

**LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL VII
NAVIGASI DAN NOTIFIKASI**



**Disusun Oleh :
Devrin Anggun Saputri / 2211104001
SE0601**

**Asisten Praktikum :
Muhammad Faza Zulian Gesit Al Barru
Aisyah Hasna Aulia**

**Dosen Pengampu :
Yudha Islami Sulistya, S.Kom., M.Cs.**

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

GUIDED

1. Model

Secara umum, hampir semua aplikasi yang dikembangkan berinteraksi dengan data. Data dalam aplikasi dapat memiliki berbagai bentuk, tergantung pada jenis aplikasi yang dibuat. Agar data yang diterima atau dikirimkan memiliki konsistensi, sangat penting untuk menerapkan standar yang seragam. Memelihara proyek yang kompleks menjadi sangat sulit tanpa adanya model yang jelas.

a. Membuat Model Class

Pada file `product.dart` berisi Class `Product` sebagai tempat untuk membuat objek produk.

```
1 class Product {
2   final int id;
3   final String nama;
4   final double harga;
5   final String imageUrl;
6   final String deskripsi;
7
8   // constructor
9   Product({
10    required this.id,
11    required this.nama,
12    required this.harga,
13    required this.imageUrl,
14    required this.deskripsi,
15  });
16
17  // method untuk mengkonversi JSON ke objek product
18  factory Product.fromJson(Map<String, dynamic> json) {
19    return Product(
20      id: json['id'],
21      nama: json['nama'],
22      harga: json['harga'].toDouble(),
23      imageUrl: json['imageUrl'],
24      deskripsi: json['deskripsi']
25    );
26  }
27
28  // method untuk mengkonversi object product ke JSON
29  Map<String, dynamic> toJson() {
30    return {
31      'id': id,
32      'nama': nama,
33      'harga': harga,
34    };
35  }
36 }
```

Pada file `“my.package.dart”` terdapat list yang berisi beberapa objek “Produk” dan menyimpan data berupa id, nama, harga, imageurl, dan deskripsi untuk setiap produk.

```
8 final List<Product> products = [
9   Product(
10    id: 1,
11    nama: 'Mouse',
12    harga: 300000,
13    imageUrl: 'https://resource.logitech.com/...',
14    deskripsi: 'Mouse keren andalan Mahasiswa',
15  ),
16  Product(
17    id: 2,
18    nama: 'Keyboard Mechanical',
19    harga: 350000,
20    imageUrl: 'https://resource.logitech.com/...',
21    deskripsi: 'Nyaman dipakai, Garansi 5 tahun',
22  ),
23  Product(
24    id: 3,
25    nama: 'Headphone',
26    harga: 700000,
27    imageUrl: 'https://m.media-amazon.com/imag...',
28    deskripsi: 'Garansi 5 tahun, Rekomendasi M...',
29  ),
30 ];
```

2. Navigation

Dalam Flutter, navigation merujuk pada cara berpindah antar halaman (atau tampilan) di aplikasi. Sistem navigasi di Flutter berbasis route dan navigator. Setiap halaman atau layar di Flutter disebut sebagai route, dan Navigator adalah widget yang mengelola stack dari route tersebut.

a. main.dart

```

1 import 'package:flutter/material.dart';
2 import 'package:praktikum_7/pages/mypage.dart';
3
4 void main() => runApp(const MyApp());
5
6 class MyApp extends StatelessWidget {
7   const MyApp({super.key});
8
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      title: 'Material App',
13      home: Scaffold(
14        appBar: AppBar(
15          title: const Text('Material App Bar'),
16        ),
17        body: Mypage(),
18      ),
19    );
20  }
21 }

