

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

**MODUL VII  
NAVIGASI DAN NOTIFIKASI**



**Disusun Oleh :**

**Rizky Hanifa Afania / 2211104017**

**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**

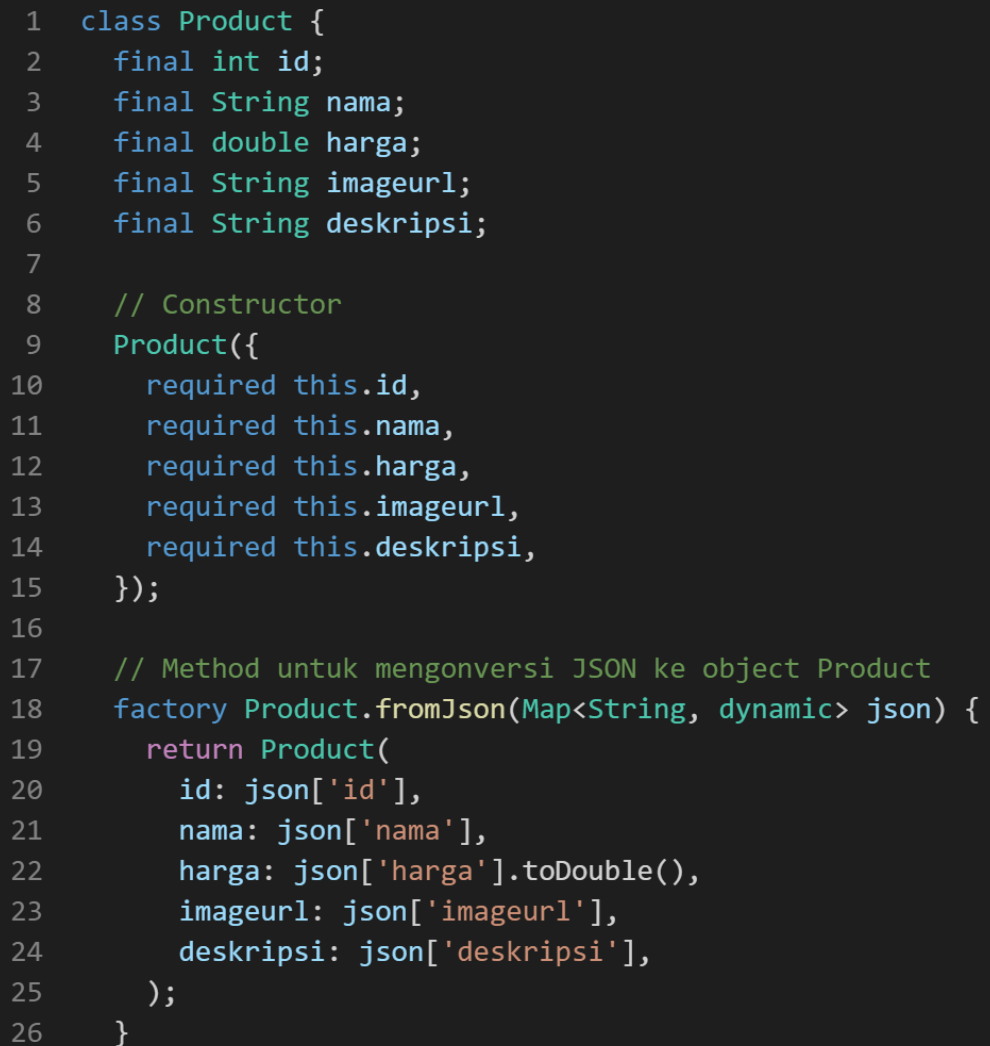
## GUIDED

### 1. Model

Secara umum, hampir semua aplikasi yang dikembangkan akan memanfaatkan data. Data dalam sebuah aplikasi bisa berbentuk berbagai macam, tergantung jenis aplikasinya. Akan lebih baik jika setiap data yang diterima atau dikirimkan mengikuti standar yang sama. Tanpa model, pemeliharaan proyek yang kompleks hampir tidak mungkin dilakukan.

