**LAPORAN WORKSHOP PEMROGRAMAN**

**PERANGKAT BERGERAK**

**WEEK 6**

**Notes App**

Dosen Pengampu : Prasetyo Wibowo S.ST., M.Kom



Disusun oleh :

Nama : Muhammad Daffa Erfiansyah

NRP : 3123500006

Kelas : 2 D3 IT A

# Program Studi Teknik Informatika

# Departemen Teknik

# Informatika dan Komputer

# Politeknik Elektronika Negeri Surabaya

**2024/2025**

1. **Dasar Teori**

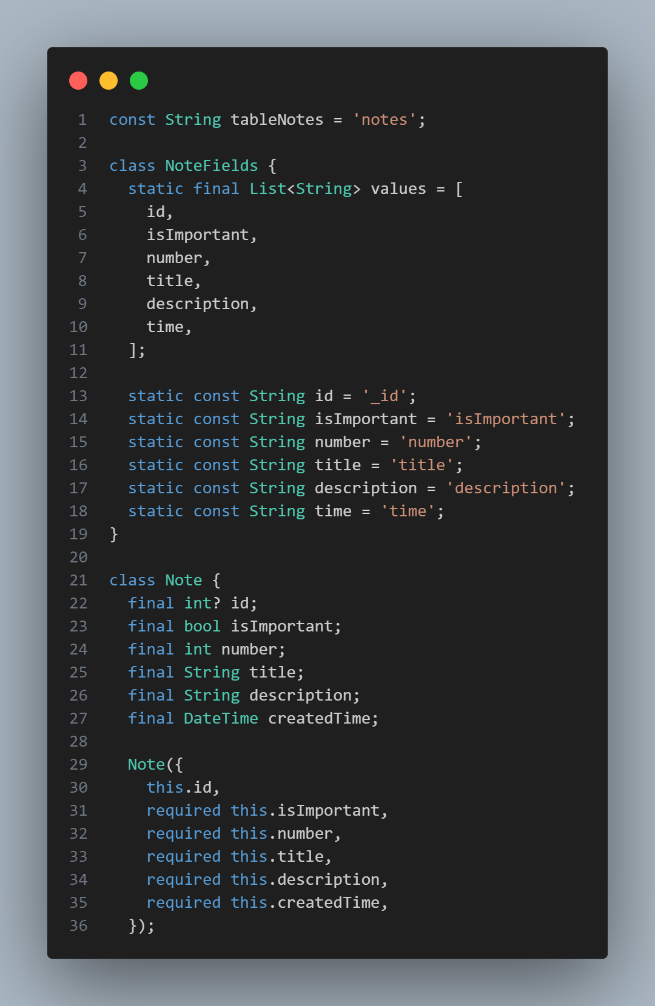
Flutter menyediakan berbagai widget untuk membangun tampilan antarmuka yang fleksibel dan responsif. Layout dalam Flutter didasarkan pada widget seperti Container, Row, Column, Stack, dan ListView, yang memungkinkan pengaturan elemen secara vertikal, horizontal, atau bertumpuk. Widget Expanded dan Flexible digunakan untuk mengatur proporsi ukuran elemen dalam suatu tata letak. Selain itu, penggunaan Padding, Margin, dan Align membantu dalam mengatur posisi serta jarak antar elemen agar tampilan lebih rapi dan estetis. Flutter menggunakan sistem rendering berbasis widget yang memungkinkan pembaruan UI secara efisien. Dengan konsep widget tree, setiap elemen UI dalam Flutter direpresentasikan sebagai widget yang dapat bersarang dan diperbarui secara dinamis. Penggunaan MediaQuery dan LayoutBuilder membantu dalam membuat tampilan yang adaptif terhadap berbagai ukuran layar. Oleh karena itu, pemahaman tentang widget dan cara mengaturnya dalam layout sangat penting dalam membangun aplikasi Flutter yang responsif dan menarik.

1. **Pratikum**
2. Inisialisasi Database





1. Models Database



Analisa:  
Class NoteDatabase menerapkan pola singleton dengan lazy initialization untuk memastikan satu instance database di seluruh aplikasi. Database diinisialisasi dengan \_initDB untuk penyimpanan dan \_createDB untuk skema. Metode CRUD seperti create, getNoteById, getAllNotes, updateNote, dan deleteNoteById tersedia, serta metode tambahan close dan getAllNotesSortedByImportance. Model Note merepresentasikan catatan dengan field seperti id, isImportant, number, title, description, dan time. Model ini bersifat immutable dengan constructor wajib, serta menyediakan metode copy, fromJson, dan toJson untuk manipulasi data.

1. Note Page



Analisa:

Class NotePage menampilkan daftar catatan dalam grid dinamis menggunakan MasonryGridView. Fitur yang tersedia mencakup pencarian, animasi transisi, indikator loading, tampilan kosong saat tidak ada catatan, serta tombol tambah catatan. State dikelola dengan baik menggunakan AnimationController, dengan pembersihan resource di dispose(). Class NoteCardWidgets menampilkan setiap catatan dalam kartu dengan judul, deskripsi, waktu, dan indikator kepentingan. Tampilan diperindah dengan warna latar dari \_lightColors, variasi tinggi kartu melalui getMinHeight(), serta format tanggal yang mudah dibaca menggunakan DateFormat.

