

LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK

MODUL 7

NAVIGASI DAN NOTIFIKASI



Disusun Oleh :

Satria Ariq Adelard Dompas/2211104033

SE06-2

Asisten Praktikum :

Ayu Susilowati

Noviana Rizki Anisa Putri

Dosen Pengampu :

Yudha Islami Sulistya

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

A. GUIDED

- Model

Code

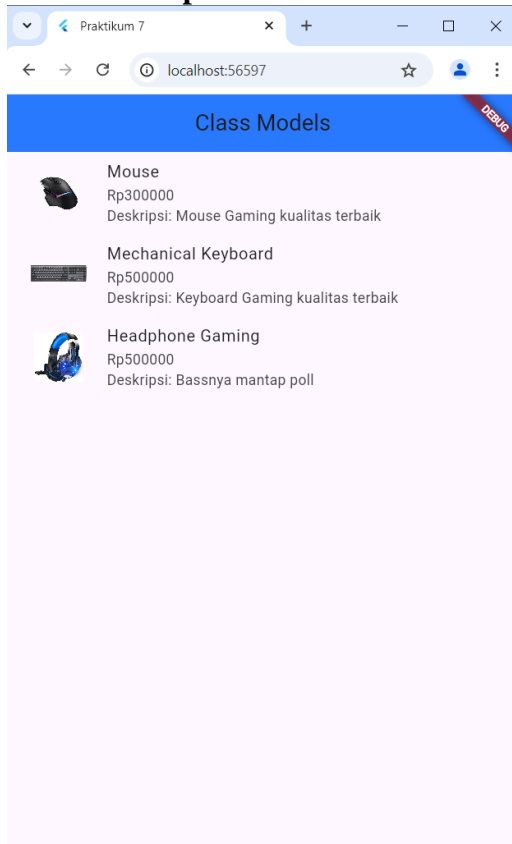
```
class Product {
    final int id;
    final String nama;
    final double harga;
    final String gambarUrl;
    final String deskripsi;

    // constructor
    Product(
        {required this.id,
        required this.nama,
        required this.harga,
        required this.gambarUrl,
        required this.deskripsi});

    // method untuk mengkonversi kedalam objek dengan nama produk
    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 mengkonversi kedalam objek json
    Map<String, dynamic> toJson() => {
        'id': id,
        'nama': nama,
        'harga': harga,
        'gambarUrl': gambarUrl,
        'deskripsi': deskripsi,
    };
}
```

Output



Deskripsi Program

Kode di atas mendefinisikan model yang disebut "Produk" yang mewakili produk dengan beberapa atribut penting seperti ID, nama, harga, URL gambar, dan deskripsi. Model ini menggunakan konstruktor untuk menginisialisasi atribut dengan parameter yang diperlukan. Selain itu, ada dua metode utama: `fromJson` dan `toJson`. Metode `fromJson` digunakan untuk mengonversi data dari format JSON ke objek produk, dan `toJson` mengonversi objek produk ke format JSON. Oleh karena itu, model ini memfasilitasi interaksi antara data yang disimpan dalam format JSON dan objek dalam kode Anda, sehingga memudahkan pengelolaan data produk di aplikasi Anda.

