LAPORAN

Implementasi Firebase Authentication pada Aplikasi Flutter

Disusun oleh :

Suci Indrawati

(5230411113)

PROGRAM STUDI INFORMATIKA FAKULTAS SAINS & TEKNOLOGI

UNIVERSITAS TEKNOLOGI YOGYAKARTA

2025/2026

1. Bab I

1.1. Pendahuluan

Implementasi Firebase Authentication pada aplikasi Flutter untuk portal berita berfungsi untuk menghadirkan sistem login yang aman, modern, dan mudah digunakan. Dengan integrasi ini, aplikasi dapat mengatur akses pengguna terhadap fitur seperti menyimpan artikel favorit, memberikan komentar, atau mengakses konten khusus. Prosesnya dimulai dengan menghubungkan proyek Flutter ke Firebase, menambahkan paket *firebase_core* dan *firebase_auth*, serta melakukan inisialisasi di fungsi `main()`. Setelah itu, aplikasi dapat membuat fitur login, registrasi, dan logout menggunakan API Firebase, serta memanfaatkan *authStateChanges()* untuk memantau status pengguna secara real-time. Setiap pengguna yang login mendapatkan UID unik yang dapat digunakan untuk menyimpan preferensi atau aktivitas membaca di Firestore. Autentikasi ini juga penting bagi admin untuk membatasi akses ke halaman pengelolaan berita. Secara keseluruhan, implementasi Firebase Authentication meningkatkan keamanan dan memberikan pengalaman pengguna yang lebih personal dan terstruktur dalam portal berita berbasis Flutter.

1.2. Tujuan

Berikut tujuan implementasi Firebase Authentication pada aplikasi Flutter untuk portal berita:

1. Menyediakan sistem login yang aman dan terpercaya sehingga hanya pengguna terverifikasi yang dapat mengakses fitur tertentu.
2. Mengatur hak akses pengguna, misalnya membedakan pengguna biasa dan admin dalam mengelola atau membaca berita.
3. Memudahkan proses autentikasi dengan menyediakan berbagai metode login seperti email, Google, atau nomor telepon.
4. Mendukung personalisasi konten, seperti menyimpan artikel favorit, riwayat bacaan, atau kategori berita favorit berdasarkan UID pengguna.
5. Meningkatkan pengalaman pengguna melalui proses masuk yang cepat, stabil, dan terintegrasi dengan baik dalam aplikasi Flutter.
6. Mempermudah pengelolaan data pengguna karena akun yang terautentikasi tersinkron dengan layanan Firebase lainnya seperti Firestore.
7. Menjaga keamanan portal berita, terutama untuk fitur yang sensitif seperti pengunggahan atau pengeditan berita oleh admin.

1.3. Manfaat

Implementasi Firebase Authentication pada aplikasi Flutter memberikan berbagai manfaat penting bagi portal berita, terutama dalam hal keamanan dan kemudahan pengelolaan pengguna. Dengan sistem autentikasi yang kuat, aplikasi dapat memastikan bahwa hanya pengguna yang terverifikasi yang dapat mengakses

fitur tertentu, sehingga mengurangi risiko penyalahgunaan akun. Firebase juga menyediakan proses login yang cepat dan fleksibel melalui berbagai metode, seperti email, Google, atau nomor telepon, yang meningkatkan kenyamanan pengguna ketika mengakses aplikasi. Selain itu, Firebase Authentication mempermudah integrasi dengan layanan Firebase lainnya, sehingga pengembang dapat menyimpan data pengguna seperti artikel favorit dan riwayat bacaan dengan lebih terstruktur. Manfaat lain yang dirasakan adalah meningkatnya pengalaman pengguna karena mereka mendapatkan akses yang lebih personal dan aman, sementara admin dapat bekerja lebih efisien dalam mengelola konten berita tanpa harus membuat sistem autentikasi dari nol.

2. Bab II

2.1. Isi kode/Langkah langkah

1. Lib/Models

Ini berupa folder pada proyek Flutter yang digunakan untuk menyimpan model data.File berita.dart merupakan model data yang digunakan untuk merepresentasikan struktur berita, termasuk ID, judul, konten, penulis, tanggal publikasi, dan URL gambar. Class ini juga menyediakan method untuk konversi data dari dan ke format Map, serta factory constructor untuk mengolah data dari berbagai sumber API.

aplikasi. Yang berupa :

a. **Berita.dart**

File ini berisi class model Berita yang digunakan untuk merepresentasikan data berita di aplikasi portal berita Flutter. Model ini bisa

- Menyimpan Data berita
- Mengubah data ke format Map (untuk Firebase/database)
- Mengambil data dari berbagai sumebr API (NewsAPI & NewsData.io

Kode ini beruapa

```
class Berita {
  final String id;
  final String judul;
  final String konten;
  final String penulis;
  final DateTime tanggal;
  final String? gambarUrl;

  Berita({
    required this.id,
    required this.judul,
    required this.konten,
    required this.penulis,
    required this.tanggal,
    this.gambarUrl,
```

```
});

// Convert to Map for storage
Map<String, dynamic> toMap() {
  return {
    'id': id,
    'judul': judul,
    'konten': konten,
    'penulis': penulis,
    'tanggal': tanggal.toIso8601String(),
    'gambarUrl': gambarUrl,
  };
}

// Create from Map
factory Berita.fromMap(Map<String, dynamic> map) {
  return Berita(
    id: map['id'] ?? '',
    judul: map['judul'] ?? '',
    konten: map['konten'] ?? '',
    penulis: map['penulis'] ?? '',
    tanggal: DateTime.parse(map['tanggal']),
    gambarUrl: map['gambarUrl'],
  );
}

// Create from NewsAPI response (backward compatibility)
factory Berita.fromNewsApi(Map<String, dynamic> article) {
  // Parse published date
  DateTime? publishedDate;
  try {
    if (article['publishedAt'] != null) {
      publishedDate = DateTime.parse(article['publishedAt']);
    }
  } catch (e) {
    publishedDate = DateTime.now();
  }

  // Get author or source name
  String author = article['author'] ??
                  article['source']?['name'] ??
                  'Tidak diketahui';

  // Get content or description
  String content = article['content'] ??
                   article['description'] ??
                   article['title'] ??
                   'Tidak ada konten';

  // Remove [Removed] or [Source] tags from content
  content = content.replaceAll(RegExp(r'\[.*?\]'), '').trim();
  if (content.isEmpty) {
    content = article['description'] ?? article['title'] ?? 'Tidak ada konten';
  }

  return Berita(
    id: article['url'] ?? DateTime.now().millisecondsSinceEpoch.toString(),
    judul: article['title'] ?? 'Tanpa judul',
    konten: content,
    penulis: author,
    tanggal: publishedDate ?? DateTime.now(),
```

```
      gambarUrl: article['urlToImage'],
    );
  }

  // Create from NewsData.io response
  factory Berita.fromNewsDataIo(Map<String, dynamic> article) {
    // Parse published date (format: "2025-12-06 18:33:52")
    DateTime? publishedDate;
    try {
      if (article['pubDate'] != null) {
        final dateStr = article['pubDate'].toString();
        // Convert format "2025-12-06 18:33:52" to ISO format
        publishedDate = DateTime.parse(dateStr.replaceAll(' ', 'T'));
      }
    } catch (e) {
      publishedDate = DateTime.now();
    }

    // Get author/creator (bisa array atau string)
    String author = 'Tidak diketahui';
    if (article['creator'] != null) {
      if (article['creator'] is List) {
        final creators = article['creator'] as List;
        if (creators.isNotEmpty) {
          author = creators.first.toString();
        }
      } else {
        author = article['creator'].toString();
      }
    } else if (article['source_name'] != null) {
      author = article['source_name'].toString();
    }

    // Get content or description
    String content = article['description'] ??
                     article['content'] ??
                     article['title'] ??
                     'Tidak ada konten';

    // Remove tags if content contains "ONLY AVAILABLE IN PAID PLANS"
    if (content.contains('ONLY AVAILABLE')) {
      content = article['description'] ?? article['title'] ?? 'Tidak ada konten';
    }

    // Get image URL - ini yang penting untuk menampilkan gambar!
    String? imageUrl = article['image_url'];

    return Berita(
      id: article['article_id'] ??
          article['link'] ??
          DateTime.now().millisecondsSinceEpoch.toString(),
      judul: article['title'] ?? 'Tanpa judul',
      konten: content,
      penulis: author,
      tanggal: publishedDate ?? DateTime.now(),
      gambarUrl: imageUrl, // Menggunakan image_url dari newsdata.io
    );
  }
}
```

2. **User_profil.dart**

   File ini berisi **model data profil pengguna (`UserProfile`)** yang digunakan untuk menyimpan informasi akun user dalam aplikasi Flutter (portal berita). File user_profile.dart merupakan model data yang digunakan untuk menyimpan informasi profil pengguna seperti UID, email, nama tampilan, foto profil, bio, dan nomor telepon. Class ini menyediakan method untuk mengonversi data ke dan dari format Map, serta method copyWith() untuk mempermudah proses pembaruan data profil pengguna.

   Model ini membantu aplikasi:

   - Menyimpan data user dengan struktur
   - Mengirim data ke Firebase/databe
   - Mengambil dan memperbarui data profil user

```dart
class Berita {
  final String id;
  final String judul;
  final String konten;
  final String penulis;
  final DateTime tanggal;
  final String? gambarUrl;

  Berita({
    required this.id,
    required this.judul,
    required this.konten,
    required this.penulis,
    required this.tanggal,
    this.gambarUrl,
  });

  // Convert to Map for storage
  Map<String, dynamic> toMap() {
    return {
      'id': id,
      'judul': judul,
      'konten': konten,
      'penulis': penulis,
      'tanggal': tanggal.toIso8601String(),
      'gambarUrl': gambarUrl,
    };
  }

  // Create from Map
  factory Berita.fromMap(Map<String, dynamic> map) {
    return Berita(
      id: map['id'] ?? '',
      judul: map['judul'] ?? '',
      konten: map['konten'] ?? '',
      penulis: map['penulis'] ?? '',
      tanggal: DateTime.parse(map['tanggal']),
      gambarUrl: map['gambarUrl'],
    );
  }

  // Create from NewsAPI response (backward compatibility)
```

```dart
  factory Berita.fromNewsApi(Map<String, dynamic> article) {
    // Parse published date
    DateTime? publishedDate;
    try {
      if (article['publishedAt'] != null) {
        publishedDate = DateTime.parse(article['publishedAt']);
      }
    } catch (e) {
      publishedDate = DateTime.now();
    }

    // Get author or source name
    String author = article['author'] ??
                    article['source']?['name'] ??
                    'Tidak diketahui';

    // Get content or description
    String content = article['content'] ??
                     article['description'] ??
                     article['title'] ??
                     'Tidak ada konten';

    // Remove [Removed] or [Source] tags from content
    content = content.replaceAll(RegExp(r'\[.*?\]'), '').trim();
    if (content.isEmpty) {
      content = article['description'] ?? article['title'] ?? 'Tidak ada konten';
    }

    return Berita(
      id: article['url'] ?? DateTime.now().millisecondsSinceEpoch.toString(),
      judul: article['title'] ?? 'Tanpa judul',
      konten: content,
      penulis: author,
      tanggal: publishedDate ?? DateTime.now(),
      gambarUrl: article['urlToImage'],
    );
  }

  // Create from NewsData.io response
  factory Berita.fromNewsDataIo(Map<String, dynamic> article) {
    // Parse published date (format: "2025-12-06 18:33:52")
    DateTime? publishedDate;
    try {
      if (article['pubDate'] != null) {
        final dateStr = article['pubDate'].toString();
        // Convert format "2025-12-06 18:33:52" to ISO format
        publishedDate = DateTime.parse(dateStr.replaceAll(' ', 'T'));
      }
    } catch (e) {
      publishedDate = DateTime.now();
    }

    // Get author/creator (bisa array atau string)
    String author = 'Tidak diketahui';
    if (article['creator'] != null) {
      if (article['creator'] is List) {
        final creators = article['creator'] as List;
        if (creators.isNotEmpty) {
          author = creators.first.toString();
        }
      } else {
```

```
        author = article['creator'].toString();
      }
    } else if (article['source_name'] != null) {
      author = article['source_name'].toString();
    }

    // Get content or description
    String content = article['description'] ??
                     article['content'] ??
                     article['title'] ??
                     'Tidak ada konten';

    // Remove tags if content contains "ONLY AVAILABLE IN PAID PLANS"
    if (content.contains('ONLY AVAILABLE')) {
      content = article['description'] ?? article['title'] ?? 'Tidak ada konten';
    }

    // Get image URL - ini yang penting untuk menampilkan gambar!
    String? imageUrl = article['image_url'];

    return Berita(
      id: article['article_id'] ??
          article['link'] ??
          DateTime.now().millisecondsSinceEpoch.toString(),
      judul: article['title'] ?? 'Tanpa judul',
      konten: content,
      penulis: author,
      tanggal: publishedDate ?? DateTime.now(),
      gambarUrl: imageUrl, // Menggunakan image_url dari newsdata.io
    );
  }
}
```

