

**LAPORAN PRAKTIKUM  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL VII  
NAVIGASI DAN NOTIFIKASI**



**Disusun Oleh :**

**Dhiya Ulhaq Ramadhan 2211104053**

**Kelas :**

**SE-06-02**

**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**

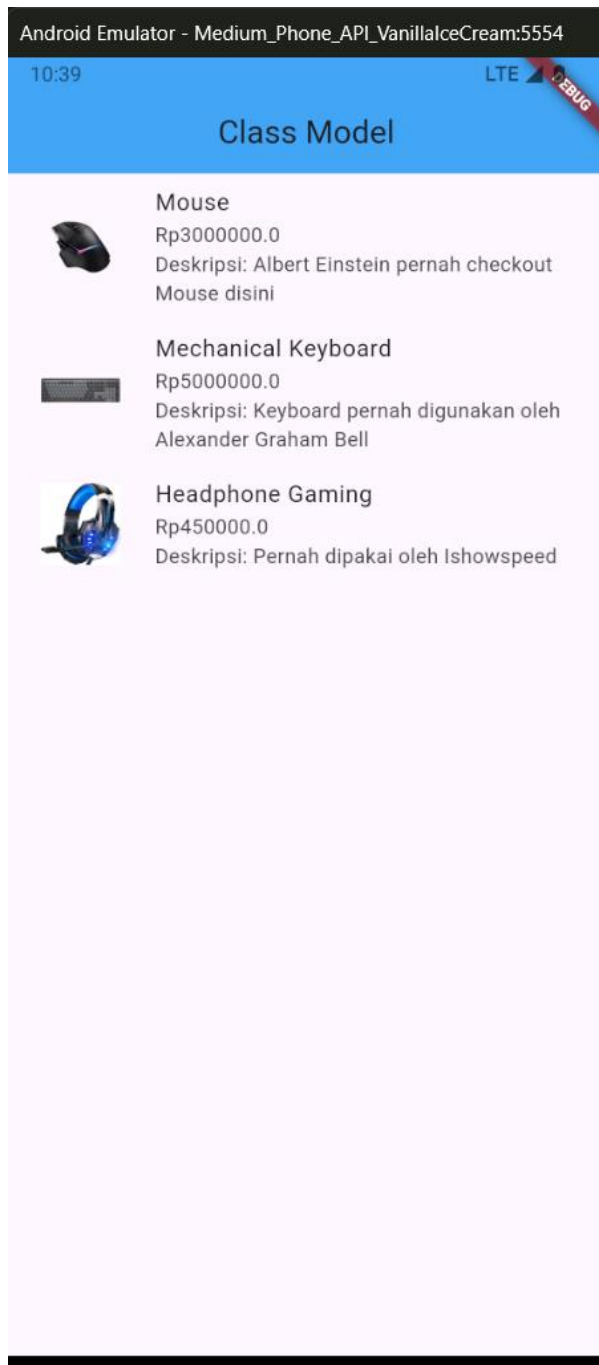
## A. GUIDED

### 1. Models

Source code :

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

Output :



#### Penjelasan Program :

membuat sebuah kelas Product dalam bahasa Dart atau Flutter. Kelas ini digunakan untuk mewakili informasi tentang sebuah produk, termasuk id, nama, harga, gambarUrl, dan deskripsi.

Kelas ini memiliki beberapa fungsionalitas yang dapat dijelaskan sebagai berikut:

1. Constructor: Kelas Product memiliki sebuah konstruktor yang menerima argumen-argumen yang dibutuhkan untuk membuat objek Product. Konstruktor ini akan menetapkan nilai-nilai tersebut ke properti-properti dari objek.
2. Konversi dari JSON: Kelas Product juga memiliki sebuah factory constructor yang menerima sebuah Map<String, dynamic> yang mewakili data JSON. Konstruktor ini akan mengambil nilai-nilai dari map tersebut dan membuat sebuah objek Product baru.
3. Konversi ke JSON: Kelas Product juga memiliki sebuah metode toJson() yang akan mengembalikan sebuah Map<String, dynamic> yang berisi nilai-nilai dari properti-properti objek Product. Hal ini memungkinkan objek Product untuk dikonversi ke format JSON.

