

Nama : Afad Fath Musyarof Halim  
NIM : 2211104030  
Kelas : 2211104030

## Tugas 8 - REST API

Note: Menggunakan [JSONPlaceholder - Free Fake REST API](#)

Kodingan

### 1. Main.dart

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

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

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

  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      title: 'Catatan',
      initialRoute: '/',
      getPages: [
        GetPage(name: '/', page: () => HomeScreen()),
      ],
    );
  }
}
```

## 2. Homescreen

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import '../service/getx_controller.dart';

void main() => runApp(HomeScreen());

class HomeScreen extends StatelessWidget {
  HomeScreen({super.key});

  final ControllerX_controller = Get.put(ControllerX());

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: const Text('JSONPlaceholder'),
        ),
        body: Padding(
          padding: const EdgeInsets.all(8.0),
          child: Column(
            children: [
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: [
                  ElevatedButton(
                    onPressed: _controller.fetchPosts,
                    child: const Text('Fetch Posts'),
                  ),
                  ElevatedButton(
                    onPressed: _controller.createPost,
                    child: const Text('Create Post'),
                  ),
                ],
              ),
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: [
                  ElevatedButton(
                    onPressed: _controller.updatePost,
                    child: const Text('Update Post'),
                  ),
                ],
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

```
ElevatedButton(  
  onPressed: _controller.deletePost,  
  child: const Text('Delete Post'),  
),  
],  
),  
const SizedBox(height: 10),  
Obx(  
  () {  
    if (_controller.isLoading.value) {  
      return const CircularProgressIndicator();  
    } else {  
      return Expanded(  
        child: ListView.builder(  
          itemCount: _controller.posts.length,  
          itemBuilder: (context, index) {  
            final post = _controller.posts[index];  
            return ListTile(  
              title: Text(post['title']),  
              subtitle: Text(post['body']),  
            );  
          },  
        ),  
      );  
    }  
  },  
),  
],  
),  
),  
));  
}  
}
```

### 3. API Service

```
import 'dart:convert';

import 'package:http/http.dart' as http;

class ApiService {
  final String baseUrl = 'https://jsonplaceholder.typicode.com';
  List<dynamic> posts = [];

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

    if (response.statusCode == 200) {
      posts = jsonDecode(response.body);
    } else {
      throw Exception('Failed to load posts');
    }
  }

  // Fungsi untuk POST data
  Future<void> createPost() async {
    final response = await http.post(
      Uri.parse('$baseUrl/posts'),
      headers: {'Content-Type': 'application/json'},
      body: json.encode({
        'title': 'Flutter Post',
        'body': 'Ini contoh POST.',
        'userId': 1,
      })),
    );
    if (response.statusCode == 201) {
      posts.add({
        'title': 'Flutter Post',
        'body': 'Ini contoh POST.',
        'id': posts.length + 1,
      });
    } else {
      throw Exception('Failed to create post');
    }
  }

  // Fungsi untuk UPDATE data
  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) {
            final updatedPost = posts.firstWhere((post) => post['id'] == 1);
            updatedPost['title'] = 'Updated Title';
            updatedPost['body'] = 'Updated Body';
        } else {
            throw Exception('Failed to update post');
        }
    }

    // Fungsi untuk DELETE data
    Future<void> deletePost() async {
        final response = await http.delete(
            Uri.parse('$baseUrl/posts/1'),
        );
        if (response.statusCode == 200) {
            posts.removeWhere((post) => post['id'] == 1);
        } else {
            throw Exception('Failed to delete post');
        }
    }
}

```

#### 4. getx\_controller

```
import 'package:get/get.dart';
import '../service/api_service.dart';

class ControllerX extends GetxController {
  var posts = <dynamic>[].obs;
  var isLoading = false.obs;

  final ApiService _apiService = ApiService();

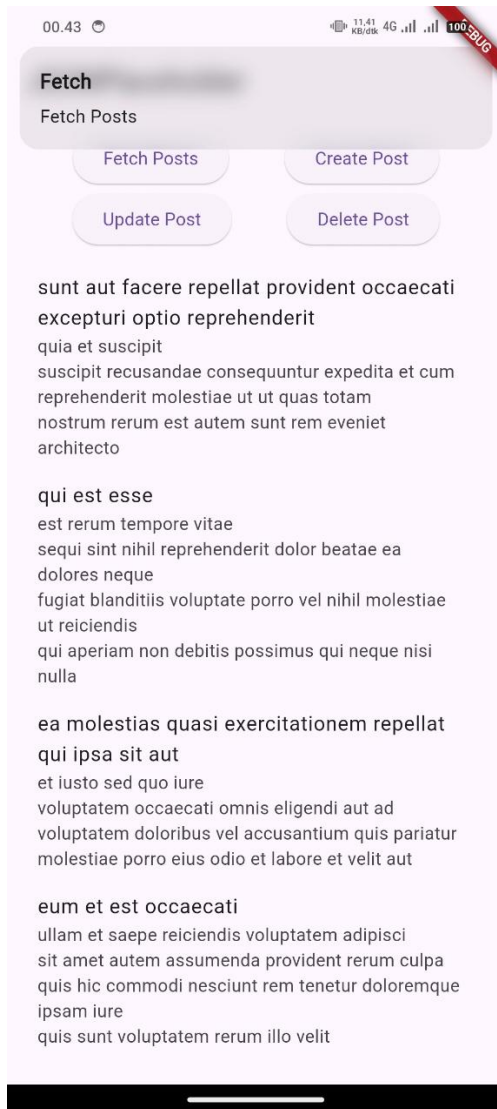
  void showSnackBar(String title, String message) {
    Get.snackbar(title, message, snackPosition: SnackPosition.TOP);
  }

