

UNGUIDED 13



Disusun Oleh :
Nama : Ganes Gemi Putra
Kelas : SE-07-02
NIM : (2311104075)

Dosen : YUDHA ISLAMI SULISTYA S.Kom M.Cs

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2025

BAB I – PENDAHULUAN

1.1 Latar Belakang

(teori singkat kenapa aplikasi catatan dan GetX digunakan — aku bisa bantu tulis jika kamu mau)

1.2 Tujuan

- Mengimplementasikan Data Storage API dalam aplikasi Flutter
- Menggunakan GetX untuk state management dan routing
- Mampu menambahkan dan menghapus data secara dinamis
- Membuat aplikasi sesuai standar arsitektur MVx sederhana (Model-Controller-View)

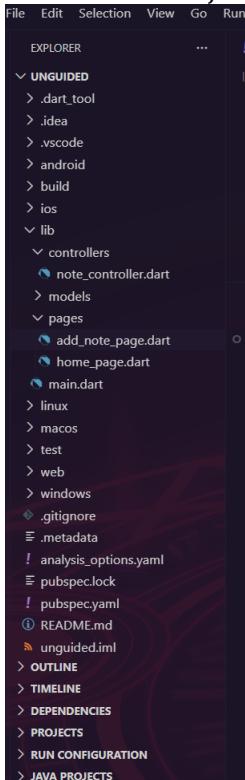
BAB II – LANDASAN TEORI

Isi teori singkat tentang:

- Web Service (SOAP & REST)
- Data Storage API
- Flutter & GetX (State Management & Routing)

BAB III – PERANCANGAN & IMPLEMENTASI

3.1 Struktur Project



3.2 Flow Program (Diagram sederhana)

Misal:

HomePage → AddNotePage → Save → Back to HomePage

3.3 Source Code Program

➔ **main.dart :**

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'controllers/note_controller.dart';
```

```
import 'pages/home_page.dart';
import 'pages/add_note_page.dart';

void main() {
    // Register controller sekali saja, bisa diakses di seluruh app
    Get.put(NoteController(), permanent: true);
    runApp(const MyApp());
}

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

    @override
    Widget build(BuildContext context) {
        return GetMaterialApp(
            debugShowCheckedModeBanner: false,
            title: 'Catatan GetX',
            theme: ThemeData(
                primarySwatch: Colors.indigo,
                scaffoldBackgroundColor: const Color(0xFFFF5F5F7),
                appBarTheme: const AppBarTheme(
                    backgroundColor: Colors.indigo,
                    elevation: 0,
                    centerTitle: true,
                    titleTextStyle: TextStyle(
                        fontSize: 20,
                        fontWeight: FontWeight.w600,
                        color: Colors.white,
                    ),
                ),
                floatingActionButtonTheme: const FloatingActionButtonThemeData(
                    backgroundColor: Colors.indigo,
                ),
                elevatedButtonTheme: ElevatedButtonThemeData(
                    style: ElevatedButton.styleFrom(
                        padding: const EdgeInsets.symmetric(vertical: 14),
                        shape: RoundedRectangleBorder(
                            borderRadius: BorderRadius.circular(14),
                        ),
                    ),
                ),
                snackBarTheme: const SnackBarThemeData(
                    behavior: SnackBarBehavior.floating,
                ),
            ),
            initialRoute: '/',
            getPages: [
                GetPage(
                    name: '/',
                    page: () => const HomePage(),
                ),
                GetPage(
                    name: '/add',
                    page: () => AddNotePage(),
                ),
            ],
        );
    }
}
```

◆ note_controller.dart :

```
import 'package:get/get.dart';
import '../models/note_model.dart';

class NoteController extends GetxController {
    // List catatan yang reaktif
    final RxList<Note> notes = <Note>[].obs;

    // Tambah catatan baru
    void addNote({
        required String title,
        required String description,
    }) {
        final note = Note(
            id: DateTime.now().millisecondsSinceEpoch,
            title: title,
            description: description,
            createdAt: DateTime.now(),
        );
        notes.add(note);
    }

