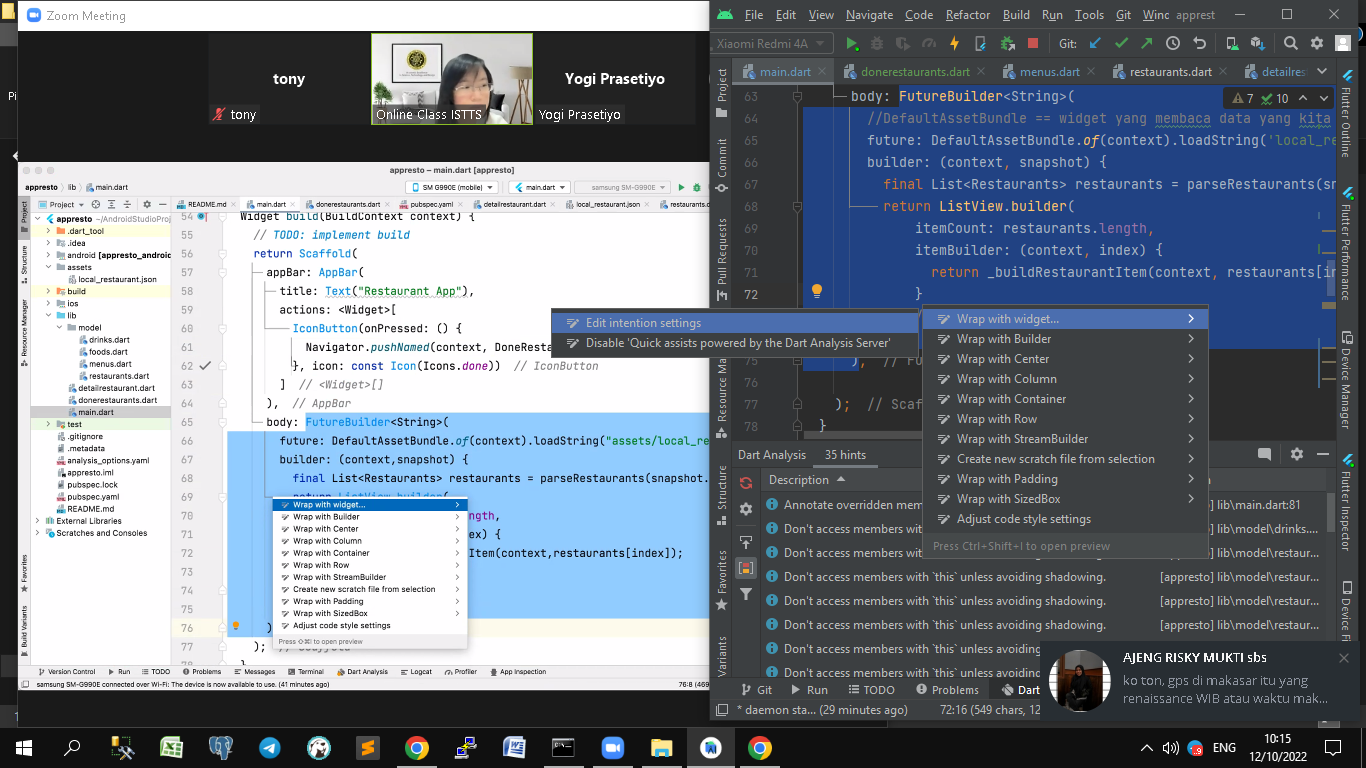
**UpdateFlutter**   
- edit pubspec.yaml

sdk: ">=2.18.2 <3.3.4"



**Shortcut**   
- wrap with widget => masukkan code body ke Widget

right click / alt + enter **-- Wrap with widget ..**



**Shortcut**   
- convert statelessWidget to statefullWidget =>

right click / alt + enter **-- Convert to StatefullWidget ..**

**File :main.dart**

import 'package:flutter/material.dart';  
import 'package:appresto/model/restaurants.dart';  
import 'dart:convert';  
import 'dart:core';  
  
import 'detailrestaurant.dart';  
import 'donerestaurants.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',  
 theme: ThemeData(  
 // This is the theme of your application.  
 //  
 // Try running your application with "flutter run". You'll see the  
 // application has a blue toolbar. Then, without quitting the app, try  
 // changing the primarySwatch below to Colors.green and then invoke  
 // "hot reload" (press "r" in the console where you ran "flutter run",  
 // or simply save your changes to "hot reload" in a Flutter IDE).  
 // Notice that the counter didn't reset back to zero; the application  
 // is not restarted.  
 primarySwatch: Colors.*blue*,  
 ),  
 initialRoute: RestaurantListPage.*routeName*,  
 routes: {  
 RestaurantListPage.*routeName*: (context) => const RestaurantListPage(),  
 DetailRestaurantPage.*routeName*: (context) => DetailRestaurantPage(  
 resto: ModalRoute.*of*(context)?.settings.arguments as Restaurants  
 ),  
 DoneRestaurantsListPage.*routeName*: (context) => DoneRestaurantsListPage(  
 doneRestaurantsList: ModalRoute.*of*(context)?.settings.arguments as List<Restaurants>  
 ),  
 },  
 // home: const RestaurantListPage(),  
 );  
 }  
}  
  
// done restaurant di simpan di state  
class RestaurantListPage extends StatefulWidget {  
 static const *routeName*='restaurant\_list';  
 // final List<Restaurants> doneRestaurantsList;

const RestaurantListPage({Key? key}): super(key:key);  
  
 @override  
 State<RestaurantListPage> createState() => \_RestaurantListPageState();  
}  
  
class \_RestaurantListPageState extends State<RestaurantListPage> {  
 // final String title;  
 final List<Restaurants> doneRestaurantsList = [];  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 // Here we take the value from the MyHomePage object that was created by  
 // the App.build method, and use it to set our appbar title.  
 title: const Text("Restaurant App"),  
 actions: <Widget>[  
 IconButton(onPressed: (){  
 Navigator.*pushNamed*(context, DoneRestaurantsListPage.*routeName*, arguments: doneRestaurantsList  
 );  
 }, icon: Icon(Icons.*done*))  
 ],  
 ),  
  
// done restaurant di lempar2  
 body: RestaurantList(doneRestaurantsList: doneRestaurantsList),  
  
 );  
 }  
  
 // @override  
 State<StatefulWidget> createState() {  
 // *TODO: implement createState* throw UnimplementedError();  
 }  
}  
  
