

**LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL 7
NAVIGASI DAN NOTIFIKASI**



Disusun Oleh :

Aditya prabu mukti / 2211104037

SE-06-02

Asisten Praktikum :

Muhammad Faza Zulian Gesit Al Barru

Aisyah Hasna Aulia

Dosen Pengampu :

Yudha Islami Sulistya

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK FAKULTAS
INFORMATIKA**

2024

GUIDED

1. BAGIAN LOGIC

A. MODELS (PRODUCT.DART)

Sourcecode

```
class Product {  
  final int id;  
  final String nama;  
  final double harga;  
  final String gambarUrl;  
  final String deskripsi;  
  
  // Constructor  
  Product({  
    required this.id,  
    required this.nama,  
    required this.harga,  
    required this.gambarUrl,  
    required this.deskripsi,  
  });  
  
  // Method untuk mengonversi Json => Object Product  
  factory Product.fromJson(Map<String, dynamic> json) {  
    return Product(  
      id: json['id'],  
      nama: json['nama'],  
      harga: json['harga'],  
      gambarUrl: json['gambarUrl'],  
      deskripsi: json['deskripsi'],  
    );  
  }  
  
  // Method untuk mengonversi Object Product => Json  
  Map<String, dynamic> toJson() {  
    return {  
      "id": id,  
      "nama": nama,  
      "harga": harga,  
      "gambarUrl": gambarUrl,  
      "deskripsi": deskripsi,  
    };  
  }  
}
```

Deskripsi Program

Kelas Product ini berfungsi sebagai struktur data untuk produk di aplikasi e-commerce, dengan properti seperti id, nama, harga, gambarUrl, dan deskripsi. Konstruktor pada kelas ini memastikan bahwa semua properti terisi saat objek Product dibuat. Kelas ini memiliki metode fromJson untuk mengonversi data JSON menjadi objek Product, yang berguna ketika data diambil dari API, serta metode toJson untuk mengonversi objek Product menjadi JSON, mempermudah pengiriman data ke API atau penyimpanan data. Dengan struktur ini, pengelolaan data produk dalam aplikasi menjadi lebih efisien.

B. BUILD GRADLE

Kita membuat 2 Build Gradle yang 1 untuk Android/App dan Android

Sourcecode android

```
allprojects {
    repositories {
        google()
        mavenCentral()
    }
}

rootProject.buildDir = "../build"

subprojects {
    project.buildDir = "${rootProject.buildDir}/${project.name}"
}

subprojects {
    project.evaluationDependsOn(":app")
}

tasks.register("clean", Delete) {
    delete rootProject.buildDir
}

buildscript {
    repositories {
        google()
        mavenCentral()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:7.3.1' // Ensure this is inside the
buildscript block
    }
}
```

Sourcecode android/app

```
plugins {
    id "com.android.application"
    id "kotlin-android"
    // The Flutter Gradle Plugin must be applied after the Android and Kotlin Gradle
    plugins.
    id "dev.flutter.flutter-gradle-plugin"
}

android {
    namespace = "com.example.pertemuan73"
    compileSdk = flutter.compileSdkVersion
    ndkVersion = flutter.ndkVersion

    defaultConfig {
        applicationId = "com.example.praktikum73"
        minSdkVersion 21
        targetSdkVersion flutter.targetSdkVersion
        versionCode flutter.versionCode
        versionName flutter.versionName
        multiDexEnabled true
    }

    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }

    kotlinOptions {
        jvmTarget = "1.8"
    }
}

dependencies {
    coreLibraryDesugaring 'com.android.tools:desugar_jdk_libs:1.2.2'
}
```

Deskripsi Program

File build.gradle dalam proyek Android berfungsi untuk mengatur konfigurasi build dan dependensi. Terdapat dua file utama: build.gradle di tingkat root proyek, yang mengelola pengaturan global seperti versi Gradle, repositori, dan plugin untuk seluruh modul, serta build.gradle di folder app, yang menangani pengaturan spesifik aplikasi, seperti versi SDK, versi aplikasi, dependensi aplikasi, serta konfigurasi tipe build (debug, release) dan signing untuk produksi. Kedua file ini bekerja secara

bersama untuk memastikan proyek dapat dibangun dengan benar dan memiliki semua dependensi yang dibutuhkan.

