LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK

MODUL 7

NAVIGASI DAN NOTIFIKASI

Disusun Oleh :

Dimas Cahyo Margono / 2211104060

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

```dart
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 mendefinisikan struktur data untuk produk dalam aplikasi e-commerce, dengan properti seperti `id`, `nama`, `harga`, `gambarUrl`, dan `deskripsi`. Konstruktor pada kelas ini memastikan semua properti diisi ketika objek `Product` dibuat. Terdapat metode `fromJson` untuk mengonversi data JSON menjadi objek `Product`, yang berguna saat mengambil data dari API, dan metode `toJson` untuk mengonversi objek `Product` menjadi JSON, mempermudah pengiriman data ke API atau penyimpanan data. Struktur ini memudahkan pengelolaan data produk secara efisien dalam aplikasi.

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 {
        multiDexEnabled true
  }

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

    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 mengatur pengaturan global seperti versi Gradle, repositori, dan plugin untuk seluruh modul, serta `build.gradle` di folder `app`, yang mengelola pengaturan spesifik aplikasi, seperti versi SDK, versi aplikasi, dependensi aplikasi, serta konfigurasi tipe build (debug, release) dan signing untuk produksi. Kedua file ini bekerja bersama untuk memastikan proyek terbangun dengan benar dan memiliki semua dependensi yang diperlukan.

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

D. PUBSPEC.YAML

Sourcecode

```
name: pertemuan72
description: "A new Flutter project."
# The following line prevents the package from being accidentally published to
# pub.dev using `flutter pub publish`. This is preferred for private packages.
publish_to: 'none' # Remove this line if you wish to publish to pub.dev

# The following defines the version and build number for your application.
# A version number is three numbers separated by dots, like 1.2.43
# followed by an optional build number separated by a +.
# Both the version and the builder number may be overridden in flutter
# build by specifying --build-name and --build-number, respectively.
# In Android, build-name is used as versionName while build-number used as
versionCode.
# Read more about Android versioning at
https://developer.android.com/studio/publish/versioning
# In iOS, build-name is used as CFBundleShortVersionString while build-number is used
as CFBundleVersion.
# Read more about iOS versioning at
#
https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistK
eyReference/Articles/CoreFoundationKeys.html
# In Windows, build-name is used as the major, minor, and patch parts
# of the product and file versions while build-number is used as the build suffix.
version: 1.0.0+1

environment:
  sdk: ^3.5.3
```

```yaml
# Dependencies specify other packages that your package needs in order to work.
# To automatically upgrade your package dependencies to the latest versions
# consider running `flutter pub upgrade --major-versions`. Alternatively,
# dependencies can be manually updated by changing the version numbers below to
# the latest version available on pub.dev. To see which dependencies have newer
# versions available, run `flutter pub outdated`.
dependencies:
  permission_handler: ^10.2.0
  flutter:
    sdk: flutter



  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.8
  flutter_local_notifications: ^17.2.4

dev_dependencies:
  flutter_test:
    sdk: flutter

  # The "flutter_lints" package below contains a set of recommended lints to
  # encourage good coding practices. The lint set provided by the package is
  # activated in the `analysis_options.yaml` file located at the root of your
  # package. See that file for information about deactivating specific lint
  # rules and activating additional ones.
  flutter_lints: ^4.0.0

# For information on the generic Dart part of this file, see the
# following page: https://dart.dev/tools/pub/pubspec

# The following section is specific to Flutter packages.
flutter:

  # The following line ensures that the Material Icons font is
  # included with your application, so that you can use the icons in
  # the material Icons class.
  uses-material-design: true

  # To add assets to your application, add an assets section, like this:
  # assets:
  #   - images/a_dot_burr.jpeg
  #   - images/a_dot_ham.jpeg

  # An image asset can refer to one or more resolution-specific "variants", see
  # https://flutter.dev/to/resolution-aware-images