```

b. mypackage.dart

c. product.dart

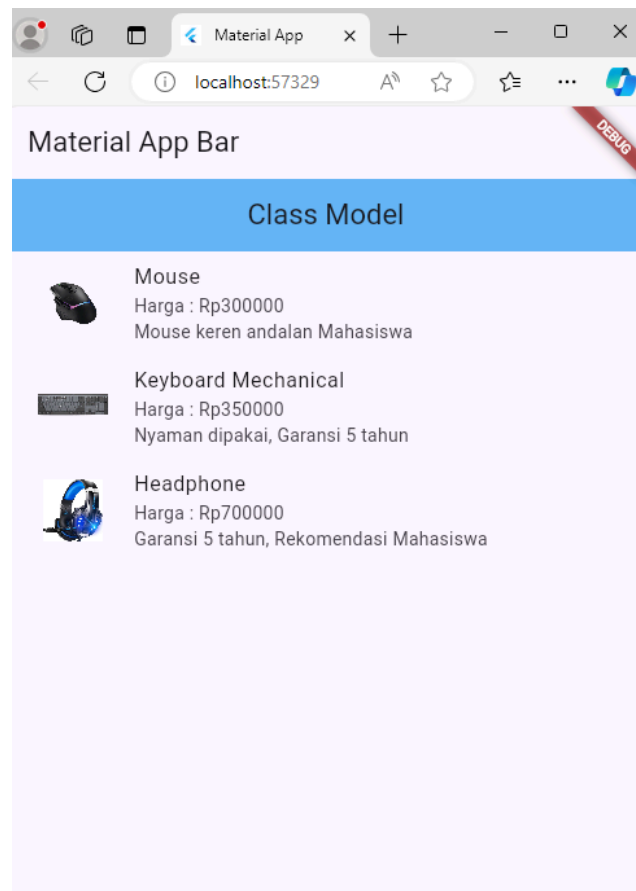
```
1 class Product {
2   final int id;
3   final String nama;
4   final double harga;
5   final String imageUrl;
6   final String deskripsi;
7
8   // constructor
9   Product({
10     required this.id,
11     required this.nama,
12     required this.harga,
13     required this.imageUrl,
14     required this.deskripsi,
15   });
16
17   // method untuk mengkonversi JSON ke objek product
18   factory Product.fromJson(Map<String, dynamic> json) {
19     return Product(
20       id: json['id'],
21       nama: json['nama'],
22       harga: json['harga'].toDouble(),
23       imageUrl: json['imageUrl'],
24       deskripsi: json['deskripsi']
25     );
26   }
27
28   // method untuk mengkonversi object product ke JSON
29   Map<String, dynamic> toJson() {
30     return {
31       'id': id,
32       'nama': nama,
33       'harga': harga,
34     };
35   }
36 }
```

d. detail.dart

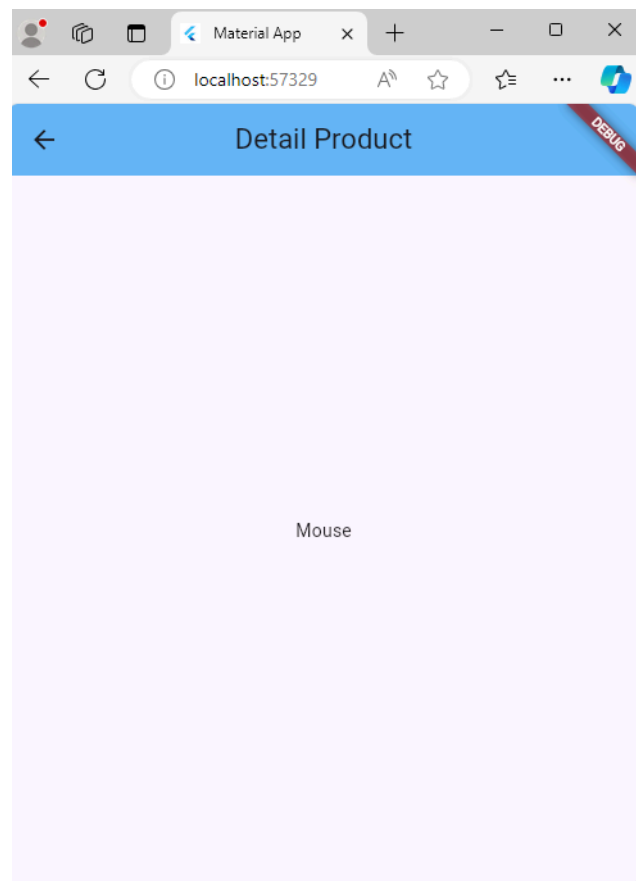
```
1 import 'package:flutter/material.dart';
2
3 class DetailProduct extends StatelessWidget {
4   const DetailProduct({super.key, required this.data});
5
6   final String data;
7
8   @override
9   Widget build(BuildContext context) {
10     return Scaffold(
11       appBar: AppBar(
12         title: Text('Detail Product'),
13         centerTitle: true,
14         backgroundColor: Colors.blue[300],
15       ),
16       body: Center(
17         child: Text(data),
18       ),
19     );
20   }
21 }
```

