

LAPORAN UJIAN TENGAH SEMESTER

MOBILE PROGRAMMING

EduConnect



Nama : Safri Firzan Sururi

NIM : 240605110084

Kelas : B

Tanggal : 25, Oktober 2025

JURUSAN TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM

MALANG

GANJIL 2025/2026

I. Pendahuluan

Aplikasi *EduConnect* merupakan aplikasi mobile yang dirancang menggunakan framework Flutter dengan tujuan untuk memberikan pengalaman belajar yang interaktif dan mudah diakses. Tema aplikasi ini diambil dari kebutuhan mahasiswa dan pelajar masa kini yang membutuhkan platform pembelajaran ringan namun tetap menarik secara visual.

Pemilihan tema ini dilatarbelakangi oleh pesatnya perkembangan teknologi mobile dalam dunia pendidikan. Dengan konsep antarmuka yang sederhana dan navigasi yang intuitif, pengguna dapat dengan cepat memahami alur aplikasi dan mengakses berbagai materi belajar secara efisien.

Aplikasi ini menggunakan data dummy API sebagai sumber data sementara yang dihubungkan ke tampilan antarmuka. Dengan pendekatan ini, pengembang dapat mensimulasikan alur aplikasi berbasis data tanpa harus terhubung ke backend sesungguhnya.

II. Perancangan Antarmuka(UI Design)

Perancangan antarmuka aplikasi *EduConnect* mengikuti prinsip dasar desain modern dengan pendekatan simpel, intuitif, dan responsif. Desain awal dibuat menggunakan Figma, lalu diimplementasikan ke dalam kode Flutter.

a. Konsep Desain

- Warna utama: biru muda (#4A90E2) dan putih sebagai latar, memberikan kesan tenang dan profesional.
Aksen warna: abu-abu muda untuk teks sekunder dan elemen nonfokus.

- Tipografi: menggunakan *Poppins* atau *Roboto*, dua jenis font yang umum dalam aplikasi edukasi modern.
- Ikon: menggunakan bawaan Icons Flutter untuk kesederhanaan dan konsistensi gaya.
- Tata letak (layout): kombinasi Column, ListView, Card, dan Container agar konten mudah dibaca dan responsif di berbagai ukuran layar.

b. Struktur Desain Halaman

1. Splash Page

Menampilkan logo aplikasi dengan efek animasi `AnimatedOpacity`. Tujuannya untuk memperkenalkan aplikasi secara singkat sebelum pengguna diarahkan ke *Home Page*.

2. Home Page

Berisi informasi pengguna yang diambil dari API, daftar menu belajar, dan tombol navigasi ke halaman lain. Menggunakan komponen ListView dan Card agar layout tetap rapi.

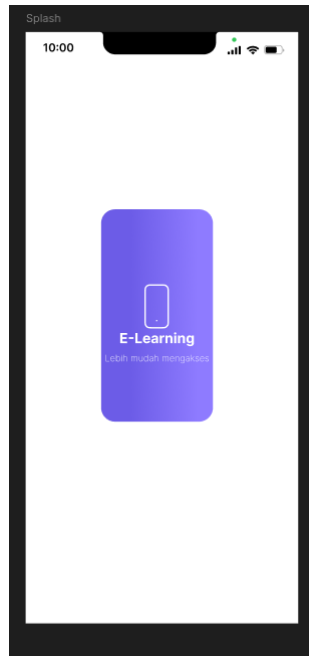
3. Study Page

Menampilkan daftar materi belajar dalam format list. Menggunakan `FutureBuilder` untuk menampilkan data dari *dummy API*.

