

LAPORAN TUGAS KELOMPOK

MATA KULIAH: MOBILE PROGRAMMING

Identitas Kelompok

Nama Kelompok	: Kelompok 1
Anggota Kelompok	: 1. M. Wahyu Alvy Kusuma (221080200117) 2. M. Haffiz Fattah (221080200086) 3. Arry Wahyu Setiawan (221080200158) 4. Rizky Budi Aprianto (221080200072) 5. Ditta Adelia (221080200087) 6. Luluk Asti Qomariah (221080200106) 7. Nadia Tsabita (221080200113) 8. Sherly Mazyda Rizqi (221080200122)
Kelas	: 7A1
Dosen Pengampu	: Moch. Fauzan, S.Kom., M.Kom.
Judul/Topik Tugas	: Profil Universitas Muhammadiyah Sidoarjo
Tanggal Pengumpulan	: 06, November 2025
Link Github	: https://github.com/wahyualvyv/project-kelompok-1.git

Tujuan Tugas

Tujuan dari pengerjaan tugas proyek kelompok ini adalah sebagai berikut:

1. Memahami konsep dan alur kerja pengembangan aplikasi mobile menggunakan framework Flutter.
2. Mampu merancang dan mengimplementasikan antarmuka pengguna (UI) yang informatif dan terstruktur.
3. Menerapkan berbagai jenis layout widget (seperti Column, Row, Stack, Container) dan widget fungsional (seperti GridView, ListView, Card).
4. Mengimplementasikan navigasi antar halaman (multi-page navigation) untuk perpindahan layar.
5. Mengelola assets (gambar dan logo) dalam proyek Flutter.
6. Membangun aplikasi fungsional "Informasi Kampus UMSIDA" sesuai dengan pembagian tugas yang telah disepakati.

Deskripsi Proyek

Proyek yang kami kembangkan adalah sebuah aplikasi *mobile* "Profil UMSIDA". Aplikasi ini berfungsi sebagai pusat informasi ringkas mengenai Universitas Muhammadiyah Sidoarjo yang dapat diakses dengan mudah oleh mahasiswa.

Alur aplikasi dimulai dari *Splash Screen* sebagai pengenalan visual, yang kemudian mengarahkan pengguna ke Halaman Utama (Home). Halaman Utama berperan sebagai pusat navigasi, yang menyediakan akses terstruktur ke berbagai informasi penting, di antaranya: Profil Kampus (mencakup sejarah dan logo), Visi & Misi, daftar Fakultas & Program Studi, Galeri foto, serta informasi Kontak & Lokasi kampus.

Analisis dan Perancangan

1. Analisis kebutuhan aplikasi

Kebutuhan Fungsional (Fitur):

a. Splash Screen

Aplikasi menampilkan layar pembuka logo Universitas Muhammadiyah Sidoarjo selama beberapa detik sebelum masuk ke Halaman Utama.

b. Home Page

- Menampilkan AppBar dengan judul "Profil Universitas" dan "UMSIDA..
- Menampilkan banner dengan teks sambutan.
- Menampilkan card deskripsi singkat yang berisi logo dan deskripsi singkat tentang UMSIDA.
- Menampilkan bagian Info Unggulan dalam tiga card terpisah, yang menyajikan data statistik kampus, terdiri dari jumlah fakultas, jumlah mahasiswa, dan jumlah program studi.
- Menampilkan menu navigasi cepat, yaitu profil, visi misi, fakultas & prodi, galeri, kontak & lokasi) dalam bentuk card.
- Menampilkan footer di bagian bawah.

c. Profil

- Menampilkan AppBar dengan judul "Profil UMSIDA"
- Menampilkan Card pengantar berisi logo umsida dan nama kampus.
- Menampilkan status akreditasi label akreditasi unggul.
- Menampilkan deskripsi singkat tentang UMSIDA.

- Menampilkan motto kampus “Disini Pencerahan Bersemi”.
- Merinici fasilitas yang tersedia seperti perpustakaan, laboratorium dan lainnya.
- Memberikan informasi alamat lengkap kampus 1, kampus 2 dan kampus 3.
- Menampilkan nomor telepon sebagai kontak utama.

d. Visi Misi

- Menampilkan AppBar dengan judul "Visi Misi UMSIDA".
- Menampilkan Bagian Pengantar (Banner/Card Utama) dengan judul "Komitmen dan Arah UMSIDA" yang berisi teks bahwa Visi, Misi, Tujuan, dan Sasaran menjadi pedoman dalam pengembangan institusi.
- Menampilkan Bagian Utama Visi, yang menjelaskan cita-cita institusi (menjadi universitas unggul, berdaya saing global, dan berlandaskan nilai Islam).
- Menampilkan Fokus Utama yang ditekankan (Keunggulan akademik dan relevansi industri global).
- Menampilkan Bagian Misi dalam empat poin terpisah dan terperinci (Pendidikan Berkualitas, Riset dan Pengembangan, Nilai-nilai Islam, dan Pemberdayaan Masyarakat).
- Menampilkan Bagian Tujuan dalam tiga poin (A, B, C) yang merinci hasil yang ingin dicapai dari Misi (Lulusan Kompeten, Penelitian Berkelanjutan, dan Kemitraan Strategis).
- Menampilkan Bagian Sasaran yang mengelompokkan target spesifik ke dalam dua fokus utama: Akademik dan Penelitian & Inovasi.

e. Fakultas dan program studi

- Menampilkan judul halaman “ fakultas dan program studi” .
- Menampilkan card struktur akademik berisi logo UMSIDA, judul “Struktur Akademik UMSIDA” dan Informasi ringkas.
- Menampilkan judul bagian “Daftar Fakultas”.
- Menampilkan daftar fakultas yang dapat diklik untuk menampilkan derail program studi di bawahnya.
- Menampilkan halaman daftar program studi di bawahnya saat sebuah fakultas di klik lengkap dengan deskripsi singkat.

f. Galeri

- Menampilkan judul “Galeri” atau nama sub menu yang sedang lihat di sub halaman.
- menampilkan ikon < untuk kembali ke halaman sebelumnya.
- menampilkan tiga menu utama dalam bentuk card, yang dapat di klik berdasarkan kategori yaitu gedung, perpustakaan dan fasilitas umum.