2.2 Lib\screen\auth

1. login_screen.dart

File login_screen.dart berfungsi sebagai halaman autentikasi pengguna yang memungkinkan login menggunakan email dan password. File ini menggunakan Firebase Authentication melalui AuthService, dilengkapi validasi input, indikator loading, serta navigasi ke halaman registrasi.

```
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import '../../services/auth_service.dart';
import 'register_screen.dart';

class LoginScreen extends StatefulWidget {
  const LoginScreen({super.key});

  @override
  State<LoginScreen> createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  final _formKey = GlobalKey<FormState>();
  final _emailController = TextEditingController();
  final _passwordController = TextEditingController();
  bool _isLoading = false;
  bool _obscurePassword = true;
```

```dart
@override
void dispose() {
  _emailController.dispose();
  _passwordController.dispose();
  super.dispose();
}

Future<void> _handleLogin() async {
  if (_formKey.currentState!.validate()) {
    setState(() => _isLoading = true);
    try {
      final authService = Provider.of<AuthService>(context, listen: false);
      await authService.signInWithEmailAndPassword(
        _emailController.text.trim(),
        _passwordController.text,
      );
      if (mounted) {
        Navigator.of(context).pushReplacementNamed('/home');
      }
    } catch (e) {
      if (mounted) {
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
            content: Text('Login gagal: ${e.toString()}'),
            backgroundColor: Colors.red,
          ),
        );
      }
    } finally {
      if (mounted) {
        setState(() => _isLoading = false);
      }
    }
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      decoration: BoxDecoration(
        gradient: LinearGradient(
          begin: Alignment.topLeft,
          end: Alignment.bottomRight,
          colors: [
            Colors.blue.shade700,
            Colors.purple.shade700,
          ],
        ),
      ),
      child: SafeArea(
        child: Center(
          child: SingleChildScrollView(
            padding: const EdgeInsets.all(24.0),
            child: Form(
              key: _formKey,
              child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  // Logo/Icon
```

```
                Container(
                  padding: const EdgeInsets.all(20),
                  decoration: BoxDecoration(
                    color: Colors.white,
                    shape: BoxShape.circle,
                  ),
                  child: Icon(
                    Icons.newspaper,
                    size: 60,
                    color: Colors.blue.shade700,
                  ),
                ),
                const SizedBox(height: 30),
                // Title
                const Text(
                  'Portal Berita',
                  style: TextStyle(
                    fontSize: 32,
                    fontWeight: FontWeight.bold,
                    color: Colors.white,
                  ),
                ),
                const SizedBox(height: 10),
                const Text(
                  'Masuk ke akun Anda',
                  style: TextStyle(
                    fontSize: 16,
                    color: Colors.white70,
                  ),
                ),
                const SizedBox(height: 40),
                // Email Field
                Container(
                  decoration: BoxDecoration(
                    borderRadius: BorderRadius.circular(16),
                    boxShadow: [
                      BoxShadow(
                        color: Colors.black.withOpacity(0.08),
                        blurRadius: 15,
                        offset: const Offset(0, 5),
                      ),
                    ],
                  ),
                  child: TextFormField(
                    controller: _emailController,
                    keyboardType: TextInputType.emailAddress,
                    style: const TextStyle(
                      fontSize: 16,
                      fontWeight: FontWeight.w500,
                    ),
                    decoration: InputDecoration(
                      labelText: 'Email',
                      hintText: 'contoh@email.com',
                      prefixIcon: Container(
                        margin: const EdgeInsets.all(12),
                        padding: const EdgeInsets.all(8),
                        decoration: BoxDecoration(
                          color: Colors.blue.shade50,
                          borderRadius: BorderRadius.circular(10),
                        ),
                        child: Icon(
```

```
            Icons.email_outlined,
            color: Colors.blue.shade700,
            size: 20,
          ),
        ),
        filled: true,
        fillColor: Colors.white,
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(16),
          borderSide: BorderSide.none,
        ),
        enabledBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(16),
          borderSide: BorderSide(
            color: Colors.grey.shade200,
            width: 1.5,
          ),
        ),
        focusedBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(16),
          borderSide: BorderSide(
            color: Colors.blue.shade700,
            width: 2,
          ),
        ),
        errorBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(16),
          borderSide: BorderSide(
            color: Colors.red.shade300,
            width: 1.5,
          ),
        ),
        focusedErrorBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(16),
          borderSide: BorderSide(
            color: Colors.red.shade700,
            width: 2,
          ),
        ),
        labelStyle: TextStyle(
          color: Colors.grey.shade600,
          fontSize: 16,
        ),
        floatingLabelStyle: TextStyle(
          color: Colors.blue.shade700,
          fontWeight: FontWeight.w600,
        ),
        contentPadding: const EdgeInsets.symmetric(
          horizontal: 20,
          vertical: 18,
        ),
      ),
    ),
    validator: (value) {
      if (value == null || value.isEmpty) {
        return 'Email tidak boleh kosong';
      }
      if (!value.contains('@')) {
        return 'Email tidak valid';
      }
      return null;
    },
```

```
        ),
      ),
      const SizedBox(height: 20),
      // Password Field
      Container(
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(16),
          boxShadow: [
            BoxShadow(
              color: Colors.black.withOpacity(0.08),
              blurRadius: 15,
              offset: const Offset(0, 5),
            ),
          ],
        ),
        child: TextFormField(
          controller: _passwordController,
          obscureText: _obscurePassword,
          style: const TextStyle(
            fontSize: 16,
            fontWeight: FontWeight.w500,
          ),
          decoration: InputDecoration(
            labelText: 'Password',
            hintText: 'Minimal 6 karakter',
            prefixIcon: Container(
              margin: const EdgeInsets.all(12),
              padding: const EdgeInsets.all(8),
              decoration: BoxDecoration(
                color: Colors.purple.shade50,
                borderRadius: BorderRadius.circular(10),
              ),
              child: Icon(
                Icons.lock_outlined,
                color: Colors.purple.shade700,
                size: 20,
              ),
            ),
            suffixIcon: Container(
              margin: const EdgeInsets.only(right: 8),
              child: IconButton(
                icon: Icon(
                  _obscurePassword
                      ? Icons.visibility_outlined
                      : Icons.visibility_off_outlined,
                  color: Colors.grey.shade600,
                ),
                onPressed: () {
                  setState(() {
                    _obscurePassword = !_obscurePassword;
                  });
                },
              ),
            ),
            filled: true,
            fillColor: Colors.white,
            border: OutlineInputBorder(
              borderRadius: BorderRadius.circular(16),
              borderSide: BorderSide.none,
            ),
            enabledBorder: OutlineInputBorder(
```

```
                borderRadius: BorderRadius.circular(16),
                borderSide: BorderSide(
                  color: Colors.grey.shade200,
                  width: 1.5,
                ),
              ),
              focusedBorder: OutlineInputBorder(
                borderRadius: BorderRadius.circular(16),
                borderSide: BorderSide(
                  color: Colors.purple.shade700,
                  width: 2,
                ),
              ),
              errorBorder: OutlineInputBorder(
                borderRadius: BorderRadius.circular(16),
                borderSide: BorderSide(
                  color: Colors.red.shade300,
                  width: 1.5,
                ),
              ),
              focusedErrorBorder: OutlineInputBorder(
                borderRadius: BorderRadius.circular(16),
                borderSide: BorderSide(
                  color: Colors.red.shade700,
                  width: 2,
                ),
              ),
              labelStyle: TextStyle(
                color: Colors.grey.shade600,
                fontSize: 16,
              ),
              floatingLabelStyle: TextStyle(
                color: Colors.purple.shade700,
                fontWeight: FontWeight.w600,
              ),
              contentPadding: const EdgeInsets.symmetric(
                horizontal: 20,
                vertical: 18,
              ),
            ),
            validator: (value) {
              if (value == null || value.isEmpty) {
                return 'Password tidak boleh kosong';
              }
              if (value.length < 6) {
                return 'Password minimal 6 karakter';
              }
              return null;
            },
          ),
        ),
        const SizedBox(height: 30),
        // Login Button
        SizedBox(
          width: double.infinity,
          height: 50,
          child: ElevatedButton(
            onPressed: _isLoading ? null : _handleLogin,
            style: ElevatedButton.styleFrom(
              backgroundColor: Colors.white,
              foregroundColor: Colors.blue.shade700,
```

```
                        shape: RoundedRectangleBorder(
                          borderRadius: BorderRadius.circular(12),
                        ),
                        elevation: 5,
                      ),
                      child: _isLoading
                          ? const SizedBox(
                              height: 20,
                              width: 20,
                              child: CircularProgressIndicator(strokeWidth: 2),
                            )
                          : const Text(
                              'Masuk',
                              style: TextStyle(
                                fontSize: 18,
                                fontWeight: FontWeight.bold,
                              ),
                            ),
                    ),
                  ),
                  const SizedBox(height: 20),
                  // Register Link
                  Row(
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: [
                      const Text(
                        'Belum punya akun? ',
                        style: TextStyle(color: Colors.white70),
                      ),
                      TextButton(
                        onPressed: () {
                          Navigator.push(
                            context,
                            MaterialPageRoute(
                              builder: (context) => const RegisterScreen(),
                            ),
                          );
                        },
                        child: const Text(
                          'Daftar',
                          style: TextStyle(
                            color: Colors.white,
                            fontWeight: FontWeight.bold,
                            decoration: TextDecoration.underline,
                          ),
                        ),
                      ),
                    ],
                  ),
                ],
              ),
            ),
          ),
        ),
      ),
    );
  }
}
```

2. Register_screen.dart

File register_screen.dart berfungsi sebagai halaman pendaftaran akun baru pada aplikasi portal berita. Halaman ini memungkinkan pengguna membuat akun menggunakan email dan password, melakukan validasi input, menampilkan indikator loading saat proses berlangsung, serta menampilkan pesan ketika pendaftaran berhasil atau gagal. Setelah registrasi berhasil, pengguna diarahkan ke halaman utama atau halaman login melalui integrasi dengan Firebase Authentication melalui AuthService. Kode tersebut adalah fitur **registrasi akun** pada aplikasi Flutter.Fungsinya untuk:

- Mnedaftarkan pengguna baru menggunakan email&password melalui Firebase Auth
- Menyimpan data profil ke Firestore dalam bentuk model UserProfil
- Mengarahkab pengguna ke halaaman login setelah berhasil daftar

```
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import '../../services/auth_service.dart';
import '../../services/profile_service.dart';
import '../../models/user_profile.dart';

class RegisterScreen extends StatefulWidget {
  const RegisterScreen({super.key});

  @override
  State<RegisterScreen> createState() => _RegisterScreenState();
}

class _RegisterScreenState extends State<RegisterScreen> {
  final _formKey = GlobalKey<FormState>();
  final _emailController = TextEditingController();
  final _passwordController = TextEditingController();
  final _confirmPasswordController = TextEditingController();
  bool _isLoading = false;
  bool _obscurePassword = true;
  bool _obscureConfirmPassword = true;

  @override
  void dispose() {
    _emailController.dispose();
    _passwordController.dispose();
    _confirmPasswordController.dispose();
    super.dispose();
  }

  Future<void> _handleRegister() async {
    if (_formKey.currentState!.validate()) {
      setState(() => _isLoading = true);
      try {
        final authService = Provider.of<AuthService>(context, listen: false);
        final profileService = Provider.of<ProfileService>(context, listen:
false);

        // Register user
```

```
      final userCredential = await authService.registerWithEmailAndPassword(
        _emailController.text.trim(),
        _passwordController.text,
      );

      // Buat profil di Firestore
      if (userCredential?.user != null) {
        final user = userCredential!.user!;
        final profile = UserProfile(
          uid: user.uid,
          email: user.email ?? '',
          displayName: user.displayName,
          photoURL: user.photoURL,
        );
        await profileService.createProfile(profile);
      }

      // Logout setelah registrasi agar user harus login manual
      await authService.signOut();
      if (mounted) {
        ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(
            content: Text('Registrasi berhasil! Silakan login.'),
            backgroundColor: Colors.green,
          ),
        );
        // Alihkan ke halaman login
        Navigator.of(context).pushReplacementNamed('/login');
      }
    } catch (e) {
      if (mounted) {
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
            content: Text('Registrasi gagal: ${e.toString()}'),
            backgroundColor: Colors.red,
          ),
        );
      }
    } finally {
      if (mounted) {
        setState(() => _isLoading = false);
      }
    }
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Daftar Akun'),
      backgroundColor: Colors.transparent,
      elevation: 0,
    ),
    body: Container(
      decoration: BoxDecoration(
        gradient: LinearGradient(
          begin: Alignment.topLeft,
          end: Alignment.bottomRight,
          colors: [
            Colors.purple.shade700,
```

```
            Colors.blue.shade700,
          ],
        ),
      ),
      child: SafeArea(
        child: Center(
          child: SingleChildScrollView(
            padding: const EdgeInsets.all(24.0),
            child: Form(
              key: _formKey,
              child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  const Text(
                    'Buat Akun Baru',
                    style: TextStyle(
                      fontSize: 28,
                      fontWeight: FontWeight.bold,
                      color: Colors.white,
                    ),
                  ),
                  const SizedBox(height: 40),
                  // Email Field
                  _buildModernTextField(
                    controller: _emailController,
                    label: 'Email',
                    hint: 'contoh@email.com',
                    icon: Icons.email_outlined,
                    iconColor: Colors.blue.shade700,
                    keyboardType: TextInputType.emailAddress,
                    validator: (value) {
                      if (value == null || value.isEmpty) {
                        return 'Email tidak boleh kosong';
                      }
                      if (!value.contains('@')) {
                        return 'Email tidak valid';
                      }
                      return null;
                    },
                  ),
                  const SizedBox(height: 20),
                  // Password Field
                  _buildModernTextField(
                    controller: _passwordController,
                    label: 'Password',
                    hint: 'Minimal 6 karakter',
                    icon: Icons.lock_outlined,
                    iconColor: Colors.purple.shade700,
                    obscureText: _obscurePassword,
                    suffixIcon: IconButton(
                      icon: Icon(
                        _obscurePassword
                            ? Icons.visibility_outlined
                            : Icons.visibility_off_outlined,
                        color: Colors.grey.shade600,
                      ),
                      onPressed: () {
                        setState(() {
                          _obscurePassword = !_obscurePassword;
                        });
                      },
```

```
    ),
    validator: (value) {
      if (value == null || value.isEmpty) {
        return 'Password tidak boleh kosong';
      }
      if (value.length < 6) {
        return 'Password minimal 6 karakter';
      }
      return null;
    },
  ),
  const SizedBox(height: 20),
  // Confirm Password Field
  _buildModernTextField(
    controller: _confirmPasswordController,
    label: 'Konfirmasi Password',
    hint: 'Ulangi password Anda',
    icon: Icons.lock_outline,
    iconColor: Colors.orange.shade700,
    obscureText: _obscureConfirmPassword,
    suffixIcon: IconButton(
      icon: Icon(
        _obscureConfirmPassword
            ? Icons.visibility_outlined
            : Icons.visibility_off_outlined,
        color: Colors.grey.shade600,
      ),
      onPressed: () {
        setState(() {
          _obscureConfirmPassword = !_obscureConfirmPassword;
        });
      },
    ),
    validator: (value) {
      if (value == null || value.isEmpty) {
        return 'Konfirmasi password tidak boleh kosong';
      }
      if (value != _passwordController.text) {
        return 'Password tidak cocok';
      }
      return null;
    },
  ),
  const SizedBox(height: 30),
  // Register Button
  SizedBox(
    width: double.infinity,
    height: 50,
    child: ElevatedButton(
      onPressed: _isLoading ? null : _handleRegister,
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.white,
        foregroundColor: Colors.purple.shade700,
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(12),
        ),
        elevation: 5,
      ),
      child: _isLoading
          ? const SizedBox(
              height: 20,
```

```
                                width: 20,
                                child: CircularProgressIndicator(strokeWidth: 2),
                              )
                            : const Text(
                                'Daftar',
                                style: TextStyle(
                                  fontSize: 18,
                                  fontWeight: FontWeight.bold,
                                ),
                              ),
                      ),
                    ),
                  ],
                ),
              ),
            ),
          ),
        ),
      ),
    );
  }

  Widget _buildModernTextField({
    required TextEditingController controller,
    required String label,
    required String hint,
    required IconData icon,
    required Color iconColor,
    TextInputType? keyboardType,
    bool obscureText = false,
    Widget? suffixIcon,
    String? Function(String?)? validator,
  }) {
    return Container(
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(16),
        boxShadow: [
          BoxShadow(
            color: Colors.black.withOpacity(0.08),
            blurRadius: 15,
            offset: const Offset(0, 5),
          ),
        ],
      ),
      child: TextFormField(
        controller: controller,
        keyboardType: keyboardType,
        obscureText: obscureText,
        style: const TextStyle(
          fontSize: 16,
          fontWeight: FontWeight.w500,
        ),
        decoration: InputDecoration(
          labelText: label,
          hintText: hint,
          prefixIcon: Container(
            margin: const EdgeInsets.all(12),
            padding: const EdgeInsets.all(8),
            decoration: BoxDecoration(
              color: iconColor.withOpacity(0.1),
              borderRadius: BorderRadius.circular(10),
```

```
            ),
            child: Icon(
              icon,
              color: iconColor,
              size: 20,
            ),
          ),
          suffixIcon: suffixIcon,
          filled: true,
          fillColor: Colors.white,
          border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(16),
            borderSide: BorderSide.none,
          ),
          enabledBorder: OutlineInputBorder(
            borderRadius: BorderRadius.circular(16),
            borderSide: BorderSide(
              color: Colors.grey.shade200,
              width: 1.5,
            ),
          ),
          focusedBorder: OutlineInputBorder(
            borderRadius: BorderRadius.circular(16),
            borderSide: BorderSide(
              color: iconColor,
              width: 2,
            ),
          ),
          errorBorder: OutlineInputBorder(
            borderRadius: BorderRadius.circular(16),
            borderSide: BorderSide(
              color: Colors.red.shade300,
              width: 1.5,
            ),
          ),
          focusedErrorBorder: OutlineInputBorder(
            borderRadius: BorderRadius.circular(16),
            borderSide: BorderSide(
              color: Colors.red.shade700,
              width: 2,
            ),
          ),
          labelStyle: TextStyle(
            color: Colors.grey.shade600,
            fontSize: 16,
          ),
          floatingLabelStyle: TextStyle(
            color: iconColor,
            fontWeight: FontWeight.w600,
          ),
          contentPadding: const EdgeInsets.symmetric(
            horizontal: 20,
            vertical: 18,
          ),
        ),
        validator: validator,
      ),
    );
  }
}
```

2.3 Lib\screen\berita

1. berita_list_screen.dart

File berita_list_screen.dart berfungsi untuk menampilkan daftar berita terkini dengan mengambil data dari API. Halaman ini mendukung fitur loading, error handling, refresh, tampilan kartu berita, serta detail berita dalam bentuk modal. Selain itu, terdapat fungsi pemformatan waktu agar lebih mudah dipahami oleh pengguna. Halaman daftar berita

