

LAPORAN FLUTTER NEWS PORTAL

disusun untuk memenuhi
Mata Kuliah Pemograman Berbasis Mobile

Oleh:

WIDYA NURUL SUKMA
2208107010054



JURUSAN INFORMATIKA
FAKULTAS METEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SYIAH KUALA
DARUSSALAM, BANDA ACEH
2025

1. PENDAHULUAN

1.1 Latar Belakang

Dalam era digital saat ini, akses terhadap informasi dan berita menjadi kebutuhan yang sangat penting. Aplikasi portal berita mobile memberikan kemudahan bagi pengguna untuk mengakses berita terbaru kapan saja dan di mana saja. Proyek ini dikembangkan sebagai implementasi dari pembelajaran Flutter untuk membangun aplikasi portal berita mobile.

1.2 Tujuan

Tujuan dari pembuatan aplikasi ini adalah:

- Menerapkan konsep-konsep dasar Flutter dalam pengembangan aplikasi mobile
- Mengimplementasikan integrasi dengan REST API untuk mengambil data berita
- Membangun sistem autentikasi pengguna dengan menggunakan Shared Preferences
- Menerapkan konsep navigasi dan routing dalam aplikasi Flutter
- Mengembangkan antarmuka pengguna yang responsif dan menarik

2. DASAR TEORI

2.1 Flutter

Flutter adalah framework aplikasi mobile open source yang dikembangkan oleh Google. Flutter memungkinkan pengembangan aplikasi untuk platform Android dan iOS dengan codebase yang sama. Flutter menggunakan bahasa pemrograman Dart dan menerapkan konsep "everything is a widget" dalam pengembangan UI.

2.2 Widget dalam Flutter

Widget adalah komponen dasar dalam Flutter yang digunakan untuk membangun antarmuka pengguna. Terdapat dua jenis widget utama:

- **Stateless Widget:** Widget yang statis dan tidak dapat berubah-ubah.
- **Stateful Widget:** Widget yang dinamis dan dapat berubah-ubah sesuai dengan interaksi pengguna atau perubahan data.

2.3 Layout Widget

Layout widget digunakan untuk mengatur posisi dan tata letak widget lain. Beberapa layout widget yang umum digunakan:

- **Row:** Mengatur widget secara horizontal.
- **Column:** Mengatur widget secara vertikal.
- **Container:** Membungkus widget lain dan memberikan properti tambahan seperti padding, margin, dan dekorasi.

2.4 HTTP Request

HTTP Request digunakan untuk berkomunikasi dengan server API. Dalam Flutter, package http digunakan untuk melakukan operasi HTTP seperti GET, POST, PUT, dan DELETE.

2.5 Shared Preferences

Shared Preferences adalah mekanisme penyimpanan lokal sederhana dalam aplikasi Flutter. Digunakan untuk menyimpan data dalam bentuk key-value, biasanya untuk menyimpan preferensi pengguna atau token autentikasi.

3. METODOLOGI

3.1 Persiapan Lingkungan Pengembangan

Persiapan awal meliputi:

- Instalasi Flutter SDK
- Konfigurasi IDE (Visual Studio Code)
- Instalasi package yang diperlukan:
 - curved_navigation_bar
 - http
 - shared_preferences
 - fluttertoast

3.2 Pengembangan Aplikasi

Pengembangan aplikasi mengikuti langkah-langkah berikut:

1. Pembuatan struktur proyek dan file-file utama
2. Pengembangan UI untuk setiap halaman
3. Implementasi logika bisnis dan integrasi API
4. Pengembangan sistem autentikasi pengguna
5. Pengujian dan debugging

4. HASIL DAN PEMBAHASAN

4.1 Struktur Aplikasi

Aplikasi terdiri dari beberapa komponen utama:

- **main.dart**: Entry point aplikasi dan navigasi utama.
- **home.dart**: Halaman beranda yang menampilkan daftar berita.
- **detail.dart**: Halaman detail untuk artikel berita.
- **kategori.dart**: Halaman untuk memilih kategori berita.
- **profil.dart**: Halaman profil pengguna.
- **formlogin.dart**: Halaman login pengguna.

- **formregist.dart:** Halaman registrasi pengguna.