e. Output

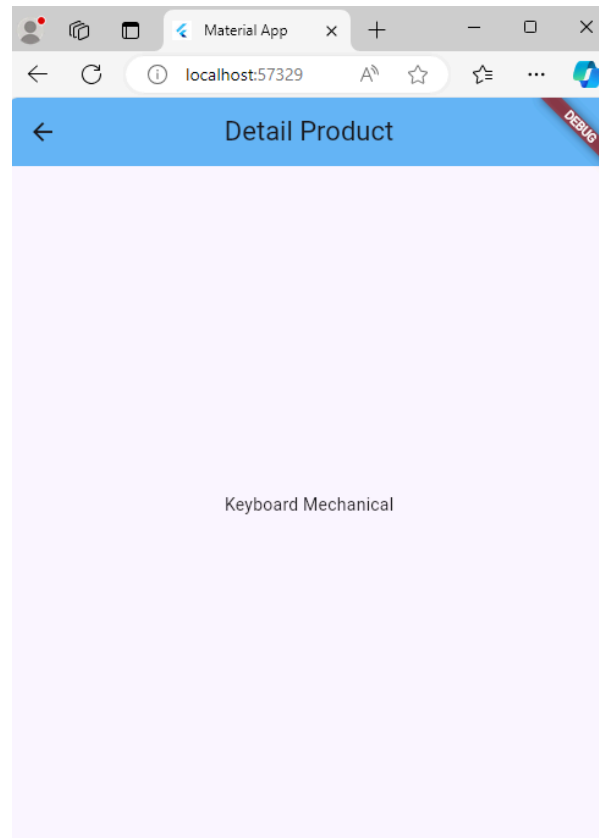
- Halaman 1



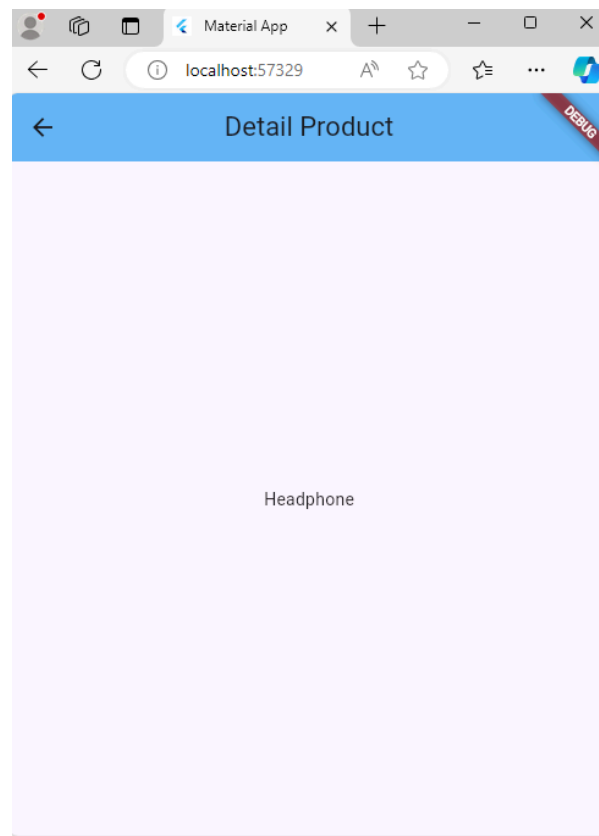
- Halaman 2



- **Halaman 3**



- **Halaman 4**



3. Notification

- a. Untuk mengirimkan notifikasi dalam aplikasi Flutter, dapat digunakan package bernama `flutter_local_notifications`. Tambahkan terlebih dahulu package tersebut ke dalam `pubspec.yaml`.

```
30 dependencies:
31   flutter:
32     sdk: flutter
33   cupertino_icons: ^1.0.8
34   flutter_local_notifications: ^17.2.4
35
```

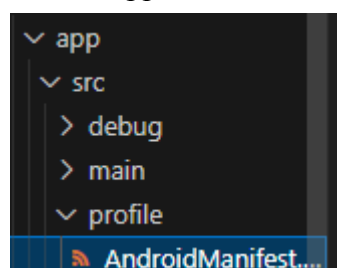
- b. Lalu unduh dan install package tersebut di terminal.

```
[navigasi_notifikasi] flutter pub [Terminal (Ctrl+)] ple
Resolving dependencies...
Downloading packages...
+ args 2.6.0
+ async 2.11.0 (2.12.0 available)
+ boolean_selector 2.1.1 (2.1.2 available)
+ characters 1.3.0 (1.3.1 available)
+ clock 1.1.1 (1.1.2 available)
+ collection 1.18.0 (1.19.1 available)
+ dbus 0.7.10
+ fake_async 1.3.1 (1.3.2 available)
+ ffi 2.1.3
+ flutter_lints 4.0.0 (5.0.0 available)
+ flutter_local_notifications 17.2.4 (18.0.0 available)
+ flutter_local_notifications_linux 4.0.1 (5.0.0 available)
+ flutter_local_notifications_platform_interface 7.2.0 (8.0.0 available)
+ leak_tracker 10.0.5 (10.0.8 available)
+ leak_tracker_flutter_testing 3.0.5 (3.0.9 available)
+ lints 4.0.0 (5.1.0 available)
+ matcher 0.12.16+1 (0.12.17 available)
+ material_color_utilities 0.11.1 (0.12.0 available)
+ meta 1.15.0 (1.16.0 available)
+ path 1.9.0 (1.9.1 available)