// stateless => stateful == supaya bisa nambah variable  
// class RestaurantList extends StatefulWidget  
class RestaurantList extends StatefulWidget{  
 final List<Restaurants> doneRestaurantsList;  
 const RestaurantList({Key ? key, required this.doneRestaurantsList}):super(key :key);  
 @override  
 State<RestaurantList> createState()=> \_RestaurantListState();  
}  
  
class \_RestaurantListState extends State<RestaurantList>{  
 // const RestaurantList({Key? key}): super (key:key);  
 // disimpan dalam widget bukan state  
 // final List<Restaurants> \_doneRestaurants = const [];  
 @override  
 Widget build(BuildContext context) {  
 return FutureBuilder<String>(  
 //DefaultAssetBundle == widget yang membaca data yang kita berikan  
 future: DefaultAssetBundle.*of*(context).loadString('local\_restaurant.json'),  
 builder: (context, snapshot) {  
 final List<Restaurants> restaurants = parseRestaurants(snapshot.data);  
 return ListView.builder(  
 itemCount: restaurants.length,  
 itemBuilder: (context, index) {  
 // return restaurantItem(restaurants[index]);  
 return restaurantItem(  
 resto: restaurants[index],  
 // widget.doneRestaurantsList ==> hasil disimpan di widget  
 isDone: widget.doneRestaurantsList.contains(restaurants[index]),  
 onClick: (){  
 setState(() {  
 widget.doneRestaurantsList.add(restaurants[index]);  
 });  
 },  
 );  
 }  
 );  
 },  
 );  
 }  
}  
  
class restaurantItem extends StatelessWidget{  
 final Restaurants resto;  
 final bool isDone;  
 // final function() onClick;  
 final Function() onClick;  
 const restaurantItem({  
 Key? key  
 , required this.resto  
 , required this.isDone  
 , required this.onClick  
 }):super(key:key);  
  
 @override  
 Widget build(BuildContext context) {  
 return Material (  
 child: ListTile(  
 contentPadding: const EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0),  
 leading: Image.network(  
 resto.pictureId,  
 width: 100,  
 ),  
 title: Text(resto.name),  
 subtitle: Column(  
 children: [  
 Row(  
 children: [  
 Icon(Icons.*add\_location*),  
 Text(resto.city),  
 ],  
 ),  
 Row(  
 children: [  
 Icon(Icons.*star*),  
 Text(resto.rating.toString()),  
 ],  
 ),  
 ],  
 ),  
 trailing: isDone? const Icon(Icons.*done*) :  
 ElevatedButton(  
 child: Text("Done"),  
 onPressed: onClick,  
 ),  
 onTap: () {  
 Navigator.*pushNamed*(context, DetailRestaurantPage.*routeName*,arguments: resto);  
 },  
 ),  
 );  
 }  
}  
  
List<Restaurants> parseRestaurants(String? data){  
 if (data==null){  
 return [];  
 }  
 final List parsed=jsonDecode(data)['restaurants'];  
 return parsed.map((data)=>Restaurants.fromJson(data)).toList();  
}

**File :detailrestaurant.dart**

import 'package:flutter/material.dart';  
import 'model/restaurants.dart';  
  
class DetailRestaurantPage extends StatelessWidget{  
 static const *routeName* = "/restaurant\_detail";  
 final Restaurants resto;  
 const DetailRestaurantPage({Key? key, required this.resto}) : super (key:key);  
  
 @override  
 Widget build(BuildContext context) {  
 // *TODO : implemend build* return Scaffold(  
 appBar: AppBar(  
 title: Text(resto.name),  
 ),  
 body: SingleChildScrollView(  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Image.network(resto.pictureId),  
 Padding(  
 padding: const EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0),  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Text(  
 resto.description,  
 maxLines: 4,  
 // overflow: TextOverFlow.ellipsis,  
 ),  
 Divider(color: Colors.*grey*),  
 Container(  
 width: 1000,  
 height: 100,  
 child:ListView.builder(  
 shrinkWrap: true,  
 scrollDirection: Axis.horizontal,  
 itemCount: resto.menus.foods.length,  
 itemBuilder: (BuildContext context,int index)=> Padding(  
 padding: const EdgeInsets.all(8.0),  
 child: Card(  
 child: Center(  
 child: Text(resto.menus.foods[index].name.toString()),  
 ),  
 ),  
 ),  
 ),  
 ),  
 // Text(  
 // resto.name,  
 // style: const TextStyle(  
 // color: Colors.black,  
 // fontWeight: FontWeight.w100,  
 // fontSize: 24  
 // ),  
 // ),  
 // const Divider(color: Colors.grey,),  
 // Text(resto.city),  
 // // Text('Date: ${resto.publishedAt}', ),  
 // const SizedBox(height: 10),  
 ],  
 )  
 )  
 ],  
 ),  
 )  
 );  
 }  
}

**File :donerestaurants.dart**

import 'dart:convert';  
  
import 'package:flutter/material.dart';  
import 'detailrestaurant.dart';  
import 'model/restaurants.dart';  
  
class DoneRestaurantsListPage extends StatelessWidget {  
 static const *routeName*='restaurant\_done';  
 final List<Restaurants> doneRestaurantsList;  
  
 const DoneRestaurantsListPage({Key? key, required this.doneRestaurantsList}): super(key:key);  
 // final String title;  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 // Here we take the value from the MyHomePage object that was created by  
 // the App.build method, and use it to set our appbar title.  
 title: const Text("Restaurant App"),  
 ),  
 body: ListView.builder(  
 itemCount: doneRestaurantsList.length,  
 itemBuilder: (context, index) {  
 return \_buildRestaurantItem(context, doneRestaurantsList[index]);  
 }  
 ),  
 );  
 }  
  
 // @override  
 State<StatefulWidget> createState() {  
 // *TODO: implement createState* throw UnimplementedError();  
 }  
}  
  
Widget \_buildRestaurantItem(BuildContext context, Restaurants resto) {  
 return Material (  
 child: ListTile(  
 contentPadding: const EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0),  
 leading: Image.network(  
 resto.pictureId,  
 width: 100,  
 ),  
 title: Text(resto.name),  
 subtitle: Column(  
 children: [  
 Row(  
 children: [  
 Icon(Icons.*add\_location*),  
 Text(resto.city),  
 ],  
 ),  
 Row(  
 children: [  
 Icon(Icons.*star*),  
 Text(resto.rating.toString()),  
 ],  
 ),  
 ],  
 ),  
 trailing: ElevatedButton(  
 child: Text("Done"),  
 onPressed: (){  
  
 },  
 ),  
 onTap: () {  
 Navigator.*pushNamed*(context, DetailRestaurantPage.*routeName*,arguments: resto);  
 },  
 ),  
 );  
}  
  
List<Restaurants> parseRestaurants(String? data){  
 if (data==null){  
 return [];  
 }  
 final List parsed=jsonDecode(data)['restaurants'];  
 return parsed.map((data)=>Restaurants.fromJson(data)).toList();  
}

**File : restaurants.dart**

import 'package:equatable/equatable.dart';  
import 'menus.dart';  
  
//  
class Restaurant{  
 List<Restaurants>? restaurants;  
  
 Restaurant({this.restaurants});  
  