## 2. Navigation

Source code :

```
1 import 'package:flutter/material.dart';
2 import 'package:interaksinavigasi/models/product.dart';
3 import 'package:interaksinavigasi/pages/detailpage.dart';
4
5 class MyPage extends StatelessWidget {
6   MyPage({super.key});
7
8   final List<Product> products = [
9     Product(
10      id: 1,
11      nama: 'Mouse',
12      harga: 3000000.00,
13      gambarUrl:
14        'https://resource.logitechg.com/w_386,ar_1.0,c_limit,f_auto,q_auto,dpr_2.0',
15      deskripsi: 'Albert Einstein pernah checkout Mouse disini'),
16     Product(
17      id: 2,
18      nama: 'Mechanical Keyboard',
19      harga: 5000000.00,
20      gambarUrl:
21        'https://resource.logitech.com/w_1600,c_limit,q_auto,f_auto,dpr_1.0/d_tran',
22      deskripsi: 'Keyboard pernah digunakan oleh Alexander Graham Bell'),
23     Product(
24      id: 3,
25      nama: 'Headphone Gaming',
26      harga: 450000.00,
27      gambarUrl:
28        'https://m.media-amazon.com/images/I/61CGHv6kmWL.AC_UF894,1000_QL80.jpg',
29      deskripsi: 'Pernah dipakai oleh Ishowspeed'),
30   ];
31
32   @override
33   Widget build(BuildContext context) {
34     return Scaffold(
35       appBar: AppBar(
36         title: const Text('Class Model'),
37         centerTitle: true,
38         backgroundColor: Colors.blue[400],
39       ), // AppBar
40       body: ListView.builder(
41         itemCount: products.length,
42         itemBuilder: (context, index) {
43           final product = products[index];
44           return ListTile(
45             leading: Image.network(
46               product.gambarUrl,
47               width: 70,
48               height: 70,
49             ), // Image.network
50             title: Text(product.nama),
51             subtitle: Column(
52               crossAxisAlignment: CrossAxisAlignment.start,
53               children: [
54                 Text('Rp${product.harga}'),
55                 Text('Deskripsi: ${product.deskripsi}'),
56               ],
57             ), // Column
58             onTap: () {
59               print('Tap Layar Berhasil');
60               Navigator.push(
61                 context,
62                 MaterialPageRoute(
63                   builder: (_) => DetailPage(
64                     data: Icon(Icons.notifications_outlined),
65                   ), // DetailPage
66                 ), // MaterialPageRoute
67               );
68             },
69           ); // ListTile
70         }); // ListView.builder // Scaffold
71     }
72   }
```

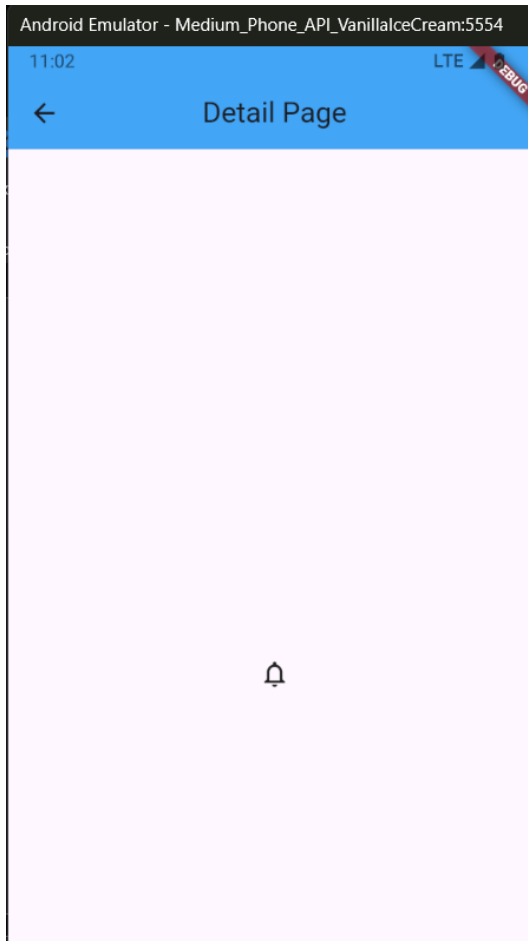
```

1 import 'package:flutter/material.dart';
2
3 class DetailPage extends StatelessWidget {
4   const DetailPage({super.key, required this.data});
5
6   final Widget data;
7
8   @override
9   Widget build(BuildContext context) {
10    return Scaffold(
11      appBar: AppBar(
12        title: Text('Detail Page'),
13        centerTitle: true,
14        backgroundColor: Colors.blue[400],
15      ), // AppBar
16      body: Center(
17        child: data,
18      ), // Center
19    ); // Scaffold
20  }
21 }

