

UAS

MOBILE PROGRAMMING

Ade Putra Prima Suhendri, S.Kom, M.Kom

Nama : Novita Putri

Kelas : 06TPLE011

NIM : 181021400190

1. Jelaskan apa yang dimaksud dengan Mobile Programming?

Seperti namanya Mobile Programming adalah **pemrograman** yang diterapkan di mobile/Handphone atau Laptop. Macam alat mobile yaitu antara lain : Pengembangan aplikasi mobile adalah proses dimana aplikasi perangkat lunak yang dikembangkan untuk daya rendah perangkat genggam , seperti asisten pribadi digital , perusahaan asisten digital atau ponsel .

2. Jelaskan apa yang dimaksud dengan User Interface(UI)?

User Interface adalah **tampilan visual sebuah produk yang menjembatani sistem dengan pengguna** (user). Tampilan UI dapat berupa bentuk, warna, dan tulisan yang didesain semenarik mungkin. Secara sederhana, UI adalah bagaimana tampilan sebuah produk dilihat oleh pengguna. Analoginya bisa dijelaskan seperti ini.

3. Jelaskan apa yang dimaksud dengan API? jelaskan fungsinya!

API didefinisikan sebagai spesifikasi kemungkinan prediksi interaksi dengan komponen perangkat lunak. Sederhananya **API** bisa dicontohkan sebagai sebuah kendaraan yang berisi komponen perangkat lunak. **API** akan mencakup informasi tentang apa yang dapat dilakukannya yakni mempercepat, mengerem, menyalakan radio, dan lain- lain.

Application Programming Interface atau API berfungsi menyediakan *function* dan juga perintah dengan bahasa yang lebih terstruktur dan lebih mudah bagi dipahami bagi *programmer*.

4. Jelaskan perbedaan Native dan Hybrid pada mobile programming

Aplikasi hybrid dinilai lebih mudah dan cepat untuk dikembangkan dibanding dengan native app. Namun **kecepatan aplikasi hybrid bekerja lebih lambat** daripada aplikasi native karena bergantung pada kecepatan browser user. Jika Anda membutuhkan jasa pengembangan aplikasi mobile, Anda dapat menghubungi Logique Digital Indonesia.

5. Jelaskan apa fungsi github

Git adalah inti atau jantung **GitHub**. **Git** merupakan sistem pengontrol versi yang dikembangkan oleh Linus Torvalds (yang juga menciptakan dan mengembangkan Linux). Github berfungsi untuk **menyimpan repository** yang tadi sudah saya jelaskan diatas dan juga sebagai **alat kolaborasi dalam pengerjaan suatu project**.

6. Apa output dari script berikut?

Text(1)
Text(2)
Text(3)
Text(4)
Text(5)
Text(6)
Text(7)
Text(8)
Text(9)
Text(10)

7. Apa output dari script berikut !

8
16
8
4 dst ditulis 8
4 dst ditulis 16
2 x 2 di tulis 8

8. Apa output dari script berikut

```
[  
{  
  "id": 1,  
  "name": "Leanne Graham",  
  "username": "Bret",  
  "email": "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",
  "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 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.yam

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 data dari JSON menggunakan ListView.builder dimana setiap ListTile nya dapat diubah secara custom sesuai keinginan masing-masing. Untuk melihat contoh ListView custom

```
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']`