- Menampilkan daftar foto foto gedung kampus, perpustakaan dan fasilitas umum, setiap fotonya di sajikan dengan format card dengan detail di bawahnya.

g. Kontak & Lokasi

- Menampilkan informasi tampilan yang berisi nomor telepon, email, website dan alamat tentang Universitas Muhammadiyah Sidoarjo
- Menampilkan informasi maps yang langsung terhubung ke alamat Universitas Muhammadiyah Sidoarjo
- Menampilkan Informasi sosial media facebook, Youtube, Instagram, WA universitas Muhammadiyah Sidoarjo
- Menampilkan informasi jam operasional Universitas Muhammadiyah Sidoarjo

Tampilan

Desain antarmuka (UI) aplikasi dibuat dengan bersih, profesional, dan konsisten di semua halaman, yang diimplementasikan melalui keseragaman penggunaan warna, jenis *font*, dan gaya *header*. Semua *assets* visual yang digunakan, seperti logo, ikon, dan foto disesuaikan, dan selaras dengan identitas Universitas Muhammadiyah Sidoarjo.

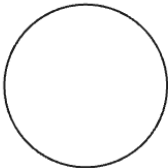
Interaksi

Aplikasi memberikan pengalaman pengguna (UX) yang lancar dan fungsional. Untuk itu, layout aplikasi dirancang agar adaptif (responsif) terhadap berbagai ukuran layar, memastikan semua elemen tetap terlihat jelas. Selain itu, untuk konten yang panjang atau melebihi ukuran layar, terutama di Halaman Utama dan halaman detail.

2. Sketsa Aplikasi

Sebelum memulai tahap implementasi (coding), kami terlebih dahulu merancang arsitektur informasi dan tata letak visual aplikasi. Proses ini kami tuangkan dalam bentuk sketsa (wireframe low-fidelity) untuk memvisualisasikan struktur, alur navigasi, dan penempatan komponen di setiap halaman menggunakan aplikasi Excalidraw.

Berikut adalah sketsa desain aplikasi Profil UMSIDA:



Logo Umsida

UMSIDA

Banner Selamat Datang
Universitas Muhammadiyah
Sidoarjo

Deskripsi Universitas

Fakultas

Prodi

Maha
siswa

Menu Informasi

Profil

Visi
misi

Fak&
Prodi

Galeri

Kontak
lokasi

More

Foter bawah

← Profil


☐ Tentang Umsida
Isi tentang umsida...

☐ Semboyan
Dari Sini Pencerahan Bersani

☐ Fasilitas Kampus
Isi fasilitas
Isi fasilitas
Isi fasilitas

☐ Lokasi Kampus
Button

← Visi Misi



Teks Sambutan


☐ Visi
Isi visi ...

☐ Misi
1 Isi misi ...
2 Isi misi ...
3 Isi misi ...

☐ Tujuan
Isi tujuan ...

☐ Sasaran
Isi sasaran ...

← Fakultas & Prodi



Teks Sambutan

☐ Daftar Fakultas
☐ Fakultas Sains dan Teknologi v
☐ Fakultas FBHIS v
☐ Fakultas FPIP v

← Kontak & Lokasi

Gambar Peta

Informasi Kontak
Telepon
Email
Website
Alamat

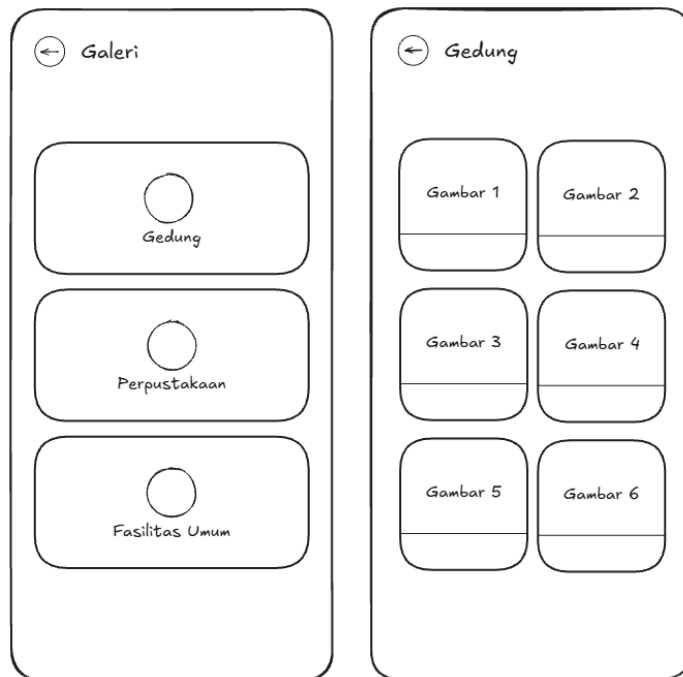
facebook

Instagram

Youtube

WhatsApp

Jam Operasional
Hari & Jam
Hari & Jam
Hari & Jam



Penjelasan struktur widget yang digunakan (Row, Column, Container, dll)

Struktur antarmuka pengguna (UI) pada Flutter DIBUAT dengan menggabungkan dan menyusun berbagai widget. Berikut adalah penjelasan widget-widget yang kami gunakan di seluruh halaman:

1. Widget Struktur Aplikasi & Halaman:

- **MaterialApp**
Widget akar (root) dari aplikasi kami, yang dideklarasikan di `main.dart`. `MaterialApp` membungkus seluruh aplikasi dan menyediakan fungsionalitas dasar seperti tema (skema warna biru UMSIDA) dan, yang terpenting, sistem navigasi bernama (*named routes*) yang kami gunakan untuk berpindah halaman (contoh: dari `/home` ke `/profil` atau `/kontak`).
- **Scaffold**
Setiap halaman di aplikasi kami (Home, Profil, Visi Misi, dll.) dibungkus dengan `Scaffold`. *Widget* ini menyediakan kerangka kerja standar halaman, seperti `AppBar` (bilah judul di atas) dan properti `body` (tempat kami meletakkan semua konten halaman).