  # For details regarding adding assets from package dependencies, see
  # https://flutter.dev/to/asset-from-package

  # To add custom fonts to your application, add a fonts section here,
```

```
  # in this "flutter" section. Each entry in this list should have a
  # "family" key with the font family name, and a "fonts" key with a
  # list giving the asset and other descriptors for the font. For
  # example:
  # fonts:
  #   - family: Schyler
  #     fonts:
  #       - asset: fonts/Schyler-Regular.ttf
  #       - asset: fonts/Schyler-Italic.ttf
  #         style: italic
  #   - family: Trajan Pro
  #     fonts:
  #       - asset: fonts/TrajanPro.ttf
  #       - asset: fonts/TrajanPro_Bold.ttf
  #         weight: 700
  #
  # For details regarding fonts from package dependencies,
  # see https://flutter.dev/to/font-from-package
```

Deskripsi Program

Program ini adalah konfigurasi `pubspec.yaml` untuk proyek Flutter bernama pertemuan72. File ini mengatur metadata proyek, seperti versi aplikasi (`version: 1.0.0+1`) dan dependensi, termasuk paket `permission_handler` untuk mengelola izin aplikasi, `flutter_local_notifications` untuk mengirim notifikasi, serta `cupertino_icons` untuk ikon bergaya iOS. Versi SDK Dart yang digunakan adalah `^3.5.3`, dan `flutter_lints` ditambahkan sebagai alat bantu untuk mengikuti praktik pengkodean yang baik. Pengaturan ini memungkinkan aplikasi untuk menggunakan ikon Material, mengelola izin perangkat, dan mengirim notifikasi, sesuai dengan fungsionalitas dasar proyek Flutter.

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

```dart
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});
```

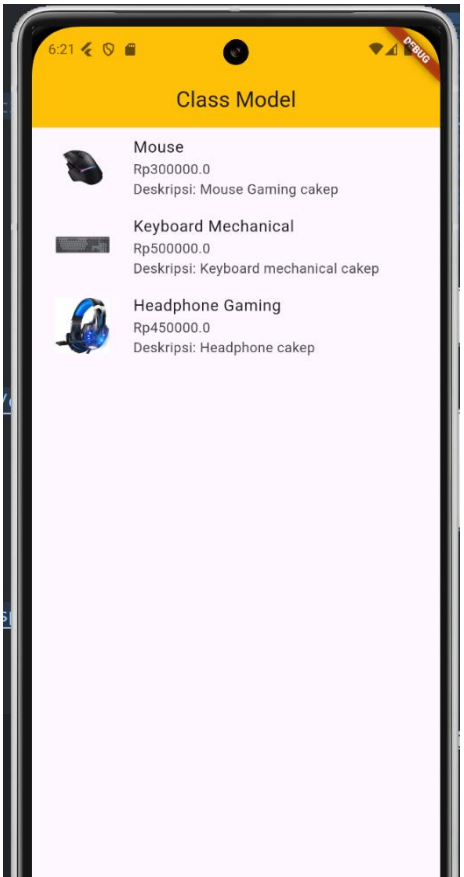
```
  final List<Product> products = [
    Product(
      id: 1,
      nama: "Mouse",
      harga: 300000.00,
      gambarUrl:
          'https://resource.logitechg.com/w_386,ar_1.0,c_limit,f_auto,q_auto,dpr_2.0/d
_transparent.gif/content/dam/gaming/en/products/g502x-plus/gallery/g502x-plus-gallery-
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_transp
arent.gif/content/dam/logitech/en/products/keyboards/mx-mechanical/gallery/mx-
mechanical-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`, halaman Flutter yang menampilkan daftar produk menggunakan `ListView.builder`. Produk-produk didefinisikan dalam `List<Product> products`, setiap produk memiliki properti seperti `id`, `nama`, `harga`, `gambarUrl`, dan `deskripsi`. Pada `build` method, setiap item dalam daftar ditampilkan sebagai `ListTile` dengan gambar, nama, harga, dan deskripsi produk. Ketika item diklik, `showNotification` dipanggil (diimpor dari `main.dart`) untuk menampilkan notifikasi dengan nama dan harga produk, dan kemudian aplikasi menavigasi ke halaman `DetailPage` menggunakan `Navigator.push`, menampilkan detail produk lebih lanjut. Tampilan keseluruhan disertai `AppBar` dengan judul "Class Model" dan latar belakang berwarna amber.

