

**PRAKTIKUM PEMROGRAMAN PERANGKAT BERGERAK**  
**MODUL VII**  
**NAVIGASI & NOTIFIKASI**



**Disusun Oleh:**

**Aorinka Anendya Chazanah / 2211104013**

**S1 SE-06-01**

**Asisten Praktikum:**

**MuhammadFazaZulian 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. Data dalam aplikasi memiliki berbagai bentuk, bergantung pada jenis aplikasinya. Akan lebih baik jika setiap data yang diterima atau dikirim mengikuti standar yang sama. Pemeliharaan proyek yang kompleks hampir mustahil dilakukan tanpa adanya model.

- **Membuat Model Class**

Untuk membuat model, mulailah dengan membuat direktori baru di dalam folder lib pada proyek Flutter. Selanjutnya, buat file kelas Dart dengan nama file sesuai dengan data yang akan dijadikan model. Misalnya, saat membuat model untuk \*user\* dengan respons JSON berikut:

```
Product{
  id: 1,
  nama: "Mouse",
  harga: 65000,
  imageUrl:
    "https://resource.logitechg.com/w_386,ar_1.0,c_limit,f_auto,q_auto,dpr_2.0/d_transparent.gif/content/dam/gaming/en/pro-
deskripsi:
  "Mouse ini menawarkan kenyamanan dan kebebasan bergerak tanpa kabel",
  data:
    "Mouse ini menawarkan kenyamanan dan kebebasan bergerak tanpa kabel",
}
```

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   final String data;
9
10  // Konstruktor
11  Product({
12    required this.id,
13    required this.nama,
14    required this.harga,
15    required this.imageUrl,
16    required this.deskripsi,
17    required this.data,
18  });
19
20  // Method untuk konversi JSON ke objek Product
21  factory Product.fromJson(Map<String, dynamic> json) {
22    return Product(
23      id: json['id'],
24      nama: json['nama'],
25      harga: json['harga'].toDouble(),
26      imageUrl: json['imageUrl'],
27      deskripsi: json['deskripsi'],
28      data: json['data'],
29    );
30  }
```

## 2. Navigation

Dalam Flutter, navigasi mengacu pada metode untuk berpindah antar halaman (atau tampilan) dalam aplikasi. Sistem navigasi di Flutter menggunakan konsep route dan navigator. Setiap halaman atau layar di Flutter dikenal sebagai route sementara Navigator adalah widget yang mengelola tumpukan dari route 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       title: 'Flutter Demo',
63       theme: ThemeData(
64         colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
65         useMaterial3: true,
66       ),
67       home: MyPage(),
68     );
69   }
70 }
71
```