4.2 Implementasi Fitur

4.2.1 Navigasi Utama

Navigasi utama diimplementasikan menggunakan `CurvedNavigationBar` yang memberikan tampilan yang menarik dan responsif. Terdapat tiga tab utama: Kategori, Home, dan Profil.

```
bottomNavigationBar: CurvedNavigationBar(
  height: 50.0,
  color: Colors.blue,
  backgroundColor: Colors.white,
  onTap: (index) {
    setState(() {
      _page = index;
    });
  },
  items: <Widget>[
    Icon(Icons.category, color: Colors.white),
    Icon(Icons.home, color: Colors.white),
    Icon(Icons.account_circle, color: Colors.white),
  ], // <Widget>[]
```

4.2.2 Halaman Beranda (Home Page)

Halaman beranda menampilkan daftar berita yang diambil dari API TechCrunch. Setiap berita ditampilkan dalam bentuk card yang berisi judul, tanggal publikasi, deskripsi singkat, dan gambar.

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: Text('Portal Berita')),
    body: FutureBuilder<List>(
      future: getData(),
      builder: (context, snapshot) {
        if (snapshot.connectionState == ConnectionState.waiting) {
          return Center(child: CircularProgressIndicator());
        }
        if (snapshot.hasError) {
          return Center(child: Text('Error: ${snapshot.error}'));
        }
        if (!snapshot.hasData || snapshot.data == null || snapshot.data!.isEmpty) {
          return Center(child: Text('No data available'));
        }
        return ListView.builder(
          itemCount: snapshot.data!.length,
          itemBuilder: (context, i) {
            var article = snapshot.data![i];
            return containerBerita(
              article['publishedAt'] ?? '',
              article['title'] ?? 'No Title',
              article['description'] ?? 'No Description',
              article['urlToImage'] as String? ??
                'https://mgmall.s3.amazonaws.com/img/062023/390bed03e54f6440416f0568f61a82b563176996.jpg',
            );
          },
        );
      },
    ); // ListView.builder
```

4.2.3 Halaman Detail

Halaman detail menampilkan artikel berita secara lengkap dengan judul, tanggal publikasi, gambar, dan konten. Halaman ini diakses ketika pengguna menekan salah satu berita di halaman beranda.

```
class _DetailPageState extends State<DetailPage> {
  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        floatingActionButton: FloatingActionButton(
          child: Icon(Icons.home),
          onPressed: () {
            Navigator.pop(context);
          },
        ), // FloatingActionButton
        body: Container(
          padding: EdgeInsets.fromLTRB(20, 20, 20, 80),
          child: ListView(
            children: [
              Text(
                widget.judul ?? 'No Title', // Menangani null dengan memberi default value
                style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
                textAlign: TextAlign.center,
              ), // Text
              SizedBox(height: 20),
              Text(
                widget.datetime ?? 'No Date Provided', // Menangani null
                style: TextStyle(fontWeight: FontWeight.w300),
              ), // Text
              SizedBox(height: 20),
              widget.image != null
                ? Image.network(
                    widget.image!,
                    height: 200,

```

4.2.4 Halaman Kategori

Halaman kategori menampilkan berbagai kategori berita yang dapat dipilih oleh pengguna. Kategori dibagi menjadi General, Entertainment, dan Transportasi. Code berada pada file kategori.dart.

```
Container(
  padding: EdgeInsets.all(5),
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(10),
    border: Border.all(color: Colors.blue, width: 3),
  ), // BoxDecoration
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text('General'),
      Container(height: 1.5, color: Colors.grey),
      Row(
        mainAxisAlignment: MainAxisAlignment.spaceAround,
        children: [
          iconTag(icon: Icon(Icons.attach_money), tag: 'General'),
          iconTag(icon: Icon(Icons.card_travel), tag: 'General'),
          iconTag(icon: Icon(Icons.local_hospital), tag: 'General'),
        ],
      ),
    ],
  ),
)
```

4.2.5 Sistem Autentikasi

Sistem autentikasi diimplementasikan menggunakan Shared Preferences untuk menyimpan token dan informasi pengguna. Terdapat halaman login dan registrasi untuk manajemen pengguna. Code berada pada file formregist.dart