C. ANDROIDMANIFEST.XML

Sourcecode

```

<manifest xmlns:android="http://schemas.android.com/apk/res/android">

    <application
        android:label="praktikum_07"
        android:name="${applicationName}"
        android:icon="@mipmap/ic_launcher">

        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:launchMode="singleTop"
            android:taskAffinity=""
            android:theme="@style/LaunchTheme"

            android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallest
ScreenSize|locale|layoutDirection|fontScale|screenLayout|density|uiMode"
            android:hardwareAccelerated="true"
            android:windowSoftInputMode="adjustResize">

                <!-- Specifies an Android theme to apply to this Activity as soon as
                the Android process has started. This theme is visible to the user
                while the Flutter UI initializes. After that, this theme continues
                to determine the Window background behind the Flutter UI. -->
                <meta-data
                    android:name="io.flutter.embedding.android.NormalTheme"
                    android:resource="@style/NormalTheme" />

                <intent-filter>
                    <action android:name="android.intent.action.MAIN"/>
                    <category android:name="android.intent.category.LAUNCHER"/>
                </intent-filter>
            </activity>

            <!-- Don't delete the meta-data below.
            This is used by the Flutter tool to generate
            GeneratedPluginRegistrant.java -->
            <meta-data
                android:name="flutterEmbedding"
                android:value="2" />
        </application>

        <!-- Required to query activities that can process text, see:

```


<https://developer.android.com/training/package-visibility> and

https://developer.android.com/reference/android/content/Intent#ACTION_PROCESS_TEXT.

In particular, this is used by the Flutter engine in `io.flutter.plugin.text.ProcessTextPlugin`. -->

```
<queries>
  <intent>
    <action android:name="android.intent.action.PROCESS_TEXT"/>
    <data android:mimeType="text/plain"/>
  </intent>
</queries>

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

</manifest>
```

Deskripsi program

Dalam kode `AndroidManifest.xml`, terdapat tiga izin yang memberikan akses aplikasi ke fitur tertentu di perangkat: `android.permission.VIBRATE` memungkinkan aplikasi mengontrol fungsi getar, yang sering digunakan untuk efek haptic atau notifikasi; `android.permission.RECEIVE_BOOT_COMPLETED` memungkinkan aplikasi menerima notifikasi ketika perangkat selesai melakukan booting, berguna untuk memulai layanan otomatis seperti sinkronisasi data; dan `android.permission.POST_NOTIFICATIONS` memberikan izin kepada aplikasi untuk menampilkan notifikasi kepada pengguna, yang diperlukan pada versi Android terbaru agar aplikasi dapat mengirim pemberitahuan. Ketiga izin ini mendukung pengalaman pengguna dengan menyediakan fitur notifikasi, getaran, dan layanan otomatis sesuai kebutuhan aplikasi.

2. BAGIAN TAMPILAN APLIKASI

Setelah kita mengkonfigurasi perizinan-perizinan dan mendefinisikan class untuk `Product`, kita tinggal merancang tampilan aplikasi beserta notifications untuk menandakan bahwa aplikasi beserta notifikasi berjalan dengan baik

A. MYPAGE

Sourcecode:

```

import 'package:flutter/material.dart';
import 'package:pertemuan72/models/product.dart';
import 'package:pertemuan72/pages/detailpage.dart';
import 'package:pertemuan72/main.dart'; // Import main.dart untuk akses
showNotification

class MyPage extends StatelessWidget {
  MyPage({super.key});

  final List<Product> products = [
    Product(
      id: 1,
      nama: "Mouse",
      harga: 300000.00,
      gambarUrl:
'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/products/g502x-plus/gallery/g502x-plus-gallery-1-black.png?v=1',
      deskripsi: 'Mouse Gaming cakep',
    ),
    Product(
      id: 2,
      nama: "Keyboard Mechanical",
      harga: 500000.00,
      gambarUrl:
'https://resource.logitech.com/w_1600,c_limit,q_auto,f_auto,dpr_1.0/d_transparent.gif/content/dam/logitech/en/products/keyboards/mx-mechanical/gallery/mxmechanical-keyboard-top-view-graphite-us.png?v=1',
      deskripsi: 'Keyboard mechanical cakep',
    ),
    Product(
      id: 3,
      nama: "Headphone Gaming",
      harga: 450000.00,
      gambarUrl: 'https://m.media-amazon.com/images/I/61CGHv6kmWL.AC_UF894,1000_QL80.jpg',
      deskripsi: 'Headphone cakep',
    ),
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Class Model"),
        centerTitle: true,
        backgroundColor: Colors.amber,