2. Widget Layout:

- **Column**

Widget layout yang paling sering kami gunakan. Column berfungsi untuk menyusun anak-anak widget (widgets di dalamnya) secara vertikal, dari atas ke bawah.

Contoh Penggunaan: Column menjadi tulang punggung di Halaman Utama untuk menyusun AppBar kustom, card sambutan, card deskripsi, highlight info, dan menu navigasi secara berurutan. Begitu pula di halaman Kontak & Lokasi untuk menyusun card informasi, Peta Lokasi, Media Sosial, dan Jam Operasional.

- **SingleChildScrollView**

Sebuah Column akan error (menampilkan overflow pixel) jika kontennya lebih panjang dari layar. Untuk mengatasi ini, kami membungkus Column utama di halaman-halaman yang panjang (seperti Home Page dan Kontak & Lokasi) dengan SingleChildScrollView. Widget ini secara otomatis membuat konten di dalamnya dapat digulir (scrollable) oleh pengguna.

- **Row**

Kebalikan dari Column, Row digunakan untuk menyusun widget secara horizontal, dari kiri ke kanan.

Contoh Penggunaan: Kami menggunakan Row pada bagian Highlight Info (di Home Page) untuk menyejajarkan tiga card (Fakultas, Mahasiswa, Prodi). Kami juga menggunakannya di halaman Kontak untuk menyejajarkan item "Jam Operasional" dengan jamnya.

- **GridView**

Untuk tata letak yang membutuhkan kisi-kisi (grid), kami menggunakan GridView.

Contoh Penggunaan: GridView sangat ideal untuk Menu Informasi di Halaman Utama (6 tombol menu) dan Halaman Galeri Gedung untuk menampilkan kumpulan foto dalam 2 kolom.

3. Widget Tampilan & Kontainer:

- **Container**

Ini adalah widget "kotak serbaguna". Kami menggunakannya untuk memberikan styling kustom seperti margin (jarak di luar), padding (jarak di dalam), warna latar belakang (misalnya footer biru), atau untuk

mengatur bentuk (misalnya `borderRadius` untuk membuat sudut melengkung).

- Card

Untuk membuat blok-blok informasi yang "mengambang" dengan bayangan (elevation), kami menggunakan Card. Widget ini sangat penting untuk estetika aplikasi kami. Contoh Penggunaan, semua bagian di Home Page (Sambutan, Deskripsi, Highlight Info) dibungkus dengan Card. Begitu pula di halaman Kontak untuk setiap bagian informasi.

- Image.asset

Digunakan setiap kali kami perlu menampilkan gambar yang tersimpan di dalam folder assets proyek, seperti logo UMSIDA di Splash Screen, foto-foto di Halaman Galeri, dan ikon-ikon di menu.

- Text

Digunakan untuk menampilkan semua teks. Kami sering menambahkan properti style (`TextStyle`) untuk mengatur ukuran font (contoh: `fontSize`), ketebalan (`fontWeight.bold`), dan warna.

- Icon

Digunakan untuk menampilkan ikon-ikon bawaan Flutter (seperti ikon telepon, email, dan website di halaman Kontak).

4. Widget Interakti & Fungsional:

- Container

Widget "kotak serbaguna", kami menggunakannya untuk memberikan styling kustom seperti margin (jarak di luar), padding (jarak di dalam), warna latar belakang (misalnya footer biru), atau untuk mengatur bentuk (misalnya `borderRadius` untuk membuat sudut melengkung).

- Card

Untuk membuat blok-blok informasi yang "mengambang" dengan bayangan (elevation), kami menggunakan Card. Widget ini sangat penting untuk estetika aplikasi kami.

Contoh Penggunaan: Semua bagian di Home Page (Sambutan, Deskripsi, Highlight Info) dibungkus dengan Card. Begitu pula di halaman Kontak untuk setiap bagian informasi.

- Image.asset

Digunakan setiap kali kami perlu menampilkan gambar yang tersimpan di dalam folder assets proyek, seperti logo UMSIDA di Splash Screen, foto-foto di Halaman Galeri, dan ikon-ikon di menu.

- Text

Digunakan untuk menampilkan semua teks. Kami sering menambahkan properti style (TextStyle) untuk mengatur ukuran font (cth: fontSize), ketebalan (fontWeight.bold), dan warna.

- Icon

Digunakan untuk menampilkan ikon-ikon bawaan Flutter (seperti ikon telepon, email, dan website di halaman Kontak).

Implementasi Program

1. Inisialisasi dan Struktur Aplikasi (main.dart)

Setiap aplikasi Flutter dimulai dari fungsi main(). Pada file main.dart, aplikasi menginisialisasi MaterialApp dan menetapkan SplashScreen sebagai halaman pertama yang dimuat.

Dart

```
// main.dart

import 'package:flutter/material.dart';

import '/pages/splash_screen.dart'; // Impor SplashScreen

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {
```

```

@override

Widget build(BuildContext context) {

  return MaterialApp(

    debugShowCheckedModeBanner: false, // Menghilangkan banner debug

    title: 'UMSIDA App',

    theme: ThemeData(

      primarySwatch: Colors.blue,

      fontFamily: 'Roboto', // Mengatur font default

    ),

    home: SplashScreen(), // Halaman pertama yang dibuka

  );

}

}

```

Penjelasan:

- runApp(MyApp()): Memulai aplikasi dengan me-render widget MyApp.
- MaterialApp: Ini adalah widget dasar yang membungkus sebagian besar aplikasi. Ini menyediakan fungsionalitas seperti navigasi (rute), tema aplikasi, dan judul.
- home: SplashScreen(): Menetapkan bahwa SplashScreen adalah widget yang akan ditampilkan saat aplikasi pertama kali dibuka.

