PEMROGRAMAN MOBILE
PENGANTAR BAHASA PEMROGRAMAN DART - BAGIAN 7 – TUGAS 2

OLEH:

Nama : Agung Rizky S

NIM : 2241720187

Kelas : TI – 3C

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

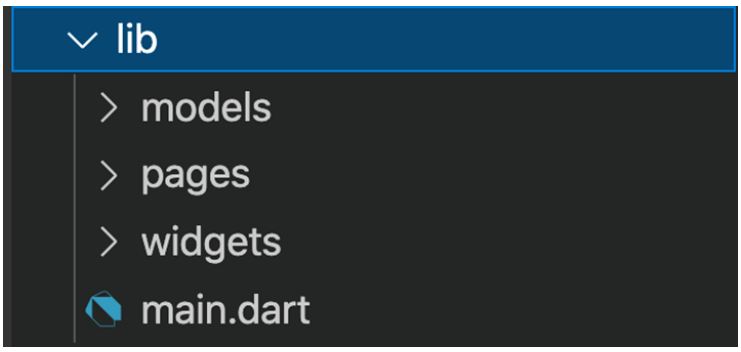
POLITEKNIK NEGERI MALANG

2024

Praktikum 1: Membangun Navigasi di Flutter

Langkah 1: Siapkan project baru

Sebelum melanjutkan praktikum, buatlah sebuah project baru Flutter dengan nama **belanja** dan susunan folder seperti pada gambar berikut. Penyusunan ini dimaksudkan untuk mengorganisasi kode dan widget yang lebih mudah.



Langkah 2: Mendefinisikan Route

Buatlah dua buah file dart dengan nama home_page.dart dan item_page.dart pada folder **pages**. Untuk masing-masing file, deklarasikan class HomePage pada file home_page.dart dan ItemPage pada item_page.dart. Turunkan class dari StatelessWidget. Gambaran potongan kode dapat anda lihat sebagai berikut.

```dart
class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    // TODO: implement build
    throw UnimplementedError();
  }
}
```

Langkah 3: Lengkapi Kode di **main.dart**

Setelah kedua halaman telah dibuat dan didefinisikan, bukalah file main.dart. Pada langkah ini anda akan mendefinisikan **Route** untuk kedua halaman tersebut. Definisi penamaan **route** harus bersifat **unique**. Halaman **HomePage** didefinisikan sebagai **/**. Dan halaman **ItemPage** didefinisikan sebagai **/item**. Untuk mendefinisikan halaman awal, anda dapat menggunakan named argument initialRoute. Gambaran tahapan ini, dapat anda lihat pada potongan kode berikut.

```
void main() {
  runApp(MaterialApp(
    initialRoute: '/',
    routes: {
      '/': (context) ⇒ HomePage(),
      '/item': (context) ⇒ ItemPage(),
    },
  )); // MaterialApp
}
```

Langkah 4: Membuat data model

Sebelum melakukan perpindahan halaman dari HomePage ke ItemPage, dibutuhkan proses pemodelan data. Pada desain mockup, dibutuhkan dua informasi yaitu nama dan harga. Untuk menangani hal ini, buatlah sebuah file dengan nama item.dart dan letakkan pada folder **models**. Pada file ini didefinisikan pemodelan data yang dibutuhkan. Ilustrasi kode yang dibutuhkan, dapat anda lihat pada potongan kode berikut.

```
class Item {
  String name;
  int price;

  Item({this.name, this.price});
}
```

Langkah 5: Lengkapi kode di class **HomePage**

Pada halaman HomePage terdapat ListView widget. Sumber data ListView diambil dari model List dari object Item. Gambaran kode yang dibutuhkan untuk melakukan definisi model dapat anda lihat sebagai berikut.

```dart
class HomePage extends StatelessWidget {
  final List<Item> items = [
    Item(name: 'Sugar', price: 5000),
    Item(name: 'Salt', price: 2000)
  ];
```

