

**PRAKTIKUM PEMROGRAMAN PERANGKAT BERGERAK
TUGAS GUIDED & UNGUIDED**

**MODUL V
ANTARMUKA PENGGUNA LANJUTAN**



Disusun Oleh :

Nadia Putri Rahmaniar / 2211104012

S1 SE-06-01

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 beroperasi dengan data, dan data dalam aplikasi memiliki berbagai bentuk, bergantung pada jenis aplikasinya. Akan lebih baik jika setiap data yang diterima atau dikirim mengikuti standar yang sama, dan pemeliharaan proyek yang kompleks hampir mustahil dilakukan tanpa adanya model. Sangat bermanfaat jika setiap pengiriman dan penerimaan data dilakukan dengan standar yang sama. Tanpa model, hampir mustahil untuk memelihara proyek yang kompleks.

a. Membuat Model Class

Mulailah dengan membuat direktori baru di dalam folder lib proyek Flutter. Kemudian, beri nama file kelas Dart sesuai dengan data yang akan digunakan untuk membuat model. Misalnya, ketika Anda membuat model untuk "user" dengan tanggapan JSON berikut:

```
1 Product(  
2   id: 1,  
3   nama: 'Chunky Hand-Knit Pink Bag',  
4   harga: 300000.00,  
5   imageUrl:  
6     'https://lapetitecrocheterie.com/cdn/shop/files/IMG_2537.jpg?v=1686520598&width=713',  
7   deskripsi:  
8     'Perfectly handmade using tube yarn, the chunky purse measures 10''wide and 8''tall.',  
9 ),
```

Kemudian buatlah file seperti berikut:

```
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   Product({  
9     required this.id,  
10    required this.nama,  
11    required this.harga,  
12    required this.imageUrl,  
13    required this.deskripsi,  
14  });  
15  
16  factory Product.fromJson(Map<String, dynamic> json) {  
17    return Product(  
18      id: json['id'],  
19      nama: json['nama'],  
20      harga: json['harga'].toDouble(),  
21      imageUrl: json['imageUrl'],  
22      deskripsi: json['deskripsi'],  
23    );  
24  }  
25  
26  Map<String, dynamic> toJson() {  
27    return {  
28      'id': id,  
29      'nama': nama,  
30      'harga': harga,  
31      'imageUrl': imageUrl,  
32      'deskripsi': deskripsi,  
33    };  
34  }  
35 }
```

2. Navigation

Navigasi dalam Flutter mengacu pada cara berpindah antar halaman atau tampilan dalam aplikasi. Dalam sistem navigasi Flutter, istilah "route" mengacu pada setiap halaman atau layar Flutter, dan "navigator" adalah widget yang mengatur tumpukan rute tersebut.

a. main.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:flutter_local_notifications/flutter_local_notifications.dart';
3 import 'package:pertemuan_7/pages/mypage.dart';
4
5 final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =
6     FlutterLocalNotificationsPlugin();
7
8 Future<void> initNotifications() async {
9     const AndroidInitializationSettings initializationSettingsAndroid =
10         AndroidInitializationSettings('@mipmap/ic_launcher');
11
12     const InitializationSettings initializationSettings = InitializationSettings(
13         android: initializationSettingsAndroid,
14     );
15
16     await flutterLocalNotificationsPlugin.initialize(
17         initializationSettings,
18         onDidReceiveNotificationResponse: (NotificationResponse response) async {
19             print('Notification clicked with payload: ${response.payload}');
20         },
21     );
22 }
23
24 Future<void> showNotification({
25     required String title,
26     required String body,
27     required String payload,
28 }) async {
29     const AndroidNotificationDetails androidPlatformChannelSpecifics =
30         AndroidNotificationDetails(
31             'product_channel',
32             'Product Notifications',
33             importance: Importance.max,
34             priority: Priority.high,
35             showWhen: true,
36         );
37
38     const NotificationDetails platformChannelSpecifics =
39         NotificationDetails(android: androidPlatformChannelSpecifics);
40
41     await flutterLocalNotificationsPlugin.show(
42         0,
43         title,
44         body,
45         platformChannelSpecifics,
46         payload: payload,
47     );
48 }
49
50 void main() async {
51     WidgetsFlutterBinding.ensureInitialized();
52     await initNotifications();
53     runApp(const MyApp());
54 }
```

