Laporan Pemrograman Berbasis Mobile B

News Portal App

disusun untuk memenuhi tugas

mata kuliah Pemrograman Berbasis Mobile B

oleh :

Farah Nasywa

(2208107010051)

JURUSAN INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS SYIAH KUALA

DARUSSALAM, BANDA ACEH

2025

1. TUJUAN TUGAS

Setelah melakukan Tugas ini, mahasiswa diharapkan dapat:

- Memahami konsep dasar Flutter framework untuk pengembangan aplikasi mobile
- Mengimplementasikan aplikasi berita dengan menggunakan API eksternal
- Menerapkan sistem autentikasi login dan registrasi dalam aplikasi Flutter
- Menggunakan state management dan navigation dalam Flutter
- Mengintegrasikan package-package eksternal seperti HTTP client dan shared preferences
- Membuat aplikasi dengan multiple screens dan navigation yang baik

2. DASAR TEORI

2.1 Flutter Framework

Flutter adalah UI toolkit revolusioner dari Google yang digunakan untuk membangun aplikasi yang dikompilasi secara native untuk mobile, web, dan desktop dari satu basis kode tunggal (single codebase). Flutter menggunakan bahasa pemrograman Dart yang juga dikembangkan oleh Google. Yang membuat Flutter unik adalah pendekatan "compile-to-native" yang memungkinkan aplikasi Flutter memiliki performance yang hampir sama dengan aplikasi native.

Arsitektur Flutter terdiri dari beberapa layer utama: Framework Layer (yang berisi widget library, rendering library, dan foundation library), Engine Layer (yang berisi Skia graphics engine, Dart runtime, dan text rendering), dan Embedder Layer (yang berisi platform-specific code). Flutter menggunakan Skia graphics engine yang sama dengan yang digunakan oleh Google Chrome, sehingga memberikan konsistensi visual across platforms.

Konsep fundamental dalam Flutter adalah "everything is a widget". Widget adalah building block utama dalam Flutter, mulai dari elemen UI sederhana seperti text dan button, hingga layout dan styling. Ada dua jenis widget utama: StatelessWidget (untuk UI yang tidak berubah) dan StatefulWidget (untuk UI yang dapat berubah berdasarkan state). Widget tree adalah representasi hierarkis dari semua widget dalam aplikasi, dan Flutter menggunakan reactive programming model dimana UI secara otomatis update ketika state berubah.

Hot reload adalah salah satu fitur paling powerful dalam Flutter yang memungkinkan developer untuk melihat perubahan kode secara real-time tanpa kehilangan application state. Fitur ini sangat mempercepat development cycle karena developer tidak perlu menunggu full compilation dan restart aplikasi setiap kali melakukan perubahan kode.

2.2 HTTP Request dan REST API

HTTP (HyperText Transfer Protocol) adalah protokol komunikasi fundamental yang digunakan untuk mentransfer data antara client dan server di internet. Dalam konteks aplikasi mobile, HTTP requests digunakan untuk berkomunikasi dengan web services dan APIs untuk mengambil atau mengirim data. HTTP menggunakan berbagai methods seperti GET (untuk mengambil data), POST (untuk mengirim data baru), PUT (untuk update data), dan DELETE (untuk menghapus data).

REST API (Representational State Transfer) adalah arsitektur yang menggunakan HTTP methods untuk mengakses dan memanipulasi data. REST API bersifat stateless, meaning setiap request harus berisi semua informasi yang diperlukan untuk memproses request tersebut. REST API biasanya menggunakan format JSON untuk data exchange karena JSON lightweight dan mudah di-parse oleh berbagai programming languages.

Dalam Flutter, HTTP requests dilakukan menggunakan http package yang menyediakan functions seperti http.get(), http.post(), dll. Response dari API biasanya berupa JSON yang kemudian di-parse menggunakan dart:convert library. Error handling sangat penting dalam HTTP requests karena network issues, server errors, atau invalid responses dapat terjadi kapan saja.

Untuk aplikasi News Portal ini, NewsAPI.org digunakan sebagai sumber data berita. NewsAPI menyediakan endpoints untuk mengakses berita dari berbagai sumber dengan format JSON yang konsisten. API key diperlukan untuk authentication dan rate limiting.

2.3 State Management

State management adalah konsep fundamental dalam Flutter yang mengatur bagaimana data (state) disimpan, dimodifikasi, dan dibagikan antar widgets dalam aplikasi. State dapat berupa user input, API response, navigation state, atau data lainnya yang dapat berubah selama aplikasi berjalan. Proper state management sangat penting untuk memastikan UI selalu menampilkan data yang akurat dan up-to-date.

