

**LAPORAN PRAKTIKUM  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL 7**

**NAVIGASI DAN NAVIGASI**



**Disusun Oleh :  
YOGI HAFIDH MAULANA/2211104061  
SE06-2**

**Asisten Praktikum :  
Ayu Susilowati  
Noviana Rizki Anisa Putri**

**Dosen Pengampu :  
Yudha Islami Sulistya**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2024**

## A. GUIDED

- **Model**

### Code

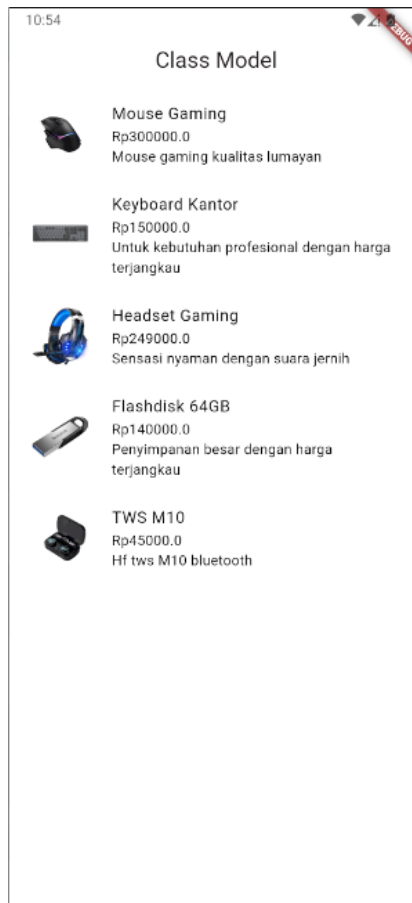
```
class Product {
    final int id;
    final String nama;
    final double harga;
    final String gambarUrl;
    final String deskripsi;

    // Constructor
    Product({
        required this.id,
        required this.nama,
        required this.harga,
        required this.gambarUrl,
        required this.deskripsi,
    });

    // Method untuk mengkonversi JSON => object product
    factory Product.fromJson(Map<String, dynamic> json) {
        return Product(
            id: json['id'],
            nama: json['nama'],
            harga: json['harga'],
            gambarUrl: json['gambarUrl'],
            deskripsi: json['deskripsi'],
        );
    }

    // Method untuk mengkonversi object product => JSON
    Map<String, dynamic> toJson() {
        return {
            'id': id,
            'nama': nama,
            'harga': harga,
            'gambarUrl': gambarUrl,
            'deskripsi': deskripsi,
        };
    }
}
```

## Output



## Deskripsi Program

Kode di atas mendefinisikan sebuah model bernama Product yang merepresentasikan produk dengan beberapa atribut penting, yaitu id, nama, harga, gambarUrl, dan deskripsi. Model ini menggunakan constructor untuk menginisialisasi atribut-atributnya dengan parameter yang diperlukan. Selain itu, terdapat dua metode utama: fromJson dan toJson. Metode fromJson berfungsi untuk mengkonversi data dari format JSON menjadi objek Product, sedangkan toJson mengkonversi objek Product kembali ke format JSON. Dengan demikian, model ini memfasilitasi interaksi antara data yang disimpan dalam format JSON dan objek dalam kode, memudahkan pengelolaan data produk dalam aplikasi.

