

**LAPORAN GUIDED & UNGUIDED**  
**PEMROGRAMAN PERANGKAT BERGERAK**  
**MODUL XIII**  
**NETWORKING**



**Disusun Oleh :**  
**Ahmad Junaidi / 2211104002**  
**SE-06-01**

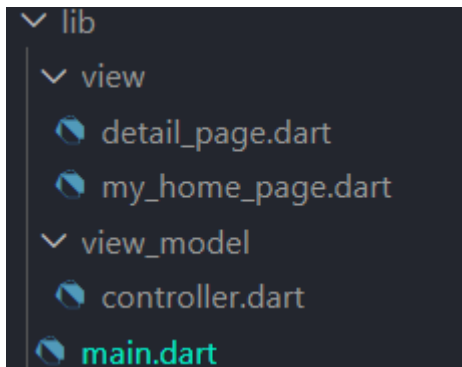
**Asisten Praktikum :**  
**Ayu Susilowati**  
**Noviana Rizki Anisa Putri**

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

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024**

## GUIDED

### Struktur File:



### Source Code main.dart:

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:pertemuan_13/view/detail_page.dart';
import 'package:pertemuan_13/view/my_home_page.dart';

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

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

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

### Source Code controller.dart :

```
import 'package:flutter/material.dart';
```

```

import 'package:get/get.dart';

class CounterController extends GetxController {
  var counter = 0.obs;

  //Fungsi untuk menambah
  void incrementCounter() {
    counter++;
  }

  //Fungsi untuk mengurangi
  void decrementCounter() {
    counter--;
  }

  void getSnackBar() {
    Get.snackbar(
      'GetX Snackbr',
      'Ini adalah getX',
      backgroundColor: Colors.amber,
      colorText: Colors.black,
    );
  }

  void getbottomshet() {
    Get.bottomSheet(
      Container(
        height: 200,
        color: Colors.green,
        child: const Center(
          child: Text(
            'Ini adalah getX BottomSheet',
            style: TextStyle(
              color: Colors.white,
              fontSize: 20,
            ),
          ),
        ),
      ),
    );
  }
}

```

Source Code my\_home\_page.dart:

```

import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:pertemuan_13/view_model/controller.dart';

class MyHomePage extends StatelessWidget {
  MyHomePage({super.key, required this.title});

  final String title;
  final CounterController controller = Get.put(CounterController());

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: Text(title),
      ),
      body: Center(
        child: Obx(
          () => Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
              const Text(
                'You have pushed the button this many times:',
              ),
              Text(
                controller.counter.toString(),
                style: Theme.of(context).textTheme.headlineMedium,
              ),
              ElevatedButton(
                onPressed: () {
                  Get.toNamed('/detail');
                },
                child: Text('Ke Halaman Detail'))
            ],
          ),
        ),
      ),
      floatingActionButton: Row(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: [
          FloatingActionButton(
            onPressed: controller.incrementCounter,
            tooltip: 'Increment',
            child: const Icon(Icons.add),
          ),
        ],
      ),
    );
  }
}

```

```

    ),
    FloatingActionButton(
      onPressed: controller.decrementCounter,
      tooltip: 'Decrement',
      child: const Icon(Icons.remove),
    ),
    FloatingActionButton(
      onPressed: controller.getSnackBar,
      tooltip: 'Get SnackBar',
      child: const Icon(Icons.message),
    ),
    FloatingActionButton(
      onPressed: controller.getbottomshet,
      tooltip: 'Get Bottom Sheet',
      child: const Icon(Icons.bolt_outlined),
    ),
  ],
),
);
}
}

```

Source Code detail\_page.dart:

```

import 'package:flutter/material.dart';

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Text('Detail Page'),
      ),
    );
  }
}

```

Source Code pubspec.yaml :

```

name: pertemuan_13
description: "A new Flutter project."