Flutter menyediakan beberapa approached untuk state management, mulai dari yang sederhana seperti setState() untuk local state management, hingga yang kompleks seperti Provider, Bloc, Riverpod, atau GetX untuk global state management. setState() adalah cara paling basic untuk update state dalam StatefulWidget, dimana calling setState() akan trigger rebuild widget dan semua child widgets nya.

Untuk aplikasi yang lebih kompleks dengan multiple screens dan shared state, dibutuhkan state management solution yang lebih sophisticated. Provider pattern adalah salah satu yang paling popular karena recommended oleh Flutter team dan relatively easy to learn. Provider menggunakan InheritedWidget untuk efficiently propagate data down the widget tree.

Local state vs global state adalah konsep penting dalam state management. Local state adalah state yang hanya diperlukan oleh satu widget atau screen, sementara global state adalah state

yang perlu diakses oleh multiple widgets atau screens. User authentication status, theme preferences, dan shopping cart contents adalah contoh global state yang umum.

2.4 Navigation dan Routing

Navigation adalah proses berpindah antar halaman (screens/routes) dalam aplikasi mobile. Flutter menggunakan Navigator class untuk mengelola route stack menggunakan konsep stack data structure (LIFO - Last In First Out). Setiap kali user navigate ke screen baru, screen tersebut di-push ke stack. Ketika user kembali ke screen sebelumnya, screen current di-pop dari stack.

Navigator.push() digunakan untuk navigate ke screen baru, sementara Navigator.pop() digunakan untuk kembali ke screen sebelumnya. Navigator juga menyediakan methods lain seperti pushReplacement() (untuk replace current screen), pushAndRemoveUntil() (untuk navigate dan remove multiple screens dari stack), dan popUntil() (untuk pop multiple screens).

Named routes adalah cara yang lebih organized untuk mengelola navigation dalam aplikasi besar. Dengan named routes, setiap screen diberi nama unique, dan navigation dilakukan menggunakan route name instead of widget instance. Ini membuat code lebih maintainable dan memungkinkan features seperti deep linking.

Route parameters dan return values juga penting dalam navigation. Seringkali kita perlu pass data dari satu screen ke screen lain, atau mendapatkan result dari screen yang di-navigate. Flutter menyediakan mechanisms untuk handle kedua scenarios ini dengan elegant way.

Bottom navigation dan drawer navigation adalah pola navigation yang umum dalam aplikasi mobile. Bottom navigation biasanya digunakan untuk primary navigation antar main sections aplikasi, sementara drawer navigation digunakan untuk secondary navigation atau settings.

2.5 Shared Preferences

Shared Preferences adalah cara standard untuk menyimpan data sederhana secara persisten di dalam aplikasi mobile. Data yang disimpan dalam shared preferences akan tetap ada bahkan setelah aplikasi ditutup atau device di-restart. Shared preferences ideal untuk menyimpan user preferences, settings, authentication tokens, atau small pieces of data lainnya.

Dalam Flutter, shared_preferences package menyediakan interface yang clean untuk mengakses platform-specific persistent storage (NSUserDefaults di iOS dan SharedPreferences di Android). Package ini mendukung berbagai data types seperti bool, int, double, String, dan List<String>.

SharedPreferences bekerja secara asynchronous, sehingga semua operations (get, set, remove) return Future. Ini penting untuk UI responsiveness karena disk I/O operations tidak akan block main UI thread. Best practice adalah selalu use await atau .then() ketika accessing SharedPreferences.

Untuk security-sensitive data seperti authentication tokens, sebaiknya menggunakan flutter_secure_storage instead of SharedPreferences karena flutter_secure_storage encrypt data sebelum menyimpannya ke persistent storage. Namun untuk simple app preferences atau non-sensitive data, SharedPreferences sudah sufficient.

Data yang disimpan dalam SharedPreferences sebaiknya di-cache dalam memory untuk performance optimization, terutama jika data tersebut frequently accessed. Ini menghindari repeated disk I/O operations yang dapat mempengaruhi app performance.