**b) mypage.dart**

**c) product.dart**

```

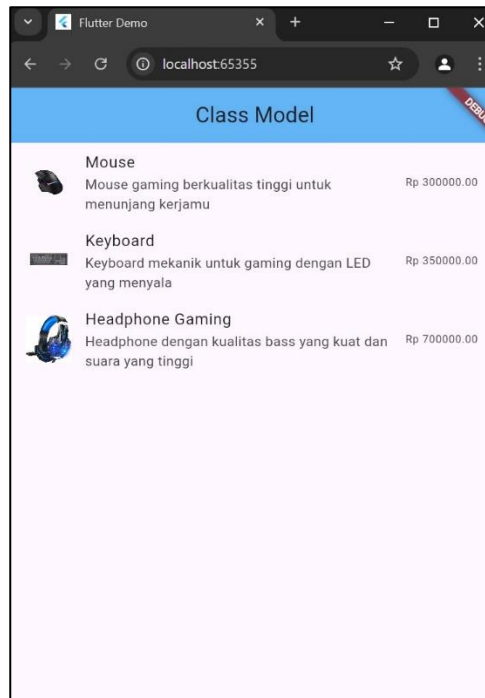
1 class Product {
2     final int id;
3     final String nama;
4     final double harga;
5     final String imageUrl;
6     final String deskripsi;
7
8     // Constructor
9     Product({
10         required this.id,
11         required this.nama,
12         required this.harga,
13         required this.imageUrl,
14         required this.deskripsi,
15     });
16
17     // Method untuk 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

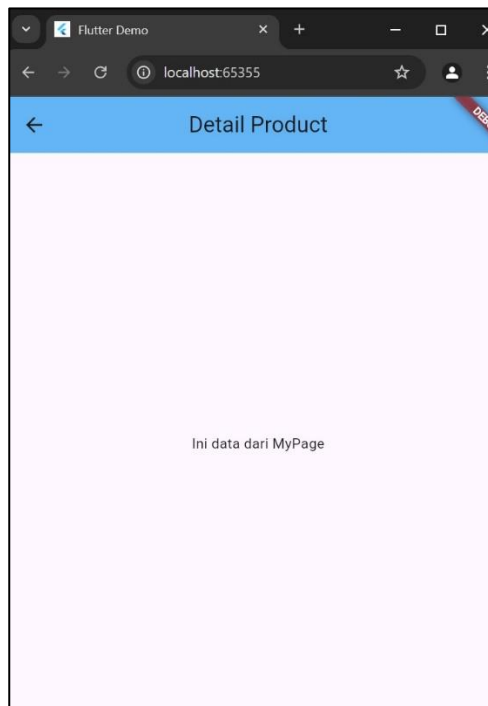
```
1 class Product {
2   final int id;
3   final String nama;
4   final double harga;
5   final String imageUrl;
6   final String deskripsi;
7
8   // Constructor
9   Product({
10     required this.id,
11     required this.nama,
12     required this.harga,
13     required this.imageUrl,
14     required 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 }
```

**e) Output halaman depan**



**f) Output detail produk**

Jika salah satu produk di klik maka akan muncul tampilan seperti berikut



### 3. Notification

- a) Untuk mengirim notifikasi di aplikasi Flutter, Anda dapat menggunakan package yang bernama `flutter\_local\_notifications` yang tersedia di situs pub.dev. Setelah itu, tambahkan package ini ke dalam file `pubspec.yaml`.

```
cupertino_icons: ^1.0.8
flutter_local_notifications: ^17.2.4
permission_handler: ^11.3.1

dev_dependencies:
  flutter_test:
    sdk: flutter
```

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

