

LEMBAR JAWABAN

NAMA : HADI SAPUTRO

NIM : 181011400138

KELAS : O6TPLE010

MK : MOBILE PROGRAMING

Dosen:AdePutra PrimaSuhendri,S.kom.,M.kom

assalamualaikum,,,,,,

bismillahirrahmanirrahhim,,,

1.Mobile programming adalah proses pembuatan aplikasi untuk perangkat mobile baik aplikasi yang bersifat offline maupun online diperangkat mobile yang dapat kita buat dengan menggunakan Java.

2.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. Misalnya Anda mengunjungi sebuah rumah, Anda pasti akan melihat tampilan rumah tersebut, bukan? Anda bisa melihat halaman depan rumah, letak pintu utama, dan jendela.

Nah, tampilan rumah yang Anda lihat itu adalah user interface atau UI.

Tampilan UI ini diterapkan pada sistem operasi, aplikasi, website, maupun blog.

3.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.

Perumpamaan yang bisa digunakan untuk menjelaskan API adalah seorang pelayan di restoran. Tugas pelayan tersebut adalah menghubungkan tamu restoran dengan juru masak. Tamu cukup memesan makanan sesuai daftar menu yang ada dan pelayan memberitahukannya ke juru masak. Nantinya, pelayan akan kembali ke tamu tadi dengan masakan yang sudah siap sesuai pesanan.

fungsi dan manfaat api sbb

a. Memudahkan Membangun Aplikasi yang Fungsional

Dengan menggunakan API, akan lebih mudah untuk membuat aplikasi yang fungsional dan kompleks.

b. Pengembangan Aplikasi Menjadi Lebih Efisien

Dengan adanya API, Anda tidak perlu melakukan komunikasi langsung dengan aplikasi lain yang ingin dihubungkan.

c. Meringankan Beban Server

Dengan menggunakan API, Anda tidak perlu menyimpan semua data yang dibutuhkan di server Anda sendiri.

4. aplikasi native adalah aplikasi yang dibangun dengan bahasa pemrograman yang spesifik dan hanya dapat digunakan di platform tertentu. Aplikasi native bisa juga disebut aplikasi asli.

Salah satu kelebihan dari aplikasi native adalah UI/UX-nya yang sangat baik.

Sayangnya, aplikasi ini hanya dapat digunakan di satu platform dan biaya pengembangannya pun relatif tinggi.

IDE yang digunakan Android adalah Android Studio. Sementara itu, iOS menggunakan IDE iOS Xcode.

Salah satu perbedaan mencolok antara aplikasi native, hybrid, dan web adalah dari segi platform yang digunakan.

Jika aplikasi native hanya bisa digunakan di salah satu platform, aplikasi hybrid justru dapat digunakan di berbagai platform.

Secara definisi, aplikasi hybrid adalah aplikasi yang dibuat menggunakan bahasa pemrograman web dengan bantuan software development kit (SDK) native dari berbagai platform sistem operasi.

Salah satu kelebihan aplikasi hybrid adalah biasanya lebih mudah dan lebih cepat untuk dikembangkan.

Aplikasi ini juga membutuhkan lebih sedikit maintenance atau perawatan.

Namun, umumnya performa aplikasi hybrid belum bisa mengungguli aplikasi native.

6. apa saja fungsi GitHub? Dengan platform ini, Anda bisa bekerja bersama-sama dengan rekan dari berbagai belahan dunia, merencanakan proyek, dan bahkan tracking (melacak) pekerjaan Anda.

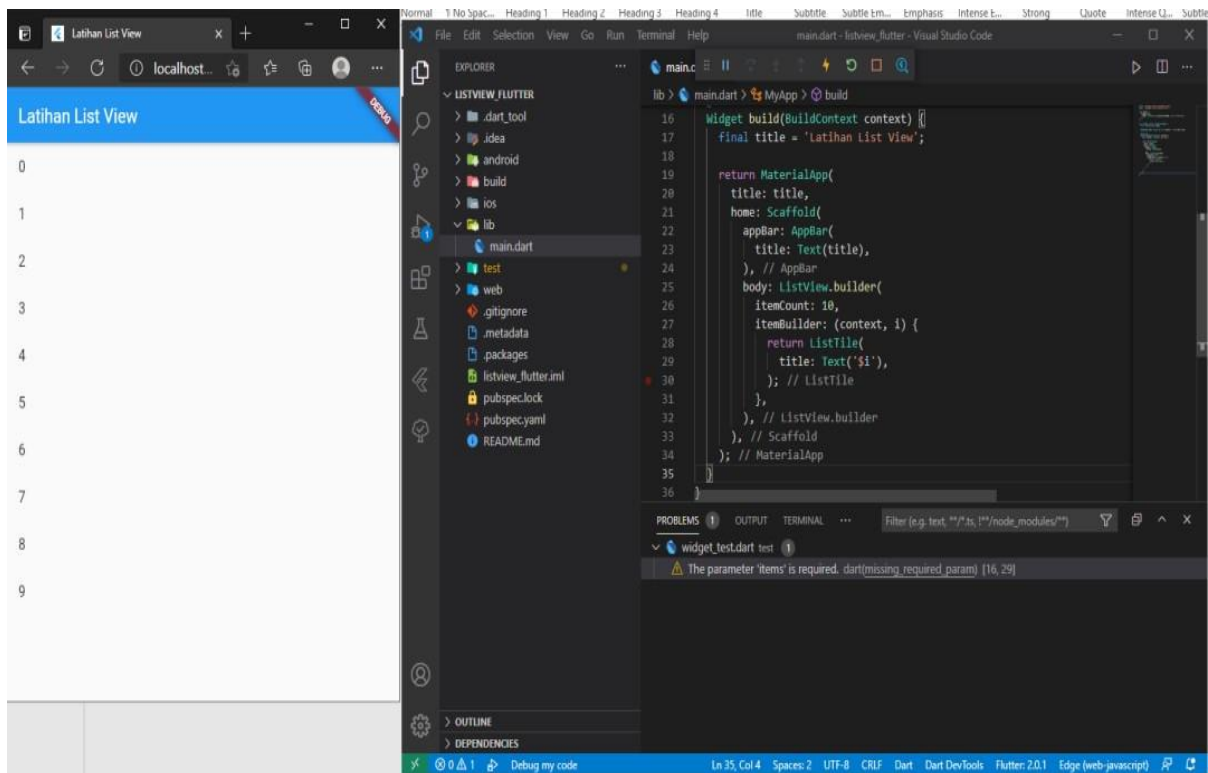
GitHub juga merupakan salah satu storehouse online terbesar di dunia untuk pekerjaan kolaborasi.

