

**UNGUIDED  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL XV  
DESIGN PATTERN**



**Disusun Oleh :**

**Ilham Lii Assidaq**

**2311104068**

**Asisten Praktikum :**

**Yoga Eka Pratama**

**Zulfa Mustafa Akhyar Iswahyudi**

**Dosen Pengampu :**

**Yudha Islami Sulistya, S.Kom., M.Cs.**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

**TUGAS PENDAHULUAN**

## 1. SOAL

### Tugas Mandiri (Unguided)

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

*Note: Jangan lupa sertakan source code, screenshot output, dan deskripsi program.*

*Kreatifitas menjadi nilai tambah*

Main.dart

```
1  import 'package:flutter/material.dart';
2  import 'package:get/get.dart';
3  import 'view/home_screen.dart';
4
5  void main() {
6    runApp(const MyApp());
7  }
8
9  class MyApp extends StatelessWidget {
10    const MyApp({super.key});
11
12    @override
13    Widget build(BuildContext context) {
14      return GetMaterialApp(
15        title: 'FORM MODUL 15',
16        debugShowCheckedModeBanner: false,
17        theme: ThemeData(
18          useMaterial3: true,
19          primarySwatch: Colors.blue,
20
21          appBarTheme: const AppBarTheme(
22            elevation: 0,
23            centerTitle: true,
24          ),
25
26          cardTheme: CardThemeData(
27            elevation: 2,
28            shape: RoundedRectangleBorder(
29              borderRadius: BorderRadius.circular(12),
30            ),
31          ),
32
33          inputDecorationTheme: InputDecorationTheme(
34            contentPadding: const EdgeInsets.symmetric(
35              horizontal: 16,
36              vertical: 14,
37            ),
38            border: OutlineInputBorder(
39              borderRadius: BorderRadius.circular(8),
40              borderSide: const BorderSide(color: Colors.grey),
41            ),
42            focusedBorder: OutlineInputBorder(
43              borderRadius: BorderRadius.circular(8),
44              borderSide: const BorderSide(
45                color: Colors.blue,
46                width: 2,
47              ),
48            ),
49          ),
50        ),
51        home: HomeScreen(),
52      );
53    }
54  }
```

## action\_buttons.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import 'package:unguidedlimabelas/viewmodel/mahasiswa_viewmodel.dart';
4
5 class ActionButtons extends StatelessWidget {
6   final MahasiswaViewModel viewModel;
7
8   const ActionButtons({super.key, required this.viewModel});
9
10  @override
11  Widget build(BuildContext context) {
12    return Row(
13      children: [
14
15        Expanded(
16          child: Obx(
17            () => ElevatedButton(
18              onPressed: viewModel.isLoading.value
19                ? null
20                : () {
21                  if (viewModel.isEditMode.value) {
22                    viewModel.updateData();
23                  } else {
24                    viewModel.addData();
25                  }
26                },
27              style: ElevatedButton.styleFrom(
28                backgroundColor: viewModel.isEditMode.value
29                  ? Colors.green
30                  : Colors.blue,
31                padding: const EdgeInsets.symmetric(vertical: 16),
32                shape: RoundedRectangleBorder(
33                  borderRadius: BorderRadius.circular(10),
34                ),
35                elevation: 2,
36              ),
37              child: Row(
38                mainAxisAlignment: MainAxisAlignment.center,
39                children: [
40                  Icon(
41                    viewModel.isEditMode.value
42                      ? Icons.update
43                      : Icons.add_circle,
44                    color: Colors.white,
45                    size: 20,
46                  ),
47                  const SizedBox(width: 8),
48                  Text(
49                    viewModel.isEditMode.value ? 'UPDATE' : 'TAMBAH',
50                    style: const TextStyle(
51                      fontWeight: FontWeight.bold,
52                      fontSize: 16,
53                      color: Colors.white,
```

## mahasiswa\_form.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import '../viewmodel/mahasiswa_viewmodel.dart';
4
5 class MahasiswaForm extends StatelessWidget {
6   final MahasiswaViewModel viewModel;
7
8   const MahasiswaForm({super.key, required this.viewModel});
9
10  @override
11  Widget build(BuildContext context) {
12    return Card(
13      elevation: 3,
14      shape: RoundedRectangleBorder(
15        borderRadius: BorderRadius.circular(12),
16      ),
17      child: Padding(
18        padding: const EdgeInsets.all(16),
19        child: Column(
20          crossAxisAlignment: CrossAxisAlignment.start,
21          children: [
22
23            Obx(
24              () => Text(
25                viewModel.isEditMode.value
26                  ? 'Edit Data'
27                  : 'Tambah Data Baru',
28                style: TextStyle(
29                  fontSize: 18,
30                  fontWeight: FontWeight.bold,
31                  color: Colors.blue[800],
32                ),
33            ),
34
35            const SizedBox(height: 16),
36
37            _buildTextField(
38              label: 'Nama',
39              icon: Icons.person,
40              controller: viewModel.nameController,
41            ),
42
43            const SizedBox(height: 12),
44
45            _buildTextField(
46              label: 'NIM',
47              icon: Icons.badge,
```