```
minSdkVersion 21
}
```

- c) Tambahkan izin pada file *AndroidManifest.xml*

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

#### d) File notification.dart

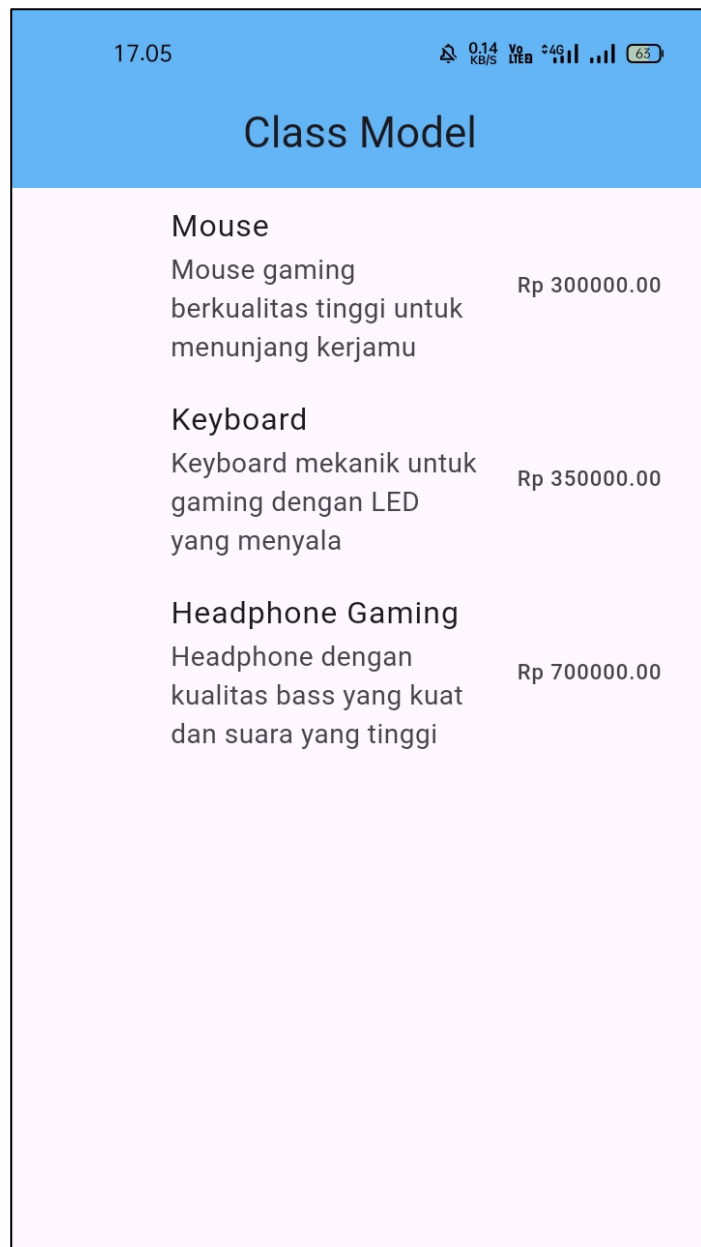
```
1  import 'dart:async';
2  import 'package:flutter/material.dart';
3  import 'package:flutter_local_notifications/flutter_local_notifications.dart';
4
5  void main() => runApp(MaterialApp(
6    theme: ThemeData(
7      appBarTheme: const AppBarTheme(
8        color: Colors.amber,
9      ),
10   ),
11   home: const MyApp(),
12   debugShowCheckedModeBanner: false,
13 ));
14
15 class MyApp extends StatefulWidget {
16   const MyApp({Key? key}) : super(key: key);
17
18   @override
19   _MyAppState createState() => _MyAppState();
20 }
21
22 class _MyAppState extends State<MyApp> {
23   final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =
24     FlutterLocalNotificationsPlugin();
25
26   @override
27   void initState() {
28     super.initState();
29     _initializeNotifications();
30   }
31
32   Future<void> _initializeNotifications() async {
33     const AndroidInitializationSettings initializationSettingsAndroid =
34       AndroidInitializationSettings('@mipmap/ic_launcher');
35
36     const DarwinInitializationSettings initializationSettingsIOS =
37       DarwinInitializationSettings();
38
39     const InitializationSettings initSettings = InitializationSettings(
40       android: initializationSettingsAndroid,
41       iOS: initializationSettingsIOS,
42     );
43
44     await flutterLocalNotificationsPlugin.initialize(
45       initSettings,
46       onDidReceiveNotificationResponse: (NotificationResponse response) async {
47         await _onSelectNotification(response.payload);
48       },
49     );
50   }
51
52   Future<void> _onSelectNotification(String? payload) async {
53     if (payload != null) {
54       Navigator.of(context).push(MaterialPageRoute(
55         builder: (_) => NewScreen(payload: payload),
56       ));
57     }
58   }
59
60   @override
61   Widget build(BuildContext context) {
62     return Scaffold(
63       appBar: AppBar(
64         backgroundColor: Colors.amber,
65         title: const Text('Flutter Notification Demo'),
66       ),
```