 Restaurant.fromJson(Map<String, dynamic> json) {  
 if (json['restaurants'] != null) {  
 restaurants = <Restaurants>[];  
 json['restaurants'].forEach((v) {  
 restaurants!.add(new Restaurants.fromJson(v));  
 });  
 }  
 }  
  
 Map<String, dynamic> toJson() {  
 final Map<String, dynamic> data = new Map<String, dynamic>();  
 if (this.restaurants != null) {  
 data['restaurants'] = this.restaurants!.map((v) => v.toJson()).toList();  
 }  
 return data;  
 }  
  
}  
  
// Equatable ditambah u/ bandingan 2 object sama / tidak  
class Restaurants extends Equatable {  
 String id;  
 String name;  
 String description;  
 String pictureId;  
 String city;  
 double rating;  
 Menus menus;  
  
 Restaurants(  
 {  
 required this.id,  
 required this.name,  
 required this.description,  
 required this.pictureId,  
 required this.city,  
 required this.rating,  
 required this.menus});  
  
 // ditambah u/ bandingan 2 object sama / tidak  
 @override  
 List<Object?> get props=> [id];  
  
 factory Restaurants.fromJson(Map<String, dynamic> json) => Restaurants (  
 id : json['id'],  
 name : json['name'],  
 description : json['description'],  
 pictureId : json['pictureId'],  
 city : json['city'],  
 rating : json['rating'],  
 menus : Menus.fromJson(json['menus'])  
 );  
  
 Map<String, dynamic> toJson() {  
 final Map<String, dynamic> data = new Map<String, dynamic>();  
 data['id'] = this.id;  
 data['name'] = this.name;  
 data['description'] = this.description;  
 data['pictureId'] = this.pictureId;  
 data['city'] = this.city;  
 data['rating'] = this.rating;  
 if (this.menus != null) {  
 data['menus'] = this.menus.toJson();  
 }  
 return data;  
 }  
  
}

**Day 11**

Tools Convert json to dart class :

1. <https://javiercbk.github.io/json_to_dart/>

2. <https://app.quicktype.io/>

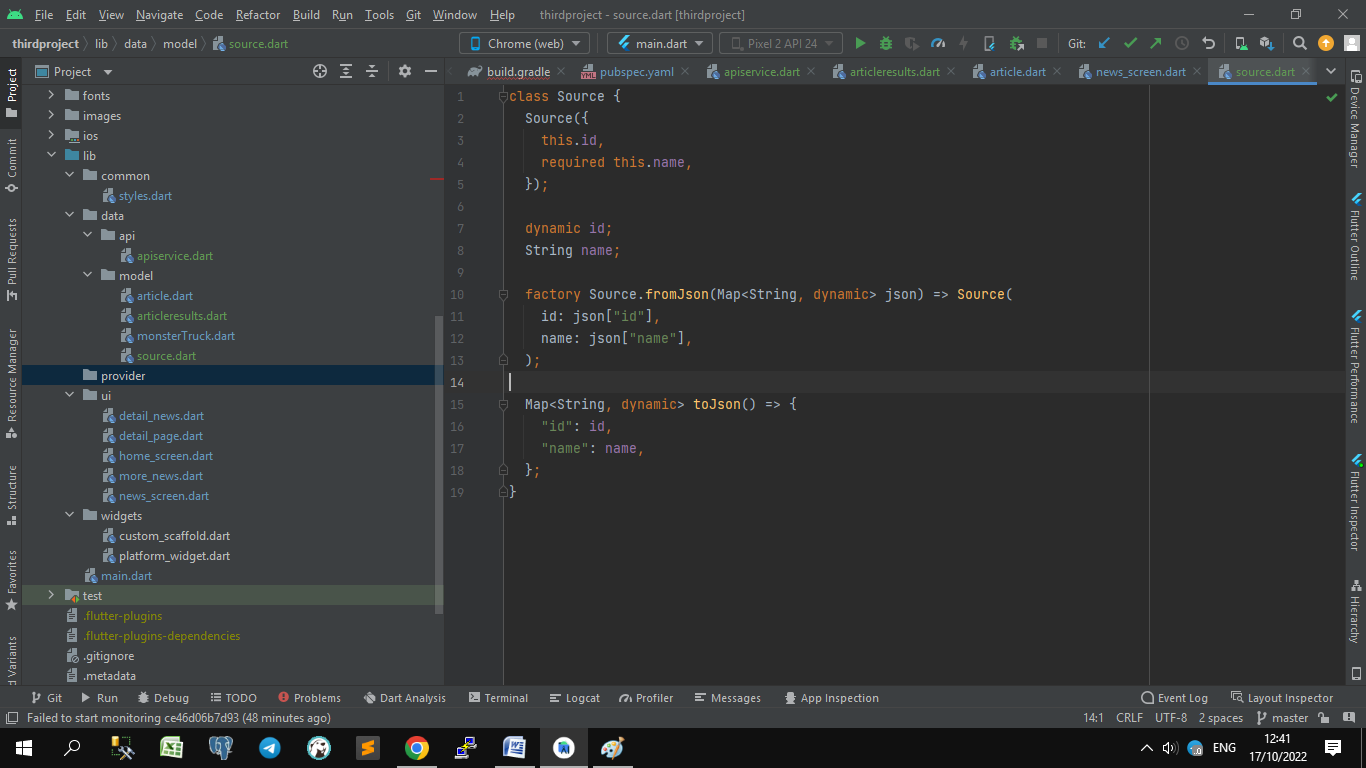


**1.Networking (lanjutan part 3)**

**2.Api**

Penempatan File / Folder

1. Common
   1. Style.dart
2. Data
   1. Api : apiservice.dart
   2. Model
      1. Article.dart
      2. articleResults.dart
      3. source.dart
3. Provider
4. Ui
   1. Halaman selain main.dart
5. Widget
   1. custom\_scaffold.dart
   2. platform\_widget.dart
6. Main.dart