+ petitparser 6.0.2
+ plugin_platform_interface 2.1.8
+ stack_trace 1.11.1 (1.12.0 available)
+ string_scanner 1.2.0 (1.4.0 available)
+ test_api 0.7.2 (0.7.3 available)
+ timezone 0.9.4
+ vm_service 14.2.5 (14.3.1 available)
+ xdg_directories 1.1.0
+ xml 6.5.0
Changed 11 dependencies!
```

- c. Kemudian konfigurasi Android di dalam direktori `android/app/build.gradle` untuk mengatur versi `minSdk Version` ke 21 atau yang lebih tinggi.

```
22 defaultConfig {
23     // TODO: Specify your own unique Application
24     applicationId = "com.example.navigasi_no
25     // You can update the following values t
26     // For more information, see: https://fl
27     minSdk = flutter.minSdkVersion 21
```

- d. Menambahkan izin getar (bersifat opsional) di dalam direktori `android/app/AndroidManifest.xml`.



Tambahkan code:

```
<uses-permission android:name="android.permission.VIBRATE" />
```

```
pubspec.yaml build.gradle AndroidManifest.xml X
android > app > src > profile > AndroidManifest.xml
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android">
2 <!-- The INTERNET permission is required for development. Specifically,
3 | the Flutter tool needs it to communicate with the running application
4 | to allow setting breakpoints, to provide hot reload, etc.
5 | -->
6 <uses-permission android:name="android.permission.INTERNET"/>
7 <!-- Menambahkan izin getar (opsional) -->
8 <uses-permission android:name="android.permission.VIBRATE" />
9 </manifest>
10
```