```
1  body: Center(  
2    child: Column(  
3      mainAxisAlignment: MainAxisAlignment.center,  
4      children: <Widget>[  
5        SizedBox(  
6          width: 250.0,  
7          child: ElevatedButton(  
8            style: ElevatedButton.styleFrom(  
9              backgroundColor: Colors.blueAccent,  
10             ),  
11            onPressed: _showNotification,  
12            child: const Text('Show Notification'),  
13          ),  
14        ],  
15      ),  
16    ),  
17  );  
18  );  
19  }  
20  
21  Future<void> _showNotification() async {  
22    const AndroidNotificationDetails androidPlatformChannelSpecifics =  
23      AndroidNotificationDetails(  
24        'id',  
25        'channel',  
26        importance: Importance.max,  
27        priority: Priority.high,  
28      );  
29  
30    const DarwinNotificationDetails iOSPlatformChannelSpecifics =  
31      DarwinNotificationDetails();  
32  
33    const NotificationDetails platformChannelSpecifics = NotificationDetails(  
34      android: androidPlatformChannelSpecifics,  
35      iOS: iOSPlatformChannelSpecifics,  
36    );  
37  
38    await flutterLocalNotificationsPlugin.show(  
39      0,  
40      'Flutter Devs',  
41      'Flutter Local Notification Demo',  
42      platformChannelSpecifics,  
43      payload: 'Welcome to the Local Notification demo',  
44    );  
45  }  
46  }  
47  
48  class NewScreen extends StatelessWidget {  
49    final String payload;  
50  
51    const NewScreen({  
52      Key? key,  
53      required this.payload,  
54    }) : super(key: key);  
55  
56    @override  
57    Widget build(BuildContext context) {  
58      return Scaffold(  
59        appBar: AppBar(  
60          title: Text(payload),  
61        ),  
62        body: Center(  
63          child: Text('Notification Payload: $payload'),  
64        ),  
65      );  
66    }  
67  }
```

e) Output contoh notification



### Produk Dipilih

Anda memilih produk Mouse

## Mouse

Rp 300000.00

Mouse gaming berkualitas tinggi untuk menunjang kerjamu


## 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, screenshot output, dan deskripsi program. Kreatifitas menjadi nilai tambah*

### Source code:

a) main.dart



```
1  import 'package:flutter/material.dart';
2  import './pages/homepage.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       title: 'Aplikasi E-Commerce',
13       theme: ThemeData(primarySwatch: Colors.blue),
14       home: ProductListPage(),
15     );
16   }
17 }
18
```

## b) homepage.dart

```
1 import 'package:flutter/material.dart';
2 import '../models/product.dart';
3 import 'details_page.dart';
4
5 class ProductListPage extends StatelessWidget {
6   final List<Product> products = [
7     Product(
8       id: 1,
9       name: 'Tiffany Wire Bangle',
10      price: 1100000.00,
11      imageUrl:
12        'https://media.tiffany.com/is/image/Tiffany/EcomBrowse/tiffany-knotwire-bangle-09526820_18894400_ED_M.jpg?defaultImage=hoImageAvailableInternal&fwt=webp',
13      description:
14        'Tiffany Knot is an expression of love's unwavering bonds. Inspired by an archival bow crafted in 1889, Knot is a symbol of life's most enduring ties and meaningful connections',
15    ),
16    Product(
17      id: 2,
18      name: 'Tiffany T Narrow Diamond',
19      price: 1500000.00,
20      imageUrl:
21        'https://media.tiffany.com/is/image/Tiffany/EcomBrowse/tiffany-tt1-narrow-diamond-linget-bangle-68335697_1816636_ED.jpg?defaultImage=hoImageAvailableInternal&fwt=webp',
22      description:
23        'Tiffany T is an expression of love's endless potential. Inspired by an archival bracelet from 1875, T is an homage to the House's iconic motif and the spirit of New York',
24    ),
25    Product(
26      id: 3,
27      name: 'Tiffany Coquette Gold',
28      price: 2400000.00,
29      imageUrl:
30        'https://media.tiffany.com/is/image/Tiffany/EcomBrowse/tiffany-knotwire-bangle-78453363_1876316_ED.jpg?defaultImage=hoImageAvailableInternal&fwt=webp',
31      description:
32        'Tiffany Coquette Gold is an expression of love's unwavering bonds. Inspired by an archival bow crafted in 1889, Knot is a symbol of life's most enduring ties and meaningful connections',
33    ),
34    Product(
35      id: 4,
36      name: 'Tiffany T Wire Bracelet',
37      price: 100000.00,
38      imageUrl:
39        'https://media.tiffany.com/is/image/Tiffany/EcomBrowse/tiffany-twire-bracelet-35263385_P97785_ED_M.jpg?defaultImage=hoImageAvailableInternal&fwt=webp',
40      description:
41        'Tiffany T is an expression of love's endless potential. Inspired by an archival bracelet from 1875, T is an homage to the House's iconic motif and the spirit of New York',
42    ),
43  ];
44
45   @override
46   Widget build(BuildContext context) {
47     return Scaffold(
48       appBar: AppBar(title: Text('Tiffany & Co Product')),
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   }
60 }
```