- Navigation

## Code:

```
import 'package:flutter/material.dart';
import 'package:praktikum_7/models/product.dart';
import 'package:praktikum_7/pages/detailpage.dart';

class MyPage extends StatelessWidget {
  MyPage({super.key});

  final List<Product> products = [
    Product(
      id: 1,
      nama: 'Mouse Gaming',
      harga: 300000.00,
      gambarUrl:
        'https://resource.logitech.com/w_386,ar_1.0,c_limit,f_auto,q_auto',
      deskripsi: 'Mouse gaming kualitas lumayan'),
    Product(
      id: 2,
      nama: 'Keyboard Kantor',
      harga: 150000.00,
      gambarUrl:
        'https://resource.logitech.com/w_1600',
      deskripsi: 'Untuk kebutuhan profesional dengan harga terjangkau'),
    Product(
      id: 3,
      nama: 'Headset Gaming',
      harga: 249000.00,
      gambarUrl:
        'https://m.media-amazon.com/images/I/61CGHv6kmWL.AC_UF894,1000_QL80.jpg',
      deskripsi: 'Sensasi nyaman dengan suara jernih'),
    Product(
      id: 4,
      nama: 'Flashdisk 64GB',
      harga: 140000.00,
      gambarUrl:
        'https://www.bhphotovideo.com',
      deskripsi: 'Penyimpanan besar dengan harga terjangkau'),
    Product(
      id: 5,
      nama: 'TWS M10',
      harga: 45000.00,
      gambarUrl:
        'https://res.cloudinary.com',
      deskripsi: 'Hf tws M10 bluetooth'),
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Class Model'),
        centerTitle: true,
        backgroundColor: Colors.white, // Warna header
      ),
      body: Container(
        color: Colors.white,
        child: ListView.builder(
          itemCount: products.length,
          itemBuilder: (context, index) {
            final product = products[index];
            return Container(
              margin:
                EdgeInsets.symmetric(vertical: 5.0), // Margin antar ListTile
              child: ListTile(
                tileColor: Colors.white,
                leading: Container(
                  color: Colors.white, // Background foto produk
                  child: Image.network(
                    product.gambarUrl,
                    width: 70,
                    height: 70,
                  ),
                ),
                title: Text(
                  product.nama,
                  style: TextStyle(color: Colors.black), // Teks judul produk
                ),
                subtitle: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text(
                      'Rp${product.harga}',
                      style: TextStyle(color: Colors.black), // Teks harga
                    ),
                    Text(
                      product.deskripsi,
                      style: TextStyle(
                        color: Colors.black), // Teks deskripsi produk
                    ),
                  ],
                ),
              ),
            ),
          ),
        onTap: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (_) => DetailPage(product: product),
            ),
          );
        },
      ),
    );
  }
}
```

```