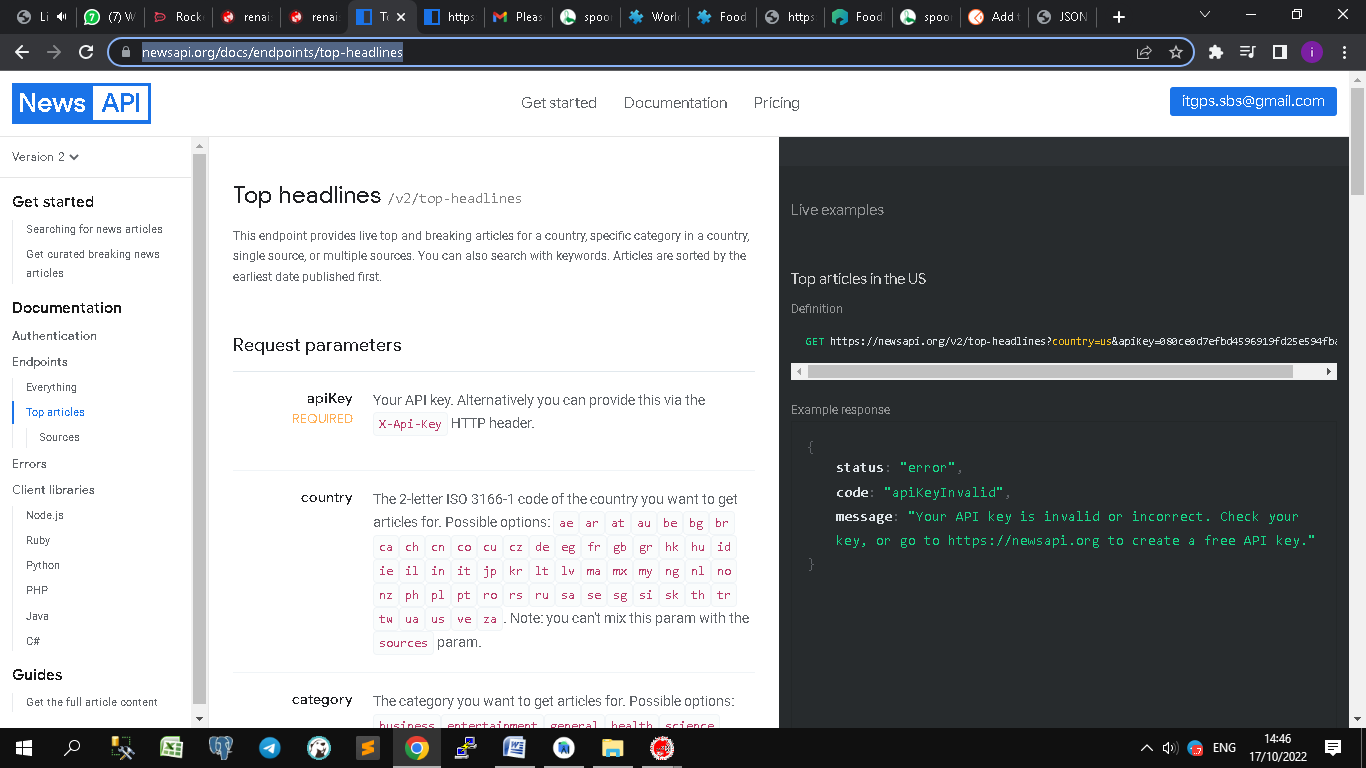
**File : Pubspecs.yaml**

dependencies:  
 http: 0.13.5  
 provider: 6.0.4

**Materi**

endpoint : <https://newsapi.org/>

apiKey : 080ce0d7efbd4596919fd25e594fba8d



**Tugas** : cari endpoint api restaurant

Contoh2 api di <https://www.programmableweb.com/category/restaurants/api>

Endpoint : <https://spoonacular.com/food-api/console#Profile>

apiKey : ef36f267a514448c84f50361e8532ea9



**File : apiservice.dart**

import 'dart:convert';  
  
import '../model/articleresults.dart';  
import 'package:http/http.dart' as http;  
  
class ApiService{  
 static const String *url*="https://newsapi.org/v2/";  
 static const String *apiKey*="080ce0d7efbd4596919fd25e594fba8d";  
  
 Future<ArticleResults> topHeadLines(String country, String category) async{  
 //https://newsapi.org/v2/top-headlines?country=us&apiKey=080ce0d7efbd4596919fd25e594fba8d  
 final response = await http.get(Uri.*parse*("${*url*}top-headlines?country=${country}&category=${category}&apiKey=${*apiKey*}"));  
 if (response.statusCode==200){  
 return ArticleResults.fromJson(json.decode(response.body));  
 }else{  
 throw Exception("Failed to Load data from Newsapi");  
 }  
 }  
}

**File : articleResults.dart**

// To parse this JSON data, do  
//  
// final articleResult = articleResultFromJson(jsonString);  
  
import 'dart:convert';  
  
import 'article.dart';  
  
ArticleResults articleResultsFromJson(String str) => ArticleResults.fromJson(json.decode(str));  
  
String articleResultsToJson(ArticleResults data) => json.encode(data.toJson());  
  
class ArticleResults {  
 ArticleResults({  
 required this.status,  
 required this.totalResults,  
 required this.articles,  
 });  
  
 String status;  
 int totalResults;  
 List<Article> articles;  
  
 factory ArticleResults.fromJson(Map<String, dynamic> json) => ArticleResults(  
 status: json["status"],  
 totalResults: json["totalResults"],  
 articles: List<Article>.from(json["articles"].map((x) => Article.fromJson(x)).where(  
 (article)=>article.author!=null&&article.urlToImage!=null&&article.publishedAt!=null&&article.content!=null  
 )),  
 );  
  
 Map<String, dynamic> toJson() => {  
 "status": status,  
 "totalResults": totalResults,  
 "articles": List<dynamic>.from(articles.map((x) => x.toJson())),  
 };  
}

**File : article.dart**

List<Article> parseArticles(String? json){  
 if (json==null){  
 return [];  
 }  
 final List parsed=jsonDecode(json);  
 return parsed.map((json)=>Article.fromJson(json)).toList();  
}  
  
  
  
class Article {  
 Article({  
 required this.source,  
 required this.author,  
 required this.title,  
 required this.description,  
 required this.url,  
 required this.urlToImage,  
 required this.publishedAt,  
 required this.content,  
 });  
  
 Source source;  
 String author;  
 String title;  
 String description;  
 String url;  
 String urlToImage;  
 DateTime publishedAt;  
 String content;  
  
 factory Article.fromJson(Map<String, dynamic> json) => Article(  
 source: Source.fromJson(json["source"]),  
 author: json["author"],  
 title: json["title"],  
 description: json["description"],  
 url: json["url"],  
 urlToImage: json["urlToImage"],  
 publishedAt: DateTime.*parse*(json["publishedAt"]),  
 content: json["content"] == null ? null : json["content"],  
 );  
  
 Map<String, dynamic> toJson() => {  
 "source": source.toJson(),  
 "author": author,  
 "title": title,  
 "description": description,  
 "url": url,  
 "urlToImage": urlToImage,  
 "publishedAt": publishedAt.toIso8601String(),  
 "content": content == null ? null : content,  
 };  
}

**File : source.dart**

class Source {  
 Source({  
 this.id,  
 required this.name,  
 });  
  
 dynamic id;  
 String name;  
  
 factory Source.fromJson(Map<String, dynamic> json) => Source(  
 id: json["id"],  
 name: json["name"],  
 );  
  
 Map<String, dynamic> toJson() => {  
 "id": id,  
 "name": name,  
 };  
}

**Day 12**

Tools Convert json to dart class :

