TUGAS PENDAHULUAN
PEMROGRAMAN PERANGKAT BERGERAK
MODUL VII
NAVIGASI DAN NOTIFIKASI

Disusun Oleh :

Nama: Arwin Nabiel Arrofif

NIM: 2211104057

Kelas: 06 02

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
NAVIGASI DAN NOTIFIKASI

1. Model

Pada umumnya, hampir seluruh aplikasi yang dibuat akan bekerja dengan data. Data dalam sebuah aplikasi memiliki sangat banyak bentuk, tergantung dari aplikasi yang dibuat. Setiap data yang diterima atau dikirimkan akan lebih baik apabila memiliki standar yang sama. Hampir mustahil melakukan peneliharaan *project* yang kompleks tanpa model.

• Membuat Model Class

Untuk membuat model, buatlah direktori baru pada folder lib project flutter, kemudian buat sebuah file class dart dengan nama filenya adalah nama data yang ingin dijadikan model. Sebagai contoh, ketika mambuat model user dengan response json di bawah ini:

```
{
  "user_id": 1,
  "id": 13,
  "title": "Bedroom Pop"
}
```

Maka buatlah file user.dart di dalam folder models, dengan code seperti di bawah ini:

```
class Album {
final int userId;
final int id;
final String title;


  const Album({
required this.userId,
required this.id,
```

```
    required this.title,
  });
  factory Album.fromJson(Map<String, dynamic> json) {
return Album(
      userId:
json['user_id'],        id:
json['id'],        title:
json['title'],
    );
  }
}
```

2. Navigation

Dalam Flutter, navigation merujuk pada cara berpindah antar halaman (atau tampilan) di aplikasi. Sistem navigasi di Flutter berbasis route dan navigator. Setiap halaman atau layar di Flutter disebut sebagai route, dan Navigator adalah widget yang mengelola stack dari route tersebut.

- **Navigation Pindah Halaman**

  Untuk melakukan navigasi ke halaman lain pada Flutter, dapat gunakan code seperti di bawah ini:

```
Navigator.push(
  Context, MaterialPageRoute(builder: (context) =>
SecondRoute()),
);
```

  Untuk melakukan navigasi kembali ke halaman sebelumnya, dapat gunakan code seperti di bawah ini:

```
Navigator.pop(context);
```

  Potongan code diatas harus diletakkan dalam function sebuah widget, misalnya pada *onPressed* milik ElevatedButton. SecondRoute pada contoh dapat diubah menjadi halaman bari yang dituju.

- **Navigation Mengirim Data**

Untuk dapat melakukan navigasi dengan mengirimkan data ke halaman lain, perlu disiapkan 2 hal, yakni:

1) Halaman baru memiliki parameter data yang diminta.
2) Halaman awal mengirimlan data melalui parameter.

Untuk dapat melakukan hal tersebut, kita dapat membuat sebuah halaman baru seperti di bawah ini:

```
class DetailScreen extends StatelessWidget {    const
DetailScreen({Key? key, required this.title}) :
super(key: key);

  final String title;

  @override
  Widget build(BuildContext context)
{      return Scaffold(
appBar: AppBar(          title:
Text(title),
      ),
    );
  }
}
```

Pada code di atas, telah dibuat halaman baru yang memiliki parameter data yang diinginkan yaitu atribut title dengan tipe data string.

```
Navigator.push(
context,
    MaterialPageRoute(builder: (context) => DetailScreen(title:
“Detail User”)),
);
```

Berbeda dengan contoh navigasi sebelumnya, pada navigasi di atas ditambahkan parameter title berisi string “Detail User” yang akan dikirimkan ke halaman baru.

3. Notification

Untuk mengirimkan notifikasi dalam aplikasi Flutter, dapat digunakan package bernama flutter_local_notifications. Tambahkan terlebih dahulu package tersebut ke dalam *pubspec.yaml*.