```
1 class ProductCard extends StatelessWidget {
2   final Product product;
3
4   ProductCard(this.product);
5
6   @override
7   Widget build(BuildContext context) {
8     return Card(
9       color: Colors.grey.shade200,
10      shape: RoundedRectangleBorder(
11        borderRadius: BorderRadius.circular(8),
12      ),
13      child: Column(
14        crossAxisAlignment: CrossAxisAlignment.start,
15        children: [
16          ClipRRect(
17            borderRadius: BorderRadius.vertical(top: Radius.circular(8)),
18            child: Image.network(
19              product.imageUrl,
20              height: 200,
21              width: double.infinity,
22              fit: BoxFit.cover,
23            ),
24          ),
25          Padding(
26            padding: const EdgeInsets.all(6.0),
27            child: Column(
28              crossAxisAlignment: CrossAxisAlignment.start,
29              children: [
```

```
1   Text(
2       product.nama,
3       style: TextStyle(
4         fontWeight: FontWeight.bold,
5         fontSize: 14,
6       ),
7     ),
8   Text(
9     'Rp ${product.harga.toStringAsFixed(2)}',
10    style: TextStyle(
11      color: Colors.black,
12      fontSize: 12,
13    ),
14  ),
15  SizedBox(height: 10), // Memberikan jarak antara teks dan tombol
16  Row(
17    children: [
18      Expanded(
19        child: ElevatedButton(
20          onPressed: () {
21            Navigator.of(context).push(
22              MaterialPageRoute(
23                builder: (context) => ProductDetailPage(product),
24              ),
25            );
26          },
27          child: Text('Shop Now'),
28          style: ElevatedButton.styleFrom(
29            shape: RoundedRectangleBorder(
30              borderRadius: BorderRadius.circular(
31                5), // Mengatur border radius
32            ),
33          ),
34        ),
35      ],
36    ),
37  ],
38 ),
39 ),
40 ),
41 ],
42 ),
43 );
44 }
45 }
```

c) detailpage.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 }
42
```

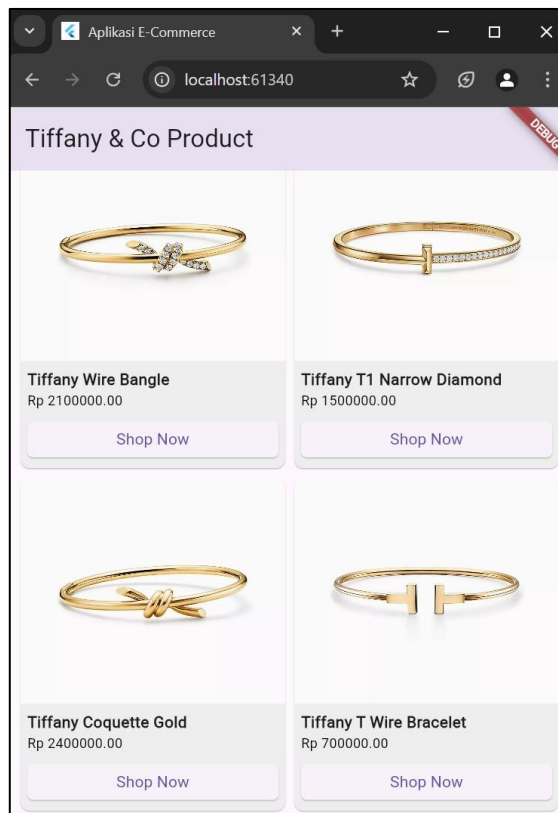
d) product.dart

```
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    // 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  }
39
```