B. DETAILPAGE

Sourcecode

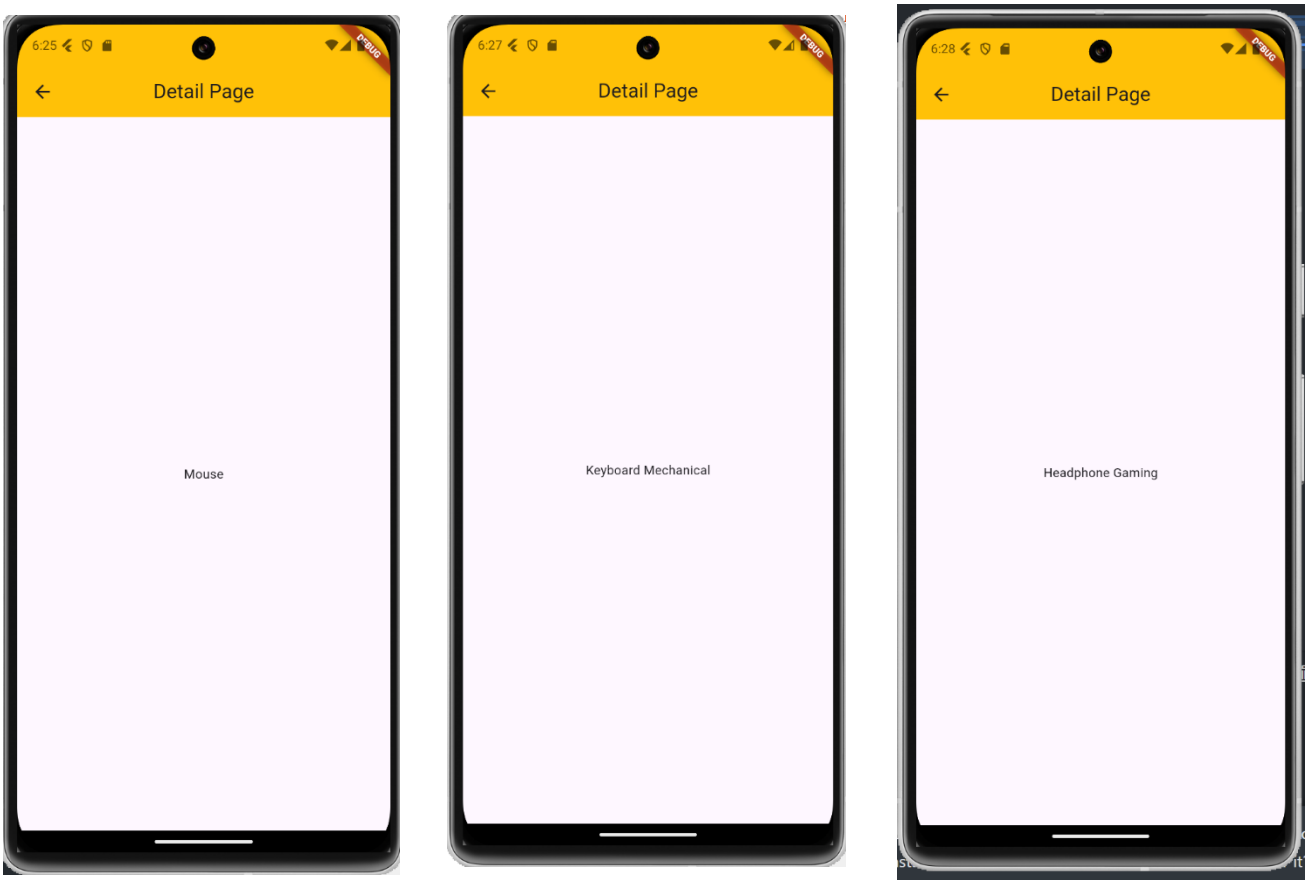
```dart
import 'package:flutter/material.dart';

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 detail konten. `DetailPage` menampilkan `AppBar` di bagian atas dengan judul "Detail Page" dan latar belakang berwarna amber, yang membuat tampilannya konsisten dengan halaman sebelumnya. Di dalam `body`, halaman ini menggunakan `Center` untuk menempatkan `data` di tengah layar. Widget ini berguna untuk menampilkan detail atau konten tertentu 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. Berikut adalah kodenya untuk local_notifications

Sourcecode

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

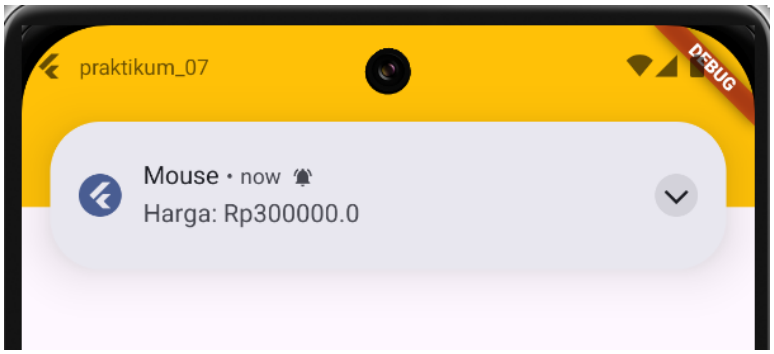
Selanjutnya, kita buat kode notification_screen.dart sebagai berikut

Sourcecode