#### a. Membuat Model Class

Di dalam file `product.dart` berisi Class `Product` sebagai wadah 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 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  }
```

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

```
Product(  
  id: 1,  
  nama: 'Mouse',  
  harga: 300000.00,  
  imageUrl:   
    'https://resource.logitechg.com/w_386,ar_1.0,c_limit,f_auto,q_auto,dpr_2.0/d_386x386px/logitech-gaming-mouse',  
  deskripsi: 'Mouse gaming berkualitas tinggi',  
),  
Product(  
  id: 2,  
  nama: 'Keyboard',  
  harga: 350000.00,  
  imageUrl:   
    'https://resource.logitechg.com/w_1600,c_limit,q_auto,f_auto,dpr_1.0/d_transp_1600x1600px/logitech-keyboard',  
  deskripsi: 'Keyboard mekanik untuk gaming',  
),  
Product(  
  id: 3,  
  nama: 'Headphone Gaming',  
  harga: 700000.00,  
  imageUrl:   
    'https://m.media-amazon.com/images/I/61CGHv6kmWL.AC_UF894,1000_QL80.jpg',  
  deskripsi: 'Headphone dengan kualitas suara tinggi',  
),
```

## 2. Navigation

Dalam Flutter, navigasi adalah proses untuk berpindah antar halaman (atau tampilan) dalam aplikasi. Sistem navigasi di Flutter didasarkan pada konsep route dan navigator. Setiap halaman atau tampilan disebut sebagai route, sedangkan Navigator adalah widget yang mengatur stack dari route tersebut.

### a. main.dart

```
1 import 'package:flutter/material.dart';  
2 import 'package:navigasi_notifikasi/mypage.dart';  
3  
4 void main() {  
5   runApp(const 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: 'Flutter Demo',  
15       theme: ThemeData(  
16         colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),  
17         useMaterial3: true,  
18       ),  
19       home: MyPage());  
20   }  
21 }
```

## b. mypagae.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:navigasi_notifikasi/models/product.dart';
3 import 'package:navigasi_notifikasi/pages/detail.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: 300000.00,
13      imageUrl:
14        'https://resource.logitechg.com/w_386,ar_1.0,c_limit,f_auto,q_auto,dpr_2.
15      deskripsi: 'Mouse gaming berkualitas tinggi',
16    ),
17    Product(
18      id: 2,
19      nama: 'Keyboard',
20      harga: 350000.00,
21      imageUrl:
22        'https://resource.logitech.com/w_1600,c_limit,q_auto,f_auto,dpr_1.0/d_tra
23      deskripsi: 'Keyboard mekanik untuk gaming',
24    ),
25    Product(
26      id: 3,
27      nama: 'Headphone Gaming',
28      harga: 700000.00,
29      imageUrl:
30        'https://m.media-amazon.com/images/I/61CGHv6kmWL.AC_UF894,1000_QL80.jpg',
31      deskripsi: 'Headphone dengan kualitas suara tinggi',
32    ),
33  ];
34
35  @override
36  Widget build(BuildContext context) {
37    return Scaffold(
38      appBar: AppBar(
39        title: const Text('Class Model'),
40        centerTitle: true,
41        backgroundColor: Colors.blue[300],
42      ),
43      body: ListView.builder(
44        itemCount: products.length,
45        itemBuilder: (context, index) {
46          final product = products[index];
47          return ListTile(
48            leading: Image.network(product.imageUrl, width: 50, height: 50),
49            title: Text(product.nama),
50            subtitle: Text(product.deskripsi),
51            trailing: Text('Rp ${product.harga.toStringAsFixed(2)}'),
52            //
53            onTap: () {
54              Navigator.push(
55                context, MaterialPageRoute(builder: (_) => DetailProduct());
56            },
57          );
58        },
59      ),
60    );
61  }
62 }
63
```