```

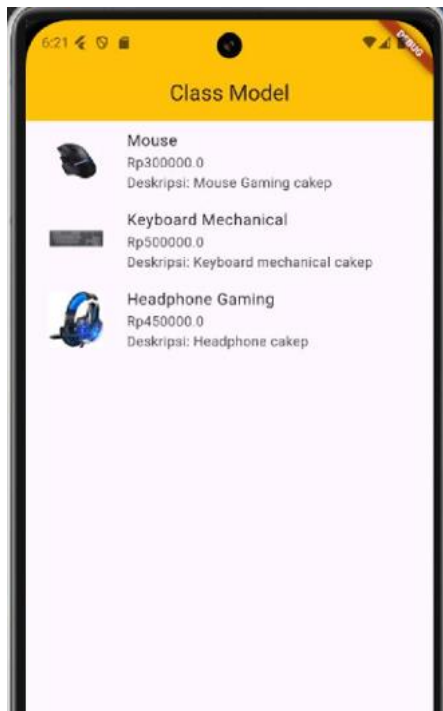
```

    ),
    body: ListView.builder(
      itemCount: products.length,
      itemBuilder: (context, index) {
        final product = products[index];
        return ListTile(
          leading: Image.network(
            product.gambarUrl,
            width: 70,
            height: 70,
          ),
          title: Text(product.nama),
          subtitle: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text('Rp$ {product.harga}'),
              Text('Deskripsi: $ {product.deskripsi}'),
            ],
          ),
          onTap: () {
            // Memanggil notifikasi saat item diklik
            showNotification(product.nama, "Harga: Rp$ {product.harga}");

            // Navigasi ke DetailPage
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (_) => DetailPage(
                  data: Text(product.nama),
                ),
              ),
            );
          },
        );
      },
    );
  },
);
}
}

```

Screenshoot output



Deskripsi Program

Kode ini mendefinisikan widget MyPage, sebuah halaman Flutter yang menampilkan daftar produk menggunakan ListView.builder. Daftar produk disimpan dalam List<Product> products, di mana setiap produk memiliki properti seperti id, nama, harga, gambarUrl, dan deskripsi. Pada build method, setiap item dalam daftar ditampilkan sebagai ListTile yang mencakup gambar, nama, harga, dan deskripsi produk. Ketika item dipilih, fungsi showNotification (diimpor dari main.dart) akan dipanggil untuk menampilkan notifikasi berisi nama dan harga produk, kemudian aplikasi menavigasi ke halaman DetailPage melalui Navigator.push untuk menampilkan detail produk lebih lengkap. Tampilan keseluruhan memiliki AppBar dengan judul "Class Model" dan latar belakang berwarna amber.

B. DETAILPAGE

Sourcecode

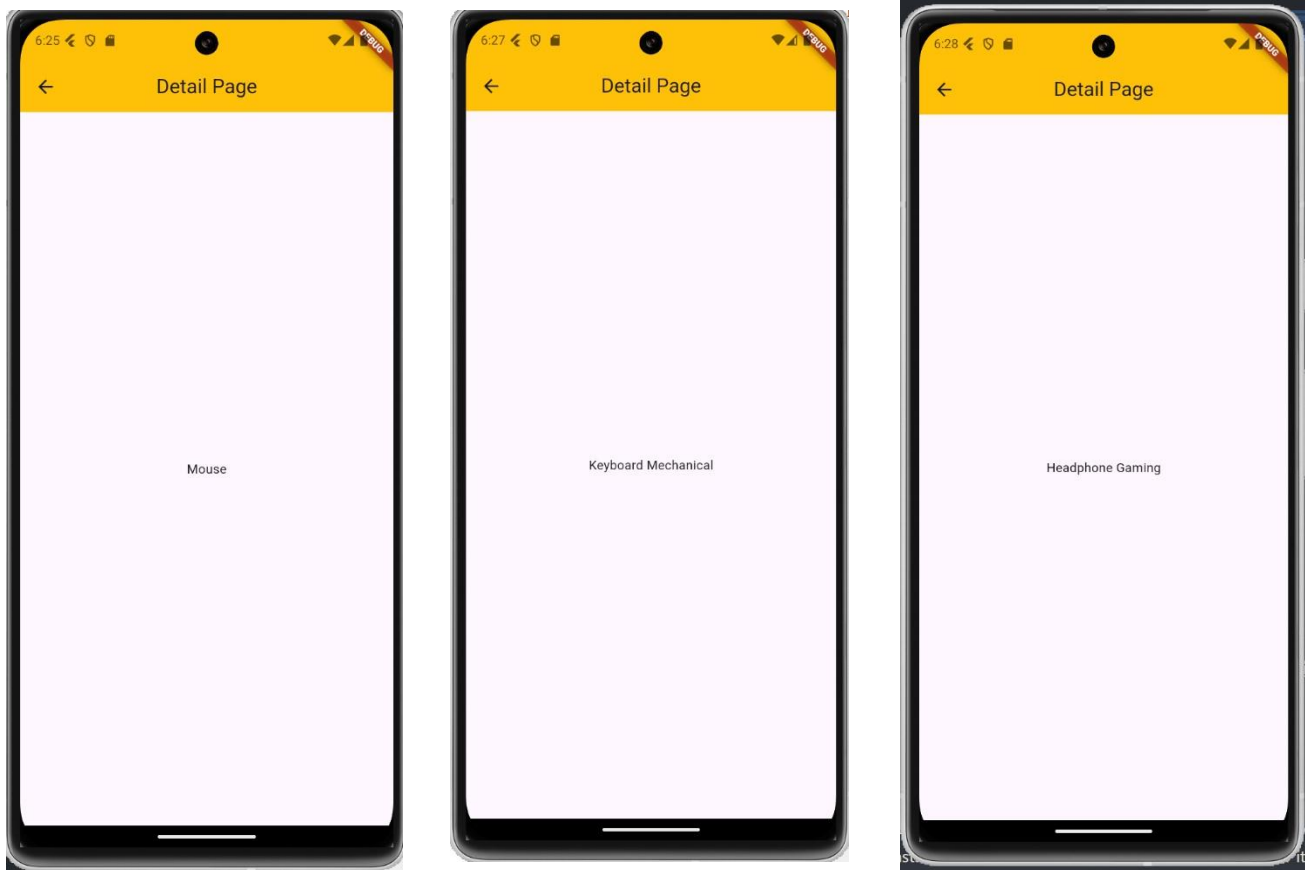
```
import 'package:flutter/material.dart';

class DetailPage extends StatelessWidget {
  const DetailPage({super.key, required this.data});

  final Widget data;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Detail Page'),
        centerTitle: true,
        backgroundColor: Colors.amber,
      ),
      body: Center(
        child: data,
      ),
    );
  }
}
```