2. Logika SplashScreen dan Navigasi Otomatis (splash_screen.dart)

SplashScreen berfungsi sebagai layar pemuatan awal. Halaman ini menggunakan AnimationController untuk menganimasikan logo dan teks. Fitur utamanya adalah Timer yang secara otomatis mengarahkan pengguna ke HomePage setelah durasi tertentu.

```

// splash_screen.dart

class _SplashScreenState extends State<SplashScreen>

  with SingleTickerProviderStateMixin {

  late AnimationController _animationController;

  late Animation<double> _fadeAnimation;

```

```
// ... (definisi animasi lainnya)
```

```
@override
```

```
void initState() {  
  super.initState();  
  _animationController = AnimationController(  
    vsync: this,  
    duration: Duration(milliseconds: 2000),  
  );
```

```
// ... (konfigurasi animasi)
```

```
_animationController.forward(); // Memulai animasi
```

```
// Navigasi ke halaman utama setelah 4 detik
```

```
Timer(Duration(seconds: 4), () {  
  Navigator.pushReplacement(  
    context,  
    MaterialPageRoute(  
      pageBuilder: (context, animation, secondaryAnimation) => HomePage(),  
      transitionsBuilder: (context, animation, secondaryAnimation, child) {  
        // Efek transisi fade  
        return FadeTransition(  
          opacity: animation.drive(Tween(begin: 0.0, end: 1.0)),  
          child: child,  
        );  
      },  
    ),  
  );
```

```

);

});

}

// ... (build method menggunakan FadeTransition, ScaleTransition, dll.)

}

```

Penjelasan:

- with `SingleTickerProviderStateMixin`: Diperlukan oleh `AnimationController` untuk mengatur *refresh rate* animasi.
- `initState()`: Metode ini dipanggil sekali saat widget dibuat. Ini adalah tempat yang ideal untuk:
 1. Menginisialisasi `AnimationController`.
 2. Menjalankan `Timer`.
- `Timer`: Menjalankan kode di dalamnya setelah 4 detik.
- `Navigator.pushReplacement`: Pindah ke `HomePage` dan mengganti tumpukan navigasi. Ini mencegah pengguna menekan tombol "kembali" dan kembali ke `SplashScreen`.

3. Halaman Utama: `CustomScrollView` dan Manajemen State (`home_page.dart`)

`HomePage` adalah halaman paling kompleks. Halaman ini menggunakan `CustomScrollView` untuk UI yang dinamis dan mengelola *state* untuk menampilkan/menyembunyikan Tombol Aksi Mengambang (FAB).

Dart

```

// home_page.dart

class _HomePageState extends State<HomePage> with TickerProviderStateMixin {

  late ScrollController _scrollController;

  late AnimationController _fabAnimationController;

  bool _showFab = false; // Variabel state untuk FAB

  @override

  void initState() {

    super.initState();
  }
}

```

```

_scrollController = ScrollController();

_fabAnimationController = AnimationController(
  vsync: this,
  duration: Duration(milliseconds: 300),
);

// Listener untuk mendeteksi scroll
_scrollController.addListener(() {
  if (_scrollController.offset > 200 && !_showFab) {
    setState(() => _showFab = true); // Memperbarui state
    _fabAnimationController.forward(); // Memulai animasi FAB
  } else if (_scrollController.offset <= 200 && _showFab) {
    setState(() => _showFab = false); // Memperbarui state
    _fabAnimationController.reverse(); // Membalik animasi FAB
  }
});
}

```

`@override`

```

Widget build(BuildContext context) {
  return Scaffold(
    body: CustomScrollView(
      controller: _scrollController, // Menghubungkan controller
      slivers: [
        _buildAppBar(), // Ini adalah SliverAppBar
        SliverToBoxAdapter(
          child: Column(
            children: [

```

```

        _buildWelcomeBanner(),
        _buildMenuInformasi(),
        // ... (konten halaman lainnya)

    ],
),
),
],
),
floatingActionButton: ScaleTransition( // FAB dengan animasi
    scale: _fabAnimationController,
    child: FloatingActionButton(
        onPressed: () {
            // Aksi untuk scroll ke atas
            _scrollController.animateTo(
                0,
                duration: Duration(milliseconds: 500),
                curve: Curves.easeInOut,
            );
        },
        child: Icon(Icons.arrow_upward),
    ),
),
);
}
// ...
}

```

Penjelasan:

- Struktur UI: Halaman ini menggunakan CustomScrollView dan SliverAppBar (di dalam _buildAppBar()) untuk menciptakan efek di mana *app bar* bisa mengecil saat halaman di-scroll.
- Manajemen State:
 1. Sebuah ScrollController (_scrollController) memantau posisi scroll.
 2. _scrollController.addListener akan dipanggil setiap kali pengguna melakukan scroll.
 3. Di dalam *listener*, kita memanggil setState() untuk mengubah nilai _showFab.
 4. setState() memberi tahu Flutter untuk me-render ulang UI dengan nilai _showFab yang baru.
- Animasi FAB: FloatingActionButton dibungkus dengan ScaleTransition. Visibilitasnya dikontrol oleh _fabAnimationController yang dipicu oleh *scroll listener*.

4. Navigasi dan Pengiriman Data (Contoh: galeri.dart ke galeri_detail_page.dart)

Fitur penting dalam aplikasi adalah berpindah halaman sambil membawa data. Contoh terbaik adalah saat pengguna menekan kategori di GaleriPage, aplikasi akan pindah ke GaleriDetailPage sambil membawa nama kategori tersebut.

