

UAS  
MOBILE PROGRAMMING  
Ade Putra Prima Suhendri, S.Kom, M.Kom

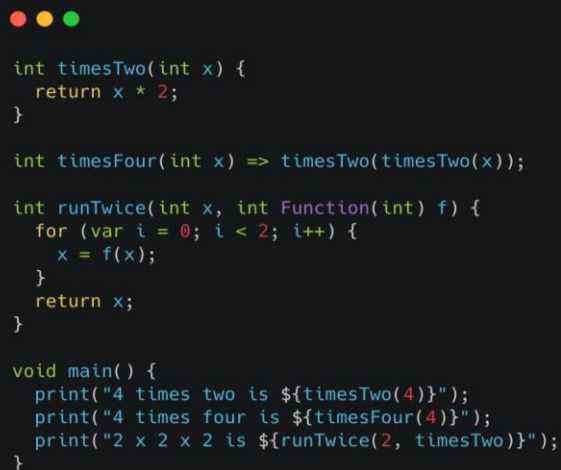
SOAL

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:



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

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



**UNIVERSITAS PAMULANG**

**KARTU UJIAN AKHIR SEMESTER GENAP 2020/2021**

**NOMOR UJIAN : 505793631778**

---

FAK/PROG : TEKNIK / TEKNIK INFORMATIKA

NAMA : PUTRI HANA SALSABILA

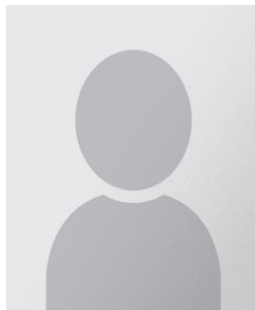
NIM : 181011401012

SHIFT : REGULER C

NO	HARI / TANGGAL	WAKTU	RUANG	KELAS	MATA KULIAH	PARAF
1	-			06TPLE011	KOMPUTER GRAFIK I	
2	-			06TPLE011	KERJA PRAKTEK	
3	-			06TPLE011	MOBILE PROGRAMMING	

**Peraturan dan Tata Tertib Peserta Ujian**

1. Peserta ujian harus berpakaian rapi, sopan dan memakai jaket Almamater
2. Peserta ujian sudah berada di ruangan sepuluh menit sebelum ujian dimulai
3. Peserta ujian yang terlambat diperkenankan mengikuti ujian setelah mendapat ijin, tanpa perpanjangan waktu
4. Peserta ujian hanya diperkenankan membawa alat-alat yang ditentukan oleh panitia ujian
5. Peserta ujian dilarang membantu teman, mencontoh dari teman dan tindakan-tindakan lainnya yang mengganggu peserta ujian lain
6. Peserta ujian yang melanggar tata tertib ujian dikenakan sanksi akademik



Pamulang, 02 Juni 2021  
Ketua Panitia Ujian

**UBAID AL FARUQ, S.Pd., M. Pd**  
**NIDK. 041802870**

## **JAWABAN :**

1. Mobile Programming adalah suatu proses pembuatan aplikasi berbasis mobile atau android, yang bersifat online ataupun offline. Dan berbagai jenis aplikasi yang kita gunakan dalam handphone kita adalah salah satu hasil akhir dari mobile programming ini.
2. User Interface adalah tampilan antar muka yang memiliki elemen yang mudah di akses serta digunakan oleh para pengguna, user interface di dalam nya terdapat elemen seperti teks, tombol, link, dan beberapa gambar. Peran User Intreface adalah sebagai penghubung agar memudahkan pengguna pada saat mengakses suatu program, web ataupun aplikasi.
3. API adalah Applcation Programming Interface, API adalah penghubung antara aplikasi suatu aplikasi yang kita buat dengan aplikasi lain nya. API ini bertujuan agar aplikasi yang berbeda dapat saling berbagi data dan berkomunikasi satu sama lain, dan semua itu dapat disimpan dalam bentuk library. Tidak hanya itu peran API bisa dibilang sangat berat karna API digunakan agar membuat sebuah tampilan menjadi lebih mudah digunakan, dan interaktif, selain itu API juga bisa digunakan sebagai komunikasi antar bebrapa layanan.
4. NATIVE adalah aplikasi yang menggunakan bahasa pemrograman yang spesifik untuk sebuah platform tertentu, contohnya penggunaan bahasa pemrograman Objective C dan Swift untuk penggunaan platform IOS. Dan untuk andorid menggunakan bahasa pemrograman Java.  
-HYBRID aplikasi ini bisa menggabungkan elemen dari aplikasi NATIVE dan Web, selain itu juga dapat menggabungkan beberapa fitur sistem operasi lainnya. Aplikasi HYBRID biasa digunakan dengan menggunakan browser untuk mengakses fitur fitur pada device mobile contohnya seperti contact, storage dll. Aplikasi HYBRID juga merupakan aplikasi yang di transformasi kan untuk kode NATIVE pada sebuah platform seperti Android ataupun IOS.
5. Fungsi Github adalah untuk membantu dalam penyimpanan repository serta github juga bisa digunakan untuk kolaborasi dalam mengerjakan project agar dapat melakukan update secara rutin, aplikasi github di lengkapi dengan beberapa elemen diantara nya : halaman profile, dapat memuat informasi berupa foto, email dll. Fungsi lain yang terdapat dalam aplikasi github yaitu :
  - Bisa untuk menyimpan repository

