

**TUGAS PENDAHULUAN / TUGAS UNGUIDED
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XIV
DATA STORAGE API**



Disusun Oleh :
Farhan Kurniawan / 2311104073
SE-07-02

Asisten Praktikum :
Zulfa Mustafa Akhyar Iswahyudi
Yoga Eka Pratama

Dosen Pengampu :
Yudha Islami Sulistyaa, S.Kom., M.Cs.

**PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

TUGAS PENDAHULUAN / TUGAS UNGUIDED

A. SOAL

Tugas Mandiri (Unguided)

Modifikasi tampilan Guided dari praktikum di atas:

a. **Gunakan State Management dengan GetX:**

- Atur data menggunakan *state management* GetX agar lebih mudah dikelola.
- Implementasi GetX meliputi pembuatan controller untuk mengelola data dan penggunaan widget Obx untuk menampilkan data secara otomatis setiap kali ada perubahan.

b. **Tambahkan Snackbar untuk Memberikan Respon Berhasil:**

- Tampilkan snackbars setelah setiap operasi berhasil, seperti menambah atau memperbarui data.
- Gunakan Get.snackbar agar pesan sukses muncul di layar dan mudah dipahami oleh pengguna.

Note: Jangan lupa sertakan source code, screenshot output, dan deskripsi program. Kreatifitas menjadi nilai tambah.

B. JAWABAN

Source Code pada Main Dart

```
//controllers/post_controller.dart

import 'package:get/get.dart';

import '../services/api_service.dart';

class PostController extends GetxController {
    final ApiService apiService = ApiService();

    var posts = <dynamic>[].obs;
    var isLoading = false.obs;

    void getPosts() async {
        try {
            isLoading(true);
            posts.value = await apiService.fetchPosts();
            Get.snackbar("Sukses", "Data berhasil diambil");
        } catch (e) {
            print(e);
        }
    }
}
```

```
    } catch (e) {
        Get.snackbar("Error", e.toString());
    } finally {
        isLoading(false);
    }
}

void addPost() async {
    await apiService.createPost();
    Get.snackbar("Sukses", "Data berhasil ditambahkan");
}

void updatePost() async {
    await apiService.updatePost();
    Get.snackbar("Sukses", "Data berhasil diperbarui");
}

void deletePost() async {
    await apiService.deletePost();
    Get.snackbar("Sukses", "Data berhasil dihapus");
}

//screens/home_screen.dart
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import '../controllers/post_controller.dart';

class HomeScreen extends StatelessWidget {
    final PostController controller = Get.put(PostController());

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(title: Text("Unguided REST API")),

```

```
body: Column(  
    children: [  
        Wrap(  
            spacing: 8,  
            children: [  
                ElevatedButton(  
                    onPressed: controller.getPosts,  
                    child: Text("GET"),  
                ),  
                ElevatedButton(  
                    onPressed: controller.addPost,  
                    child: Text("POST"),  
                ),  
                ElevatedButton(  
                    onPressed: controller.updatePost,  
                    child: Text("PUT"),  
                ),  
                ElevatedButton(  
                    onPressed: controller.deletePost,  
                    child: Text("DELETE"),  
                ),  
            ],  
        ),  
        Expanded(  
            child: Obx(() {  
                if (controller.isLoading.value) {  
                    return Center(child: CircularProgressIndicator());  
                }  
  
                return ListView.builder(  
                    itemCount: controller.posts.length,  
                    itemBuilder: (context, index) {  
                        return Card(  
                            child: ListTile(  
                                title: Text(controller.posts[index]['title'] ?? ''),  
                            ),  
                        );  
                    },  
                );  
            }),  
        ),  
    ],  
);
```



```
'body': 'Ini data POST dari Flutter',
'userId': 1,
}),
);

if (response.statusCode != 201) {
    throw Exception('Gagal menambah data');
}
}

Future<void> updatePost() async {
    final response = await http.put(
        Uri.parse('$baseUrl/posts/1'),
        headers: {'Content-Type': 'application/json'},
        body: json.encode({
            'title': 'Post Diupdate',
            'body': 'Ini hasil update',
            'userId': 1,
        }),
    );

    if (response.statusCode != 200) {
        throw Exception('Gagal update data');
    }
}

Future<void> deletePost() async {
    final response = await http.delete(Uri.parse('$baseUrl/posts/1'));

    if (response.statusCode != 200) {
        throw Exception('Gagal hapus data');
    }
}
```

