

1. Jelaskan apa yang dimaksud dengan Mobile Programming?

Mobile Programming adalah pemrograman yang digunakan untuk perangkat mobile. Mobile Programming adalah pemrograman aplikasi mobile bisa menggunakan J2ME, mobile merupakan aplikasi yang bisa jalan di handphone seperti PDA atau yang handphone yang berbasis java.

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

User interface (antar muka pengguna) merupakan bentuk tampilan yang berfungsi untuk menghubungkan antara pengguna (user) dengan aplikasi atau sistem operasi, sehingga pengguna (user) dapat mengerti dan berkomunikasi. User interface dapat berupa teks maupun grafis. Untuk aplikasi yang menekankan pada kecepatan proses dan ditujukan untuk pengguna (user) dengan pengetahuan lebih baik, biasanya dibuat bentuk teks. Sedangkan untuk aplikasi yang ditujukan untuk pengguna akhir (end user) dan lebih menekankan user friendly, maka digunakan bentuk grafis.

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

API adalah antarmuka komputasi yang mendefinisikan interaksi antara beberapa perantara perangkat lunak. Dengan cara mendefinisikan jenis panggilan atau permintaan yang dapat dibuat, bagaimana membuatnya, format data yang harus digunakan, konvensi yang harus diikuti, dan lain- lain. API juga dapat menyediakan mekanisme ekstensi sehingga pengguna dapat memperluas fungsionalitas yang ada dengan berbagai cara dan ke berbagai tingkatan.

Beberapa fungsi API yang umum bagi setiap pengembang yang menggunakannya :

1. Jika Anda ingin menyematkan peramban web untuk menampilkan satu atau beberapa laman web, misalnya, Anda tidak harus memprogram peramban web Anda sendiri dari awal hanya untuk aplikasi Anda. Anda dapat menggunakan WKWebView API untuk menyematkan objek browser WebKit (Safari) di aplikasi Anda.
2. Jika Anda ingin mengambil foto atau video dari kamera iPhone, Anda tidak perlu menulis antarmuka kamera Anda sendiri. Anda menggunakan API kamera untuk menanamkan kamera bawaan iPhone di aplikasi Anda. Jika API tidak ada untuk mempermudah metode ini, pengembang aplikasi harus membuat perangkat lunak kamera sendiri dan menginterpretasikan input perangkat keras kamera.

4. Jelaskan perbedaan Native dan Hybrid pada mobile programming?

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.

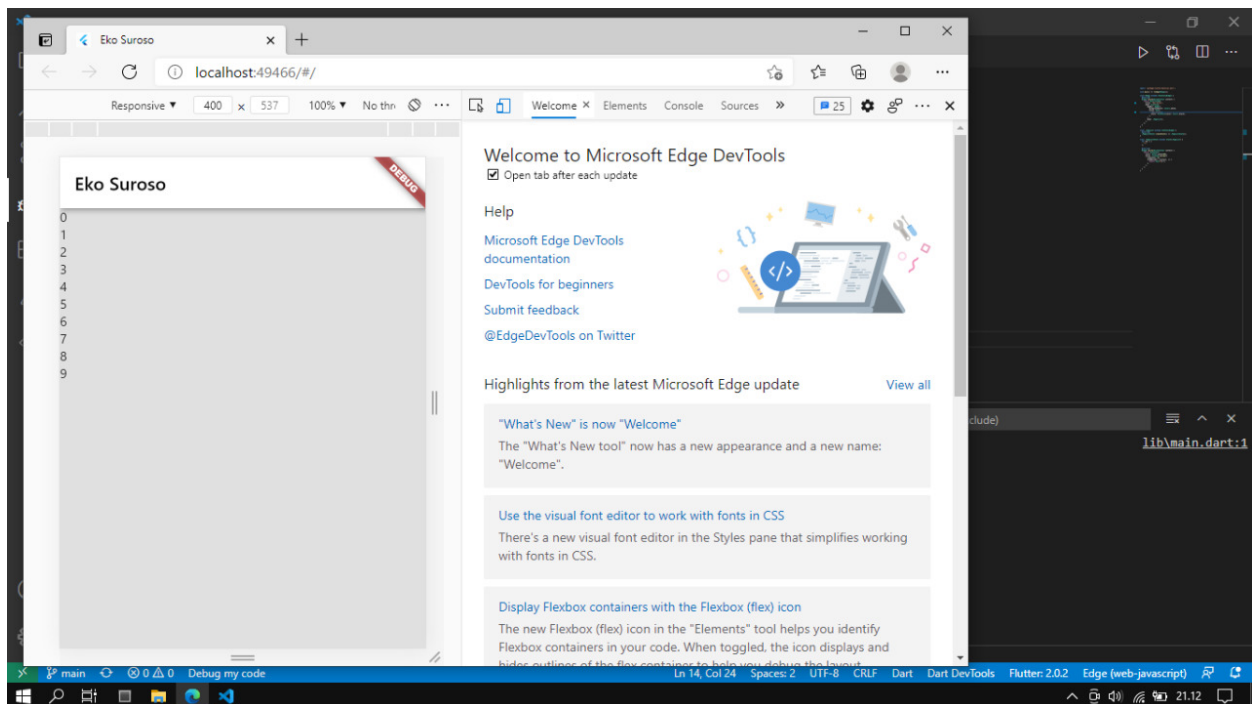
Hybrid adalah aplikasi web yang ditransformasikan menjadi kode native pada platform seperti iOS atau Android. Aplikasi hybrid biasanya menggunakan browser untuk memungkinkan aplikasi web mengakses berbagai fitur di device mobile seperti Push Notification, Contacts, atau Offline Data Storage. Beberapa tools untuk mengembangkan aplikasi hybrid antara lain Phonegap, Rubymotion dan lain-lain.

