

Nama : Fauzan Ardiansyah
NIM : 181011402194
Kelas : 06TPLE022

Jawaban

1. Mobile programming adalah proses pembuatan aplikasi untuk perangkat mobile baik aplikasi yang bersifat offline maupun online.
2. User Interface adalah tampilan visual sebuah produk yang menjembatani sistem dengan pengguna (user).
3. API atau Application Programming Interface adalah sebuah interface yang dapat menghubungkan aplikasi satu dengan aplikasi lainnya, dan berikut adalah fungsinya :
 - a. Memudahkan Membangun Aplikasi yang Fungsional
 - b. Pengembangan Aplikasi Menjadi Lebih Efisien
 - c. Meringankan Beban Server
4. Berikut perbedaan mobile programming native dan hybrid :
 - a. **Aplikasi native** adalah aplikasi yang dibangun dengan bahasa pemrograman yang spesifik untuk platform tertentu.
 - b. Aplikasi hybrid adalah aplikasi web yang ditransformasikan menjadi kode native pada platform seperti iOS atau Android.
5. Github adalah sebuah layanan host website yang dirancang khusus untuk developer atau programmer. Fungsi utama github adalah untuk memudahkan manajemen proyek.

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
9

7. **A. Source Code**

```
int timesTwo(int x) {
```

```

    return x * 2;
}

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
2 x 2 x 2 is 8

```

8. Pada contoh kali ini saya akan menggunakan sampel data json dari <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",
    "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 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']),
                        );
                    }
                ),
            ),
        );
    }
}

```