A. Pengirim Data (galeri.dart)

```
Dart

// galeri.dart

// ... di dalam _buildMenuCard

Widget _buildMenuCard({
  // ...

  required String label,
  // ...

}) {

  final String title = label.replaceAll('\n', ' ');

  return Expanded(
    child: InkWell(
      onTap: () {
        // Aksi pindah halaman
```

```

Navigator.push(
  context,
  MaterialPageRoute(
    builder: (context) => GaleriDetailPage(
      categoryTitle: title, // Mengirim data 'title' ke constructor
    ),
  ),
);
},
// ... (sisa widget)
),
);
}

```

Penjelasan:

- onTap: Saat kartu menu di-tap, Navigator.push dipanggil.
- MaterialPageRoute: Membungkus halaman baru yang akan dibuka.
- GaleriDetailPage(categoryTitle: title): Ini adalah bagian kuncinya. Kita membuat *instance* dari GaleriDetailPage dan mengisi parameternya (categoryTitle) dengan nilai title dari kartu yang ditekan.

B. Penerima Data (galeri_detail_page.dart)

Dart

```

// galeri_detail_page.dart

class GaleriDetailPage extends StatefulWidget {
  final String categoryTitle; // 1. Variabel untuk menerima data

  // 2. Constructor yang mewajibkan 'categoryTitle' diisi
  const GaleriDetailPage({Key? key, required this.categoryTitle})
    : super(key: key);

```



```
@override

_GaleriDetailPageState createState() => _GaleriDetailPageState();

}
```

```
class _GaleriDetailPageState extends State<GaleriDetailPage> {

  // 3. State untuk menyimpan daftar foto yang sudah difilter

  List<Map<String, String>> filteredList = [];
```

```
@override

void initState() {

  super.initState();

  // 4. Data yang diterima diakses melalui 'widget.categoryTitle'

  filteredList = allPhotosData

    .where((item) => item['category'] == widget.categoryTitle)

    .toList(); // Memfilter data master

}
```

```
@override

Widget build(BuildContext context) {

  return Scaffold(

    appBar: AppBar(

      // 5. Data yang diterima digunakan sebagai judul AppBar

      title: Text(widget.categoryTitle),

    ),

    body: _buildPhotoGrid(), // 6. Menampilkan data yang sudah difilter

  );

}
```

```

Widget _buildPhotoGrid() {
  // 7. GridView me-render data dari 'filteredList'

  return GridView.count(
    // ...

    children: filteredList.map((item) { // Looping data

      return _buildPropertyCard(

        imageUrl: item['imageUrl']!,

        title: item['title']!,

        location: item['location']!,

      );

    }).toList(),

  );
}

// ...
}

```

Penjelasan:

1. GaleriDetailPage (widget) memiliki variabel final String categoryTitle untuk menampung data yang dikirim.
2. Constructor-nya mewajibkan variabel ini diisi.
3. Di dalam _GaleriDetailPageState (state), kita mengakses data tersebut menggunakan widget.categoryTitle.
4. Di initState, data ini digunakan untuk memfilter daftar foto allPhotosData dan hasilnya disimpan ke *state* filteredList.
5. Fungsi build kemudian menggunakan widget.categoryTitle untuk judul AppBar dan filteredList untuk membangun GridView.

5. Tampilan Data Interaktif (ExpansionTile di Fakultas_dan_prodi.dart)

Untuk menampilkan data yang memiliki relasi (seperti Fakultas dan Program Studi), ExpansionTile adalah widget yang sangat berguna. Ini memungkinkan pengguna untuk menampilkan atau menyembunyikan detail (Prodi) dengan satu ketukan.

Dart

```
// Fakultas_dan_prodi.dart

Widget _buildFakultasCard({
  required String nama,
  required IconData icon,
  required Color color,
  required List<ProgramData> prodi,
}) {
  return Container(
    margin: EdgeInsets.only(bottom: 16),
    // ... (styling)
    child: Theme(
      data: Theme.of(context).copyWith(
        dividerColor: Colors.transparent, // Menghilangkan garis pembatas
      ),
      child: ExpansionTile( // Widget utama
        tilePadding: EdgeInsets.all(16),
        leading: Container(
          // ... (icon fakultas)
        ),
        title: Text(
          nama, // Judul yang selalu terlihat
          style: TextStyle(
            fontSize: 15,
            fontWeight: FontWeight.bold,
          ),
        ),
        subtitle: Text('${prodi.length} Program Studi'), // Subtitle
```

```

// 'children' adalah widget yang akan muncul saat di-klik
children: [

  Padding(

    padding: EdgeInsets.symmetric(horizontal: 16, vertical: 12),

    child: Column(

      crossAxisAlignment: CrossAxisAlignment.start,

      // Melakukan looping data prodi untuk ditampilkan

      children: List.generate(

        prodi.length,

        (index) => Padding(

          // ... (Widget untuk menampilkan detail prodi)

          child: Text(prodi[index].name),

        ),

      ),

    ),

  ],

);
}

```

Penjelasan:

- ExpansionTile: Widget ini secara otomatis mengelola *state* (terbuka atau tertutup).
- title, subtitle, dan leading: Ini adalah bagian yang selalu terlihat.
- children: Ini adalah List<Widget> yang hanya akan ditampilkan saat *tile* diekspansi (dibuka). Dalam kode ini, kita membuat daftar detail prodi secara dinamis menggunakan List.generate berdasarkan data prodi yang diterima.

6. Interaktivitas UI: Dialog dan Modal (home_page.dart)

Aplikasi ini menggunakan dua jenis *popup* untuk interaksi pengguna: `showDialog` untuk pencarian dan `showModalBottomSheet` untuk notifikasi.

