

**PEMROGRAMAN MOBILE**

**“Pertemuan 15 REST API”**



Di susun oleh :

Andika Ainur Wibowo

NIM 2141720238

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

Source code main.dart:



```
1 @override
2 Widget build(BuildContext context) {
3   return Scaffold(
4     appBar: AppBar(
5       title: Text(widget.title),
6     ),
7     body: ListView.separated(
8       itemBuilder: (context, index) {
9         return Text(
10          "${listBlog[index].id} | ${listBlog[index].name} | ${listBlog[index].year} | ${listBlog[index].color} | ${listBlog[index].pantone_value}";
11        ),
12        separatorBuilder: (context, index) {
13          return const Divider();
14        },
15        itemCount: listBlog.length,
16      );
17 }
```

Penjelasan code :

1. Widget build membuat tampilan menggunakan Scaffold, yang merupakan struktur dasar aplikasi Flutter.
2. Terdapat AppBar sebagai bagian atas layar dengan judul sesuai nilai yang diberikan oleh widget widget.title.
3. Bagian body menggunakan ListView.separated untuk menampilkan daftar item dengan pemisah (divider) antara setiap item.
4. Dalam metode itemBuilder, setiap elemen dari listBlog direpresentasikan sebagai widget Text yang menampilkan informasi dari objek Blog.
5. Metode separatorBuilder membuat pemisah antara setiap item menggunakan widget Divider.
6. Jumlah item dalam ListView ditentukan oleh itemCount yang sesuai dengan panjang listBlog.

Source code model.dart:



```
1 class Blog {
2   late String color;
3   late int id;
4   late int year;
5   late String pantone_value;
6   late String name;
7
8   Blog(this.id, this.name, this.color, this.year, this.pantone_value);
9
10  Blog.fromJson(Map<String, dynamic> json) {
11    id = json['id'];
12    color = json['color'];
13    year = json['year'];
14    pantone_value = json['pantone_value'];
15    name = json['name'];
16  }
17 }
```

Penjelasan code :

1. Kelas Blog memiliki beberapa properti yang merepresentasikan atribut dari suatu objek blog, seperti id, name, color, year, dan pantone\_value.
2. Konstruktor Blog digunakan untuk membuat objek Blog baru dengan memberikan nilai awal pada propertinya.
3. Metode fromJson adalah konvensi umum untuk membuat objek dari representasi JSON. Metode ini mengambil objek Map<String, dynamic> sebagai parameter dan menginisialisasi propertinya berdasarkan nilai yang diambil dari map tersebut.

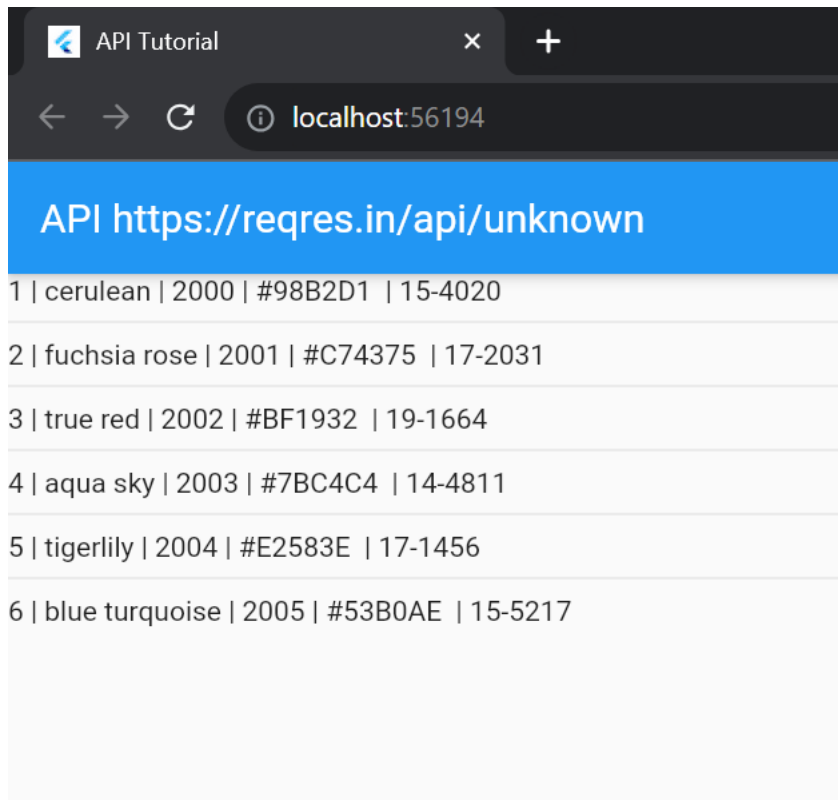
Source code Repository.dart:

```
1 import 'package:flutter/foundation.dart';
2 import 'dart:io';
3 import 'dart:convert';
4 import 'package:http/http.dart' as http;
5 import 'package:week_15/model/model.dart';
6
7
8 class Repository{
9   String final_baseUrl = "https://reqres.in/api/unknown";
10
11   Future getData() async{
12     try {
13       http.Response response = await http.get(Uri.parse(final_baseUrl));
14
15       // print(response.body);
16       //print("masuk api")
17
18       if(response.statusCode == HttpStatus.ok){
19         if(kDebugMode){}
20         //print(response.body);
21         final jsonResponse = json.decode(response.body);
22         final datablog= jsonResponse['data'];
23         List blog = datablog.map((i)>=> Blog.fromJson(i)).toList();
24         return blog;
25       }
26     } catch(e){
27       print(e.toString());
28     }
29   }
30 }
```

Penjelasan code :

1. Repository adalah kelas untuk mengelola permintaan data dari API.
2. Metode getData menggunakan pustaka http untuk mengambil data dari URL yang ditentukan.
3. Data JSON diuraikan dan diubah menjadi objek Blog.
4. Kesalahan dicetak ke konsol jika terjadi masalah.

Output :



|   |                |      |         |         |
|---|----------------|------|---------|---------|
| 1 | cerulean       | 2000 | #98B2D1 | 15-4020 |
| 2 | fuchsia rose   | 2001 | #C74375 | 17-2031 |
| 3 | true red       | 2002 | #BF1932 | 19-1664 |
| 4 | aqua sky       | 2003 | #7BC4C4 | 14-4811 |
| 5 | tigerlily      | 2004 | #E2583E | 17-1456 |
| 6 | blue turquoise | 2005 | #53B0AE | 15-5217 |