Langkah 6: Membuat ListView dan itemBuilder

Untuk menampilkan ListView pada praktikum ini digunakan itemBuilder. Data diambil dari definisi model yang telah dibuat sebelumnya. Untuk menunjukkan batas data satu dan berikutnya digunakan widget Card. Kode yang telah umum pada bagian ini tidak ditampilkan. Gambaran kode yang dibutuhkan dapat anda lihat sebagai berikut.

```dart
      body: Container(
        margin: EdgeInsets.all(8),
        child: ListView.builder(
          padding: EdgeInsets.all(8),
          itemCount: items.length,
          itemBuilder: (context, index) {
            final item = items[index];
            return Card(
              child: Container(
                margin: EdgeInsets.all(8),
                child: Row(
                  children: [
                    Expanded(child: Text(item.name)),
                    Expanded(
                      child: Text(
                        item.price.toString(),
                        textAlign: TextAlign.end,
                      ), // Text
                    ) // Expanded
                  ],
                ), // Row
              ), // Container
            ); // Card
          },
        ), // ListView.builder
      ), // Container
```

Jalankan aplikasi pada emulator atau pada device anda.

Perhatian: Pastikan pada halaman awal telah berhasil menampilkan ListView. Jika ada kesalahan, segera perbaiki sebelum melanjutkan ke langkah berikutnya.

Langkah 7: Menambahkan aksi pada **ListView**

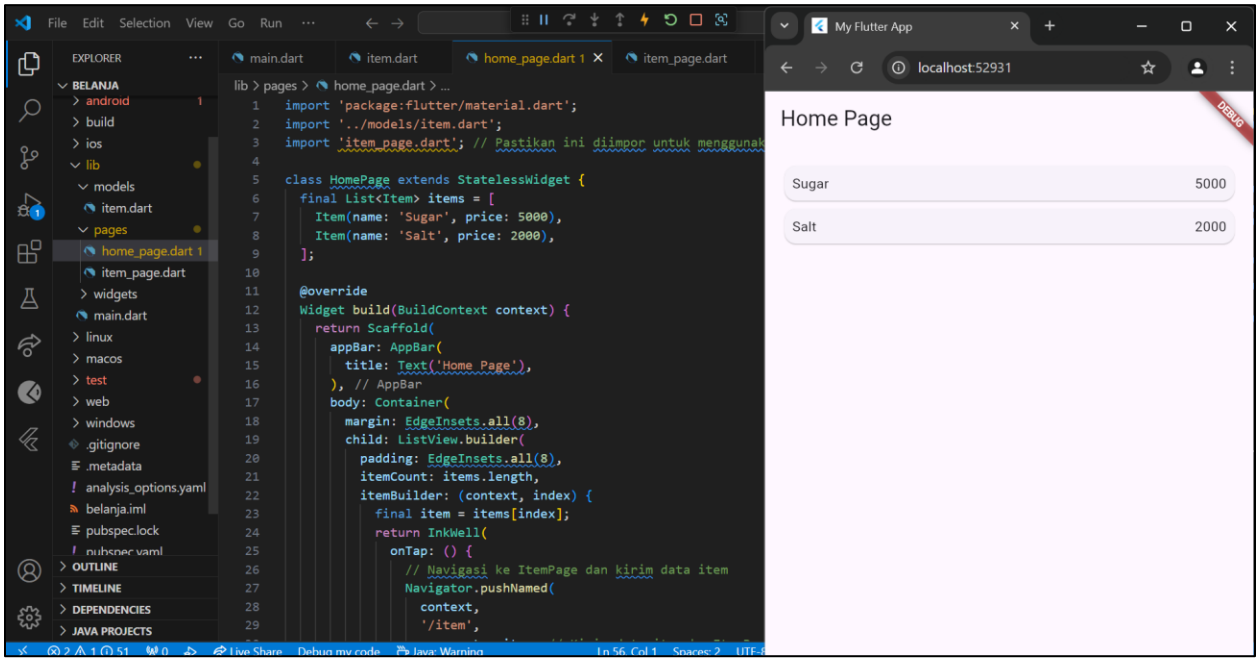
Item pada ListView saat ini ketika ditekan masih belum memberikan aksi tertentu. Untuk menambahkan aksi pada ListView dapat digunakan widget InkWell atau GestureDetector. Perbedaan utamanya InkWell merupakan material widget yang memberikan efek ketika ditekan. Sedangkan GestureDetector bersifat umum dan bisa juga digunakan untuk gesture lain selain sentuhan. Pada praktikum ini akan digunakan widget InkWell.