```
Future<void> loginUser(String username, String password) async {
  setState(() => isLoading = true);
  final url = Uri.parse("https://e-commerce-store.glitch.me/login");
  try {
    final response = await http.post(
      url,
      headers: {"Content-Type": "application/json"},
      body: jsonEncode({"username": username, "password": password}),
    );
    final statusCode = response.statusCode;
    final body = jsonDecode(response.body);
    if (statusCode == 200 && body['status'] == true) {
      final user = body['data']['user'];
      final token = body['data']['token'];
      SharedPreferences prefs = await SharedPreferences.getInstance();
      await prefs.setString('username', user['username']);
      await prefs.setString('token', token);
      showToast("Login berhasil", isSuccess: true);
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (_) => MyApp(initialPage: 0)),
      );
    } else {
      showToast(
        "Login gagal: ${body['message'] ?? 'Username atau password salah'}",
      );
    }
  }
}
```

4.2.6 Halaman Profil

Halaman profil menampilkan informasi pengguna jika sudah login atau opsi untuk login/registrasi jika belum login. Code berada pada file profil.dart

```
@override
Widget build(BuildContext context) {
  return FutureBuilder<String?>(
    future: _loginFuture,
    builder: (context, snapshot) {
      if (snapshot.connectionState == ConnectionState.waiting) {
        return Center(child: CircularProgressIndicator());
      }
      final username = snapshot.data;
      return Center(
        child: username != null ? sudahLogin(username) : belumLogin(),
      ); // Center
    },
  ); // FutureBuilder
}
```

4.3 Integrasi API

Aplikasi mengintegrasikan API News untuk mendapatkan data berita terbaru dari TechCrunch. Data diambil menggunakan HTTP request dan ditampilkan dalam format yang sesuai. Code ini terdapat pada file formregist.dart

```
class _HomePageState extends State<HomePage> {  
  Future<List> getData() async {  
    final response = await http.get(Uri.parse(linkAPI));  
    if (response.statusCode == 200) {  
      // Dekode JSON dan ambil data dari key 'articles'  
      var data = json.decode(response.body);  
      // Pastikan mengambil data dari key 'articles' yang merupakan List  
      return data['articles']; // Mengembalikan data sebagai List<dynamic>  
    } else {  
      throw Exception('Failed to load data');  
    }  
  }  
}
```

4.4 Tantangan dan Solusi

Selama pengembangan, beberapa tantangan yang dihadapi:

1. Menangani Data Null dan Error

- Tantangan: API terkadang mengembalikan data yang tidak lengkap atau null.
- Solusi: Menggunakan operator null-aware dan nilai default untuk menangani data yang mungkin null.

2. Sistem Autentikasi

- Tantangan: Mengimplementasikan alur autentikasi yang aman dan efisien.
- Solusi: Menggunakan Shared Preferences untuk menyimpan token dan informasi pengguna secara lokal.

3. Performa Aplikasi

- Tantangan: Memastikan performa aplikasi tetap baik saat menampilkan banyak data.
- Solusi: Menggunakan FutureBuilder dan optimasi loading untuk meningkatkan performa aplikasi.

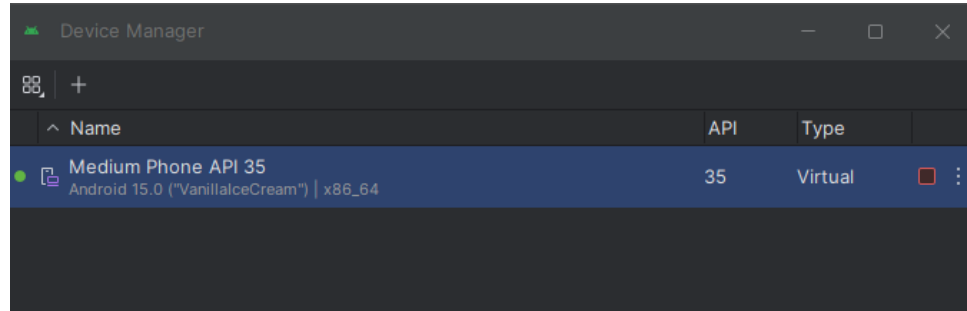
5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

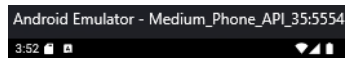
Proyek Flutter News Portal berhasil diimplementasikan dengan berbagai fitur seperti menampilkan berita dari API, sistem autentikasi pengguna, dan navigasi yang responsif. Aplikasi ini mendemonstrasikan kemampuan Flutter dalam membangun aplikasi mobile yang efisien dan menarik dengan waktu pengembangan yang relatif singkat. Konsep-konsep dasar Flutter seperti widget, state management, dan integrasi API berhasil diterapkan dalam proyek ini.

LAMPIRAN

Lampiran 1: Screenshot Run Aplikasi



Buka Device Manager pada Android Studio, kemudian jalan kan Medium Phone API Sehingga Android Emulator muncul, seperti gambar dibawah ini.



Android Emulator