```

Output :





### Penjelasan Program

Aplikasi ini terdiri dari dua halaman utama: MyPage dan DetailPage.

Pada halaman MyPage, terdapat sebuah list produk yang disimpan dalam variabel `products`. Setiap produk direpresentasikan oleh objek `Product` yang memiliki informasi seperti `id`, `nama`, `harga`, `gambarUrl`, dan `deskripsi`. Halaman ini menggunakan `ListView.builder` untuk menampilkan daftar produk, di mana setiap produk ditampilkan dalam bentuk `ListTile` yang menampilkan gambar, nama, harga, dan deskripsi produk. Ketika pengguna mengetuk (tap) salah satu produk dalam daftar, aplikasi akan berpindah ke halaman `DetailPage`.

Halaman `DetailPage` adalah halaman yang sederhana, hanya menampilkan sebuah `Scaffold` dengan `AppBar` dan `Center` yang menampilkan sebuah `Widget` yang dikirimkan melalui parameter data.

Saat pengguna mengetuk salah satu produk, aplikasi akan mencetak pesan "Tap Layar Berhasil" ke konsol dan kemudian berpindah ke halaman `DetailPage`. Di halaman `DetailPage`, aplikasi akan menampilkan sebuah `Icon` dengan tipe `Icons.notifications_outlined` yang dikirimkan melalui parameter data.

### 3. Notification

Source code :

```

1 import 'package:flutter/material.dart';
2 import 'package:flutter_local_notifications/flutter_local_notifications.dart';
3 import 'package:interaksinavigasi/models/product.dart';
4 import 'package:interaksinavigasi/pages/detailpage.dart';
5
6 > class MyPage extends StatefulWidget { ...
10
11 class _MyPageState extends State<MyPage> {}
12   final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =
13     | FlutterLocalNotificationsPlugin();
14
15 > final List<Product> products = [ ...
38
39 void showNotification() async {
40   const AndroidNotificationDetails androidNotificationDetails =
41     | AndroidNotificationDetails('channel_id', 'Channel Name',
42     | | priority: Priority.high, importance: Importance.max);
43   const NotificationDetails notificationDetails =
44     | NotificationDetails(android: androidNotificationDetails);
45   await flutterLocalNotificationsPlugin.show(0, 'Flutter Notification',
46     | 'This is a local notification', notificationDetails,
47     | payload: 'Notification Payload');
48 }
49
50 @override
51 Widget build(BuildContext context) {
52   return Scaffold(
53     appBar: AppBar(
54       title: const Text('Class Model'),
55       centerTitle: true,
56       backgroundColor: Colors.blue[400],
57     ), // AppBar
58     body: ListView.builder(
59       itemCount: products.length,
60       itemBuilder: (context, index) {
61         final product = products[index];
62         return ListTile(
63           leading: Image.network(
64             product.gambarUrl,
65             width: 70,
66             height: 70,
67           ), // Image.network
68           title: Text(product.nama),
69           subtitle: Column(
70             crossAxisAlignment: CrossAxisAlignment.start,
71             children: [
72               Text('Rp${product.harga}'),
73               Text('Deskripsi: ${product.deskripsi}'),
74             ],
75           ), // Column
76           onTap: () {
77             print('Tap Layar Berhasil');
78             Navigator.push(
79               context,
80               MaterialPageRoute(
81                 builder: (_) => DetailPage(
82                   data: Icon(Icons.notifications_outlined),
83                 ), // DetailPage
84               ), // MaterialPageRoute
85             );
86           },
87         ); // ListTile
88       }, // ListView.builder
89       floatingActionButton: FloatingActionButton(
90         onPressed: showNotification,
91         child: const Icon(Icons.notifications),
92       ), // FloatingActionButton // Scaffold
93     ],
94   );

