

NAMA : ANGGIT BAGAS PAMINTO

NIM : 181011402443

MATKUL : MOBILE PROGRAMMING

UAS

MOBILE PROGRAMMING

Ade Putra Prima Suhendri, S.Kom, M.Kom

SOAL

1. Jelaskan apa yang dimaksud dengan Mobile Programming? Point 5
2. Jelaskan apa yang dimaksud dengan User Interface (UI)? Point 5
3. Jelaskan apa yang dimaksud dengan API? jelaskan fungsinya! Point 5
4. Jelaskan perbedaan Native dan Hybrid pada mobile programming? Point 5
6. Jelaskan apa fungsi github! Point 5
7. Apa output dari script berikut ! Point 10:



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)}");
}

```

9. Tuliskan sintak cara parsing JSON pada flutter ! Poin 55

JAWABAN :

1. Mobile programming adalah pemrograman yang di gunakan untuk perangkat mobile.
2. User Interface merupakan tampilan visual dari sebuah produk yang berfungsi untuk menjembatani sistem dengan user atau pengguna. Dimana tampilan UI bisa berupa warna, bentuk serta tulisan yang didesain dengan semenarik mungkin. Namun secara sederhana, UI dapat diartikan sebagai bagaimana tampilan suatu produk dilihat oleh pengguna atau user. Untuk lebih jelasnya, UI dapat dianalogikan berupa sebuah rumah yang sedang Anda kunjungi kemudian Anda akan melihat tampilan tersebut. Dimana tampilan yang bisa Anda lihat antara lain halaman depan, jendela, pintu dan dinding. Tampilan rumah yang Anda lihat itulah yang dinamakan dengan User Interface.

Biasanya tampilan UI diterapkan untuk sistem operasi, website, aplikasi dan 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.

Fungsi API : Membantu beban kerja pada server, Mengembangkan aplikasi lebih cepat & efektif, Menciptakan aplikasi yang bersifat fungsional.

4. Native : Di sini kita membuat aplikasi menggunakan bahasa pemrograman native yang sudah didukung penuh oleh penyedia platform yaitu Google dan Apple. Bahasa native tersebut berarti java untuk android dan objective C untuk iOS.

Hybrid : Dengan hybrid kita bisa membangun aplikasi android dan iOS secara mudah menggunakan teknologi HTML, CSS, dan Javascript yaitu menggunakan React Native sebagai frameworknya.

Kedua metode tersebut tentu saja memiliki keunggulan dan kelemahan.

- Jika ingin menggunakan native berarti kita harus mempelajari dua bahasa pemrograman sekaligus agar bisa membuat aplikasi yang berbeda platform.
- Begitupun dengan hybrid masalah utamanya biasanya pada performa dan memori, tapi kita cukup menguasai satu framework saja yang menggunakan teknologi di atas.

6. Fungsi utama github adalah membantu penyimpanan repository. Namun tak hanya sebatas itu saja, masih ada lebih banyak fungsi dari github untuk mendukung *project* yang Anda garap.

Beberapa fungsi github adalah:

- Memungkinkan Anda untuk berkolaborasi dengan orang lain;
- Menyimpan dan mengawasi *repository*;
- Merencanakan, menyimpan dan melacak proses kerja dari proyek;
- Berkomunikasi dengan sesama programmer;
- Melacak bug dan manajemen tugas. hingga;
- Menampilkan profil dan update dari Anda ke khalayak banyak.

7. Text(1)

Text(2)

Text(3)

Text(4)

Text(5)

Text(6)

Text(7)

Text(8)

Text(9)

Text(10)

8. 8

16

8

4 dst ditulis 8

4 dst ditulis 16

2 x 2 di tulis 8

9. [

```
{
  "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 data dari JSON;
```

```
Future ambil data() async {  
  http.Response hasil = await http.get(  
    Uri.encodeFull("https://jsonplaceholder.typicode.com/users"),  
    headers: {"Accept": "application/json"});  
  
  this.setState(() {  
    data dari JSON = json.decode(hasil.body);  
  });  
}
```

Pada contoh **Future** di atas, sebelumnya kita telah membuat List terlebih dahulu yang bernama **data dari JSON** yang dimaksudkan akan menampung data yang akan diparsing dari url. Jadi sebelum Future **ambil data** dijalankan maka List **data dari JSON** masih bernilai **null**. **setState** di sini berfungsi untuk merubah state dari **data dari JSON** yang tadinya null menjadi berisi data dari hasil parsing. Lalu untuk menjalankan Future ambil data kita menggunakan **initState**.

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

Singkatnya Full Code-nya akan menjadi seperti di bawah ini. Di sini saya menampilkan List **data dari JSON** menggunakan **ListView.builder** dimana setiap ListTile nya dapat diubah secara custom sesuai keinginan masing-masing. Untuk melihat contoh ListView custom

```
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']**.