3. ALAT DAN BAHAN

3.1 Software

- Flutter SDK (versi terbaru)
- Android Studio
- Android Emulator / Device fisik
- Git untuk version control
- NewsAPI.org account untuk API key

3.2 Hardware

- Laptop/PC dengan RAM minimal 8GB
- Storage minimal 10GB untuk Flutter SDK dan emulator
- Smartphone Android (opsional untuk testing)

3.3 API dan Services

- NewsAPI.org untuk sumber berita TechCrunch
- API sederhana untuk sistem login/registrasi

4. LANGKAH KERJA

4.1 Setup Environment

Install Flutter SDK adalah langkah pertama yang critical untuk Flutter development. Download Flutter SDK dari official website dan extract ke directory yang appropriate (avoid spaces dalam path). Add Flutter bin directory ke system PATH environment variable sehingga flutter commands dapat dijalankan dari anywhere dalam command line. Verify installation dengan running `flutter --version` untuk ensure Flutter properly installed.

Setup Android Studio dengan Flutter plugin requires downloading dan installing Android Studio, kemudian installing Flutter dan Dart plugins dari plugin marketplace. Configure Android SDK location dalam Android Studio settings dan accept license agreements untuk

various SDK components. Setup at least one Android Virtual Device (AVD) dengan recent Android version untuk testing purposes.

Konfigurasi Android emulator involves creating AVD dengan appropriate specifications (RAM, storage, screen resolution) yang representative dari target devices. Enable hardware acceleration (Intel HAXM atau AMD Hypervisor) untuk better emulator performance. Test emulator startup dan ensure dapat connect dengan adb (Android Debug Bridge).

Verifikasi instalasi dengan flutter doctor adalah diagnostic command yang checks untuk common issues dalam Flutter setup. Running `flutter doctor` akan show status dari Flutter installation, connected devices, IDE setup, dan any issues yang perlu di-resolve. Address any issues yang reported oleh flutter doctor sebelum proceeding dengan development.

Daftar akun di NewsAPI.org untuk mendapatkan API key requires creating free account dan obtaining API key untuk accessing news data. API key harus di-store securely dan tidak di-hardcode dalam source code. Consider using environment variables atau configuration files yang tidak di-commit ke version control untuk API key management.

4.2 Inisialisasi Project

Clone repository dari GitHub menggunakan command `git clone https://github.com/farahNasywa/Flutter_FarahNasywa_2208107010051.git` akan download entire project dengan semua files dan commit history. Ensure git is properly installed dan configured dengan user name dan email sebelum cloning. Navigate ke project directory setelah cloning completed.

Change directory ke project folder dengan `cd Flutter_FarahNasywa_2208107010051` untuk ensure subsequent commands dijalankan dalam correct context. Verify bahwa current directory contains pubspec.yaml file yang indicates Flutter project root.

Install dependencies dengan running `flutter pub get` akan download dan install semua packages yang specified dalam pubspec.yaml file. Command ini akan create .packages file dan .dart_tool directory yang contains package metadata. Check untuk any errors during dependency resolution dan resolve version conflicts jika ada.

Jalankan aplikasi dengan `flutter run` akan compile aplikasi dan deploy ke connected device atau emulator. Ensure device atau emulator is running dan recognized oleh Flutter dengan checking `flutter devices`. First run akan take longer karena initial compilation, tapi subsequent runs akan faster dengan hot reload capability.

Monitor console output untuk any compilation errors, warnings, atau runtime issues. Hot reload dapat di-trigger dengan pressing 'r' dalam console atau saving files dalam IDE dengan hot reload enabled.

4.3 Implementasi Fitur

Membuat sistem Login & Registrasi melibatkan beberapa components yang complex. **Implementasi form login dan registrasi** requires creating TextFormField widgets dengan proper validation untuk email format, password strength, dan required fields. Form validation harus user-friendly dengan clear error messages dan real-time feedback. TextEditingController digunakan untuk managing input state dan accessing user input values.

Integrasi dengan API untuk autentikasi involves creating HTTP requests ke authentication endpoints dengan user credentials. Response handling harus cover success scenarios (dengan token return) dan error scenarios (invalid credentials, network errors, server errors). Proper error messages harus ditampilkan ke user based on API response.

Penyimpanan token dengan shared_preferences ensures user session persistence across app restarts. Token harus di-store securely dan checked pada app startup untuk determine authentication state. Token expiration harus di-handle gracefully dengan automatic logout atau token refresh functionality.

Implementasi News Feed adalah core feature aplikasi yang complex. **Integrasi dengan NewsAPI.org** requires understanding API endpoints, parameters, dan response format. HTTP requests harus include proper headers (API key, content-type) dan handle various response scenarios including successful data retrieval, empty results, dan API errors.

