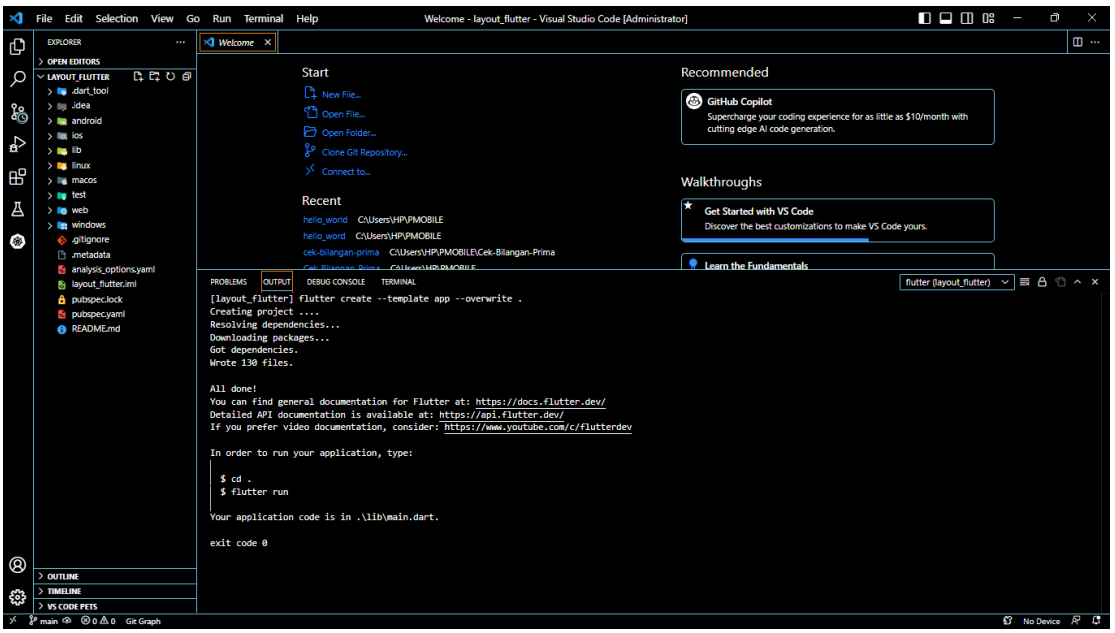
KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
https://jti.polinema.ac.id

Mata Kuliah	:	Pemrograman Mobile
Program Studi	:	D4 – Sistem Informasi Bisnis
Semester	:	5

Kelas	:	SIB-3D
NIM	:	2341760178
Nama	:	Saria Fauzani
Jobsheet Ke-	:	5
Link Github	:	https://github.com/Sariafauzani/layout-flutter

Laporan Jobsheet

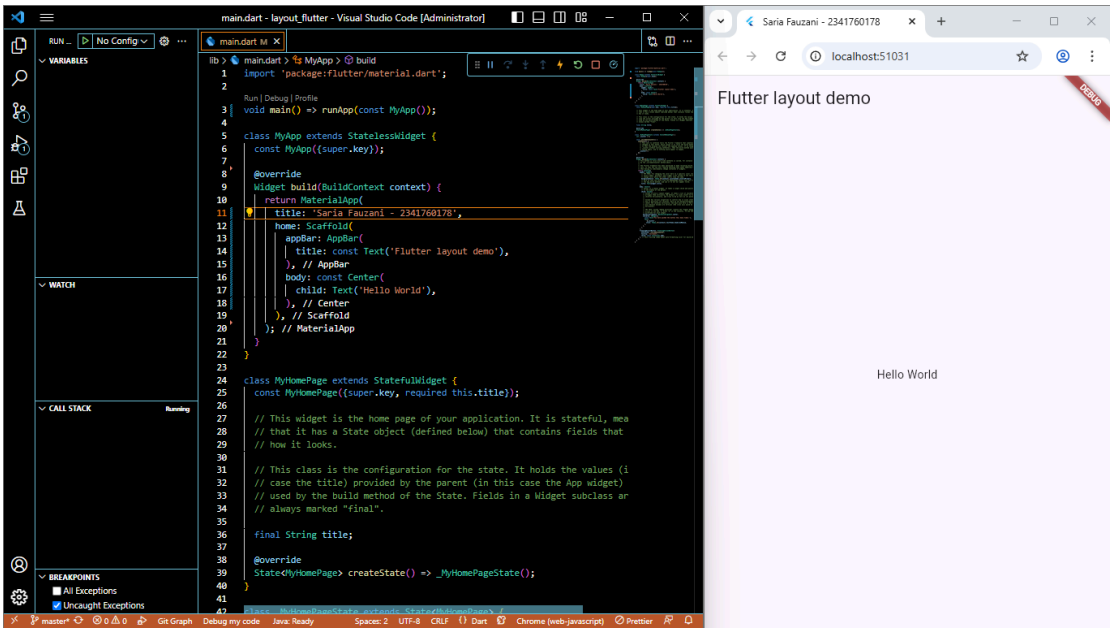
Praktikum Ke-1 - Membangun Layout di Flutter

