

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

MODUL XV
PENGENALAN DESIGN PATTERN



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

Dari tugas unguided modul 14 yang telah dikerjakan, ubahlah kodennya sehingga sesuai dengan salah satu design pattern yang telah dipelajari.

B. JAWABAN

Source code

```
class Post {  
    int id;  
    String title;  
    String body;  
  
    Post({  
        required this.id,  
        required this.title,  
        required this.body,  
    });  
  
    factory Post.fromJson(Map<String, dynamic> json) {  
        return Post(  
            id: json['id'],  
            title: json['title'],  
            body: json['body'],  
        );  
    }  
}  
  
import 'dart:convert';  
import 'package:http/http.dart' as http;  
import '../models/post_model.dart';  
  
class ApiService {  
    final String baseUrl = 'https://jsonplaceholder.typicode.com';  
  
    Future<List<Post>> fetchPosts() async {  
        final response = await http.get(Uri.parse('$baseUrl/posts'));  
        if (response.statusCode == 200) {  
            final List data = json.decode(response.body);  
            return data.map((e) => Post.fromJson(e)).toList();  
        } else {  
            throw Exception('Gagal mengambil data');  
        }  
    }  
}  
  
import 'package:get/get.dart';  
import '../models/post_model.dart';  
import '../services/api_service.dart';  
  
class PostViewModel extends GetxController {  
    final ApiService _apiService = ApiService();
```

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

Future<void> getPosts() async {
    try {
        isLoading.value = true;
        posts.value = await _apiService.fetchPosts();
        Get.snackbar('Sukses', 'Data berhasil diambil');
    } catch (e) {
        Get.snackbar('Error', 'Gagal mengambil data');
    } finally {
        isLoading.value = false;
    }
}

String get totalPost => 'Total Post: ${posts.length}';
}

import 'package:flutter/material.dart';
import 'package:get/get.dart';
import '../viewmodels/post_viewmodel.dart';

class HomeView extends StatelessWidget {
    final PostViewModel viewModel = Get.put(PostViewModel());

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(title: const Text('MVVM – API Post')),
            body: Column(
                children: [
                    ElevatedButton(
                        onPressed: viewModel.getPosts,
                        child: const Text('Ambil Data'),
                    ),
                    Obx(() => Text(viewModel.totalPost)),
                    Expanded(
                        child: Obx(() {
                            if (viewModel.isLoading.value) {
                                return const Center(child: CircularProgressIndicator());
                            }
                            return ListView.builder(
                                itemCount: viewModel.posts.length,
                                itemBuilder: (context, index) {
                                    final post = viewModel.posts[index];
                                    return Card(
                                        child: ListTile(
                                            title: Text(post.title),
                                            subtitle: Text(post.body),
                                        ),
                                    );
                                },
                            );
                        }));
                ],
            ),
        );
    }
}
```

```

        },
        ],
        ),
        );
    }
}

import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'views/home_view.dart';

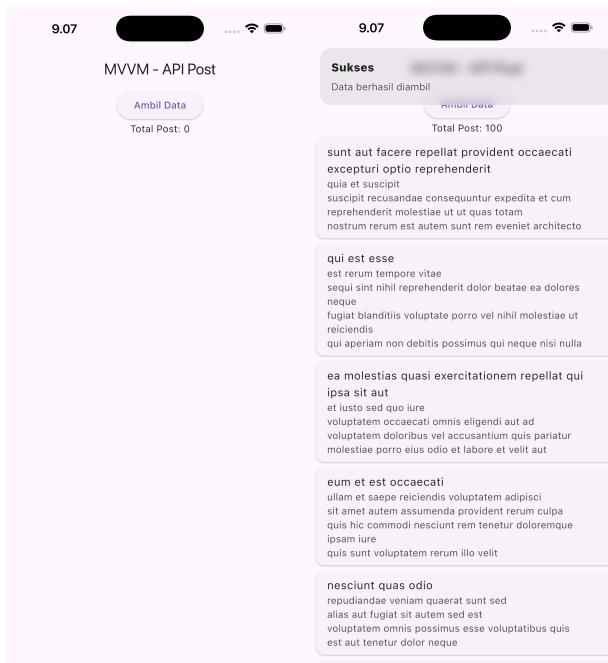
void main() {
    runApp(const MyApp());
}

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

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

```

Screenshot Output



Deskripsi Program

Aplikasi ini merupakan pengembangan dari tugas unguided modul sebelumnya dengan menerapkan design pattern MVVM. Aplikasi memisahkan komponen Model, View, dan ViewModel untuk meningkatkan keterbacaan dan pemeliharaan kode. Data diambil dari REST API dan dikelola menggunakan ViewModel berbasis GetX, kemudian ditampilkan secara reaktif pada View menggunakan Obx.