Jika notifikasi ingin tetap aktif, tambahkan izin berikut:

```
<use-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
```

```
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android">
2 <!-- The INTERNET permission is required for development. Specifically,
3 | the Flutter tool needs it to communicate with the running application
4 | to allow setting breakpoints, to provide hot reload, etc.
5 | -->
6 <uses-permission android:name="android.permission.INTERNET"/>
7 <!-- Menambahkan izin getar (opsional) -->
8 <uses-permission android:name="android.permission.VIBRATE" />
9 <!-- Agar notifikasi tetap aktif setelah perangkat di-reboot -->
10 <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
11 </manifest>
12
```

- e. Inisialisasi flutter_local_notifications dengan menambahkan inisialisasi di main.dart.

```
1 import 'package:flutter/material.dart';
2 import 'package:flutter_local_notifications/flutter_local_notifications.dart';
3
4 // Buat instance
5 final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =
6   FlutterLocalNotificationsPlugin();
7
8 void main() async {
9   WidgetsFlutterBinding.ensureInitialized();
10
11   // Pengaturan inisialisasi untuk Android
12   const AndroidInitializationSettings initializationSettingsAndroid =
13     AndroidInitializationSettings('@mipmap/ic_launcher');
14
15   // Pengaturan inisialisasi untuk iOS
16   const DarwinInitializationSettings initializationSettingsIOS =
17     DarwinInitializationSettings();
18
19   // Menggabungkan semua pengaturan platform
20   const InitializationSettings initializationSettings = InitializationSettings(
21     android: initializationSettingsAndroid,
22     iOS: initializationSettingsIOS,
23   );
24
25   // Inisialisasi plugin dengan pengaturan
26   await flutterLocalNotificationsPlugin.initialize(
27     initializationSettings,
28   );
29
30   runApp(MyApp());
31 }
32
33 // Langkah 4 - Fungsi untuk menampilkan notifikasi
34 Future<void> showNotification() async {
35   const AndroidNotificationDetails androidPlatformChannelSpecifics =
36     AndroidNotificationDetails(
37       'your_channel_id', // ID unik untuk channel
38       'your_channel_name', // Nama channel
39       channelDescription: 'your_channel_description', // Deskripsi channel
40       importance: Importance.max,
41       priority: Priority.high,
42       showWhen: true,
43     );
44
45   const NotificationDetails platformChannelSpecifics = NotificationDetails(
46     android: androidPlatformChannelSpecifics,
47   );
48
49   await flutterLocalNotificationsPlugin.show(
50     0, // ID notifikasi
51     'Hello!', // Judul notifikasi
52     'This is a local notification.', // Isi notifikasi
53     platformChannelSpecifics,
54   );
55 }
56
57 class MyApp extends StatelessWidget {
58   @override
59   Widget build(BuildContext context) {
60     return MaterialApp(
61       title: 'Flutter Local Notifications',
62       theme: ThemeData(primarySwatch: Colors.blue),
63       home: NotificationScreen(),
64     );
65   }
66 }
67
68 // Langkah 5: Buat UI dengan Tombol untuk Menampilkan Notifikasi
69 class NotificationScreen extends StatelessWidget {
70   @override
71   Widget build(BuildContext context) {
72     return Scaffold(
73       appBar: AppBar(title: Text("Local Notifications")),
74       body: Center(
75         child: ElevatedButton(
76           onPressed:
77             showNotification, // Memanggil fungsi notifikasi saat tombol ditekan
78           child: Text("Show Notification"),
79         ),
80       ),
81     );
82   }
83 }
84
```