```dart
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"),
        ),
      ),
    );
  }
}
```

Screenshoot Output



Deskripsi Program

Kode ini terdiri dari dua bagian untuk mengimplementasikan local notifications di Flutter. Pada bagian pertama, widget `NotificationScreen` menampilkan halaman dengan tombol "Show

Notification". Saat tombol ditekan, metode `showNotification` dari `NotificationService` dipanggil untuk menampilkan notifikasi. Bagian kedua mendefinisikan `NotificationService`, yang menggunakan `FlutterLocalNotificationsPlugin` untuk mengelola notifikasi lokal. Fungsi `initialize` mengatur pengaturan awal untuk notifikasi di Android dan iOS, termasuk ikon aplikasi sebagai ikon notifikasi. Metode `showNotification` mengonfigurasi detail notifikasi dengan `AndroidNotificationDetails`, menetapkan ID channel, nama, dan deskripsi, serta tingkat kepentingan dan prioritasnya. Notifikasi kemudian ditampilkan dengan ID `0`, judul "Hello!", dan isi "This is a local notification." Kode ini memungkinkan aplikasi menampilkan notifikasi lokal dengan konfigurasi yang ditentukan 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
  await InitializationSettings();

  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 menggunakan `flutter_local_notifications` untuk menampilkan notifikasi lokal. `FlutterLocalNotificationsPlugin` diinisialisasi untuk menampilkan notifikasi di aplikasi, dan fungsi `showNotification` dibuat agar dapat dipanggil di seluruh aplikasi dengan judul dan isi notifikasi dinamis. Pada `main()`, aplikasi memeriksa izin notifikasi untuk Android 13 ke atas, dan jika izin ditolak, aplikasi meminta pengguna untuk memberikan izin. Konfigurasi `InitializationSettings` mengatur notifikasi pada Android dan iOS, termasuk ikon aplikasi sebagai ikon notifikasi. Aplikasi ini dijalankan melalui widget `MyApp` yang menampilkan halaman `MyPage`, di mana `showNotification` dapat dipanggil untuk menampilkan notifikasi sesuai interaksi pengguna.

UNGUIDED

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

Sourcecode

1. MAIN PROGRAM