Konfigurasi Gradle

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

```
defaultConfig {     minSdkVersion 21
}
```

Konfigurasi iOS

Buka file *ios/Runner/Info.plist* dan tambahkan kode berikut untuk mendapatkan izin notifikasi pada iOS:

```
<key>UIBackgroundModes</key>
<array>
    <string>fetch</string>
    <string>remote-notification</string>
</array>
<key>NSAppTransportSecurity</key>
<dict>
    <key>NSAllowsArbitraryLoads</key>
    <true/>
</dict>
```

```
      <key>NSUserNotificationUsageDescription</key>
<string>We need your permission to send notifications.</string>
```

Setelah menambahkan package, ubah file *AndroidManifest.xml* dengan menambahkan barisan code seperti di bawah ini:

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

Tambahkan potongan kode di bawah ini di luar build dari stateful widget tersebut:

```
FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin
=FlutterLocalNotificationsPlugin();
```

Dengan membuat sebuah object Flutter LocalNotificationsPlugin, operasi yang terdapaat dalam package flutter_notification dapat digunakan.

Override method initState dari widget tersebut dengan menambahkan code seperti di bawah ini:

```
void initState() {
super.initState();
    var initializationSettingsAndroid =
    AndroidInitializationSettings('flutter_devs');
    var initializationSettingsIOs = IOSInitializationSettings();
var initSetttings = InitializationSettings(
        initializationSettingsAndroid, initializationSettingsIOs);
flutterLocalNotificationsPlugin.initialize(initSetttings,
onSelectNotification: onSelectNotification);
}
```

Kemudian buat sebuah function yang akan mengendalikan ketika notifikasi dipilih:

```
Future onSelectNotification(String payload) {
    Navigator.of(context).push(MaterialPageRoute(builder: (_) {
return NewScreen(              payload: payload,
        );
    }));
}
```

Buatlah sebuah widget baru yang akan menjadi halaman berikutnya setelah notifikasi dipilih:

```
class NewScreen extends StatelessWidget {
    String payload;
    NewScreen({
        @required this.payload,
    });
    @override
    Widget build(BuildContext context) {          return
Scaffold(              appBar: AppBar(              title:
Text(payload),
        ));
    }
}
```

Kemudian buat sebuah function yang fungsinya untuk menampilkan notifikasi sederhana untuk android:

```
showNotification() async {
    var android = new AndroidNotificationDetails(
'id', 'channel ', 'description',
        priority: Priority.High, importance:
Importance.Max);      var iOS = new
IOSNotificationDetails();      var platform = new
NotificationDetails(android, iOS);      await
flutterLocalNotificationsPlugin.show(
```

```
        0,  'Flutter  devs',  'Flutter  Local  Notification  Demo',
platform,
        payload: 'Welcome to the Local Notification demo ');
}
```

- AndroidNotificationDetails akan berisi mengenai detail notifikasi pada android.
- IOSNotificationDetails akan berisi mengenai detail notifikasi pada iOS.
- Kunci untuk menampilkan notifikasi terletak pada pemanggilan function flutterLocalNotificationsPlugin yang berfungsi untuk menampilkan notifikasi sesuai dengan platform yang digunakan.

Keseluruhan code di atas akan membentuk sebuah file seperti di bawah ini:

main.dart

```
import 'dart:async';
import 'package:flutter/material.dart';  import
'package:flutter_local_notifications/flutter_local_notifications.d
art';
void main() => runApp(new MaterialApp(
theme: ThemeData(
          appBarTheme: AppBarTheme(
color: Colors.amber,
      )),
      home: new MyApp(),
      debugShowCheckedModeBanner: false,
    ));
class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => _MyAppState();
}
class _MyAppState extends State<MyApp> {
```

```
  FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin
= FlutterLocalNotificationsPlugin();

 @override    void
initState() {
super.initState();
    var initializationSettingsAndroid =
        AndroidInitializationSettings('flutter_devs');      var
initializationSettingsIOs = IOSInitializationSettings();
var initSetttings = InitializationSettings(
        initializationSettingsAndroid, initializationSettingsIOs);

    flutterLocalNotificationsPlugin.initialize(initSetttings,
onSelectNotification: onSelectNotification);
  }

  Future onSelectNotification(String payload) {
    Navigator.of(context).push(MaterialPageRoute(builder: (_)
{        return NewScreen(          payload: payload,
      );
    }));
  }

  @override
  Widget build(BuildContext context)
{      return Scaffold(
appBar: AppBar(
        backgroundColor: Colors.amber,
        title: new Text('Flutter notification demo'),
      ),
      body: new Center(          child:
Column(            children: <Widget>[
ButtonTheme(                minWidth:
250.0,               child:
RaisedButton(                  color:
Colors.blueAccent,
```

```
                onPressed: showNotification,
child: new Text(
                  'showNotification',
                ),
              ),
            ),
          ],
        ),
      ),
    );
}
 showNotification() async {
    var android = new AndroidNotificationDetails(
        'id', 'channel ', 'description',
        priority: Priority.High, importance: Importance.Max);
var iOS = new IOSNotificationDetails();
    var platform = new NotificationDetails(android, iOS);
await flutterLocalNotificationsPlugin.show(
        0, 'Flutter devs', 'Flutter Local Notification Demo',
platform,
        payload: 'Welcome to the Local Notification demo ');
  }
class NewScreen extends StatelessWidget {
  String payload;

  NewScreen({
    @required this.payload,
  });

  @override
  Widget build(BuildContext context)
{      return Scaffold(
appBar: AppBar(          title:
Text(payload),
      ),
    );
  }
```

```
}
```