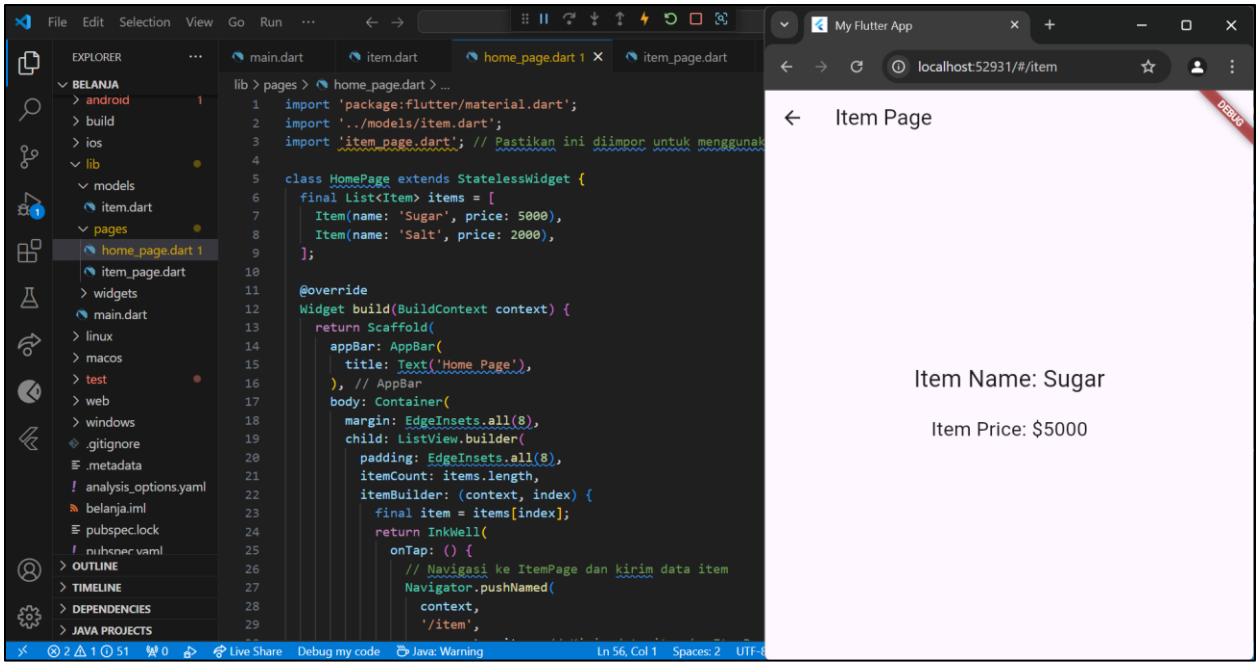
Untuk menambahkan sentuhan, letakkan cursor pada widget pembuka Card. Kemudian gunakan shortcut quick fix dari VSCode (**Ctrl + .** pada Windows atau **Cmd + .** pada MacOS). Sorot menu wrap with widget... Ubah nilai widget menjadi InkWell serta tambahkan named argument onTap yang berisi fungsi untuk berpindah ke halaman ItemPage. Ilustrasi potongan kode dapat anda lihat pada potongan berikut.

```
return InkWell(
  onTap: () {
    Navigator.pushNamed(context, '/item');
  },
```

Jalankan aplikasi kembali dan pastikan ListView dapat disentuh dan berpindah ke halaman berikutnya. Periksa kembali jika terdapat kesalahan.