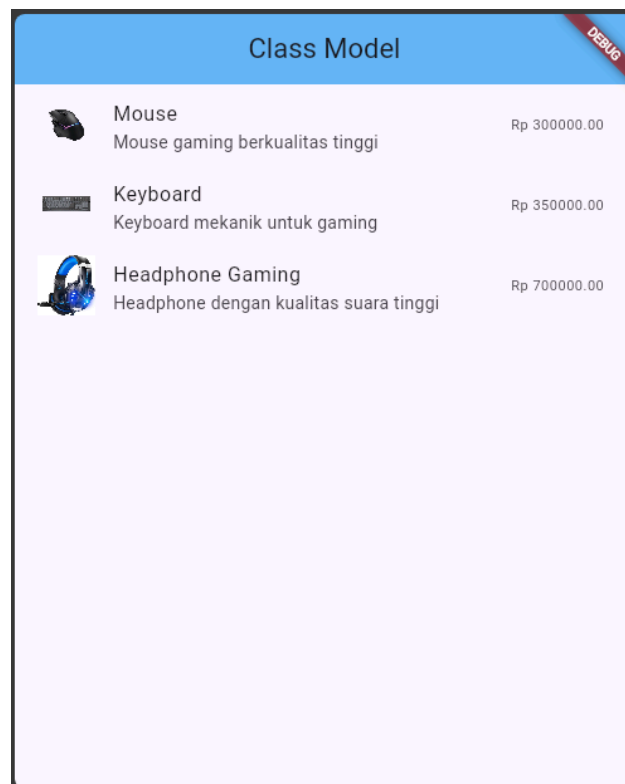
c. product.dart

```
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 }
39
```

#### d. detail.dart

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

#### e. Output Halaman Utama



Ketika di klik salah satu produk, maka akan berpindah ke halaman detail produk

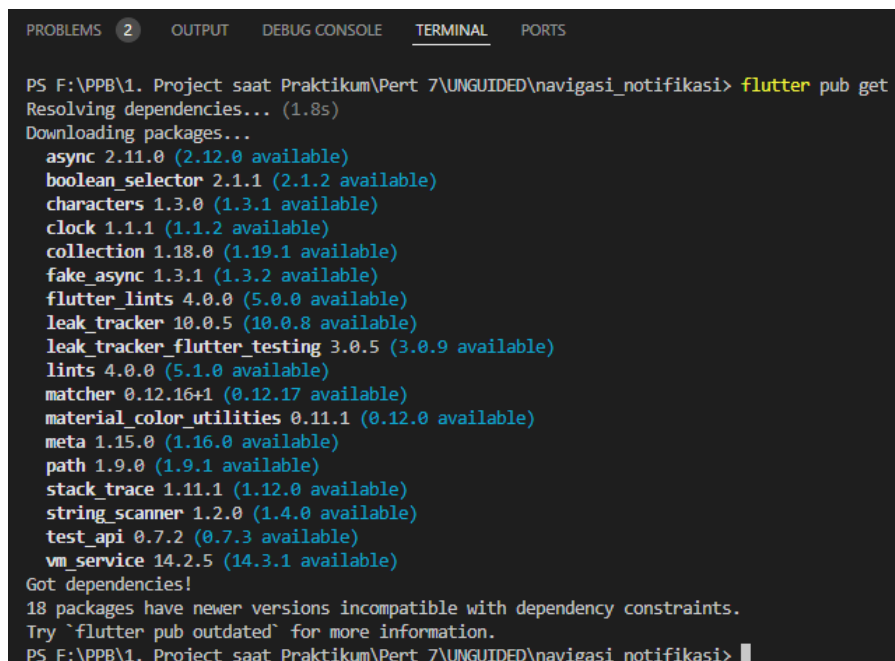


### 3. Notification

- a. Untuk mengirim notifikasi dalam aplikasi Flutter, bisa menggunakan package bernama **flutter\_local\_notifications** di website **pub.dev**. Kemudian menambahkan package ini ke dalam file pubspec.yaml.