Langkah	Jawaban/Deskripsi
	Langkah 1: Buat Project Baru
	Buatlah sebuah project flutter baru dengan nama layout_flutter. Atau sesuaikan style laporan praktikum yang Anda buat. 
	Langkah 2: Buka file lib/main.dart

Buka file main.dart lalu ganti dengan kode berikut. Isi nama dan NIM Anda di text title.

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

Run | Debug | Profile
void main() => runApp(const MyApp());

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Saria Fauzani - 2341760178',
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Flutter layout demo'),
        ), // AppBar
        body: const Center(
          child: Text('Hello World'),
        ), // Center
      ), // Scaffold
    ); // MaterialApp
  }
}
```

Output



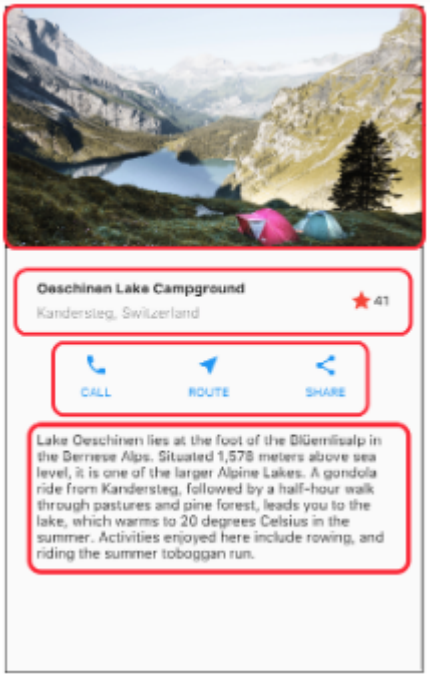
Langkah 3: Identifikasi layout diagram

Langkah pertama adalah memecah tata letak menjadi elemen dasarnya:

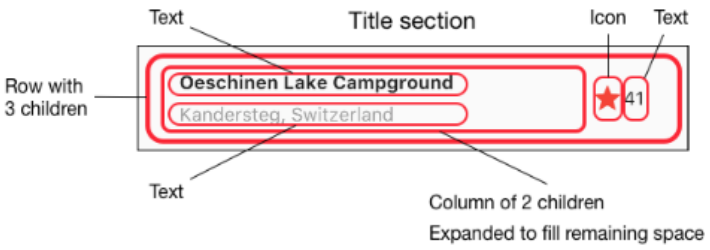
- Identifikasi baris dan kolom.
- Apakah tata letaknya menyertakan kisi-kisi (grid)?
- Apakah ada elemen yang tumpang tindih?

- Apakah UI memerlukan tab?
- Perhatikan area yang memerlukan alignment, padding, atau borders.

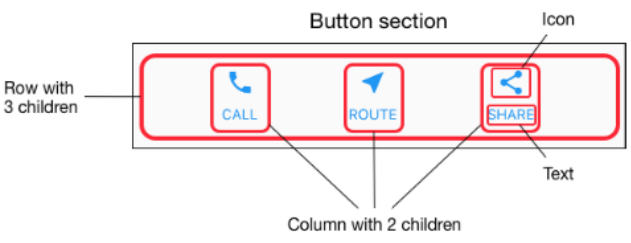
Pertama, identifikasi elemen yang lebih besar. Dalam contoh ini, empat elemen disusun menjadi sebuah kolom: sebuah gambar, dua baris, dan satu blok teks.



Selanjutnya, buat diagram setiap baris. Baris pertama, disebut bagian Judul, memiliki 3 anak: kolom teks, ikon bintang, dan angka. Anak pertamanya, kolom, berisi 2 baris teks. Kolom pertama itu memakan banyak ruang, sehingga harus dibungkus dengan widget yang Diperluas.