1. Add & Edit Page







Form.dart





Analisa:  
Class AddEditNotePage berfungsi sebagai halaman formulir untuk menambah atau mengedit catatan, dengan parameter note yang opsional. State dikelola untuk field catatan menggunakan Form dan GlobalKey untuk validasi. Metode addNote dan updateNote dipisahkan untuk menangani penyimpanan dan pembaruan catatan, serta Snackbar digunakan untuk menampilkan notifikasi keberhasilan atau kegagalan. NoteFormWidget memisahkan UI formulir dari logika dengan menerima nilai dan callback. Formulir mencakup switch untuk menandai catatan penting, slider prioritas (0-5), serta field judul dan deskripsi. Desainnya diperhatikan dengan Card, border radius, padding yang konsisten, serta fokus border ungu saat field aktif. Validasi memastikan judul dan deskripsi tidak kosong.

1. DetailPage

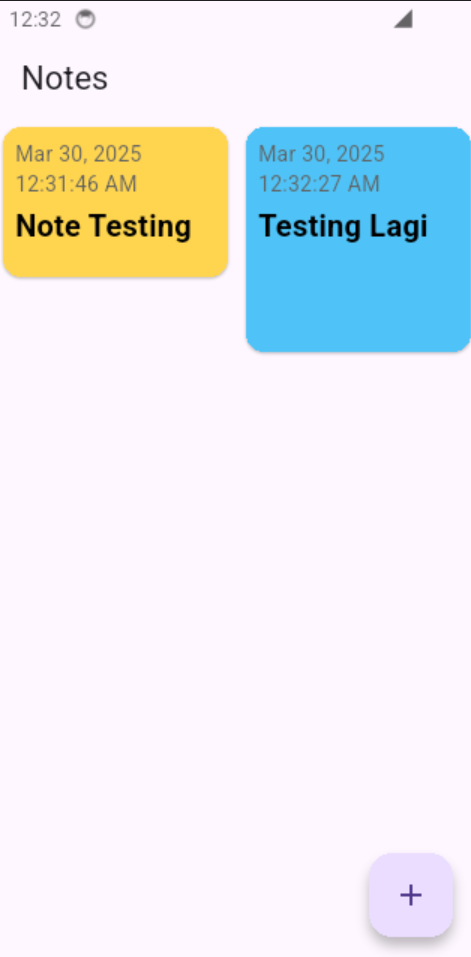




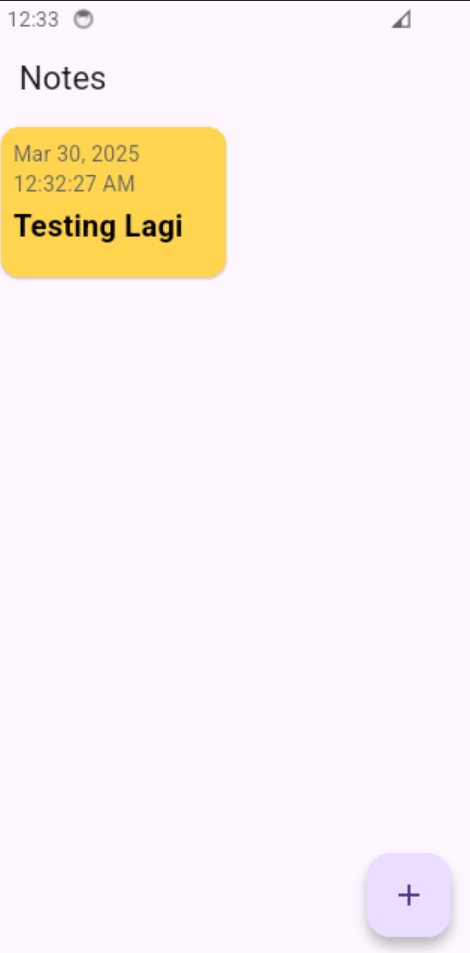


Analisa :

Kelas NoteDetailPage dirancang untuk menampilkan informasi lengkap dari sebuah catatan yang dipilih berdasarkan ID. Halaman ini menggunakan StatefulWidget karena perlu memuat data catatan secara asinkron dari database. Metode refreshNote() mengambil data catatan berdasarkan ID dan mengelola state loading dengan baik.

1. Output :







1. Kesimpulan

Dalam praktikum ini untuk menampilkan informasi tempat wisata dapat dilakukan dengan pendekatan yang responsif dan estetis. Dengan memanfaatkan ListView, Card, dan Stack, aplikasi mampu menyajikan daftar tempat wisata serta detail informasi dengan tampilan yang menarik dan mudah diakses oleh pengguna. Selain itu, penggunaan Navigator.push memungkinkan navigasi antar halaman dengan lancar, sementara elemen visual seperti ClipRRect, BoxShadow, dan LinearGradient memberikan tampilan yang lebih profesional. Praktikum ini juga menunjukkan pentingnya pengorganisasian kode dengan metode terpisah, seperti \_buildListItem dan \_buildInfoCard, untuk meningkatkan keterbacaan dan pemeliharaan kode.