Screenshoot Output



Deskripsi Program

Program ini mendefinisikan widget `DetailPage`, sebuah halaman sederhana di Flutter yang menerima parameter data dalam bentuk Widget untuk menampilkan konten detail. `DetailPage` memiliki `AppBar` di bagian atas dengan judul "Detail Page" dan latar belakang berwarna amber, mempertahankan konsistensi tampilan dengan halaman sebelumnya. Di dalam body, halaman ini menggunakan `Center` untuk menempatkan data di tengah layar. Widget ini berfungsi untuk menampilkan detail atau konten spesifik yang diteruskan dari halaman lain, seperti produk yang dipilih dari daftar.

C. NOTIFICATIONS

Untuk notifikasi ini dibuat 2 file yaitu `local_notifications.dart` dan `notification_screen.dart`.

local_notifications :

Sourcecode

```
import 'package:flutter_local_notifications/flutter_local_notifications.dart';

// Inisialisasi instance dari FlutterLocalNotificationsPlugin
final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =
  FlutterLocalNotificationsPlugin();

// Inisialisasi notifikasi di dalam class NotificationService
class NotificationService {
  static Future<void> initialize() async {
    const AndroidInitializationSettings initializationSettingsAndroid =
      AndroidInitializationSettings('@mipmap/ic_launcher');

    const DarwinInitializationSettings initializationSettingsIOS =
      DarwinInitializationSettings();

    const InitializationSettings initializationSettings = InitializationSettings(
      android: initializationSettingsAndroid,
      iOS: initializationSettingsIOS,
    );

    await flutterLocalNotificationsPlugin.initialize(initializationSettings);
  }

  static Future<void> showNotification() async {
    const AndroidNotificationDetails androidPlatformChannelSpecifics =
      AndroidNotificationDetails(
        'your_channel_id', // ID unik untuk channel
        'your', // Nama channel
        channelDescription: 'your_channel_description', // Deskripsi channel
        importance: Importance.max,
        priority: Priority.high,
        showWhen: true,
      );

    const NotificationDetails platformChannelSpecifics = NotificationDetails(
      android: androidPlatformChannelSpecifics,
    );

    await flutterLocalNotificationsPlugin.show(
      0, // ID notifikasi
      'Hello!', // Judul notifikasi
      'This is a local notification.', // Isi notifikasi
      platformChannelSpecifics,
    );
  }
}
```