```

C:\project\flutter_application_1>flutter run
Launching lib\main.dart on sdk gphone64 x86 64 in debug mode...
Running Gradle task 'assembleDebug'... 123.0s\
✓ Built build\app\outputs\flutter-apk\app-debug.apk
Installing build\app\outputs\flutter-apk\app-debug.apk... 81.3s\
I/flutter ( 4689): [IMPORTANT:flutter/shell/platform/android/android_context_gl_impeller.cc(94)]
Syncing files to device sdk gphone64 x86 64... 4.5s\

Flutter run key commands.
r Hot reload.
R Hot restart.
h List all available interactive commands.
d Detach (terminate "flutter run" but leave application running).
c Clear the screen
q Quit (terminate the application on the device).

A Dart VM Service on sdk gphone64 x86 64 is available at:
http://127.0.0.1:51823/hVvPvuEcS7s=/
The Flutter DevTools debugger and profiler on sdk gphone64 x86 64 is
available at:
http://127.0.0.1:9101?uri=http://127.0.0.1:51823/hVvPvuEcS7s=/

```

Buka CMD, Masuk ke dalam file project, kemudian jalankan command ‘flutter run’. Sehingga nantinya flutter project akan berjalan pada Android Emulator.

```

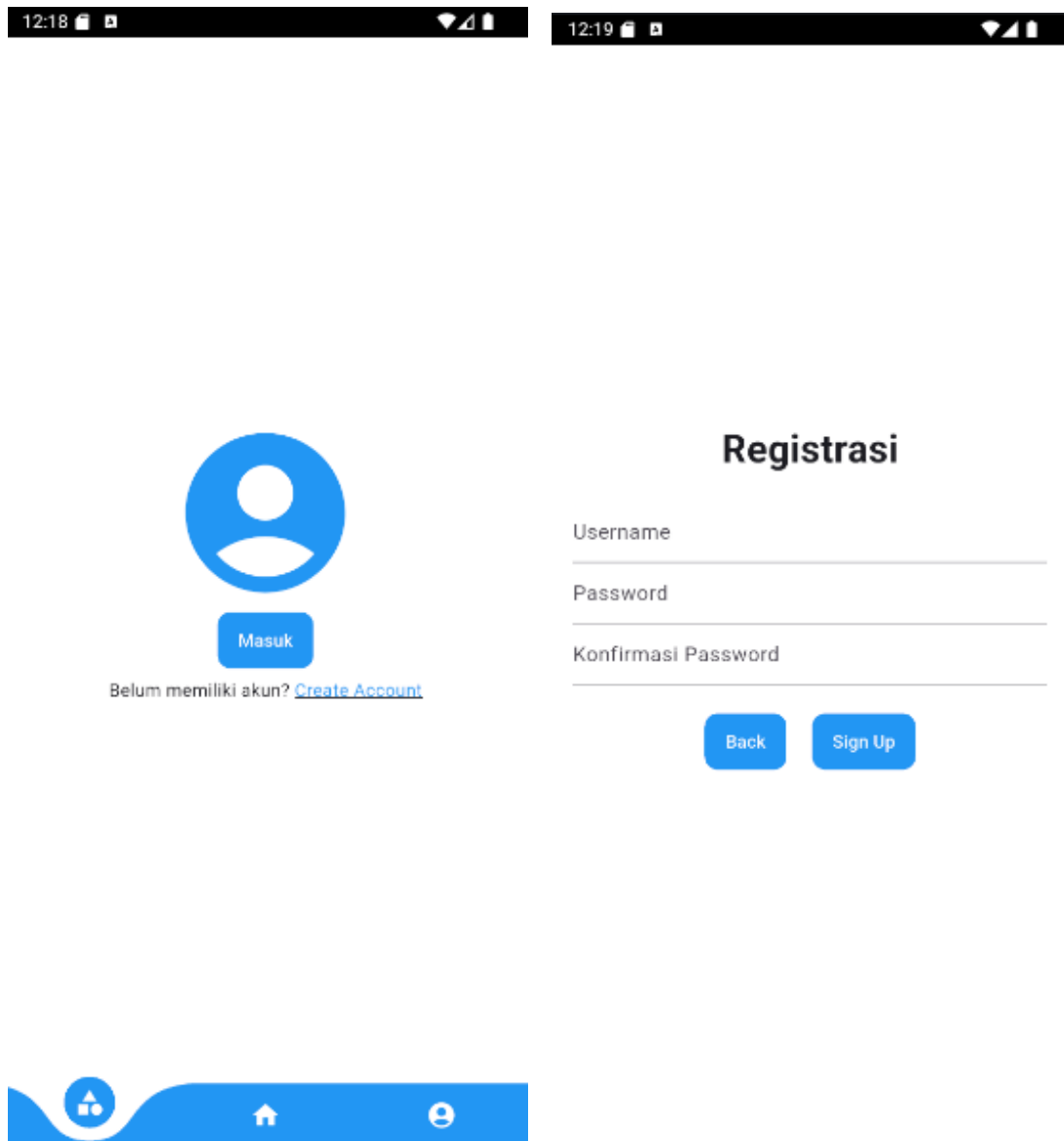
C:\project\flutter_application_1>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.29.3, on Microsoft Windows [Version
    10.0.22631.5335], locale en-GB)
[✓] Windows Version (11 Pro 64-bit, 23H2, 2009)
[✓] Android toolchain - develop for Android devices (Android SDK version
    35.0.1)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop Windows apps (Visual Studio Community 2022
    17.14.0)
[✓] Android Studio (version 2024.2)
[✓] VS Code (version 1.74.3)
[✓] Connected device (4 available)
[✓] Network resources