Dart

```
// home_page.dart
```

```
// 1. Fungsi untuk menampilkan Dialog (Pop-up di tengah)
```

```
void _showSearchDialog(BuildContext context) {  
  
  showDialog(  
  
    context: context,  
  
    builder: (context) {  
  
      return Dialog( // Widget Dialog  
  
        shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(20)),  
  
        child: Container(  
  
          padding: EdgeInsets.all(20),  
  
          child: Column(  
  
            mainAxisAlignment: MainAxisAlignment.min, // Agar ukuran dialog pas  
  
            children: [  
  
              Text('Pencarian', ...),  
  
              TextField(...),  
  
              // ...  
  
            ],  
  
          ),  
  
        ),  
  
      );  
  
    },  
  
  );  
  
}
```

// 2. Fungsi untuk menampilkan Modal (Pop-up dari bawah)

```
void _showNotificationDialog(BuildContext context) {  
  
  showModalBottomSheet(  
  
    context: context,  
  
    backgroundColor: Colors.transparent, // Latar belakang transparan  
  
    builder: (context) {  
  
      return Container( // Widget kustom sebagai modal  
  
        decoration: BoxDecoration(  
  
          color: Colors.white,  
  
          borderRadius: BorderRadius.vertical(top: Radius.circular(25)),  
  
        ),  
  
        child: Column(  
  
          mainAxisAlignment: MainAxisAlignment.min, // Agar ukuran modal pas  
  
          children: [  
  
            // ... (Handle, judul, dan ListView notifikasi)  
  
            ListView(  
  
              shrinkWrap: true,  
  
              children: [  
  
                _buildNotificationItem(...),  
  
                _buildNotificationItem(...),  
  
              ],  
  
            ),  
  
          ],  
  
        ),  
  
      );  
  
    },  
  
  );  
  
}
```

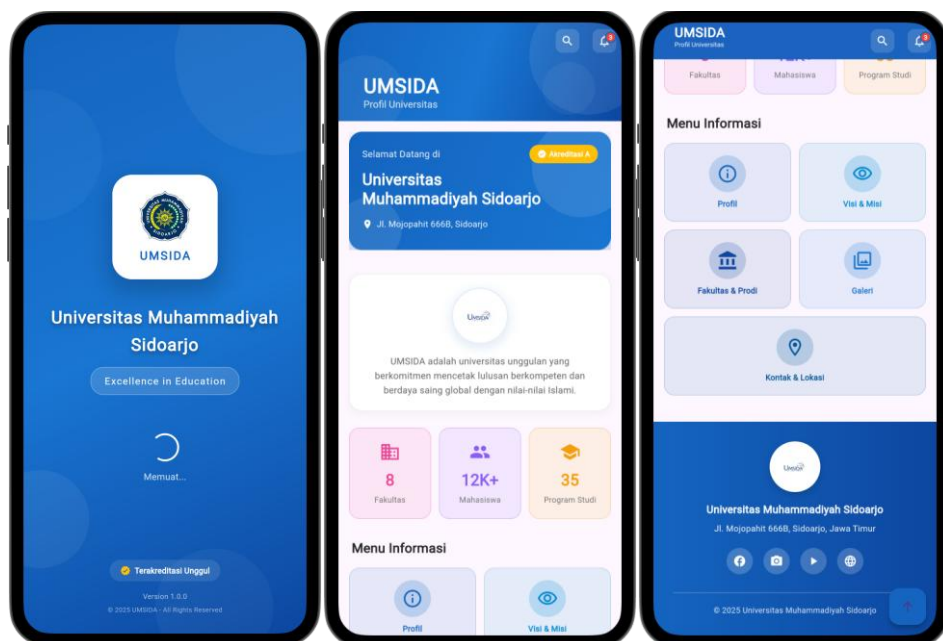
Penjelasan:

- `showDialog`: Fungsi bawaan Flutter untuk menampilkan *overlay* di tengah layar. Kode ini menggunakannya untuk menampilkan widget Dialog kustom berisi TextField pencarian.
- `showModalBottomSheet`: Fungsi bawaan Flutter untuk menampilkan *overlay* yang meluncur dari bawah layar. Ini adalah pola UI yang umum digunakan untuk notifikasi atau menu aksi tambahan.

Hasil dan Pembahasan

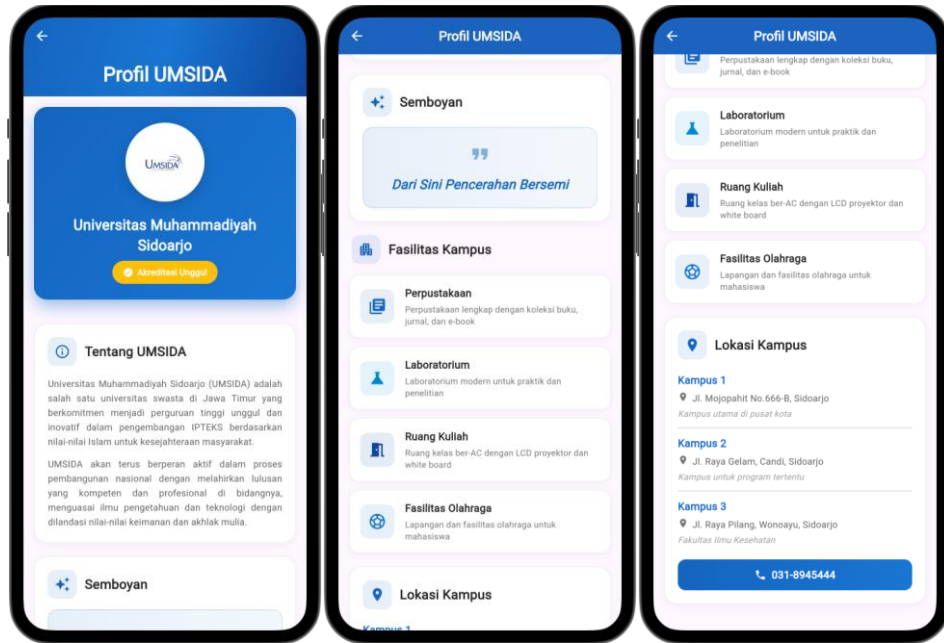
1. Splash Screen dan Home Page

Aplikasi dimulai dengan Splash Screen saat pengguna membukanya. Layar ini menampilkan logo utama universitas, dan sebuah progress bar yang memberikan indikasi proses loading. Setelah beberapa detik, aplikasi secara otomatis mengarahkan pengguna ke Home Page (Halaman Utama). Halaman utama menampilkan beberapa icon secara vertikal dalam `SingleChildScrollView`.