```

```
# The following line prevents the package from being accidentally
published to
# pub.dev using `flutter pub publish`. This is preferred for private
packages.
publish_to: 'none' # Remove this line if you wish to publish to pub.dev

# The following defines the version and build number for your
application.
# A version number is three numbers separated by dots, like 1.2.43
# followed by an optional build number separated by a +.
# Both the version and the builder number may be overridden in flutter
# build by specifying --build-name and --build-number, respectively.
# In Android, build-name is used as versionName while build-number
used as versionCode.
# Read more about Android versioning at
https://developer.android.com/studio/publish/versioning
# In iOS, build-name is used as CFBundleShortVersionString while
build-number is used as CFBundleVersion.
# Read more about iOS versioning at
#
https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html
# In Windows, build-name is used as the major, minor, and patch parts
# of the product and file versions while build-number is used as the build
suffix.
version: 1.0.0+1

environment:
  sdk: ^3.5.4

# Dependencies specify other packages that your package needs in order
to work.
# To automatically upgrade your package dependencies to the latest
versions
# consider running `flutter pub upgrade --major-versions`. Alternatively,
# dependencies can be manually updated by changing the version
numbers below to
# the latest version available on pub.dev. To see which dependencies have
newer
# versions available, run `flutter pub outdated`.
dependencies:
  flutter:
    sdk: flutter
```

**# The following adds the Cupertino Icons font to your application.  
# Use with the CupertinoIcons class for iOS style icons.  
cupertino\_icons: ^1.0.8  
get: ^4.6.6**

**dev\_dependencies:  
flutter\_test:  
 sdk: flutter**

**# The "flutter\_lints" package below contains a set of recommended lints  
to  
# encourage good coding practices. The lint set provided by the package  
is  
# activated in the `analysis\_options.yaml` file located at the root of your  
# package. See that file for information about deactivating specific lint  
# rules and activating additional ones.  
flutter\_lints: ^4.0.0**

**# For information on the generic Dart part of this file, see the  
# following page: <https://dart.dev/tools/pub/pubspec>**

**# The following section is specific to Flutter packages.  
flutter:**

**# The following line ensures that the Material Icons font is  
# included with your application, so that you can use the icons in  
# the material Icons class.  
uses-material-design: true**

**# To add assets to your application, add an assets section, like this:  
# assets:  
# - images/a\_dot\_burr.jpeg  
# - images/a\_dot\_ham.jpeg**

**# An image asset can refer to one or more resolution-specific "variants",  
see  
# <https://flutter.dev/to/resolution-aware-images>**