## mahasiswa\_list.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import '../model/mahasiswa_model.dart';
4 import '../viewmodel/mahasiswa_viewmodel.dart';
5
6 class MahasiswaList extends StatelessWidget {
7   final MahasiswaViewModel viewModel;
8
9   const MahasiswaList({super.key, required this.viewModel});
10
11   @override
12   Widget build(BuildContext context) {
13     return Obx(() {
14       if (viewModel.isLoading.value) {
15         return _buildLoadingIndicator();
16       }
17
18       if (viewModel.mahasiswaList.isEmpty) {
19         return _buildEmptyState();
20       }
21
22       return ListView.builder(
23         itemCount: viewModel.mahasiswaList.length,
24         itemBuilder: (context, index) {
25           final mahasiswa = viewModel.mahasiswaList[index];
26           return _buildListItem(mahasiswa);
27         },
28       );
29     });
30   }
31
32   Widget _buildLoadingIndicator() {
33     return Center(
34       child: Column(
35         mainAxisAlignment: MainAxisAlignment.center,
36         children: [
37           const CircularProgressIndicator(
38             strokeWidth: 3,
39             valueColor: AlwaysStoppedAnimation(Colors.blue),
40           ),
41           const SizedBox(height: 16),
42           Text(
43             'Memuat data...',
44             style: TextStyle(
45               fontSize: 14,
46               color: Colors.grey[600],
47             ),
48           ),
49         ],
50       ),
51     );
52   }
53
54   Widget _buildEmptyState() {
55     return Center(
56       child: Column(
57         mainAxisAlignment: MainAxisAlignment.center,
58         children: [
59           Icon(
60             Icons.list_alt,
61             size: 64,
62             color: Colors.grey[400],
63           ),
64           const SizedBox(height: 16),
65           const Text(
66             'Belum ada data',
67             style: TextStyle(
68               fontSize: 16,
69               color: Colors.grey,
70               fontWeight: FontWeight.w500,
71             ),
72           ),
73           const SizedBox(height: 8),
74           Text(
75             'Tekan tombol GET untuk mengambil data',
76             style: TextStyle(
77               fontSize: 12,
78               color: Colors.grey[500],
79             ),
80           ),
81         ],
82       ),
83     );
84   }
85
86   Widget _buildListItem(Mahasiswa mahasiswa) {
87     return Card(
88       elevation: 2,
89       margin: const EdgeInsets.only(bottom: 12),
90       shape: RoundedRectangleBorder(
91         borderRadius: BorderRadius.circular(10),
92       ),
93       child: Padding(
94         padding: const EdgeInsets.all(16),
95         child: Row(
96           children: [
97             Container(
98               width: 40,
99               height: 40,
100              decoration: BoxDecoration(
101                color: Colors.blue[50],
102                borderRadius: BorderRadius.circular(20),
103              ),
104              child: Icon(
105                Icons.person,
106                color: Colors.blue[600],
107              ),
108            ),
109            const SizedBox(width: 16),
110          ],
111        ),
112      ),
113
114      Expanded(
115        child: Column(
116          crossAxisAlignment: CrossAxisAlignment.start,
117          children: [
118            Text(
119              mahasiswa.name,
120              maxLines: 1,
121              overflow: TextOverflow.ellipsis,
122              style: const TextStyle(
123                fontWeight: FontWeight.bold,
124                fontSize: 14,
125              ),
126            ),
127            const SizedBox(height: 4),
128            Text(
129              mahasiswa.nim,
130              style: TextStyle(
131                fontSize: 14,
132                color: Colors.grey[600],
133              ),
134            ),
135            if (mahasiswa.createdAt != null)
136              Padding(
137                padding: const EdgeInsets.only(top: 4),
138                child: Text(
139                  'Dibuatkan: ${mahasiswa.createdAt.toLocal().toString().split('.')[0]}',
