

Modul 5

Getting Data from the Web (2)

Adding a ListView to show data, passing data through screens

Module Overview

Memprogram aplikasi untuk dapat menampilkan data dari web API pada ListView dan meneruskannya pada *screen* yang berbeda.

Module Objectives

Setelah mempelajari dan mempraktikkan modul ini, mahasiswa diharapkan dapat:

- Mengetahui bagaimana cara menampilkan data dari web API pada ListView
- Meneruskan data yang dipilih pada *screen* yang berbeda

Pada pertemuan sebelumnya Anda telah belajar bagaimana cara memperoleh data dari sebuah website dengan menggunakan API (*Application Programming Interface*).

Biasanya setiap developer akan diberikan sebuah kunci yang unik sebagai verifikasi akses bagi Anda untuk dapat menggunakan API yang disediakan oleh website tersebut.



Gambar 1. Hasil Pembacaan API themoviedb.org dalam Format JSON

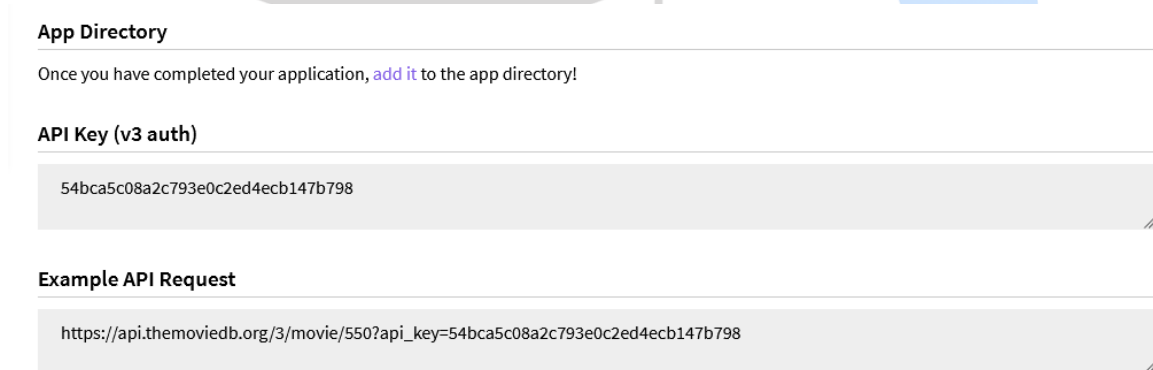
Data yang diperoleh dari API pada umumnya akan dikirimkan dalam format *JavaScript Object Notation* (JSON). JSON adalah format data yang dibuat agar mudah dibaca baik oleh manusia maupun komputer seperti yang sudah Anda coba menggunakan di minggu lalu. JSON merupakan cara paling populer bagi aplikasi untuk bertukar data dan paling sering ditemui saat bekerja dengan API.

Akan tetapi, anda mungkin menyadari bahwa data dalam format JSON tersebut cukup sulit untuk dipahami terlebih lagi jika jumlah data yang diperoleh dari API tersebut sangat banyak. Untuk itu kali ini Anda akan belajar bagaimana cara mengolah dan menampilkan data yang Anda peroleh dari API tersebut menjadi sebuah informasi yang mudah untuk dipahami dengan memanfaatkan widget-widget yang ada pada Flutter.

Perlu Anda ingat bahwa mungkin tidak semua data yang Anda peroleh dari API harus Anda tampilkan. Beberapa bagian data mungkin cukup Anda tempatkan di belakang layar dan dipanggil ketika Anda ingin meneruskan data tersebut pada *screen* yang berbeda. Menggunakan API dari themoviedb.org yang minggu lalu sudah Anda coba gunakan, kali ini Anda akan belajar bagaimana menampilkan data dari API tersebut pada sebuah ListView.

Movie Database API Web Service

Pada contoh ini, kita masih akan mengambil data dari salah satu penyedia data JSON untuk film dan serial tv dengan beberapa Bahasa yang tersedia, yaitu <https://www.themoviedb.org/>. Untuk melakukan koneksi ke dalam aplikasi, anda diwajibkan memiliki akun di dalam <https://www.themoviedb.org/> tanpa diperlukan biaya. Pastikan Anda masih menyimpan API Key yang minggu lalu sudah Anda dapatkan seperti pada gambar di bawah ini.



