

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

**MODUL XIV
DATA STROAGE BAGIAN 3**



Disusun Oleh :
Dimas Abhipraya Ramansyah /
2311104069
SE-07-02

Asisten Praktikum :
Zulfa Mustafa Akhyar Iswahyudi
Yoga Eka Pratama

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

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

TUGAS PENDAHULUAN / TUGAS UNGUIDED

A. SOAL

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

B. JAWABAN

Source code

```
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;

    Future<void> getPosts() async {
        isLoading.value = true;
        await apiService.fetchPosts();
        posts.value = apiService.posts;
        isLoading.value = false;

        Get.snackbar('Sukses', 'Data berhasil diambil');
    }

    Future<void> addPost() async {
        await apiService.createPost();
        posts.value = apiService.posts;
        Get.snackbar('Sukses', 'Data berhasil ditambahkan');
    }

    Future<void> updatePost() async {
        await apiService.updatePost();
        posts.refresh();
        Get.snackbar('Sukses', 'Data berhasil diperbarui');
    }
}
```

```
Future<void> deletePost() async {
  await apiService.deletePost();
  posts.refresh();
  Get.snackbar('Sukses', 'Data berhasil dihapus');
}

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: const Text('Praktikum 14 - API GetX')),
      body: Column(
        children: [
          Wrap(
            spacing: 8,
            children: [
              ElevatedButton(
                onPressed: controller.getPosts,
                child: const Text('GET'),
              ),
              ElevatedButton(
                onPressed: controller.addPost,
                child: const Text('POST'),
              ),
              ElevatedButton(
                onPressed: controller.updatePost,
                child: const Text('UPDATE'),
              ),
              ElevatedButton(
                onPressed: controller.deletePost,
                child: const Text('DELETE'),
              ),
            ],
          ),
          Expanded(
            child: Obx(() {
              if (controller.isLoading.value) {
                return const Center(child: CircularProgressIndicator());
              }

              if (controller.posts.isEmpty) {
                return const Center(child: Text('Data belum ada'));
              }

              return ListView.builder(
                itemCount: controller.posts.length,
```



```
}

Future<void> updatePost() async {
    final response = await http.put(
        Uri.parse('$baseUrl/posts/1'),
        body: json.encode({
            'title': 'Updated Title',
            'body': 'Updated Body',
            'userId': 1,
        }),
    );

    if (response.statusCode == 200) {
        posts[0]['title'] = 'Updated Title';
        posts[0]['body'] = 'Updated Body';
    } else {
        throw Exception('Failed to update post');
    }
}

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

    if (response.statusCode == 200) {
        posts.removeAt(0);
    } else {
        throw Exception('Failed to delete post');
    }
}

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

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

class MyApp extends StatelessWidget {
    const MyApp({super.key});

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

Screenshot Output



Deskripsi Program

Aplikasi ini merupakan aplikasi Flutter yang mengimplementasikan REST API menggunakan state management GetX. Aplikasi dapat melakukan operasi GET, POST, PUT, dan DELETE terhadap data dari API. Data dikelola melalui GetX Controller dan ditampilkan secara reaktif menggunakan widget Obx. Setiap operasi menampilkan notifikasi Snackbar sebagai respon keberhasilan.