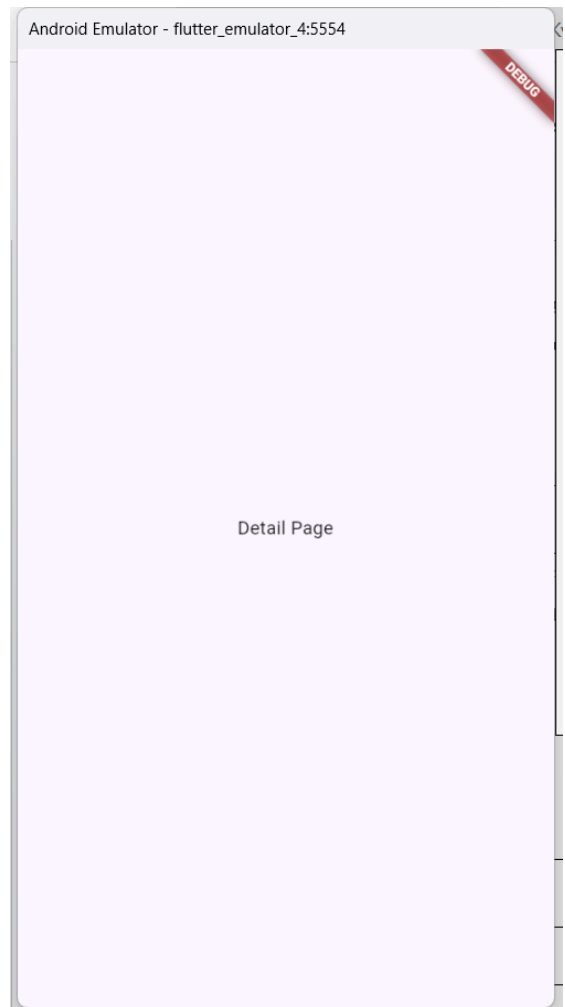
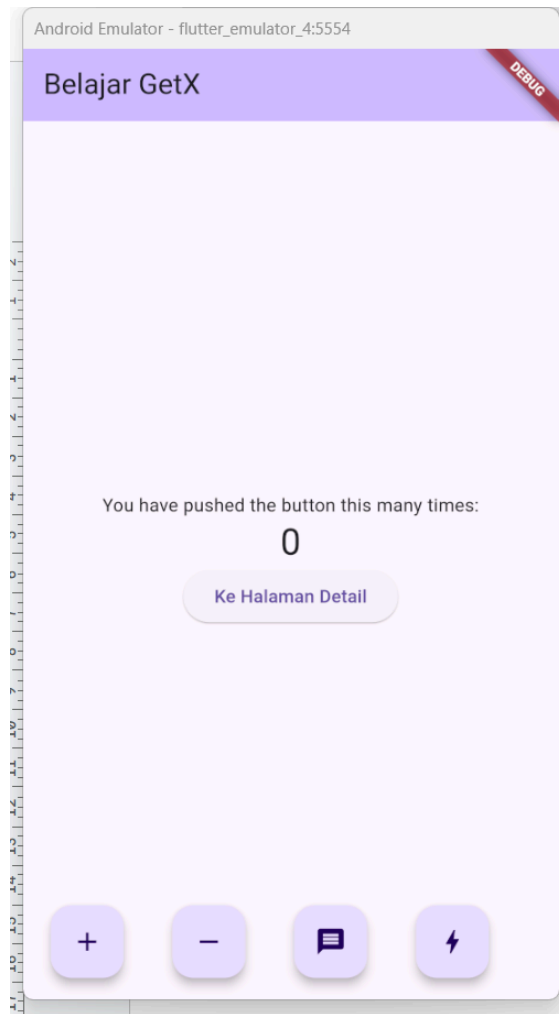
**# For details regarding adding assets from package dependencies, see  
# <https://flutter.dev/to/asset-from-package>**

**# To add custom fonts to your application, add a fonts section here,  
# in this "flutter" section. Each entry in this list should have a  
# "family" key with the font family name, and a "fonts" key with a  
# list giving the asset and other descriptors for the font. For**

```
# example:
# fonts:
#   - family: Schyler
#     fonts:
#       - asset: fonts/Schyler-Regular.ttf
#       - asset: fonts/Schyler-Italic.ttf
#         style: italic
#   - family: Trajan Pro
#     fonts:
#       - asset: fonts/TrajanPro.ttf
#       - asset: fonts/TrajanPro_Bold.ttf
#         weight: 700
#
# For details regarding fonts from package dependencies,
# see https://flutter.dev/to/font-from-package
```