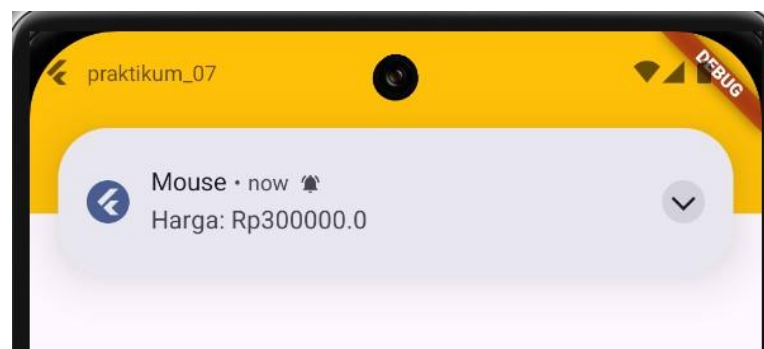
```
}  
}
```

notification_screen.dart:

Sourcecode

```
import 'package:flutter/material.dart';  
import 'package:pertemuan72/local_notifications.dart';  
  
class NotificationScreen extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text("Local Notifications"),  
      ),  
      body: Center(  
        child: ElevatedButton(  
          onPressed: NotificationService.showNotification, // Panggil fungsi notifikasi  
          child: Text("Show Notification"),  
        ),  
      ),  
    );  
  }  
}
```

Screenshot Output



Deskripsi Program

Kode ini terdiri dari dua bagian untuk mengimplementasikan notifikasi lokal di Flutter. Pada bagian pertama, widget NotificationScreen menampilkan halaman dengan tombol

"Show Notification". Ketika tombol ditekan, metode showNotification dari NotificationService dipanggil untuk menampilkan notifikasi. Bagian kedua mendefinisikan NotificationService, yang memanfaatkan FlutterLocalNotificationsPlugin untuk mengelola notifikasi lokal. Fungsi initialize melakukan konfigurasi awal notifikasi di Android dan iOS, termasuk menampilkan ikon aplikasi sebagai ikon notifikasi. Metode showNotification mengatur detail notifikasi dengan AndroidNotificationDetails, menetapkan ID channel, nama, deskripsi, serta tingkat kepentingan dan prioritasnya. Notifikasi ditampilkan dengan ID 0, judul "Hello!", dan pesan "This is a local notification." Kode ini memungkinkan aplikasi menampilkan notifikasi lokal dengan konfigurasi yang sudah ditetapkan setiap kali tombol ditekan.

D. MAIN PROGRAM

Sourcecode:

```
import 'package:flutter/material.dart';
import 'package:flutter_local_notifications/flutter_local_notifications.dart';
import 'package:pertemuan72/pages/mypage.dart';
import 'package:permission_handler/permission_handler.dart';

final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =
FlutterLocalNotificationsPlugin();

// Pindahkan fungsi showNotification ke luar main agar dapat diakses di seluruh
aplikasi
Future<void> showNotification(String title, String body) async {
  const AndroidNotificationDetails androidPlatformChannelSpecifics =
AndroidNotificationDetails(
    'yovvv', // ID unik untuk channel
    'your_channel_name', // Nama channel
    channelDescription: 'your_channel_description', // Deskripsi channel
    importance: Importance.max,
    priority: Priority.high,
    showWhen: true,
  );

  const NotificationDetails platformChannelSpecifics = NotificationDetails(
    android: androidPlatformChannelSpecifics,
  );

  await flutterLocalNotificationsPlugin.show(
    0, // ID notifikasi
    title, // Judul notifikasi
    body, // Isi notifikasi
    platformChannelSpecifics,
  );
}
```

```

}

void main() async {
  WidgetsFlutterBinding.ensureInitialized();

  // Minta izin notifikasi untuk Android 13 dan lebih tinggi
  if (await Permission.notification.isDenied) {
    await Permission.notification.request();
  }

  // Inisialisasi notifikasi lokal
  const AndroidInitializationSettings initializationSettingsAndroid =
  AndroidInitializationSettings('@mipmap/ic_launcher');
  const DarwinInitializationSettings initializationSettingsIOS =
  DarwinInitializationSettings();
  const InitializationSettings initializationSettings = InitializationSettings(
    android: initializationSettingsAndroid,
    iOS: initializationSettingsIOS,
  );

  await flutterLocalNotificationsPlugin.initialize(initializationSettings);

  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Local Notifications',
      theme: ThemeData(primarySwatch: Colors.blue),
      home: MyPage(),
    );
  }
}