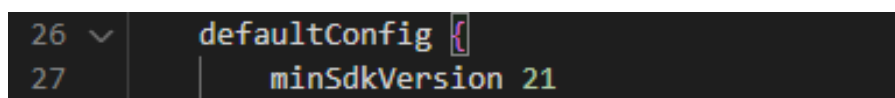
```
! pubspec.yaml X
! pubspec.yaml
28 # the latest version available on pub.dev. To see w
29 # versions available, run `flutter pub outdated`.
30 dependencies:
31   flutter:
32     sdk: flutter
33   cupertino_icons: ^1.0.8
34   flutter_local_notifications: ^17.2.4
```

- b. Lalu unduh dan install package tersebut di terminal



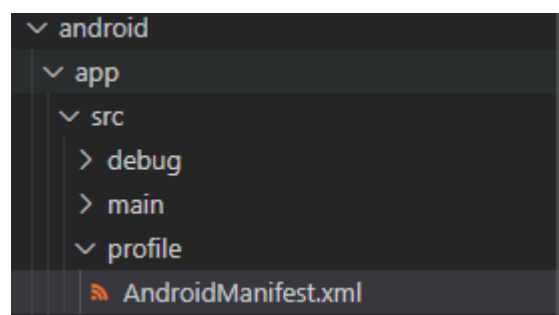
```
PS F:\PPB\1. Project saat Praktikum\Pert 7\UNGUIDED\navigasi_notifikasi> flutter pub get
Resolving dependencies... (1.8s)
Downloading packages...
  async 2.11.0 (2.12.0 available)
  boolean_selector 2.1.1 (2.1.2 available)
  characters 1.3.0 (1.3.1 available)
  clock 1.1.1 (1.1.2 available)
  collection 1.18.0 (1.19.1 available)
  fake_async 1.3.1 (1.3.2 available)
  flutter_lints 4.0.0 (5.0.0 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)
  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)
  vm_service 14.2.5 (14.3.1 available)
Got dependencies!
18 packages have newer versions incompatible with dependency constraints.
Try 'flutter pub outdated' for more information.
PS F:\PPB\1. Project saat Praktikum\Pert 7\UNGUIDED\navigasi_notifikasi>
```

- c. Kemudian Konfigurasi Android di dalam direktori **android/app/build.gradle** untuk mengatur versi minSdkVersion ke 21 atau yang lebih tinggi



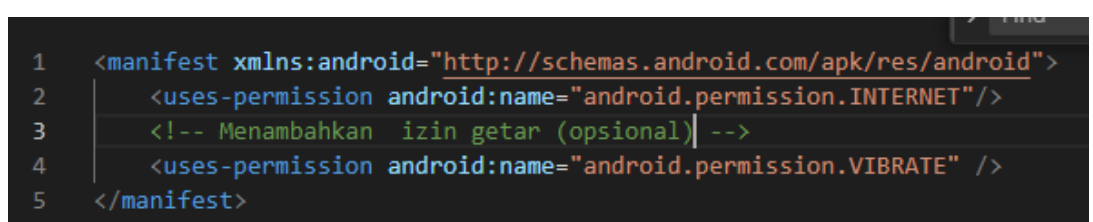
```
26  defaultConfig {
27      minSdkVersion 21
```

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



Lalu tambahkan code:

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



```
1  <manifest xmlns:android="http://schemas.android.com/apk/res/android">
2      <uses-permission android:name="android.permission.INTERNET"/>
3      <!-- Menambahkan izin getar (opsional) -->
4      <uses-permission android:name="android.permission.VIBRATE" />
5  </manifest>
```



Jika Anda ingin notifikasi tetap aktif setelah perangkat di-reboot (misalnya, untuk notifikasi yang terjadwal), tambahkan izin berikut:

**<uses-permission**

**android:name="android.permission.RECEIVE\_BOOT\_COMPLETED"/>**

```
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android">
2   <uses-permission android:name="android.permission.INTERNET"/>
3   <!-- Menambahkan izin getar (opsional) -->
4   <uses-permission android:name="android.permission.VIBRATE" />
5   <!-- Agar notifikasi tetap aktif setelah perangkat di-reboot -->
6   <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
7 </manifest>
```