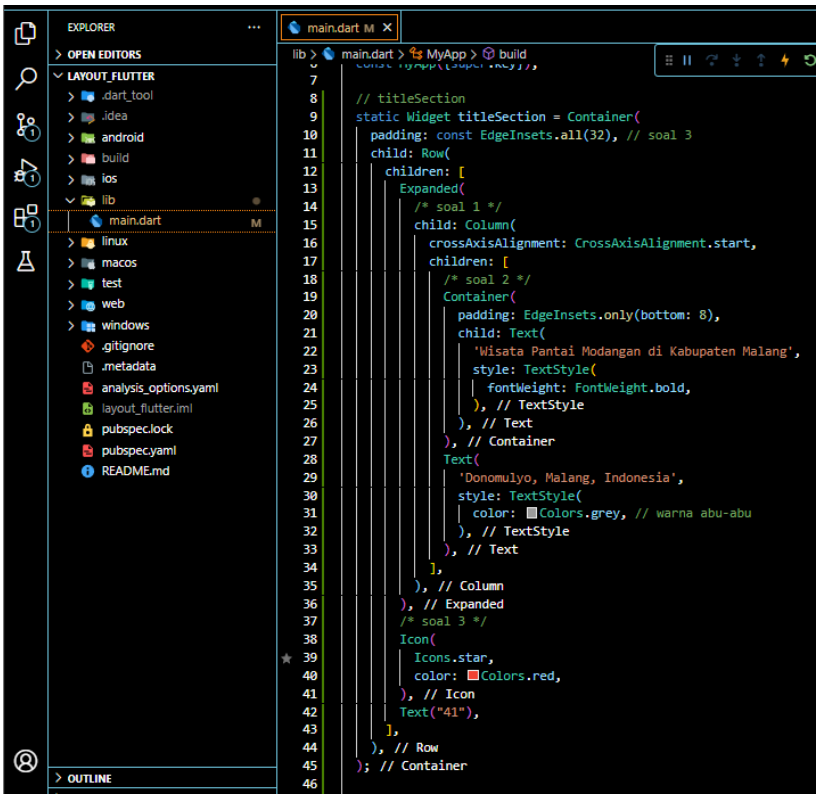
Baris kedua, disebut bagian Tombol, juga memiliki 3 anak: setiap anak merupakan kolom yang berisi ikon dan teks.



Setelah tata letak telah dibuat diagramnya, cara termudah adalah dengan menerapkan pendekatan bottom-up. Untuk meminimalkan kebingungan visual dari kode tata letak yang banyak bertumpuk, tempatkan beberapa implementasi dalam variabel dan fungsi.

Langkah 4: Implementasi title row

Pertama, Anda akan membuat kolom bagian kiri pada judul. Tambahkan kode berikut di bagian atas metode build() di dalam kelas MyApp:



/* soal 1 */ Letakkan widget Column di dalam widget Expanded agar menyesuaikan ruang yang tersisa di dalam widget Row. Tambahkan properti crossAxisAlignment ke CrossAxisAlignment.start sehingga posisi kolom berada di awal baris.

/* soal 2 */ Letakkan baris pertama teks di dalam Container sehingga memungkinkan Anda untuk menambahkan padding = 8. Teks ‘Batu, Malang, Indonesia' di dalam Column, set warna menjadi abu-abu.

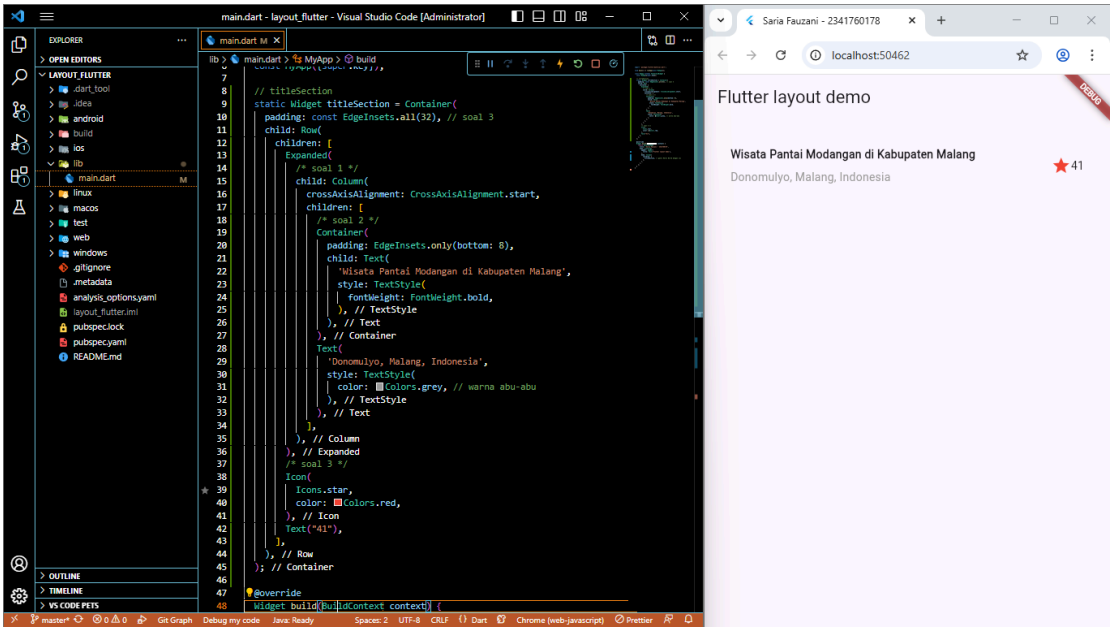
/* soal 3 */ Dua item terakhir di baris judul adalah ikon bintang, set dengan warna merah, dan teks "41". Seluruh baris ada di dalam Container dan beri padding di sepanjang setiap tepinya sebesar 32 piksel. Kemudian ganti isi body text ‘Hello World' dengan variabel titleSection seperti berikut:

<table>
<tr><td></td><td>