Gambar 2. Halaman API Key

Anda dapat melanjutkan program ini dari pertemuan sebelumnya untuk mendapatkan data JSON. Dan untuk memudahkan Anda dalam memastikan semua data dapat diwakilkan dengan baik maka Anda dapat membuahkan sebuah class model dengan membuat file baru dengan nama Movies.dart seperti berikut ini.

```
class Movie {  
  final int id;  
  final String title;  
  final double voteAverage;  
  final String releaseDate;  
  final String overview;  
  final String posterPath;  
}
```

```
Movie(this.id, this.title, this.voteAverage, this.releaseDate,
this.overview, this.posterPath);

factory Movie.fromJson(Map<String, dynamic> parsedJson) {
  final id = parsedJson['id'] as int;
  final title = parsedJson['title'] as String;
  final voteAverage = parsedJson['vote_average'] * 1.0 as double;
  final releaseDate = parsedJson['release_date'] as String;
  final overview = parsedJson['overview'] as String;
  final posterPath = parsedJson['poster_path'] as String;
  return Movie(id, title, voteAverage, releaseDate, overview, posterPath);
}
```

Pada model class Movie ini kita akan memerlukan sebuah metode fromJson untuk mengubah data dari JSON menjadi sebuah List untuk nantinya data tersebut dapat dengan mudah kita tampilkan pada bagian front-end.

1. Sama seperti minggu sebelumnya, pada class HttpHelper, masukkan API Key yang sudah Anda peroleh sebelumnya seperti berikut.

```
6. class HttpHelper {
7.   final String _urlKey = "?api_key=fdbce446784119fbe5f15f52ab3d29d7";
8.   final String _urlBase = "https://api.themoviedb.org";
```

2. Untuk dapat menggunakan paket http, pastikan Anda melakukan import paket http seperti berikut ini.

```
import 'package:http/http.dart' as http;
```

3. Masih pada class HttpHelper, tambahkan sebuah metode untuk melakukan permintaan data terhadap API dengan mengakses url API seperti di bawah ini.

```
10. Future<List?> getMovies() async {
11.   var url = Uri.parse(_urlBase + '/3/movie/now_playing' + _urlKey);
12.   http.Response result = await http.get(url);
13.   if (result.statusCode == HttpStatus.ok) {
14.     final jsonResponse = json.decode(result.body);
15.     final moviesMap = jsonResponse['results'];
16.     List movies = moviesMap.map((i) => Movie.fromJson(i)).toList();
17.     return movies;
```

```
18.   } else {  
19.     return null;  
20.   }  
21. }
```

Perhatikan pada bagian url, di sini kita akan melakukan permintaan data (request) terhadap film yang sedang tayang (now_playing). Jika pengaksesan data yang dilakukan berhasil maka data yang diterima dalam format JSON tersebut akan diubah menjadi format String karena adanya metode `json.decode` terhadap hasil response.

Untuk dapat menggunakan enum `HttpStatus.ok` (status code 200) dan melakukan konversi JSON maka Anda perlu menambahkan library-library yang dibutuhkan dengan mengimpor:

```
import 'dart:io';  
import 'dart:convert';
```

4. Sekarang kita akan merancang sebuah tampilan menggunakan `ListView` untuk menampilkan setiap item yang diterima melalui API sebelumnya. Untuk itu buatlah sebuah file `home.dart` baru dan rancang sebuah tampilan sebagai berikut ini.

```
1. import 'package:flutter/material.dart';  
2. import 'package:m05/HttpHelper.dart';  
3. import 'package:m05/detail.dart';  
4.  
5. class HomeScreen extends StatefulWidget {  
6.   const HomeScreen({super.key});  
7.  
8.   @override  
9.   State<HomeScreen> createState() => _HomeScreenState();  
10.}  
11.  
12. class _HomeScreenState extends State<HomeScreen> {  
13.   HttpHelper? helper;  
14.   List? movies;  
15.   final String iconBase = 'https://image.tmdb.org/t/p/w92/';  
16.   final String defaultImage =
```

```
17.     'https://images.freeimages.com/images/large-previews/5eb/movie-
      clapboard-1184339.jpg';
18.
19. @override
20. Widget build(BuildContext context) {
21.     NetworkImage image;
22.     return Scaffold(
23.         appBar: AppBar(title: Text('Now Playing')),
24.         body: ListView.builder(
25.             itemCount: (movies?.length == null) ? 0 : movies?.length,
26.             itemBuilder: (BuildContext context, int position) {
27.                 if (movies![position].posterPath != null) {
28.                     image = NetworkImage(iconBase +
      movies![position].posterPath);
29.                 } else {
30.                     image = NetworkImage(defaultImage);
31.                 }
32.                 return Card(
33.                     color: Colors.white,
34.                     elevation: 2.0,
35.                     child: ListTile(
36.                         onTap: () {
37.                             MaterialPageRoute route = MaterialPageRoute(
38.                                 builder: (_) =>
      DetailScreen(movies![position]));
39.                             Navigator.push(context, route);
40.                         },
41.                         leading: CircleAvatar(
42.                             backgroundImage: image,
43.                         ),
44.                         title: Text(movies![position].title),
45.                         subtitle: Text('Released: ' +
46.                             movies![position].releaseDate +
47.                             ' - Vote: ' +
48.                             movies![position].voteAverage.toString()),
49.                     ));
50.             }));
51. }
52. }
53.
```

5. Tambahkan sebuah metode Future secara asinkron untuk melakukan permintaan (request) data movies kepada API. Metode ini akan memanggil fungsi getMovies yang ada pada class HttpHelper.

```
55. Future initialize() async {  
56.   movies = await helper?.getMovies();  
57.   setState(() {  
58.     movies = movies;  
59.   });  
60. }
```

Data yang diperoleh dari API tersebut akan ditampilkan pada aplikasi dengan menggunakan metode setState.