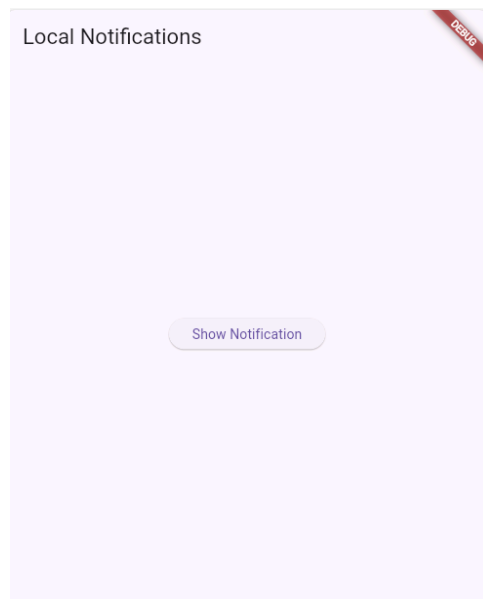

- f. Kemudian pada file main.dart, buat fungsi untuk menampilkan notifikasi dengan menambahkan kode berikut:

```
1 // Langkah 4 - Fungsi untuk menampilkan notifikasi
2 Future<void> showNotification() async {
3   const AndroidNotificationDetails androidPlatformChannelSpecifics =
4     AndroidNotificationDetails(
5       'your_channel_id', // ID unik untuk channel
6       'your_channel_name', // Nama channel
7       channelDescription: 'your_channel_description', // Deskripsi channel
8       importance: Importance.max,
9       priority: Priority.high,
10      showWhen: true,
11    );
12
13   const NotificationDetails platformChannelSpecifics = NotificationDetails(
14     android: androidPlatformChannelSpecifics,
15   );
16
17   await flutterLocalNotificationsPlugin.show(
18     0, // ID notifikasi
19     'Hello!', // Judul notifikasi
20     'This is a local notification.', // Isi notifikasi
21     platformChannelSpecifics,
22   );
23 }
24
25 class MyApp extends StatelessWidget {
26   @override
27   Widget build(BuildContext context) {
28     return MaterialApp(
29       title: 'Flutter Local Notifications',
30       theme: ThemeData(primarySwatch: Colors.blue),
31       home: NotificationScreen(),
32     );
33   }
34 }
```

- g. Lalu buat UI dengan tombol untuk menampilkan notifikasi dengan kode berikut:

```
1 // Langkah 5: Buat UI dengan Tombol untuk Menampilkan Notifikasi
2 class NotificationScreen extends StatelessWidget {
3   @override
4   Widget build(BuildContext context) {
5     return Scaffold(
6       appBar: AppBar(title: Text("Local Notifications")),
7       body: Center(
8         child: ElevatedButton(
9           onPressed:
10             showNotification, // Memanggil fungsi notifikasi saat tombol ditekan
11             child: Text("Show Notification"),
12         ),
13       ),
14     );
15   }
16 }
17
```

h. Output



UNGUIDED

Buatlah satu project untuk menampilkan beberapa produk dan halaman *e-commerce* dengan menerapkan class model serta navigasi halaman.

Note: Jangan lupa sertakan source code, screenshoot output, dan deskripsi program. Kreativitas menjadi nilai tambah.

Source Code:

a. main.dart

```
1 import 'package:flutter/material.dart';
2 import 'screens/product_homepage.dart';
3
4 void main() {
5   runApp(MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   const MyApp({super.key});
10
11   @override
12   Widget build(BuildContext context) {
13     return MaterialApp(
14       title: 'E-Commerce App',
15       theme: ThemeData(
16         primarySwatch: Colors.blue,
17       ),
18       home: HomeScreen(),
19     );
20   }
21 }
```

b. product.dart

```
1 import 'package:flutter/material.dart';
2
3 class Product {
4   final String id;
5   final String title;
6   final String description;
7   final double price;
8   final String imageUrl;
9   final Color? backgroundColor; // Membuatnya opsional
10
11   Product({
12     required this.id,
13     required this.title,
14     required this.description,
15     required this.price,
16     required this.imageUrl,
17     this.backgroundColor, // Tidak wajib
18   });
19 }
```