- **Navigation**

Code:

```
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: const Text('Detail Page'),
        centerTitle: true,
        backgroundColor: Colors.blueAccent[400],
      ),
      body: Center(
        child: data, // Menampilkan widget data secara langsung
      ),
    );
  }
}
```

```

import 'package:flutter/material.dart';
import 'package:praktikum_7/models/product.dart';
import 'package:praktikum_7/pages/detailpage.dart';

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

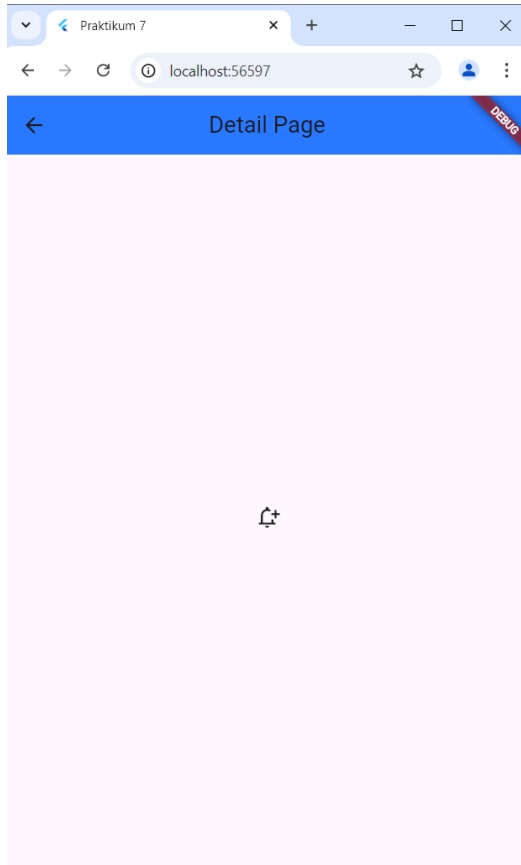
  final List<Product> products = [
    Product(
      id: 1,
      nama: 'Mouse',
      harga: 300000,
      gambarUrl:
        'https://resource.logitechg.com/w_386,ar_1.0,c_limit,f_auto,q_auto,dpr_2.0/d_transparent.gif/content/dam/gaming/en/products/g502x-plus/gallery/g502x-plus-gallery-1-black.png?v=1',
      deskripsi: 'Mouse Gaming kualitas terbaik',
    ),
    Product(
      id: 2,
      nama: 'Mechanical Keyboard',
      harga: 500000,
      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/mx-mechanical-keyboard-top-view-graphite-us.png?v=1',
      deskripsi: 'Keyboard Gaming kualitas terbaik',
    ),
    Product(
      id: 3,
      nama: 'Headphone Gaming',
      harga: 500000,
      gambarUrl:
        'https://m.media-amazon.com/images/I/61CGHv6kmWL.AC_UF894,1000_QL80.jpg',
      deskripsi: 'Bassnya mantap poll',
    ),
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Class Models'),
        centerTitle: true,
        backgroundColor: Colors.blueAccent[400],
      ),
      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: () {  
        print('Tap Layar Berhasil');  
        Navigator.push(  
          context,  
          MaterialPageRoute(  
            builder: (_) => DetailPage(  
              data: Icon(Icons.notification_add_outlined),  
            ),  
          ),  
        );  
      },  
    ),  
  ),  
);  
}  
}
```

Output



Deskripsi Program

Kode di atas terdiri dari dua halaman dalam aplikasi Flutter, `MyPage` dan `DetailPage`, dan menunjukkan cara mengimplementasikan navigasi antar halaman. `MyPage` memiliki daftar produk yang ditampilkan menggunakan `ListView.builder`, dan setiap item dalam daftar adalah `ListTile` yang berisi gambar, nama, harga, dan deskripsi produk. Saat pengguna mengetuk item, navigasi ke Halaman Detail terjadi menggunakan `Navigator.push()`. Dibutuhkan konteks saat ini dan root baru, yaitu `DetailPage`, dan meneruskan objek produk terkait sebagai parameter. `DetailPage` menampilkan informasi produk secara detail dan memiliki tombol yang memungkinkan pengguna melihat notifikasi melalui sistem notifikasi lokal. Proses ini berfokus pada penggunaan navigasi berbasis rute Flutter untuk berpindah antar halaman sambil mempertahankan status dan informasi produk yang dipilih.

- Notifikasi

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

class LocalNotifications {
  static final FlutterLocalNotificationsPlugin
    _flutterLocalNotificationsPlugin = FlutterLocalNotificationsPlugin();

  // Inisialisasi Notifikasi Lokal
  static Future init() async {
    AndroidInitializationSettings initializationSettingsAndroid =
      AndroidInitializationSettings('@mipmap/ic_launcher');
    final DarwinInitializationSettings initializationSettingsDarwin =
      DarwinInitializationSettings(
        onDidReceiveLocalNotification: (id, title, body, payload) => null,
      );
    final LinuxInitializationSettings initializationSettingsLinux =
      LinuxInitializationSettings(defaultActionName: 'Open notification');
    final InitializationSettings initializationSettings =
      InitializationSettings(
        android: initializationSettingsAndroid,
        iOS: initializationSettingsDarwin,
        linux: initializationSettingsLinux);
    _flutterLocalNotificationsPlugin.initialize(
      initializationSettings,
      onDidReceiveNotificationResponse: (details) => null,
    );
  }