```

55
56 class MyApp extends StatelessWidget {
57   const MyApp({super.key});
58
59   @override
60   Widget build(BuildContext context) {
61     return MaterialApp(
62       debugShowCheckedModeBanner: false,
63       title: 'Flutter Demo',
64       theme: ThemeData(
65         colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
66         useMaterial3: true,
67       ),
68       home: MyPage(),
69     );
70   }
71 }
72

```

b. mypage.dart

```

1 import 'package:flutter/material.dart';
2 import 'package:pertemuan_7/models/product.dart';
3 import 'package:pertemuan_7/pages/detail.dart';
4 import 'package:pertemuan_7/main.dart'; // Import main.dart untuk menggunakan showNotification
5
6 class MyPage extends StatelessWidget {
7   MyPage({super.key});
8
9   // Daftar produk yang akan ditampilkan
10  final List<Product> products = [
11    Product(
12      id: 1,
13      nama: 'Mouse',
14      harga: 300000.00,
15      imageUrl:
16        'https://resource.logitech.com/w_386_ar_1.0_c_16mil_f_auto_q_auto_dpr_2.0/d_transparent.gif/content/dam/logitech/en/products/g502x-plus/gallery/g502x-plus-gallery-1-black.png?v=1',
17      deskripsi: 'Mouse gaming berkualitas tinggi untuk menunjang kerjamu',
18    ),
19    Product(
20      id: 2,
21      nama: 'Keyboard',
22      harga: 350000.00,
23      imageUrl:
24        'https://resource.logitech.com/w_1600_c_list_q_auto_f_auto_dpr_1.0/d_transparent.gif/content/dam/logitech/en/products/keyboards/mx-mechanical/gallery/mx-mechanical-keyboard-top-view-graphite-us.png?v=1&quot',
25      deskripsi: 'Keyboard mekanik untuk gaming dengan LED yang menyala',
26    ),
27    Product(
28      id: 3,
29      nama: 'Headphone Gaming',
30      harga: 700000.00,
31      imageUrl:
32        'https://m.media-amazon.com/images/I/61GCPvKvKwM.AC_UF894,1000-Q188.jpg',
33      deskripsi:
34        'Headphone dengan kualitas bass yang kuat dan suara yang tinggi',
35    ),
36    // Produk lainnya bisa ditambahkan di sini
37  ];
38
39  @override
40  Widget build(BuildContext context) {
41    return Scaffold(
42      appBar: AppBar(
43        title: const Text('Class Model'),
44        centerTitle: true,
45        backgroundColor: const Color.fromARGB(255, 195, 195, 174, 255),
46      ),
47      body: ListView.builder(
48        itemCount: products.length,
49        itemBuilder: (context, index) {
50          final product = products[index];
51          return ListTile(
52            leading: Image.network(product.imageUrl, width: 50, height: 50),
53            title: Text(product.nama),
54            subtitle: Text(product.deskripsi),
55            trailing: Text('Rp ${product.harga.toStringAsFixed(2)}'),
56            onTap: () async {
57              // Menampilkan log saat notifikasi dipanggil
58              print('Notifikasi untuk ${product.nama} dipanggil');
59
60              // Menampilkan notifikasi saat produk ditekan
61              await showNotification(
62                title: 'Produk Dipilih',
63                body: 'Anda memilih produk ${product.nama}',
64                payload: product.id.toString(),
65              );
66
67              // Navigasi ke halaman detail produk
68              Navigator.push(
69                context,
70                MaterialPageRoute(
71                  builder: (_) => DetailProduct(product: product),
72                ),
73              );
74            },
75          );
76        },
77      ),
78    );
79  }
80 }
81

```

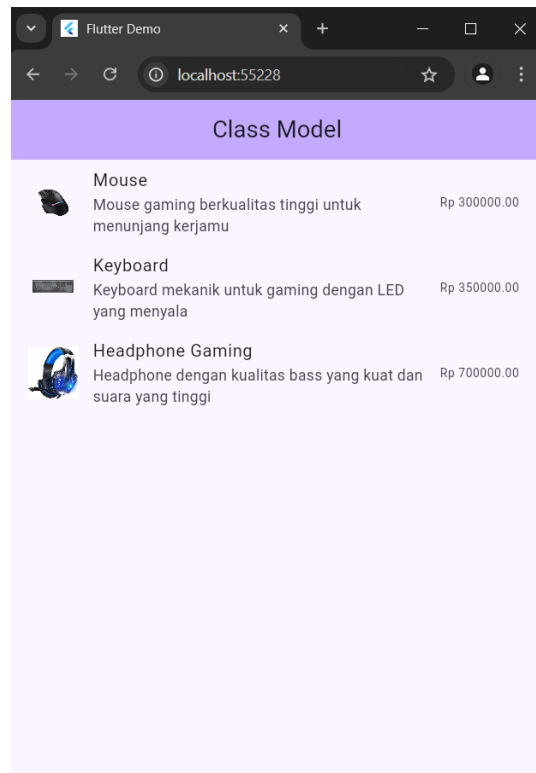
c. product.dart