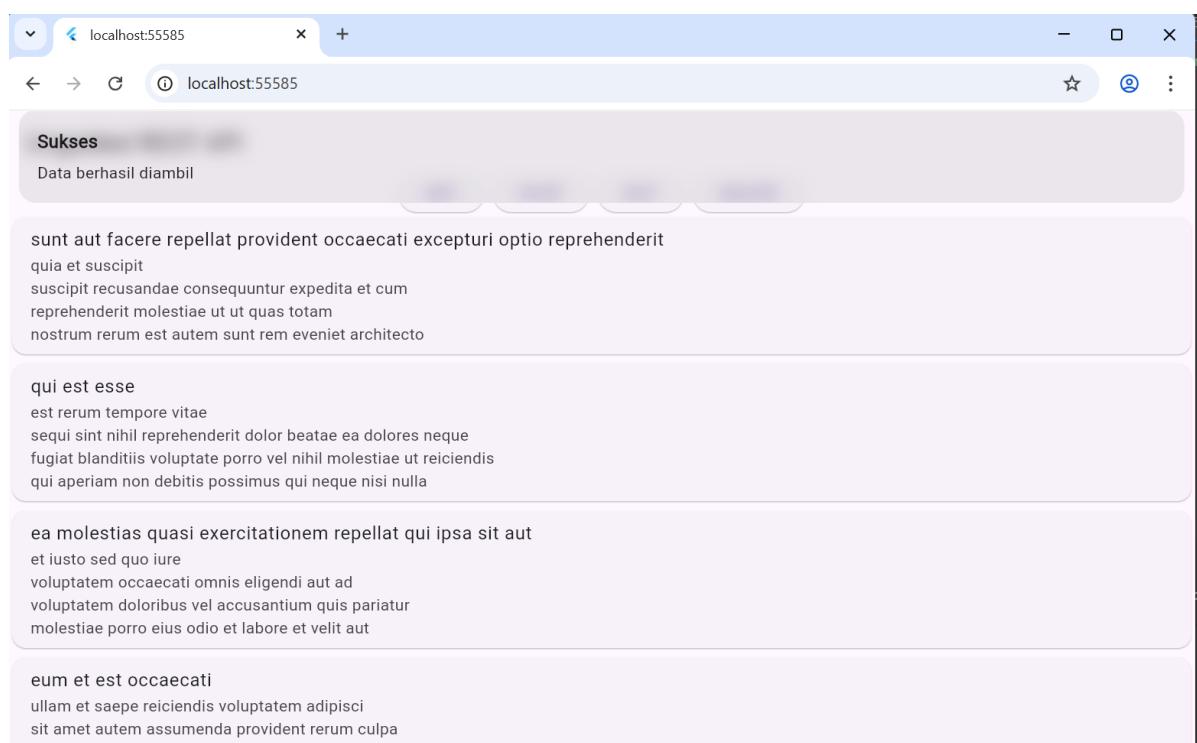
```
//main.dart

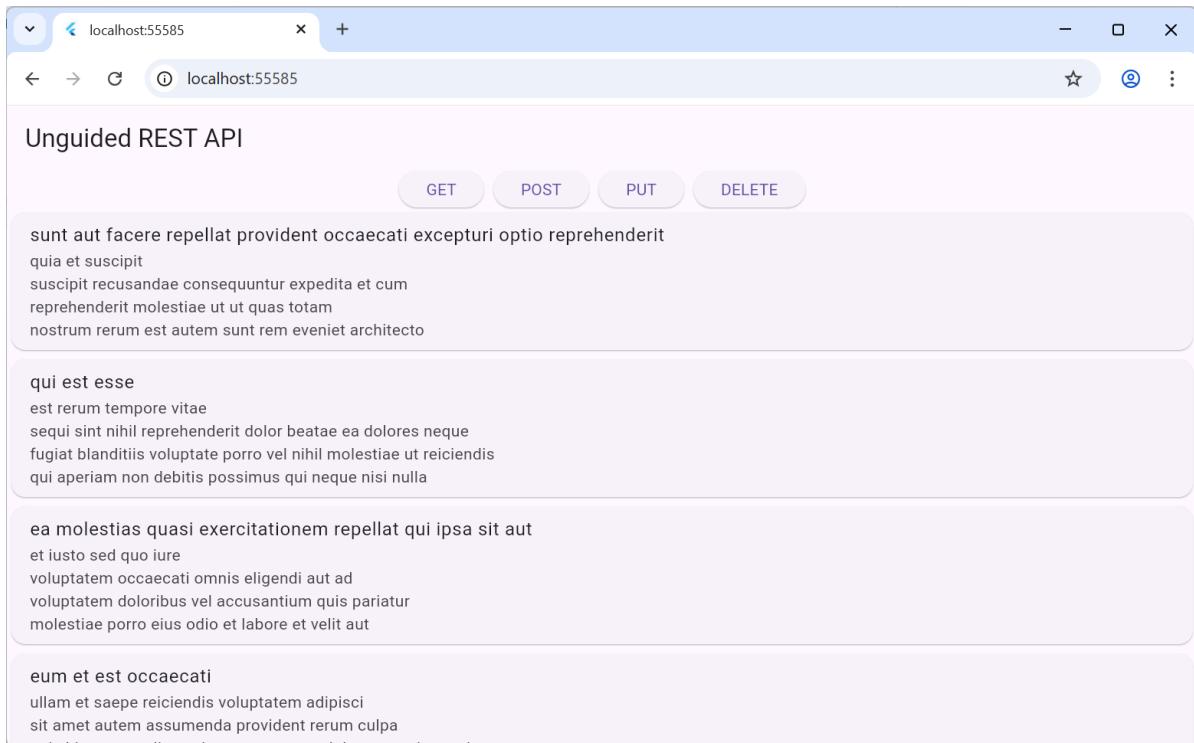
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'screens/home_screen.dart';

void main() {
    runApp(MyApp());
}

class MyApp extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return GetMaterialApp(
            debugShowCheckedModeBanner: false,
            home: HomeScreen(),
        );
    }
}
```

