

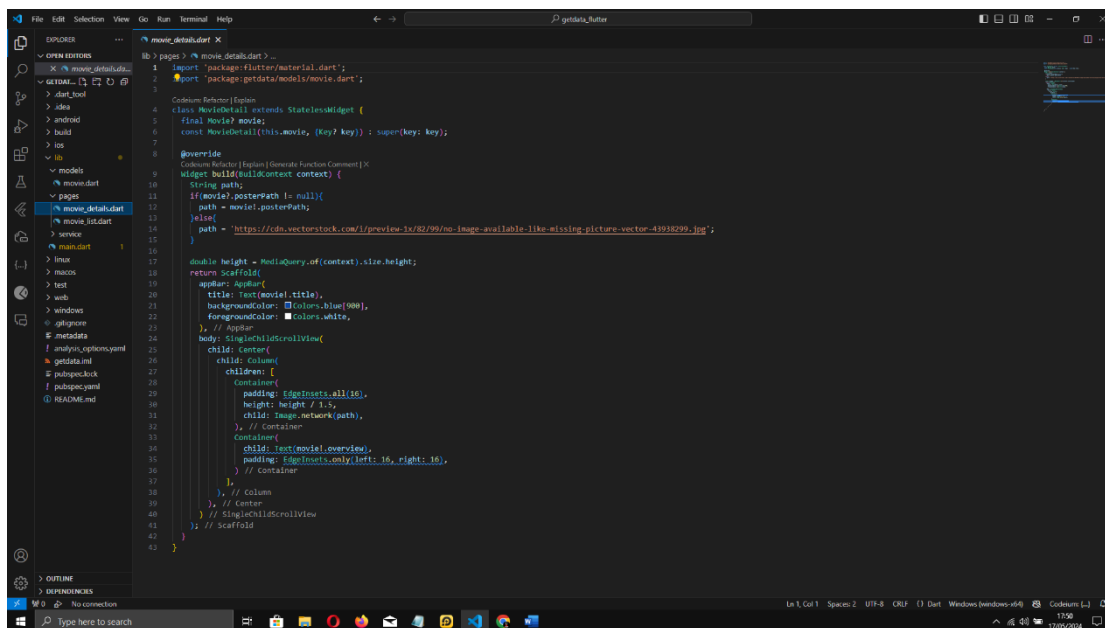
Get Data Flutter

Nama : Moch Arif Lestiawan

Kelas : XI RPL 4 (18)

Link Github : <https://github.com/MochArif99/Get-data-flutter>

Berikut code nya :



```
1 import 'package:flutter/material.dart';
2 import 'package:getdata/models/movie.dart';
3
4 class MovieDetail extends StatelessWidget {
5   final Movie? movie;
6   const MovieDetail(this.movie, {Key? key}) : super(key: key);
7
8   @override
9   Widget build(BuildContext context) {
10     String path;
11     if(movie?.posterPath != null){
12       path = movie!.posterPath;
13     } else {
14       path = 'https://cdn.vectorstock.com/1/preview-1x/02/99/no-image-available-like-missing-picture-vector-41934299.jpg';
15     }
16
17     double height = MediaQuery.of(context).size.height;
18     return Scaffold(
19       appBar: AppBar(
20         title: Text(movie!.title),
21         backgroundColor: Colors.blue(900),
22         foregroundColor: Colors.white,
23       ), // AppBar
24       body: SingleChildScrollView(
25         child: Center(
26           child: Column(
27             children: [
28               Container(
29                 padding: EdgeInsets.all(16),
30                 height: height / 1.5,
31                 child: Image.network(path),
32               ), // Container
33               Container(
34                 child: Text(movie!.overview),
35                 padding: EdgeInsets.only(left: 16, right: 16),
36               ), // Container
37             ], // Column
38           ), // Center
39         ), // SingleChildScrollView
40       ), // Scaffold
41     );
42   }
43 }
```