140                  style: TextStyle(
141                    fontSize: 10,
142                    color: Colors.grey[500],
143                  ),
144                ),
145              ),
146          ],
147        ),
148      ),
149    );
150
151    Row(
152      children: [
153        IconButton(
154          onPressed: () => viewModel.setEditMode(mahasiswa),
155          icon: Icon(Icons.edit, color: Colors.green[600]),
156          tooltip: 'Edit',
157          style: IconButton.styleFrom(
158            backgroundColor: Colors.green[50],
159            shape: RoundedRectangleBorder(
160              borderRadius: BorderRadius.circular(8),
161            ),
162          ),
163        ),
164        const SizedBox(width: 8),
165        IconButton(
166          onPressed: () => _showDeleteDialog(mahasiswa),
167          icon: Icon(Icons.delete, color: Colors.red[600]),
168          tooltip: 'Hapus',
169          style: IconButton.styleFrom(
170            backgroundColor: Colors.red[50],
171            shape: RoundedRectangleBorder(
172              borderRadius: BorderRadius.circular(8),
173            ),
174          ),
175        ),
176      ],
177    ),
178  ],
179 ),
180 ),
181 ),
182 );
183
184 void _showDeleteDialog(Mahasiswa mahasiswa) {
185   Get.defaultDialog(
186     title: 'Konfirmasi Hapus',
187     titleStyle: const TextStyle(
188       fontWeight: FontWeight.bold,
189       color: Colors.red,
190     ),
191     content: Column(
192       children: [
193         const Text('Apakah Anda yakin ingin menghapus data ini?'),
194         const SizedBox(height: 8),
195         Text(
196           '${mahasiswa.name} (${mahasiswa.nim})',
197           style: const TextStyle(
198             fontWeight: FontWeight.bold,
199           ),
200         ),
201       ],
202     ),
203     confirm: ElevatedButton(
204       onPressed: () {
205         Get.back();
206         viewModel.deleteData(mahasiswa.id);
207       },
208       style: ElevatedButton.styleFrom(
209         backgroundColor: Colors.red,
210         padding: const EdgeInsets.symmetric(
211           horizontal: 24,
212           vertical: 12,
213         ),
214       ),
215     ),
216     child: const Text(
217       'HAPUS',
218       style: TextStyle(color: Colors.white),
219     ),
220     cancel: TextButton(
221       onPressed: () => Get.back(),
222       child: const Text('BATAL'),
223     ),
224   );
225 }
```

## Home\_screen.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import '../viewModel/mahasiswa_viewmodel.dart';
4 import 'widget/mahasiswa_form.dart';
5 import 'widget/action_buttons.dart';
6 import 'widget/mahasiswa_list.dart';
7
8 class HomeScreen extends StatelessWidget {
9   HomeScreen({super.key});
10
11   final MahasiswaViewModel viewModel =
12     Get.put(MahasiswaViewModel());
13
14   @override
15   Widget build(BuildContext context) {
16     return Scaffold(
17       backgroundColor: Colors.grey[50],
18       appBar: AppBar(
19         centerTitle: true,
20         elevation: 0,
21         backgroundColor: Colors.blue[700],
22         foregroundColor: Colors.white,
23         title: const Text(
24           'FORM MODUL 15',
25           style: TextStyle(
26             fontWeight: FontWeight.bold,
27             fontSize: 22,
28             letterSpacing: 0.5,
29           ),
30         ),
31         actions: [
32           Obx(
33             () => viewModel.isEditMode.value
34             ? IconButton(
35               onPressed: viewModel.cancelEdit,
36               icon: const Icon(Icons.close),
37               tooltip: 'Batal Edit',
38             )
39             : const SizedBox(),
40           ),
41         ],
42       ),
43       body: Padding(
44         padding: const EdgeInsets.all(16),
45         child: Column(
46           crossAxisAlignment: CrossAxisAlignment.stretch,
47           children: [
48             MahasiswaForm(viewModel: viewModel),
49
50             const SizedBox(height: 20),
51
52             // Action Buttons
53             ActionButtons(viewModel: viewModel),
54
55             const SizedBox(height: 20),
56
```