Screenshot Output:





Deskripsi Program

Program ini merupakan aplikasi mobile berbasis **Flutter** yang dirancang untuk memahami dan mengimplementasikan konsep **Data Storage menggunakan REST API**. Aplikasi ini terhubung dengan layanan API publik jsonplaceholder.typicode.com yang berfungsi sebagai server penyedia data. Melalui aplikasi ini, pengguna dapat melakukan operasi pengelolaan data berupa **mengambil data (GET)**, **menambahkan data (POST)**, **memperbarui data (PUT)**, dan **menghapus data (DELETE)**.

Dalam pengembangannya, aplikasi ini menerapkan **State Management GetX** untuk mengelola perubahan data secara reaktif dan efisien. Setiap perubahan data yang terjadi pada controller akan langsung diperbarui pada antarmuka pengguna tanpa perlu pemanggilan ulang widget secara manual. Selain itu, aplikasi juga memanfaatkan fitur **Get.snackbar** untuk memberikan notifikasi kepada pengguna setiap kali operasi berhasil dilakukan, sehingga meningkatkan pengalaman pengguna dalam berinteraksi dengan aplikasi.

Aplikasi ini memiliki antarmuka yang sederhana dan mudah digunakan, dilengkapi dengan tombol aksi untuk setiap metode HTTP serta tampilan daftar data yang ditampilkan dalam bentuk list. Program ini diharapkan dapat membantu mahasiswa memahami cara kerja REST API, penggunaan metode HTTP, serta penerapan state management modern pada pengembangan aplikasi mobile berbasis Flutter.