Parsing JSON response involves converting API response dari JSON string ke Dart objects menggunakan dart:convert library. Data models harus di-create untuk represent news articles dengan proper null safety handling. Error handling during parsing important untuk prevent crashes dari malformed atau unexpected API responses.

Menampilkan list berita dengan ListView requires efficient widget building dengan ListView.builder untuk performance optimization dengan large datasets. Each list item harus properly display news information (title, description, image, date) dengan attractive layout. Image loading harus handle network images dengan fallback untuk loading states dan error states.

Implementasi Navigation adalah crucial untuk user experience. **Setup curved_navigation_bar** involves adding package dependency dan configuring bottom navigation bar dengan custom curved design. Navigation state harus di-manage untuk indicate current active page dan handle navigation transitions smoothly.

Konfigurasi routing antar halaman includes setting up proper route definitions dan navigation logic. Named routes recommended untuk better organization dan maintainability.

Route parameters harus di-handle untuk passing data between screens, such as passing selected news article ke detail screen.

State management untuk active page ensures consistent navigation state across bottom navigation items. Current page index harus di-track dan update accordingly ketika user navigate between different sections aplikasi.

Implementasi Detail Berita provides comprehensive view dari selected news article. **Passing data antar screens** involves proper parameter passing through route arguments atau constructor parameters. Data integrity harus di-maintain during navigation untuk ensure detail screen receives complete article information.

Layout untuk detail berita harus comprehensive dan user-friendly, displaying article image, title, publication date, source, dan full content dalam readable format. Typography, spacing, dan visual hierarchy important untuk good reading experience. Scrollable content harus handle long articles gracefully.

Handling image loading requires implementing loading states, error states, dan caching untuk better performance. Network images dapat slow to load atau fail completely, sehingga proper placeholder dan error handling essential untuk good user experience.

Implementasi Fitur Kategori adds content organization capability. **Membuat UI kategori dengan ikon** involves creating visually appealing category selection interface dengan representative icons untuk each category (General, Transportation, Entertainment). Grid layout atau horizontal scrolling list dapat digunakan untuk category display.

Implementasi toast message dengan fluttertoast provides immediate feedback ketika user select category. Toast messages harus informative dan not intrusive, giving users confirmation bahwa their selection has been registered. Consider implementing actual category filtering functionality untuk complete feature.

Implementasi Profil Page provides user account management. **Menampilkan informasi user yang login** requires retrieving user data dari storage atau API dan displaying dalam user-friendly format. Profile information dapat include username, email, registration date, atau other relevant user data.

Fungsi logout dan clear session harus comprehensive, removing authentication token dari storage, clearing any cached user data, dan redirecting user ke login screen. Logout confirmation dialog recommended untuk prevent accidental logouts. Session cleanup harus thorough untuk ensure security.

5. HASIL TUGAS

5.1 Deskripsi Aplikasi

News Portal App adalah aplikasi portal berita yang dibangun menggunakan Flutter. Aplikasi ini menampilkan daftar berita teknologi dari TechCrunch melalui NewsAPI.org, serta dilengkapi dengan fitur kategori, detail berita, login, dan registrasi pengguna.

5.2 Struktur Project

```
lib/
├── main.dart          # Entry point aplikasi
├── home.dart          # Halaman utama berita
├── detail.dart        # Detail dari berita
├── kategori.dart      # Pilihan kategori berita
├── profil.dart        # Halaman profil pengguna
├── formlogin.dart     # Form login pengguna
└── formregist.dart    # Form registrasi pengguna
```

5.3 Fitur yang Diimplementasikan

5.3.1 Login & Registrasi (🔐)

- Form login dengan validasi input
- Form registrasi untuk user baru
- Menggunakan API sederhana untuk autentikasi
- Penyimpanan token dengan `shared_preferences`
- Session management untuk maintain login state

5.3.2 Tampilan Berita Utama (📰)

- Menampilkan list berita dari NewsAPI.org
- Sumber berita dari TechCrunch
- Card design untuk setiap berita
- Pull-to-refresh functionality
- Loading indicator saat fetch data

5.3.3 Kategori Berita (📂)

- Ikon-ikon kategori: General, Transportasi, Entertainment
- Grid layout untuk menampilkan kategori
- Toast message ketika kategori dipilih
- Navigation ke halaman kategori spesifik

5.3.4 Detail Berita (📄)

- Halaman detail lengkap untuk setiap berita
- Menampilkan gambar, judul, waktu publish, dan isi berita
- Hero animation untuk transisi gambar

- Share functionality (opsional)