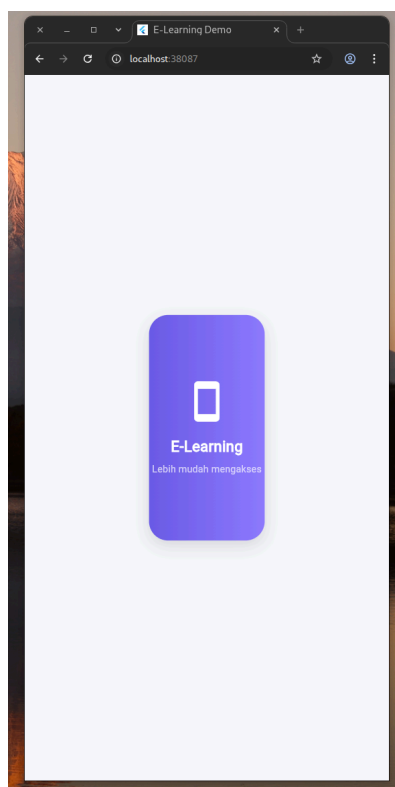
c. Contoh Tampilan (Desain Awal Figma & Implementasi)

- Splash Page:

> Figma:

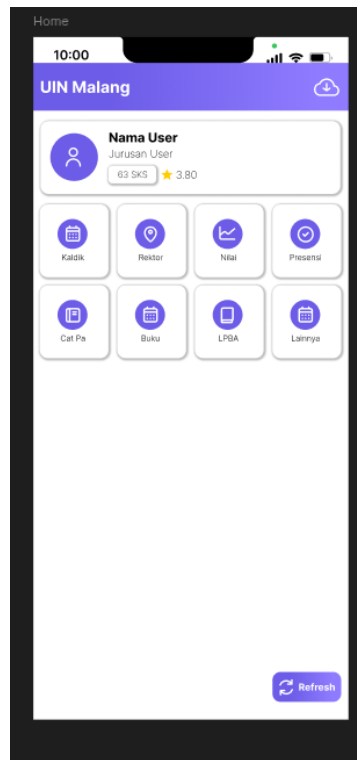


> Implementasi:

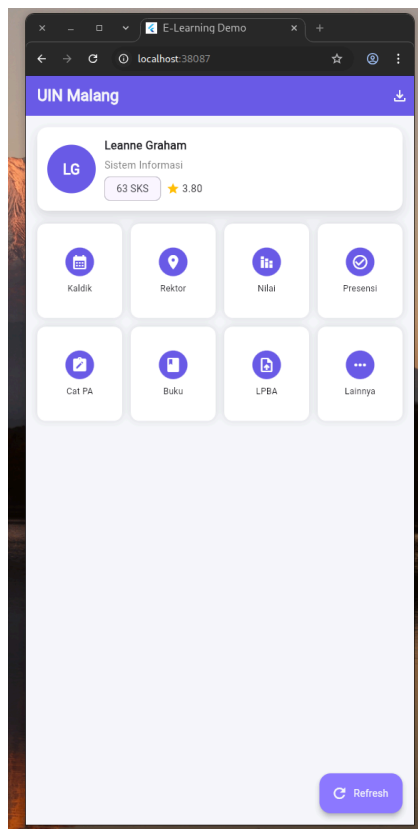


- Home Page:

> Figma:

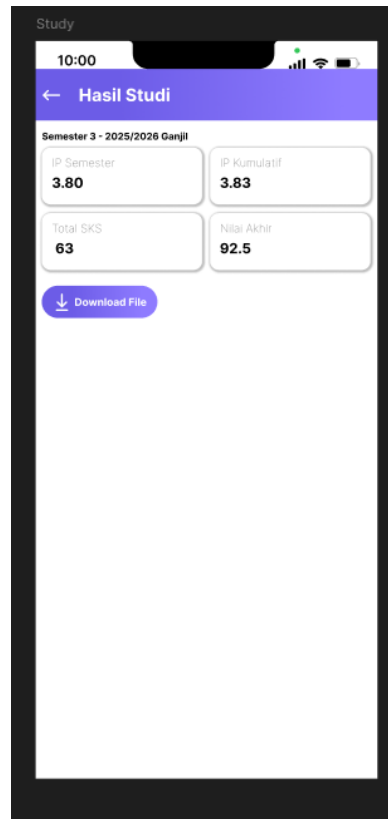


> Implementasi:

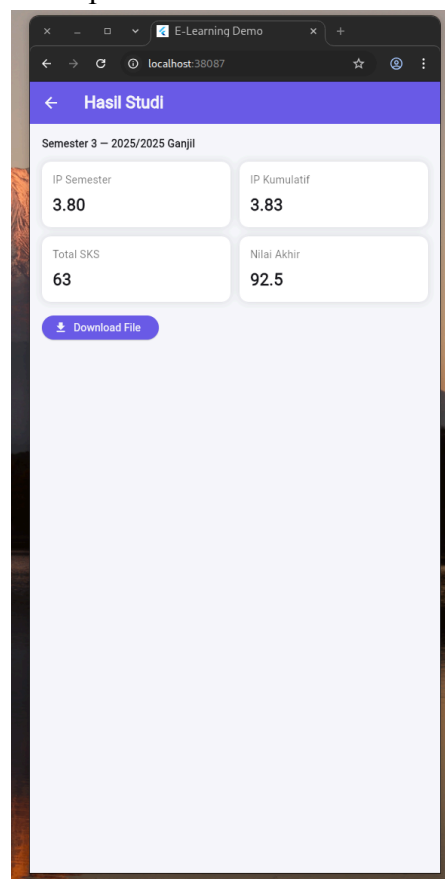


- Study Page:

> Figma :

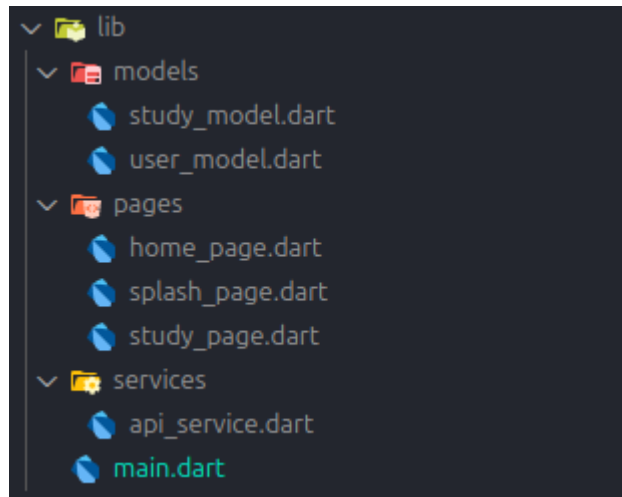


> Implementasi :



IV. Implementasi Program

a. Struktur Folder:



b. Penjelasan Komponen

1. main.dart

File main.dart berfungsi sebagai *entry point* atau titik awal eksekusi aplikasi. Di dalam file ini terdapat fungsi main() yang menjalankan widget utama bernama MyApp. Komponen MaterialApp digunakan untuk mengatur tema global, warna utama (*primaryColor*), dan halaman pertama yang ditampilkan yaitu SplashPage.

Selain itu, file ini juga mendefinisikan fungsi createRoute() yang berfungsi untuk mengatur animasi transisi antarhalaman menggunakan PageRouteBuilder. Dengan memanfaatkan kombinasi antara FadeTransition dan SlideTransition, perpindahan halaman terlihat lebih halus dan modern, sehingga meningkatkan pengalaman pengguna (*user experience*).

Berikut cuplikan kode implementasinya:

```
1 import 'package:flutter/material.dart';
2 import 'pages/splash_page.dart';
3
4 const primaryColor = Color(0xFF6C5CE7);
5 const accentColor = Color(0xFF8E7BFF);
6
7 void main() {
8   runApp(const MyApp());
9 }
10
11 class MyApp extends StatelessWidget {
12   const MyApp({super.key});
13   @override
14   Widget build(BuildContext context) {
15     return MaterialApp(
16       title: 'EduConnect',
17       debugShowCheckedModeBanner: false,
18       theme: ThemeData(
19         useMaterial3: true,
20         colorScheme: ColorScheme.fromSeed(seedColor: primaryColor),
21         scaffoldBackgroundColor: const Color(0xFFF6F7FB),
22       ),
23       home: const SplashPage(),
24     );
25   }
26 }
```