- e. Inisialisasi flutter\_local\_notifications dengan menambahkan kode 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 = FlutterLocalNotificationsPlugin();
6
7 void main() async {
8   WidgetsFlutterBinding.ensureInitialized();
9
10  // Pengaturan inisialisasi untuk Android
11  const AndroidInitializationSettings initializationSettingsAndroid =
12    AndroidInitializationSettings('@mipmap/ic_launcher');
13
14  // Pengaturan inisialisasi untuk iOS
15  const DarwinInitializationSettings initializationSettingsIOS = DarwinInitializationSettings();
16
17  // Menggabungkan semua pengaturan platform
18  const InitializationSettings initializationSettings = InitializationSettings(
19    android: initializationSettingsAndroid,
20    iOS: initializationSettingsIOS,
21  );
22
23  // Inisialisasi plugin dengan pengaturan
24  await flutterLocalNotificationsPlugin.initialize(
25    initializationSettings,
26  );
27
28  runApp(MyApp());
29 }
30
31 class MyApp extends StatelessWidget {
32   @override
33   Widget build(BuildContext context) {
34     return MaterialApp(
35       title: 'Flutter Local Notifications',
36       theme: ThemeData(primarySwatch: Colors.blue),
37       home: NotificationScreen(),
38     );
39   }
40 }
41
```

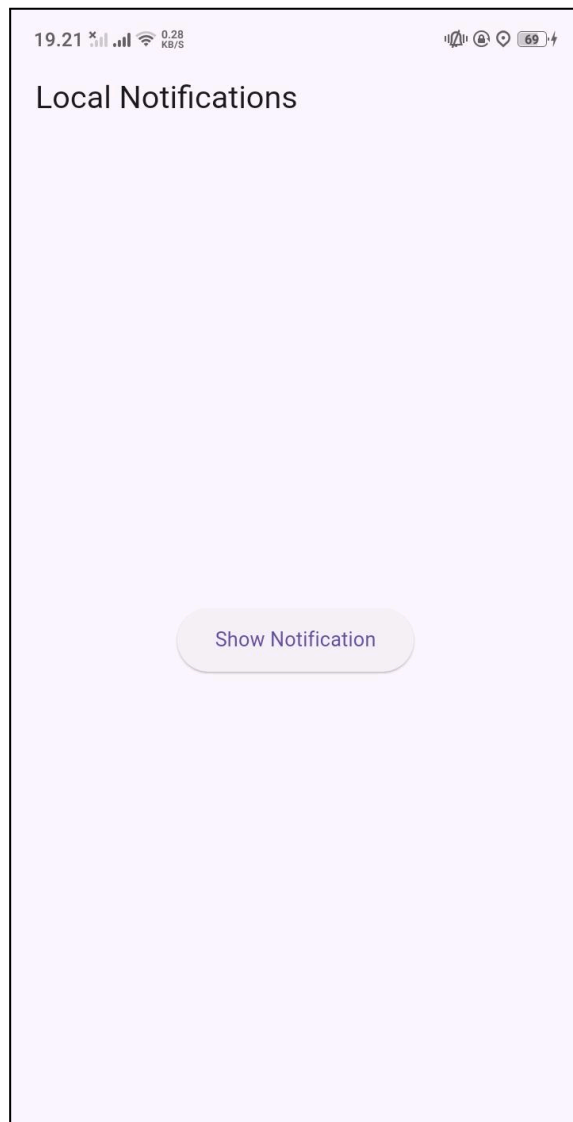
- f. Kemudian di file yang sama yaitu main.dart buat fungsi untuk menampilkan notifikasi dengan menambahkan kode berikut:

```
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 }
```

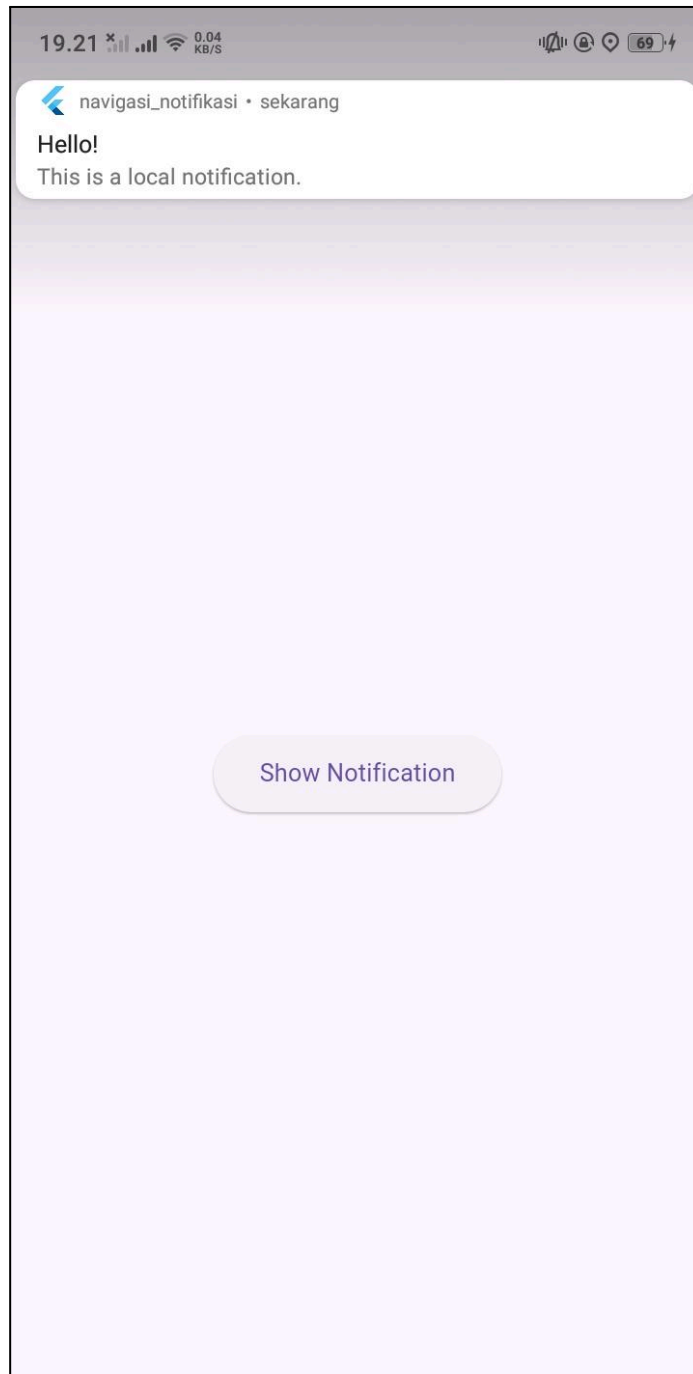
- g. Lalu buat UI dengan tombol untuk menampilkan notifikasi dengan kode berikut:

```
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 }
```

h. Output tampilan awal



- i. Output: setelah di klik button “**Show Notification**”, maka akan muncul notifikasi di atas layar.



## 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 'screens/home_screen.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  }
22
```

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:flutter/material.dart';
import '../models/product.dart';
import 'product_detail_screen.dart';

class HomeScreen extends StatelessWidget {
  final List<Product> products = [
    Product(
      id: '1',
      title: 'Suncscreen',
      description:
        'Berbeda dengan sunscreen physical, sunscreen ini biasanya  
punya kandungan zinc oxide. Bahan tersebut akan memantulkan sinar UV  
untuk menjaga kesehatan kulit. Jadi kalau kamu lebih suka menggunakan  
chemical sunscreen, kamu bisa pilih sunscreen Skintific ini.',
      price: 120000,
      imageUrl:
        'https://diary.marshabeauty.com/media/article_images/5291/Skintific_sun  
screen_physical_atau_chemical_1.jpg',
    ),
  ],
}
```

