

UAS  
MOBILE PROGRAMMING  
Ade Putra Prima Suhendri, S.Kom, M.Kom

Nama : Rayhan Fauzan   NIM : 181021400090   Kelas : 06TPLE011

---

SOAL

1. Jelaskan apa yang dimaksud dengan Mobile Programming? Point 5
2. Jelaskan apa yang dimaksud dengan User Interface (UI)? Point 5
3. Jelaskan apa yang dimaksud dengan API? jelaskan fungsinya! Point 5
4. Jelaskan perbedaan Native dan Hybrid pada mobile programming? Point 5
5. Jelaskan apa fungsi github! Point 5
6. Apa output dari script berikut ! Point 10:

```
ListView.builder(  
  itemCount: 10,  
  itemBuilder: (context, i){  
    return Text("$i");  
  },  
);
```

Apa output dari script berikut ! Point 10:

```
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)}");  
}
```

Tuliskan sintak cara parsing JSON pada flutter ! Poin 55

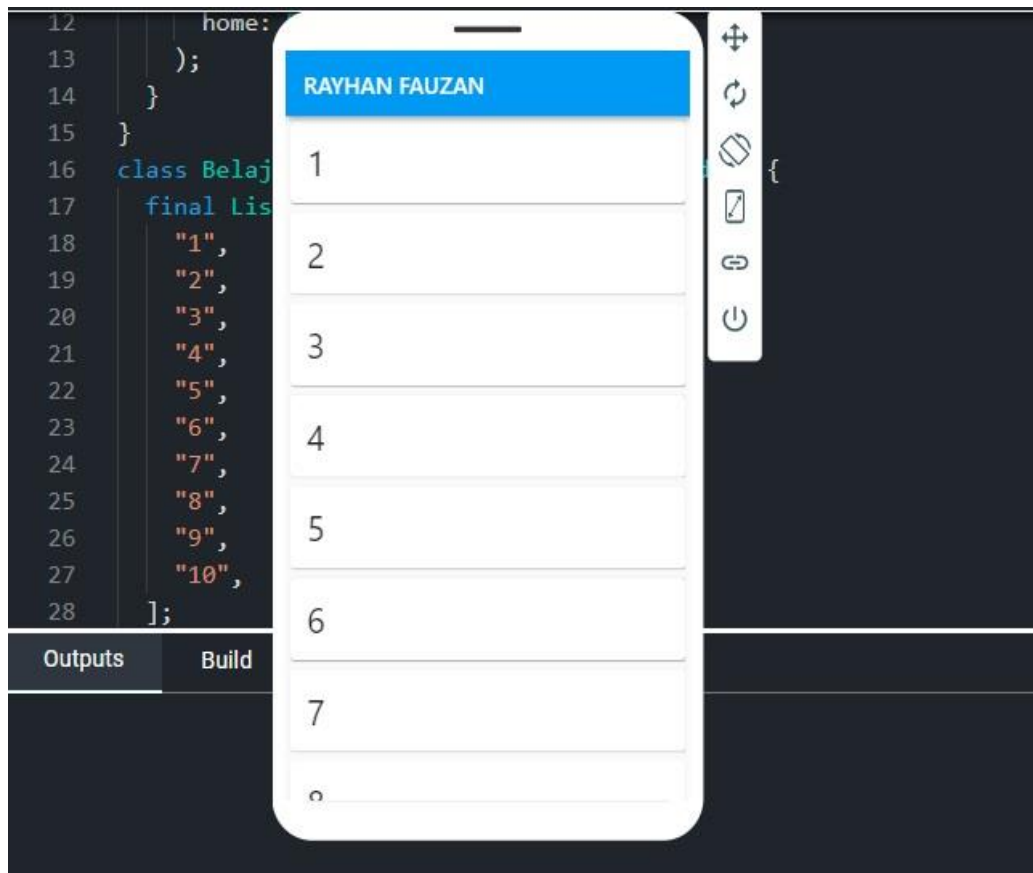
## JAWABAN

1. Mobile Programming adalah pemrograman yang ditujukan untuk pembuatan aplikasi di perangkat mobile. Mobile Programming juga pemrograman yang diterapkan di Mobile atau Laptop.
2. User Interface merupakan sebuah tampilan visual dari sebuah produk yang berfungsi untuk menjembatani sistem dengan user atau pengguna. Dimana tampilan UI bisa berupa warna, bentuk serta tulisan yang didesain semenarik mungkin. Tampilan UI biasanya didesain dari beberapa aspek yaitu gambar/logo, layout, pemilihan warna dan juga typography.
3. *Application Programming Interface* atau API adalah seperangkat antarmuka yang dapat berupa fungsi, method, maupun URL endpoint yang dapat digunakan untuk mengembangkan aplikasi baik dalam satu platform maupun lintas platform, API dapat menerima permintaan dari perangkat lunak lainnya seperti mobile apps / website yang memberikan layanan pertukaran data, dengan gunakan API memungkinkan developer untuk mengintegrasikan dan mengizinkan dua aplikasi yang berbeda saling terhubung secara bersamaan.