```

## B. UNGUIDED

Source code :

Homepage.dart

```
1 import 'package:flutter/material.dart';
2 import '../models/product.dart';
3 import 'detailpage.dart';
4 import 'cartpage.dart';
5
6 class HomePage extends StatefulWidget {
7   const HomePage({Key? key}) : super(key: key);
8
9   @override
10  State<HomePage> createState() => _HomePageState();
11 }
12
13 class _HomePageState extends State<HomePage> {
14   List<Product> products = [
15     Product(
16       id: 1,
17       name: "Pisang Goreng",
18       price: 10000,
19       description: "Pisang goreng crispy dengan topping keju dan coklat",
20       imageUrl:
21         "https://cdn1-production-images-kly.akamaized.net/vj5n-ZSFhuvchNVuKRCLHJeZpw8=/680x383/smart/
22     ),
23     Product(
24       id: 2,
25       name: "Keripik Pisang",
26       price: 15000,
27       description: "Keripik pisang renyah berbagai varian rasa",
28       imageUrl:
29         "https://amaan.co.id/wp-content/uploads/2023/10/pexels-saveurs-secretes-9191924-scaled.jpg",
30     ),
31     Product(
32       id: 3,
33       name: "Banana Cake",
34       price: 25000,
35       description: "Kue pisang lembut dengan cream cheese",
36       imageUrl:
37         "https://www.tamingtwins.com/wp-content/uploads/2024/03/banana_cake_final_1.jpg",
```



```

38 |     ),
39 | ];
40
41 | List<Product> cartItems = [];
42
43 | @override
44 | Widget build(BuildContext context) {
45 |     return Scaffold(
46 |         appBar: AppBar(
47 |             title: const Text("Banana Store"),
48 |             actions: [
49 |                 IconButton(
50 |                     icon: const Icon(Icons.shopping_cart),
51 |                     onPressed: () {
52 |                         Navigator.push(
53 |                             context,
54 |                             MaterialPageRoute(
55 |                                 builder: (context) => CartPage(cartItems: cartItems),
56 |                             ), // MaterialPageRoute
57 |                         );
58 |                     },
59 |                 ), // IconButton
60 |             ],
61 |         ), // AppBar
62 |         body: GridView.builder(
63 |             padding: const EdgeInsets.all(10),
64 |             gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
65 |                 crossAxisCount: 2,
66 |                 childAspectRatio: 0.75,
67 |                 crossAxisSpacing: 10,
68 |                 mainAxisSpacing: 10,
69 |             ), // SliverGridDelegateWithFixedCrossAxisCount
70 |             itemCount: products.length,
71 |             itemBuilder: (context, index) {
72 |                 return GestureDetector(
73 |                     onTap: () {