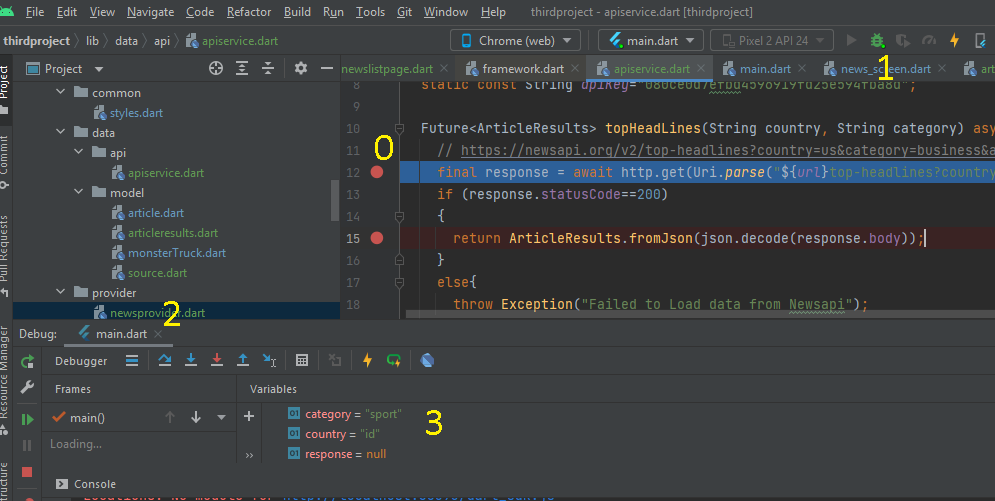
<https://javiercbk.github.io/json_to_dart/>

api news : <https://newsapi.org/>

tes : All articles about Tesla from the last month, sorted by recent first

https://newsapi.org/v2/everything?**q=tesla**&from=2022-09-19&sortBy=publishedAt&apiKey=080ce0d7efbd4596919fd25e594fba8d

compile / debug : SHIFT + F9

  
Langkah2 debug:

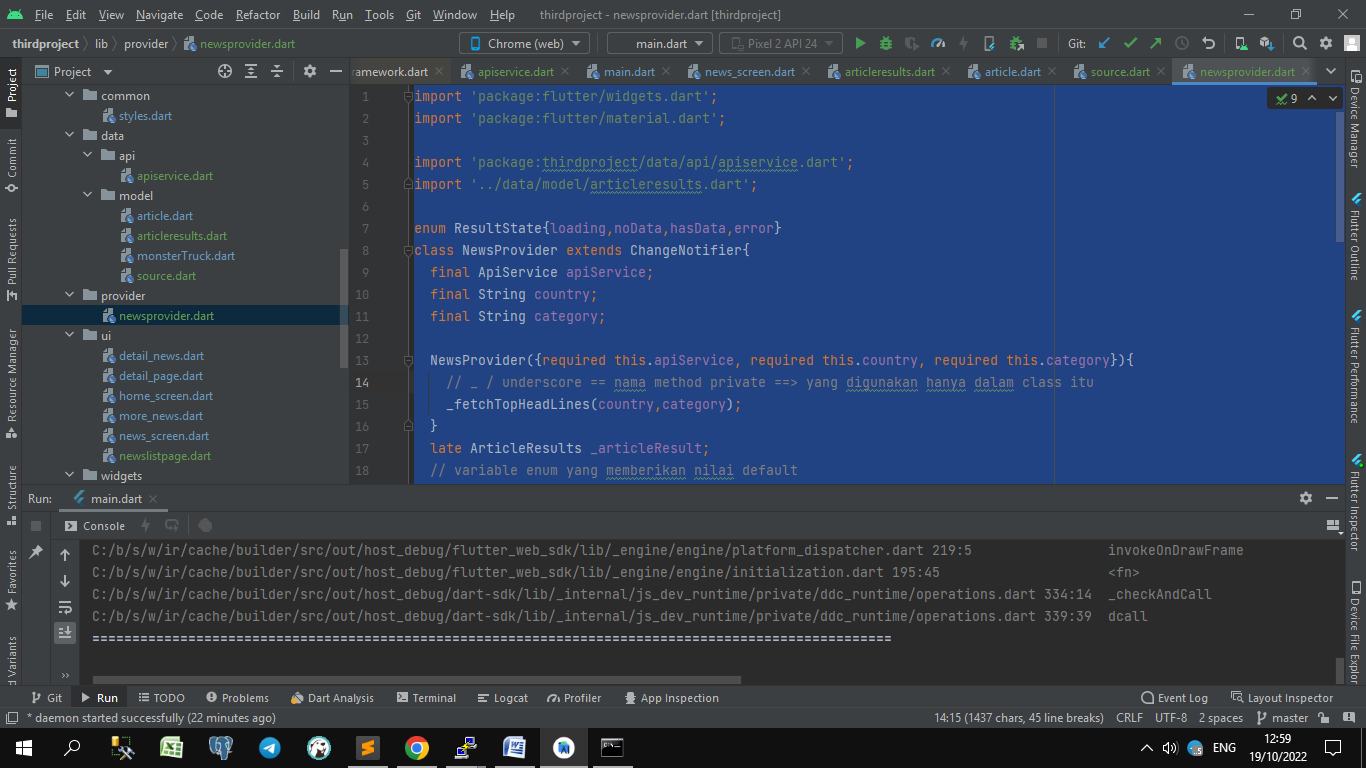
0. Beri Tanda pada code yang error

1. klik tombol debug / Shift + F9
2. tekan tombol F8 / next
3. cek isi variable apa sudah sesuai

**1.Networking (lanjutan part 4)**

Directory Provider – dapat dipanggil di setiap dart file

<https://docs.flutter.dev/development/data-and-backend/state-mgmt/simple>

****

**File : provider/newsprovider.dart**

import 'package:flutter/widgets.dart';  
import 'package:flutter/material.dart';  
  
import 'package:thirdproject/data/api/apiservice.dart';  
import '../data/model/articleresults.dart';  
  
enum ResultState{loading,noData,hasData,error}  
class NewsProvider extends ChangeNotifier{  
 final ApiService apiService;  
 final String country;  
 final String category;  
  
 NewsProvider({required this.apiService, required this.country, required this.category}){  
 // \_ / underscore == nama method private ==> yang digunakan hanya dalam class itu  
 \_fetchTopHeadLines(country,category);  
 }  
 late ArticleResults \_articleResult;  
 // variable enum yang memberikan nilai default  
 late ResultState \_state;  
 String \_message="";  
 String get message => \_message;  
 ArticleResults get result=>\_articleResult;  
 ResultState get state=>\_state;  
  