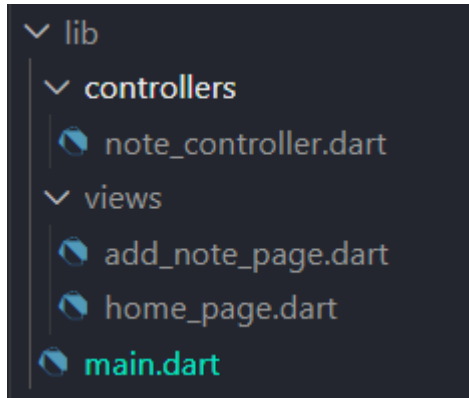


## Output GUIDED:



## UNGUIDED

### Struktur File:



### Source Code main.dart:

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return GetMaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Catatan Kekinian',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.indigo),
        useMaterial3: true,
        fontFamily: 'Roboto',
      ),
      home: HomePage(),
    );
  }
}
```

### Source Code add\_note\_page.dart:

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

class HomePage extends StatelessWidget {
  final NoteController noteController = Get.put(NoteController());

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          'Catatan SiJuna',
          style: TextStyle(fontWeight: FontWeight.bold, fontSize: 22),
        ),
        centerTitle: true,
        elevation: 4,
      ),
      body: Container(
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [Colors.indigo.shade300, Colors.indigo.shade900],
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
          ),
        ),
        child: Obx() {
          if (noteController.notes.isEmpty) {
            return Center(
              child: Text(
                'Belum ada catatan. Yuk, tambahkan!',
                style: TextStyle(
                  color: Colors.white,
                  fontSize: 18,
                  fontStyle: FontStyle.italic,
                ),
              ),
            );
          }
          return ListView.builder(
            itemCount: noteController.notes.length,
            padding: EdgeInsets.symmetric(horizontal: 16, vertical: 12),
            itemBuilder: (context, index) {
              final note = noteController.notes[index];
```

```

return Card(
  elevation: 5,
  shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(15),
  ),
  margin: EdgeInsets.symmetric(vertical: 8),
  child: ListTile(
    contentPadding:
      EdgeInsets.symmetric(vertical: 8, horizontal: 16),
    title: Text(
      note['title'] ?? 'Tidak ada judul',
      style: TextStyle(
        fontWeight: FontWeight.w600,
        fontSize: 18,
        color: Colors.indigo.shade800,
      ),
    ),
    subtitle: Text(
      note['description'] ?? 'Tidak ada deskripsi',
      style: TextStyle(
        fontSize: 14,
        color: Colors.indigo.shade600,
      ),
    ),
    trailing: IconButton(
      icon: Icon(Icons.delete, color: Colors.redAccent),
      onPressed: () {
        noteController.removeNoteAt(index);
      },
    ),
  ),
);
},
);
},
);
});
),
floatingActionButton: FloatingActionButton(
  onPressed: () {
    Get.to() => AddNotePage();
  },
  backgroundColor: Colors.indigo.shade700,
  child: Icon(Icons.add, color: Colors.white),
),
);
}

```

```
}
```

Source Code add\_note\_page.dart :

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

class AddNotePage extends StatelessWidget {
  final NoteController noteController = Get.find();
  final TextEditingController titleController = TextEditingController();
  final TextEditingController descriptionController =
    TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          'Tambah Catatan',
          style: TextStyle(fontWeight: FontWeight.bold),
        ),
        centerTitle: true,
      ),
      body: Padding(
        padding: const EdgeInsets.all(20.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text(
              'Judul Catatan',
              style: TextStyle(
                fontSize: 16,
                fontWeight: FontWeight.w500,
                color: Colors.indigo.shade700,
              ),
            ),
            SizedBox(height: 8),
            TextField(
              controller: titleController,
              decoration: InputDecoration(
                hintText: 'Masukkan judul...',
                border: OutlineInputBorder(
```

```
borderRadius: BorderRadius.circular(10),
),
),
),
SizedBox(height: 20),
Text(
'Deskripsi',
style: TextStyle(
fontSize: 16,
fontWeight: FontWeight.w500,
color: Colors.indigo.shade700,
),
),
SizedBox(height: 8),
TextField(
controller: descriptionController,
maxLines: 5,
decoration: InputDecoration(
hintText: 'Masukkan deskripsi...',
border: OutlineInputBorder(
borderRadius: BorderRadius.circular(10),
),
),
),
),
SizedBox(height: 20),
Center(
child: ElevatedButton(
onPressed: () {
final title = titleController.text;
final description = descriptionController.text;

if (title.isNotEmpty && description.isNotEmpty) {
noteController.addNote(title, description);
Get.back();
} else {
Get.snackbar(
'Error',
'Judul dan deskripsi tidak boleh kosong.',
snackPosition: SnackPosition.BOTTOM,
backgroundColor: Colors.redAccent,
colorText: Colors.white,
);
}
},
style: ElevatedButton.styleFrom(
```