- Mengambil data berita dari API
- Menampikan berita dalam bentuk list kartu
- Menyediakan refres dan detail berita saat diklik

```dart
import 'package:flutter/material.dart';
import '../../models/berita.dart';
import '../../services/news_api_service.dart';

class BeritaListScreen extends StatefulWidget {
  const BeritaListScreen({super.key});

  @override
  State<BeritaListScreen> createState() => _BeritaListScreenState();
}

class _BeritaListScreenState extends State<BeritaListScreen> {
  final NewsApiService _newsApiService = NewsApiService();
  List<Berita> _beritaList = [];
  bool _isLoading = true;
  String? _errorMessage;

  @override
  void initState() {
    super.initState();
    _loadBerita();
  }

  Future<void> _loadBerita() async {
    setState(() {
      _isLoading = true;
      _errorMessage = null;
    });

    try {
      // Menggunakan Indonesia saja
      final berita = await _newsApiService.getNews(
        country: 'id',
        category: null,
      );

      setState(() {
        _beritaList = berita;
        _isLoading = false;
      });
    } catch (e) {
```

```
      setState(() {
        _errorMessage = e.toString();
        _isLoading = false;
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        decoration: BoxDecoration(
          gradient: LinearGradient(
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
            colors: [
              Colors.blue.shade50,
              Colors.white,
            ],
          ),
        ),
        child: SafeArea(
          child: Column(
            children: [
              // Header dengan gradient
              Container(
                padding: const EdgeInsets.symmetric(horizontal: 20, vertical:
16),
                decoration: BoxDecoration(
                  gradient: LinearGradient(
                    colors: [
                      Colors.blue.shade700,
                      Colors.purple.shade700,
                    ],
                  ),
                  boxShadow: [
                    BoxShadow(
                      color: Colors.black.withOpacity(0.1),
                      blurRadius: 10,
                      offset: const Offset(0, 4),
                    ),
                  ],
                ),
                child: Row(
                  children: [
                    Container(
                      padding: const EdgeInsets.all(10),
                      decoration: BoxDecoration(
                        color: Colors.white.withOpacity(0.2),
                        borderRadius: BorderRadius.circular(12),
                      ),
                      child: const Icon(
                        Icons.newspaper,
                        color: Colors.white,
                        size: 28,
                      ),
                    ),
                    const SizedBox(width: 16),
                    const Expanded(
                      child: Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
```

```
                    children: [
                      Text(
                        'Portal Berita',
                        style: TextStyle(
                          fontSize: 24,
                          fontWeight: FontWeight.bold,
                          color: Colors.white,
                        ),
                      ),
                      Text(
                        'Berita Terkini Indonesia',
                        style: TextStyle(
                          fontSize: 14,
                          color: Colors.white70,
                        ),
                      ),
                    ],
                  ),
                ),
                IconButton(
                  icon: const Icon(Icons.refresh, color: Colors.white),
                  onPressed: _loadBerita,
                  tooltip: 'Refresh',
                ),
              ],
            ),
          ),
          // News List
          Expanded(
            child: _isLoading
                ? Center(
                    child: Column(
                      mainAxisAlignment: MainAxisAlignment.center,
                      children: [
                        CircularProgressIndicator(
                          valueColor: AlwaysStoppedAnimation<Color>(
                            Colors.blue.shade700,
                          ),
                        ),
                        const SizedBox(height: 16),
                        Text(
                          'Memuat berita...',
                          style: TextStyle(
                            color: Colors.grey.shade600,
                            fontSize: 16,
                          ),
                        ),
                      ],
                    ),
                  )
                : _errorMessage != null
                    ? Center(
                        child: Padding(
                          padding: const EdgeInsets.all(32),
                          child: Column(
                            mainAxisAlignment: MainAxisAlignment.center,
                            children: [
                              Container(
                                padding: const EdgeInsets.all(20),
                                decoration: BoxDecoration(
                                  color: Colors.red.shade50,
```

```
                                shape: BoxShape.circle,
                              ),
                              child: Icon(
                                Icons.error_outline,
                                size: 64,
                                color: Colors.red.shade300,
                              ),
                            ),
                            const SizedBox(height: 24),
                            Text(
                              'Gagal Memuat Berita',
                              style: TextStyle(
                                fontSize: 20,
                                fontWeight: FontWeight.bold,
                                color: Colors.grey.shade800,
                              ),
                            ),
                            const SizedBox(height: 12),
                            Text(
                              _errorMessage!,
                              textAlign: TextAlign.center,
                              style: TextStyle(
                                color: Colors.grey.shade600,
                                fontSize: 14,
                              ),
                            ),
                            const SizedBox(height: 32),
                            ElevatedButton.icon(
                              onPressed: _loadBerita,
                              icon: const Icon(Icons.refresh),
                              label: const Text('Coba Lagi'),
                              style: ElevatedButton.styleFrom(
                                backgroundColor: Colors.blue.shade700,
                                foregroundColor: Colors.white,
                                padding: const EdgeInsets.symmetric(
                                  horizontal: 24,
                                  vertical: 12,
                                ),
                                shape: RoundedRectangleBorder(
                                  borderRadius: BorderRadius.circular(12),
                                ),
                              ),
                            ),
                          ],
                        ),
                      ),
                    )
                  : _beritaList.isEmpty
                      ? Center(
                          child: Column(
                            mainAxisAlignment: MainAxisAlignment.center,
                            children: [
                              Icon(
                                Icons.newspaper_outlined,
                                size: 80,
                                color: Colors.grey.shade300,
                              ),
                              const SizedBox(height: 16),
                              Text(
                                'Tidak ada berita',
                                style: TextStyle(
```

```
                                    color: Colors.grey.shade600,
                                    fontSize: 16,
                                  ),
                                ),
                              ],
                            ),
                          )
                        : RefreshIndicator(
                            onRefresh: _loadBerita,
                            color: Colors.blue.shade700,
                            child: ListView.builder(
                              padding: const EdgeInsets.all(16),
                              itemCount: _beritaList.length,
                              itemBuilder: (context, index) {
                                final berita = _beritaList[index];
                                return _buildNewsCard(berita);
                              },
                            ),
                          ),
                ),
              ],
            ),
          ),
        ),
      );
    }

    Widget _buildNewsCard(Berita berita) {
      return Container(
        margin: const EdgeInsets.only(bottom: 20),
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(16),
          boxShadow: [
            BoxShadow(
              color: Colors.black.withOpacity(0.08),
              blurRadius: 10,
              offset: const Offset(0, 4),
            ),
          ],
        ),
        child: Material(
          color: Colors.white,
          borderRadius: BorderRadius.circular(16),
          child: InkWell(
            onTap: () => _showBeritaDetail(berita),
            borderRadius: BorderRadius.circular(16),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                // Image dengan overlay gradient
                if (berita.gambarUrl != null)
                  Stack(
                    children: [
                      ClipRRect(
                        borderRadius: const BorderRadius.vertical(
                          top: Radius.circular(16),
                        ),
                        child: Image.network(
                          berita.gambarUrl!,
                          width: double.infinity,
                          height: 220,
```

```
                    fit: BoxFit.cover,
                    errorBuilder: (context, error, stackTrace) {
                      return Container(
                        height: 220,
                        decoration: BoxDecoration(
                          gradient: LinearGradient(
                            colors: [
                              Colors.blue.shade100,
                              Colors.purple.shade100,
                            ],
                          ),
                        ),
                        child: Center(
                          child: Icon(
                            Icons.image_not_supported,
                            size: 56,
                            color: Colors.grey.shade400,
                          ),
                        ),
                      );
                    },
                    loadingBuilder: (context, child, loadingProgress) {
                      if (loadingProgress == null) return child;
                      return Container(
                        height: 220,
                        decoration: BoxDecoration(
                          gradient: LinearGradient(
                            colors: [
                              Colors.blue.shade50,
                              Colors.purple.shade50,
                            ],
                          ),
                        ),
                        child: Center(
                          child: CircularProgressIndicator(
                            value: loadingProgress.expectedTotalBytes != null
                                ? loadingProgress.cumulativeBytesLoaded /
                                    loadingProgress.expectedTotalBytes!
                                : null,
                            valueColor: AlwaysStoppedAnimation<Color>(
                              Colors.blue.shade700,
                            ),
                          ),
                        ),
                      );
                    },
                  ),
                ),
                // Gradient overlay
                Positioned(
                  bottom: 0,
                  left: 0,
                  right: 0,
                  child: Container(
                    height: 80,
                    decoration: BoxDecoration(
                      gradient: LinearGradient(
                        begin: Alignment.topCenter,
                        end: Alignment.bottomCenter,
                        colors: [
                          Colors.transparent,
```

```
                    Colors.black.withOpacity(0.3),
                  ],
                ),
              ),
            ),
          ),
        ],
      ),
      // Content
      Padding(
        padding: const EdgeInsets.all(20),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            // Title
            Text(
              berita.judul,
              style: TextStyle(
                fontSize: 20,
                fontWeight: FontWeight.bold,
                color: Colors.grey.shade900,
                height: 1.3,
              ),
              maxLines: 2,
              overflow: TextOverflow.ellipsis,
            ),
            const SizedBox(height: 12),
            // Description
            Text(
              berita.konten,
              style: TextStyle(
                fontSize: 14,
                color: Colors.grey.shade700,
                height: 1.5,
              ),
              maxLines: 3,
              overflow: TextOverflow.ellipsis,
            ),
            const SizedBox(height: 16),
            // Footer info
            Row(
              children: [
                Container(
                  padding: const EdgeInsets.symmetric(
                    horizontal: 10,
                    vertical: 6,
                  ),
                  decoration: BoxDecoration(
                    color: Colors.blue.shade50,
                    borderRadius: BorderRadius.circular(8),
                  ),
                  child: Row(
                    mainAxisSize: MainAxisSize.min,
                    children: [
                      Icon(
                        Icons.person_outline,
                        size: 14,
                        color: Colors.blue.shade700,
                      ),
                      const SizedBox(width: 6),
                      Flexible(
```

```
                              child: Text(
                                berita.penulis,
                                style: TextStyle(
                                  fontSize: 12,
                                  fontWeight: FontWeight.w500,
                                  color: Colors.blue.shade700,
                                ),
                                maxLines: 1,
                                overflow: TextOverflow.ellipsis,
                              ),
                            ),
                          ],
                        ),
                      ),
                      const Spacer(),
                      Container(
                        padding: const EdgeInsets.symmetric(
                          horizontal: 10,
                          vertical: 6,
                        ),
                        decoration: BoxDecoration(
                          color: Colors.purple.shade50,
                          borderRadius: BorderRadius.circular(8),
                        ),
                        child: Row(
                          mainAxisSize: MainAxisSize.min,
                          children: [
                            Icon(
                              Icons.access_time,
                              size: 14,
                              color: Colors.purple.shade700,
                            ),
                            const SizedBox(width: 6),
                            Text(
                              _formatDate(berita.tanggal),
                              style: TextStyle(
                                fontSize: 12,
                                fontWeight: FontWeight.w500,
                                color: Colors.purple.shade700,
                              ),
                            ),
                          ],
                        ),
                      ),
                    ],
                  ),
                ),
              ],
            ),
          ),
        ),
      ),
    );
  }

  void _showBeritaDetail(Berita berita) {
    showModalBottomSheet(
      context: context,
      isScrollControlled: true,
      backgroundColor: Colors.transparent,
```

```
    builder: (context) => Container(
      height: MediaQuery.of(context).size.height * 0.95,
      decoration: BoxDecoration(
        color: Colors.white,
        borderRadius: const BorderRadius.vertical(
          top: Radius.circular(25),
        ),
        boxShadow: [
          BoxShadow(
            color: Colors.black.withOpacity(0.2),
            blurRadius: 20,
            offset: const Offset(0, -5),
          ),
        ],
      ),
      child: Column(
        children: [
          // Handle bar
          Container(
            margin: const EdgeInsets.only(top: 12),
            width: 40,
            height: 4,
            decoration: BoxDecoration(
              color: Colors.grey.shade300,
              borderRadius: BorderRadius.circular(2),
            ),
          ),
          // Content
          Expanded(
            child: SingleChildScrollView(
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  // Image
                  if (berita.gambarUrl != null)
                    ClipRRect(
                      borderRadius: const BorderRadius.vertical(
                        top: Radius.circular(25),
                      ),
                      child: Stack(
                        children: [
                          Image.network(
                            berita.gambarUrl!,
                            width: double.infinity,
                            height: 280,
                            fit: BoxFit.cover,
                            errorBuilder: (context, error, stackTrace) {
                              return Container(
                                height: 280,
                                decoration: BoxDecoration(
                                  gradient: LinearGradient(
                                    colors: [
                                      Colors.blue.shade100,
                                      Colors.purple.shade100,
                                    ],
                                  ),
                                ),
                                child: Center(
                                  child: Icon(
                                    Icons.image_not_supported,
                                    size: 64,
```

```
                        color: Colors.grey.shade400,
                      ),
                    ),
                  );
                },
              ),
              // Gradient overlay
              Positioned(
                bottom: 0,
                left: 0,
                right: 0,
                child: Container(
                  height: 100,
                  decoration: BoxDecoration(
                    gradient: LinearGradient(
                      begin: Alignment.topCenter,
                      end: Alignment.bottomCenter,
                      colors: [
                        Colors.transparent,
                        Colors.black.withOpacity(0.5),
                      ],
                    ),
                  ),
                ),
              ),
            ],
          ),
        ),
        // Content section
        Padding(
          padding: const EdgeInsets.all(24),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              // Title
              Text(
                berita.judul,
                style: TextStyle(
                  fontSize: 26,
                  fontWeight: FontWeight.bold,
                  color: Colors.grey.shade900,
                  height: 1.3,
                ),
              ),
              const SizedBox(height: 20),
              // Author and date info
              Row(
                children: [
                  Expanded(
                    child: Container(
                      padding: const EdgeInsets.all(12),
                      decoration: BoxDecoration(
                        color: Colors.blue.shade50,
                        borderRadius: BorderRadius.circular(12),
                      ),
                      child: Row(
                        children: [
                          Icon(
                            Icons.person,
                            size: 18,
                            color: Colors.blue.shade700,
```

```
                    ),
                    const SizedBox(width: 8),
                    Expanded(
                      child: Text(
                        berita.penulis,
                        style: TextStyle(
                          fontSize: 14,
                          fontWeight: FontWeight.w600,
                          color: Colors.blue.shade700,
                        ),
                        maxLines: 1,
                        overflow: TextOverflow.ellipsis,
                      ),
                    ),
                  ],
                ),
              ),
            ),
            const SizedBox(width: 12),
            Container(
              padding: const EdgeInsets.all(12),
              decoration: BoxDecoration(
                color: Colors.purple.shade50,
                borderRadius: BorderRadius.circular(12),
              ),
              child: Row(
                mainAxisSize: MainAxisSize.min,
                children: [
                  Icon(
                    Icons.access_time,
                    size: 18,
                    color: Colors.purple.shade700,
                  ),
                  const SizedBox(width: 8),
                  Text(
                    _formatDate(berita.tanggal),
                    style: TextStyle(
                      fontSize: 14,
                      fontWeight: FontWeight.w600,
                      color: Colors.purple.shade700,
                    ),
                  ),
                ],
              ),
            ),
          ],
        ),
        const SizedBox(height: 24),
        // Divider
        Divider(
          color: Colors.grey.shade200,
          thickness: 1,
        ),
        const SizedBox(height: 24),
        // Content
        Text(
          berita.konten,
          style: TextStyle(
            fontSize: 16,
            height: 1.8,
            color: Colors.grey.shade800,
```

```
                            letterSpacing: 0.3,
                          ),
                        ),
                        const SizedBox(height: 32),
                      ],
                    ),
                  ),
                ],
              ),
            ),
          ),
          // Close button
          Container(
            padding: const EdgeInsets.all(20),
            decoration: BoxDecoration(
              color: Colors.white,
              boxShadow: [
                BoxShadow(
                  color: Colors.black.withOpacity(0.05),
                  blurRadius: 10,
                  offset: const Offset(0, -2),
                ),
              ],
            ),
            child: SizedBox(
              width: double.infinity,
              child: ElevatedButton(
                onPressed: () => Navigator.pop(context),
                style: ElevatedButton.styleFrom(
                  backgroundColor: Colors.blue.shade700,
                  foregroundColor: Colors.white,
                  padding: const EdgeInsets.symmetric(vertical: 16),
                  shape: RoundedRectangleBorder(
                    borderRadius: BorderRadius.circular(12),
                  ),
                  elevation: 0,
                ),
                child: const Text(
                  'Tutup',
                  style: TextStyle(
                    fontSize: 16,
                    fontWeight: FontWeight.bold,
                  ),
                ),
              ),
            ),
          ),
        ],
      ),
    ),
  );
}

String _formatDate(DateTime date) {
  final now = DateTime.now();
  final difference = now.difference(date);

  if (difference.inDays == 0) {
    if (difference.inHours == 0) {
      if (difference.inMinutes == 0) {
        return 'Baru saja';
```

```
        }
        return '${difference.inMinutes} menit lalu';
      }
      return '${difference.inHours} jam lalu';
    } else if (difference.inDays == 1) {
      return 'Kemarin';
    } else if (difference.inDays < 7) {
      return '${difference.inDays} hari lalu';
    } else {
      return '${date.day}/${date.month}/${date.year}';
    }
  }
}
```

2.4 Lib\screen\home

1.Home_screen.dart

file home_screen.dart berfungsi sebagai halaman utama aplikasi portal berita yang menampilkan informasi pengguna secara real-time melalui Firebase dan menyediakan menu navigasi utama seperti daftar berita, halaman profil, dan logout. Implementasi ini menggunakan Provider untuk manajemen state dan integrasi Firebase Authentication. Dihalaman ini pengguna bisa:

- Melihat nama profil
- Mengaskes daftar berita
- Membuka halaman profil
- Logout dari aplikasi

```
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:firebase_auth/firebase_auth.dart';
import '../../services/auth_service.dart';
import '../../services/profile_service.dart';
import '../berita/berita_list_screen.dart';
import '../profile/profile_screen.dart';

class HomeScreen extends StatefulWidget {
  const HomeScreen({super.key});

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  @override
  Widget build(BuildContext context) {
    final authService = Provider.of<AuthService>(context);
    final profileService = Provider.of<ProfileService>(context);
    final user = authService.currentUser;

    return Scaffold(
      body: Container(
        decoration: BoxDecoration(
          gradient: LinearGradient(
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
            colors: [
```

```
            Colors.blue.shade50,
            Colors.purple.shade50,
          ],
        ),
      ),
      child: SafeArea(
        child: Column(
          children: [
            // Header
            Container(
              padding: const EdgeInsets.all(20),
              decoration: BoxDecoration(
                gradient: LinearGradient(
                  colors: [
                    Colors.blue.shade700,
                    Colors.purple.shade700,
                  ],
                ),
                borderRadius: const BorderRadius.only(
                  bottomLeft: Radius.circular(30),
                  bottomRight: Radius.circular(30),
                ),
              ),
              child: Row(
                children: [
                  CircleAvatar(
                    radius: 30,
                    backgroundColor: Colors.white,
                    child: Icon(
                      Icons.person,
                      size: 30,
                      color: Colors.blue.shade700,
                    ),
                  ),
                  const SizedBox(width: 15),
                  Expanded(
                    child: Column(
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
                        const Text(
                          'Selamat Datang,',
                          style: TextStyle(
                            color: Colors.white70,
                            fontSize: 14,
                          ),
                        ),
                        // Stream untuk listen perubahan profil
                        if (user != null)
                          StreamBuilder(
                            stream: profileService.getProfileStream(user.uid),
                            builder: (context, snapshot) {
                              String displayName = user.displayName ??
                                                   user.email ??
                                                   'Pengguna';

                              if (snapshot.hasData && snapshot.data != null) {
                                final profile = snapshot.data!;
                                displayName = profile.displayName ??
                                              user.displayName ??
                                              user.email ??
                                              'Pengguna';
```

```
                    }

                    return Text(
                      displayName,
                      style: const TextStyle(
                        color: Colors.white,
                        fontSize: 20,
                        fontWeight: FontWeight.bold,
                      ),
                      maxLines: 1,
                      overflow: TextOverflow.ellipsis,
                    );
                  },
                )
              else
                Text(
                  'Pengguna',
                  style: const TextStyle(
                    color: Colors.white,
                    fontSize: 20,
                    fontWeight: FontWeight.bold,
                  ),
                  maxLines: 1,
                  overflow: TextOverflow.ellipsis,
                ),
            ],
          ),
        ),
        IconButton(
          icon: const Icon(Icons.person, color: Colors.white),
          onPressed: () async {
            final result = await Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => const ProfileScreen(),
              ),
            );
            // Refresh jika profil diupdate
            if (result == true && mounted) {
              setState(() {});
            }
          },
        ),
      ],
    ),
  ),
  // Content
  Expanded(
    child: Padding(
      padding: const EdgeInsets.all(20),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          const Text(
            'Menu Utama',
            style: TextStyle(
              fontSize: 24,
              fontWeight: FontWeight.bold,
            ),
          ),
          const SizedBox(height: 20),
```

```
                    Expanded(
                      child: GridView.count(
                        crossAxisCount: 2,
                        crossAxisSpacing: 15,
                        mainAxisSpacing: 15,
                        children: [
                          _buildMenuCard(
                            context,
                            icon: Icons.newspaper,
                            title: 'Daftar Berita',
                            color: Colors.blue,
                            onTap: () {
                              Navigator.push(
                                context,
                                MaterialPageRoute(
                                  builder: (context) =>
                                      const BeritaListScreen(),
                                ),
                              );
                            },
                          ),
                          _buildMenuCard(
                            context,
                            icon: Icons.person,
                            title: 'Profil',
                            color: Colors.purple,
                            onTap: () {
                              Navigator.push(
                                context,
                                MaterialPageRoute(
                                  builder: (context) => const ProfileScreen(),
                                ),
                              );
                            },
                          ),
                          _buildMenuCard(
                            context,
                            icon: Icons.logout,
                            title: 'Logout',
                            color: Colors.red,
                            onTap: () async {
                              final confirm = await showDialog<bool>(
                                context: context,
                                builder: (context) => AlertDialog(
                                  title: const Text('Logout'),
                                  content: const Text(
                                    'Apakah Anda yakin ingin logout?',
                                  ),
                                  actions: [
                                    TextButton(
                                      onPressed: () =>
                                          Navigator.pop(context, false),
                                      child: const Text('Batal'),
                                    ),
                                    TextButton(
                                      onPressed: () =>
                                          Navigator.pop(context, true),
                                      child: const Text('Logout'),
                                    ),
                                  ],
                                ),
```

```
                                );
                                if (confirm == true) {
                                  await authService.signOut();
                                  if (context.mounted) {
                                    Navigator.of(context)
                                        .pushReplacementNamed('/login');
                                  }
                                }
                              },
                            ),
                          ],
                        ),
                      ),
                    ),
                  ),
                ),
              ],
            ),
          ),
        ),
      ),
    );
  }

  Widget _buildMenuCard(
    BuildContext context, {
    required IconData icon,
    required String title,
    required MaterialColor color,
    required VoidCallback onTap,
  }) {
    return Card(
      elevation: 5,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(15),
      ),
      child: InkWell(
        onTap: onTap,
        borderRadius: BorderRadius.circular(15),
        child: Container(
          decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(15),
            gradient: LinearGradient(
              begin: Alignment.topLeft,
              end: Alignment.bottomRight,
              colors: [
                color.shade400,
                color.shade700,
              ],
            ),
          ),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Icon(
                icon,
                size: 50,
                color: Colors.white,
              ),
              const SizedBox(height: 10),
              Text(
```

```
                title,
                style: const TextStyle(
                  color: Colors.white,
                  fontSize: 16,
                  fontWeight: FontWeight.bold,
                ),
                textAlign: TextAlign.center,
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

2.5 Lib\screen\profile

1. Profil_srcreen.dart

File ini berfungsi sebagai halaman profil pengguna dalam aplikasi Flutter. Fungsinya meliputi:

a. Menampilkan data profil user
- Nama (display name)
- Email (hanya baca / read-only)
- Bio
- Nomor telepon
- Foto profil

b. Mengambil dan menyimpan data

c. Edit profil

d. Update data

e. hapus akun

Halaman ini menghubungkan data profil pengguna dengan Firebase Authentication dan Firestore sehingga data tetap tersimpan dan bisa diperbarui secara real-time.