 Future<dynamic> \_fetchTopHeadLines(String country, String category) async{  
 try {  
 \_state = ResultState.loading;  
 notifyListeners();  
 final article = await apiService.topHeadLines(country, category);  
 if (article.articles.isEmpty){  
 \_state = ResultState.noData;  
 notifyListeners();  
 return \_message='Empty Data';  
 }  
 else{  
 \_state = ResultState.hasData;  
 notifyListeners();  
 return \_articleResult=article;  
 }  
 }catch(e){  
 \_state = ResultState.error;  
 notifyListeners();  
 return \_message = 'Error --> $e';  
 }  
 }  
}

**File : ui/newslistpage.dart**

import 'dart:js';  
  
import 'package:flutter/material.dart';  
import 'package:provider/provider.dart';  
import 'package:thirdproject/data/model/articleresults.dart';  
import 'package:thirdproject/provider/newsprovider.dart';  
import 'package:thirdproject/ui/detail\_news.dart';  
  
import '../data/model/article.dart';  
  
class NewsListPage extends StatelessWidget{  
 static const *routeName*='news\_list';  
 const NewsListPage({Key? key}): super(key:key);  
 @override  
 Widget \_buildList(){  
 return Consumer<NewsProvider>(  
 builder: (context, state,\_) {  
 if (state.state == ResultState.loading) {  
 return const Center(child: CircularProgressIndicator());  
 }  
 else if (state.state == ResultState.hasData) {  
 return ListView.builder(  
 itemCount: state.result.articles.length,  
 itemBuilder: (context,index){  
 return \_buildArticleItem(context, state.result.articles[index]);  
 }  
 );  
 }  
 else if (state.state == ResultState.noData) {  
 return Center(  
 child: Material(  
 child: Text(state.message),  
 )  
 );  
 }  
 else {  
 return Center(  
 child: Material(  
 child: Text(''),  
 )  
 );  
 }  
 }  
 );  
 }  
  
 @override  
 Widget build(BuildContext context){  
 return Scaffold(  
 body:\_buildList(),  
 );  
 }  
}  
  
Widget \_buildArticleItem(BuildContext context, Article article){  
 return Material(  
 child: ListTile(  
 contentPadding: const EdgeInsets.symmetric(horizontal: 16.0, vertical: 8.0),  
 leading: Image.network(  
 article.urlToImage,  
 width: 100,  
 ),  
 title: Text(article.title),  
 subtitle: Text(article.author),  
 onTap: (){  
 Navigator.*pushNamed*(context, DetailNewsScreen.*routeName*,arguments: article);  
 },  
 )  
 );  
}

**File : ui/news\_screen.dart**

// platforms IOS  
// import 'package:flutter/cupertino.dart';  
  
import 'package:flutter/material.dart';  
import 'package:provider/provider.dart';  
import 'package:thirdproject/provider/newsprovider.dart';  
import 'package:thirdproject/ui/detail\_news.dart';  
import 'package:thirdproject/widgets/platform\_widget.dart';  
  
import '../data/api/apiservice.dart';  
import '../data/model/article.dart';  
import 'newslistpage.dart';  
  
class NewsScreen extends StatefulWidget {  
 static const *routeName* = '/articles\_list';  
 const NewsScreen({Key? key}) : super(key: key);  
  
 @override  
 State<NewsScreen> createState() => \_NewsScreenState();  
}  
  
class \_NewsScreenState extends State<NewsScreen>{  
 @override  
 Widget build(BuildContext context){  
 return Scaffold(  
 body:  
 ChangeNotifierProvider<NewsProvider>(  
 // parameter \_ adalah parameter yang gak akan dipakai  
 create: (\_)=> NewsProvider(apiService: ApiService(),country:'id', category: 'sport'),  
 child: const NewsListPage(),  
 )  
 );  
 }  
}

**File : data/api/news\_screen.dart**

import 'dart:convert';  
  
import '../model/articleresults.dart';  
import 'package:http/http.dart' as http;  
  
class ApiService{  
 static const String *url*="https://newsapi.org/v2/";  
 static const String *apiKey*="080ce0d7efbd4596919fd25e594fba8d";  
  
 Future<ArticleResults> topHeadLines(String country, String category) async{  
 // https://newsapi.org/v2/top-headlines?country=us&category=business&apiKey=080ce0d7efbd4596919fd25e594fba8d  
 final response = await http.get(Uri.*parse*("${*url*}top-headlines?country=${country}&category=${category}&apiKey=${*apiKey*}"));  
 if (response.statusCode==200)  
 {  
 return ArticleResults.fromJson(json.decode(response.body));  
 }  
 else{  
 throw Exception("Failed to Load data from Newsapi");  
 }  
 }  
  
 Future<ArticleResults> searchHeadLines(String date, String query) async{  
 //https://newsapi.org/v2/everything?q=tesla&from=2022-09-19&sortBy=publishedAt&apiKey=080ce0d7efbd4596919fd25e594fba8d  
 final response = await http.get(Uri.*parse*("${*url*}everything?q=${query}&from=${date}&sortBy=publishedAt&apiKey=${*apiKey*}"));  
 if (response.statusCode==200)  
 {  
 return ArticleResults.fromJson(json.decode(response.body));  
 }  
 else{  
 throw Exception("Failed to Load data from Newsapi");  
 }  
 }  
}

**File : data/model/source.dart**

class Source {  
 Source({  
 this.id,  
 required this.name,  
 });

dynamic id;  
 String name;  
  
 factory Source.fromJson(Map<String, dynamic> json) => Source(  
 id: json["id"] ?? 'null',  
 name: json["name"] ?? 'null',  
 );  
  
 Map<String, dynamic> toJson() => {  
 "id": id,  
 "name": name,  
 };  
}

**File : data/model/articleresults.dart**

// To parse this JSON data, do  
//  
// final articleResult = articleResultFromJson(jsonString);  
  
import 'dart:convert';  
  
import 'article.dart';  
  
ArticleResults articleResultsFromJson(String str) => ArticleResults.fromJson(json.decode(str));  
  
String articleResultsToJson(ArticleResults data) => json.encode(data.toJson());  
  
class ArticleResults {  
 ArticleResults({  
 required this.status,  
 required this.totalResults,  
 required this.articles,  
 });  
  
 String status;  
 int totalResults;  
 List<Article> articles;  
  
 factory ArticleResults.fromJson(Map<String, dynamic> json) => ArticleResults(  
 status: json["status"],  
 totalResults: json["totalResults"],  
 articles: List<Article>.from(json["articles"].map((x) => Article.fromJson(x)).where(  
 (article)=>  
 article.author!=null  
 &&article.urlToImage!=null  
 &&article.publishedAt!=null  
 &&article.content!=null  
  
 )),  
 );  
  
 Map<String, dynamic> toJson() => {  
 "status": status,  
 "totalResults": totalResults,  
 "articles": List<dynamic>.from(articles.map((x) => x.toJson())),  
 };  
}