```

        backgroundColor: Colors.indigo.shade700,
        padding: EdgeInsets.symmetric(horizontal: 40, vertical: 12),
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(10),
        ),
      ),
      child: Text(
        'Simpan Catatan',
        style: TextStyle(fontSize: 16, color: Colors.white),
      ),
    ),
  ],
),
);
}
}

```

Source Code note\_controller.dart :

```

import 'package:get/get.dart';

class NoteController extends GetxController {
  var notes = <Map<String, String>>[].obs;

  void addNote(String title, String description) {
    notes.add({'title': title, 'description': description});
  }

  void removeNoteAt(int index) {
    notes.removeAt(index);
  }
}

```

Source Code pubspec.yaml :

```

name: pertemuan_13
description: "A new Flutter project."
# The following line prevents the package from being accidentally
published to
# pub.dev using `flutter pub publish`. This is preferred for private

```

```
packages.  
publish_to: 'none' # Remove this line if you wish to publish to pub.dev  
  
# The following defines the version and build number for your  
application.  
# A version number is three numbers separated by dots, like 1.2.43  
# followed by an optional build number separated by a +.  
# Both the version and the builder number may be overridden in flutter  
# build by specifying --build-name and --build-number, respectively.  
# In Android, build-name is used as versionName while build-number  
used as versionCode.  
# Read more about Android versioning at  
https://developer.android.com/studio/publish/versioning  
# In iOS, build-name is used as CFBundleShortVersionString while  
build-number is used as CFBundleVersion.  
# Read more about iOS versioning at  
#  
https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html  
# In Windows, build-name is used as the major, minor, and patch parts  
# of the product and file versions while build-number is used as the build  
suffix.  
version: 1.0.0+1  
  
environment:  
  sdk: ^3.5.4  
  
# Dependencies specify other packages that your package needs in order  
to work.  
# To automatically upgrade your package dependencies to the latest  
versions  
# consider running `flutter pub upgrade --major-versions`. Alternatively,  
# dependencies can be manually updated by changing the version  
numbers below to  
# the latest version available on pub.dev. To see which dependencies have  
newer  
# versions available, run `flutter pub outdated`.  
dependencies:  
  flutter:  
    sdk: flutter  
  