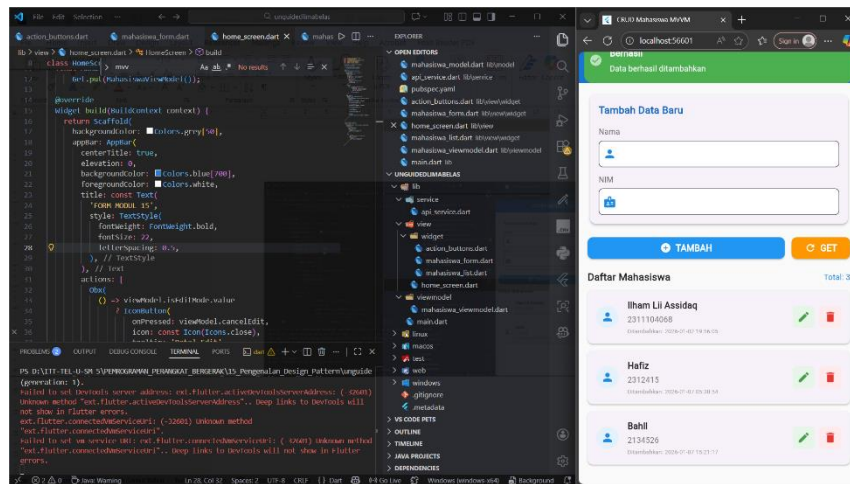
```
57           // List Header
58           _buildListHeader(),
59
60           const SizedBox(height: 10),
61
62           // List
63           Expanded(
64             child: MahasiswaList(viewModel: viewModel),
65           ),
66         ],
67       ),
68     ),
69   );
70 }
71
72 Widget _buildListHeader() {
73   return Row(
74     mainAxisAlignment: MainAxisAlignment.spaceBetween,
75     children: [
76       const Text(
77         'Daftar Mahasiswa',
78         style: TextStyle(
79           fontSize: 18,
80           fontWeight: FontWeight.bold,
81           color: Colors.black87,
82         ),
83     ),
84     Obx(
85       () => Text(
86         'Total: ${viewModel.totalMahasiswa}',
87         style: TextStyle(
88           fontSize: 14,
89           fontWeight: FontWeight.w500,
90           color: Colors.blue[600],
91         ),
92     ),
93   ),
94 ],
95 );
96 }
97 }
98
```

