

Laporan
Pemograman Mobile

**PENGEMBANGAN APLIKASI MOBILE "BOOKSAPP" BERBASIS
FLUTTER**



Nama : Melvin Pawaka Tio

Npm : 231510003

Dosen : Saut Pintubipar Saragih, S.Kom., M.MSI

PROGRAM STUDI
SISTEM INFORMASI
FAKULTAS TEKNIK DAN KOMPUTER
UNIVERSITAS PUTERA BATAM

2026/2027

Kata Pengantar

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas selesainya laporan teknis ini. Laporan ini disusun sebagai dokumentasi pengembangan aplikasi "BooksApp" yang menerapkan arsitektur modular, manajemen tema, dan integrasi data menggunakan Flutter. Penulis berterima kasih kepada semua pihak yang telah membantu hingga tugas ini selesai.

Batam, 25 Desember 2025

Penyusun

Melvin Pawaka Tio

Daftar Isi

Daftar Isi	3
Daftar Tabel	4
BAB I	5
PENDAHULUAN	5
I.A. Latar Belakang	5
I.B. Rumusan Masalah	5
I.C. Tujuan.....	6
BAB II.....	7
PEMBAHASAN	7
II.A. HASIL	7
II.A.1. Workflow Understanding.....	7
II.A.2. Environment Setup	7
II.A.3. Set Up Project & File Structure.....	8
II.A.4. Application Core	10
II.A.5. Application Features.....	10
II.A.6. Presentation Layer	10
II.A.7. Application Entry Point.....	10
II.B. PEMBAHASAN	11
BAB III	12
PENUTUP	12
III.A. KESIMPULAN	12
III.B. SARAN.....	12

Daftar Tabel

BAB I

PENDAHULUAN

I.A. Latar Belakang

Dalam pengembangan perangkat lunak *mobile* modern, arsitektur yang bersih dan modular sangat penting untuk menjamin kemudahan pemeliharaan dan pengembangan fitur di masa depan. Laporan ini membahas proses teknis pengembangan "BooksApp", sebuah aplikasi katalog buku yang dibangun menggunakan kerangka kerja (*framework*) Flutter.

Pengembangan aplikasi ini difokuskan pada pengorganisasian kode yang terstruktur, di mana logika antarmuka (*User Interface*), manajemen tema visual, dan pengelolaan data dipisahkan ke dalam modul-modul spesifik. Pendekatan ini diambil untuk mensimulasikan lingkungan pengembangan aplikasi skala industri yang mengutamakan skalabilitas dan keterbacaan kode.

I.B. Rumusan Masalah

Berdasarkan kebutuhan teknis sistem, rumusan masalah dalam pengembangan ini adalah:

1. Bagaimana menyusun struktur direktori proyek Flutter yang memisahkan logika fitur ke dalam modul terpisah?
2. Bagaimana mengimplementasikan manajemen tema (*theme manager*) agar aplikasi mendukung mode tampilan yang dinamis?
3. Bagaimana mengelola status data (*state management*), khususnya untuk fitur favorit dalam aplikasi?
4. Bagaimana mengintegrasikan navigasi antara halaman daftar buku dan halaman detail buku secara efisien?

I.C. Tujuan

Tujuan dari pengembangan aplikasi ini adalah:

1. Menghasilkan aplikasi *mobile* BooksApp yang fungsional dan terintegrasi penuh.
2. Menerapkan struktur kode modular untuk manajemen tema, data favorit, dan tampilan detail.
3. Menyediakan pengalaman pengguna yang responsif melalui navigasi halaman yang mulus.

BAB II

PEMBAHASAN

II.A. HASIL

II.A.1. Workflow Understanding

Alur kerja sistem BooksApp dirancang sebagai berikut:

1. **Inisialisasi Sistem:** Aplikasi dijalankan melalui titik masuk utama, yang memuat konfigurasi global seperti tema dan penyedia data (*data provider*).
2. **Tampilan Daftar:** Pengguna disajikan antarmuka daftar buku yang diambil dari modul data.
3. **Personalisasi:** Pengguna dapat mengubah preferensi visual (tema) yang dikelola oleh manajer tema.
4. **Interaksi Item:** Pengguna dapat menandai buku sebagai favorit atau memilih buku untuk melihat rinciannya.
5. **Navigasi Detail:** Sistem mengarahkan pengguna ke halaman detail yang memuat informasi spesifik buku terkait.

II.A.2. Environment Setup

Lingkungan pengembangan yang digunakan meliputi:

- **Framework:** Flutter SDK.
- **Bahasa:** Dart.
- **Editor:** Visual Studio Code.
- **Platform Target:** Android/iOS.