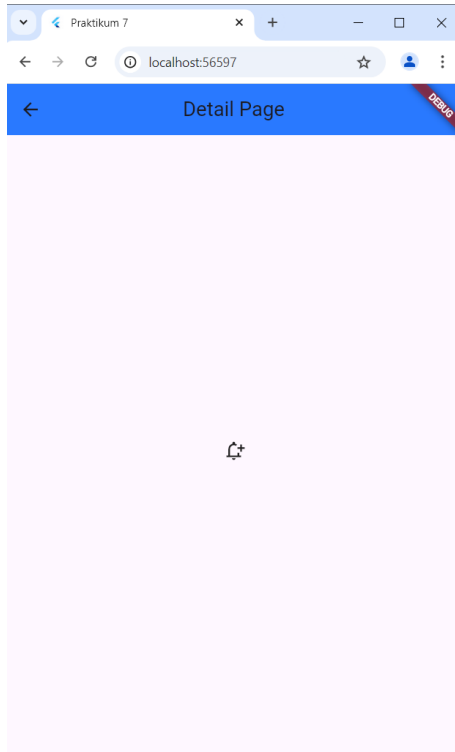
  // Menampilkan notifikasi sederhana
  static Future showSimpleNotification({
    required String title,
    required String body,
    required String payload,
  }) async {
    AndroidNotificationDetails androidNotificationDetails =
      AndroidNotificationDetails(
        'your_channel_id', // Ganti dengan ID channel Anda
        'your_channel_name', // Ganti dengan nama channel Anda
        channelDescription: 'your_channel_description', // Deskripsi channel
        importance: Importance.max,
        priority: Priority.high,
        ticker: 'ticker',
        styleInformation: BigTextStyleInformation(
          body, // Teks yang ditampilkan saat diperluas
          contentType: title, // Judul notifikasi
          summaryText: 'Tap to see more', // Teks ringkasan
        ),
      );

    NotificationDetails notificationDetails =
      NotificationDetails(android: androidNotificationDetails);
```



```
    await _flutterLocalNotificationsPlugin.show(  
      0,  
      title,  
      body,  
      notificationDetails,  
      payload: payload,  
    );  
  }  
}
```

Output



Deskripsi program

Kode di atas terdiri dari dua bagian utama. `DetailPage`, yang menampilkan detail produk, dan kelas `LocalNotifications`, yang mengelola konfigurasi dan tampilan notifikasi lokal. Saat pengguna mengklik tombol Tampilkan Notifikasi di Halaman Detail, fungsi `showSimpleNotification` dari kelas `LocalNotifications` dipanggil untuk menampilkan notifikasi. Kelas `LocalNotifications` menggunakan paket `flutter_local_notifications` untuk menginisialisasi pengaturan notifikasi di seluruh platform (Android, iOS, Linux). Metode `showSimpleNotification` menyiapkan notifikasi dengan detail seperti judul, isi, payload, dan menggunakan `AndroidNotificationDetails` untuk menyetel tampilan dan prioritas notifikasi. Saat notifikasi muncul, pengguna dapat melihat informasi tentang produk, seperti harga dan deskripsi, dan secara opsional dapat mengklik notifikasi tersebut untuk melihat detail lebih lanjut. Oleh karena itu, kode ini menunjukkan cara menerapkan notifikasi lokal di aplikasi Flutter Anda untuk memberikan masukan tentang produk Anda kepada pengguna.

B. Unguided

1. Soal 1

Code

Class model

```
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,  
  });  
}
```

Cartpage

```
import 'package:flutter/material.dart';  
  
class CartPage extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(title: Text('Keranjang Belanja')),  
      body: Center(  
        child: Text('Belum Ada Produk Di Keranjang'),  
      ),  
    );  
  }  
}
```

Homepage

```
import 'package:flutter/material.dart';
import 'package:praktikum_7/models/product.dart';
import 'package:praktikum_7/pages/detailpage.dart';

class HomePage extends StatelessWidget {
  final List<Product> products = [
    Product(
      id: '1',
      name: 'Teh Poci',
      description: 'Teh Poci Khas Tegal',
      price: 18000,
      imageUrl:
        'https://tse2.mm.bing.net/th?id=OIP.ezXU3q_5aMhLtf0ZfvEZHwHaF0&pid=Api&P=0&h=220',
    ),
    Product(
      id: '2',
      name: 'Tahu Aci',
      description: 'Tahu Aci Khas Tegal',
      price: 5000,
      imageUrl:
        'https://2.bp.blogspot.com/-WVPDpA0kqyU/VxhurzYRslI/AAAAAAAAAmU/zlMyE2L3PRYt1RZioD91lHBc8A7bj6NBACLCB/s1600/Resep-Tahu-Aci.jpg',
    ),
    // Tambahkan lebih banyak produk jika diperlukan
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Center(child: Text('Daftar Produk Yang Tersedia')),
        backgroundColor: const Color.fromARGB(255, 75, 255, 108),
      ),
      body: Container(
        color: const Color.fromARGB(255, 132, 148, 255),
        child: ListView.builder(
          itemCount: products.length,
          itemBuilder: (context, index) {
            final product = products[index];
            return Card(
              elevation: 4,
              margin: const EdgeInsets.symmetric(vertical: 10),
              shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(15),
              ),
              child: InkWell(
                onTap: () {
                  Navigator.push(
                    context,
```