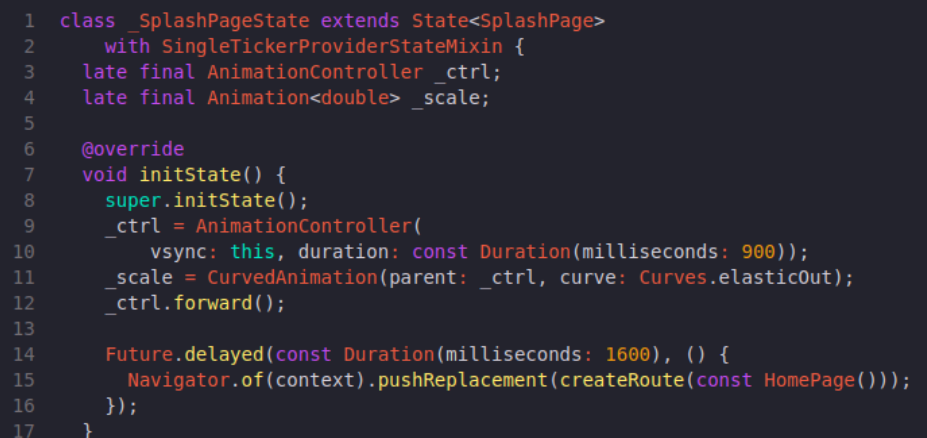
Untuk navigasi dengan efek animasi, digunakan fungsi berikut:

```
1 Route createRoute(Widget page) {
2   return PageRouteBuilder(
3     transitionDuration: const Duration(milliseconds: 520),
4     pageBuilder: (context, animation, secondaryAnimation) => page,
5     transitionsBuilder: (context, animation, sec, child) {
6       final tween = Tween(begin: const Offset(0, 0.08), end: Offset.zero)
7         .chain(CurveTween(curve: Curves.easeOut));
8       final opacity = Tween<double>(begin: 0, end: 1).animate(animation);
9       return FadeTransition(
10         opacity: opacity,
11         child: SlideTransition(position: animation.drive(tween), child: child),
12       );
13     },
14   );
15 }
```


Dengan fungsi ini, setiap kali halaman berpindah, tampilan akan mengalami animasi *fade-in* disertai *slide-up* yang lembut. Pendekatan ini memberikan kesan profesional dan memperkaya aspek estetika dari aplikasi.

Secara keseluruhan, file `main.dart` berperan sebagai fondasi utama dari aplikasi, mencakup pengaturan tema, warna, dan navigasi antarhalaman yang menjadi dasar bagi seluruh bagian lain seperti halaman *home*, *study*, dan *profile*.

2. `splash_page.dart`



```
1 class _SplashPageState extends State<SplashPage>
2   with SingleTickerProviderStateMixin {
3   late final AnimationController _ctrl;
4   late final Animation<double> _scale;
5
6   @override
7   void initState() {
8     super.initState();
9     _ctrl = AnimationController(
10       vsync: this, duration: const Duration(milliseconds: 900));
11     _scale = CurvedAnimation(parent: _ctrl, curve: Curves.elasticOut);
12     _ctrl.forward();
13
14     Future.delayed(const Duration(milliseconds: 1600), () {
15       Navigator.of(context).pushReplacement(createRoute(const HomePage()));
16     });
17   }
```

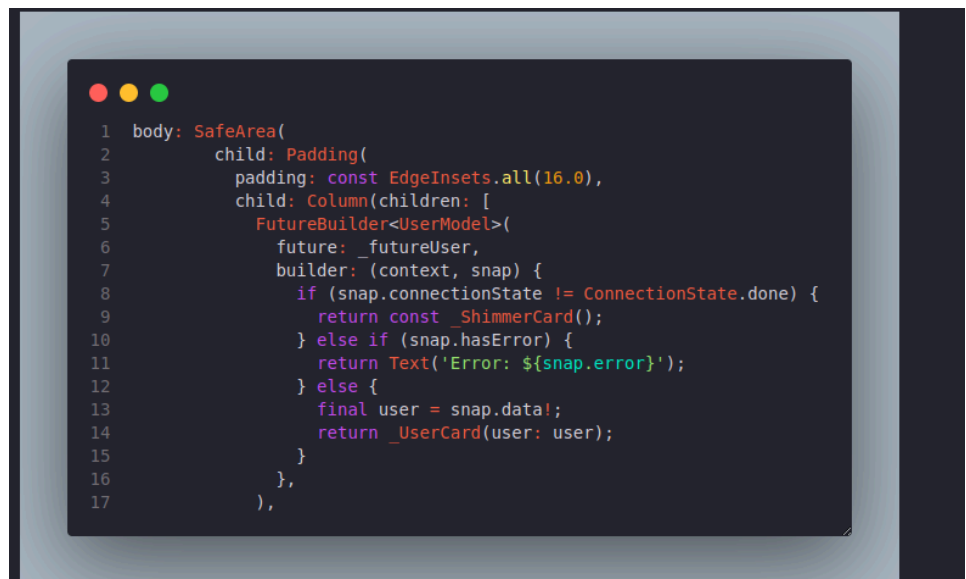
Cuplikan ini menunjukkan dua hal utama:

- Animasi Logo (ScaleTransition) dikontrol oleh AnimationController dan CurvedAnimation dengan efek elastis (Curves.elasticOut), membuat logo muncul dengan animasi lembut.

- Navigasi Otomatis – setelah 1,6 detik, aplikasi berpindah ke HomePage menggunakan `Navigator.of(context).pushReplacement()` yang memanggil `createRoute()` dari `main.dart`. Fungsi `createRoute()` menambahkan efek fade dan slide, sehingga transisi halaman terasa halus.

3. home_page.dart

Cuplikan utama yang menunjukkan logika inti halaman adalah:



Pada bagian ini, `FutureBuilder` digunakan untuk menunggu hasil dari `ApiService.fetchUser()`, yaitu data dummy user yang disimulasikan dari API.

- Ketika data belum selesai dimuat, widget `_ShimmerCard` akan muncul sebagai placeholder animasi.
- Jika data berhasil diambil, widget `_UserCard` menampilkan informasi pengguna seperti nama, program studi, SKS, dan IPK.
- Struktur ini menunjukkan konsep pengolahan data asinkron dengan feedback visual yang baik untuk pengguna.

4. api_service.dart

Cuplikan kode utama yang merepresentasikan inti dari file ini adalah:

A screenshot of a code editor with a dark background and light-colored text. The code is written in Dart and defines a class named ApiService. It includes two static constants for URLs, a static asynchronous method fetchUser() that uses http.get() to fetch data from a dummy API, and a try-catch block to handle exceptions. The code is numbered from 1 to 13.

```
1 class ApiService {
2   static const String userUrl = 'https://jsonplaceholder.typicode.com/users/1';
3   static const String studyUrl = 'https://jsonplaceholder.typicode.com/posts/1';
4
5   static Future<UserModel> fetchUser() async {
6     final resp = await http.get(Uri.parse(userUrl));
7     if (resp.statusCode == 200) {
8       return UserModel.fromJson(json.decode(resp.body));
9     } else {
10      throw Exception('Failed to load user (${resp.statusCode})');
11    }
12  }
13 }
```

Bagian ini menggunakan HTTP GET request untuk mengambil data dari endpoint dummy API dari :

<https://jsonplaceholder.typicode.com/users/1>.

Data yang diterima dalam format JSON kemudian diubah menjadi objek UserModel menggunakan metode fromJson().

- Fungsi ini bersifat asinkron, artinya aplikasi tidak akan membeku selama proses pemanggilan API berlangsung.
- Jika server merespons dengan status kode selain 200, fungsi akan melempar exception agar error dapat ditangani oleh widget seperti FutureBuilder di halaman Home.

Selain mengambil data dari internet, file ini juga menyediakan data dummy secara manual untuk menampilkan hasil akademik pengguna:



```
1 static Future<StudyModel> fetchStudy() async {
2     final resp = await http.get(Uri.parse(studyUrl));
3     if (resp.statusCode == 200) {
4         return StudyModel(
5             ipSemester: 3.80,
6             ipKumulatif: 3.83,
7             totalSks: 63,
8             totalNilai: 92.5,
9         );
10    } else {
11        throw Exception('Failed to load study (${resp.statusCode})');
12    }
13 }
```

Bagian ini tetap melakukan request ke endpoint studyUrl, tetapi data yang ditampilkan tidak berasal dari hasil response API melainkan dari nilai dummy lokal (nilai yang ditulis langsung di kode).