## Mahasiswa\_viewmodel.dart

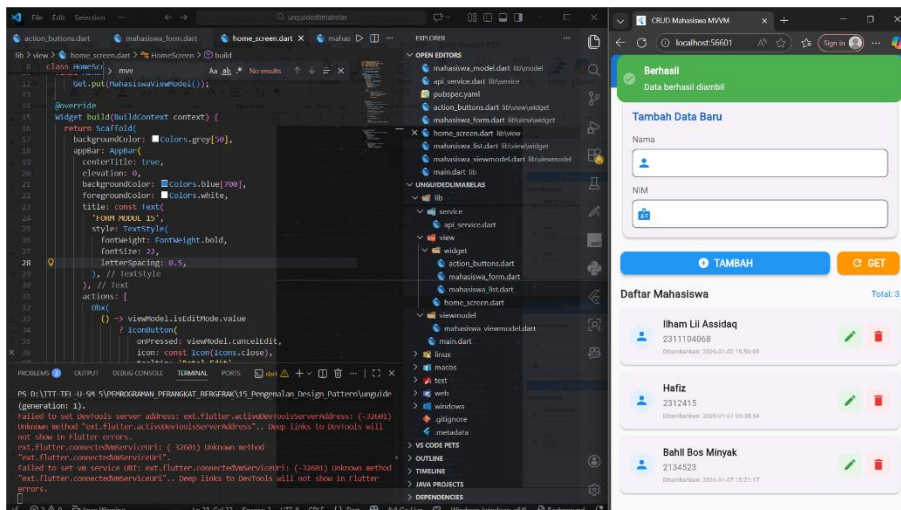
```
1 import 'package:flutter/material.dart';
2 import 'package:get/get.dart';
3 import '../model/mahasiswa_model.dart';
4 import '../service/api_service.dart';
5
6 class MahasiswaViewModel extends GetxController {
7   final ApiService _apiService = ApiService();
8
9   final RxList<Mahasiswa> mahasiswaList = <Mahasiswa>[].obs;
10   final RxBool isLoading = false.obs;
11   final RxBool isEditMode = false.obs;
12   final RxString selectedId = ''.obs;
13   final RxString errorMessage = ''.obs;
14
15   final TextEditingController nameController =
16     TextEditingController();
17   final TextEditingController nimController =
18     TextEditingController();
19
20   int get totalMahasiswa => mahasiswaList.length;
21
22   bool get hasError => errorMessage.isNotEmpty;
23
24   bool get isValid =>
25     nameController.text.isNotEmpty &&
26     nimController.text.isNotEmpty;
27
28   @override
29   void onClose() {
30     nameController.dispose();
31     nimController.dispose();
32     super.onClose();
33   }
34
35   Future<void> fetchData() async {
36     try {
37       isLoading(true);
38       errorMessage('');
39
40       final data = await _apiService.fetchMahasiswa();
41       mahasiswaList.assignAll(data);
42
43       showSuccessSnackBar('Data berhasil diambil');
44     } catch (e) {
45       errorMessage(e.toString());
46       showErrorSnackBar('Gagal mengambil data');
47     } finally {
48       isLoading(false);
49     }
50   }
51
52   Future<void> addData() async {
53     if (!isValid) {
54       showWarningSnackBar('Nama dan NIM tidak boleh kosong');
55       return;
56     }
57
58     try {
59       isLoading(true);
60       errorMessage('');
61
62       await _apiService.addMahasiswa(
63         nameController.text,
64         nimController.text,
65       );
66
67       await fetchData();
68       clearForm();
69
70       showSuccessSnackBar('Data berhasil ditambahkan');
71     } catch (e) {
72       errorMessage(e.toString());
73       showErrorSnackBar('Gagal menambah data');
74     } finally {
75       isLoading(false);
76     }
77   }
78
79   Future<void> updateData() async {
80     if (selectedId.isEmpty) return;
81
82     if (!isValid) {
83       showWarningSnackBar('Nama dan NIM tidak boleh kosong');
84       return;
85     }
86
87     try {
88       isLoading(true);
89       errorMessage('');
90
91       await _apiService.updateMahasiswa(
92         selectedId.value,
93         nameController.text,
94         nimController.text,
95       );
96
97
98       await fetchData();
99       cancelEdit();
100
101       showSuccessSnackBar('Data berhasil diperbarui');
102     } catch (e) {
103       errorMessage(e.toString());
104       showErrorSnackBar('Gagal memperbarui data');
105     } finally {
106       isLoading(false);
107     }
108   }
109
110   Future<void> deleteData(String id) async {
111     try {
112       isLoading(true);
113       errorMessage('');
114
115       await _apiService.deleteMahasiswa(id);
116       await fetchData();
117
118       showSuccessSnackBar('Data berhasil dihapus');
119     } catch (e) {
120       errorMessage(e.toString());
121       showErrorSnackBar('Gagal menghapus data');
122     } finally {
123       isLoading(false);
124     }
125   }
126
127   void setEditMode(Mahasiswa mahasiswa) {
128     isEditMode(true);
129     selectedId(mahasiswa.id);
130     nameController.text = mahasiswa.name;
131     nimController.text = mahasiswa.nim;
132   }
133
134   void cancelEdit() {
135     isEditMode(false);
136     selectedId('');
137     clearForm();
138   }
139
140   void clearForm() {
141     nameController.clear();
142     nimController.clear();
143   }
144
145   void showSuccessSnackBar(String message) {
146     Get.snackbar(
147       'Berhasil',
148       message,
149       snackPosition: SnackPosition.TOP,
150       backgroundColor: Colors.green,
151       colorText: Colors.white,
152       duration: const Duration(seconds: 2),
153       icon: const Icon(Icons.check_circle, color: Colors.white),
154     );
155   }
156
157   void showErrorSnackBar(String message) {
158     Get.snackbar(
159       'Error',
160       message,
161       snackPosition: SnackPosition.BOTTOM,
162       backgroundColor: Colors.red,
163       colorText: Colors.white,
164       duration: const Duration(seconds: 3),
165     );
166   }
167
168   void showWarningSnackBar(String message) {
169     Get.snackbar(
170       'Peringatan',
171       message,
172       snackPosition: SnackPosition.TOP,
173       backgroundColor: Colors.orange,
174       colorText: Colors.white,
175     );
176   }
177
178   void updateMahasiswa() {}
179
180   void addMahasiswa() {}
181
182
183
184
185
```