```
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 mengonversi JSON ke object 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 mengonversi object Product ke JSON
29   Map<String, dynamic> toJson() {
30     return {
31       'id': id,
32       'nama': nama,
33       'harga': harga,
34       'imageUrl': imageUrl,
35       'deskripsi': deskripsi,
36     };
37   }
38 }
```

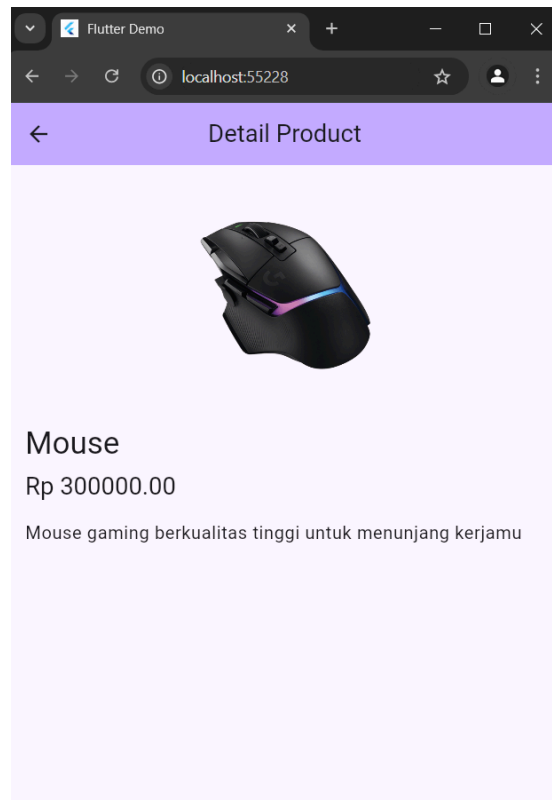
d. detail.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:pertemuan_7/models/product.dart'; // Tambahkan import Product model
3
4 class DetailProduct extends StatelessWidget {
5   const DetailProduct(
6     {super.key, required this.product}); // Ubah dari data ke product
7
8   final Product product; // Ubah tipe data dari String ke Product
9
10  @override
11  Widget build(BuildContext context) {
12    return Scaffold(
13      appBar: AppBar(
14        title: Text("Detail Product"),
15        centerTitle: true,
16        backgroundColor: const Color.fromARGB(255, 195, 174, 255),
17      ),
18      body: Padding(
19        padding: const EdgeInsets.all(16.0),
20        child: Column(
21          crossAxisAlignment: CrossAxisAlignment.start,
22          children: [
23            Image.network(
24              product.imageUrl,
25              height: 200,
26              width: double.infinity,
27              fit: BoxFit.contain,
28            ),
29            SizedBox(height: 16),
30            Text(
31              product.nama,
32              style: Theme.of(context).textTheme.headlineMedium,
33            ),
34            SizedBox(height: 8),
35            Text(
36              'Rp ${product.harga.toStringAsFixed(2)}',
37              style: Theme.of(context).textTheme.titleLarge,
38            ),
39            SizedBox(height: 16),
40            Text(
41              product.deskripsi,
42              style: Theme.of(context).textTheme.bodyLarge,
43            ),
44          ],
45        ),
46      ),
47    );
48  }
49 }
50
```

e. Output home




f. Output detail produk



3. Notification

- a. Untuk mengirim notifikasi di aplikasi Flutter, Anda dapat menggunakan paket "flutter_local_notifications", yang tersedia di situs pub.dev, dan kemudian dimasukkan ke dalam file "pubspec.yaml."




```
1  cupertino_icons: ^1.0.8
2  flutter_local_notifications: ^17.2.4
3  permission_handler: ^11.3.1
4
5  dev_dependencies:
6    flutter_test:
7      sdk: flutter
```