• No issues found!

```

Tampilan saat menjalankan command ‘flutter doctor’. Tidak ada lagi masalah pada semua details.

Lampiran 2: Screenshot Aplikasi



Tampilan diatas adalah halaman untuk masuk dan juga saat menekan ‘Create Account’ di bawa ke halaman Registrasi.

12:18

Login

Username

Password

[Back](#) [Login](#)

11:55

Kategori Page

General

\$  

Entertainment

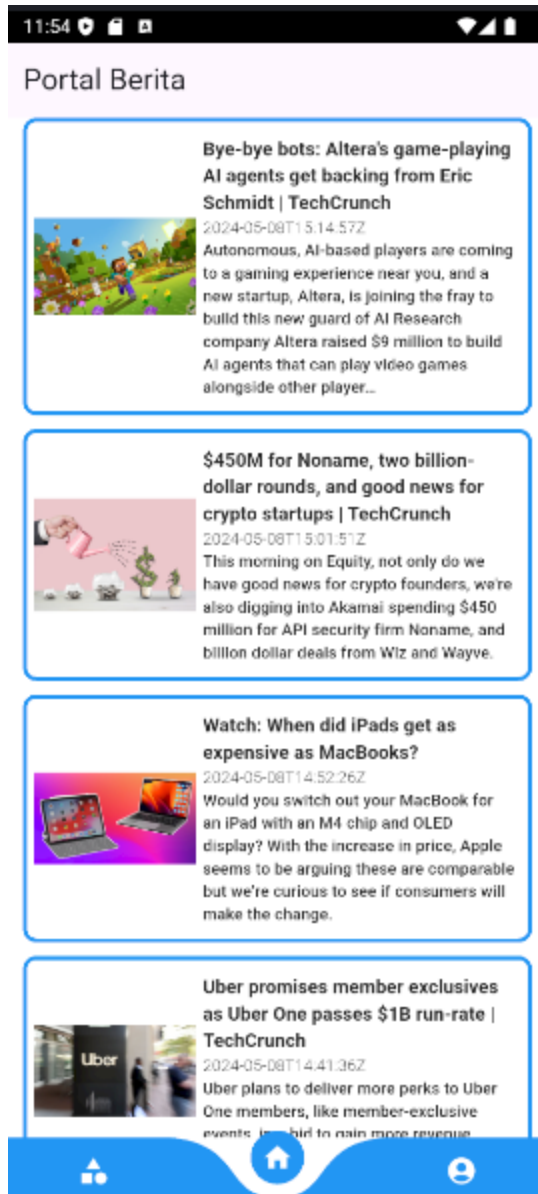
   

Transportasi

Setelah melakukan Registrasi, langsung saja masuk dengan Masukkan Username serta Password. Setelah berhasil login, Halaman nya beralih ke Halaman Kategori Page.



Diatas merupakan Halaman Portal Berita, yang berisi banyak berita dan disampingnya merupakan Halaman User. Halaman user menampilkan nama pengguna serta button untuk keluar dari aplikasi.