```
),
Product(
  id: '2',
  title: 'Moisturizer',
  description:
    'Vitamin B5 atau panthenol punya peran penting untuk
menyehatkan skin barrier sekaligus melembapkannya. Kandungan panthenol
yang cukup di dalam produk skincare akan mengurangi resiko skin barrier
rusak karena paparan zat kimia hingga polusi.',
  price: 110000,
  imageUrl:
'https://diary.marshabeauty.com/media/article_images/None/Skintific_5X_
Ceramide_Skin_Barrier_Repair_Moisture_Gel_2.jpg',
),
Product(
  id: '3',
  title: 'Serum',
  description:
    'Serum Skintific 5X Ceramide Barrier Repair memiliki
kandungan utama 5 jenis ceramide, yaitu ceramide NP, ceramide AP,
ceramide NS, ceramide AS, dan ceramide EOP. Ceramide adalah komponen
penting dari skin barrier, atau lapisan pelindung kulit. Skin barrier
yang sehat dapat melindungi kulit dari berbagai faktor lingkungan yang
dapat merusak kulit, seperti polusi, debu, dan sinar UV.',
  price: 145000,
  imageUrl:
'https://diary.marshabeauty.com/media/review_images/Skintific_4D_Hyalur
onic_Acid_Hydrating_Serum.jpg',
),
Product(
  id: '4',
  title: 'Toner',
  description:
    'Seperti yang disebutkan, 4D Hyaluronic Acid Barrier Essence
Toner dari Skintific menggunakan bahan-bahan terbaik untuk
memformulasikan TTE. Kandungannya terdiri atas empat jenis hyaluronic
acid yang digabungkan menjadi HymagicTMD-4D untuk memaksimalkan
penyerapan hingga ke pori-pori terdalam. Selain itu, keempat jenis
hyaluronic acid ini akan mengunci kelembapan secara intensif dan cepat
dalam hitungan sepuluh detik saja.',
  price: 139000,
```

```

        imageUrl:

'https://diary.marshabeauty.com/media/review_images/Skintific_4D_Hyalur
onic_Acid_Barrier_Essence_Toner.jpg',
    ),
];

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text('Belanjain Store'),
            centerTitle: true,
            backgroundColor: const Color.fromARGB(255, 102, 223, 183)),
        body: GridView.builder(
            padding: const EdgeInsets.all(10),
            gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
                crossAxisCount: 2,
                childAspectRatio: 0.75,
                crossAxisSpacing: 10,
                mainAxisSpacing: 10,
            ),
            itemCount: products.length,
            itemBuilder: (ctx, i) {
                return GestureDetector(
                    onTap: () {
                        Navigator.of(context).push(
                            MaterialPageRoute(
                                builder: (ctx) => ProductDetailScreen(product:
products[i]),
                            ),
                        );
                    },
                    child: Container(
                        decoration: BoxDecoration(
                            color: products[i]
                                .backgroundColor, // Warna latar belakang untuk
kontainer produk
                            borderRadius: BorderRadius.circular(10),
                        ),
                        child: GridTile(
                            child: ClipRRect(
                                borderRadius: BorderRadius.circular(10),

```



```

        child: Image.network(products[i].imageUrl, fit:
BoxFit.cover),
      ),
      footer: Container(
        color: products[i].backgroundColor ??
          const Color.fromARGB(255, 166, 231,
            209), // Terapkan backgroundColor pada footer
        child: GridTileBar(
          backgroundColor: Colors.transparent, // Set agar
transparan

          title: Text(
            products[i].title,
            textAlign: TextAlign.center,
            style: TextStyle(
              color: Colors.black), // Warna teks agar
kontras