- b. Pastikan minSdkVersion di android/app/build.gradle diatur ke 21 atau lebih tinggi:



```
1  minSdkVersion 21
```

- c. Tambahkan izin pada file AndroidManifest.xml



```
1  <!-- Izin untuk menghidupkan kembali notifikasi setelah reboot -->
2  <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
3  <!-- Izin untuk mengaktifkan getaran pada notifikasi -->
4  <uses-permission android:name="android.permission.VIBRATE" />
5  <uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
```


d. File notification.dart

```
1 // lib/notification_page.dart
2 import 'dart:async';
3 import 'package:flutter/material.dart';
4 import 'package:flutter_local_notifications/flutter_local_notifications.dart';
5
6 void main() => runApp(MaterialApp(
7   theme: ThemeData(
8     appBarTheme: const AppBarTheme(
9       color: Colors.amber,
10     ),
11   ),
12   home: const MyApp(),
13   debugShowCheckedModeBanner: false,
14 ));
15
16 class MyApp extends StatefulWidget {
17   const MyApp({Key? key}) : super(key: key);
18
19   @override
20   _MyAppState createState() => _MyAppState();
21 }
22
23 class _MyAppState extends State<MyApp> {
24   final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =
25     FlutterLocalNotificationsPlugin();
26
27   @override
28   void initState() {
29     super.initState();
30     _initializeNotifications();
31   }
32
33   Future<void> _initializeNotifications() async {
34     const AndroidInitializationSettings initializationSettingsAndroid =
35       AndroidInitializationSettings('@mipmap/ic_launcher');
36
37     const DarwinInitializationSettings initializationSettingsIOS =
38       DarwinInitializationSettings();
39
40     const InitializationSettings initSettings = InitializationSettings(
41       android: initializationSettingsAndroid,
42       iOS: initializationSettingsIOS,
43     );
44
45     await flutterLocalNotificationsPlugin.initialize(
46       initSettings,
47       onDidReceiveNotificationResponse: (NotificationResponse response) async {
48         await _onSelectNotification(response.payload);
49       },
50     );
51   }
52
53   Future<void> _onSelectNotification(String? payload) async {
54     if (payload != null) {
55       Navigator.of(context).push(MaterialPageRoute(
56         builder: (_) => NewScreen(payload: payload),
57       ));
58     }
59   }
60 }
```

```

61   @override
62   Widget build(BuildContext context) {
63     return Scaffold(
64       appBar: AppBar(
65         backgroundColor: Colors.amber,
66         title: const Text('Flutter Notification Demo'),
67       ),
68       body: Center(
69         child: Column(
70           mainAxisAlignment: MainAxisAlignment.center,
71           children: <Widget>[
72             SizedBox(
73               width: 250.0,
74               child: ElevatedButton(
75                 style: ElevatedButton.styleFrom(
76                   backgroundColor: Colors.blueAccent,
77                 ),
78                 onPressed: _showNotification,
79                 child: const Text('Show Notification'),
80               ),
81             ),
82           ],
83         ),
84       ),
85     );
86   }
87
88   Future<void> _showNotification() async {
89     const AndroidNotificationDetails androidPlatformChannelSpecifics =
90       AndroidNotificationDetails(
91         'id',
92         'channel',
93         importance: Importance.max,
94         priority: Priority.high,
95       );
96
97     const DarwinNotificationDetails iOSPlatformChannelSpecifics =
98       DarwinNotificationDetails();
99
100    const NotificationDetails platformChannelSpecifics = NotificationDetails(
101      android: androidPlatformChannelSpecifics,
102      iOS: iOSPlatformChannelSpecifics,
103    );
104
105    await flutterLocalNotificationsPlugin.show(
106      0,
107      'Flutter Devs',
108      'Flutter Local Notification Demo',
109      platformChannelSpecifics,
110      payload: 'Welcome to the Local Notification demo',
111    );
112  }
113 }
114
115 class NewScreen extends StatelessWidget {
116   final String payload;
117
118   const NewScreen({
119     Key? key,
120     required this.payload,
121   }) : super(key: key);
122

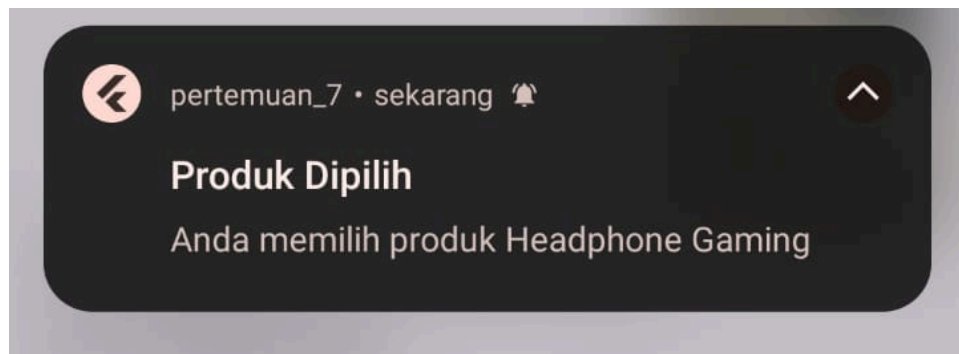
```

```

123   @override
124   Widget build(BuildContext context) {
125     return Scaffold(
126       appBar: AppBar(
127         title: Text(payload),
128       ),
129       body: Center(
130         child: Text('Notification Payload: $payload'),
131       ),
132     );
133   }
134 }
135