c. home screen.dart

```
import 'package:ecommerce_navigation_notification/screens/product_detail_page.dart';
import 'package:flutter/material.dart';
import '../models/product.dart';

class HomeScreen extends StatelessWidget {
  final List<Product> products = [
    Product(
      id: '1',
      title: 'Basreng Pedas',
      description:
        'Basreng, singkatan dari "bakso goreng," adalah camilan khas Sunda yang terbuat dari bakso ikan atau bakso sapi',
      price: 12000,
      imageUrl:
        'https://akcdn.detik.net.id/community/media/visual/2022/01/06/bakso-gorengbasreng_43.jpeg?w=700&q=90',
    ),
    Product(
      id: '2',
      title: 'Mie Lidi',
      description:
        'Mie lidi adalah jajanan khas Indonesia yang terbuat dari adonan tepung sagu dan tepung terigu, dibentuk menyeru',
      price: 11000,
      imageUrl:
        'https://images.tokopedia.net/img/cache/500-square/VqbcmM/2021/1/29/ccf283ce-4b80-41b9-bbbb-a1c7ec352361.jpg.webp',
    ),
    Product(
      id: '3',
      title: 'Keripik Cireng',
      description:
        'Keripik cireng pedas adalah camilan khas Indonesia yang terbuat dari adonan tepung tapioka yang dibentuk pipih',
      price: 10000,
      imageUrl:
        'https://cdn.orderonline.id/uploads/5272361660557311691-original.png',
    ),
    Product(
      id: '4',
      title: 'Makaroni Pedas',
      description:
        'Makaroni pedas adalah camilan lezat yang terbuat dari makaroni yang digoreng hingga renyah, kemudian dibumbui d',
      price: 13500,
      imageUrl:
        'https://i.gojekapi.com/darkroom/gofood-indonesia/v2/images/uploads/4a4a467b-ffbf-4419-b862-fe5627e9e46c_4e6af63',
    ),
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Camilanku'),
        centerTitle: true,
        backgroundColor: const Color.fromARGB(255, 230, 154, 13)), // AppBar
      body: GridView.builder(
        padding: const EdgeInsets.all(10),
        gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
```

```

1      crossAxisCount: 2,
2      childAspectRatio: 0.75,
3      crossAxisSpacing: 10,
4      mainAxisSpacing: 10,
5    ),
6    itemCount: products.length,
7    itemBuilder: (ctx, i) {
8      return GestureDetector(
9        onTap: () {
10          Navigator.of(context).push(
11            MaterialPageRoute(
12              builder: (ctx) => ProductDetailScreen(product: products[i]),
13            ),
14          );
15        },
16        child: Container(
17          decoration: BoxDecoration(
18            color: products[i]
19              .backgroundColor, // Warna latar belakang untuk kontainer produk
20            borderRadius: BorderRadius.circular(10),
21          ),
22          child: GridTile(
23            child: ClipRect(
24              borderRadius: BorderRadius.circular(10),
25              child: Image.network(products[i].imageUrl, fit: BoxFit.cover),
26            ),
27            footer: Container(
28              color: products[i].backgroundColor ??
29                const Color.fromARGB(255, 230, 154, 13), // Terapkan backgroundColor pada footer
30              child: GridTileBar(
31                backgroundColor: Colors.transparent, // Set agar transparan
32                title: Text(
33                  products[i].title,
34                  textAlign: TextAlign.center,
35                  style: TextStyle(
36                    color: Colors.black, // Warna teks agar kontras
37                  ),
38              ),
39            ),
40          ),
41        ),
42      );
43    },
44  ),
45 );
46 }
47 }