```

        MaterialPageRoute(
          builder: (context) => DetailPage(product: product),
        ),
      );
    },
    child: Padding(
      padding: const EdgeInsets.all(10.0),
      child: Row(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          ClipRRect(
            borderRadius: BorderRadius.circular(10),
            child: Image.network(
              product.imageUrl,
              width: 80,
              height: 80,
              fit: BoxFit.cover,
            ),
          ),
          SizedBox(width: 15),
          Expanded(
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text(
                  product.name,
                  style: TextStyle(
                    fontSize: 18,
                    fontWeight: FontWeight.bold,
                    color: Colors.black87,
                  ),
                ),
                SizedBox(height: 5),
                Text(
                  product.description,
                  maxLines: 2,
                  overflow: TextOverflow.ellipsis,
                  style: TextStyle(
                    fontSize: 14,
                    color: Colors.black54,
                  ),
                ),
                SizedBox(height: 10),
                Text(
                  'Rp${product.price}',
                  style: TextStyle(
                    fontSize: 16,
                    fontWeight: FontWeight.w600,
                    color: Colors.blueAccent,
                  ),
                ),
              ],
            ),
          ),
        ],
      ),
    ),
  ),
);

```

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

detailpage

```
import 'package:flutter/material.dart';
import 'package:praktikum_7/models/product.dart';

class DetailPage extends StatelessWidget {
  final Product product;

  DetailPage({required this.product});

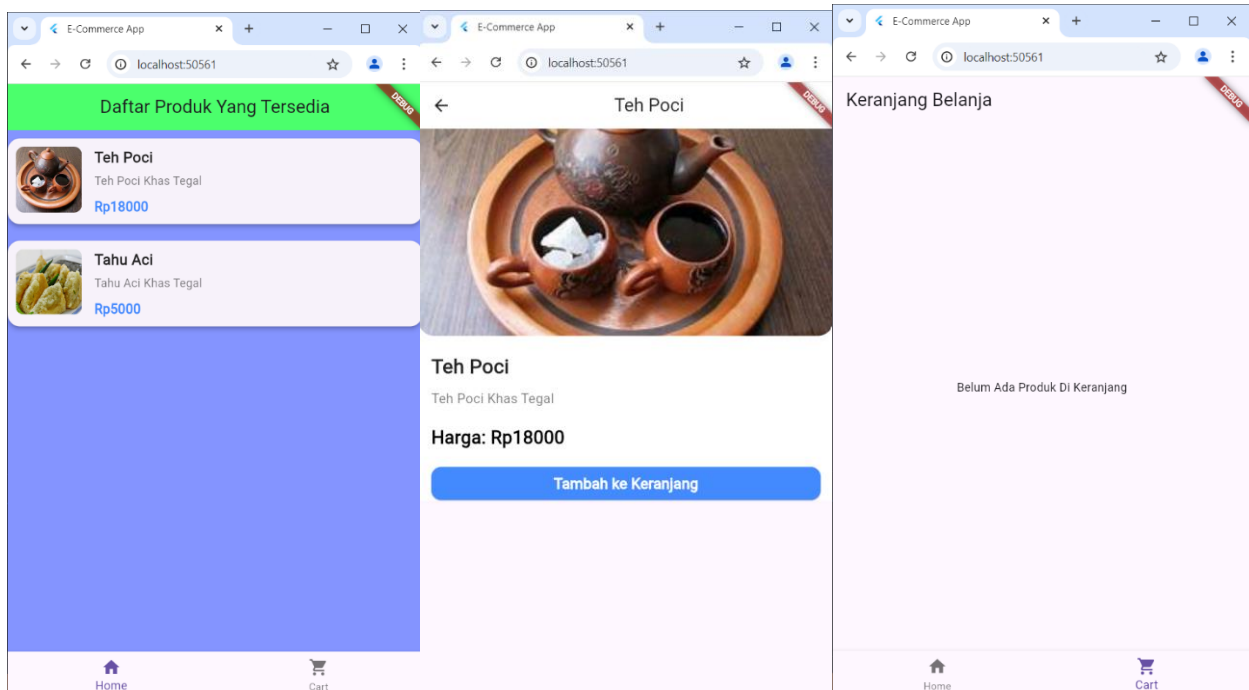
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Center(child: Text(product.name)),
        backgroundColor: Colors.white,
      ),
      body: SingleChildScrollView(
        child: Container(
          color: Colors.white,
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              // Gambar produk
              ClipRRect(
                borderRadius: BorderRadius.only(
                  bottomLeft: Radius.circular(20),
                  bottomRight: Radius.circular(20),
                ),
                child: Image.network(
                  product.imageUrl,
                  width: double.infinity,
                  height: 250,
                  fit: BoxFit.cover,
                ),
              ),
              SizedBox(height: 20),
              // Informasi produk
              Padding(
                padding: const EdgeInsets.symmetric(horizontal: 16.0),
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text(
                      product.name,
                      style: TextStyle(
                        fontSize: 24,
                        fontWeight: FontWeight.bold,
                        color: Colors.black87,
                      ),
                    ),
                    SizedBox(height: 10),
                    Text(
```