    // Hapus catatan berdasarkan id
    void removeNoteById(int id) {
        notes.removeWhere((note) => note.id == id);
    }
}
```

◆ note_model.dart :

```
class Note {
    final int id;
    final String title;
    final String description;
    final DateTime createdAt;

    Note({
        required this.id,
        required this.title,
        required this.description,
        required this.createdAt,
    });
}
```

◆ home_page.dart :

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import '../controllers/note_controller.dart';
import '../models/note_model.dart';

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

    @override
    Widget build(BuildContext context) {
        final NoteController noteController = Get.find<NoteController>();

        return Scaffold(
            appBar: AppBar(
                title: const Text('Catatan Sederhana'),
            ),
            body: Container(
                padding: const EdgeInsets.all(16),
                child: Column(
                    children: [
                        Text('Catatan Sederhana'),
                        Text('Tampilan awal aplikasi Catatan Sederhana. Di sini, pengguna dapat menambahkan catatan baru dengan memberikan judul dan deskripsi. Setiap catatan akan ditunjukkan dengan detailnya, termasuk tanggal pembuatan.'),
                    ],
                ),
            ),
        );
    }
}
```



```

),
body: Obx(() {
if (noteController.notes.isEmpty) {
return Center(
child: Padding(
padding: const EdgeInsets.all(32.0),
child: Column(
mainAxisSize: MainAxisSize.min,
children: const [
Icon(
Icons.note_alt_outlined,
size: 80,
color: Color(0xFF5A55DA),
),
SizedBox(height: 16),
Text(
'Belum ada catatan',
style: TextStyle(
fontSize: 20,
fontWeight: FontWeight.w600,
),
),
SizedBox(height: 8),
Text(
'Tekan tombol "Catatan" di kanan bawah\nuntuk menambahkan catatan baru.',
textAlign: TextAlign.center,
style: TextStyle(
fontSize: 13,
color: Colors.black54,
),
),
],
),
],
),
),
);
}

return ListView.separated(
padding: const EdgeInsets.fromLTRB(16, 16, 16, 80),
itemCount: noteController.notes.length,
separatorBuilder: (__, __) => const SizedBox(height: 10),
itemBuilder: (context, index) {
final Note note = noteController.notes[index];

return Card(
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(18),
),
elevation: 3,
shadowColor: Colors.black12,
child: ListTile(
contentPadding:
const EdgeInsets.symmetric(vertical: 10, horizontal: 14),
leading: Container(
width: 42,
height: 42,
decoration: BoxDecoration(
color: Colors.indigo.withOpacity(0.1),
borderRadius: BorderRadius.circular(12),
),
),
child: Center(
child: Text(
(index + 1).toString(),
style: const TextStyle(
fontWeight: FontWeight.bold,