- Bisa juga digunakan untuk mengawasi repository tertentu
- Dapat menjadi tempat atau alat untuk melakukan kolaborasi project
- Juga dapat memuat informasi profil yang memudahkan untuk saling mengikuti atau follow programmer lain di dalam aplikasi tersebut.

6. (1)

(2)

(3)

(4)

(5)

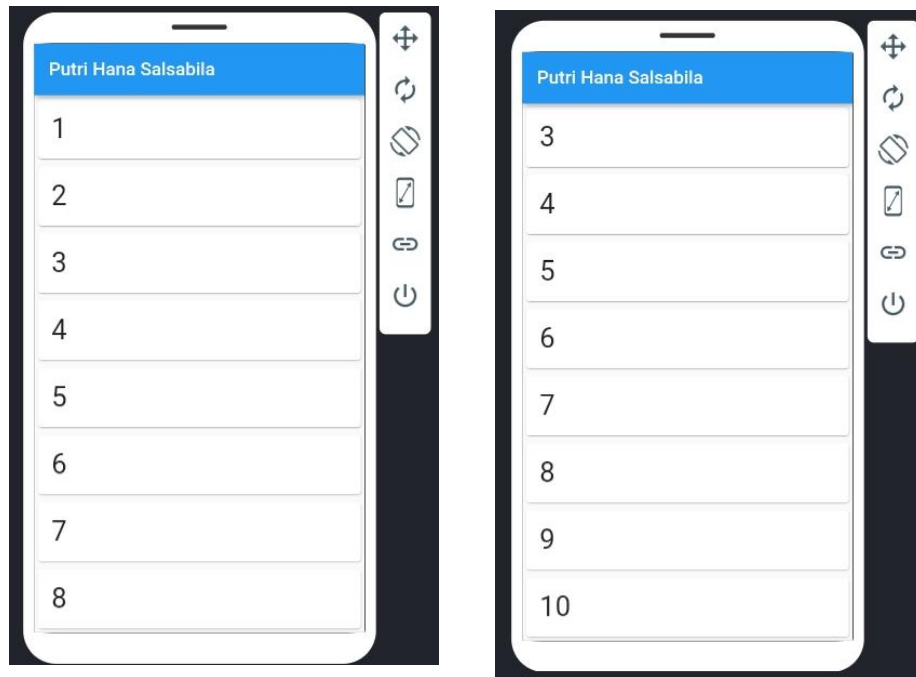
(6)

(7)

(8)

(9)

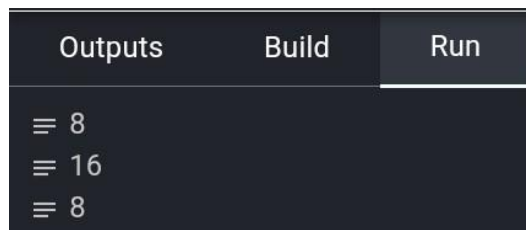
(10)



7. 8

16

8



8. `import 'views/halamanutama.dart';`  
`import 'package:flutter/material.dart';`

```
void main() {
  runApp(new MaterialApp(
    title: "My App",
    home: new MyApp(),
```

```
));  
}
```

Lalu kita buat halaman utamanya (**HalamanUtama**) pada folder views seperti berikut.

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'MY App',  
      home: HalamanUtama()  
    );  
  }  
}
```

```
import 'package:flutter/material.dart';
```

```
class HalamanUtama extends StatefulWidget {  
  @override  
  _HalamanUtamaState createState() => _HalamanUtamaState();  
}
```

```
class _HalamanUtamaState extends State {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text("Halaman Utama"),  
      ),  
      body: Text("data"),  
    );  
  }  
}
```

Selanjutnya kita siapkan API dan Modelnya.

```
[
  {
    "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"
  }
},]

```

Untuk membuat model biasanya saya pakai yang instan menggunakan <https://app.quicktype.io/>. Caranya tinggal kita paste saja json kita ke form yang sudah disediakan dan kita pilih bahasa pemrograman **dart**. Kemudian tinggal kita copy paste saja hasilnya ke dalam file model kita yaitu di sini saya beri nama userModel.dart.

