

Nama : Eko Febriansyah

NIM : 181011402157

Kelas : 06TPLE022

#### UAS Mobile Programming.

1. Jelaskan apa yang dimaksud dengan Mobile Programming?  
Mobile Programming adalah pemrograman yang digunakan untuk perangkat mobile
2. Jelaskan apa yang dimaksud dengan User Interface (UI)?  
User Interface adalah tampilan visual sebuah produk yang menjembatani sistem dengan pengguna (user). Secara sederhana, UI adalah bagaimana tampilan sebuah produk dilihat oleh pengguna.
3. Jelaskan apa yang dimaksud dengan API? jelaskan fungsinya!  
API atau Application Programming Interface adalah sebuah interface yang dapat menghubungkan aplikasi satu dengan aplikasi lainnya. Jadi, API berperan sebagai perantara antar berbagai aplikasi berbeda, baik dalam satu platform yang sama atau lintas platform.
4. Jelaskan perbedaan Native dan Hybrid pada mobile programming?  
Aplikasi hybrid adalah aplikasi web yang ditransformasikan menjadi kode native pada platform seperti iOS atau Android.  
Aplikasi native adalah aplikasi yang dibangun dengan bahasa pemrograman yang spesifik untuk platform tertentu. Contoh populernya yakni penggunaan bahasa pemrograman Objective-C atau Swift untuk platform iOS (Apple). Adapun platform Android yang menggunakan bahasa pemrograman Java.
5. Jelaskan apa fungsi github!  
GitHub adalah layanan hosting web bersama untuk proyek pengembangan perangkat lunak yang menggunakan sistem kendali versi Git dan layanan hosting internet.
6. Output!



## 7. Output!

## 8. Parsing JSON pada Flutter!

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']**.