```

```
        color: Colors.indigo,
    ),
),
),
),
title: Text(
    note.title,
style: const TextStyle(
    fontWeight: FontWeight.w600,
    fontSize: 16,
),
),
subtitle: Padding(
    padding: const EdgeInsets.only(top: 4.0),
child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
children: [
    Text(
        note.description,
        maxLines: 2,
        overflow: TextOverflow.ellipsis,
        style: const TextStyle(
            fontSize: 14,
        ),
    ),
    const SizedBox(height: 6),
    Text(
        _formatDate(note.createdAt),
        style: const TextStyle(
            fontSize: 11,
            color: Colors.black54,
        ),
    ),
],
),
),
),
),
trailing: IconButton(
    icon: const Icon(
        Icons.delete_outline,
        color: Colors.redAccent,
    ),
    onPressed: () {
        _showDeleteDialog(
            context: context,
            note: note,
            controller: noteController,
        );
    },
),
),
),
),
),
),
floatingActionButton: Container(
decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(16),
gradient: const LinearGradient(
    colors: [
        Color(0xFF5A55DA),
        Color(0xFF3B1F8A),
    ],
begin: Alignment.topLeft,
end: Alignment.bottomRight,
),
```



```
        ],
        child: FloatingActionButton.extended(
            backgroundColor: Colors.transparent,
            elevation: 0,
            onPressed: () => Get.toNamed('/add'),
            icon: const Icon(Icons.add, color: Colors.white),
            label: const Text(
                'Catatan',
                style: TextStyle(color: Colors.white),
            ),
        ),
    ],
),
);
}
}

String _formatDate(DateTime dateTime) {
// Format sederhana: 12/10/2025 14:30
final d = dateTime.day.toString().padLeft(2, '0');
final m = dateTime.month.toString().padLeft(2, '0');
final y = dateTime.year.toString();
final h = dateTime.hour.toString().padLeft(2, '0');
final min = dateTime.minute.toString().padLeft(2, '0');
return '$d/$m/$y $h:$min';
}

void _showDeleteDialog({
    required BuildContext context,
    required Note note,
    required NoteController controller,
}) {
    showDialog(
        context: context,
        builder: (ctx) => AlertDialog(
            title: const Text('Hapus Catatan'),
            content: const Text('Yakin ingin menghapus catatan ini?'),
            actions: [
                TextButton(
                    onPressed: () => Navigator.of(ctx).pop(),
                    child: const Text('Batal'),
                ),
                TextButton(
                    onPressed: () {
                        controller.removeNoteById(note.id);
                        Navigator.of(ctx).pop();
                        Get.snackbar(
                            'Berhasil',
                            'Catatan dihapus',
                            snackPosition: SnackPosition.BOTTOM,
                            margin: const EdgeInsets.all(12),
                        );
                    },
                    child: const Text(
                        'Hapus',
                        style: TextStyle(color: Colors.red),
                    ),
                ),
            ],
        );
    );
}
```

📌 add_note_page.dart :

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import '../controllers/note_controller.dart';

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

  final _formKey = GlobalKey<FormState>();
  final TextEditingController titleController = TextEditingController();
  final TextEditingController descController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    final NoteController noteController = Get.find<NoteController>();

    return Scaffold(
      appBar: AppBar(
        title: const Text('Tambah Catatan'),
      ),
      body: SingleChildScrollView(
        padding: const EdgeInsets.all(20.0),
        child: Form(
          key: _formKey,
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              const Text(
                'Buat catatan baru',
                style: TextStyle(
                  fontSize: 18,
                  fontWeight: FontWeight.w600,
                ),
              ),
              const SizedBox(height: 6),
              const Text(
                'Isi judul dan deskripsi singkat untuk menyimpan catatanmu.',
                style: TextStyle(
                  fontSize: 13,
                  color: Colors.black54,
                ),
              ),
              const SizedBox(height: 24),
              TextFormField(
                controller: titleController,
                decoration: InputDecoration(
                  labelText: 'Judul Catatan',
                  hintText: 'Misal: Belajar GetX',
                  filled: true,
                  fillColor: Colors.white,
                  border: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(14),
                  ),
                ),
                validator: (value) {
                  if (value == null || value.trim().isEmpty) {
                    return 'Judul tidak boleh kosong';
                  }
                  if (value.trim().length < 3) {
                    return 'Judul minimal 3 karakter';
                  }
                  return null;
                },
              ),
              const SizedBox(height: 16),
```

```
TextField(  
  controller: descController,  
  maxLines: 5,  
  decoration: InputDecoration(  
    labelText: 'Deskripsi',  
    alignLabelWithHint: true,  
    hintText: 'Tuliskan detail atau poin penting di sini...',  
    filled: true,  
    fillColor: Colors.white,  
    border: OutlineInputBorder(  
      borderRadius: BorderRadius.circular(14),  
    ),  
  ),  
  validator: (value) {  
    if (value == null || value.trim().isEmpty) {  
      return 'Deskripsi tidak boleh kosong';  
    }  
    return null;  
  },  
),  
const SizedBox(height: 28),  
SizedBox(  
  width: double.infinity,  
  child: ElevatedButton.icon(  
    onPressed: () {  
      if (_formKey.currentState!.validate()) {  
        return;  
      }  
  
      final title = titleController.text.trim();  
      final desc = descController.text.trim();  
  
      noteController.addNote(  
        title: title,  
        description: desc,  
      );  
  