```

// To parse this JSON data, do
//
//   final userModel = userModelFromJson(jsonString);

```

```

import 'dart:convert';

```

```

List userModelFromJson(String str) => List.from(json.decode(str).map((x) =>
UserModel.fromJson(x)));

```

```
String userModelToJson(List data) => json.encode(List.from(data.map((x) =>
x.toJson())));
```

```
class UserModel {
  int id;
  String name;
  String username;
  String email;
  Address address;
  String phone;
  String website;
  Company company;
```

```
  UserModel({
    this.id,
    this.name,
    this.username,
    this.email,
    this.address,
    this.phone,
    this.website,
    this.company,
  });
```

```
factory UserModel.fromJson(Map json) => UserModel(
  id: json["id"],
  name: json["name"],
  username: json["username"],
  email: json["email"],
  address: Address.fromJson(json["address"]),
  phone: json["phone"],
```



```
        website: json["website"],
        company: Company.fromJson(json["company"]),
    );
```

```
Map toJson() => {
    "id": id,
    "name": name,
    "username": username,
    "email": email,
    "address": address.toJson(),
    "phone": phone,
    "website": website,
    "company": company.toJson(),
};
}
```

```
class Address {
    String street;
    String suite;
    String city;
    String zipcode;
    Geo geo;
```

```
Address({
    this.street,
    this.suite,
    this.city,
    this.zipcode,
    this.geo,
});
```

```
factory Address.fromJson(Map json) => Address(  
  street: json["street"],  
  suite: json["suite"],  
  city: json["city"],  
  zipcode: json["zipcode"],  
  geo: Geo.fromJson(json["geo"]),  
);
```

```
Map toJson() => {  
  "street": street,  
  "suite": suite,  
  "city": city,  
  "zipcode": zipcode,  
  "geo": geo.toJson(),  
};  
}
```

```
class Geo {  
  String lat;  
  String lng;  
  
  Geo({  
    this.lat,  
    this.lng,  
  });
```

```
factory Geo.fromJson(Map json) => Geo(  
  lat: json["lat"],  
  lng: json["lng"],  
);
```

```
Map toJson() => {  
    "lat": lat,  
    "lng": lng,  
};  
}
```

```
class Company {  
    String name;  
    String catchPhrase;  
    String bs;
```

```
    Company({  
        this.name,  
        this.catchPhrase,  
        this.bs,  
    });
```

```
factory Company.fromJson(Map json) => Company(  
    name: json["name"],  
    catchPhrase: json["catchPhrase"],  
    bs: json["bs"],  
);
```

```
Map toJson() => {  
    "name": name,  
    "catchPhrase": catchPhrase,  
    "bs": bs,  
};  
}
```

Nah sekarang kita harus membuat fungsi untuk memanggil api tersebut yaitu dengan Future dengan nama file **userviewmodel.dart**. Contohnya seperti berikut.

```
import 'dart:convert';
import 'package:blog_sample/models/usermodel.dart';
import 'package:http/http.dart' as http;

class UserViewModel {

  Future<getUsers() async {
    try {
      http.Response hasil = await http.get(
        Uri.encodeFull("https://jsonplaceholder.typicode.com/users"),
        headers: {"Accept": "application/json"});
      if (hasil.statusCode == 200) {
        print("data category success");
        final data = userModelFromJson(hasil.body);
        return data;
      } else {
        print("error status " + hasil.statusCode.toString());
        return null;
      }
    } catch (e) {
      print("error catch $e");
      return null;
    }
  }
}
```

Karena kita pakai package http maka jangan lupa menambahkan dependency pada pubspec.yaml seperti berikut.

dependencies:

### **http: ^0.12.1**

Langkah terakhir tinggal kita panggil Future dari UserViewModel untuk ditampilkan ke dalam view dengan cara menambahkan kode berikut pada file **halamanutama.dart**.

```
import 'package:blog_sample/models/usermodel.dart';
import 'package:blog_sample/viewmodels/userviewmodel.dart';
import 'package:flutter/material.dart';
```

```
class HalamanUtama extends StatefulWidget {
  @override
  _HalamanUtamaState createState() => _HalamanUtamaState();
}
```

```
class _HalamanUtamaState extends State {
  List dataUser = new List();
  void getDataUser() {
    UserViewModel().getUsers().then((value) {
      setState() {
        dataUser = value;
      });
    });
  }
}
```

```
@override
void initState() {
  // TODO: implement initState
  super.initState();
  getDataUser();
}
```

```
@override
Widget build(BuildContext context) {
```

```
return Scaffold(  
  appBar: AppBar(  
    title: Text("Halaman Utama"),  
  ),  
  body: dataUser == null ? Center(child: CircularProgressIndicator(),) :  
  ListView.builder(  
    itemCount: dataUser.length,  
    itemBuilder: (context, i) {  
      return ListTile(  
        title: Text(dataUser[i].name),  
      );  
    },  
  ),  
);  
}
```