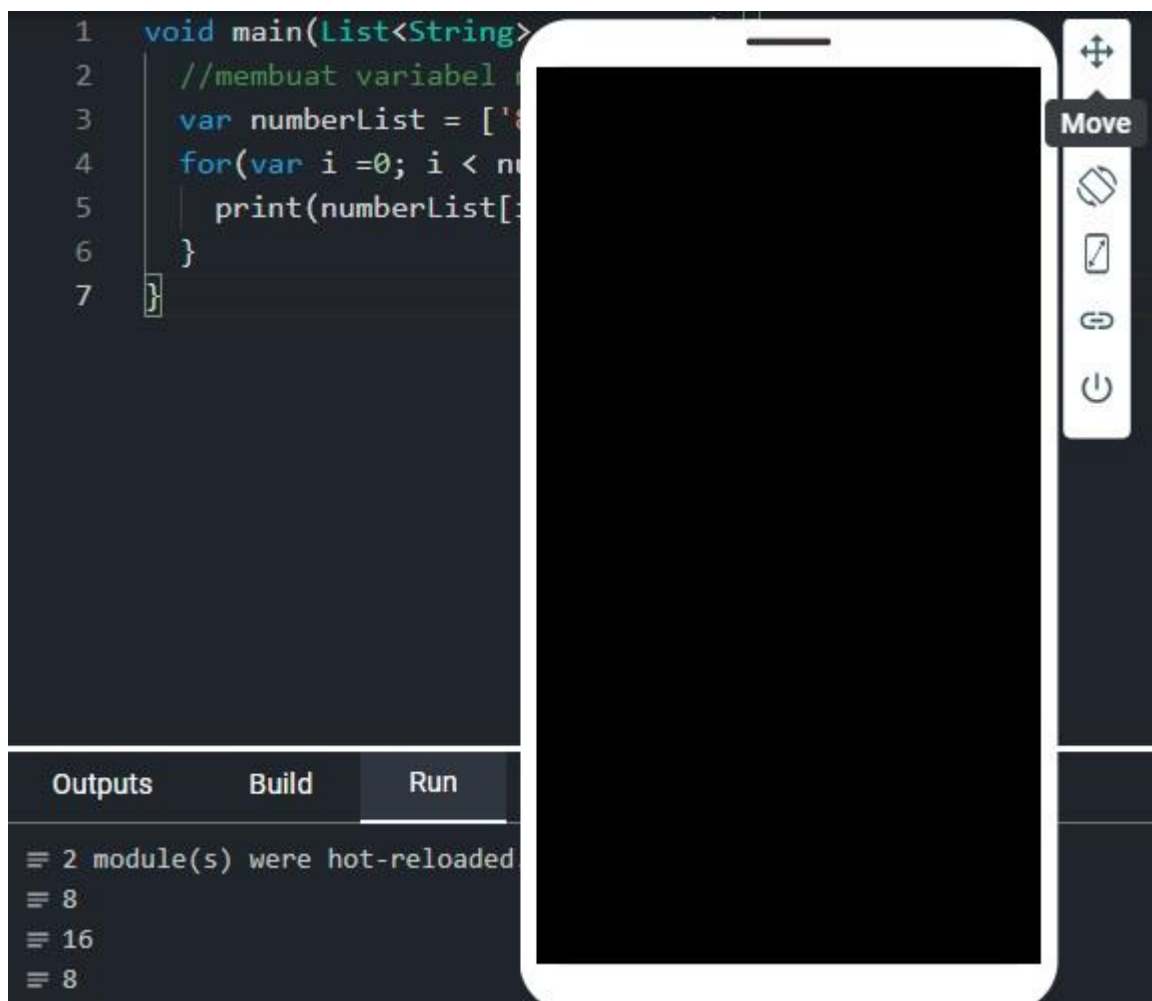
Fungsi API:

- 1) **Memudahkan membangun aplikasi yang fungsional:** Dengan menggunakan API akan lebih mudah untuk membuat aplikasi yang fungsional dan kompleks. Tanpa perlu menambahkan data secara manual, aplikasi yang dikembangkan akan memiliki fitur dari aplikasi tujuan.
  - 2) **Pengembangan aplikasi menjadi lebih efisien:** Dengan adanya API kita tidak perlu untuk melakukan komunikasi langsung dengan aplikasi lain yang ingin dihubungkan. Cukup dengan komunikasi melalui API hal ini sangat membantu, terutama jika anda ingin membangun aplikasi lintas platform (Cross Platform) dengan berbagai layanan sekaligus.
  - 3) **Meringankan beban server:** Dengan menggunakan API kita tidak perlu menyimpan semua data yang dibutuhkan di server kita sendiri, cukup meminta API untuk mendapatkan data terbaru dari server aplikasi asal. Dengan kondisi seperti ini server tidak akan terbebani dan pada akhirnya mengurangi resiko website tidak dapat diakses karena server yang down.
4. **Native** di Mobile Programming adalah sebuah aplikasi yang dibangun dengan bahasa pemrograman yang spesifik untuk platform-platform tertentu. Contoh populernya penggunaan bahasa pemrograman Objective-C atau swift untuk platform Ios (apple) adapun platform android yang menggunakan bahasa pemrograman seperti hal-nya (JAVA).
- Hybrid** di Mobile Programming adalah aplikasi web yang di transformasikan menjadi kode native pada platform seperti ios atau android. Aplikasi hybrid biasanya menggunakan browser untuk mengizinkan aplikasi web mengakses berbagai fitur di device mobile seperti hal-nya Push Notification, Contacts, atau Offline Data Storage. Beberapa tools untuk mengembangkan aplikasi hybrid antara lain yaitu, Phonegap, Rubymotion dan lain- lain.
5. Fungsi utama dari Github adalah membantu menyimpan dan mengawasi repository bahkan merencanakan dan melacak proses kerja dari suatu project yang sedang dilakukan.

6. Berikut outputnya



7. Berikut outputnya



Ket: 4x2: 8

4x4: 16

2x2x2: 8

```
8. [
  {
    "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",
```

```

import 'package:flutter/material.dart';

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

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

  "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.

```

}

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 masingmasing. Untuk melihat contoh Listview custom



```
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

import 'dart:convert';
```

```

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']**.