```

e. Output contoh notification



UNGUIDED

Buatlah satu project untuk menampilkan beberapa produk dan halaman ecommerce dengan menerapkan class model serta navigasi halaman.

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

1. Source Code

a. main.dart

```
1 import 'package:flutter/material.dart';
2 import './pages/home.dart';
3
4 void main() {
5   runApp(MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      debugShowCheckedModeBanner: false,
13      title: 'Nadia E-Commerce',
14      theme: ThemeData(primarySwatch: Colors.orange),
15      home: ProductListPage(),
16    );
17  }
18 }
```

b. home.dart

```
1 import 'package:flutter/material.dart';
2 import '../models/product.dart';
3 import 'product_detail.dart';
4
5 class ProductListPage extends StatelessWidget {
6   final List<Product> products = [
7     Product(
8       id: 1,
9       nama: 'Chunky Hand-Knit Pink Bag',
10      harga: 300000.00,
11      imageUrl:
12        'https://lapetitecrocheterie.com/cdn/shop/files/IMG_2537.jpg?v=1686520598&width=713',
13      deskripsi:
14        'Perfectly handmade using tube yarn, the chunky purse measures 10''wide and 8''tall.',
15    ),
16    Product(
17      id: 2,
18      nama: 'Chunky Hand-Knit Pink Bag',
19      harga: 300000.00,
20      imageUrl:
21        'https://lapetitecrocheterie.com/cdn/shop/files/IMG_2537.jpg?v=1686520598&width=713',
22      deskripsi:
23        'Perfectly handmade using tube yarn, the chunky purse measures 10''wide and 8''tall.',
24    ),
25    Product(
26      id: 3,
27      nama: 'Chunky Hand-Knit Pink Bag',
28      harga: 300000.00,
29      imageUrl:
30        'https://lapetitecrocheterie.com/cdn/shop/files/IMG_2537.jpg?v=1686520598&width=713',
31      deskripsi:
32        'Perfectly handmade using tube yarn, the chunky purse measures 10''wide and 8''tall.',
33    ),
34    Product(
35      id: 4,
36      nama: 'Chunky Hand-Knit Pink Bag',
37      harga: 300000.00,
38      imageUrl:
39        'https://lapetitecrocheterie.com/cdn/shop/files/IMG_2537.jpg?v=1686520598&width=713',
40      deskripsi:
41        'Perfectly handmade using tube yarn, the chunky purse measures 10''wide and 8''tall.',
42    ),
43  ];
44
45  @override
46  Widget build(BuildContext context) {
47    return Scaffold(
48      appBar: AppBar(title: Text('La Petite Crocheterie')),
49      body: GridView.builder(
50        padding: const EdgeInsets.all(8.0),
51        itemCount: products.length,
52        gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
53          crossAxisCount: 2,
54          childAspectRatio: 0.8,
55        ),
56        itemBuilder: (ctx, i) => ProductCard(products[i]),
57      ),
58    );
59  }
```

```


59     }
60 }
61
62 class ProductCard extends StatelessWidget {
63   final Product product;
64
65   ProductCard(this.product);
66
67   @override
68   Widget build(BuildContext context) {
69     return Card(
70       color: Colors.grey.shade200,
71       shape: RoundedRectangleBorder(
72         borderRadius: BorderRadius.circular(8),
73       ),
74       child: Column(
75         crossAxisAlignment: CrossAxisAlignment.start,
76         children: [
77           ClipRect(
78             borderRadius: BorderRadius.vertical(top: Radius.circular(8)),
79             child: Image.network(
80               product.imageUrl,
81               height: 200,
82               width: double.infinity,
83               fit: BoxFit.cover,
84             ),
85           ),
86           Padding(
87             padding: const EdgeInsets.all(6.0),
88             child: Column(
89               crossAxisAlignment: CrossAxisAlignment.start,
90               children: [
91                 Text(
92                   product.nama,
93                   style: TextStyle(
94                     fontWeight: FontWeight.bold,
95                     fontSize: 14,
96                   ),
97                 ),
98                 Text(
99                   'Rp ${product.harga.toStringAsFixed(2)}',
100                 style: TextStyle(
101                   color: Colors.black,
102                   fontSize: 12,
103                 ),
104               ),
105               SizedBox(height: 10), // Memberikan jarak antara teks dan tombol
106             Row(
107               children: [
108                 Expanded(
109                   child: ElevatedButton(
110                     onPressed: () {
111                       Navigator.of(context).push(
112                         MaterialPageRoute(
113                           builder: (context) => ProductDetailPage(product),
114                         ),
115                     ),
116                   ),
117                   child: Text('Shop Now'),
118                   style: ElevatedButton.styleFrom(
119                     shape: RoundedRectangleBorder(
120                       borderRadius: BorderRadius.circular(
121                         5), // Mengatur border radius
122                     ),
123                   ),
124                 ),
125               ],
126             ),
127           ],
128         ),
129       ),
130     ),
131   ],
132 );
133 }
134 }
135 }