import 'package:flutter/material.dart';
import 'package:praktikum_7/models/product.dart';
import 'package:praktikum_7/systems/local_notification.dart';

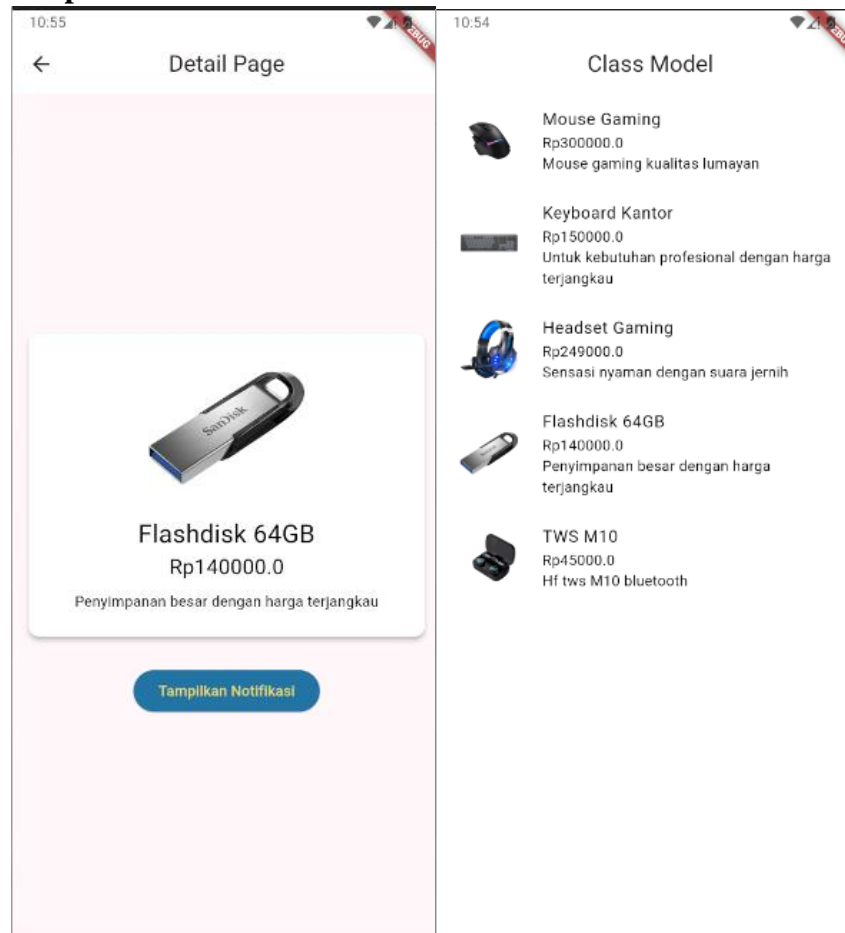
class DetailPage extends StatelessWidget {
  const DetailPage({super.key, required this.product});

  final Product product;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Detail Page'),
        centerTitle: true,
        backgroundColor: Colors.white,
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Card(
              margin: EdgeInsets.all(16.0),
              elevation: 4,
              color: Colors.white,
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(10),
              ),
              child: Padding(
                padding: const EdgeInsets.all(16.0),
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.min,
                  children: [
                    Stack(
                      alignment: Alignment.center,
                      children: [
                        Container(
                          width: double.infinity,
                          height: 150,
                          color: Colors.white10,
                        ),
                        Image.network(
                          product.gambarUrl,
                          width: 150,
                          height: 150,
                          fit: BoxFit.cover,
                        ),
                      ],
                    ),
                    SizedBox(height: 10),
                    Text(
                      product.nama,
                      style: TextStyle(
                        fontSize: 24,
                        color: Colors.black,
                      ),
                      textAlign: TextAlign.center,
                    ),
                    Text(
                      'Rp${product.harga}',
                      style: TextStyle(
                        fontSize: 20, color: Colors.black),
                    ),
                    Padding(
                      padding: const EdgeInsets.all(8.0),
                      child: Text(
                        product.deskripsi,
                        style: TextStyle(
                          color: Colors.black),
                      ),
                    ),
                  ],
                ),
              ),
            ),
            SizedBox(height: 12),
            ElevatedButton(
              onPressed: () {
                LocalNotifications.showSimpleNotification(
                  title: product.nama,
                  body: 'Harga: Rp${product.harga}\n${product.deskripsi}',
                  payload: 'Detail untuk ${product.nama}',
                );
              },
              style: ElevatedButton.styleFrom(
                backgroundColor: Color(0xFF2274A5),
              ),
              child: Text(
                'Tampilkan Notifikasi',
                style: TextStyle(color: Color(0xFFFFADF6)),
              ),
            ),
          ],
        ),
      ),
    );
  }
}