```
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:pertemuan73/models/product.dart';
import 'package:pertemuan73/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:pertemuan73/models/product.dart';
import 'package:pertemuan73/pages/product_detail.dart';
import 'package:pertemuan73/main.dart'; // Import untuk akses showNotification

class ProductListScreen extends StatelessWidget {
  final List<Product> products = [
    Product(
      id: 'p1',
      name: 'Kabel LAN',
      description: 'Kabel yang dapat menghubungkan koneksi internet',
      price: 30000,
      imageUrl:
          'https://www.vention.id/wp-content/uploads/2022/04/Vention-Network-Cable-
Cat6-RJ45-Cable-Ethernet-Patch-Cable-For-XBox-Computer-Router-1m-2m-3m-1-1-
600x600.jpg',
    ),
    Product(
      id: 'p2',
      name: 'Webcam HD',
      description:
          'Webcam dengan resolusi tinggi untuk panggilan video yang jernih.',
      price: 150000,
      imageUrl:
```

```
          'https://m.media-amazon.com/images/I/71iNwni9TsL._AC_SL1500_.jpg',
    ),
    Product(
      id: 'p3',
      name: 'Modem',
      description: 'Modem untuk koneksi internet yang cepat dan stabil.',
      price: 200000,
      imageUrl:
          'https://images.tokopedia.net/img/cache/200-
square/VqbcmM/2024/7/16/954f4533-bff5-469f-a4ca-77b64c121dcd.jpg.webp?ect=4g',
    ),
    Product(
      id: 'p4',
      name: 'USB Flashdisk',
      description: 'USB Flashdisk berkapasitas besar untuk penyimpanan data.',
      price: 50000,
      imageUrl:
          'https://images.tokopedia.net/img/cache/500-
square/VqbcmM/2022/10/20/a37a23cf-5d33-40bd-a91d-723af7050625.jpg.webp?ect=4g',
    ),
    Product(
      id: 'p5',
      name: 'Charger Laptop',
      description: 'Charger Laptop baru yang dapat mengecas dengan hemat daya',
      price: 250000,
      imageUrl:
          'https://images.tokopedia.net/img/cache/500-
square/VqbcmM/2022/9/24/1395ffe8-2ce6-42a8-b7fd-901d00ffd782.png.webp?ect=4g',
    ),
  ];

  void onProductTap(BuildContext context, Product product) {
    // Menampilkan notifikasi saat produk diklik
    showNotification(product.name, "Harga: Rp${product.price}");

    // Navigasi ke halaman detail produk
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (_) => ProductDetailScreen(product: product),
      ),
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          'Walawe Store',
          style: TextStyle(
            color: Colors.white,
```

```
            fontWeight: FontWeight.bold,
          ),
        ),
        backgroundColor: Colors.green, // AppBar berwarna hijau
      ),
      body: ListView.builder(
        itemCount: products.length,
        itemBuilder: (context, index) {
          final product = products[index];
          return ListTile(
            leading: Image.network(
              product.imageUrl,
              width: 70,
              height: 70,
              fit: BoxFit.cover,
            ),
            title: Text(product.name),
            subtitle: Text('Rp${product.price.toStringAsFixed(0)}'),
            onTap: () => onProductTap(context, product),
          );
        },
      ),
    );
  }
}
```

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

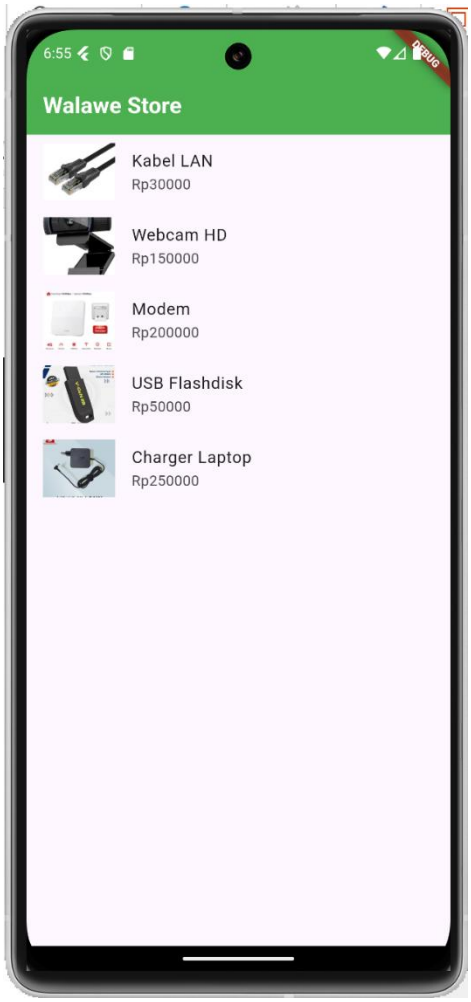
Sourcecode