6. Agar data movies tersebut dapat ditampilkan ketika aplikasi dijalankan maka tambahkan metode initState dan panggil metode Future initialize sebelumnya seperti berikut ini.

```
21. @override  
22. void initState() {  
23.   helper = HttpHelper();  
24.   initialize();  
25.   super.initState();  
26. }
```

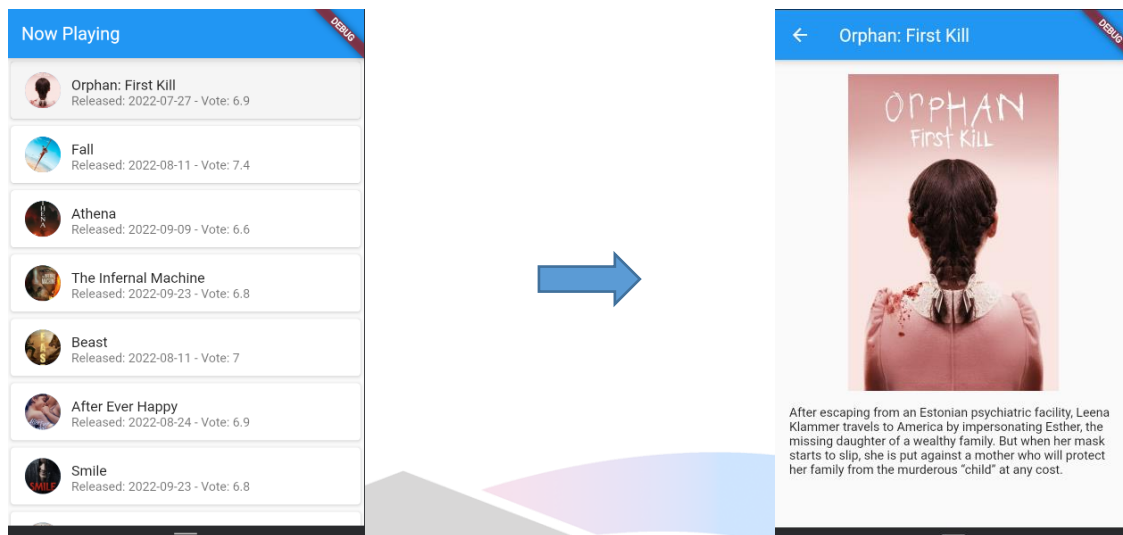
Dengan demikian ketika aplikasi dijalankan maka aplikasi akan melakukan permintaan data (request) terlebih dahulu lalu dilanjutkan dengan meload front-end dari aplikasi

7. Untuk dapat menampilkan detail dari item movies yang dipilih maka buatlah sebuah file dart baru bernama detail.dart, lalu rancang tampilannya sebagai berikut ini.

```
1. import 'package:flutter/material.dart';  
2. import 'package:m05/Movies.dart';  
3.  
4. class DetailScreen extends StatelessWidget {  
5.   final Movie movie;  
6.   final String imagePath = 'https://image.tmbd.org/t/p/w500/';  
7.   const DetailScreen(this.movie, {Key? key}) : super(key: key);
```

```
8.
9.  @override
10.  Widget build(BuildContext context) {
11.    double height = MediaQuery.of(context).size.height;
12.    String path;
13.    if (movie.posterPath != null) {
14.      path = imagePath + movie.posterPath;
15.    } else {
16.      path = 'https://images.freeimages.com/images/large-
      previews/5eb/movie-clapboard-1184339.jpg';
17.    }
18.    return Scaffold(
19.      appBar: AppBar(title: Text(movie.title),),
20.      body: SingleChildScrollView(
21.        child: Center(
22.          child: Column(
23.            children: <Widget>[
24.              Container(
25.                padding: EdgeInsets.all(16),
26.                height: height / 1.5,
27.                child: Image.network(path)),
28.              Container(
29.                padding: EdgeInsets.only(left: 16, right: 16),
30.                child: Text(movie.overview)),
31.            ]));
32.  }
33.}
34.
```

File ini akan menerima data movies yang diteruskan dari file home.dart sebelumnya sehingga jika project Anda dijalankan akan menghasilkan tampilan seperti di bawah ini.



Gambar 3. Hasil ListView dan Passing Data

Latihan

1. Kembangkanlah contoh pada pembacaan data, sehingga dapat menampilkan data film, berdasarkan : 'Latest', 'Now Playing', 'Popular', 'Top Rated', 'Upcoming', dengan menggunakan pilihan seperti Chips, Dropdownmenu, atau Radio Button.
2. Tambahkan sebuah fitur pencarian sehingga aplikasi dapat menampilkan film sesuai dengan kata kunci yang diinput oleh pengguna.