II.A.3. Set Up Project & File Structure

Struktur direktori proyek disusun secara hirarkis untuk memisahkan tanggung jawab setiap berkas. Berikut adalah struktur pohon (*tree structure*) dari folder lib aplikasi:

Lib

```
|  main.dart
|
|-----chapter09
|    main_09.dart
|
|-----chapter10
|    theme_manager.dart
|
|-----chapter11
|    favorites_manager.dart
|
|-----chapter15
|    book_details.dart
|    main_15.dart
```

Penjelasan Fungsionalitas Modul:

1. **lib/main.dart**: Merupakan titik masuk utama (*entry point*) aplikasi. Berkas ini berfungsi menginisialisasi MaterialApp dan menghubungkan seluruh komponen modul agar dapat berjalan sebagai satu sistem yang utuh.
2. **lib/chapter09/main_09.dart (Modul UI Dasar)**: Berkas ini berisi implementasi dasar antarmuka pengguna (*User Interface*). Fokus utamanya adalah menyusun kerangka tampilan daftar buku (*Listing*) menggunakan komponen *widget* standar Flutter.
3. **lib/chapter10/theme_manager.dart (Modul Manajemen Tema)**: Berkas ini menangani logika *theming*. Di dalamnya terdapat definisi warna, tipografi, dan mekanisme untuk mengganti tampilan aplikasi antara mode terang (*Light Mode*) dan gelap (*Dark Mode*).
4. **lib/chapter11/favorites_manager.dart (Modul Data & State)**: Bertanggung jawab atas pengelolaan data lokal, khususnya fitur "Favorit". Berkas ini memastikan bahwa buku yang ditandai oleh pengguna disimpan statusnya dalam memori aplikasi.
5. **lib/chapter15/ (Modul Integrasi & Detail)**:
 - **book_details.dart**: Menangani tampilan halaman kedua (*Second Page*). Halaman ini dirancang untuk menerima data buku yang dikirim dari halaman utama dan menampilkannya secara rinci (Judul, Deskripsi, Penulis).
 - **main_15.dart**: Berkas logika integrasi yang menyempurnakan navigasi. Berkas ini menghubungkan daftar buku dengan halaman detail, memungkinkan perpindahan data (*passing data*) antar layar.

II.A.4. Application Core

Inti aplikasi dibangun dengan konsep *Widget Tree* yang saling terhubung. `main.dart` bertindak sebagai *root*, yang kemudian memanggil logika tampilan dari folder `chapter15` (sebagai versi final UI), sambil menyuntikkan ketergantungan (*dependency injection*) dari `theme_manager.dart` dan `favorites_manager.dart` ke seluruh aplikasi.

II.A.5. Application Features

Fitur-fitur yang berhasil diimplementasikan dalam aplikasi ini meliputi:

1. **Katalog Buku:** Penampilan data buku dalam format senarai (*list*) yang dapat digulir.
2. **Detail View:** Halaman informasi mendalam untuk setiap item buku.
3. **Dynamic Theming:** Kemampuan mengubah nuansa warna aplikasi secara dinamis.
4. **Favorites System:** Logika untuk memisahkan buku pilihan pengguna.

II.A.6. Presentation Layer

- **Halaman Utama:** Menggunakan `ListView.builder` untuk merender daftar buku secara efisien. Setiap elemen daftar direpresentasikan oleh *widget* kartu yang informatif.
- **Halaman Detail:** Menggunakan `SingleChildScrollView` untuk mengakomodasi konten teks yang panjang pada deskripsi buku, serta `AppBar` yang menyediakan tombol kembali (*back button*) secara otomatis.

II.A.7. Application Entry Point

Fungsi `main()` pada `main.dart` menjalankan `runApp()`, yang membungkus aplikasi dalam *widget* utama. Di sinilah konfigurasi rute navigasi dan penyedia tema ditetapkan sebelum antarmuka visual dirender ke layar pengguna.

II.B. PEMBAHASAN

Analisis Hasil Implementasi Struktur berkas yang diterapkan menunjukkan pemisahan yang jelas antara logika tampilan (*Presentation*) dan logika bisnis (*Business Logic*). Penggunaan `theme_manager.dart` dan `favorites_manager.dart` secara terpisah memungkinkan pengembang untuk memperbarui fitur tersebut tanpa mengganggu kode antarmuka di `book_details.dart` atau `main_15.dart`.

Integrasi Antar Fitur Sistem navigasi berhasil menghubungkan daftar buku dengan halaman detail. Data objek buku dikirimkan melalui konstruktor kelas `BookDetails`, sehingga informasi yang muncul di halaman detail selalu sinkron dengan buku yang dipilih di halaman utama.

Kelebihan dan Keterbatasan

- **Kelebihan:** Aplikasi memiliki struktur yang sangat modular dan mudah dipahami. Pemisahan fitur ke dalam folder spesifik memudahkan proses *debugging*.
- **Keterbatasan:** Saat ini data buku masih bersifat lokal atau statis dalam implementasi dasar. Pengembangan lanjut dapat dilakukan dengan menghubungkan modul ini ke API eksternal secara penuh.

BAB III

PENUTUP

III.A. KESIMPULAN

Pengembangan aplikasi "BooksApp" telah berhasil diselesaikan dengan baik. Aplikasi ini mendemonstrasikan kemampuan Flutter dalam menangani struktur proyek yang kompleks melalui pembagian modul yang sistematis. Integrasi antara manajemen tema, pengelolaan data favorit, dan navigasi halaman detail telah berjalan lancar membentuk satu kesatuan aplikasi yang utuh.

III.B. SARAN

Disarankan untuk pengembangan selanjutnya agar menambahkan fitur pencarian (*Search*) yang lebih kompleks dan menerapkan penyimpanan persisten (seperti SQLite) pada `favorites_manager.dart` agar data favorit tidak hilang saat aplikasi ditutup sepenuhnya.