# The following adds the Cupertino Icons font to your application.  
# Use with the CupertinoIcons class for iOS style icons.  
cupertino_icons: ^1.0.8
```



**get: ^4.6.6**

**dev\_dependencies:**

**flutter\_test:**

**sdk: flutter**

**# The "flutter\_lints" package below contains a set of recommended lints to**

**# encourage good coding practices. The lint set provided by the package is**

**# activated in the `analysis\_options.yaml` file located at the root of your  
# package. See that file for information about deactivating specific lint  
# rules and activating additional ones.**

**flutter\_lints: ^4.0.0**

**# For information on the generic Dart part of this file, see the  
# following page: <https://dart.dev/tools/pub/pubspec>**

**# The following section is specific to Flutter packages.  
flutter:**

**# The following line ensures that the Material Icons font is  
# included with your application, so that you can use the icons in  
# the material Icons class.  
uses-material-design: true**

**# To add assets to your application, add an assets section, like this:**

**# assets:**

**# - images/a\_dot\_burr.jpeg**

**# - images/a\_dot\_ham.jpeg**

**# An image asset can refer to one or more resolution-specific "variants",  
see**

**# <https://flutter.dev/to/resolution-aware-images>**

**# For details regarding adding assets from package dependencies, see  
# <https://flutter.dev/to/asset-from-package>**

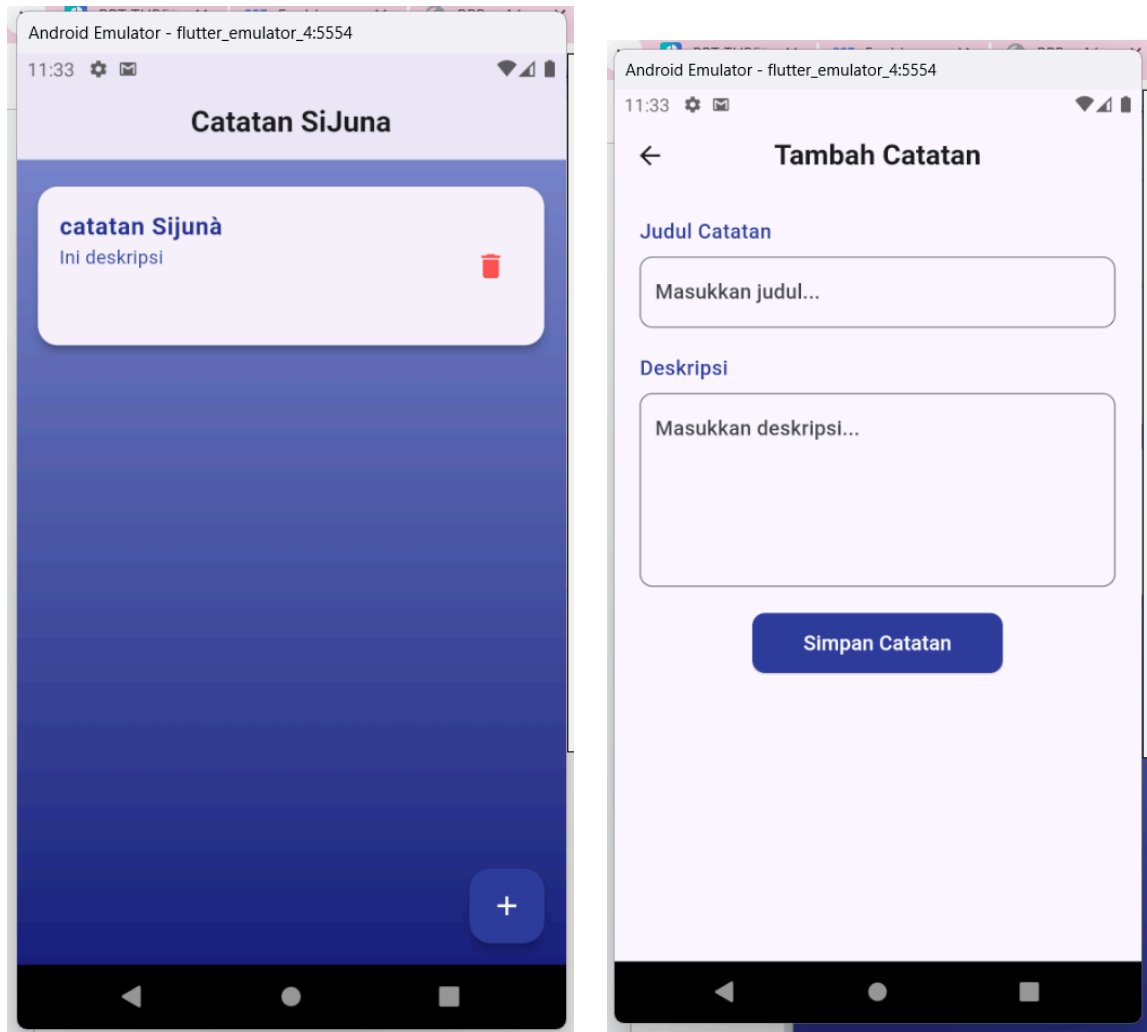
**# To add custom fonts to your application, add a fonts section here,  
# in this "flutter" section. Each entry in this list should have a  
# "family" key with the font family name, and a "fonts" key with a  
# list giving the asset and other descriptors for the font. For  
# example:**

**# fonts:**

**# - family: Schyler**