```
                Icon(
                  Icons.star,
                  color: Colors.red,
                ), // Icon
                Text("41"),
              ],
```

</td></tr>
<tr><td></td><td>

Output



</td></tr>
</table>

Praktikum Ke-2 - Implementasi button row

Langkah	Jawaban/Deskripsi
	Langkah 1: Buat method Column _buildButtonColumn
	Bagian tombol berisi 3 kolom yang menggunakan tata letak yang sama—sebuah ikon di atas baris teks. Kolom pada baris ini diberi jarak yang sama, dan teks serta ikon diberi warna primer. Karena kode untuk membangun setiap kolom hampir sama, buatlah metode pembantu pribadi bernama buildButtonColumn(), yang mempunyai parameter warna, Icon dan Text, sehingga dapat mengembalikan kolom dengan widgetnya sesuai dengan warna tertentu. **lib/main.dart (_buildButtonColumn)**

```dart
Column _buildButtonColumn(Color color, IconData icon, String label) {
  return Column(
    mainAxisSize: MainAxisSize.min,
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Icon(icon, color: color),
      Container(
        margin: const EdgeInsets.only(top: 8),
        child: Text(
          label,
          style: TextStyle(
            fontSize: 12,
            fontWeight: FontWeight.w400,
            color: color,
          ), // TextStyle
        ), // Text
      ), // Container
    ],
  ); // Column
}
```

Langkah 2: Buat widget buttonSection

Buat Fungsi untuk menambahkan ikon langsung ke kolom. Teks berada di dalam Container dengan margin hanya di bagian atas, yang memisahkan teks dari ikon.

Bangun baris yang berisi kolom-kolom ini dengan memanggil fungsi dan set warna, Icon, dan teks khusus melalui parameter ke kolom tersebut. Sejajarkan kolom di sepanjang sumbu utama menggunakan MainAxisAlignment.spaceEvenly untuk mengatur ruang kosong secara merata sebelum, di antara, dan setelah setiap kolom. Tambahkan kode berikut tepat di bawah deklarasi titleSection di dalam metode build():

lib/main.dart (buttonSection)

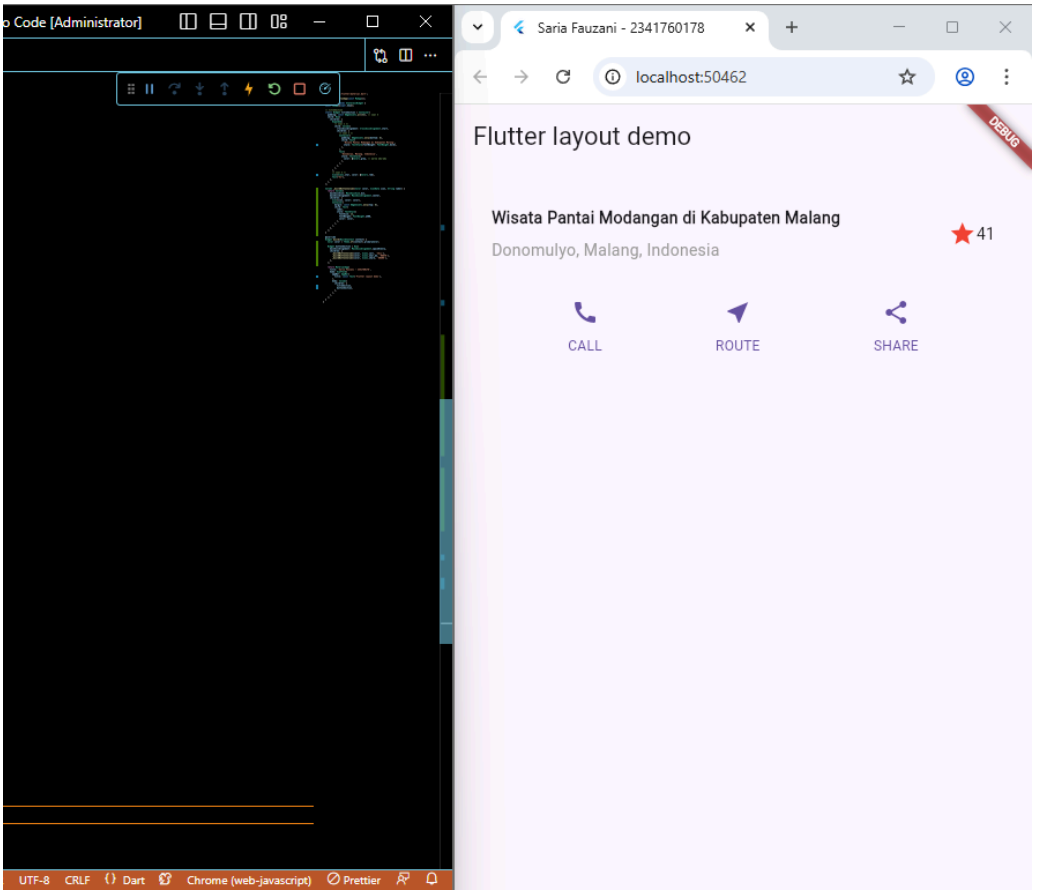
```dart
@override
Widget build(BuildContext context) {
  Color color = Theme.of(context).primaryColor;

  Widget buttonSection = Row(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: [
      _buildButtonColumn(color, Icons.call, 'CALL'),
      _buildButtonColumn(color, Icons.near_me, 'ROUTE'),
      _buildButtonColumn(color, Icons.share, 'SHARE'),
    ],
  );
```

Langkah 3: Tambah button section ke body

Tambahkan variabel buttonSection ke dalam body seperti berikut:

```
76      return MaterialApp(
77        title: 'Saria Fauzani - 2341760178',
78        home: Scaffold(
79          appBar: AppBar(
80            title: const Text('Flutter layout demo'),
81          ), // AppBar
82          body: Column(
83            children: [
84              titleSection,
85              buttonSection,
86            ],
87          ), // Column
88        ), // Scaffold
89      ); // MaterialApp
90    }
91  }
```

Output



Praktikum Ke-3 - Implementasi text section

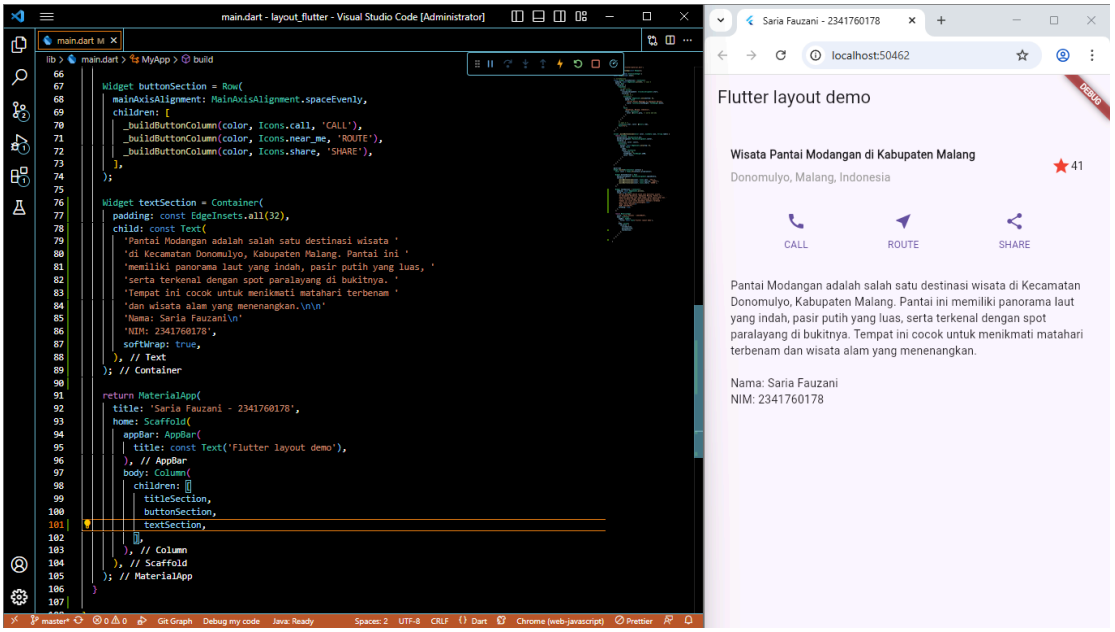
Langkah	Jawaban/Deskripsi
	Langkah 1: Buat widget textSection

	Tentukan bagian teks sebagai variabel. Masukkan teks ke dalam Container dan tambahkan padding di sepanjang setiap tepinya. Tambahkan kode berikut tepat di bawah deklarasi buttonSection:
	`Widget textSection = Container(`
	`  padding: const EdgeInsets.all(32),`
	`  child: const Text(`
	`  'Pantai Modangan adalah salah satu destinasi wisata '`
	`  'di Kecamatan Donomulyo, Kabupaten Malang. Pantai ini '`
	`  'memiliki panorama laut yang indah, pasir putih yang luas, '`
	`  'serta terkenal dengan spot paralayang di bukitnya. '`
	`  'Tempat ini cocok untuk menikmati matahari terbenam '`