5.3.5 Profil Page (👤)

- Menampilkan nama user yang sedang login
- Avatar/profile picture placeholder
- Informasi akun user
- Tombol logout dengan konfirmasi
- Clear session data saat logout

5.4 Teknologi & Package yang Digunakan

5.4.1 Core Framework

- **Flutter**: Framework utama untuk pengembangan UI
- **Dart**: Bahasa pemrograman yang digunakan

5.4.2 HTTP & API Integration

- **http package**: Untuk melakukan HTTP requests ke NewsAPI
- **NewsAPI.org**: Sumber data berita TechCrunch

5.4.3 Local Storage

- **shared_preferences**: Menyimpan data user dan token secara lokal

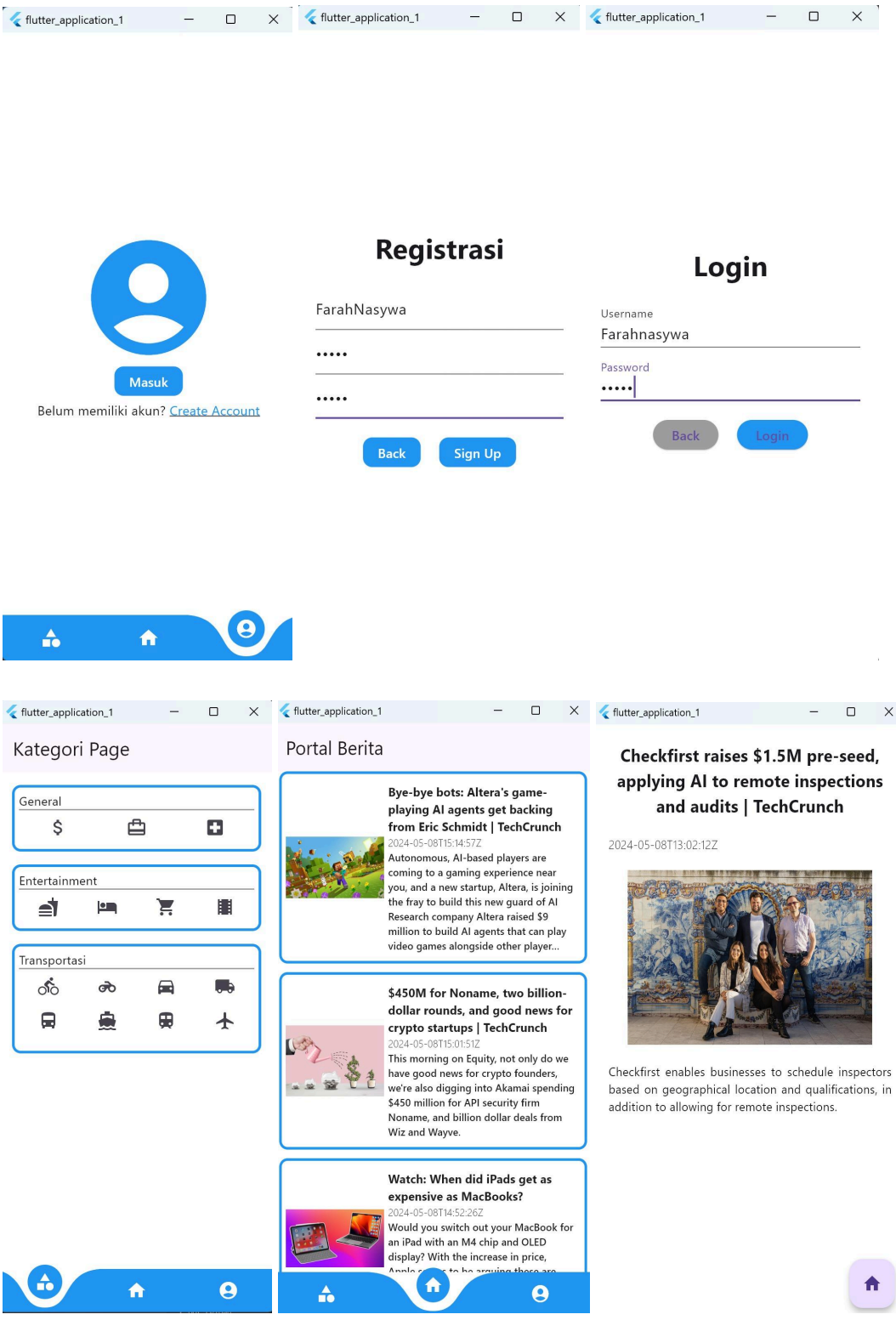
5.4.4 UI/UX Enhancement

- **curved_navigation_bar**: Bottom navigation dengan desain curved
- **fluttertoast**: Menampilkan toast message yang user-friendly