Jawaban :



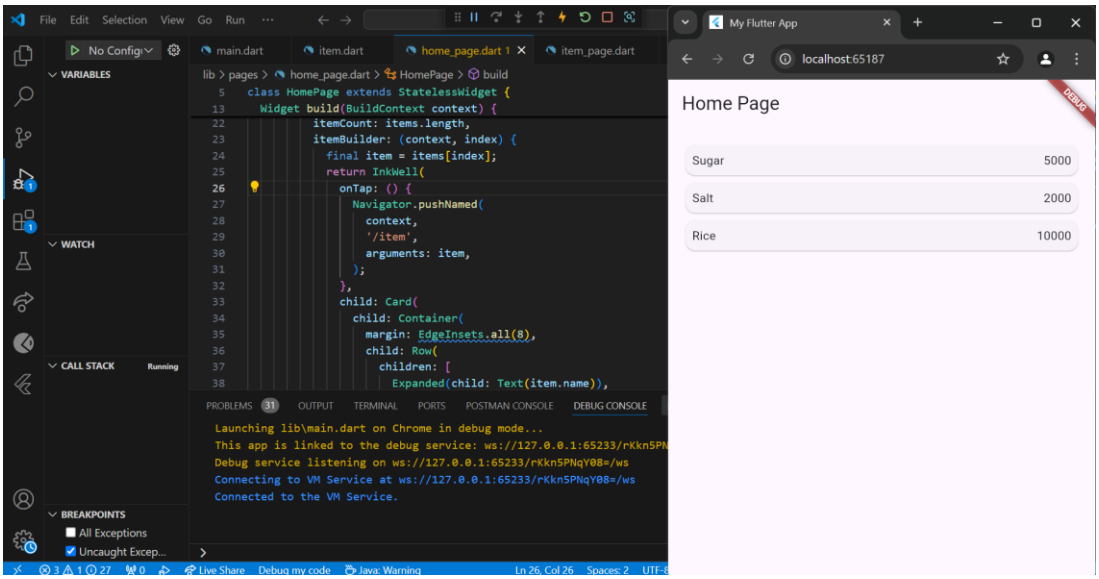

File Edit Selection View Go Run
EXPLORER
BELANJA
android
build
ios
lib
models
item.dart
pages
home_page.dart 1
item_page.dart
widgets
main.dart
linux
macos
test
web
windows
.gitignore
.metadata
analysis_options.yaml
belanja.iml
pubspec.lock
pubspec.yaml
OUTLINE
TIMELINE
DEPENDENCIES
JAVA PROJECTS
main.dart
item.dart
home_page.dart 1
item_page.dart
lib > pages > home_page.dart > ...
import 'package:flutter/material.dart';
import '../models/item.dart';
import 'item_page.dart'; // Pastikan ini diimpor untuk menggunak
class HomePage extends StatelessWidget {
final List<Item> items = [
Item(name: 'Sugar', price: 5000),
Item(name: 'Salt', price: 2000),
];
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text('Home Page'),
), // AppBar
body: Container(
margin: EdgeInsets.all(8),
child: ListView.builder(
padding: EdgeInsets.all(8),
itemCount: items.length,
itemBuilder: (context, index) {
final item = items[index];
return InkWell(
onTap: () {
// Navigasi ke ItemPage dan kirim data item
Navigator.pushNamed(
context,
'/item',
Live Share
Debug my code
Java: Warning
Ln 56, Col 1
Spaces: 2
My Flutter App
localhost:52931/#/item
Item Page
DEBUG
Item Name: Sugar
Item Price: $5000

Tugas 2: Tugas Praktikum 2

1. Untuk melakukan pengiriman data ke halaman berikutnya, cukup menambahkan informasi arguments pada penggunaan Navigator. Perbarui kode pada bagian Navigator menjadi seperti berikut.

```
Navigator.pushNamed(context, '/item', arguments: item);
```