```

## Output



## Deskripsi Program

Kode di atas terdiri dari dua halaman dalam aplikasi Flutter: `MyPage` dan `DetailPage`, yang menunjukkan cara implementasi navigasi antar halaman. Pada `MyPage`, terdapat daftar produk yang ditampilkan menggunakan `ListView.builder`, di mana setiap item dalam daftar adalah `ListTile` yang mencakup gambar, nama, harga, dan deskripsi produk. Saat pengguna mengetuk salah satu item, navigasi ke `DetailPage` dilakukan dengan menggunakan `Navigator.push()`, yang mengambil konteks saat ini dan rute baru yang ingin dituju, yaitu `DetailPage`, sambil mengirimkan objek `Product` yang relevan sebagai parameter. Di `DetailPage`, informasi detail produk ditampilkan, dan terdapat tombol yang memungkinkan pengguna untuk memunculkan notifikasi menggunakan sistem notifikasi lokal. Proses ini menekankan penggunaan navigasi berbasis rute dalam Flutter untuk berpindah antar halaman sambil mempertahankan status dan informasi produk yang dipilih.

- **Notifikasi**

```
import 'package:flutter_local_notifications/flutter_local_notifications.dart';

class LocalNotifications {
  static final FlutterLocalNotificationsPlugin
    _flutterLocalNotificationsPlugin = FlutterLocalNotificationsPlugin();

  // Inisialisasi Notifikasi Lokal
  static Future init() async {
    AndroidInitializationSettings initializationSettingsAndroid =
      AndroidInitializationSettings('@mipmap/ic_launcher');
    final DarwinInitializationSettings initializationSettingsDarwin =
      DarwinInitializationSettings(
        onDidReceiveLocalNotification: (id, title, body, payload) => null,
      );
    final LinuxInitializationSettings initializationSettingsLinux =
      LinuxInitializationSettings(defaultActionName: 'Open notification');
    final InitializationSettings initializationSettings =
      InitializationSettings(
        android: initializationSettingsAndroid,
        iOS: initializationSettingsDarwin,
        linux: initializationSettingsLinux);
    _flutterLocalNotificationsPlugin.initialize(
      initializationSettings,
      onDidReceiveNotificationResponse: (details) => null,
    );
  }

  // Menampilkan notifikasi sederhana
  static Future showSimpleNotification({
    required String title,
    required String body,
    required String payload,
  }) async {
    AndroidNotificationDetails androidNotificationDetails =
      AndroidNotificationDetails(
        'your_channel_id', // Ganti dengan ID channel Anda
        'your_channel_name', // Ganti dengan nama channel Anda
        channelDescription: 'your_channel_description', // Deskripsi channel
        importance: Importance.max,
        priority: Priority.high,
        ticker: 'ticker',
        styleInformation: BigTextStyleInformation(
          body, // Teks yang ditampilkan saat diperluas
          contentType: title, // Judul notifikasi
          summaryText: 'Tap to see more', // Teks ringkasan
        ),
      );

    NotificationDetails notificationDetails =
      NotificationDetails(android: androidNotificationDetails);

    await _flutterLocalNotificationsPlugin.show(
      0,
      title,
      body,
      notificationDetails,
      payload: payload,
    );
  }
}
```

## Output



## Deskripsi program

Kode di atas terdiri dari dua bagian utama: `DetailPage`, yang menampilkan rincian produk, dan kelas `LocalNotifications`, yang mengelola pengaturan dan tampilan notifikasi lokal. Di dalam `DetailPage`, saat pengguna menekan tombol "Tampilkan Notifikasi," fungsi `showSimpleNotification` dari kelas `LocalNotifications` dipanggil untuk menampilkan notifikasi. Kelas `LocalNotifications` menggunakan paket `flutter_local_notifications` untuk menginisialisasi pengaturan notifikasi di berbagai platform (Android, iOS, dan Linux). Dalam metode `showSimpleNotification`, notifikasi disiapkan dengan rincian seperti judul, isi, dan payload, serta mengatur tampilan dan prioritas notifikasi menggunakan `AndroidNotificationDetails`. Ketika notifikasi ditampilkan, pengguna dapat melihat informasi tentang produk, termasuk harga dan deskripsi, dengan opsi untuk mengklik notifikasi dan melihat detail lebih lanjut. Dengan demikian, kode ini menunjukkan bagaimana mengimplementasikan notifikasi lokal untuk memberikan umpan balik kepada pengguna tentang produk dalam aplikasi Flutter.



## B. Unguided

### 1. Soal 1

#### Code

Class model

```
class Product {  
  final String id;  
  final String name;  
  final String description;  
  final double price;  
  final String imageUrl;  
  
  Product({  
    required this.id,  
    required this.name,  
    required this.description,  
    required this.price,  
    required this.imageUrl,  
  });  
}
```

Cartpage

```
import 'package:flutter/material.dart';  
  
class CartPage extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(title: Text('Keranjang')),  
      body: Center(  
        child: Text('Belum ada produk di keranjang'),  
      ),  
    );  
  }  
}
```