```

Deskripsi Program

Program ini adalah aplikasi Flutter yang memanfaatkan `flutter_local_notifications` untuk menampilkan notifikasi lokal. `FlutterLocalNotificationsPlugin` diinisialisasi untuk memungkinkan notifikasi dalam aplikasi, dan fungsi `showNotification` dirancang agar dapat dipanggil di seluruh aplikasi dengan judul dan isi notifikasi yang bersifat dinamis. Dalam fungsi `main()`, aplikasi memeriksa izin notifikasi untuk Android 13 dan versi lebih tinggi; jika izin ditolak, aplikasi akan meminta pengguna untuk memberikan izin. Konfigurasi `InitializationSettings` digunakan untuk mengatur notifikasi pada Android dan iOS, termasuk penentuan ikon aplikasi sebagai ikon notifikasi. Aplikasi ini dijalankan melalui widget `MyApp`, yang menampilkan halaman `MyPage`, di

mana fungsi `showNotification` dapat dipanggil untuk menampilkan notifikasi sesuai dengan interaksi pengguna.

UNGUIDED

1. (Soal) Buatlah satu project untuk menampilkan beberapa produk dan halaman e-commerce dengan menerapkan class model serta navigasi halaman.

1. MAIN PROGRAM

Sourcecode

```
import 'package:flutter/material.dart';
import 'package:flutter_local_notifications/flutter_local_notifications.dart';
import 'package:permission_handler/permission_handler.dart';
import 'pages/product_list_screen.dart';

final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =
  FlutterLocalNotificationsPlugin();

// Fungsi untuk menampilkan notifikasi di seluruh aplikasi
Future<void> showNotification(String title, String body) async {
  const AndroidNotificationDetails androidPlatformChannelSpecifics =
    AndroidNotificationDetails(
      'channel_id', // ID channel unik
      'channel_name', // Nama channel
      channelDescription: 'Deskripsi channel',
      importance: Importance.max,
      priority: Priority.high,
      showWhen: true,
    );

  const NotificationDetails platformChannelSpecifics = NotificationDetails(
    android: androidPlatformChannelSpecifics,
  );
}
```

```

await flutterLocalNotificationsPlugin.show(
  0, // ID notifikasi
  title, // Judul notifikasi
  body, // Isi notifikasi
  platformChannelSpecifics,
);
}

void main() async {
  WidgetsFlutterBinding.ensureInitialized();

  // Minta izin notifikasi (khususnya untuk Android 13+)
  if (await Permission.notification.isDenied) {
    await Permission.notification.request();
  }

  // Inisialisasi notifikasi lokal
  const AndroidInitializationSettings initializationSettingsAndroid =
    AndroidInitializationSettings('@mipmap/ic_launcher');

  const InitializationSettings initializationSettings = InitializationSettings(
    android: initializationSettingsAndroid,
  );

  await flutterLocalNotificationsPlugin.initialize(initializationSettings);

  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Simple E-commerce App',
      theme: ThemeData(primarySwatch: Colors.blue),
      home: ProductListScreen(),
    );
  }
}

```

2. MODELS (PRODUCT.DART)

Sourcecode

```

class Product {
  final String id;
  final String name;
  final String description;
  final double price;
  final String imageUrl;

  Product({
    required this.id,
    required this.name,
    required this.description,
    required this.price,
    required this.imageUrl,
  });
}

```

3. PRODUCT DETAIL SCREEN

Sourcecode

```

import 'package:flutter/material.dart';
import 'package:pertemuan7_4/models/product.dart';
import 'package:pertemuan7_4/main.dart'; // Import untuk akses showNotification

class ProductDetailScreen extends StatelessWidget {
  final Product product;

  ProductDetailScreen({required this.product});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(product.name),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start, // Teks tidak di tengah
          children: [
            Center(
              child: Image.network(
                product.imageUrl,
                height: 250, // Memperbesar ukuran gambar
                fit: BoxFit.cover,

```

```

    ),
    ),
    SizedBox(height: 20),
    Text(
      product.name,
      style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
    ),
    SizedBox(height: 10),
    Text(
      'Rp$ {product.price.toStringAsFixed(0)}',
      style: TextStyle(fontSize: 20, color: Colors.grey),
    ),
    SizedBox(height: 20),
    Text(
      product.description,
      style: TextStyle(fontSize: 16),
    ),
    Spacer(),
    Center(
      child: ElevatedButton(
        onPressed: () {
          // Tampilkan notifikasi "Produk berhasil dimasukkan ke keranjang!"
          showNotification("Terima Kasih!",
            "Produk berhasil dimasukkan ke keranjang!");
        },
        child: Text("Beli Produk"),
        style: ElevatedButton.styleFrom(
          padding: EdgeInsets.symmetric(horizontal: 30, vertical: 15),
          textStyle: TextStyle(fontSize: 18),
        ),
      ),
    ),
  ],
),
);
}
}