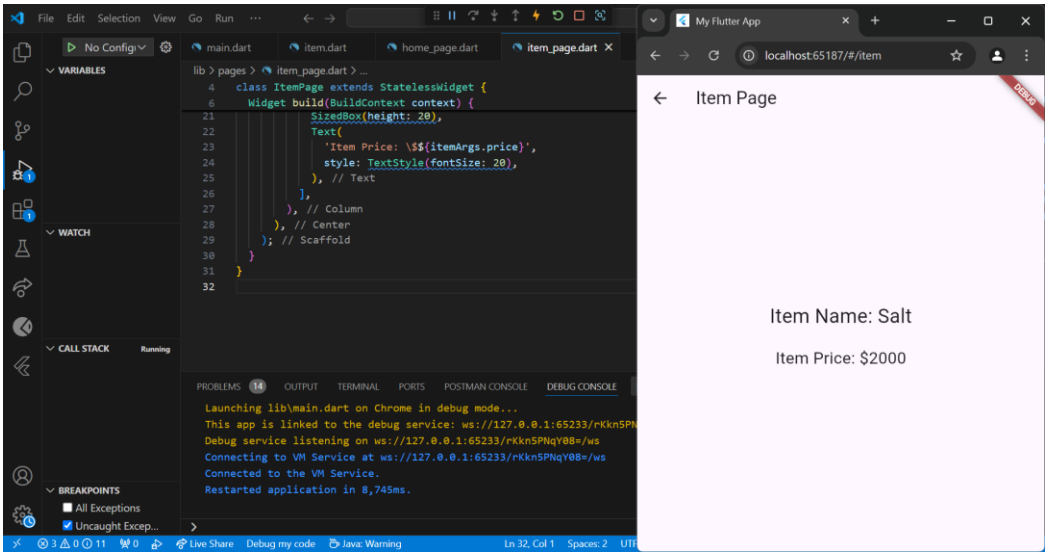
Jawaban :



2. Pembacaan nilai yang dikirimkan pada halaman sebelumnya dapat dilakukan menggunakan ModalRoute. Tambahkan kode berikut pada blok fungsi build dalam halaman ItemPage. Setelah nilai didapatkan, anda dapat menggunakannya seperti penggunaan variabel pada umumnya. (https://docs.flutter.dev/cookbook/navigation/navigate-with-arguments)

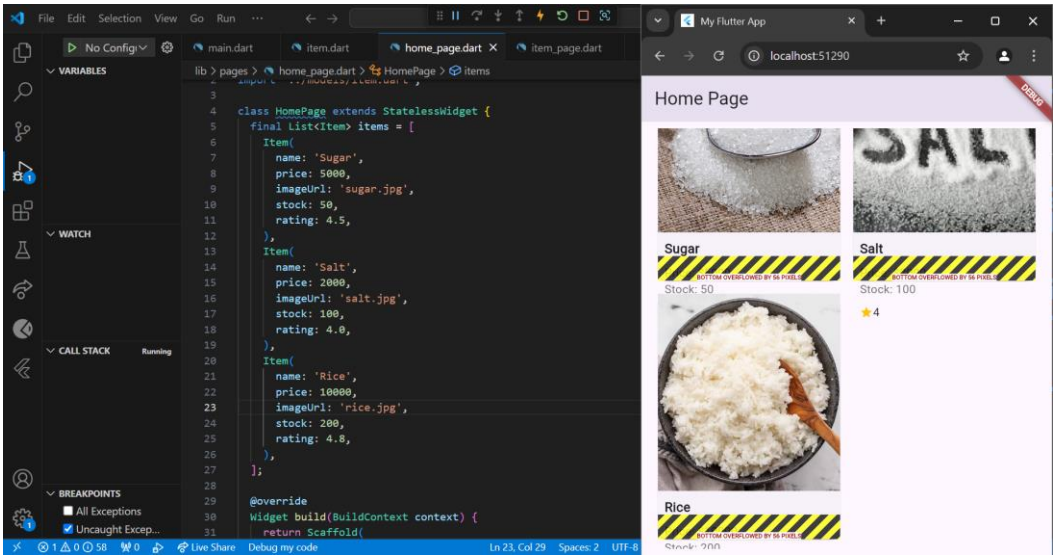
```
final itemArgs = ModalRoute.of(context)!.settings.arguments as Item;
```