```

d. product_detail_screen.dart

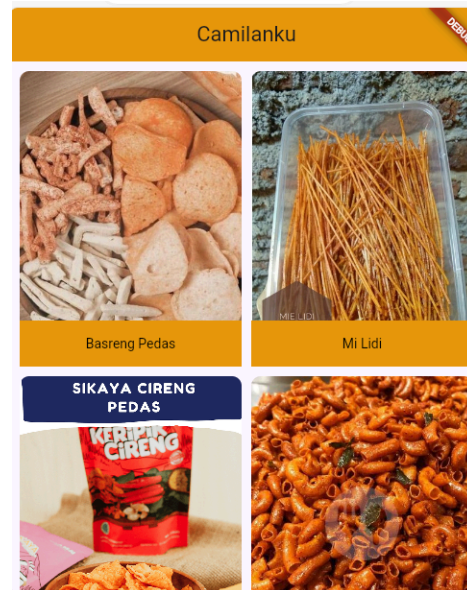
```

1 import 'package:flutter/material.dart';
2 import '../models/product.dart';
3
4 class ProductDetailScreen extends StatelessWidget {
5   final Product product;
6
7   ProductDetailScreen({required this.product});
8
9   @override
10  Widget build(BuildContext context) {
11    return Scaffold(
12      appBar: AppBar(
13        title: Text(product.title),
14        backgroundColor: const Color.fromARGB(255, 230, 154, 13), // Sesuaikan dengan warna utama aplikasi
15      ),
16      body: SingleChildScrollView(
17        child: Column(
18          crossAxisAlignment:
19            CrossAxisAlignment.center, // Mengatur posisi ke tengah
20          children: <Widget>[
21            Padding(
22              padding: const EdgeInsets.all(
23                16.0), // Kotak dengan padding di sekitar gambar
24            ),
25            child: Container(
26              decoration: BoxDecoration(
27                borderRadius: BorderRadius.circular(10),
28                color: Colors.grey[200], // Warna latar belakang kotak gambar
29              ),
30              padding: const EdgeInsets.all(8.0),
31              child: ClipRect(
32                borderRadius: BorderRadius.circular(10),
33                child: Image.network(product.imageUrl),
34              ),
35            ),
36            SizedBox(height: 10),
37            Text(
38              'Rp ${product.price}',
39              style: TextStyle(
40                fontSize: 24,
41                fontWeight: FontWeight.bold,
42                color: const Color.fromARGB(255, 223, 167, 102),
43              ),
44            ),
45            SizedBox(height: 20),
46            Padding(
47              padding: const EdgeInsets.symmetric(horizontal: 16.0),
48              child: Text(
49                product.description,
50                textAlign:
51                  TextAlign.center, // Mengatur teks deskripsi ke tengah
52                style: TextStyle(fontSize: 16),
53              ),
54            ),
55          ],
56        ),
57      );
58    }
59  }
60 }

```

e. Output

- Home Screen



- Detail Produk




- Detail Produk 2

A wooden bowl filled with golden-brown, crispy fried snacks (keripik cireng) sits on a wooden surface. In the background, there are bags of the product, one pink and one red.

Rp 10000

Keripik cireng pedas adalah camilan khas Indonesia yang terbuat dari adonan tepung tapioka yang dibentuk pipih dan digoreng hingga garing. Camilan ini dikenal dengan rasa pedas yang menggugah selera, biasanya ditambahkan dengan bumbu cabai dan rempah-rempah yang membuatnya semakin nikmat.

A close-up shot of a bowl of spicy macaroni (makaroni pedas). The macaroni is coated in a thick, orange-red sauce and garnished with green herbs.

Rp 13500

Makaroni pedas adalah camilan lezat yang terbuat dari makaroni yang digoreng hingga renyah, kemudian dibumbui dengan berbagai rempah dan cabai untuk memberikan rasa pedas yang menggugah selera.

Deskripsi Program:

Program ini bertujuan untuk mengimplementasikan penggunaan class model dan navigasi halaman dalam menampilkan produk camilan pedas. Setiap produk diwakili oleh kelas "**Product**" yang memiliki atribut seperti id, title, description, price, dan imageUrl, yang berguna untuk menyimpan informasi lengkap tentang setiap produk. Daftar produk ini kemudian ditampilkan pada "**Home Screen**" dalam format grid.

Ketika pengguna memilih salah satu produk, mereka akan diarahkan ke halaman detail produk menggunakan fungsi **navigator.push**. Halaman ini menyajikan informasi produk secara lebih rinci, termasuk gambar, harga, dan deskripsi singkat. Pengguna juga dapat kembali ke halaman utama dengan menekan tombol kembali di bagian kiri atas layar.