      Get.snackbar(  
        'Berhasil',  
        'Catatan ditambahkan',  
        snackPosition: SnackPosition.BOTTOM,  
        margin: const EdgeInsets.all(12),  
      );  
  
      Get.offNamed('');  
    },  
    icon: const Icon(Icons.check),  
    label: const Text(  
      'SIMPAN CATATAN',  
      style: TextStyle(  
        fontWeight: FontWeight.w600,  
        letterSpacing: 0.5,  
      ),  
    ),  
  ),  
),  
],  
),  
);  
};  
}  
}
```

BAB IV – HASIL & PEMBAHASAN

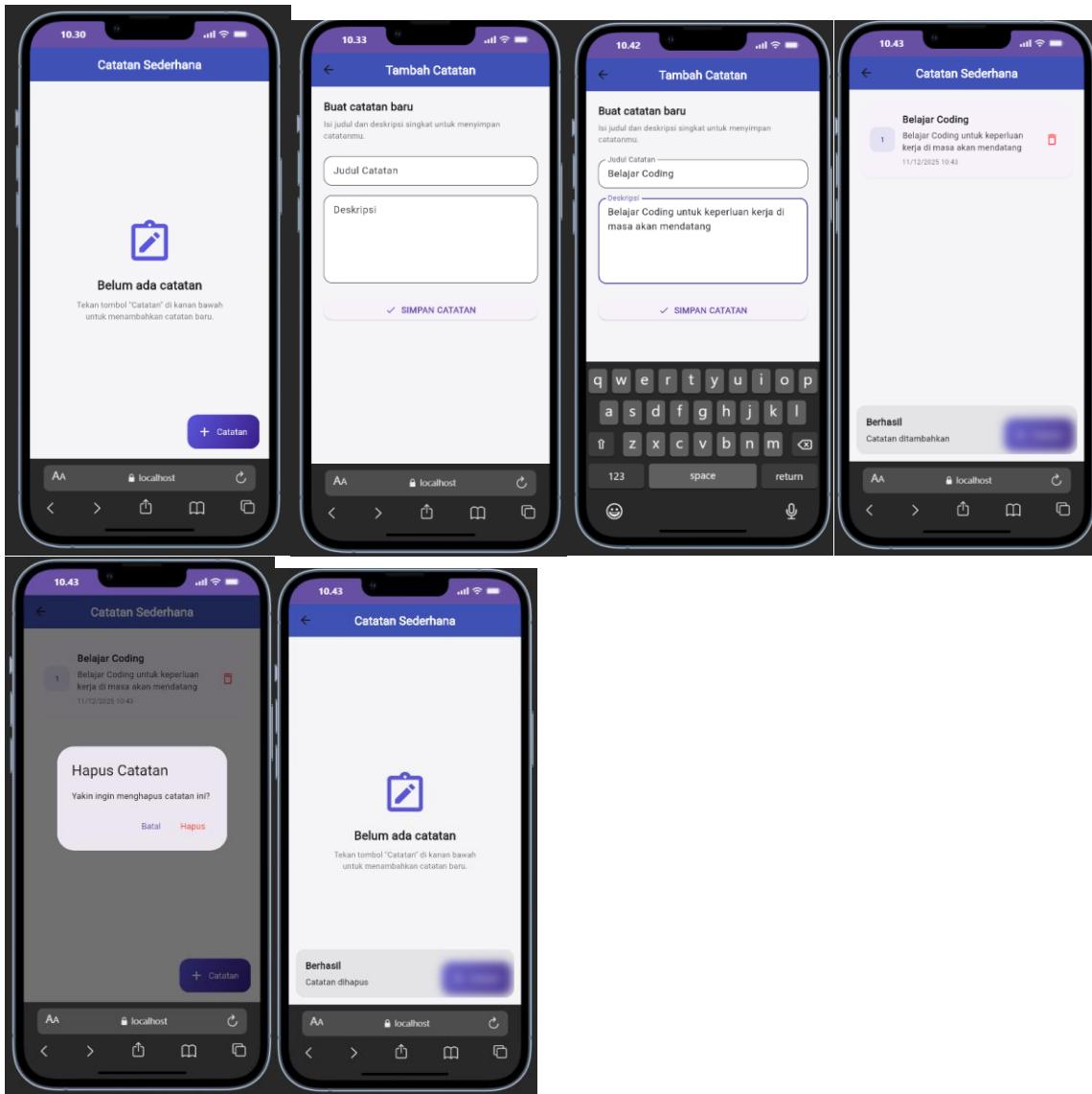
4.1 Screenshot Hasil Aplikasi

Flutter run :

