UAS

MOBILE PROGRAMMING

Nama : Rendi Arif Hermawan

NIM : 181011401698

Kelas : 06TPLE017

1. Mobile Programming adalah pemograman yang ditujukan untuk pembuatan aplikasi diperangkat mobile. Banyak sekali platform mobile yang dapat kita coba,diantaranya iOS,BB RIM,J2ME,QT Mobile,Symbian,dan adroid.
2. UI (User Interface) adalah tampilan visual sebuah produk yang menjembatani system dengan pengguna (user).
3. API adalah sebuah antarmuka yang digunakan untuk menghubungkan antara satu aplikasi dengan aplikasi yang lain. Fungsi atau peran API adalah sebagai perantara yang menghubungkan aplikasi berbeda,baik dari platform yang sama maupun lintas platform. Fungsi yang lain yaitu membantu beban kerja dari server,mengembangkan aplikasi lebih cepat dan efektif,dan menciptakan aplikasi yang bersifat fungsional.
4. Aplikasi Native adalah aplikasi yang dibangun dengan bahasa pemograman yang spesifik dan hanya dapat digunakan diplatform tertentu. Aplikasi Ntive bisa disebut juga dengan aplikasi asli. Sedangkan aplikasi Hybrid adalah aplikasi yang dibuat menggunakan bahasa pemograman web dengan bantuan software development kit (SDK) native dari berbagai platform system operasi.
5. Aplikasi dengan berbasis website ini berfungsi untuk membantu menyimpan repository. Namun tak hanya itu ,dengan aplikasi ini anda juga dapat mengerjakan project tertentu agar dapat melakukan kolaborasi dalam mengerjakan project tertentu agar dapat terus melakukan update secara rutin,termasuk melacak dan menyimpan perubahan-perubahan yang terjadi dalam project anda tersebut.
6. **A. Source Code** body: ListView.builder( itemCount: 10, itemBuilder: (context, i) { return Text("$i");

},

**B. Output**

0

1

2

3

4

5

6

7

8

1. **A. Source Code** int timesTwo(int x) { return

int timesFour(int x) => timesTwo(timesTwo(x)); int runTwice(int x, int Function(int) f) {

for (var i=0;i<2;i++) { x = f(x);

}

return x;

}

void main() { print("4 times two is

${timesTwo(4)}"); print("4 times four is

${timesFour(4)}"); print("2 x 2 x 2 is

${runTwice(2, timesTwo)}");

}

**B. Output**

4 times two is 8

4 times four is 16

1. x 2 x 2 is 8

1.Pada contoh kali ini saya akan menggunakan sampel data json dari [https://jsonplaceholder.typicode.com/users/.](https://jsonplaceholder.typicode.com/users/) Jika dilihat struktur datanya kira-kira seperti di bawah ini. Total data ada 10.

[

{

"id": 1,

"name": "Leanne Graham", "username": "Bret",

"email": "[Sincere@april.biz](mailto:Sincere@april.biz)", "address": {

"street": "Kulas Light", "suite": "Apt. 556", "city": "Gwenborough", "zipcode": "92998-3874", "geo": {

"lat": "-37.3159",

"lng": "81.1496"

}

},

"phone": "1-770-736-8031 x56442",

"website": "hildegard.org", "company": {

"name": "Romaguera-Crona",

"catchPhrase": "Multi-layered client-server neural-net", "bs": "harness real-time e-markets"

}

},

{

"id": 2,

"name": "Ervin Howell", "username": "Antonette", "email": "[Shanna@melissa.tv](mailto:Shanna@melissa.tv)", "address": {

"street": "Victor Plains", "suite": "Suite 879", "city": "Wisokyburgh", "zipcode": "90566-7771", "geo": {

"lat": "-43.9509",

"lng": "-34.4618"

}

},

"phone": "010-692-6593 x09125",

"website": "anastasia.net", "company": {

"name": "Deckow-Crist",

"catchPhrase": "Proactive didactic contingency", "bs": "synergize scalable supply-chains"

}

},

Tahap Pertama membuat file halamanJson.dart dalam project.

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")

),

);

}

}

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';

Gunakan **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);

});

}

Untuk menjalankan Future ambil data menggunakan **initState**

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

}

Singkatnya Full Code-nya akan menjadi seperti di bawah ini

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']),

);

}

),

),

);

}

}