  Future<void> handleApiOperation(
    Future<void> operation, String title, String successMessage) async {
    isLoading.value = true;
    try {
      await operation;
      posts.value = _apiService.posts;
      showSnackBar(title, successMessage);
    } catch (e) {
      showSnackBar("Error", e.toString());
    } finally {
      isLoading.value = false;
    }
  }

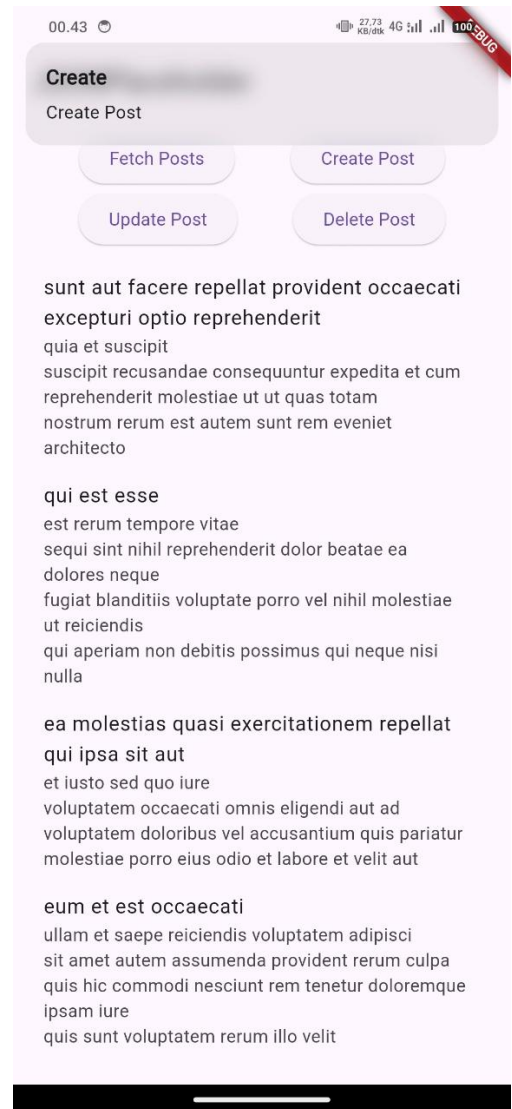
  void fetchPosts() =>
    handleApiOperation(_apiService.fetchPosts(), "Fetch", 'Fetch Posts');
  void createPost() =>
    handleApiOperation(_apiService.createPost(), "Create", 'Create Post');
  void updatePost() =>
    handleApiOperation(_apiService.updatePost(), "Update", 'Update Post');
  void deletePost() =>
    handleApiOperation(_apiService.deletePost(), "Delete", 'Delete Post');
}
```

## Hasil

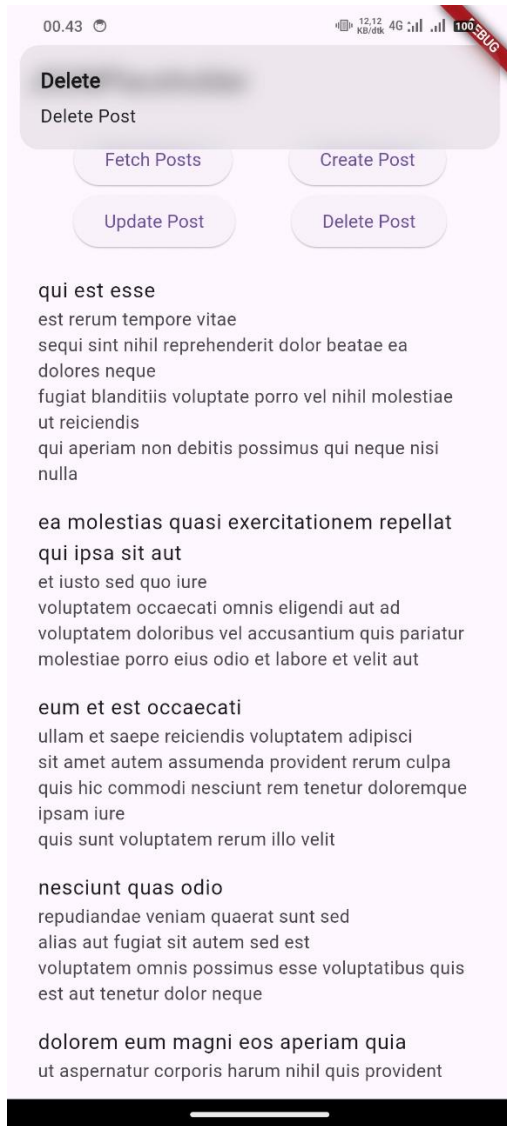
### 1. Home – Get



### 2. Home – Create



### 3. Home – Delete



### 4. Home – Update

