

UAS MOBILE PROGRAMMING

NAMA : MULIA MAULANA

NIM : 171011402177

KELAS : 08TPLE007

1. Yang dimaksud dengan **Pemrograman Mobile** adalah pemrograman yang ditujukan untuk pembuatan aplikasi diperangkat mobile. Banyak sekali platform mobile yang dapat kita coba, diantaranya iOS, BB RIM, J2ME, QT Mobile, Symbian, dan Android. Salah satunya yang sedang booming adalah Android Mobile. Android merupakan sistem operasi berbasis linux yang bahasa pemrograman aplikasinya dapat kita buat menggunakan java.
2. Yang dimaksud dengan **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.
3. Yang dimaksud dengan **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. Fungsi dari API itu sendiri antara lain :
 - 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.

- Pengembangan Aplikasi Menjadi Lebih Efisien

Dengan adanya API, Anda tidak perlu 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 dengan berbagai layanan sekaligus.

- Meringankan Beban Server

Dengan menggunakan API, Anda tidak perlu menyimpan semua data yang dibutuhkan di server Anda sendiri. Cukup meminta API untuk mendapatkan data terbaru dari server aplikasi asal. Dengan kondisi ini, server Anda tidak akan terbebani. Pada akhirnya, mengurangi resiko website tidak dapat diakses karena server down.

4. Perbedaan aplikasi Native dan aplikasi Hybrid adalah sebagai berikut:

- Aplikasi Native

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.

- Aplikasi Hybrid

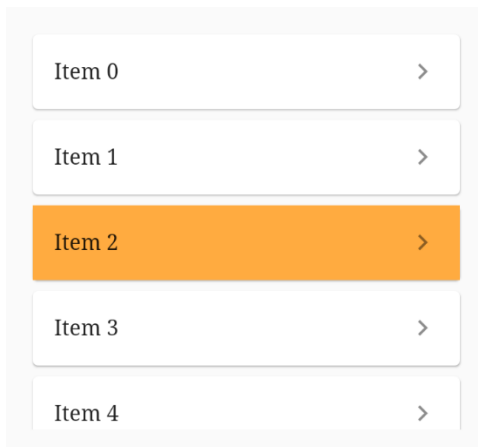
Aplikasi hybrid adalah aplikasi web yang ditransformasikan menjadi kode native pada platform seperti iOS atau Android. Aplikasi hybrid biasanya menggunakan browser untuk mengijinkan aplikasi web mengakses berbagai fitur di device mobile seperti Push Notification, Contacts, atau Offline Data Storage.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

Keuntungan membangun aplikasi hybrid diantaranya pemeliharaan project menjadi semakin mudah jika dibandingkan dengan aplikasi native. Aplikasi hybrid juga, bisa dibangun secara cepat untuk keperluan cross platform dan dana yang bisa menjadi lebih hemat jika dibandingkan dengan native.

5. Singkatnya, **GitHub** adalah manajemen proyek dan sistem *versioning code* sekaligus platform jaringan sosial yang dirancang khusus bagi para developer.

Fungsi dari **GitHub** itu sendiri adalah kita bisa bekerja bersama-sama dengan rekan dari berbagai belahan dunia, merencanakan proyek, dan bahkan *tracking* (melacak) pekerjaan yg sudah/sedang dikerjakan.

7. Outputnya adalah sebagai berikut



8. Berikut adalah penulisan JSON

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:async';

void main() {
  runApp(new MaterialApp(
    title: "Berkebun",
    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']),
                    );
                }
            ),
        ),
    );
}
}

```