```
PS D:\PPB_231104075_GAMES_GEMI_PURWOKERTO13_Networking\UNGUIDED\unguided flutter run
Debug service listening on ws://127.0.0.1:54015/y0jy208n4h-/ws
this app is linked to the debug service: ws://127.0.0.1:54015/y0jy208n4h-/ws
Debug service listening on ws://127.0.0.1:54015/y0jy208n4h-/ws

flutter run key commands:
? flutter run key commands.
? flutter run key commands.
e Hot reload.
r Hot restart.
l List (available interactive commands).
D Dash (minimize "flutter run" but leave application running).
c Clear the screen.
q Quit (terminate the application on the device).

A dart VM Service on Edge is available at: http://127.0.0.1:54015/y0jy208n4h-
Starting application from main method in: org.dartlang.app/web_entrypoint.dart.
The flutter devtools debugger and profiler on Edge is available at: http://127.0.0.1:54015/y0jy208n4h-
```



Gambar 1 – Halaman Utama Saat Tidak Ada Catatan

Menampilkan kondisi awal aplikasi. Karena belum ada data, aplikasi menunjukkan *empty state* dengan pesan “Belum ada catatan” dan tombol **+ Catatan** untuk menambah catatan baru.

Gambar 2 – Halaman Tambah Catatan

Pengguna mengisi **Judul Catatan** dan **Deskripsi** pada form yang disediakan. Setelah tombol **Simpan Catatan** ditekan, data dikirim ke controller dan aplikasi otomatis kembali ke halaman utama.

Gambar 3 – Catatan Berhasil Ditambahkan

Catatan yang telah disimpan muncul di daftar dengan:

- Judul catatan
- Deskripsi
- Tanggal dan waktu dibuat
- Tombol hapus

Terdapat notifikasi “*Berhasil – Catatan ditambahkan*”, menandakan proses penyimpanan berhasil.

Gambar 4 – Dialog Konfirmasi Hapus

Saat tombol hapus ditekan, muncul dialog konfirmasi untuk memastikan pengguna benar-benar ingin menghapus catatan, agar tidak terjadi penghapusan tanpa sengaja.

Gambar 5 – Catatan Dihapus

Setelah dikonfirmasi, catatan terhapus dan UI kembali menampilkan *empty state*. Notifikasi “*Berhasil – Catatan dihapus*” muncul sebagai tanda bahwa penghapusan berhasil.

4.2 Pembahasan

- Ketika tombol simpan ditekan → data disimpan di Controller → list berubah → UI update otomatis dengan Obx
- **Tidak perlu refresh** → reaktif dari GetX
- Routing bekerja dengan baik (Get.toNamed() dan Get.back())

BAB V – KESIMPULAN

1. Aplikasi **Catatan Sederhana** berhasil dibuat menggunakan Flutter dan GetX dengan fitur utama menambah dan menghapus catatan secara dinamis.
2. Penggunaan **GetX Controller** memungkinkan pengelolaan data (state management) menjadi lebih mudah, reaktif, dan efisien tanpa melakukan refresh manual.
3. Implementasi **GetX Routing** mempermudah navigasi antar halaman, sehingga proses berpindah dari halaman utama ke halaman tambah catatan berjalan lancar.
4. Fitur penyimpanan dan penghapusan catatan dapat berjalan dengan baik, ditandai dengan perubahan tampilan yang otomatis berkat penggunaan Obx.
5. Aplikasi mampu menampilkan *empty state* ketika tidak ada catatan dan memperbarui tampilan saat data ditambahkan atau dihapus, membuktikan bahwa integrasi Data Storage API dan GetX berjalan sesuai kebutuhan.