5.5 Alur Kerja Aplikasi

1. **Splash Screen** → Cek status login user
2. **Login/Register** → Jika belum login, tampilkan form autentikasi
3. **Home Screen** → Fetch dan tampilkan berita dari API
4. **Navigation** → Gunakan curved bottom navigation
5. **Detail Screen** → Tampilkan detail berita yang dipilih
6. **Category Screen** → Tampilkan kategori berita dengan toast feedback
7. **Profile Screen** → Tampilkan info user dan opsi logout

5.6 Screenshots dan Demo



6. ANALISIS

6.1 Kelebihan Implementasi

6.1.1 Arsitektur yang Terstruktur

- Pembagian file yang jelas sesuai fungsi masing-masing
- Separation of concerns antara UI dan logic
- Modular design yang mudah di-maintain

6.1.2 User Experience yang Baik

- Curved navigation bar memberikan tampilan modern
- Toast feedback memberikan konfirmasi aksi user
- Loading states yang jelas
- Smooth navigation antar halaman

6.1.3 Integration dengan External Services

- Sukses mengintegrasikan NewsAPI.org
- Implementasi HTTP requests yang proper
- Error handling untuk network issues

6.1.4 State Management

- Penggunaan shared_preferences untuk persistence
- Session management yang baik
- State handling untuk login/logout flow

6.2 Tantangan yang Dihadapi

6.2.1 API Integration

- **Challenge**: Handling different response formats dari API
- **Solution**: Implementasi proper JSON parsing dan error handling

6.2.2 State Management

- **Challenge**: Managing state across multiple screens
- **Solution**: Menggunakan StatefulWidget dan callback functions

6.2.3 UI Responsiveness

- **Challenge**: Membuat UI yang responsive di berbagai ukuran layar
- **Solution**: Menggunakan MediaQuery dan Flexible widgets

6.2.4 Data Persistence

- **Challenge**: Menyimpan session user
- **Solution**: Implementasi shared_preferences untuk local storage

6.3 Aspek Keamanan

- Token-based authentication

- Session timeout handling
- Input validation pada form
- API key management

6.4 Performance Considerations

- Lazy loading untuk images
- Efficient ListView dengan builder
- Caching untuk frequently accessed data
- Minimal rebuild dengan proper state management

7. KESIMPULAN

Dari tugas pembuatan News Portal App dengan Flutter ini dapat disimpulkan bahwa:

7.1 Pencapaian Tujuan Tugas

1. **Berhasil mengimplementasikan aplikasi Flutter kompleks** dengan multiple screens dan navigasi yang baik
2. **Sukses mengintegrasikan external API** (NewsAPI.org) untuk mengambil data berita real-time
3. **Implementasi sistem autentikasi** dengan login/registrasi dan session management
4. **Penerapan best practices** dalam struktur project dan code organization
5. **Penggunaan package eksternal** yang tepat untuk meningkatkan functionality

7.2 Pembelajaran yang Diperoleh

1. **HTTP Integration**: Memahami cara melakukan API calls dan parsing JSON response
2. **State Management**: Mengelola state aplikasi across multiple screens
3. **Navigation**: Implementasi routing dan navigation yang smooth
4. **UI/UX Design**: Membuat interface yang user-friendly dengan package tambahan
5. **Data Persistence**: Menyimpan data lokal dengan shared_preferences

7.3 Keunggulan Flutter

1. **Hot Reload**: Mempercepat development process significantly
2. **Rich Ecosystem**: Package manager yang lengkap untuk berbagai kebutuhan
3. **Cross Platform**: Single codebase untuk multiple platforms
4. **Performance**: Native performance dengan smooth animations
5. **Documentation**: Dokumentasi yang lengkap dan community support yang baik

7.4 Nilai Praktis

Aplikasi News Portal ini menunjukkan implementasi real-world scenario dimana:

- User dapat login dan maintain session
- Data diambil dari external API secara real-time
- Navigation yang intuitive dengan modern UI components
- Proper error handling dan user feedback

Tugas ini memberikan pengalaman lengkap dalam mengembangkan aplikasi mobile dengan Flutter, dari setup environment hingga deployment-ready application.

8. LAMPIRAN

Lampiran : Repository GitHub

GitHub Repository: https://github.com/farahNasywa/Flutter_FarahNasywa_2208107010051