```

c. product_detail.dart

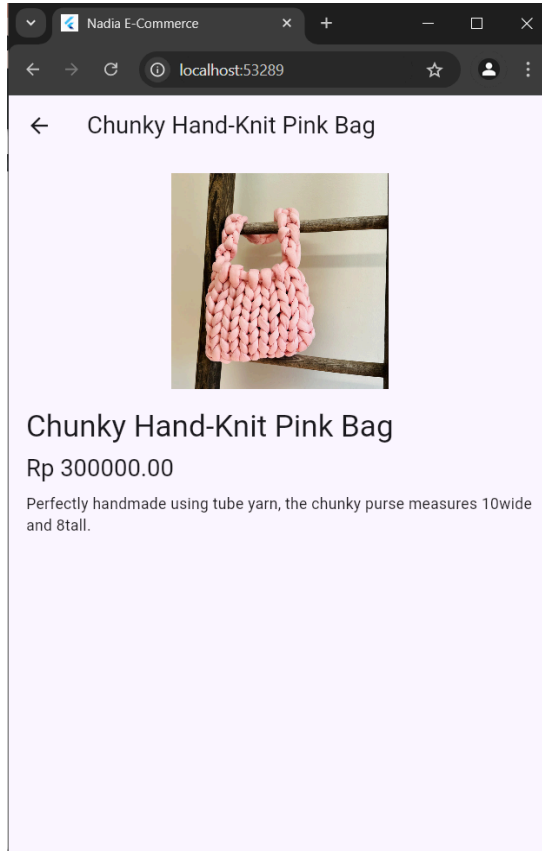
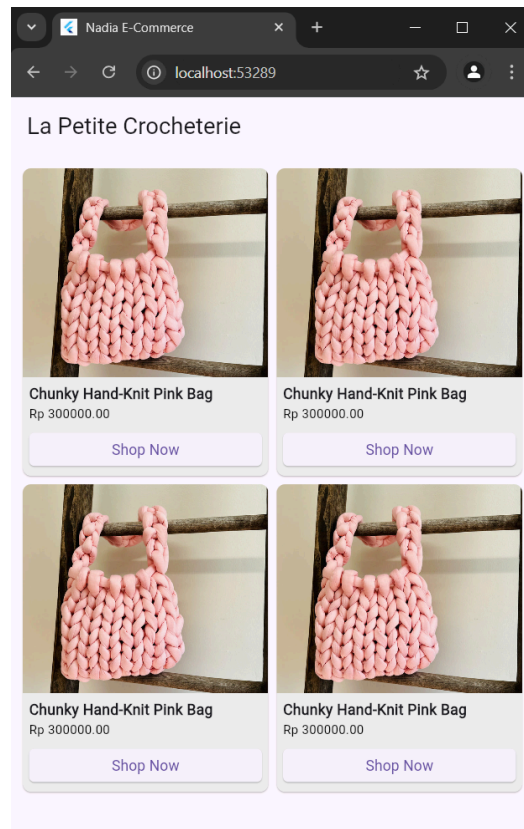
```
1 import 'package:flutter/material.dart';
2 import '../models/product.dart';
3
4 class ProductDetailPage extends StatelessWidget {
5   final Product product;
6
7   ProductDetailPage(this.product);
8
9   @override
10  Widget build(BuildContext context) {
11    return Scaffold(
12      appBar: AppBar(title: Text(product.nama)),
13      body: Padding(
14        padding: const EdgeInsets.all(16.0),
15        child: Column(
16          crossAxisAlignment: CrossAxisAlignment.start,
17          children: [
18            Image.network(
19              product.imageUrl,
20              height: 200,
21              width: double.infinity,
22              fit: BoxFit.contain,
23            ),
24            SizedBox(height: 16),
25            Text(
26              product.nama,
27              style: Theme.of(context).textTheme.headlineMedium,
28            ),
29            SizedBox(height: 8),
30            Text(
31              'Rp ${product.harga.toStringAsFixed(2)}',
32              style: Theme.of(context).textTheme.titleLarge,
33            ),
34            SizedBox(height: 8),
35            Text(product.deskripsi),
36          ],
37        ),
38      ),
39    );
40  }
41 }
```