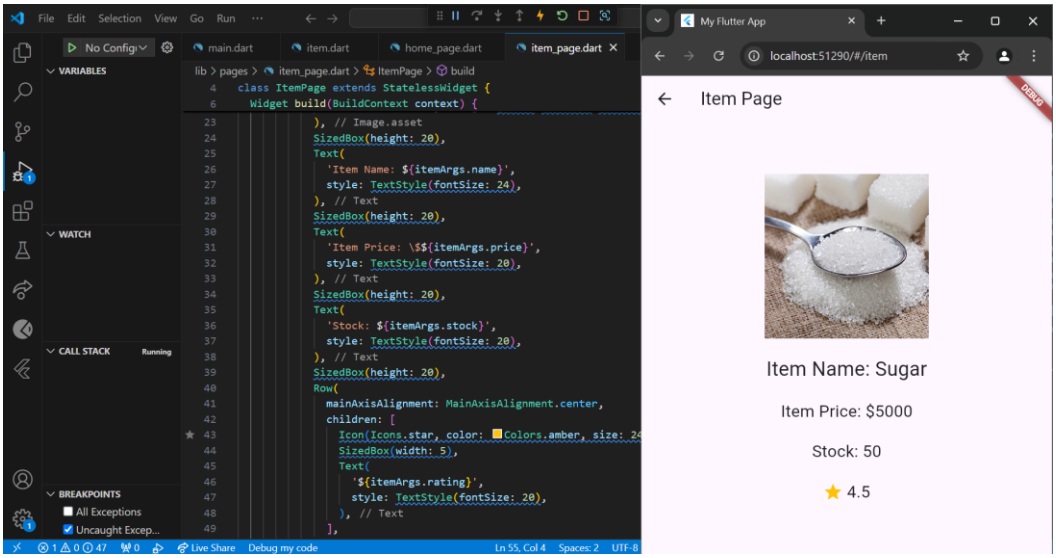
jawaban :



3. Pada hasil akhir dari aplikasi **belanja** yang telah anda selesaikan, tambahkan atribut foto produk, stok, dan rating. Ubahlah tampilan menjadi GridView seperti di aplikasi marketplace pada umumnya.

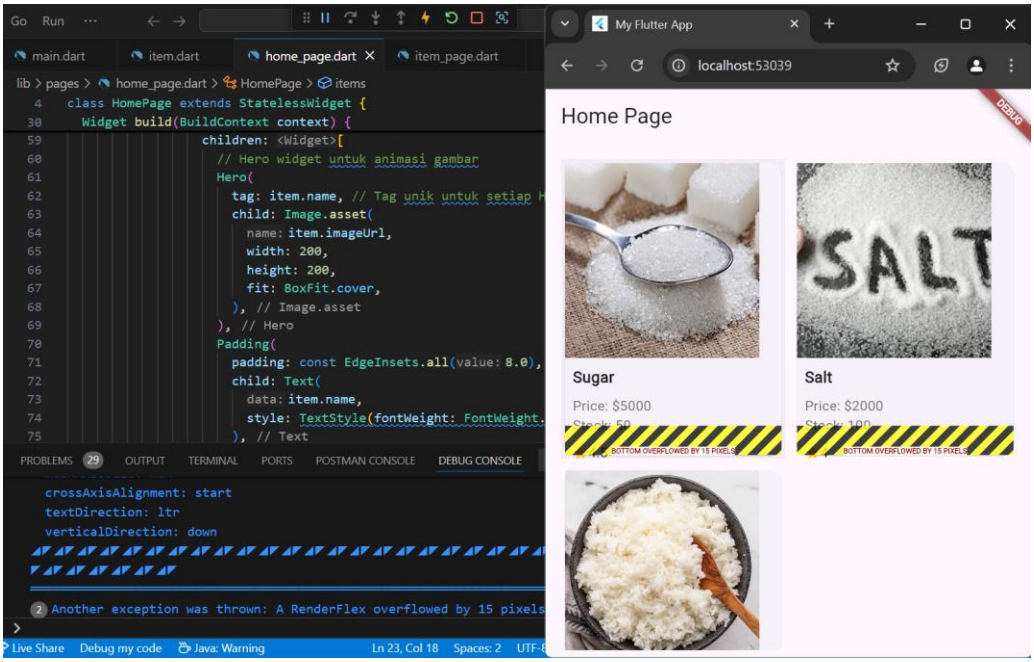
   Jawaban :