          ),
        ),
      ),
    ),
  ),
);
},
),
);
}
}
}
}

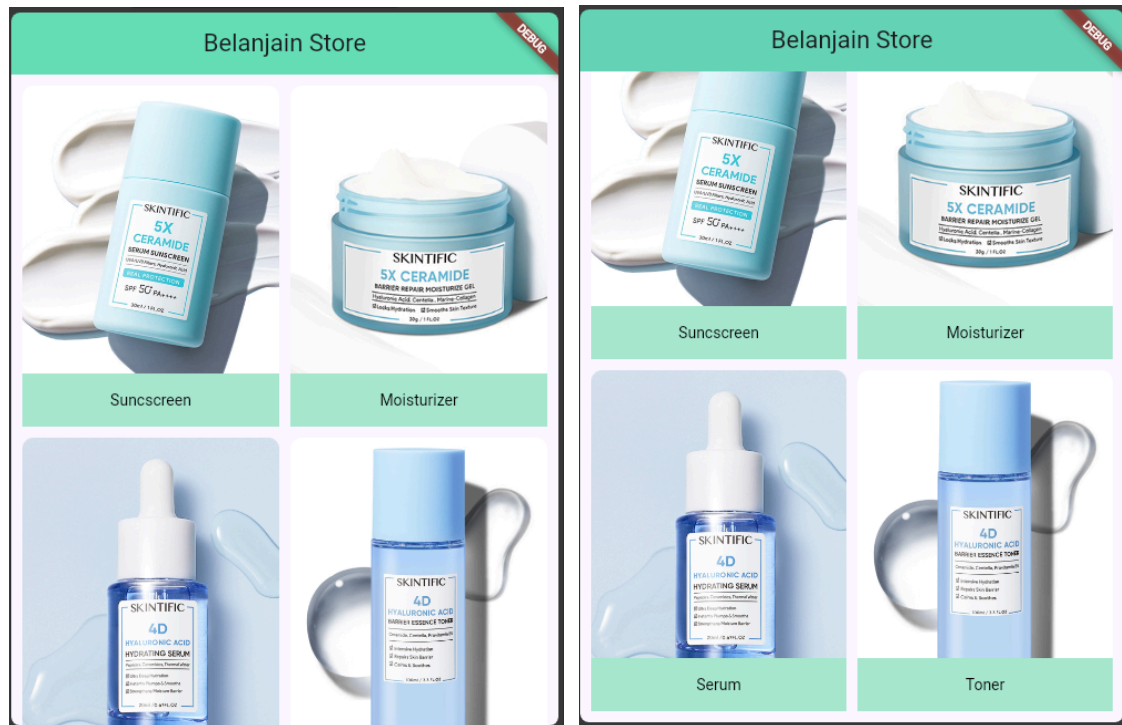
```

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(
15          255, 102, 223, 183), // Sesuaikan dengan warna utama aplikasi
16      ),
17      body: SingleChildScrollView(
18        child: Column(
19          crossAxisAlignment:
20            CrossAxisAlignment.center, // Mengatur posisi ke tengah
21          children: <Widget>[
22            Padding(
23              padding: const EdgeInsets.all(
24                16.0), // Kotak dengan padding di sekitar gambar
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: ClipRRect(
32                  borderRadius: BorderRadius.circular(10),
33                  child: Image.network(product.imageUrl),
34                ),
35              ),
36            ),
37            SizedBox(height: 10),
38            Text(
39              'Rp ${product.price}',
40              style: TextStyle(
41                fontSize: 24,
42                fontWeight: FontWeight.bold,
43                color: Colors.teal,
44              ),
45            ),
46            SizedBox(height: 20),
47            Padding(
48              padding: const EdgeInsets.symmetric(horizontal: 16.0),
49              child: Text(
50                product.description,
51                textAlign:
52                  TextAlign.center, // Mengatur teks deskripsi ke tengah
53                style: TextStyle(fontSize: 16),
54              ),
55            ),
56          ],
57        ),
58      ),
59    );
60  }
61 }
62
```

## Output:

### a. Home Screen



### b. Detail Produk





### Deskripsi Program:

Program ini mengimplementasikan materi class model dan navigasi halaman untuk menampilkan beberapa produk skincare. Setiap produk skincare ditampung dalam kelas **“Product”** yang memiliki beberapa atribut berupa id, title, description, price, dan imageUrl. Atribut ini berfungsi untuk menyimpan informasi dari setiap produk. Selanjutnya daftar produk tersebut ditampilkan di bagian **“Home Screen”** dalam bentuk grid.

Ketika *user* memilih salah satu produk maka akan dialihkan ke halaman detail produk dengan menjalankan fungsi **navigator.push**. Pada bagian ini menerapkan materi dari navigasi halaman. Di halaman detail produk informasi yang lebih lengkap seperti gambar, harga, dan deskripsi singkat dari produk yang dipilih. Ketika ingin kembali ke home screen, *user* bisa menekan tombol panah di bagian pojok kiri atas.