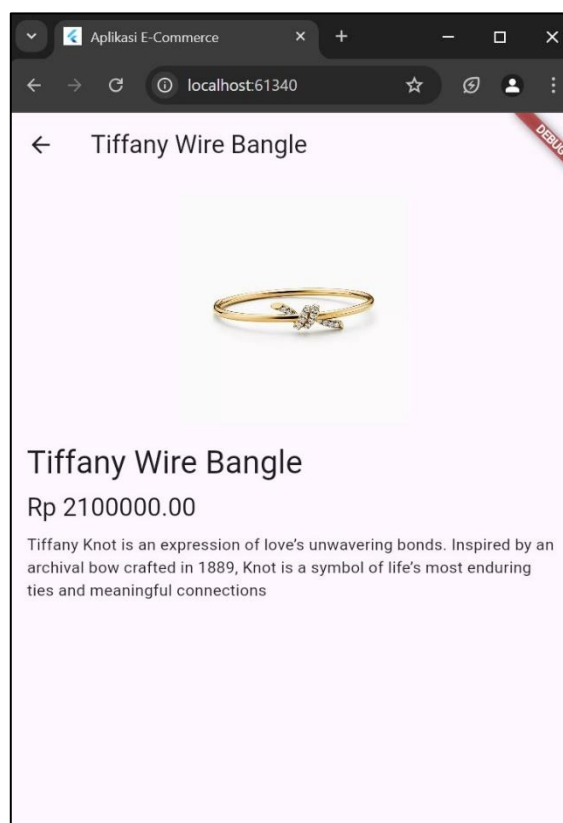


## Output

Halaman depan produk



Jika salah satu produk di klik maka muncul tampilan seperti ini



## Deskripsi Program

Program ini adalah sebuah aplikasi e-commerce pengguna dapat melihat daftar produk dalam format grid, lengkap dengan gambar, nama, harga, dan deskripsi singkat. Setiap produk dapat diklik untuk membuka halaman detail yang menampilkan informasi lebih lanjut, seperti gambar produk yang lebih besar, deskripsi lengkap, dan harga.

- **main.dart:** fungsi main dijalankan untuk meluncurkan aplikasi Flutter. Di dalam kelas MyApp, aplikasi ini mengonfigurasi judul dan tema dengan skema warna biru. Halaman awal aplikasi diatur untuk menampilkan ProductListPage, yang merupakan halaman utama di mana daftar produk ditampilkan.
- **detailpage.dart:** mendefinisikan kelas ProductDetailPage, yang bertanggung jawab untuk menampilkan detail produk ketika pengguna mengklik salah satu produk dari daftar. Di dalam halaman ini, terdapat AppBar yang menampilkan nama produk, dan konten utama berupa gambar produk, nama, harga, dan deskripsi.
- **homepage.dart:** berisi kelas ProductListPage, yang menampilkan daftar produk dalam bentuk grid. Di dalamnya, terdapat daftar objek Product yang didefinisikan dengan atribut seperti id, nama, harga, URL gambar, dan deskripsi. Setiap produk ditampilkan menggunakan kelas ProductCard, yang membungkus setiap produk dalam sebuah kartu dengan gambar, nama, harga, dan tombol "Shop Now" yang memungkinkan pengguna untuk melihat detail lebih lanjut.
- **product.dart:** mendefinisikan class Product, yang merupakan model data untuk produk dalam aplikasi. Kelas ini memiliki atribut seperti id, nama, harga, URL gambar, dan deskripsi. Selain itu, terdapat dua metode yang memungkinkan konversi antara objek Product dan format JSON, yaitu fromJson untuk mengonversi data JSON menjadi objek Product, dan toJson untuk mengonversi objek Product kembali ke format JSON.