Jika Git adalah jantung, maka Hub adalah jiwa GitHub. Sistem Hub yang ada pada GitHub berfungsi untuk mengubah baris perintah (command line), seperti Git, menjadi jaringan media sosial terbesar bagi para developer.

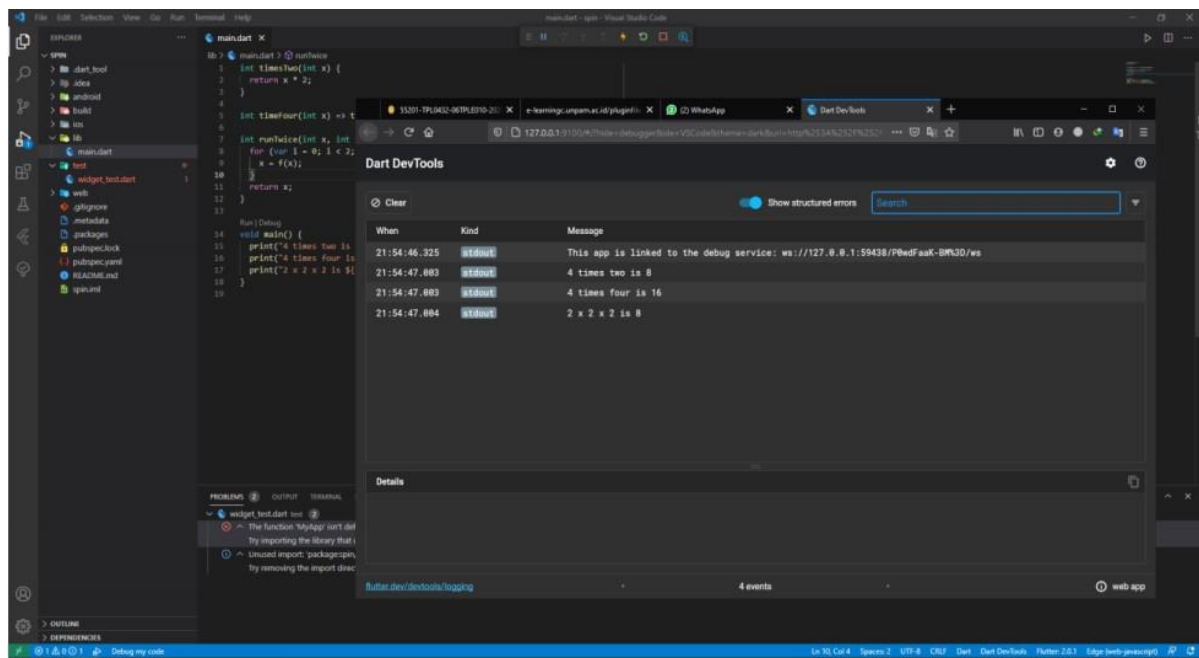
Selain berkontribusi dalam proyek tertentu, GitHub juga memungkinkan usernya untuk berkomunikasi dengan orang-orang yang memiliki kesamaan visi dan misi. Anda bahkan bisa follow mereka dan melihat proyek yang mereka kerjakan atau bahkan mencari tahu siapa saja yang terhubung dengan mereka.

7.

```
body: ListView.builder(  
  itemCount: 10,  
  itemBuilder: (context, i) {  
    return ListTile(  
      title: Text('$i'),  
    );  
  },  
),  
),  
);  
}  
}
```



8.



9. Pertama kita akan 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")
      ),
    );
  }
}

```

lalu 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);  
  
    });  
  
}
```

sebelumnya saya 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.

```
@override  
  
void initState() {  
  
    this.ambildata();  
  
}
```

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']),
                    );
                }
            ),
        ),
    );
}
}

```

di sini menampilkan name dengan cara datadariJSON[i]['name'].

contoh lain nya sbb;


```

class _HomeState extends State<Home> {
  List<UsersDetail> _list = [];
  var loading = false;
  Future<Null> _fetchData() async {
    setState(() {
      loading = true;
    });
    final response =
      get("https://jsonplaceholder.typicode.com/users");
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);
      setState(() {
        for (Map i in data) {
          _list.add(UsersDetail.fromJson(i));
        }
        loading = false;
      });
    }
  }
}

```

Terima kasih ,,,,

Mohon maaf lahir batin bila ada salah salah penulisan ya pa ,,,,

Wassalamualaikum wr.wb.