```

```

74     Navigator.push(
75       context,
76       MaterialPageRoute(
77         builder: (context) => DetailPage(
78           product: products[index],
79           onAddToCart: (product) {
80             setState(() {
81               int existingIndex = cartItems
82                 .indexWhere((item) => item.id == product.id);
83               if (existingIndex >= 0) {
84                 cartItems[existingIndex].quantity++;
85               } else {
86                 product.quantity = 1;
87                 cartItems.add(product);
88               }
89             });
90           },
91         ), // DetailPage
92       ), // MaterialPageRoute
93     );
94   },
95   child: Card(
96     elevation: 3,
97     child: Column(
98       crossAxisAlignment: CrossAxisAlignment.start,
99       children: [
100         Expanded(
101           child: Container(
102             width: double.infinity,
103             color: Colors.grey[200],
104             child: const Icon(Icons.image, size: 50),
105           ), // Container
106         ), // Expanded
107         Padding(
108           padding: const EdgeInsets.all(8.0),
109           child: Column(
110             crossAxisAlignment: CrossAxisAlignment.start,
111             children: [
112               Text(
113                 products[index].name,
114                 style: const TextStyle(
115                   fontSize: 16,
116                   fontWeight: FontWeight.bold,
117                 ), // TextStyle
118               ), // Text
119               Text(
120                 "Rp ${products[index].price}",
121                 style: TextStyle(
122                   color: Colors.green[700],
123                   fontSize: 14,
124                 ), // TextStyle
125               ), // Text
126             ],
127           ), // Column
128         ), // Padding
129       ],
130     ), // Column
131   ), // Card
132 ); // GestureDetector
133 ],
134 ), // GridView.builder
135 ); // Scaffold
136 }
137 }

```

## cartpage.dart

```
1 import 'package:flutter/material.dart';
2 import '../models/product.dart';
3
4 class CartPage extends StatefulWidget {
5   final List<Product> cartItems;
6
7   const CartPage({Key? key, required this.cartItems}) : super(key: key);
8
9   @override
10  State<CartPage> createState() => _CartPageState();
11 }
12
13 class _CartPageState extends State<CartPage> {
14   double getTotalPrice() {
15     return widget.cartItems.fold(
16       0,
17       (sum, item) => sum + (item.price * item.quantity),
18     );
19   }
20
21   Widget _buildProductImage(String? imageUrl) {
22     if (imageUrl != null && imageUrl.isNotEmpty) {
23       return Container(
24         width: 50,
25         height: 50,
26         decoration: BoxDecoration(
27           color: Colors.grey[200],
28           image: DecorationImage(
29             image: NetworkImage(imageUrl),
30             fit: BoxFit.cover,
31           ), // DecorationImage
32         ), // BoxDecoration
33       ); // Container
34     } else {
35       return Container(
36         width: 50,
37         height: 50,
38         color: Colors.grey[200],
39         child: const Icon(Icons.image, size: 25),
40       ); // Container
41     }
42   }
43
44   @override
45   Widget build(BuildContext context) {
46     return Scaffold(
47       appBar: AppBar(
48         title: const Text("Keranjang"),
49       ), // AppBar
50       body: widget.cartItems.isEmpty
51         ? const Center(
52           child: Text("Keranjang kosong"),
53         ) // Center
54         : Column(
55           children: [
56             Expanded(
57               child: ListView.builder(
58                 itemCount: widget.cartItems.length,
59                 itemBuilder: (context, index) {
60                   final item = widget.cartItems[index];
61                   return ListTile(
62                     leading: _buildProductImage(item.imageUrl),
63                     title: Text(item.name),
64                     subtitle: Text("Rp ${item.price}"),
65                     trailing: Row(
66                       mainAxisAlignment: MainAxisAlignment.min,
67                       children: [
68                         IconButton(
69                           icon: const Icon(Icons.remove),
70                           onPressed: () {
71                             setState(() {
72                               if (item.quantity > 1) {
73                                 item.quantity--;
74                               } else {
```

```

74         } else {
75             widget.cartItems.removeAt(index);
76         }
77     });
78 },
79 ), // IconButton
80 Text("${item.quantity}"),
81 IconButton(
82     icon: const Icon(Icons.add),
83     onPressed: () {
84         setState(() {
85             item.quantity++;
86         });
87     },
88 ), // IconButton
89 ],
90 ), // Row
91 ); // ListTile
92 },
93 ), // ListView.builder
94 ), // Expanded
95 Container(
96     padding: const EdgeInsets.all(16),
97     child: Column(
98         children: [
99             Row(
100                 mainAxisAlignment: MainAxisAlignment.spaceBetween,
101                 children: [
102                     const Text(
103                         "Total:",
104                         style: TextStyle(
105                             fontSize: 20,
106                             fontWeight: FontWeight.bold,
107                         ), // TextStyle
108                     ), // Text
109                     Text(
110                         "Rp. ${getTotalPrice()}.",
111                         "Rp. ${getTotalPrice()}",
112                         style: const TextStyle(
113                             fontSize: 20,
114                             fontWeight: FontWeight.bold,
115                         ), // TextStyle
116                     ), // Text
117                 ],
118             ), // Row
119             const SizedBox(height: 16),
120             SizedBox(
121                 width: double.infinity,
122                 child: ElevatedButton(
123                     onPressed: () {
124                         // Implementasi checkout
125                         ScaffoldMessenger.of(context).showSnackBar(
126                             const SnackBar(
127                                 content: Text("Fitur checkout belum tersedia"),
128                             ), // SnackBar
129                         );
130                     },
131                     child: const Padding(
132                         padding: EdgeInsets.all(12.0),
133                         child: Text("Checkout"),
134                     ), // Padding
135                 ), // ElevatedButton
136             ), // SizedBox
137         ],
138     ), // Column
139 ), // Container
140 ],
141 ), // Column
142 ); // Scaffold
143 }