## Homepage

```
import 'package:flutter/material.dart';
import 'package:praktikum/models/product.dart';
import 'package:praktikum/pages/detailpage.dart';

class HomePage extends StatelessWidget {
  final List<Product> products = [
    Product(
      id: '1',
      name: 'Air Slamet',
      description: 'Air yang berasal dari gunung slamet',
      price: 10000,
      imageUrl:
        'https://tastysnack.id/cdn/shop/files/CRYSTALINAIRMINERALBTL600mL_1024x.jpg?v=1704773917',
    ),
    Product(
      id: '2',
      name: 'Air Janah',
      description: 'Air dari pegunungan Evresh',
      price: 4000,
      imageUrl:
        'https://www.static-src.com/wcsstore/Indraprastha/images/catalog/full//98/MTA-3759227/aqua_air-mineral-aqua-600-ml-botol_full02.jpg',
    ),
    // Tambahkan lebih banyak produk jika diperlukan
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Center(child: Text('Daftar Produk')),
        backgroundColor: Colors.white,
      ),
      body: Container(
        color: Colors.white,
        child: ListView.builder(
          itemCount: products.length,
          itemBuilder: (context, index) {
            final product = products[index];
            return Card(
              elevation: 4,
              margin: const EdgeInsets.symmetric(vertical: 10),
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(15),
              ),
              child: InkWell(
                onTap: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(
                      builder: (context) => DetailPage(product: product),
                    ),
                  );
                },
              ),
            );
          },
          child: Padding(
            padding: const EdgeInsets.all(10.0),
            child: Row(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                ClipRRect(
                  borderRadius: BorderRadius.circular(10),
                  child: Image.network(
                    product.imageUrl,
                    width: 80,
                    height: 80,
                    fit: BoxFit.cover,
                  ),
                ),
                SizedBox(width: 15),
                Expanded(
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      Text(
                        product.name,
                        style: TextStyle(
                          fontSize: 18,
                          fontWeight: FontWeight.bold,
                          color: Colors.black87,
                        ),
                      ),
                      SizedBox(height: 5),
                      Text(
                        product.description,
                        maxLines: 2,
                        overflow: TextOverflow.ellipsis,
                        style: TextStyle(
                          fontSize: 14,
                          color: Colors.black54,
                        ),
                      ),
                      SizedBox(height: 10),
                      Text(
                        'Rp${product.price}',
                        style: TextStyle(
                          fontSize: 16,
                          fontWeight: FontWeight.w600,
                          color: Colors.blueAccent,
                        ),
                      ),
                    ],
                  ),
                ),
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```

## detailpage

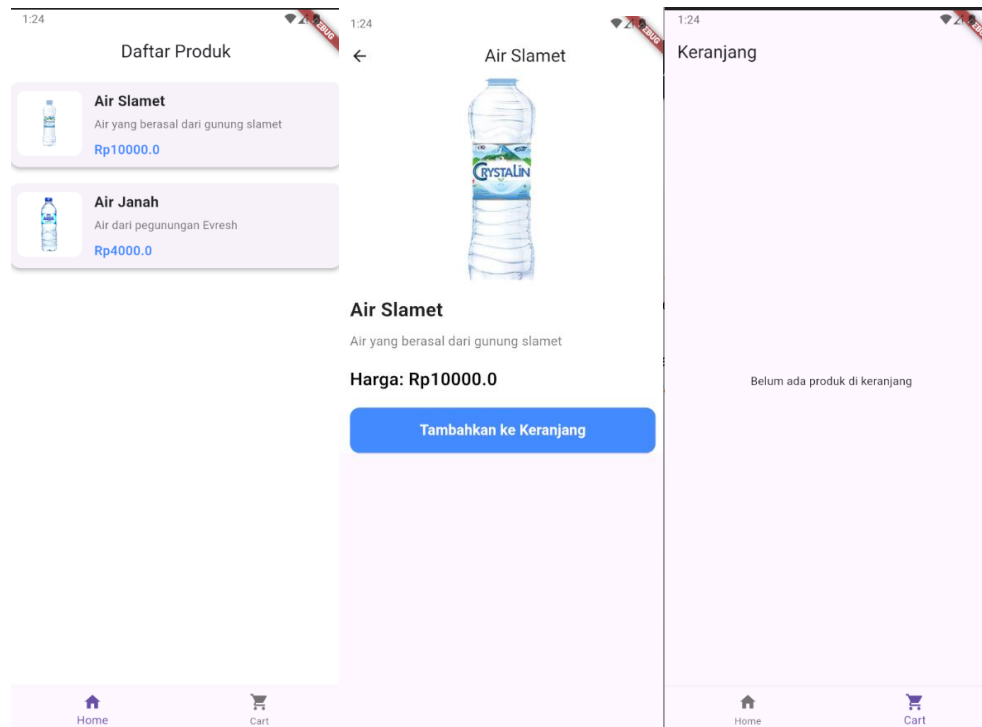
```
import 'package:flutter/material.dart';
import 'package:praktikum/models/product.dart';

class DetailPage extends StatelessWidget {
  final Product product;