```
# fonts:
#   - asset: fonts/Schyler-Regular.ttf
#   - asset: fonts/Schyler-Italic.ttf
#     style: italic
# - family: Trajan Pro
#   fonts:
#     - asset: fonts/TrajanPro.ttf
#     - asset: fonts/TrajanPro_Bold.ttf
#       weight: 700
#
# For details regarding fonts from package dependencies,
# see https://flutter.dev/to/font-from-package
```

## Output Programs **UNGUIDED**



## Deskripsi

Aplikasi Catatan SiJuna adalah sebuah aplikasi manajemen catatan berbasis Flutter, yang bertujuan untuk membantu pengguna dalam menyimpan, mengelola, dan menghapus catatan secara mudah. Aplikasi ini menggunakan framework Flutter untuk pengembangan front-end dan pustaka GetX untuk pengelolaan state dan navigasi yang efisien.

## Fitur Utama Aplikasi

1. **Tampilan Daftar Catatan:**
  - Menampilkan daftar catatan yang telah dibuat.
  - Menyediakan informasi seperti judul dan deskripsi catatan.
  - Tampilan dengan gradien warna indigo yang estetik.
2. **Penambahan Catatan:**

- Pengguna dapat menambahkan catatan baru melalui halaman *Add Note*.
  - Formulir dengan validasi memastikan bahwa judul dan deskripsi tidak kosong.
  - Catatan akan langsung ditambahkan ke daftar setelah disimpan.
3. **Penghapusan Catatan:**
    - Setiap catatan memiliki ikon hapus (*delete*) untuk menghapus catatan tertentu.
    - Data akan diperbarui secara langsung tanpa memuat ulang aplikasi.
  4. **Navigasi yang Mudah:**
    - Menggunakan **GetX** untuk transisi halaman, yang memberikan pengalaman navigasi cepat dan bebas bug.
  5. **Pemberitahuan dan Validasi:**
    - Memberikan notifikasi jika pengguna mencoba menyimpan catatan tanpa memasukkan judul atau deskripsi.

### Struktur File Aplikasi

1. **main.dart**
    - Berfungsi sebagai *entry point* aplikasi dan menginisialisasi tema.
    - Menampilkan halaman utama (*HomePage*) sebagai *home screen*.
  2. **home\_page.dart**
    - Menyediakan daftar catatan yang dapat dikelola pengguna.
    - Menggunakan **Obx** untuk *real-time state management* dari daftar catatan.
  3. **add\_note\_page.dart**
    - Halaman untuk menambahkan catatan baru dengan input formulir.
    - Validasi input dan tombol simpan tersedia di halaman ini.
  4. **note\_controller.dart**
    - Berisi logika pengelolaan catatan menggunakan **GetX Controller**.
    - Mengelola penambahan dan penghapusan catatan dalam bentuk daftar observasi (*RxList*).
  5. **pubspec.yaml**
    - Mendefinisikan konfigurasi proyek, seperti nama aplikasi, versi, dan dependensi (misalnya, **GetX**).
- 

### Cara Kerja Aplikasi

1. Saat aplikasi dibuka, pengguna diarahkan ke halaman utama (*HomePage*).
2. Jika belum ada catatan, akan muncul pesan "Belum ada catatan. Yuk, tambahkan!".
3. Pengguna dapat menambahkan catatan dengan mengetuk tombol tambah (*Floating Action Button*), yang akan membuka halaman *Add Note*.
4. Di halaman *Add Note*, pengguna dapat mengisi judul dan deskripsi, lalu menyimpannya.
5. Catatan baru akan langsung ditambahkan ke daftar di halaman utama.
6. Pengguna dapat menghapus catatan tertentu dengan mengetuk ikon hapus di setiap item daftar.