```

detailpage.dart

```
1 import 'package:flutter/material.dart';
2 import '../models/product.dart';
3
4 class DetailPage extends StatelessWidget {
5   final Product product;
6   final Function(Product) onAddToCart;
7
8   const DetailPage({
9     Key? key,
10    required this.product,
11    required this.onAddToCart,
12  }) : super(key: key);
13
14  Widget _buildProductImage(String? imageUrl) {
15    if (imageUrl != null && imageUrl.isNotEmpty) {
16      return Container(
17        height: 300,
18        width: double.infinity,
19        decoration: BoxDecoration(
20          color: Colors.grey[200],
21          image: DecorationImage(
22            image: NetworkImage(imageUrl),
23            fit: BoxFit.cover,
24          ), // DecorationImage
25        ), // BoxDecoration
26      ); // Container
27    } else {
28      return Container(
29        height: 300,
30        width: double.infinity,
31        color: Colors.grey[200],
32        child: const Icon(Icons.image, size: 150),
33      ); // Container
34    }
35  }
```

```

37 @override
38 Widget build(BuildContext context) {
39   return Scaffold(
40     appBar: AppBar(
41       title: Text(product.name),
42     ), // AppBar
43     body: Column(
44       crossAxisAlignment: CrossAxisAlignment.start,
45       children: [
46         _buildProductImage(product.imageUrl),
47         Padding(
48           padding: const EdgeInsets.all(16.0),
49           child: Column(
50             crossAxisAlignment: CrossAxisAlignment.start,
51             children: [
52               Text(
53                 product.name,
54                 style: const TextStyle(
55                   fontSize: 24,
56                   fontWeight: FontWeight.bold,
57                 ), // TextStyle
58               ), // Text
59               const SizedBox(height: 8),
60               Text(
61                 "Rp ${product.price}",
62                 style: TextStyle(
63                   fontSize: 20,
64                   color: Colors.green[700],
65                   fontWeight: FontWeight.bold,
66                 ), // TextStyle
67               ), // Text
68               const SizedBox(height: 16),
69               Text(
70                 product.description,
71                 style: const TextStyle(fontSize: 16),
72               ), // Text
73               const SizedBox(height: 24),
74               SizedBox(
75                 width: double.infinity,
76                 child: ElevatedButton(
77                   onPressed: () {
78                     onAddToCart(product);
79                     ScaffoldMessenger.of(context).showSnackBar(
80                       const SnackBar(
81                         content: Text("Produk ditambahkan ke keranjang"),
82                         duration: Duration(seconds: 1),
83                       ), // SnackBar
84                     );
85                   },
86                   child: const Padding(
87                     padding: EdgeInsets.all(12.0),
88                     child: Text("Tambah ke Keranjang"),
89                   ), // Padding
90                 ), // ElevatedButton
91               ), // SizedBox
92             ],
93           ), // Column
94         ), // Padding
95       ],
96     ), // Column
97   ); // Scaffold
98 }
99 }