**File : data/model/article.dart**

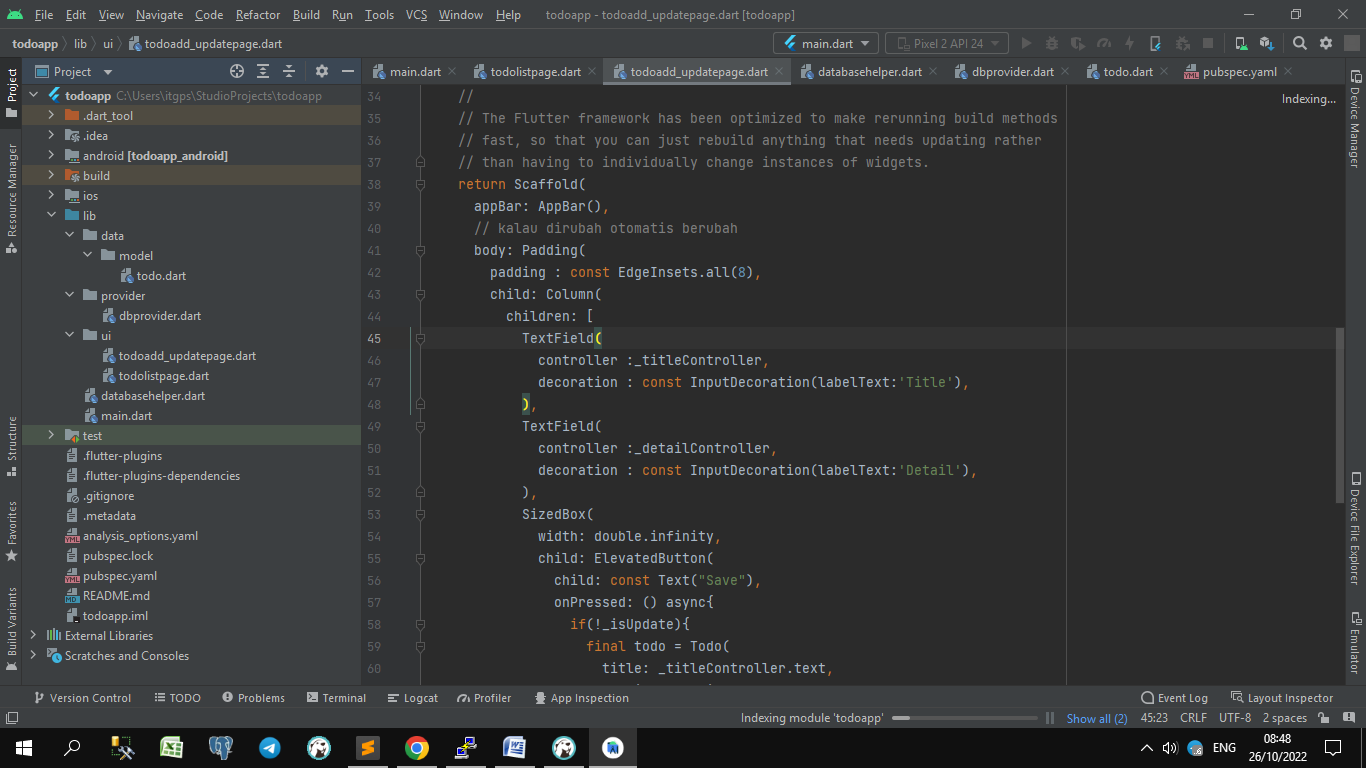
import 'dart:convert';  
import 'source.dart';

List<Article> parseArticles(String? json){  
 if (json==null){  
 return [];  
 }  
 final List parsed=jsonDecode(json);  
 return parsed.map((json)=>Article.fromJson(json)).toList();  
}  
  
class Article {  
 Article({  
 required this.source,  
 required this.author,  
 required this.title,  
 required this.description,  
 required this.url,  
 required this.urlToImage,  
 required this.publishedAt,  
 required this.content,  
 });  
  
 Source source;  
 String author;  
 String title;  
 String description;  
 String url;  
 String urlToImage;  
 DateTime publishedAt;  
 String content;  
  
 factory Article.fromJson(Map<String, dynamic> json) => Article(  
 source: json["source"] != null ? Source.fromJson(json["source"]) : Source(id: "", name: ""),  
 author: json["author"] ?? "null",  
 title: json["title"] ?? "null",  
 description: json["description"] ?? "null",  
 url: json["url"] ?? "null",  
 urlToImage: json["urlToImage"] ?? "null",  
 publishedAt: json["publishedAt"] != null ? DateTime.*parse*(json["publishedAt"]) : DateTime.*parse*('1900-01-01'),  
 content: json["content"] ?? "null",  
 );  
  
 Map<String, dynamic> toJson() => {  
 "source": source.toJson(),  
 "author": author,  
 "title": title,  
 "description": description,  
 "url": url,  
 "urlToImage": urlToImage,  
 "publishedAt": publishedAt.toIso8601String(),  
 "content": content ?? null,  
 };  
}

**Day 13**

**File : Pubspecs.yaml**

dependencies:  
 **sqflite**: ^2.0.0+3  
 **provider**: ^6.0.3



**Directory**

1. data
   1. model
      1. todo.dart
2. provider
   1. dbprovider.dart
3. ui
   1. todoadd\_updatepage.dart
   2. todolistpage.dart
4. databasehelper.dart
5. main.dart

**1. Menyimpan Data dengan SQLite**

Langkah2 buat db:

1. tambah dependencies : sqflite: ^2.0.0+3
2. buat dbprovider.dart -- provider
3. buat databasehelper.dart –- path db
4. buat todo.dart –- model
5. tambahkan akses u/ DbProvider di main.dart
6. buat home : todolistpage.dart
7. buat add page : todoadd\_updatepage.dart

**File : data/model/todo.dart**

class Todo{  
 late int? id;  
 late String title;  
 late String detail;  
  
 Todo({this.id, required this.title, required this.detail});  
  
 Map<String,dynamic> toMap(){  
 return{  
 'id':id,  
 'title':title,  
 'detail':detail,  
 };  
 }  
 Todo.fromMap(Map<String,dynamic> map){  
 id = map['id'];  
 title = map['title'];  
 detail = map['detail'];  
 }  
}

**File : provider/dbprovider.dart**

import 'package:flutter/material.dart';  
import 'package:sqflite/sqflite.dart';  
import 'package:todoapp/databasehelper.dart';  
  
import '../data/model/todo.dart';  
  
class DbProvider extends ChangeNotifier {  
 List<Todo> \_arrToDo = [];  
 late DatabaseHelper \_dbHelper;  
 List<Todo> get todos => \_arrToDo;  
 DbProvider(){  
 \_dbHelper = DatabaseHelper();  
 \_getAllTodos();  
 }  
  