```

4. PRODUCT LIST SCREEN

Sourcecode

```
import 'package:flutter/material.dart';
```

```

import 'package:pertemuan7_4/models/product.dart';
import 'package:pertemuan7_4/pages/product_detail.dart';
import 'package:pertemuan7_4/main.dart';

class ProductListScreen extends StatelessWidget {
  final List<Product> products = [
    Product(
      id: 'p1',
      name: 'ssd m.2 external 1TB',
      description: 'penyimpanan external yang praktis untuk dibawa kemana saja',
      price: 1000000,
      imageUrl:
        'https://images.tokopedia.net/img/cache/900/product-1/2020/7/12/9191266/9191266_c95aa137-3aba-47b8-9301-d39f895b0294_700_700',
    ),
    Product(
      id: 'p2',
      name: 'iphone 15 pro max',
      description:
        'menyediakan berbagai fitur menarik dan kamera yang sangat jernih',
      price: 2600000000,
      imageUrl:
        'https://p-id.ipricegroup.com/uploaded_9413b7b2fa614cae7bafa4c33ce23e2bc2ecaaec.jpg',
    ),
    Product(
      id: 'p3',
      name: 'asus ROG srtix',
      description: 'cocok untuk para gamers yang suka game dengan kualitas dan performa yang sangat bagus',
      price: 290000000,
      imageUrl:
        'https://assets.bmdstatic.com/web/image?model=product.product&field=image_1024&id=71638&unique=6586d82',
    ),
    Product(
      id: 'p4',
      name: 'Antimateri',
      description: 'benda termahal di dunia yang berfungsi sebagai bahan bakar terkuat',
      price: 9250000000000000000,
      imageUrl:
        'https://akcdn.detik.net.id/community/media/visual/2022/08/11/antimateri_169.webp?w=700&q=90',
    ),
    Product(
      id: 'p5',
      name: 'Hot Wheels VW Combi',

```



```

        description: 'hotwheels gagal produk dan menjadi hot wheels pertama di dunia',
        price: 25000000000,
        imageUrl:
            'https://akcdn.detik.net.id/community/media/visual/2023/06/18/hot-wheels-pink-rear-loading-
            beach-bomb-yang-diproduksi-1969_169.jpeg?w=700&q=90',
    ),
];

void onProductTap(BuildContext context, Product product) {
    showNotification(product.name, "Harga: Rp$ {product.price}");

    Navigator.push(
        context,
        MaterialPageRoute(
            builder: (_) => ProductDetailScreen(product: product),
        ),
    );
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text(
                'Embuh Store',
                style: TextStyle(
                    color: Colors.white,
                    fontWeight: FontWeight.bold,
                ),
            ),
            backgroundColor: const Color.fromARGB(255, 3, 217, 250),
        ),
        body: GridView.builder(
            padding: const EdgeInsets.all(10),
            itemCount: products.length,
            gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
                crossAxisCount: 2, // Jumlah kolom
                crossAxisSpacing: 10,
                mainAxisSpacing: 10,
                childAspectRatio: 0.7, // Rasio aspek kartu produk
            ),
            itemBuilder: (context, index) {
                final product = products[index];
                return GestureDetector(
                    onTap: () => onProductTap(context, product),
                    child: Card(

```

```

elevation: 5,
shape: RoundedRectangleBorder(
  borderRadius: BorderRadius.circular(10),
),
child: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    ClipRRect(
      borderRadius: BorderRadius.vertical(top: Radius.circular(10)),
      child: Image.network(
        product.imageUrl,
        width: double.infinity,
        height: 120,
        fit: BoxFit.cover,
      ),
    ),
    Padding(
      padding: const EdgeInsets.all(8.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text(
            product.name,
            style: TextStyle(
              fontSize: 14,
              fontWeight: FontWeight.bold,
            ),
            maxLines: 2,
            overflow: TextOverflow.ellipsis,
          ),
          SizedBox(height: 5),
          Text(
            'Rp${product.price.toStringAsFixed(0)}',
            style: TextStyle(
              fontSize: 12,
              fontWeight: FontWeight.bold,
              color: Colors.red, // Warna merah untuk harga
            ),
          ),
        ],
      ),
    ),
  ],
),
);