5. study_page.dart

Bagian paling penting dari halaman ini terletak pada penggunaan FutureBuilder untuk menampilkan data hasil studi yang diambil dari API dummy (ApiService.fetchStudy()), serta penataan tampilan statistik akademik menggunakan widget kustom _StatCard.



Pada bagian StudyPage, halaman ini menampilkan data hasil studi mahasiswa yang diambil melalui ApiService.fetchStudy() secara asynchronous. Data seperti IP Semester, IP Kumulatif, total SKS, dan nilai akhir ditampilkan

menggunakan FutureBuilder, yang menyesuaikan tampilan sesuai status pemanggilan data (loading, error, atau sukses). Saat data berhasil dimuat, nilai-nilai tersebut disajikan melalui widget _StatCard dalam bentuk kartu informatif dengan desain sederhana. Selain itu, tombol “Download File” berfungsi sebagai simulasi unduhan dan menampilkan notifikasi SnackBar saat ditekan.

V. Hasil dan Pembahasan

Aplikasi EduConnect (E-Learning Demo) berhasil diimplementasikan sesuai dengan rancangan awal yang menekankan tampilan sederhana, navigasi halus, dan pemuatan data berbasis API. Struktur utama aplikasi terdiri dari tiga halaman inti: SplashPage, HomePage, dan StudyPage, yang saling terhubung melalui animasi transisi lembut menggunakan fungsi createRoute() pada file main.dart.

Halaman HomePage berfungsi sebagai beranda utama yang menampilkan informasi pengguna dan menu fitur kampus. Data pengguna diambil secara asynchronous melalui ApiService.fetchUser() dan ditampilkan dalam bentuk kartu profil menggunakan widget _UserCard. Selain itu, tersedia tombol Refresh untuk memperbarui data dan ikon Download di AppBar yang menavigasi pengguna ke halaman StudyPage.

Sementara itu, halaman StudyPage menampilkan data hasil studi seperti IP semester, IP kumulatif, total SKS, dan nilai akhir yang diambil dari data dummy API menggunakan FutureBuilder. Data tersebut disajikan dalam layout yang rapi melalui widget _StatCard, sehingga pengguna dapat dengan mudah memahami performa akademiknya. Tombol Download File berfungsi sebagai simulasi aksi unduhan dan memberikan umpan balik visual menggunakan SnackBar.

Secara keseluruhan, hasil implementasi telah sesuai dengan desain awal. Navigasi antarhalaman berjalan lancar, data dummy berhasil ditampilkan dengan

baik, dan antarmuka aplikasi sudah mendukung pengalaman pengguna yang interaktif. Aplikasi ini telah memenuhi tujuan utamanya, yaitu menampilkan simulasi sistem e-learning dengan tampilan informatif dan alur interaksi yang sederhana namun fungsional.

VI. Kesimpulan

Proses pengembangan aplikasi EduConnect (E-Learning Demo) berjalan dengan baik, dimulai dari perancangan antarmuka menggunakan konsep navigasi sederhana hingga implementasi fitur utama yang berfokus pada pengalaman pengguna. Setiap halaman berhasil diimplementasikan sesuai fungsinya — SplashPage menampilkan animasi pembuka, HomePage menyajikan data pengguna serta menu navigasi, dan StudyPage menampilkan hasil studi berbasis data dummy dari API.

Pencapaian utama dari proyek ini adalah keberhasilan mengintegrasikan animasi, navigasi, dan pemanggilan data asynchronous dalam satu aplikasi yang ringan dan responsif. Tampilan antarmuka yang rapi serta interaksi pengguna yang mudah dipahami menunjukkan bahwa desain awal telah terealisasi dengan baik pada tahap implementasi.

Sebagai refleksi, meskipun aplikasi ini masih menggunakan data dummy dan belum terhubung ke sistem backend yang sesungguhnya, strukturnya sudah cukup modular untuk dikembangkan lebih lanjut. Potensi pengembangan selanjutnya mencakup penambahan autentikasi pengguna, integrasi database real-time, serta fitur interaktif seperti sistem penilaian dan notifikasi agar aplikasi semakin mendekati fungsi e-learning yang sesungguhnya.