```
import 'package:flutter/material.dart';
import '../local_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,
          child: Text("Show Notification"),
        ),
```

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

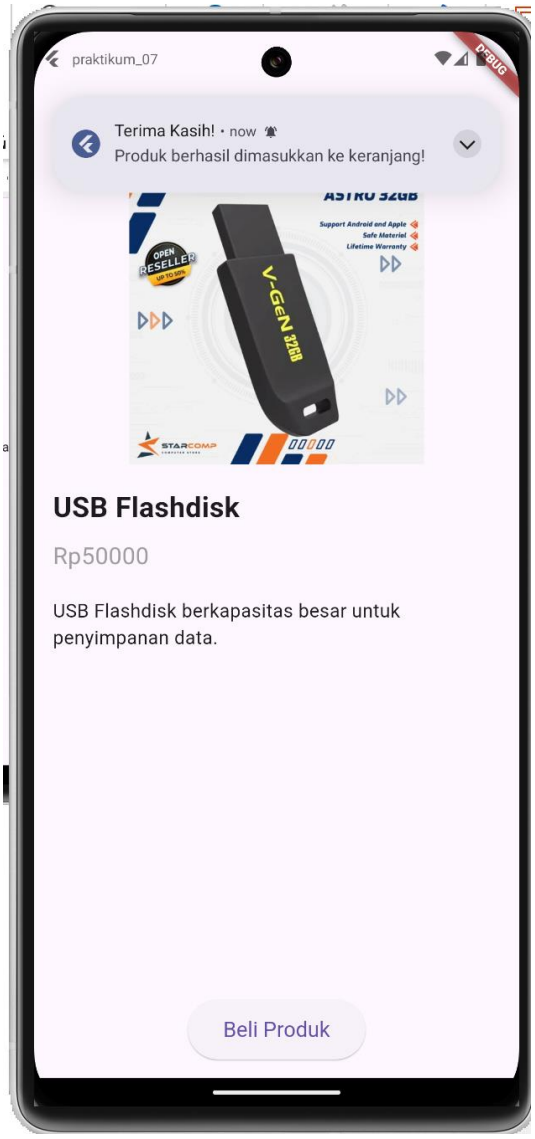
Screenshoot Output



Tampilan Utama

Detail Produk

Notifikasi pada Detail Produk



Notifikasi pada saat button “Beli Produk” diklik

Deskripsi Program

Proyek yang dibuat adalah aplikasi e-commerce sederhana menggunakan Flutter. Proyek ini terdiri dari beberapa halaman yang menampilkan daftar produk, detail produk, dan fitur notifikasi lokal. Pada halaman daftar produk, setiap item menampilkan nama, harga, dan gambar produk. Ketika pengguna mengetuk produk, aplikasi akan menampilkan notifikasi dengan judul produk dan harganya, serta menavigasi ke halaman detail produk. Di halaman detail, pengguna dapat melihat deskripsi lengkap dan memilih tombol "Beli Produk" yang juga memicu notifikasi untuk memberi tahu bahwa produk telah ditambahkan ke keranjang. Konfigurasi notifikasi menggunakan flutter_local_notifications, dan aplikasi juga memanfaatkan permission_handler untuk memastikan izin notifikasi pada perangkat Android 13 ke atas. Proyek ini menggabungkan fitur navigasi, model data produk, dan interaksi pengguna melalui notifikasi untuk pengalaman belanja yang intuitif. Untuk konfigurasi lainnya seperti pubspec.yaml, androidmanifest, dan build.gradle sama seperti project sebelumnya.