## Screenshot Program

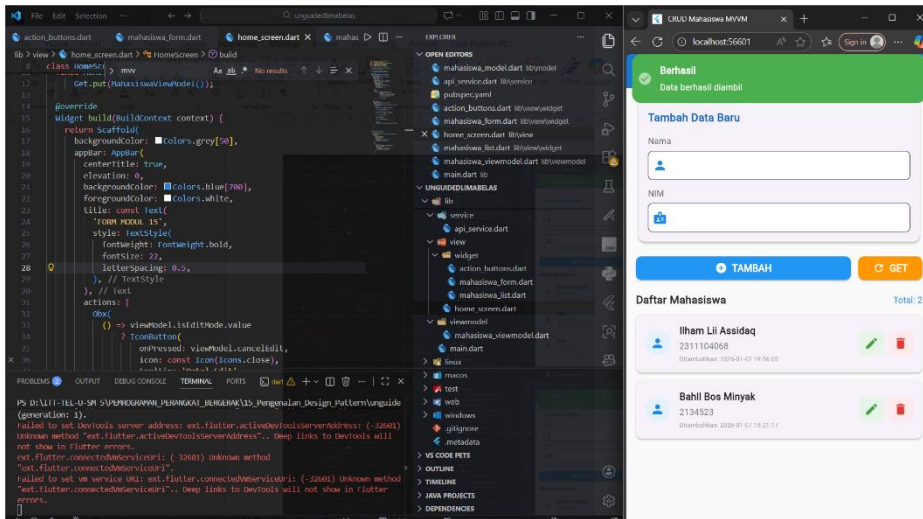
### 1. addMahasiswa



### 2. editMahasiswa



### 3. deleteMahasiswa



## Penjelasan

Program Flutter ini menerapkan arsitektur MVVM (Model–View–ViewModel) dengan bantuan GetX sebagai state management. ApiService berperan sebagai lapisan service yang menangani komunikasi HTTP ke server (MockAPI), mencakup operasi CRUD seperti mengambil, menambah, memperbarui, dan menghapus data mahasiswa menggunakan metode GET, POST, PUT, dan DELETE. Data yang diterima dari API dikonversi dari format JSON ke objek Mahasiswa sehingga mudah dikelola di dalam aplikasi. Pendekatan ini memisahkan logika jaringan dari UI, membuat kode lebih terstruktur, mudah diuji, dan mudah dikembangkan.

MahasiswaViewModel berfungsi sebagai pusat logika aplikasi dan pengelola state dengan variabel reaktif (Rx) seperti mahasiswaList, isLoading, dan isEditMode. ViewModel juga menangani validasi form, proses submit data, serta menampilkan feedback ke pengguna melalui snackbar. Sementara itu, widget ActionButtons bertugas sebagai lapisan tampilan (View) yang hanya menangani interaksi pengguna. Dengan bantuan Obx, tampilan tombol secara otomatis menyesuaikan kondisi aplikasi (mode tambah/update dan loading) tanpa perlu setState. Kombinasi ini menghasilkan alur data yang jelas, UI yang responsif, dan menghindari kesalahan logika karena pemisahan tanggung jawab antar komponen.