	`  'dan wisata alam yang menenangkan.\n\n'`
	`  'Nama: Saria Fauzani\n'`
	`  'NIM: 2341760178',`
	`  softWrap: true,`
	`  ), // Text`
	`); // Container`
	Dengan memberi nilai softWrap = true, baris teks akan memenuhi lebar kolom sebelum membungkusnya pada batas kata.

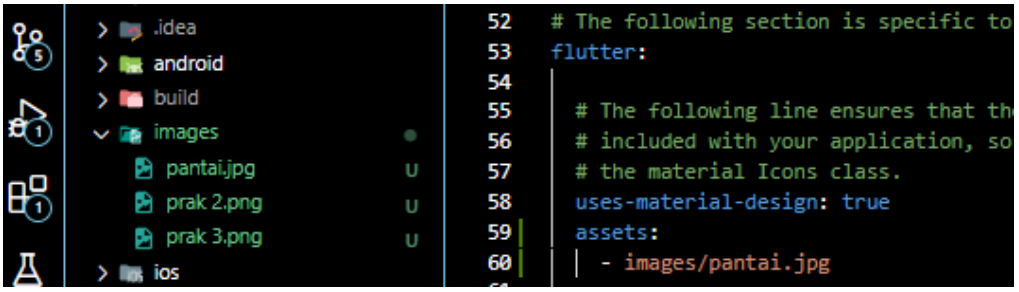
	Langkah 2: Tambahkan variabel text section ke body

	Tambahkan widget variabel textSection ke dalam body seperti berikut:
	`return MaterialApp(`
	`  title: 'Saria Fauzani - 2341760178',`
	`  home: Scaffold(`
	`  appBar: AppBar(`
	`  title: const Text('Flutter layout demo'),`
	`  ), // AppBar`
	`  body: Column(`
	`  children: [`
	`  titleSection,`
	`  buttonSection,`
	`  textSection,`
	`  ],`
	`  ), // Column`
	`  ), // Scaffold`
	`  ); // MaterialApp`

	Output



Praktikum Ke-4 - Implementasi image section

Langkah	Jawaban/Deskripsi
	Langkah 1: Siapkan aset gambar
	Anda dapat mencari gambar di internet yang ingin ditampilkan. Buatlah folder images di root project layout_flutter. Masukkan file gambar tersebut ke folder images, lalu set nama file tersebut ke file pubspec.yaml seperti berikut: 
	Langkah 2: Tambahkan gambar ke body
	Tambahkan aset gambar ke dalam body seperti berikut:

```
children: [
  Image.asset(
    'images/pantai.jpg',
    width: 600,
    height: 240,
    fit: BoxFit.cover,
  ), // Image.asset
```

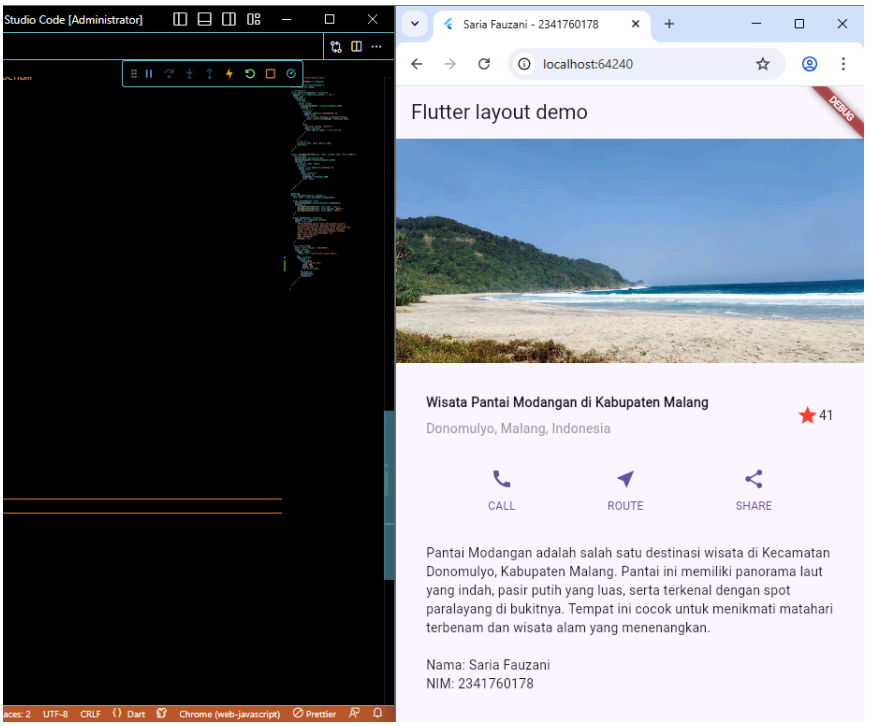
BoxFit.cover memberitahu kerangka kerja bahwa gambar harus sekecil mungkin tetapi menutupi seluruh kotak rendernya.

Langkah 3: Terakhir, ubah menjadi ListView

Pada langkah terakhir ini, atur semua elemen dalam ListView, bukan Column, karena ListView mendukung scroll yang dinamis saat aplikasi dijalankan pada perangkat yang resolusinya lebih kecil.