```
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:firebase_auth/firebase_auth.dart';
import '../../services/auth_service.dart';
import '../../services/profile_service.dart';
import '../../models/user_profile.dart';

class ProfileScreen extends StatefulWidget {
  const ProfileScreen({super.key});

  @override
  State<ProfileScreen> createState() => _ProfileScreenState();
}

class _ProfileScreenState extends State<ProfileScreen> {
  final _formKey = GlobalKey<FormState>();
  final _displayNameController = TextEditingController();
```

```dart
final _bioController = TextEditingController();
final _phoneController = TextEditingController();
bool _isEditing = false;
bool _isLoading = false;
UserProfile? _userProfile;

@override
void initState() {
  super.initState();
  _loadUserData();
}

Future<void> _loadUserData() async {
  final authService = Provider.of<AuthService>(context, listen: false);
  final profileService = Provider.of<ProfileService>(context, listen: false);
  final user = authService.currentUser;

  if (user != null) {
    setState(() => _isLoading = true);
    try {
      // Coba ambil dari Firestore
      _userProfile = await profileService.getProfile(user.uid);

      if (_userProfile != null) {
        // Data dari Firestore
        _displayNameController.text = _userProfile!.displayName ?? '';
        _bioController.text = _userProfile!.bio ?? '';
        _phoneController.text = _userProfile!.phoneNumber ?? '';
      } else {
        // Jika belum ada di Firestore, gunakan data dari Auth
        _displayNameController.text = user.displayName ?? '';
        _bioController.text = '';
        _phoneController.text = '';

        // Buat profil di Firestore
        _userProfile = UserProfile(
          uid: user.uid,
          email: user.email ?? '',
          displayName: user.displayName,
          photoURL: user.photoURL,
        );
        await profileService.createProfile(_userProfile!);
      }
    } catch (e) {
      if (mounted) {
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
            content: Text('Gagal memuat profil: ${e.toString()}'),
            backgroundColor: Colors.orange,
          ),
        );
      }
      // Fallback ke data Auth
      _displayNameController.text = user.displayName ?? '';
      _bioController.text = '';
      _phoneController.text = '';
    } finally {
      if (mounted) {
        setState(() => _isLoading = false);
      }
    }
```

```
    }
  }

  @override
  void dispose() {
    _displayNameController.dispose();
    _bioController.dispose();
    _phoneController.dispose();
    super.dispose();
  }

  Future<void> _saveProfile() async {
    if (_formKey.currentState!.validate()) {
      setState(() => _isLoading = true);
      try {
        final authService = Provider.of<AuthService>(context, listen: false);
        final profileService = Provider.of<ProfileService>(context, listen:
false);
        final user = authService.currentUser;

        if (user != null) {
          // Update di Firebase Auth
          await authService.updateUserProfile(
            displayName: _displayNameController.text.trim(),
          );

          // Update di Firestore
          await profileService.updateProfile(user.uid, {
            'displayName': _displayNameController.text.trim(),
            'bio': _bioController.text.trim(),
            'phoneNumber': _phoneController.text.trim(),
          });

          // Reload user data
          await _loadUserData();

          // Reload user di Firebase Auth untuk update displayName
          await user.reload();

          setState(() => _isEditing = false);
          if (mounted) {
            ScaffoldMessenger.of(context).showSnackBar(
              const SnackBar(
                content: Text('Profil berhasil diperbarui!'),
                backgroundColor: Colors.green,
              ),
            );
          }
        }
      } catch (e) {
        if (mounted) {
          ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(
              content: Text('Gagal memperbarui profil: ${e.toString()}'),
              backgroundColor: Colors.red,
            ),
          );
        }
      } finally {
        if (mounted) {
          setState(() => _isLoading = false);
```

```
        }
      }
    }
  }

  Future<void> _deleteAccount() async {
    final confirm = await showDialog<bool>(
      context: context,
      builder: (context) => AlertDialog(
        title: const Text('Hapus Akun'),
        content: const Text(
          'Apakah Anda yakin ingin menghapus akun? Tindakan ini tidak dapat
dibatalkan.',
        ),
        actions: [
          TextButton(
            onPressed: () => Navigator.pop(context, false),
            child: const Text('Batal'),
          ),
          TextButton(
            onPressed: () => Navigator.pop(context, true),
            style: TextButton.styleFrom(foregroundColor: Colors.red),
            child: const Text('Hapus'),
          ),
        ],
      ),
    );

    if (confirm == true) {
      setState(() => _isLoading = true);
      try {
        final authService = Provider.of<AuthService>(context, listen: false);
        final profileService = Provider.of<ProfileService>(context, listen:
false);
        final user = authService.currentUser;
        if (user != null) {
          // Hapus profil dari Firestore
          await profileService.deleteProfile(user.uid);
          // Hapus akun dari Firebase Auth
          await user.delete();
          await authService.signOut();
          if (mounted) {
            Navigator.of(context).pushReplacementNamed('/login');
          }
        }
      } catch (e) {
        if (mounted) {
          ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(
              content: Text('Gagal menghapus akun: ${e.toString()}'),
              backgroundColor: Colors.red,
            ),
          );
        }
      } finally {
        if (mounted) {
          setState(() => _isLoading = false);
        }
      }
    }
  }
```

```dart
@override
Widget build(BuildContext context) {
  final authService = Provider.of<AuthService>(context);
  final user = authService.currentUser;

  return Scaffold(
    appBar: AppBar(
      title: const Text('Profil'),
      backgroundColor: Colors.purple.shade700,
      foregroundColor: Colors.white,
      actions: [
        if (!_isEditing)
          IconButton(
            icon: const Icon(Icons.edit),
            onPressed: () {
              setState(() => _isEditing = true);
            },
          ),
      ],
    ),
    body: _isLoading
        ? const Center(child: CircularProgressIndicator())
        : Container(
            decoration: BoxDecoration(
              gradient: LinearGradient(
                begin: Alignment.topLeft,
                end: Alignment.bottomRight,
                colors: [
                  Colors.purple.shade50,
                  Colors.blue.shade50,
                ],
              ),
            ),
            child: SingleChildScrollView(
              padding: const EdgeInsets.all(20),
              child: Form(
                key: _formKey,
                child: Column(
                  children: [
                    // Profile Picture
                    Stack(
                      children: [
                        CircleAvatar(
                          radius: 60,
                          backgroundColor: Colors.purple.shade700,
                          child: user?.photoURL != null
                              ? ClipOval(
                                  child: Image.network(
                                    user!.photoURL!,
                                    width: 120,
                                    height: 120,
                                    fit: BoxFit.cover,
                                  ),
                                )
                              : Icon(
                                  Icons.person,
                                  size: 60,
                                  color: Colors.white,
                                ),
                        ),
```

```
                    if (_isEditing)
                      Positioned(
                        bottom: 0,
                        right: 0,
                        child: CircleAvatar(
                          radius: 18,
                          backgroundColor: Colors.blue,
                          child: IconButton(
                            icon: const Icon(Icons.camera_alt,
                                size: 18, color: Colors.white),
                            onPressed: () {
                              // TODO: Implement photo upload
                              ScaffoldMessenger.of(context).showSnackBar(
                                const SnackBar(
                                  content: Text(
                                      'Fitur upload foto belum tersedia'),
                                ),
                              );
                            },
                          ),
                        ),
                      ),
                  ],
                ),
                const SizedBox(height: 20),
                // Display Name
                Container(
                  decoration: BoxDecoration(
                    color: Colors.white,
                    borderRadius: BorderRadius.circular(16),
                    boxShadow: [
                      BoxShadow(
                        color: Colors.black.withOpacity(0.05),
                        blurRadius: 10,
                        offset: const Offset(0, 2),
                      ),
                    ],
                  ),
                  child: Padding(
                    padding: const EdgeInsets.all(20),
                    child: Column(
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
                        Row(
                          children: [
                            Container(
                              padding: const EdgeInsets.all(8),
                              decoration: BoxDecoration(
                                color: Colors.blue.shade50,
                                borderRadius: BorderRadius.circular(10),
                              ),
                              child: Icon(
                                Icons.person_outline,
                                color: Colors.blue.shade700,
                                size: 20,
                              ),
                            ),
                            const SizedBox(width: 12),
                            Text(
                              'Nama',
                              style: TextStyle(
```

```
                                    fontSize: 13,
                                    fontWeight: FontWeight.w600,
                                    color: Colors.grey.shade600,
                                    letterSpacing: 0.5,
                                  ),
                                ),
                              ],
                            ),
                            const SizedBox(height: 12),
                            _isEditing
                                ? Container(
                                    decoration: BoxDecoration(
                                      borderRadius: BorderRadius.circular(14),
                                      boxShadow: [
                                        BoxShadow(
                                          color:
Colors.blue.shade100.withOpacity(0.3),
                                          blurRadius: 8,
                                          offset: const Offset(0, 2),
                                        ),
                                      ],
                                    ),
                                    child: TextFormField(
                                      controller: _displayNameController,
                                      style: const TextStyle(
                                        fontSize: 16,
                                        fontWeight: FontWeight.w600,
                                        color: Colors.black87,
                                      ),
                                      decoration: InputDecoration(
                                        hintText: 'Masukkan nama lengkap',
                                        hintStyle: TextStyle(
                                          color: Colors.grey.shade400,
                                          fontWeight: FontWeight.normal,
                                        ),
                                        prefixIcon: Container(
                                          margin: const EdgeInsets.all(10),
                                          padding: const EdgeInsets.all(10),
                                          decoration: BoxDecoration(
                                            gradient: LinearGradient(
                                              colors: [
                                                Colors.blue.shade400,
                                                Colors.blue.shade600,
                                              ],
                                            ),
                                            borderRadius:
BorderRadius.circular(10),
                                          ),
                                          child: const Icon(
                                            Icons.person,
                                            color: Colors.white,
                                            size: 20,
                                          ),
                                        ),
                                        filled: true,
                                        fillColor: Colors.white,
                                        border: OutlineInputBorder(
                                          borderRadius:
BorderRadius.circular(14),
                                          borderSide: BorderSide.none,
                                        ),
```

```
                                        enabledBorder: OutlineInputBorder(
                                          borderRadius:
BorderRadius.circular(14),
                                          borderSide: BorderSide(
                                            color: Colors.grey.shade200,
                                            width: 1.5,
                                          ),
                                        ),
                                        focusedBorder: OutlineInputBorder(
                                          borderRadius:
BorderRadius.circular(14),
                                          borderSide: BorderSide(
                                            color: Colors.blue.shade700,
                                            width: 2.5,
                                          ),
                                        ),
                                        errorBorder: OutlineInputBorder(
                                          borderRadius:
BorderRadius.circular(14),
                                          borderSide: BorderSide(
                                            color: Colors.red.shade300,
                                            width: 1.5,
                                          ),
                                        ),
                                        focusedErrorBorder: OutlineInputBorder(
                                          borderRadius:
BorderRadius.circular(14),
                                          borderSide: BorderSide(
                                            color: Colors.red.shade700,
                                            width: 2.5,
                                          ),
                                        ),
                                        contentPadding: const
EdgeInsets.symmetric(
                                          horizontal: 20,
                                          vertical: 18,
                                        ),
                                      ),
                                      validator: (value) {
                                        if (value == null || value.isEmpty) {
                                          return 'Nama tidak boleh kosong';
                                        }
                                        return null;
                                      },
                                    ),
                                  )
                                : Container(
                                    padding: const EdgeInsets.symmetric(
                                      horizontal: 16,
                                      vertical: 12,
                                    ),
                                    decoration: BoxDecoration(
                                      color: Colors.grey.shade50,
                                      borderRadius: BorderRadius.circular(12),
                                    ),
                                    child: Text(
                                      user?.displayName ?? 'Belum diatur',
                                      style: TextStyle(
                                        fontSize: 17,
                                        fontWeight: FontWeight.bold,
                                        color: user?.displayName != null
```