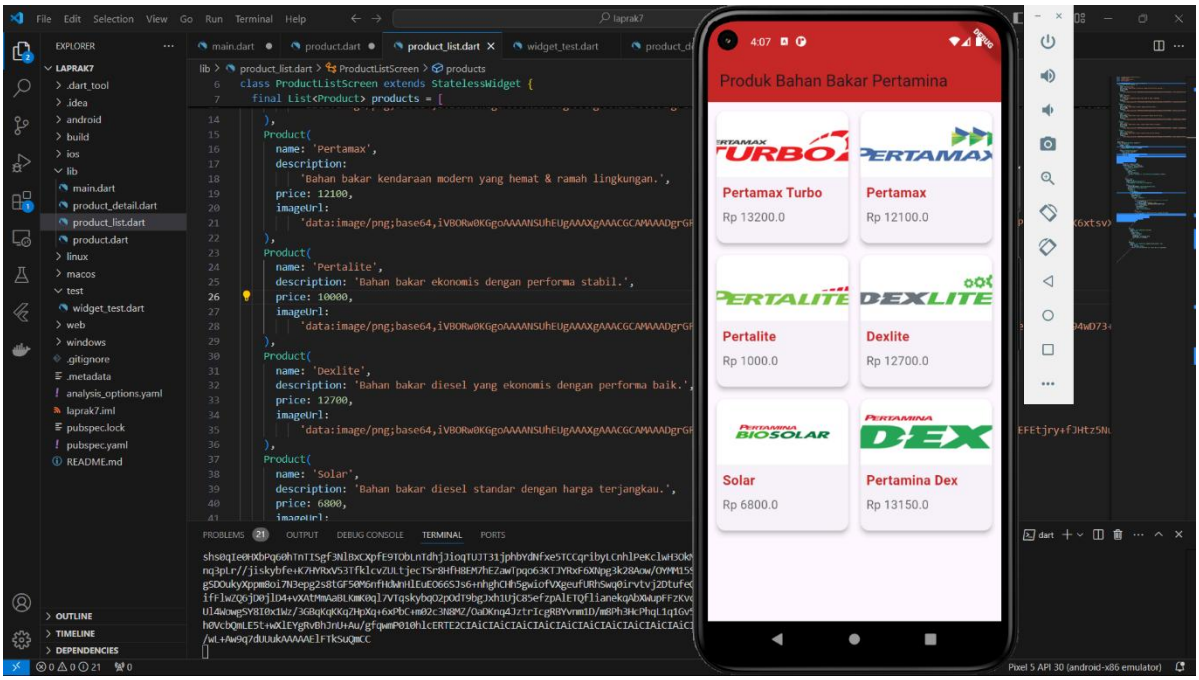
Di atas adalah keseluruhan kode yang digunakan untuk menampilkan sebuah notifikasi sederhana.