```

product.dart

```

1  class Product {
2      final int id;
3      final String name;
4      final double price;
5      final String description;
6      final String? imageUrl;
7      int quantity;
8
9      Product({
10         required this.id,
11         required this.name,
12         required this.price,
13         required this.description,
14         required this.imageUrl,
15         this.quantity = 0,
16     });
17 }

```

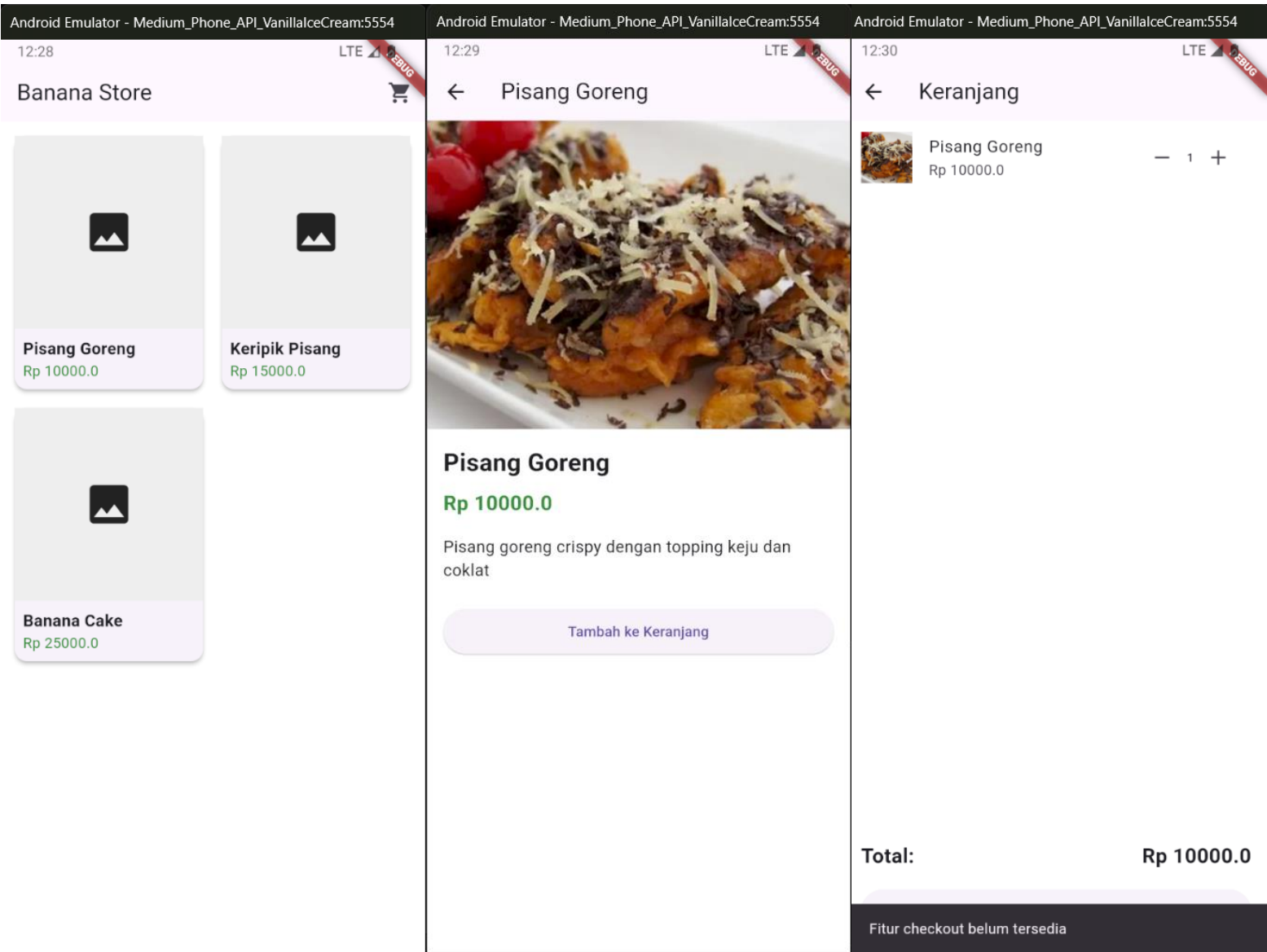
main.dart

```

1  import 'package:flutter/material.dart';
2  import 'pages/homepage.dart';
3
4  Run | Debug | Profile
5  void main() {
6      runApp(const MyApp());
7  }
8
9  class MyApp extends StatelessWidget {
10     const MyApp({Key? key}) : super(key: key);
11
12     @override
13     Widget build(BuildContext context) {
14         return MaterialApp(
15             title: 'Banana Store',
16             theme: ThemeData(
17                 primarySwatch: Colors.brown,
18                 scaffoldBackgroundColor: Colors.white,
19             ), // ThemeData
20             home: const HomePage(),
21         ); // MaterialApp
22     }
23 }

