

1 . **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.

Seperti namanya Mobile Programing adalah pemrogaman yang diterapkan di mobile/Handphone atau Laptop. Macam alat mobile yaitu antara lain :

1. Telepon Wireless
2. Tablet PC
3. Smartphone
4. Laptop
5. Handphone

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.

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.

Fungsi API

1. Memudahkan Membangun Aplikasi yang Fungsional
2. Pengembangan Aplikasi Menjadi Lebih Efisien
3. Meringankan Beban Server

4. –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.

Membangun aplikasi native harus menyediakan pengalaman produk yang optimal pada perangkat mobile. Meskipun begitu, budget yang tinggi dibutuhkan untuk membangun aplikasi cross platform yang mampu mempertahankan aplikasi native tetap update

-Hybrid

Aplikasi 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.

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. . **GitHub** adalah manajemen proyek dan sistem versioning code sekaligus platform jaringan sosial yang dirancang khusus bagi para developer. Dengan platform ini, Anda bisa bekerja bersama-sama dengan rekan dari berbagai belahan dunia, merencanakan proyek, dan bahkan tracking (melacak) pekerjaan Anda.

```
6. [
  {
    "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"
    }
},
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")
    ),
  );
}
}
dependencies:
  flutter:
    sdk: flutter

```

```

http: ^0.12.0+1
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'dart:async';
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();
}
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']),  
                    );  
                }  
            ),  
        ),  
    );  
}
```