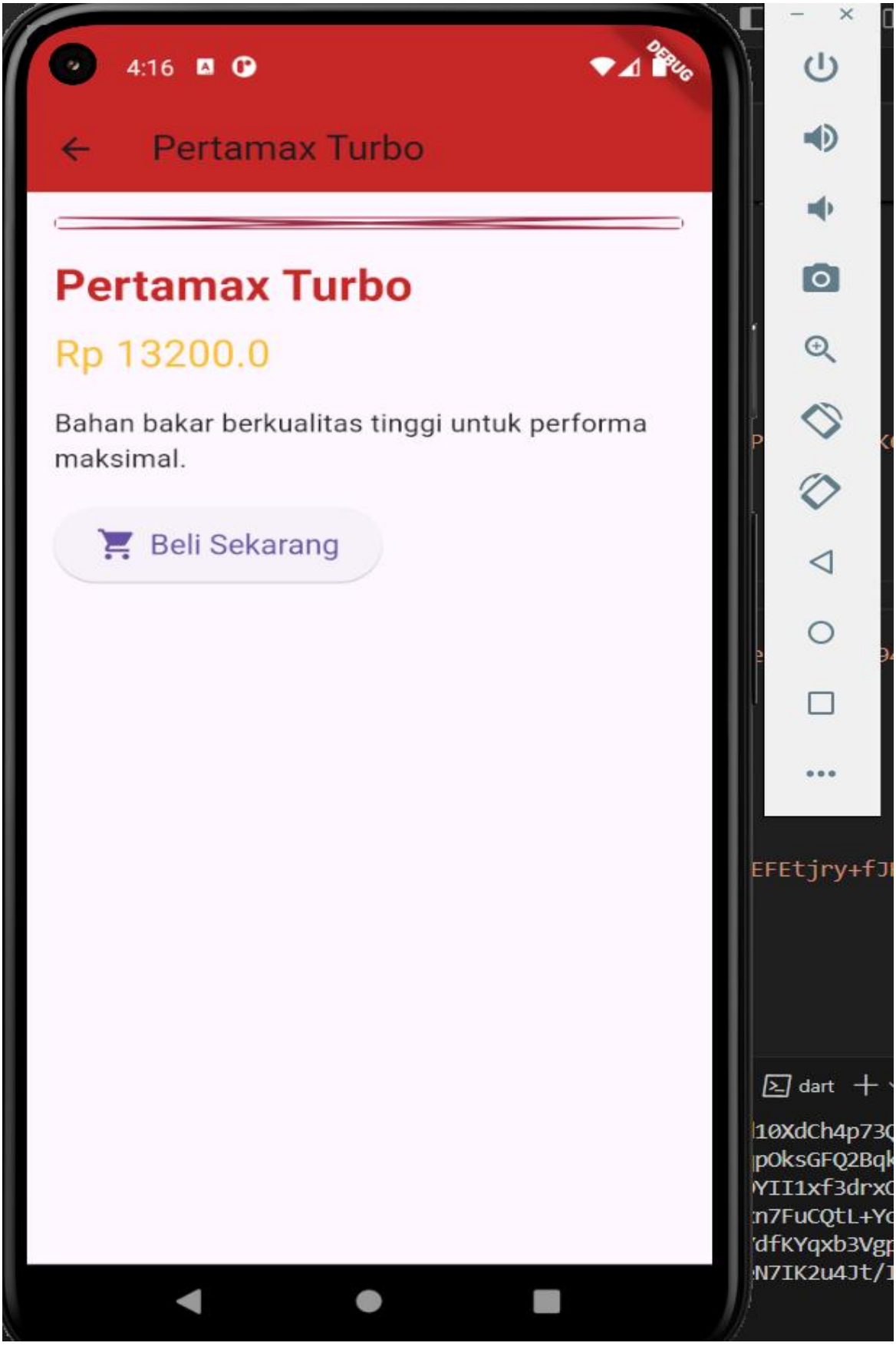
Unguided

Tugas Mandiri (Unguided)

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

Note: Jangan lupa sertakan source code, screenshoot output, dan deskripsi program. Kreatifitas menjadi nilai tambah.




4:16
DEBUG
Pertamax Turbo
Pertamax Turbo
Rp 13200.0
Bahan bakar berkualitas tinggi untuk performa maksimal.
Beli Sekarang
dart

deskripsi program.

Flutter sederhana bernama "Pertamina Fuel Store" yang bertujuan untuk menampilkan berbagai produk bahan bakar dari Pertamina. Aplikasi ini memiliki halaman daftar produk yang menampilkan berbagai jenis bahan bakar dengan harga, nama, dan gambar. Pengguna dapat memilih salah satu produk untuk melihat detailnya di halaman lain.

Fitur utama aplikasi:

1. **Halaman Daftar Produk**
   Menampilkan produk dalam tata letak grid dengan gambar, nama, dan harga.
2. **Navigasi ke Halaman Detail**
   Ketika produk diklik, pengguna akan diarahkan ke halaman detail produk untuk informasi lebih lanjut.
3. **Desain Responsif**
   Menggunakan GridView untuk menyesuaikan tampilan produk dalam bentuk grid yang rapi.