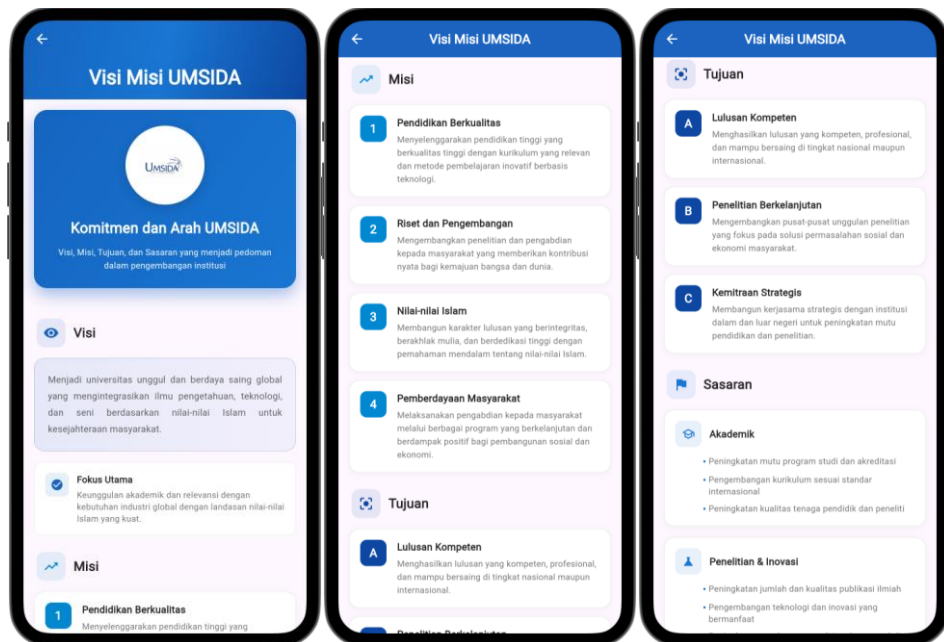
2. Halaman Profil Kampus

Ketika pengguna klik menu "Profil" pada Home Page, pengguna akan diarahkan ke halaman Profil UMSIDA. Halaman ini berisi banner profil dan deskripsi sejarah universitas. Halaman ini juga berfungsi sebagai menu navigasi sekunder, yang terdiri dari tentang UMSIDA, semboyan, dan fasilitas kampus. Terdapat juga button nomor telepon yang dapat dihubungi.



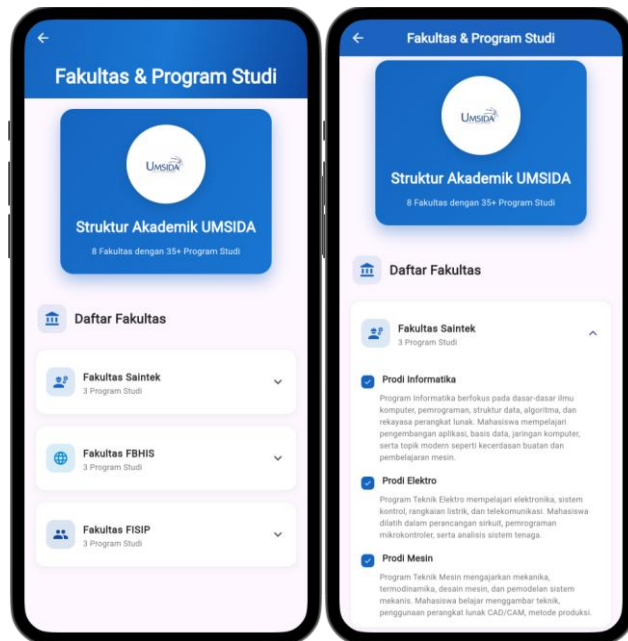
3. Halaman Visi & Misi

Halaman Visi Misi UMSIDA dapat diakses dari halaman utama. Halaman ini dirancang menggunakan widget ExpansionTile agar informasi yang padat dapat disajikan secara ringkas. Pengguna dapat melihat visi, misi, tujuan, dan sasaran. Hal ini membuat tampilan tetap bersih dan tidak membanjiri pengguna dengan teks yang terlalu panjang sekaligus.



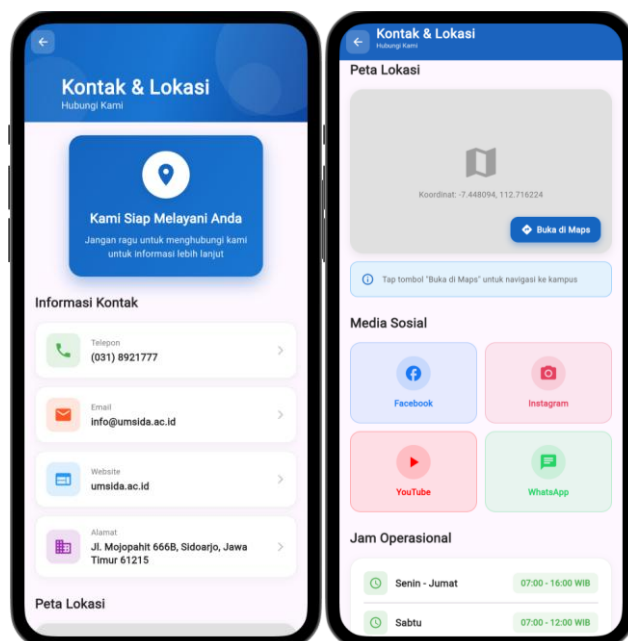
4. Halaman Fakultas & Program Studi

Pada halaman Fakultas & Program Studi, aplikasi menampilkan daftar fakultas dan program studi. Aplikasi akan menampilkan sub-daftar program studi yang ada di bawahnya beserta deskripsi singkat untuk setiap prodi.