- Statles widget untuk menampilkan detail film. Menerima parameter movie yang merupakan objek dari kelas Movie
- Variabel path diinisialisasi dengan alamat URL poster film. Jika poster film tersedia (movie?.posterPath != null), variabel path diisi dengan nilai dari movie!.posterPath. Jika tidak tersedia, variabel path diisi dengan URL gambar default yang menampilkan gambar pengganti.
- Kemudian, variabel height diinisialisasi dengan tinggi layar perangkat menggunakan MediaQuery.of(context).size.height.
- Di dalam Scaffold, terdapat AppBar yang menampilkan judul film menggunakan movie!.title.
- Pada bagian body, terdapat SingleChildScrollView yang memungkinkan konten untuk digulir jika terlalu besar.
- Di dalam SingleChildScrollView, terdapat Container yang menampilkan poster film menggunakan Image.network(path). Tinggi kontainer diatur menjadi setengah dari tinggi layar perangkat.

```

1 import 'package:flutter/material.dart';
2 import 'package:provider/provider.dart';
3 import 'package:movie_data/service/http_service.dart';
4
5 class MovieList extends StatefulWidget {
6   const MovieList({super.key});
7
8   @override
9   State<MovieList> createState() => _MovieListState();
10
11 }
12
13 class _MovieListState extends State<MovieList> {
14   int? movieCount = 0;
15   List? movies;
16   HttpService? service;
17
18   @override
19   void initState() {
20     service = HttpService();
21     super.initState();
22   }
23
24   @override
25   Widget build(BuildContext context) {
26     service!.getPopularMovies().then((value) => {
27       setState(() {
28         // result = value.toString();
29       });
30     });
31
32     return Scaffold(
33       appBar: AppBar(
34         title: Text('Popular movie'),
35         backgroundColor: Colors.deepPurple,
36         foregroundColor: Colors.white,
37       ),
38       body: ListView.builder(
39         itemCount: movieCount,
40         itemBuilder: (context, list position) {
41           return Card(
42             color: Colors.white,
43             elevation: 1.0,
44             child: ListTile(
45               leading: Image(
46                 image: NetworkImage(movies![position].posterPath),
47               ), // Image
48               title: Text(movies![position].title),
49               subtitle: Text(
50                 'Rating: ..... * movies![position].voteAverage.toString()', // Text
51               ),
52             ),
53           );
54         },
55       ), // ListTile
56     ); // Card
57   } // ListView.builder // Scaffold
58 }

```

```

12 class _MovieListState extends State<MovieList> {
13   @override
14   void initState() {
15     service!.getPopularMovies().then((value) => {
16       setState(() {
17         // result = value.toString();
18       });
19     });
20
21     return Scaffold(
22       appBar: AppBar(
23         title: Text('Popular movie'),
24         backgroundColor: Colors.deepPurple,
25         foregroundColor: Colors.white,
26       ),
27       body: ListView.builder(
28         itemCount: movieCount,
29         itemBuilder: (context, list position) {
30           return Card(
31             color: Colors.white,
32             elevation: 1.0,
33             child: ListTile(
34               leading: Image(
35                 image: NetworkImage(movies![position].posterPath),
36               ), // Image
37               title: Text(movies![position].title),
38               subtitle: Text(
39                 'Rating: ..... * movies![position].voteAverage.toString()', // Text
40               ),
41             ),
42           );
43         },
44       ), // ListTile
45     ); // Card
46   } // ListView.builder // Scaffold
47 }

```

- Kelas Movie list widget yang bertanggung jawab untuk menampilkan daftar film populer. Kelas ini memiliki satu metode, yaitu initialize(), yang digunakan untuk menginisialisasi daftar film dengan film-film populer yang diambil dari layanan HTTP.
- Metode initState() dijalankan setelah widget dibuat, di mana layanan HTTP diinisialisasi dan initialize() dipanggil untuk mengambil daftar film populer.
- Di dalam metode build, widget ListView.builder digunakan untuk menampilkan daftar film populer.
- Ketika mengklik salah satu item dalam daftar (onTap), mereka akan dialihkan ke halaman detail film menggunakan Navigator.push untuk menavigasikan ke halaman MovieDetail dan menampilkan detail film yang dipilih.