 // async – nunggu response data sampai selesai  
 void \_getAllTodos() async{  
 \_arrToDo = await \_dbHelper.getTodos();  
 // cek insert sekaligus refresh data  
 notifyListeners();  
 }  
  
 Future<void> addTodo(Todo todo) async{  
 await \_dbHelper.insertTodo(todo);  
 \_getAllTodos();  
 }  
}

**File : ui/todoadd\_updatepage.dart**

import 'package:flutter/material.dart';  
import 'package:provider/provider.dart';  
import 'package:todoapp/provider/dbprovider.dart';  
  
import '../data/model/todo.dart';  
  
class ToAddUpdatePage extends StatefulWidget {  
 final Todo? todo;  
 const ToAddUpdatePage({super.key, this.todo});  
  
 @override  
 State<ToAddUpdatePage> createState() => \_ToAddUpdatePageState();  
}  
  
class \_ToAddUpdatePageState extends State<ToAddUpdatePage> {  
 final TextEditingController \_titleController = TextEditingController();  
 final TextEditingController \_detailController = TextEditingController();  
 bool \_isUpdate=false;

// refresh data setelah insert  
 @override  
 void initState(){  
 super.initState();  
 if (widget.todo!=null){  
 \_titleController.text = widget.todo!.title;  
 \_detailController.text = widget.todo!.detail;  
 \_isUpdate = true;  
 }  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 // This method is rerun every time setState is called, for instance as done  
 // by the \_incrementCounter method above.  
 //  
 // The Flutter framework has been optimized to make rerunning build methods  
 // fast, so that you can just rebuild anything that needs updating rather  
 // than having to individually change instances of widgets.  
 return Scaffold(  
 appBar: AppBar(),  
 // kalau dirubah otomatis berubah  
 body: Padding(  
 padding : const EdgeInsets.all(8),  
 child: Column(  
 children: [  
 TextField(  
 controller :\_titleController,  
 decoration : const InputDecoration(labelText:'Title'),  
 ),  
 TextField(  
 controller :\_detailController,  
 decoration : const InputDecoration(labelText:'Detail'),  
 ),  
 SizedBox(  
 width: double.*infinity*,  
 child: ElevatedButton(  
 child: const Text("Save"),  
 onPressed: () async{  
 if(!\_isUpdate){  
 final todo = Todo(  
 title: \_titleController.text,  
 detail: \_detailController.text  
 );  
 Provider.*of*<DbProvider>(context, listen: false).addTodo(todo);  
 }  
 Navigator.*pop*(context);  
 },  
 ),  
 ),  
 ],  
 ),  
 ), // This trailing comma makes auto-formatting nicer for build methods.  
 );  
 }  
 // controller gak otomatis dispose tapi harus manual dispose  
 @override  
 void dispose(){  
 \_titleController.dispose();  
 \_detailController.dispose();  
 super.dispose();  
 }  
}

**File : ui/todolistpage.dart**

import 'package:flutter/material.dart';  
import 'package:provider/provider.dart';  
import 'package:todoapp/provider/dbprovider.dart';  
import 'package:todoapp/ui/todoadd\_updatepage.dart';  
  
class ToDoListPage extends StatelessWidget {  
 const ToDoListPage({super.key});  
  
 @override  
 Widget build(BuildContext context) {  
 // This method is rerun every time setState is called, for instance as done  
 // by the \_incrementCounter method above.  
 //  
 // The Flutter framework has been optimized to make rerunning build methods  
 // fast, so that you can just rebuild anything that needs updating rather  
 // than having to individually change instances of widgets.  
 return Scaffold(  
 appBar: AppBar(  
 // Here we take the value from the MyHomePage object that was created by  
 // the App.build method, and use it to set our appbar title.  
 title: const Text('Todo app'),  
 ),  
 // kalau dirubah otomatis berubah  
 body: Consumer<DbProvider>(  
 builder: (context, provider, child){  
 final arrToDo = provider.todos;  
 return ListView.builder(  
 itemCount: arrToDo.length,  
 itemBuilder: (context,index){  
 final todo = arrToDo[index];  
 return Dismissible(  
 key: Key(todo.id.toString()),  
 background: Container(color:Colors.*red*),  
 onDismissed: (direction){  
  
 },  
 child: Card(  
 child: ListTile(  
 title: Text(todo.title),  
 subtitle: Text(todo.detail),  
 onTap: () async{  
  
 },  
 )  
 )  
 );  
 },  
 );  
 },  
 ),  
 floatingActionButton: FloatingActionButton(  
 onPressed: (){   
 Navigator.*push*(context, MaterialPageRoute(builder: (context)=>const ToAddUpdatePage()));  
 },  
 child: const Icon(Icons.*add*),  
 ), // This trailing comma makes auto-formatting nicer for build methods.  
 );  
 }  
}

**File : databasehelper.dart**

import 'package:sqflite/sqflite.dart';  
import 'data/model/todo.dart';  
  
class DatabaseHelper{  
 static late Database *\_database*;  
 static DatabaseHelper? *\_databaseHelper*;  
 DatabaseHelper.\_internal(){  
 *\_databaseHelper* = this;  
 }  
  
 factory DatabaseHelper()=>*\_databaseHelper*??DatabaseHelper.\_internal();  
 Future<Database> get database async{  
 *\_database* = await \_initializeDb();  
 return *\_database*;  
 }  
  
 static const String *\_tableName*='todos';  
  
 Future<Database> \_initializeDb() async {  
 var path = await getDatabasesPath();  
 var db = openDatabase(  
 "$path/todo\_db.db",  
 onCreate: (db, version) async{  
 await db.execute('CREATE TABLE $*\_tableName* (ID INTEGER PRIMARY KEY, title TEXT, detail TEXT)');  
 },  
 version: 1,  
 );  
 return db;  
 }  
  
 Future<void> insertTodo(Todo todo) async{  
 final Database db = await database;  
 await db.insert(*\_tableName*, todo.toMap());  
 print('Data inserted');  
 }

Future<List<Todo>> getTodos() async{  
 final Database db = await database;  
 List<Map<String,dynamic>> results = await db.query(*\_tableName*);  
 return results.map((res)=>Todo.fromMap(res)).toList();  
 }  
}

**File : main.dart**

import 'package:flutter/material.dart';  
import 'package:provider/provider.dart';  
import 'package:todoapp/provider/dbprovider.dart';  
import 'package:todoapp/ui/todolistpage.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) {  
 // nunggu data dbprovider  
 return ChangeNotifierProvider(  
 create: (context)=>DbProvider(),  
 child: MaterialApp(  
 title: 'To Do App',  
 theme: ThemeData(  
 primarySwatch: Colors.*blue*,  
 visualDensity: VisualDensity.*adaptivePlatformDensity*,  
 ),  
 home: const ToDoListPage(),  
 ),  
 );  
 }  
}