6. Jelaskan apa fungsi github!

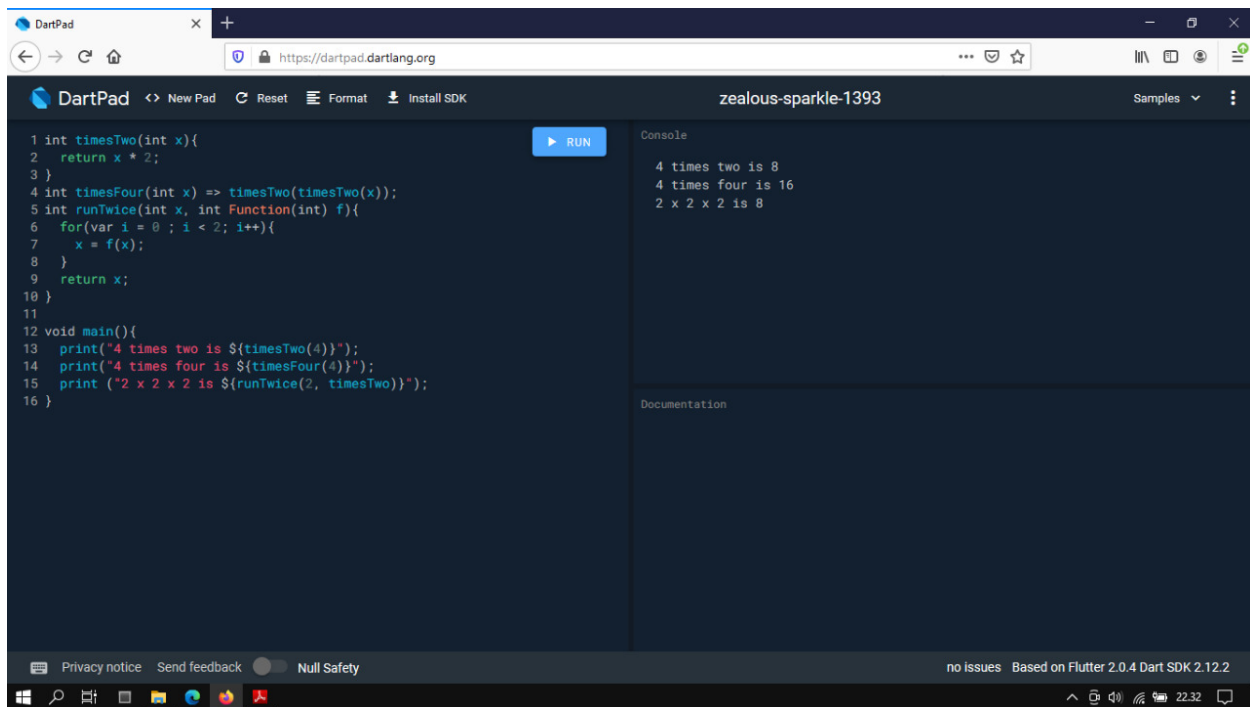
GitHub adalah manajemen proyek dan sistem versioning code sekaligus platform jaringan sosial yang dirancang khusus bagi para developer. GitHub berfungsi untuk membantu penyimpanan repository. Namun tak hanya itu, dengan aplikasi ini Anda juga dapat melakukan kolaborasi dalam mengerjakan project tertentu agar dapat terus melakukan update secara rutin, termasuk melacak dan menyimpan perubahan-perubahan yang terjadi dalam project Anda tersebut.

Aplikasi ini tidak hanya dapat digunakan oleh para programmer atau orang-orang yang sedang memiliki project saja. Aplikasi ini dapat digunakan secara luas oleh siapa saja, termasuk oleh Anda yang saat ini sedang mencari pekerjaan. Aplikasi ini memiliki laman profil yang akan memuat data pribadi Anda, seperti foto, email, bahkan followers yang Anda miliki. Informasi-informasi yang dimuat dalam profil Anda ini akan menarik pihak lain untuk mengajak Anda bekerjasama dalam sebuah project.

7. Apa output dari script berikut!



8. Apa output dari script berikut!



9. Tuliskan sintak cara parsing JSON pada flutter!

menampilkan data secara real-time dari server/website atau platform lainnya. Hal tersebut salah satunya karena adanya API yang menghubungkan data ke masing-masing platform tersebut. 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 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. Untuk melihat contoh Listview custom **di sini**.

```
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']**.