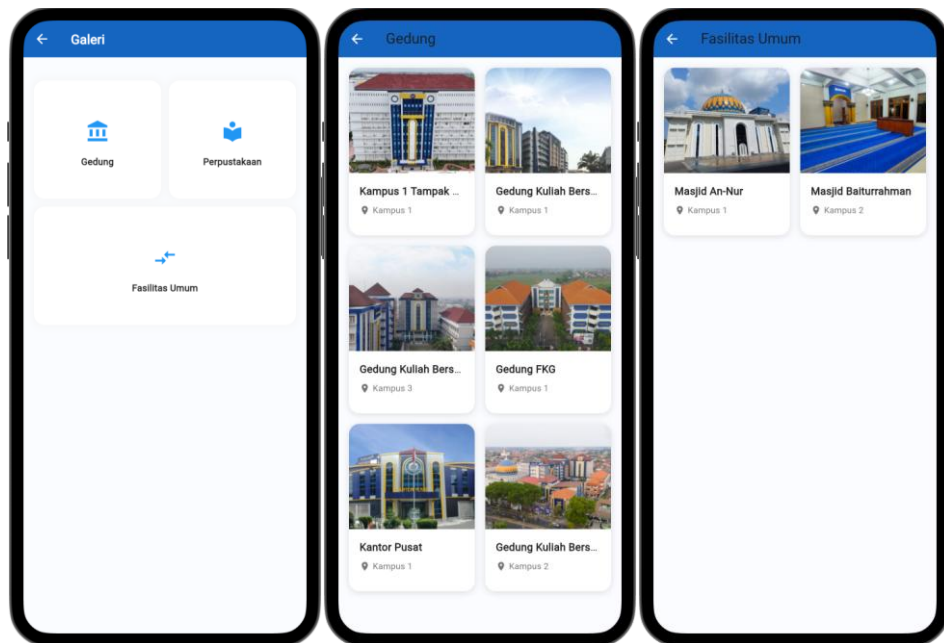
5. Halaman Kontak & Lokasi

Halaman ini dibagi menjadi beberapa bagian yaitu informasi kontak, peta lokasi, media sosial, dan jam operasional. Informasi Kontak, disajikan dalam bentuk ListView atau Column yang rapi. Jam operasional, menampilkan jadwal layanan kampus untuk hari Senin sampai Jumat dan Sabtu



6. Halaman Galeri Kampus

Halaman Galeri berisi kumpulan foto-foto. Halaman ini tidak langsung menampilkan semua foto, melainkan menyajikan tiga kategori, yaitu gedung, perpustakaan, dan fasilitas umum, memastikan foto-foto terorganisir dengan rapi.



Kesimpulan

Tuliskan kesimpulan dari hasil pengerjaan tugas kelompok ini, termasuk pengalaman yang diperoleh dan kendala yang dihadapi.

Berdasarkan hasil pengerjaan proyek, kelompok kami telah berhasil menyelesaikan pengembangan aplikasi mobile "Profil UMSIDA" menggunakan framework Flutter. Aplikasi ini berhasil diimplementasikan sesuai dengan seluruh rancangan fungsional dan visual yang telah ditetapkan, baik pada tahap wireframe (menggunakan Excalidraw) maupun pada implementasi kode.

Seluruh fitur utama yang direncanakan telah berfungsi dengan baik, mulai dari alur pembuka (Splash Screen), Halaman Utama sebagai dasbor navigasi, hingga halaman-halaman detail yang informatif (Profil, Visi Misi, Fakultas & Prodi, Galeri, dan Kontak). Aplikasi ini berhasil menyajikan informasi kampus secara terstruktur, bersih, dan konsisten.

Selama proses pengerjaan tugas kelompok ini, kami mendapatkan banyak pengalaman berharga, di antaranya:

- Kolaborasi Tim: belajar bekerja sama secara kolaboratif dalam tim yang terdiri dari 8 anggota. Setiap anggota bertanggung jawab atas *screen* atau fiturnya masing-masing dan kemudian berhasil mengintegrasikannya menjadi satu aplikasi utuh.
- Alur Kerja UI/UX: mendapatkan pengalaman praktis dalam alur kerja pengembangan aplikasi, mulai dari menerjemahkan kebutuhan *wireframe* (lo-fi) hingga menerjemahkan desain tersebut secara presisi ke dalam kode Flutter.
- Penerapan Widget Flutter: belajar menggunakan dan menggabungkan *widget-widget* layout fundamental (Column, Row, Stack, SingleChildScrollView) dan *widget* fungsional seperti GridView, Card, ExpansionTile, dan ListTile untuk membangun antarmuka yang kompleks dan interaktif.
- Navigasi Multi-Page: berhasil mengimplementasikan sistem navigasi *multi-page* menggunakan *named routes* yang diatur di main.dart, memungkinkan perpindahan antar halaman yang rapi dan terkelola.

Meskipun aplikasi berhasil diselesaikan, kami menghadapi beberapa kendala selama proses pengembangan:

- Layout Overflow : Kendala paling sering ditemui adalah *error 'bottom overflowed by pixels'* (tampilan bergaris kuning-hitam) pada halaman yang kontennya panjang, seperti Halaman Utama dan Halaman Kontak. Solusi: Kami mengatasinya dengan membungkus *widget* Column utama pada halaman-halaman tersebut menggunakan *widget* SingleChildScrollView.
- Manajemen Aset : Kami sempat mengalami kendala di mana gambar (logo, foto galeri) dan *custom font* tidak muncul saat aplikasi dijalankan. Solusi: Kendala ini terjadi karena kesalahan pengetikan *path* atau kelalaian dalam mendaftarkan direktori *assets* (cth: assets/images/) secara benar di dalam file pubspec.yaml.
- Konsistensi Desain: Dengan 8 orang yang mengerjakan file berbeda, menjaga konsistensi ukuran *font*, *padding*, dan warna menjadi tantangan.

Daftar Pustaka

1. *Flutter Documentation* (<https://flutter.dev/docs>)
2. *Website Universitas Muhammadiyah Sidoarjo* (<https://umsida.ac.id/>)
3. *Perangkat lunak yang digunakan untuk membuat wireframe / Excalidraw* (<https://excalidraw.com/>)