

**PEMROGRAMAN MOBILE**

**“Cupertino & REST API”**



Di susun oleh :

Andika Ainur Wibowo

NIM 2141720238

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

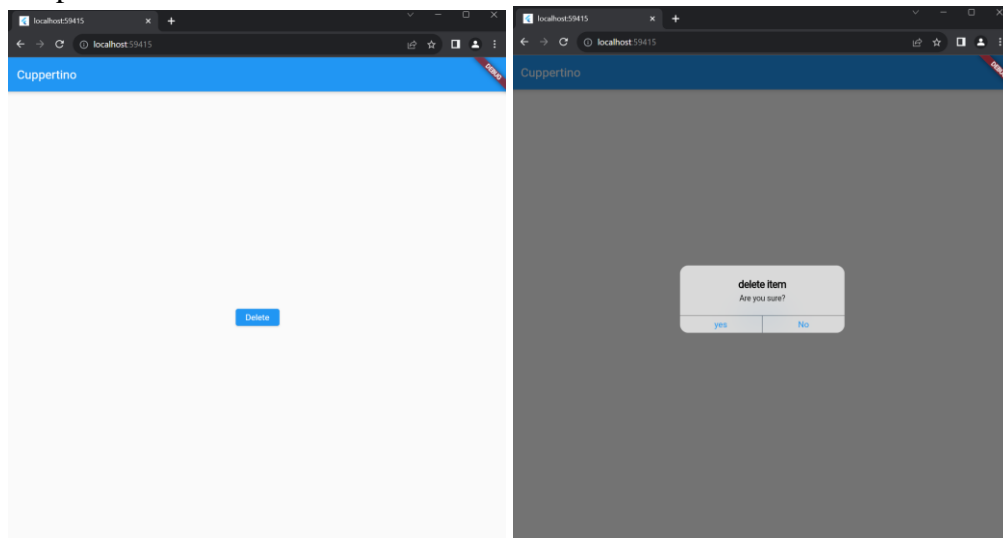
- Install Cupertino pada pubspec.yaml

```
1 dependencies:
2   flutter:
3     sdk: flutter
4   http: ^0.13.3
5
6   # The following adds the Cupertino Icons font to your application.
7   # Use with the CupertinoIcons class for iOS style icons.
8   cupertino_icons: ^1.0.2
9
```

- Kode ini memunculkan tombol delete yang menampilkan dialog style Cupertino ketika di tekan

```
1 import 'package:flutter/material.dart';
2 import 'package:flutter/cupertino.dart';
3
4 void main() {
5   runApp(MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      home: Homepage(),
13    );
14  }
15 }
16
17 class Homepage extends StatelessWidget {
18   @override
19   Widget build(BuildContext context) {
20     return Scaffold(
21       appBar: AppBar(
22         title: Text("Cupertino"),
23       ),
24       body: Center(
25         child: ElevatedButton(
26           onPressed: () {
27             showDialog(
28               context: context,
29               builder: (context) {
30                 return CupertinoAlertDialog(
31                   title: Text("delete item"),
32                   content: Text("Are you sure?"),
33                   actions: [
34                     TextButton(
35                       onPressed: () {},
36                       child: Text("yes"),
37                     ),
38                     TextButton(
39                       onPressed: () {},
40                       child: Text("No"),
41                     ),
42                   ],
43                 );
44               },
45             );
46           },
47           child: Text("Delete"),
48         ),
49       ),
50     );
51  }
52 }
```

- Output



- Kode ini untuk mengambil data dari API menggunakan HTTP GET request dan mengelola responsnya.

```
1 import '../model/mahasiswa.dart';
2 import 'package:http/http.dart' as http;
3
4 class MahasiswaService {
5   static final String _baseUrl = "https://regres.in/api/unknown/2";
6
7   var http;
8
9   Future getMahasiswa() async{
10     Uri urlApi =Uri.parse(_baseUrl);
11
12     final response = await http.get(urlApi);
13     if (response.statusCode == 200){
14       return mahasiswaFromJson(response.body.toString());
15     }else{
16       throw Exception('Failed to load data mahasiswa');
17     }
18   }
19 }
```

- Kode ini untuk implementasi konversi antara objek Dart dan representasi JSON-nya menggunakan metode serialization dan deserialization.

```
1 import 'dart:convert';
2
3 Mahasiswa mahasiswaFromJson(String str) => Mahasiswa.fromJson(json.decode(str));
4
5 String mahasiswaToJson(Mahasiswa data) => json.encode(data.toJson());
6
7 class Mahasiswa {
8   Data data;
9   Support support;
10
11   Mahasiswa({
12     required this.data,
13     required this.support,
14   });
15
16   factory Mahasiswa.fromJson(Map<String, dynamic> json) => Mahasiswa(
17     data: Data.fromJson(json["data"]),
18     support: Support.fromJson(json["support"]),
19   );
20   Map<String, dynamic> toJson() => {
21     "data": data.toJson(),
22     "support": support.toJson(),
23   };
24 }
25
26 class Data {
27   int id;
28   String name;
29   int year;
30   String color;
31   String pantoneValue;
32
33   Data({
34     required this.id,
35     required this.name,
36     required this.year,
37     required this.color,
38     required this.pantoneValue,
39   });
40
41   factory Data.fromJson(Map<String, dynamic> json) => Data(
42     id: json["id"],
43     name: json["name"],
44     year: json["year"],
45     color: json["color"],
46     pantoneValue: json["pantone_value"],
47   );
48
49   Map<String, dynamic> toJson() => {
50     "id": id,
51     "name": name,
52     "year": year,
53     "color": color,
54     "pantone_value": pantoneValue,
55   };
56 }
57
58 class Support {
59   String url;
60   String text;
61
62   Support({
63     required this.url,
64     required this.text,
65   });
66
67   factory Support.fromJson(Map<String, dynamic> json) => Support(
68     url: json["url"],
69     text: json["text"],
70   );
71
72   Map<String, dynamic> toJson() => {
73     "url": url,
74     "text": text,
75   };
76 }
```

- Main.dart

```
1 // import 'dart:ffi';
2
3 import 'package:flutter/material.dart';
4 import './model/mahasiswa.dart';
5 import './servis/mahasiswaServis.dart';
6
7 void main() {
8   runApp(MyApp());
9 }
10
11 class MyApp extends StatelessWidget {
12   @override
13   Widget build(BuildContext context) {
14     return MaterialApp(
15       title: "Rest API",
16       home: MyHomePage(),
17     );
18   }
19 }
20
21 class MyHomePage extends StatefulWidget {
22   @override
23   _MyHomePageState createState() => _MyHomePageState();
24 }
25
26 class _MyHomePageState extends State<MyHomePage> {
27   late Future data;
28   List<Mahasiswa> data2 = [];
29
30   @override
31   Widget build(BuildContext context) {
32     return Scaffold(
33       appBar: AppBar(
34         title: const Text("Rest API"),
35       ),
36       body: data2.length == 0
37         ? const Center(
38           child: CircularProgressIndicator(
39             backgroundColor: Colors.red,
40           ),
41         )
42         : ListView.builder(
43           itemCount: data2.length,
44           itemBuilder: (context, index) {
45             return ListTile(
46               title: Text(data2[index].toString()),
47             );
48           });
49   }
50 }
```

- Output

