

Pengembangan Aplikasi Mobile

TI2115

By:

Sio Jurnalís Pipin, S.Kom., M.Kom.
Ricky Paian Limbong, S.Kom., M.Kom.

Course Version: 2024
Universitas Mikroskil, Copyright ©2024

COURSE OVERVIEW

COURSE OVERVIEW

Mata kuliah ini membahas tentang dasar-dasar pemrograman mobile dimulai dari perancangan user interface hingga proses back-end dan mampu membangun aplikasi mobile menggunakan flutter yang memanfaatkan bahasa pemrograman Dart.

COURSE GOALS

Capaian Pembelajaran Lulusan yang dibebankan:

- Mahasiswa mampu **menerapkan** penggunaan API

COURSE OBJECTIVES

Capaian Pembelajaran Mata Kuliah:

- Mahasiswa mampu **memahami** dan **menerapkan** dasar-dasar dart pada pengembangan aplikasi mobil.
- Mahasiswa mampu **memahami** dan **menerapkan** objek dan class dalam dart pada pengembangan aplikasi mobil.
- Mahasiswa mampu **memahami** dan **menerapkan** exception handling pada pengembangan aplikasi mobil.
- Mahasiswa mampu **mengimplementasikan** pengembangan aplikasi mobile menggunakan flutter.
- Mahasiswa mampu **menerapkan** dasar-dasar flutter.
- Mahasiswa mampu **menerapkan** navigasi dan routing dalam flutter.
- Mahasiswa mampu **menerapkan** dan **mengimplementasikan** manajemen state di flutter.
- Mahasiswa mampu **menerapkan** dan **mengimplementasikan** plugin dan package internal.
- Mahasiswa mampu **menerapkan** dan **mengimplementasikan** integrasi API dalam flutter.
- Mahasiswa mampu **menerapkan** dan **mengimplementasikan** model data, view layer dan state management.
- Mahasiswa mampu **menerapkan** dan **mengimplementasikan** sharedPreferences pada flutter.
- Mahasiswa mampu **menerapkan** dan **mengimplentasikan** Interasi firebase pada flutter.

UNIT 12

INTEGRASI API DALAM FLUTTER

UNIT OVERVIEW

Pada minggu ke-11, mahasiswa akan memanfaatkan package flutter external dengan menginstal dan menggunakannya pada project sesuai dengan kebutuhan.

UNIT OBJECTIVES

Capaian yang diharapkan yaitu:

- Mampu menerapkan dan mengimplementasikan API pada aplikasi flutter.

UNIT CONTENTS

Lesson 1:Api pada flutter..... 4 - 5

PRE LAB

Pre lab, berisi pertanyaan mendasar terkait teori materi yang sedang diajarkan

QUESTION

1. Apa yang perlu dipersiapkan untuk koneksi API?
2. Bagaimana cara melakukan komunikasi server API melalui flutter.

CONTENT LESSON

CASE STUDY / PROJECT

-

IDENTIFICATION CONCEPT OF PROBLEM / PROJECT LESSON: KONSEP DART

[!] Gunakan dokumentasi "Fetch data from the internet" dari flutter sebagai panduan memulai pemrograman flutter.

SOLUTION

Setelah mempelajari Integrasi API dalam Flutter mahasiswa dapat melakukan koneksi API ke aplikasi flutter.

INSTRUCTION

Pada materi ini, package yang akan digunakan adalah *http*.

- Silahkan ikuti langkah-langkah berikut untuk memulai instalasi dan penggunaan:
-
- Konfigurasi dan instalasi provider
- Pastikan telah install *http*. Cek "**pubspec.yaml**", apakah telah terdapat package:
- *http*: ^1.2.1

```
30 dependencies:
31   flutter:
32     sdk: flutter
33   provider:
34
35
36   # The following adds the Cupertino Icons font to your
37   # Use with the CupertinoIcons class for iOS style icons
38   cupertino_icons: ^1.0.2
39   http: ^1.2.1
```

-
-

1. Mengimpor Library:

- **dart:convert:** Digunakan untuk mengonversi data JSON ke format yang dapat digunakan oleh program Dart.
- **package:http/http.dart:** Memungkinkan Anda membuat permintaan HTTP untuk mengambil data dari API.

2. Fungsi `fetchAlbums`:

```
Future<List<Album>> fetchAlbums() async{
  final response = await
  http.get(Uri.parse('https://jsonplaceholder.typicode.com/albums'));

  if (response.statusCode == 200) {
    final data = jsonDecode(response.body) as List;
    return data.map((album) => Album.fromJson(album as Map<String,
dynamic>)).toList();
  } else {
    throw Exception('Failed to load Album');
  }
}
```

3. Class Album

```
class Album{
  final int userId;
  final int id;
  final String title;

  const Album({ required this.userId, required this.id, required
this.title});

  factory Album.fromJson(Map<String, dynamic> object){
    return Album(userId: object["userId"], id: object["id"], title:
object["title"]);
  }
}
```

4. UI untuk menampilkan hasil data

```
class AlbumApiAllData extends StatefulWidget {
  const AlbumApiAllData({super.key});

  @override
  State<AlbumApiAllData> createState() => _AlbumApiAllDataState();
}

class _AlbumApiAllDataState extends State<AlbumApiAllData> {
  late Future<List<Album>> futureAlbums;

  @override
  void initState() {
    super.initState();
    futureAlbums = fetchAlbums();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Album Test API"),
      ),
      body: FutureBuilder(
        future: futureAlbums,
        builder: (context, snapshot){
          if (snapshot.hasData) {
            final List<Album> albums = snapshot.data!;
            return ListView.builder(
              itemCount: albums.length,
              itemBuilder: (context, index){
                final album = albums[index];
                return Text(album.title);
              },
            );
          } else if (snapshot.hasError) {
            return Text('${snapshot.error}');
          }

          return const CircularProgressIndicator();
        },
      ),
    );
  }
}
```

EXERCISE

EXERCISE OBJECTIVES

TASK 1: LENGKAPI APLIKASI DI ATAS DENGAN KETENTUAN:

1. Temukan API publik menarik yang menyediakan data yang ingin Anda tampilkan. Berikut beberapa sumber untuk membantu Anda menemukan API:
 - PokeAPI
 - Randomuser
 - Atau pilih dari daftar ini : <https://github.com/public-api-lists/public-api-lists>



UNIVERSITAS
MIKROSKIL