```
                                ? Colors.grey.shade900
                                : Colors.grey.shade500,
                          ),
                        ),
                      ),
                  ],
                ),
              ),
            ),
            const SizedBox(height: 16),
            // Email (Read-only)
            Container(
              decoration: BoxDecoration(
                color: Colors.white,
                borderRadius: BorderRadius.circular(16),
                boxShadow: [
                  BoxShadow(
                    color: Colors.black.withOpacity(0.05),
                    blurRadius: 10,
                    offset: const Offset(0, 2),
                  ),
                ],
              ),
              child: Padding(
                padding: const EdgeInsets.all(20),
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Row(
                      children: [
                        Container(
                          padding: const EdgeInsets.all(8),
                          decoration: BoxDecoration(
                            color: Colors.orange.shade50,
                            borderRadius: BorderRadius.circular(10),
                          ),
                          child: Icon(
                            Icons.email_outlined,
                            color: Colors.orange.shade700,
                            size: 20,
                          ),
                        ),
                        const SizedBox(width: 12),
                        Text(
                          'Email',
                          style: TextStyle(
                            fontSize: 13,
                            fontWeight: FontWeight.w600,
                            color: Colors.grey.shade600,
                            letterSpacing: 0.5,
                          ),
                        ),
                      ],
                    ),
                    const SizedBox(height: 12),
                    Container(
                      padding: const EdgeInsets.symmetric(
                        horizontal: 16,
                        vertical: 12,
                      ),
                      decoration: BoxDecoration(
```

```
                    color: Colors.grey.shade50,
                    borderRadius: BorderRadius.circular(12),
                  ),
                  child: Row(
                    children: [
                      Icon(
                        Icons.lock_outline,
                        size: 16,
                        color: Colors.grey.shade400,
                      ),
                      const SizedBox(width: 8),
                      Expanded(
                        child: Text(
                          user?.email ?? '',
                          style: TextStyle(
                            fontSize: 16,
                            fontWeight: FontWeight.w600,
                            color: Colors.grey.shade800,
                          ),
                        ),
                      ),
                    ],
                  ),
                ),
              ),
            ],
          ),
        ),
      ),
      const SizedBox(height: 16),
      // Bio
      Container(
        decoration: BoxDecoration(
          color: Colors.white,
          borderRadius: BorderRadius.circular(16),
          boxShadow: [
            BoxShadow(
              color: Colors.black.withOpacity(0.05),
              blurRadius: 10,
              offset: const Offset(0, 2),
            ),
          ],
        ),
        child: Padding(
          padding: const EdgeInsets.all(20),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Row(
                children: [
                  Container(
                    padding: const EdgeInsets.all(8),
                    decoration: BoxDecoration(
                      color: Colors.purple.shade50,
                      borderRadius: BorderRadius.circular(10),
                    ),
                    child: Icon(
                      Icons.description_outlined,
                      color: Colors.purple.shade700,
                      size: 20,
                    ),
                  ),
```

```
                                const SizedBox(width: 12),
                                Text(
                                  'Bio',
                                  style: TextStyle(
                                    fontSize: 13,
                                    fontWeight: FontWeight.w600,
                                    color: Colors.grey.shade600,
                                    letterSpacing: 0.5,
                                  ),
                                ),
                              ],
                            ),
                            const SizedBox(height: 12),
                            _isEditing
                                ? Container(
                                    decoration: BoxDecoration(
                                      borderRadius: BorderRadius.circular(14),
                                      boxShadow: [
                                        BoxShadow(
                                          color:
Colors.purple.shade100.withOpacity(0.3),
                                          blurRadius: 8,
                                          offset: const Offset(0, 2),
                                        ),
                                      ],
                                    ),
                                    child: TextFormField(
                                      controller: _bioController,
                                      maxLines: 4,
                                      style: const TextStyle(
                                        fontSize: 16,
                                        fontWeight: FontWeight.w500,
                                        color: Colors.black87,
                                        height: 1.5,
                                      ),
                                      decoration: InputDecoration(
                                        hintText: 'Ceritakan tentang diri
Anda...',
                                        hintStyle: TextStyle(
                                          color: Colors.grey.shade400,
                                          fontWeight: FontWeight.normal,
                                        ),
                                        prefixIcon: Container(
                                          margin: const EdgeInsets.only(
                                            top: 12,
                                            bottom: 12,
                                            left: 10,
                                          ),
                                          padding: const EdgeInsets.all(10),
                                          decoration: BoxDecoration(
                                            gradient: LinearGradient(
                                              colors: [
                                                Colors.purple.shade400,
                                                Colors.purple.shade600,
                                              ],
                                            ),
                                            borderRadius:
BorderRadius.circular(10),
                                          ),
                                          child: const Icon(
                                            Icons.edit_note,
```

```
                                        color: Colors.white,
                                        size: 20,
                                      ),
                                    ),
                                    filled: true,
                                    fillColor: Colors.white,
                                    border: OutlineInputBorder(
                                      borderRadius:
BorderRadius.circular(14),
                                      borderSide: BorderSide.none,
                                    ),
                                    enabledBorder: OutlineInputBorder(
                                      borderRadius:
BorderRadius.circular(14),
                                      borderSide: BorderSide(
                                        color: Colors.grey.shade200,
                                        width: 1.5,
                                      ),
                                    ),
                                    focusedBorder: OutlineInputBorder(
                                      borderRadius:
BorderRadius.circular(14),
                                      borderSide: BorderSide(
                                        color: Colors.purple.shade700,
                                        width: 2.5,
                                      ),
                                    ),
                                    contentPadding: const
EdgeInsets.all(16),
                                  ),
                                ),
                              )
                            : Container(
                                padding: const EdgeInsets.symmetric(
                                  horizontal: 16,
                                  vertical: 12,
                                ),
                                decoration: BoxDecoration(
                                  color: Colors.grey.shade50,
                                  borderRadius: BorderRadius.circular(12),
                                ),
                                child: Text(
                                  _userProfile?.bio?.isEmpty ?? true
                                      ? 'Belum diatur'
                                      : (_userProfile?.bio ??
_bioController.text),
                                  style: TextStyle(
                                    fontSize: 16,
                                    fontWeight: FontWeight.w500,
                                    color: (_userProfile?.bio?.isEmpty ??
true)
                                        ? Colors.grey.shade500
                                        : Colors.grey.shade800,
                                    height: 1.5,
                                  ),
                                ),
                              ),
                        ],
                      ),
                    ),
                  ),
```

```
                    const SizedBox(height: 16),
                    // Phone Number
                    Container(
                      decoration: BoxDecoration(
                        color: Colors.white,
                        borderRadius: BorderRadius.circular(16),
                        boxShadow: [
                          BoxShadow(
                            color: Colors.black.withOpacity(0.05),
                            blurRadius: 10,
                            offset: const Offset(0, 2),
                          ),
                        ],
                      ),
                      child: Padding(
                        padding: const EdgeInsets.all(20),
                        child: Column(
                          crossAxisAlignment: CrossAxisAlignment.start,
                          children: [
                            Row(
                              children: [
                                Container(
                                  padding: const EdgeInsets.all(8),
                                  decoration: BoxDecoration(
                                    color: Colors.green.shade50,
                                    borderRadius: BorderRadius.circular(10),
                                  ),
                                  child: Icon(
                                    Icons.phone_outlined,
                                    color: Colors.green.shade700,
                                    size: 20,
                                  ),
                                ),
                                const SizedBox(width: 12),
                                Text(
                                  'Nomor Telepon',
                                  style: TextStyle(
                                    fontSize: 13,
                                    fontWeight: FontWeight.w600,
                                    color: Colors.grey.shade600,
                                    letterSpacing: 0.5,
                                  ),
                                ),
                              ],
                            ),
                            const SizedBox(height: 12),
                            _isEditing
                                ? Container(
                                    decoration: BoxDecoration(
                                      borderRadius: BorderRadius.circular(14),
                                      boxShadow: [
                                        BoxShadow(
                                          color:
Colors.green.shade100.withOpacity(0.3),
                                          blurRadius: 8,
                                          offset: const Offset(0, 2),
                                        ),
                                      ],
                                    ),
                                    child: TextFormField(
                                      controller: _phoneController,
```

```
                                    keyboardType: TextInputType.phone,
                                    style: const TextStyle(
                                      fontSize: 16,
                                      fontWeight: FontWeight.w600,
                                      color: Colors.black87,
                                    ),
                                    decoration: InputDecoration(
                                      hintText: '08xx xxxx xxxx',
                                      hintStyle: TextStyle(
                                        color: Colors.grey.shade400,
                                        fontWeight: FontWeight.normal,
                                      ),
                                      prefixIcon: Container(
                                        margin: const EdgeInsets.all(10),
                                        padding: const EdgeInsets.all(10),
                                        decoration: BoxDecoration(
                                          gradient: LinearGradient(
                                            colors: [
                                              Colors.green.shade400,
                                              Colors.green.shade600,
                                            ],
                                          ),
                                          borderRadius:
BorderRadius.circular(10),
                                        ),
                                        child: const Icon(
                                          Icons.phone,
                                          color: Colors.white,
                                          size: 20,
                                        ),
                                      ),
                                      filled: true,
                                      fillColor: Colors.white,
                                      border: OutlineInputBorder(
                                        borderRadius:
BorderRadius.circular(14),
                                        borderSide: BorderSide.none,
                                      ),
                                      enabledBorder: OutlineInputBorder(
                                        borderRadius:
BorderRadius.circular(14),
                                        borderSide: BorderSide(
                                          color: Colors.grey.shade200,
                                          width: 1.5,
                                        ),
                                      ),
                                      focusedBorder: OutlineInputBorder(
                                        borderRadius:
BorderRadius.circular(14),
                                        borderSide: BorderSide(
                                          color: Colors.green.shade700,
                                          width: 2.5,
                                        ),
                                      ),
                                      contentPadding: const
EdgeInsets.symmetric(
                                        horizontal: 20,
                                        vertical: 18,
                                      ),
                                    ),
                                  ),
```

```
                                    )
                                : Container(
                                    padding: const EdgeInsets.symmetric(
                                      horizontal: 16,
                                      vertical: 12,
                                    ),
                                    decoration: BoxDecoration(
                                      color: Colors.grey.shade50,
                                      borderRadius: BorderRadius.circular(12),
                                    ),
                                    child: Text(
                                      _userProfile?.phoneNumber?.isEmpty ??
true
                                          ? 'Belum diatur'
                                          : (_userProfile?.phoneNumber ??
_phoneController.text),
                                      style: TextStyle(
                                        fontSize: 16,
                                        fontWeight: FontWeight.w600,
                                        color:
(_userProfile?.phoneNumber?.isEmpty ?? true)
                                            ? Colors.grey.shade500
                                            : Colors.grey.shade800,
                                      ),
                                    ),
                                  ),
                            ],
                          ),
                        ),
                      ),
                      const SizedBox(height: 30),
                      // Action Buttons
                      if (_isEditing) ...[
                        SizedBox(
                          width: double.infinity,
                          height: 50,
                          child: ElevatedButton(
                            onPressed: _saveProfile,
                            style: ElevatedButton.styleFrom(
                              backgroundColor: Colors.green,
                              foregroundColor: Colors.white,
                              shape: RoundedRectangleBorder(
                                borderRadius: BorderRadius.circular(12),
                              ),
                            ),
                            child: const Text(
                              'Simpan',
                              style: TextStyle(
                                fontSize: 16,
                                fontWeight: FontWeight.bold,
                              ),
                            ),
                          ),
                        ),
                        const SizedBox(height: 12),
                        SizedBox(
                          width: double.infinity,
                          height: 50,
                          child: OutlinedButton(
                            onPressed: () {
                              setState(() {
```

```
                                _isEditing = false;
                                _loadUserData();
                              });
                            },
                            style: OutlinedButton.styleFrom(
                              shape: RoundedRectangleBorder(
                                borderRadius: BorderRadius.circular(12),
                              ),
                            ),
                            child: const Text(
                              'Batal',
                              style: TextStyle(fontSize: 16),
                            ),
                          ),
                        ),
                      ],
                      const SizedBox(height: 20),
                      // Delete Account Button
                      SizedBox(
                        width: double.infinity,
                        height: 50,
                        child: OutlinedButton(
                          onPressed: _deleteAccount,
                          style: OutlinedButton.styleFrom(
                            foregroundColor: Colors.red,
                            side: const BorderSide(color: Colors.red),
                            shape: RoundedRectangleBorder(
                              borderRadius: BorderRadius.circular(12),
                            ),
                          ),
                          child: const Text(
                            'Hapus Akun',
                            style: TextStyle(fontSize: 16),
                          ),
                        ),
                      ),
                    ),
                  ],
                ),
              ),
            ),
          ),
        );
      }
    }
```

2.6 Lib\screen\services

1. firebase_options.dart

File firebase_options.dart adalah file konfigurasi Firebase yang dibuat otomatis oleh FlutterFire CLI. File ini berfungsi sebagai penghubung antara aplikasi Flutter dan project Firebase, sehingga aplikasi bisa menggunakan layanan Firebase seperti Authentication, Firestore, Storage, dan lainnya sesuai dengan platform yang digunakan (Web atau Android). Fungsi utamanya yaitu

- Menyimpan Konfigurasi Firebase

- Menemtukan konfigurasi berdasarkan platfrom
- Menghubungkan aplikasi ke firebase
- Mencegah eror pada platfrom yang belum diatur