```

        product.description,
        style: TextStyle(
          fontSize: 16,
          color: Colors.black54,
        ),
      ),
      SizedBox(height: 20),
      Text(
        'Harga: Rp${product.price}',
        style: TextStyle(
          fontSize: 22,
          fontWeight: FontWeight.w600,
          color: Colors.black,
        ),
      ),
    ],
  ),
),
SizedBox(height: 20),
// Tombol Tambah ke Keranjang
Padding(
  padding: const EdgeInsets.symmetric(horizontal: 16.0),
  child: ElevatedButton(
    onPressed: () {
      // Logika menambahkan ke keranjang
    },
    style: ElevatedButton.styleFrom(
      padding: EdgeInsets.symmetric(vertical: 15),
      backgroundColor: Colors.blueAccent,
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(12),
      ),
    ),
    child: Center(
      child: Text(
        'Tambah ke Keranjang',
        style: TextStyle(
          fontSize: 18,
          fontWeight: FontWeight.bold,
          color: Colors.white,
        ),
      ),
    ),
  ),
),
),
),
),
),
),
),
),
),
),
);
}
}

```


Output



Deskripsi Program

Pada halaman utama (HomePage), aplikasi menampilkan daftar produk berupa gambar, nama, dan harga. Mengklik setiap item akan membuka halaman detail dengan informasi lebih lengkap tentang produk, termasuk gambar produk, deskripsi, dan harga. Pengguna juga dapat menambahkan produk ke keranjang mereka menggunakan tombol di halaman detail. CartPage bertindak sebagai halaman yang menampilkan produk yang ditambahkan ke keranjang. Halaman beranda menampilkan daftar produk dalam bentuk kartu terstruktur, masing-masing berisi gambar produk, nama, dan harga. ListView.builder digunakan untuk menampilkan daftar produk secara dinamis tergantung pada jumlah produk yang ada. Setiap produk ditampilkan sebagai gambar dengan sudut membulat, dan nama serta harga disorot dalam gaya teks tebal. InkWell digunakan pada setiap kartu untuk menciptakan efek riak saat pengguna mengetuk suatu produk, memberikan umpan balik visual yang membuat interaksi terasa lebih alami. Setelah mengklik suatu produk, pengguna diarahkan ke halaman detail dengan informasi lengkap tentang produk tersebut. Pada halaman detail, aplikasi menampilkan informasi produk yang lebih lengkap. Gambar produk di atas memiliki sudut bawah membulat dan ditempatkan melintang di lebar layar agar lebih terlihat. Di bawah gambar ditampilkan nama produk dengan huruf tebal besar, deskripsi produk ditampilkan dalam warna abu-abu lembut dalam ukuran kecil, dan harga produk ditampilkan dengan warna biru dalam ukuran besar. Tombol "Tambah ke Keranjang" yang besar dan mudah diakses di bagian bawah halaman, dengan banyak bantalan dan warna tematik, memudahkan pengguna untuk menambahkan item ke keranjang mereka. CartPage adalah halaman untuk menampilkan produk yang ditambahkan ke keranjang belanja pengguna. Halaman ini saat ini berfungsi sebagai pengganti dengan teks yang menunjukkan bahwa keranjang belanja masih kosong. CartPage dapat dikembangkan lebih lanjut untuk menampilkan daftar produk di keranjang, total harga, dan tombol untuk menyelesaikan pembelian atau mengeluarkan produk dari keranjang.