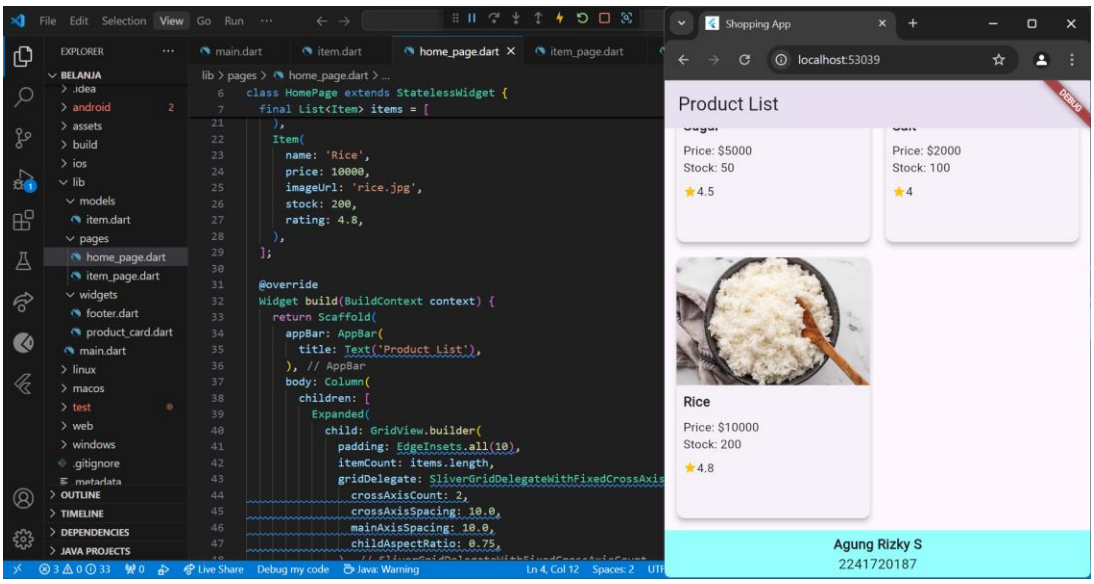
4. Silakan implementasikan Hero widget pada aplikasi **belanja** Anda dengan mempelajari dari sumber ini: https://docs.flutter.dev/cookbook/navigation/hero-animations

   Jawaban :



5. Sesuaikan dan modifikasi tampilan sehingga menjadi aplikasi yang menarik. Selain itu, pecah widget menjadi kode yang lebih kecil. Tambahkan **Nama** dan **NIM** di footer aplikasi **belanja** Anda.

Jawaban :



6. Selesaikan Praktikum 5: Navigasi dan Rute tersebut. Cobalah modifikasi menggunakan plugin go_router, lalu dokumentasikan dan push ke repository Anda berupa screenshot setiap hasil pekerjaan beserta penjelasannya di file README.md. Kumpulkan link commit repository GitHub Anda kepada dosen yang telah disepakati!