  DetailPage({required this.product});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Center(child: Text(product.name)),
        backgroundColor: Colors.white,
      ),
      body: SingleChildScrollView(
        child: Container(
          color: Colors.white,
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              // Gambar produk
              ClipRect(
                borderRadius: BorderRadius.only(
                  bottomLeft: Radius.circular(20),
                  bottomRight: Radius.circular(20),
                ),
                child: Image.network(
                  product.imageUrl,
                  width: double.infinity,
                  height: 250,
                  fit: BoxFit.cover,
                ),
              ),
              SizedBox(height: 20),
              // Informasi produk
              Padding(
                padding: const EdgeInsets.symmetric(horizontal: 16.0),
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text(
                      product.name,
                      style: TextStyle(
                        fontSize: 24,
                        fontWeight: FontWeight.bold,
                        color: Colors.black87,
                      ),
                    ),
                    SizedBox(height: 10),
                    Text(
                      product.description,
                      style: TextStyle(
                        fontSize: 16,
                        color: Colors.black54,
                      ),
                    ),
                    SizedBox(height: 20),
                    Text(
                      'Harga: Rp${product.price}',
                      style: TextStyle(
                        fontSize: 22,
                        fontWeight: FontWeight.w600,
                        color: Colors.black,
                      ),
                    ),
                  ],
                ),
              ),
              SizedBox(height: 20),
              // Tombol Tambahkan ke Keranjang
              Padding(
                padding: const EdgeInsets.symmetric(horizontal: 16.0),
                child: ElevatedButton(
                  onPressed: () {
                    // Logika menambahkan ke keranjang
                  },
                  style: ElevatedButton.styleFrom(
                    padding: EdgeInsets.symmetric(vertical: 15),
                    backgroundColor: Colors.blueAccent,
                    shape: RoundedRectangleBorder(
                      borderRadius: BorderRadius.circular(12),
                    ),
                  ),
                  child: Center(
                    child: Text(
                      'Tambahkan ke Keranjang',
                      style: TextStyle(
                        fontSize: 18,
                        fontWeight: FontWeight.bold,
                        color: Colors.white,
                      ),
                    ),
                  ),
                ),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

## Output



## Deskripsi Program

Pada halaman utama (HomePage), aplikasi menampilkan daftar produk berupa gambar, nama, dan harga. Setiap item dapat diklik untuk membuka halaman detail (DetailPage) yang memperlihatkan informasi lebih lengkap tentang produk, termasuk gambar produk, deskripsi, dan harga. Pengguna juga bisa menambahkan produk ke keranjang melalui tombol yang tersedia di halaman detail tersebut. CartPage berfungsi sebagai halaman yang menampilkan produk yang telah ditambahkan ke keranjang belanja.

Pada HomePage, daftar produk ditampilkan dalam bentuk Card yang terstruktur, masing-masing menampilkan gambar, nama, dan harga produk. ListView.builder digunakan untuk menampilkan daftar produk secara dinamis sesuai dengan jumlah produk yang ada. Setiap produk ditampilkan dengan gambar yang dilengkapi dengan sudut melengkung, serta nama dan harga yang diberi penekanan dengan gaya teks tebal. InkWell digunakan pada setiap Card untuk menambahkan efek ripple ketika pengguna mengetuk produk, memberikan umpan balik visual yang membuat interaksi terasa lebih alami. Ketika produk diklik, pengguna diarahkan ke DetailPage yang menampilkan informasi lengkap tentang produk.

Pada DetailPage, aplikasi menampilkan informasi produk yang lebih lengkap. Di bagian atas, terdapat gambar produk dengan sudut melengkung pada bagian bawahnya, yang diatur untuk memenuhi lebar layar sehingga lebih menonjol. Di bawah gambar, ditampilkan nama produk dengan teks berukuran besar dan tebal, deskripsi produk dalam ukuran yang lebih kecil dan warna abu-abu yang lembut, serta harga produk yang ditampilkan dalam ukuran lebih besar dan berwarna biru. Pada bagian bawah halaman, terdapat tombol "Tambahkan ke Keranjang" yang besar dan mudah diakses, dengan padding luas dan warna yang sesuai dengan tema, memberi pengguna akses mudah untuk menambahkan produk ke keranjang belanja.

CartPage adalah halaman yang dirancang untuk menampilkan produk yang telah ditambahkan ke keranjang belanja pengguna. Saat ini, halaman ini berfungsi sebagai placeholder dengan teks yang menunjukkan bahwa keranjang belanja masih kosong. CartPage dapat dikembangkan lebih lanjut untuk menampilkan daftar produk di keranjang, total harga, dan tombol untuk menyelesaikan pembelian atau menghapus produk dari keranjang.