```dart
// File generated by FlutterFire CLI.
// ignore_for_file: type=lint
import 'package:firebase_core/firebase_core.dart' show FirebaseOptions;
import 'package:flutter/foundation.dart'
    show defaultTargetPlatform, kIsWeb, TargetPlatform;

/// Default [FirebaseOptions] for use with your Firebase apps.
///
/// Example:
/// ```dart
/// import 'firebase_options.dart';
/// // ...
/// await Firebase.initializeApp(
///   options: DefaultFirebaseOptions.currentPlatform,
/// );
/// ```
class DefaultFirebaseOptions {
  static FirebaseOptions get currentPlatform {
    if (kIsWeb) {
      return web;
    }
    switch (defaultTargetPlatform) {
      case TargetPlatform.android:
        return android;
      case TargetPlatform.iOS:
        throw UnsupportedError(
          'DefaultFirebaseOptions have not been configured for ios - '
          'you can reconfigure this by running the FlutterFire CLI again.',
        );
      case TargetPlatform.macOS:
        throw UnsupportedError(
          'DefaultFirebaseOptions have not been configured for macos - '
          'you can reconfigure this by running the FlutterFire CLI again.',
        );
      case TargetPlatform.windows:
        throw UnsupportedError(
          'DefaultFirebaseOptions have not been configured for windows - '
          'you can reconfigure this by running the FlutterFire CLI again.',
        );
      case TargetPlatform.linux:
        throw UnsupportedError(
          'DefaultFirebaseOptions have not been configured for linux - '
          'you can reconfigure this by running the FlutterFire CLI again.',
        );
      default:
        throw UnsupportedError(
          'DefaultFirebaseOptions are not supported for this platform.',
        );
    }
  }

  static const FirebaseOptions web = FirebaseOptions(
```

```
    apiKey: 'AIzaSyDOJ_sApmwiQsrRSehaPVl-roK0zipv_v0',
    appId: '1:345822162548:web:4cce3c3e72b0360345ab21',
    messagingSenderId: '345822162548',
    projectId: 'portalb-a3010',
    authDomain: 'portalb-a3010.firebaseapp.com',
    storageBucket: 'portalb-a3010.firebasestorage.app',
    measurementId: 'G-C4MRKHDY8F',
  );

  static const FirebaseOptions android = FirebaseOptions(
    apiKey: 'AIzaSyDwpSTjdzwIDNY9rP-f2elUtrlB74t3ST4',
    appId: '1:345822162548:android:5932b887f67c8e6645ab21',
    messagingSenderId: '345822162548',
    projectId: 'portalb-a3010',
    storageBucket: 'portalb-a3010.firebasestorage.app',
  );

}
```

2. Mian.dart

Kode ini adalah entry point utama aplikasi Flutter yang menginisialisasi Firebase, menyiapkan state management dengan Provider, dan mengatur navigasi otomatis berdasarkan status login pengguna. Aplikasi akan menampilkan halaman login atau halaman utama tergantung apakah user sudah login atau belum.

Fungsi uamanta yaitu:

- Mendeteksi status login pengguna secara real time
- Jika masih loading ->tampilkan indikator loading
- Jika sudah login -> tampilan Homescreen
- Jika belum login ->tampilan loginscreen

```
       import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:provider/provider.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'firebase_options.dart';
import 'services/auth_service.dart';
import 'services/profile_service.dart';
import 'screens/auth/login_screen.dart';
import 'screens/home/home_screen.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(const MainApp());
}

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

  @override
  Widget build(BuildContext context) {
```

```
    return MultiProvider(
      providers: [
        Provider<AuthService>(
          create: (_) => AuthService(),
        ),
        Provider<ProfileService>(
          create: (_) => ProfileService(),
        ),
      ],
      child: MaterialApp(
        title: 'Portal Berita',
        debugShowCheckedModeBanner: false,
        theme: ThemeData(
          colorScheme: ColorScheme.fromSeed(seedColor: Colors.blue),
          useMaterial3: true,
        ),
        home: const AuthWrapper(),
        routes: {
          '/login': (context) => const LoginScreen(),
          '/home': (context) => const HomeScreen(),
        },
      ),
    );
  }
}

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

  @override
  Widget build(BuildContext context) {
    final authService = Provider.of<AuthService>(context);

    return StreamBuilder<User?>(
      stream: authService.authStateChanges,
      builder: (context, snapshot) {
        // Show loading while checking auth state
        if (snapshot.connectionState == ConnectionState.waiting) {
          return const Scaffold(
            body: Center(
              child: CircularProgressIndicator(),
            ),
          );
        }

        // If user is logged in, show home screen
        if (snapshot.hasData) {
          return const HomeScreen();
        }

        // If user is not logged in, show login screen
        return const LoginScreen();
      },
    );
  }
}
```

2.7 Portal_berital_android.iml

File .iml adalah file konfigurasi modul yang dibuat otomatis oleh Android Studio / IntelliJ IDEA. File ini digunakan untuk memberi tahu IDE bagaimana struktur project Android-Flutter kamu disusun, termasuk lokasi folder, SDK yang digunakan, dan dependensi library. File ini tidak berpengaruh langsung ke aplikasi saat runtime, tapi sangat penting untuk proses build dan pengelolaan project di IDE.

```
<?xml version="1.0" encoding="UTF-8"?>
<module type="JAVA_MODULE" version="4">
  <component name="FacetManager">
    <facet type="android" name="Android">
      <configuration>
        <option name="ALLOW_USER_CONFIGURATION" value="false" />
        <option name="GEN_FOLDER_RELATIVE_PATH_APT" value="/gen" />
        <option name="GEN_FOLDER_RELATIVE_PATH_AIDL" value="/gen" />
        <option name="MANIFEST_FILE_RELATIVE_PATH"
value="/app/src/main/AndroidManifest.xml" />
        <option name="RES_FOLDER_RELATIVE_PATH" value="/app/src/main/res" />
        <option name="ASSETS_FOLDER_RELATIVE_PATH" value="/app/src/main/assets"
/>
        <option name="LIBS_FOLDER_RELATIVE_PATH" value="/app/src/main/libs" />
        <option name="PROGUARD_LOGS_FOLDER_RELATIVE_PATH"
value="/app/src/main/proguard_logs" />
      </configuration>
    </facet>
  </component>
  <component name="NewModuleRootManager" inherit-compiler-output="true">
    <exclude-output />
    <content url="file://$MODULE_DIR$">
      <sourceFolder url="file://$MODULE_DIR$/app/src/main/java"
isTestSource="false" />
      <sourceFolder url="file://$MODULE_DIR$/app/src/main/kotlin"
isTestSource="false" />
      <sourceFolder url="file://$MODULE_DIR$/gen" isTestSource="false"
generated="true" />
    </content>
    <orderEntry type="jdk" jdkName="Android API 29 Platform" jdkType="Android
SDK" />
    <orderEntry type="sourceFolder" forTests="false" />
    <orderEntry type="library" name="Flutter for Android" level="project" />
    <orderEntry type="library" name="KotlinJavaRuntime" level="project" />
  </component>
</module>
```

2.8 Firebase.json

File firebase.json adalah file konfigurasi utama **Firebase** yang memberi tahu Firebase bagaimana proyek kamu dijalankan, platform apa yang terhubung (Android & Web), serta bagaimana **Firebase Hosting** bekerja untuk aplikasi kamu.

```
{
  "flutter": {
    "platforms": {
      "android": {
        "default": {
          "projectId": "portalb-a3010",
          "appId": "1:345822162548:android:5932b887f67c8e6645ab21",
```

```
          "fileOutput": "android/app/google-services.json"
        }
      },
      "dart": {
        "lib/firebase_options.dart": {
          "projectId": "portalb-a3010",
          "configurations": {
            "android": "1:345822162548:android:5932b887f67c8e6645ab21",
            "web": "1:345822162548:web:4cce3c3e72b0360345ab21"
          }
        }
      }
    }
  },
  "hosting": {
    "public": "public",
    "ignore": [
      "firebase.json",
      "**/.*",
      "**/node_modules/**"
    ],
    "rewrites": [
      {
        "source": "**",
        "destination": "/index.html"
      }
    ]
  }
}
```

2.9 Pubspec.yaml

File `pubspec.yaml` adalah file konfigurasi utama proyek Flutter yang digunakan untuk mengatur identitas aplikasi, versi SDK, dan semua dependency (library) yang dibutuhkan. File ini memastikan semua paket yang digunakan aplikasi bisa diunduh dan berjalan dengan benar.

```
name: portal_berita
description: "A new Flutter project."
publish_to: 'none'
version: 0.1.0

environment:
  sdk: ^3.9.2
```

```yaml
dependencies:
  flutter:
    sdk: flutter
  firebase_core: ^3.6.0
  firebase_auth: ^5.3.1
  cloud_firestore: ^5.4.4
  provider: ^6.1.2
  http: ^1.2.2