```

Screenshoot Output



### Penjelasan Program :

Program yang dibuat adalah sebuah aplikasi toko online untuk produk-produk berbahan dasar pisang, seperti pisang goreng, keripik pisang, dan banana cake. Aplikasi ini terdiri dari tiga halaman utama: Halaman Utama, Halaman Detail Produk, dan Halaman Keranjang Belanja.

Halaman Utama menampilkan daftar produk yang tersedia dalam bentuk grid layout. Setiap produk ditampilkan dengan thumbnail gambar, nama, dan harga. Pengguna dapat mengklik salah satu produk untuk masuk ke Halaman Detail Produk.

Halaman Detail Produk menampilkan informasi lengkap mengenai produk yang dipilih, termasuk gambar, nama, harga, dan deskripsi. Pada halaman ini, pengguna dapat menambahkan produk ke dalam keranjang belanja.

Ketika pengguna menambahkan produk ke dalam keranjang belanja, aplikasi akan menyimpan informasi produk tersebut, termasuk jumlah (quantity) yang dipilih. Jumlah produk di keranjang akan ditampilkan pada ikon keranjang di Halaman Utama. Pengguna dapat mengakses Halaman Keranjang Belanja dengan mengklik ikon keranjang di Halaman Utama.

Pada Halaman Keranjang Belanja, pengguna dapat melihat daftar produk yang telah ditambahkan, termasuk jumlah dan total harga. Pengguna juga dapat menambah atau mengurangi jumlah produk yang ada di dalam keranjang.



Secara algoritma, program ini menggunakan konsep Object-Oriented Programming (OOP) untuk mengelola data produk melalui class Product. Setiap produk memiliki atribut seperti id, name, price, description, imageUrl, dan quantity. Aplikasi menggunakan state management untuk mengelola perubahan data produk di dalam keranjang belanja.

Pada Halaman Utama, aplikasi menggunakan GridView untuk menampilkan daftar produk. Saat pengguna mengklik salah satu produk, aplikasi akan menavigasi ke Halaman Detail Produk dengan memanfaatkan Navigator.push(). Di Halaman Detail Produk, aplikasi menampilkan informasi produk dan menyediakan tombol untuk menambahkan produk ke dalam keranjang belanja.

Pada Halaman Keranjang Belanja, aplikasi menggunakan ListView untuk menampilkan daftar produk yang ada di dalam keranjang. Aplikasi juga menghitung total harga produk yang ada di dalam keranjang dan menampilkannya. Pengguna dapat menambah atau mengurangi jumlah produk di dalam keranjang, dan perubahan tersebut akan diperbarui secara langsung pada tampilan.

.