d. 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    Product({
9      required this.id,
10     required this.nama,
11     required this.harga,
12     required this.imageUrl,
13     required this.deskripsi,
14   });
15
16   factory Product.fromJson(Map<String, dynamic> json) {
17     return Product(
18       id: json['id'],
19       nama: json['nama'],
20       harga: json['harga'].toDouble(),
21       imageUrl: json['imageUrl'],
22       deskripsi: json['deskripsi'],
23     );
24   }
25
26   Map<String, dynamic> toJson() {
27     return {
28       'id': id,
29       'nama': nama,
30       'harga': harga,
31       'imageUrl': imageUrl,
32       'deskripsi': deskripsi,
33     };
34   }
35 }
```


e. Screenshot Output



2. Deskripsi Program

Aplikasi ini merupakan platform e-commerce yang memungkinkan pengguna melihat daftar produk dalam tata letak grid, dilengkapi dengan gambar, nama, harga, dan deskripsi singkat. Setiap produk dapat diklik untuk membuka halaman detail yang memuat informasi lebih rinci, termasuk gambar produk yang lebih besar, deskripsi lengkap, dan harga.

- a. **main.dart:** Fungsi main digunakan untuk menjalankan aplikasi Flutter. Di dalam kelas MyApp, aplikasi ini mengatur judul dan tema dengan skema warna biru. Halaman awal aplikasi ditetapkan ke ProductListPage, yang merupakan halaman utama untuk menampilkan daftar produk.
- b. **product_detail.dart:** Berisi kelas ProductDetailPage yang berfungsi menampilkan detail produk ketika pengguna mengklik produk dari daftar. Halaman ini mencakup AppBar yang menampilkan nama produk dan konten utama yang terdiri dari gambar produk, nama, harga, dan deskripsi.
- c. **home.dart:** Menyediakan kelas ProductListPage untuk menampilkan daftar produk dalam bentuk grid. Daftar objek Product didefinisikan dengan atribut seperti ID, nama, harga, URL gambar, dan deskripsi. Setiap produk diwakili oleh kelas ProductCard, yang menampilkan gambar, nama, harga, serta tombol "Shop Now" yang memungkinkan pengguna melihat detail lebih lanjut.
- d. **product.dart:** Mendefinisikan kelas Product, yang berperan sebagai model data produk dalam aplikasi. Kelas ini memiliki atribut ID, nama, harga, URL gambar, dan deskripsi. Terdapat dua metode untuk konversi antara objek Product dan format JSON: fromJson untuk mengubah data JSON menjadi objek Product, dan toJson untuk mengonversi objek Product kembali ke format JSON.