dev_dependencies:
  flutter_test:
    sdk: flutter
  flutter_lints: ^5.0.0

flutter:
  uses-material-design: true
```

3. BAB III

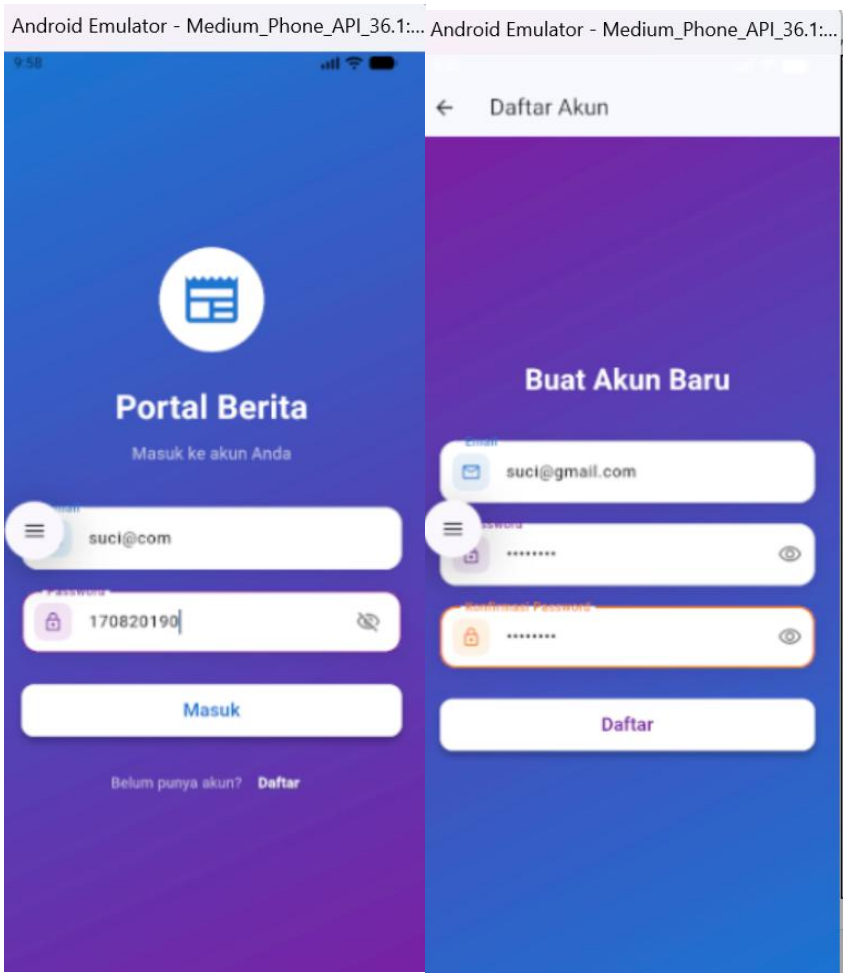
3.1 Detail Tampilan

1. daftar email & password

Elemen utamanya :

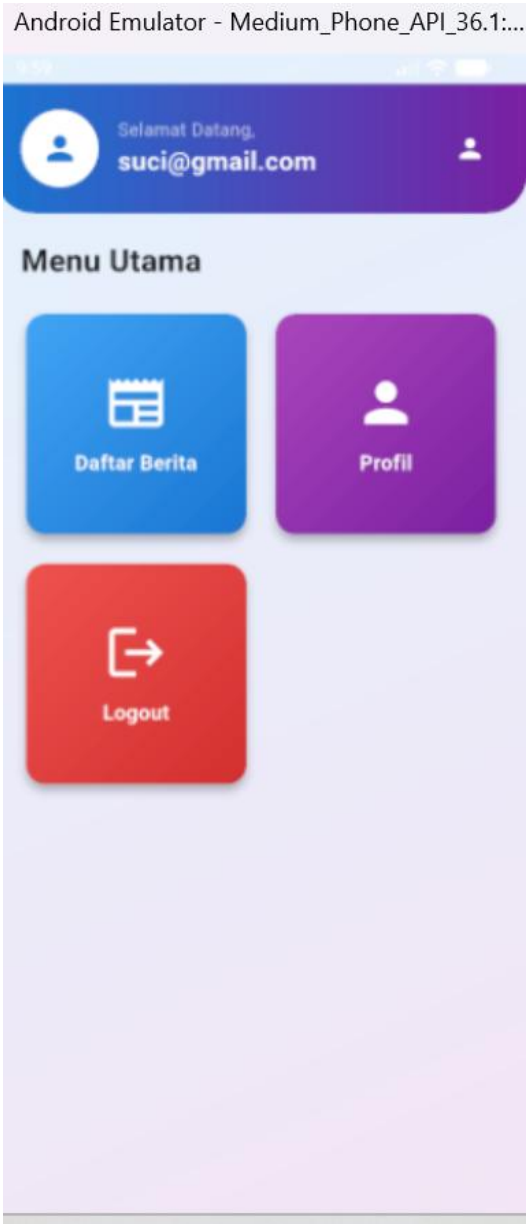
- Judul “Buat akun baru
- Input email
- Input password
- Inout konfirmasi password (dengan validasi jika tidak cocok)
- Tombol daftar untuk membuat akun baru.

Fungsi gamambr ini yaitu pengguna baru uuntuk membuat akun dengan email dann password


Android Emulator - Medium_Phone_API_36.1:...
Portal Berita
Masuk ke akun Anda
suci@com
170820190
Masuk
Belum punya akun? Daftar
Android Emulator - Medium_Phone_API_36.1:...
Daftar Akun
Buat Akun Baru
suci@gmail.com
Daftar

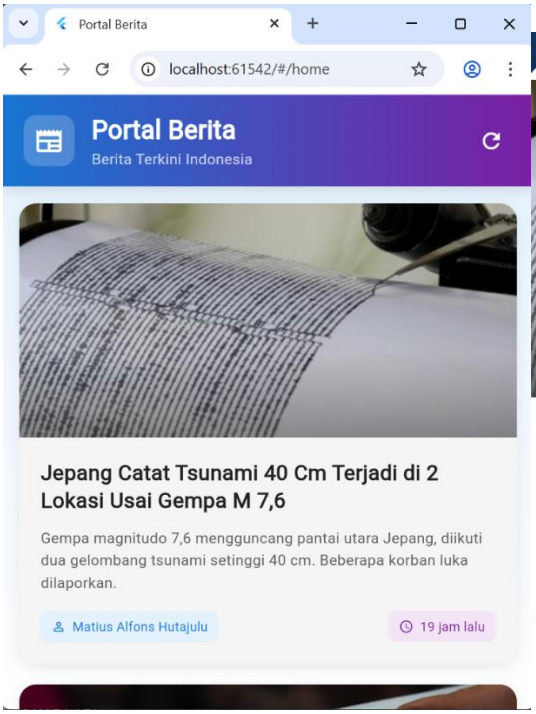
2. Menu utama

Gambar tersebut menunjukkan halaman Menu Utama (Beranda/Dashboard) aplikasi. Tampilan ini adalah halaman **Dashboard/Menu Utama** dengan jenis navigasi **Grid/Menu Grid (GridView Navigation)** yang digunakan sebagai pusat akses fitur aplikasi.



3. Sub menu (daftar berita)(news Detail/Detail page)

Tampilan ini merupakan halaman Daftar Berita yang berfungsi menampilkan kumpulan berita dalam bentuk daftar (ListView). Halaman ini adalah submenu yang diakses dari Menu Utama, dengan jenis navigasi List-based Navigation. Dan gambar ke 2 adalah halaam detail berita dimana halaman ini akan muncul ketika pengguna menekan salah satu berita dari halaman daftar berita, menampilkan informasi berita secara lebih lengkap



4. Profil

Menu Profil merupakan halaman yang digunakan pengguna untuk melihat serta mengubah informasi akun mereka. Pada halaman ini terdapat tiga informasi utama yang dapat ditampilkan dan disesuaikan: Nama, Email, dan Bio.

Pada tampilan awal, semua data pengguna masih kosong atau belum diisi.

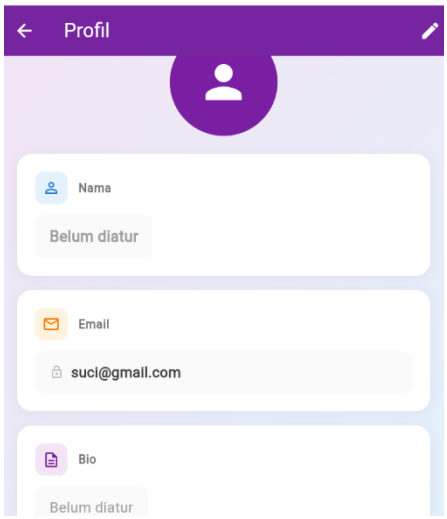
Elemen yg terlihat yaitu

- **Foto Profil**
  Masih menggunakan icon default (belum ada foto pengguna).
- **Nama**
  Menampilkan teks *“Belum diatur”*, menunjukkan bahwa nama belum dibuat.
- **Email**
  Sudah terlihat (misalnya *suci@gmail.com*), namun bersifat **tidak dapat diedit** karena digunakan sebagai identitas login.

- **Bio**
  Juga masih menampilkan *“Belum diatur”* sebagai tanda belum ada informasi tambahan yang ditambahkan.

Fungsi status “sebelum diatur’

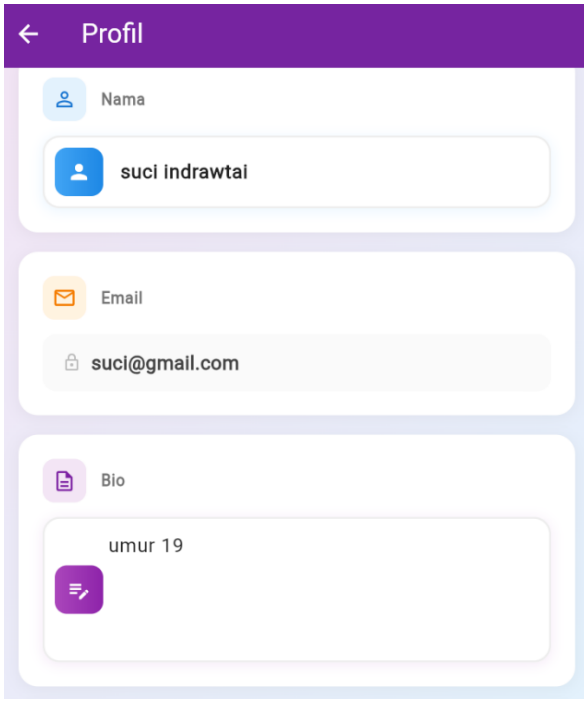
Memberikan info awal bahwa pengguna belumm melengkapi dan perlu menekan ikon edit (pensil) untuk mengubah data



Tampilan profil – setelah diatur

Elemn yg berubah yaitu :

- Nama = Suci indrawati
- Email = tetap sama karna bersifat terkunci
- Bio = umur 19



5. Logout

Fitur Logout berfungsi untuk mengakhiri sesi pengguna dan mengembalikan mereka ke halaman login. Proses ini memastikan keamanan akun serta mencegah orang lain mengakses aplikasi menggunakan akun tersebut.

a. Tampilan Ketika Tombol Logout Ditekan

Ketika pengguna menekan menu **Logout** di halaman utama, muncul **dialog konfirmasi**:

Isi dialog:

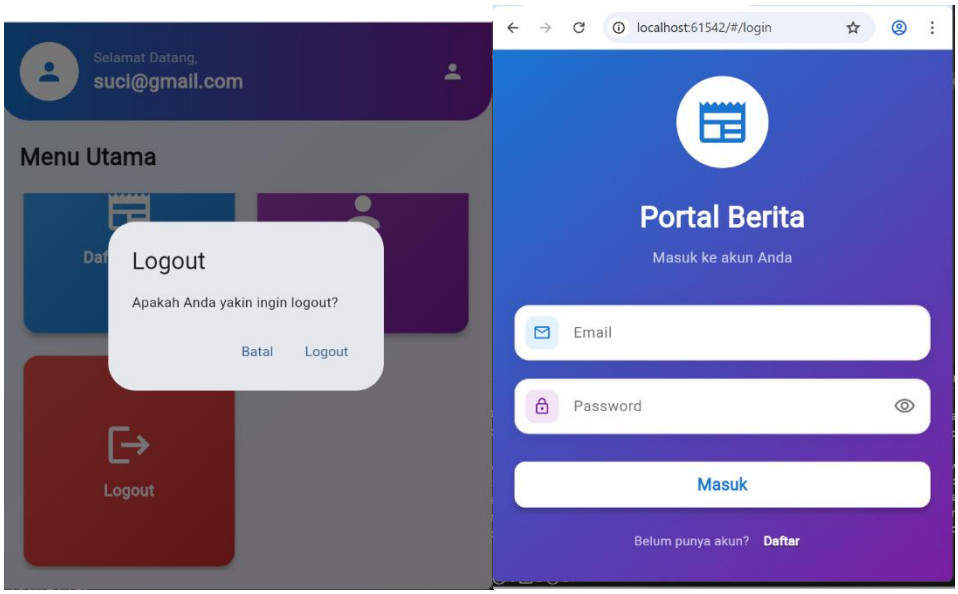
- **Pesan:**
  "Apakah Anda yakin ingin logout?"
- **Tombol Batal:**
  Jika dipilih, pengguna tetap berada di halaman utama.
- **Tombol Logout:**
  Jika dipilih, pengguna benar-benar keluar dari aplikasi.

Fungsi dari dialog ini adalah **mencegah logout tidak sengaja**.

b. Tampilan Setelah Logout Berhasil

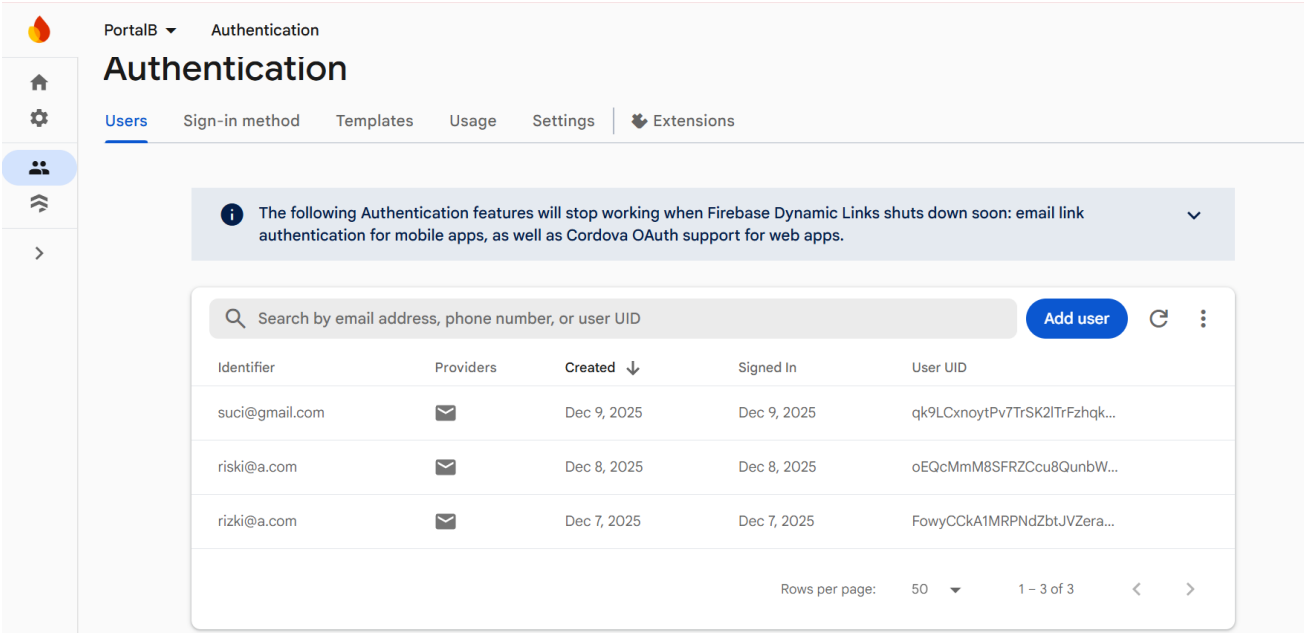
Setelah pengguna menekan tombol **Logout**, aplikasi akan:

- Menghapus sesi login pengguna
- Membersihkan data sementara (session/token)
- Mengarahkan pengguna kembali ke **halaman Login**



6. Penjelasan Firebase Authentication (Database Email Pengguna)

Firebase Authentication pada aplikasi Portal Berita berfungsi sebagai tempat penyimpanan data akun pengguna yang melakukan proses pendaftaran dan login. Pada halaman Users ini, Firebase menampilkan daftar email yang telah terdaftar beserta informasi penting lainnya seperti provider autentikasi yang digunakan, tanggal pembuatan akun, tanggal terakhir login, serta User UID yang menjadi identitas unik setiap pengguna. Data tersebut digunakan oleh aplikasi untuk mengelola proses login, logout, dan pengamanan akses ke halaman tertentu. Dengan memanfaatkan Firebase Authentication, aplikasi tidak perlu membuat sistem penyimpanan password secara manual, karena Firebase sudah menyediakan mekanisme keamanan otomatis untuk mengelola kredensial pengguna. Hal ini membuat proses autentikasi lebih aman, cepat, dan mudah diintegrasikan dengan fitur lain seperti profil pengguna dan menu utama aplikasi.



8. Penggunaan AI

(CHAT GPT, CURSOR)