```

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

5. NOTIFICATIONS

a. Local Notifications

Sourcecode

```
import 'package:flutter_local_notifications/flutter_local_notifications.dart';  
  
final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =  
  FlutterLocalNotificationsPlugin();  
  
class NotificationService {  
  // Inisialisasi notifikasi  
  static Future<void> initialize() async {  
    const AndroidInitializationSettings initializationSettingsAndroid =  
      AndroidInitializationSettings('@mipmap/ic_launcher');  
  
    const DarwinInitializationSettings initializationSettingsIOS =  
      DarwinInitializationSettings();  
  
    const InitializationSettings initializationSettings =  
      InitializationSettings(  
        android: initializationSettingsAndroid,  
        iOS: initializationSettingsIOS,  
      );  
  
    await flutterLocalNotificationsPlugin.initialize(  
      initializationSettings,  
      onDidReceiveNotificationResponse: (NotificationResponse response) async {  
        // Tambahkan logika jika ada aksi saat notifikasi di-klik  
      },  
    );  
  }  
  
  // Fungsi untuk menampilkan notifikasi  
  static Future<void> showNotification() async {  
    const AndroidNotificationDetails androidPlatformChannelSpecifics =  
      AndroidNotificationDetails(  
        'your_channel_id',  
        'your_channel_name',
```

```

        channelDescription: 'your_channel_description',
        importance: Importance.max,
        priority: Priority.high,
        showWhen: true,
    );

    const NotificationDetails platformChannelSpecifics = NotificationDetails(
      android: androidPlatformChannelSpecifics,
    );

    await flutterLocalNotificationsPlugin.show(
      0,
      'Produk Terpilih',
      'Kabel LAN telah ditambahkan!',
      platformChannelSpecifics,
    );
  }
}

```

b. Notification Screen

Sourcecode

```

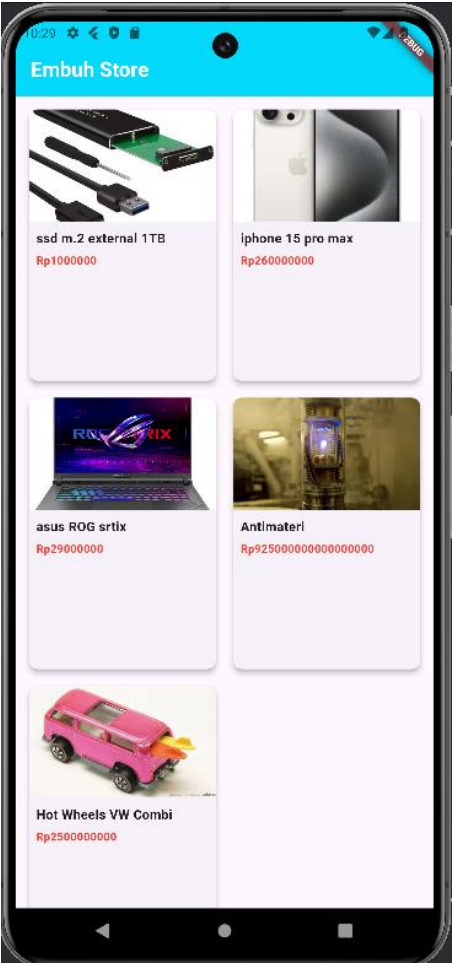
import 'package:flutter/material.dart';
import '../local_notification.dart';

class NotificationScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Local Notifications")),
      body: Center(
        child: ElevatedButton(
          onPressed: NotificationService.showNotification,
          child: Text("Show Notification"),
        ),
      ),
    );
  }
}

```

Screenshoot Output

Tampilan Utama



Detail Produk



Notifikasi pada Detail Produk



Notifikasi pada saat button “Beli Produk” diklik



Deskripsi Program

Proyek ini adalah aplikasi e-commerce sederhana menggunakan Flutter yang memiliki beberapa fitur utama seperti halaman daftar produk, halaman detail produk, dan notifikasi lokal. Di halaman daftar produk, setiap item menampilkan nama, harga, dan gambar produk. Ketika pengguna mengetuk suatu produk, aplikasi akan menampilkan notifikasi dengan judul dan harga produk tersebut, lalu mengarahkan pengguna ke halaman detail produk. Pada halaman detail, pengguna bisa melihat deskripsi lengkap produk dan memilih tombol "Beli Produk" yang juga akan memicu notifikasi untuk memberi tahu bahwa produk telah ditambahkan ke keranjang. Notifikasi diatur menggunakan paket `flutter_local_notifications`, dan `permission_handler` digunakan untuk memastikan izin notifikasi di perangkat Android 13 ke atas. Proyek ini mengintegrasikan navigasi, model data produk, dan interaksi pengguna melalui notifikasi untuk menciptakan pengalaman belanja yang sederhana dan mudah digunakan. Konfigurasi tambahan, seperti `pubspec.yaml`, `AndroidManifest.xml`, dan `build.gradle`, sama seperti pada proyek sebelumnya.