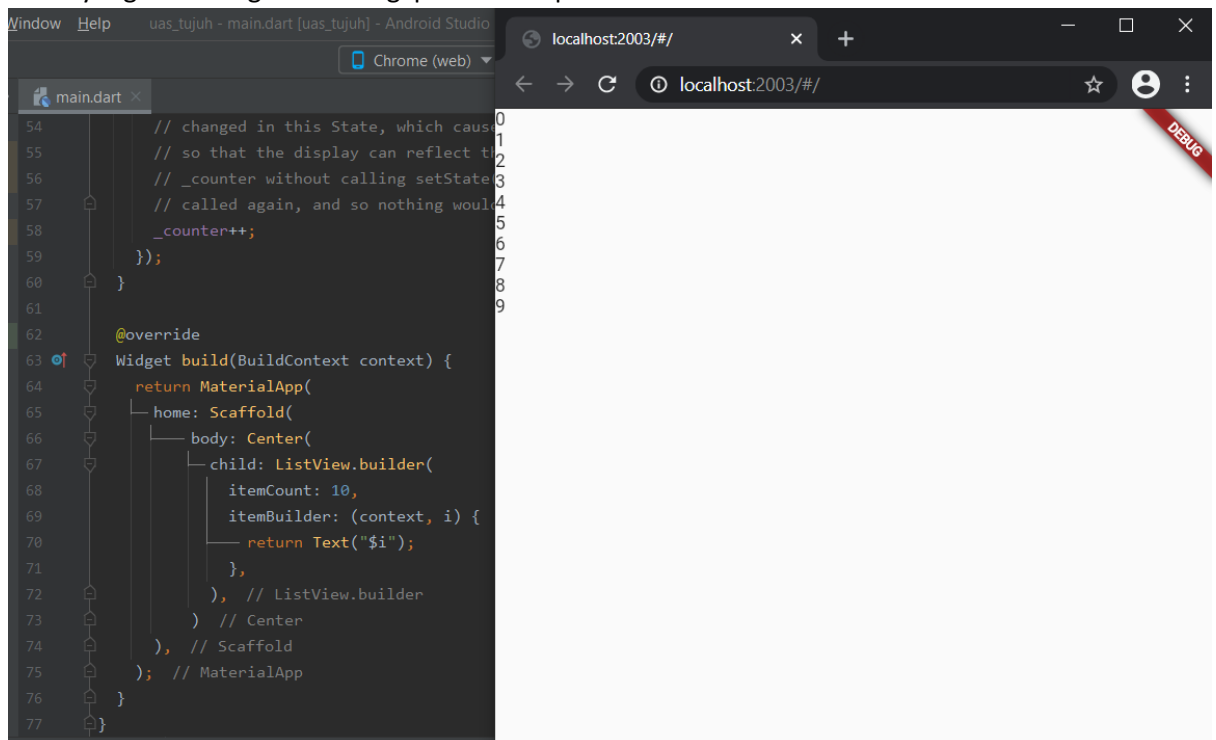


Nama : Muhamad Fauzan
Nim : 181011402222
Kelas : 06TPLE022

UAS MOBILE PRORAMMING

Jawaban

1. Mobile Programming adalah pemrograman untuk membuat aplikasi pada perangkat mobile seperti smartphone dll.
2. User Interface (UI) merupakan bagian visual dari website ataupun aplikasi yang dapat berupa teks box, radio button, drop-down list, dan komponen lain yang memungkinkan pengguna untuk memasukkan data ke dalam sistem perusahaan.
3. API atau Application Programming Interface adalah sebuah interface yang dapat menghubungkan aplikasi satu dengan aplikasi lainnya.
4. Perbedaanya hanyalah dibagian platformnya saja jika Nativehanya bisa digunakan di salah satu platform sedangkan hybrid justru dapat digunakan di berbagai platform
5. Gaada soal
6. GitHub adalah manajemen proyek dan sistem versioning code sekaligus platform jaringan sosial yang dirancang khusus bagi para developer



- 7.
8. 4 times two is 8
4 times two four is 16
2 x 2 x 2 is 8
9. Tahap Pertama kita akan membuat file halamanJson.dart dalam project kita. Kemudian isi dengan kode di bawah ini.

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

void main() {
  runApp(new MaterialApp(
```

```

    title: "My Apps",
    home: new HalamanJson(),
  ));
}

```

```

class HalamanJson extends StatefulWidget {
  @override
  _HalamanJsonState createState() => _HalamanJsonState();
}

```

```

class _HalamanJsonState extends State {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Data JSON"),
      ),
      drawer: DrawerApp(),
      body: Center(
        child: Text("Data JSON")
      ),
    );
  }
}

```

Selanjutnya kita membutuhkan beberapa package diantaranya sebagai berikut. Tambahkan http dependencies dalam file pubspecs.yaml

```

dependencies:
  flutter:
    sdk: flutter

```

```

http: ^0.12.0+1

```

Lalu import dalam file .dart

```

import 'dart:convert';
import 'package:http/http.dart' as http;
import 'dart:async';

```

Selanjutnya kita akan menggunakan **Future** untuk menjalankan http.get.

```

List datadariJSON;

```

```

Future ambildata() async {
  http.Response hasil = await http.get(
    Uri.encodeFull("https://jsonplaceholder.typicode.com/users"),
    headers: {"Accept": "application/json"});

```

```

    this.setState(() {
      datadariJSON = json.decode(hasil.body);
    });
}

```

Pada contoh **Future** di atas, sebelumnya kita telah membuat List terlebih dahulu yang bernama **datadariJSON** yang dimaksudkan akan menampung data yang akan diparsing dari url. Jadi sebelum Future **ambildata** dijalankan maka List **datadariJSON** masih bernilai **null**. **setState** di sini berfungsi untuk merubah state dari **datadariJSON** yang tadinya null menjadi berisi data dari hasil parsing. Lalu untuk menjalankan Future ambil data kita menggunakan **initState**.

```
@override
void initState() {
  this.ambildata();
}
```

Singkatnya Full Code-nya akan menjadi seperti di bawah ini. Di sini saya menampilkan List **datadariJSON** menggunakan **ListView.builder** dimana setiap ListTile nya dapat diubah secara custom sesuai keinginan masing-masing.

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:async';
```

```
void main() {
  runApp(new MaterialApp(
    title: "My Apps",
    home: new HalamanJson(),
  ));
}
```

```
class HalamanJson extends StatefulWidget {
  @override
  _HalamanJsonState createState() => _HalamanJsonState();
}
```

```
class _HalamanJsonState extends State {
  List datadariJSON;
```

```
Future ambildata() async {
  http.Response hasil = await http.get(
    Uri.encodeFull("https://jsonplaceholder.typicode.com/users"),
    headers: {"Accept": "application/json"});
```

```
  this.setState(() {
    datadariJSON = json.decode(hasil.body);
  });
}
@override
void initState() {
  this.ambildata();
}
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
```

```

appBar: AppBar(
  title: Text("Data JSON"),
),
body: Container(
  child: ListView.builder(
    itemCount: datadariJSON == null ? 0 : datadariJSON.length,
    itemBuilder: (context, i){
      return ListTile(
        title: Text(datadariJSON[i]['name']),
      );
    }
  ),
),
);
}
}

```

Saat menampilkan data dari List kita seperti menampilkan array pada umumnya dan disesuaikan dengan struktur dari JSON yang tersedia. Contohnya di sini menampilkan **name** dengan cara **datadariJSON[i]['name']**. Contoh lain jika ingin menampilkan nama jalan maka kita gunakan cara **datadariJSON[i]['address']['street']**.