

Nama : MOHAMAD KHATAMI AR  
NIM / Kelas : 181011402823 / 06TPLE022  
Mata Kuliah : UAS MOBILE PROGRAMMING

---

### 1. Jelaskan apa yang dimaksud dengan Mobile Programming?

Mobile Programming adalah pemrograman yang digunakan untuk perangkat mobile.

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

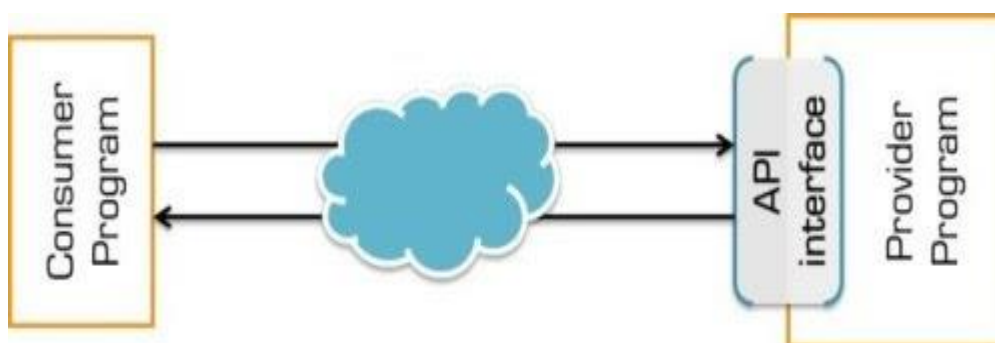
User interface (antarmuka pengguna) merupakan bentuk tampilan yang berfungsi untuk menghubungkan antara pengguna (user) dengan aplikasi atau system operasi, sehingga pengguna (user) dapat memahami dan berkomunikasi. User interface dapat berupa teks maupun grafis. Untuk aplikasi yang menekankan pada kecepatan 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.

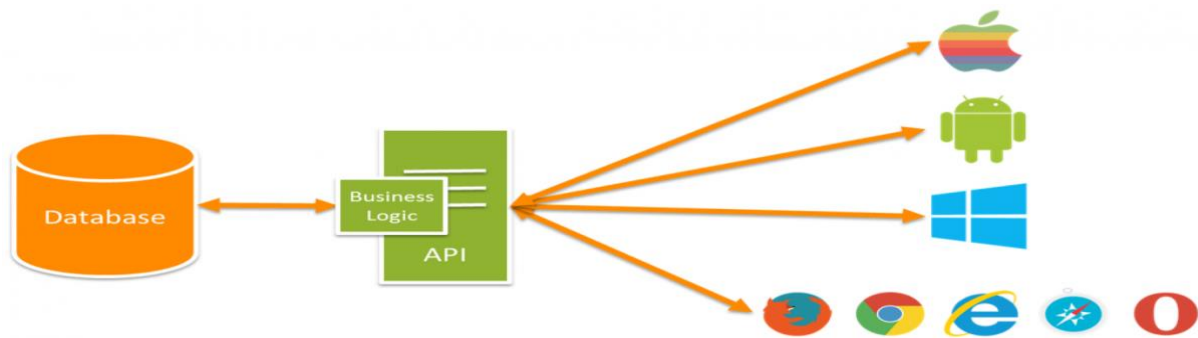
### 3. Jelaskan apa yang dimaksud dengan API? jelaskan fungsinya

API 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 memberitahunya ke juru masak. Nantinya, pelayan akan kembali ke tamu tadi dengan masakan yang sudah siap sesuai pesanan.

Itulah gambaran tugas dari API dalam pengembangan aplikasi.





Fungsi;

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

#### 4. Jelaskan perbedaan Native dan Hybrid pada mobile programming?

##### 1. Aplikasi Native

Aplikasi native adalah aplikasi mobile yang dikembangkan secara khusus untuk satu sistem operasi. Contohnya adalah pengembangan android hanya untuk android saja, atau Objective-C/Swift hanya untuk ios saja. Yang mana keduanya memiliki IDE (Integrated Development Environment) masing-masing. Untuk membuat aplikasi android, IDE yang digunakan adalah Android Studio. Sedangkan untuk membuat aplikasi ios, IDE yang digunakan adalah XCode. Untuk menggunakan produk aplikasi native, kita tinggal download di playstore untuk android dan appstore untuk ios.

Kelebihan :

1. UI dan UX yang alami sangat baik
2. Interaksi antar aplikasi sangat konsisten
3. Kualitas dan keamanannya terjamin aman

Kekurangan :

1. Pengembangan tidak mudah karena menggunakan bahasa API (Application Programming Interface) yang spesifik
2. Aplikasi hanya bekerja pada platform tertentu
3. Biaya pengembangan dan maintenance yang lebih mahal

##### 2. Aplikasi Hybrid

Aplikasi Hybrid adalah aplikasi yang pengembangannya menggabungkan antara aplikasi native dan aplikasi web mobile. Yang awalnya aplikasi web kemudian diubah dalam sebuah tool sehingga menjadi kode native. Beberapa tool untuk mengembangkan aplikasi hybrid antara lain Phonegap, Xamarin, Ionic dan lainnya. Aplikasi hybrid ini dikembangkan menggunakan HTML5 dan JavaScript. Untuk dapat menggunakannya kita dapat download di

masing-masing market platform. Inilah perbedaan utama antara aplikasi web dan aplikasi hybrid. Aplikasi web mobile tidak tersedia di market, sebaliknya hybrid sudah tersedia.

Kelebihan :

1. Dapat berfungsi dalam kondisi baik online maupun offline
2. Integrasi dengan file sistem perangkat
3. Integrasi dengan web-services
4. Embed dengan browser untuk meningkatkan akses ke konten online secara dinamis
5. Kinerja loading yang lumayan cepat
6. Hemat biaya perawatan dan pembuatan
7. Sebagian besar aset yang dibutuhkan oleh halaman web disimpan dalam paket aplikasi pada perangkat, bukan pada server

Kekurangan :

1. Biasanya pengembangan aplikasi hybrid harus menggunakan framework
2. Dalam sisi performa masih belum bisa melewati aplikasi native
3. Tidak mudah bagi pemula untuk mengembangkannya

## 6. Jelaskan apa fungsi github! Point 5

### Fungsi GitHub

Seperti yang sudah dijelaskan sebelumnya, aplikasi dengan basis website ini 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.

Secara lebih sederhana, beberapa fungsi dari **GitHub** adalah antara lain:

1. Dapat digunakan untuk menyimpan *repository*.
2. Menjadi wadah atau alat untuk kolaborasi dan mengerjakan sebuah *project*.
3. Memuat profil Anda sehingga dapat mengikuti programmer lainnya di aplikasi ini.
4. Dapat untuk mengawasi *repository* tertentu.

**7. Apa output dari script berikut ! Point 10:**



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

0

1

2

3

4

5

6

7

8

9

**8 . 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)}");
}

```

4 times two is 8

4 times two four is 16

2 x 2 x 2 is 8

## 8. Tuliskan sintak cara parsing JSON pada flutter! Poin 55

Class Your Model

```
{ String yourVar;
```

```
    YourModel({ this.yourVar});
```

```
    YourModel.fromJson(Map<String,dynamic>json){
```

```
        yourVar = json[key];
```

```
    }
```

```
}
```

```
//examplegetdatajsonfromAPI
```

```
final response = await client.post(uri,body: bodyString);
```

```
//decode json
```

```
Var responseJson = json.decode(response.body.toString());
```

```
//parsing json
```

```
final result = YourModel.fromJson(response);
```

## Parsing JSON pada Flutter

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']**.