```
), // AppBar
body: ListView(
  children: [
    Image.asset(
      'images/pantai.jpg',
      width: 600,
      height: 240,
      fit: BoxFit.cover,
    ), // Image.asset
```

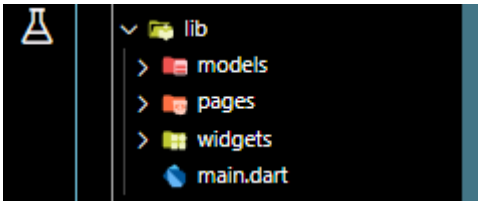
Output



Tugas Praktikum 1

Langkah	Jawaban/Deskripsi
1	Selesaikan Praktikum 1 sampai 4, lalu dokumentasikan dan push ke repository Anda berupa screenshot setiap hasil pekerjaan beserta penjelasannya di file README.md! https://github.com/Sariafauzani/layout-flutter/commit/aa70c850aba17480a6439e7eac1a72a4ebd18bf8
2	Silahkan implementasikan di project baru "**basic_layout_flutter**" dengan mengakses sumber ini: https://docs.flutter.dev/codelabs/layout-basics
3	Kumpulkan link commit repository GitHub Anda kepada dosen yang telah disepakati!

Praktikum 5: Membangun Navigasi di Flutter

Langkah	Jawaban/Deskripsi
	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. **item_page.dart**

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

class ItemPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {

    throw UnimplementedError();
  }
}
```

home_page.dart

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

class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {

    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.

```dart
Run | Debug | Profile
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      initialRoute: '/',
      routes: {
        '/': (context) => HomePage(),
        '/item': (content) => 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.

belanja > lib > models > item.dart > ...

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

  Item({required this.name, required 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.

belanja > lib > pages > home_page.dart > ...

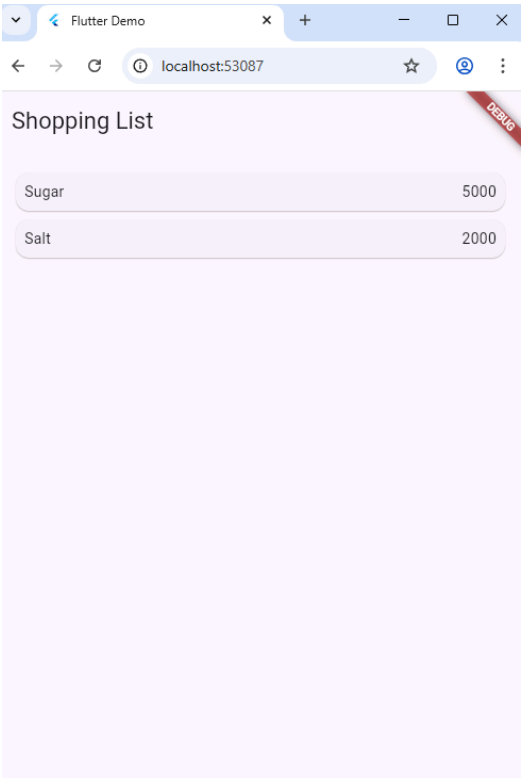
```dart
import 'package:flutter/material.dart';
import '../models/item.dart';

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

  final List<Item> items = [
    Item(name: 'Sugar', price: 5000),
    Item(name: 'Salt', price: 2000)
  ];

  @override
  Widget build(BuildContext context) {

    throw UnimplementedError();
  }
}
```

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.

```
    return Scaffold(
      appBar: AppBar(title: const Text("Shopping List")),
      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
    ); // Scaffold
  }
}
```

Jalankan aplikasi pada emulator atau pada device anda.



Langkah 7: Menambahkan aksi pada ListView

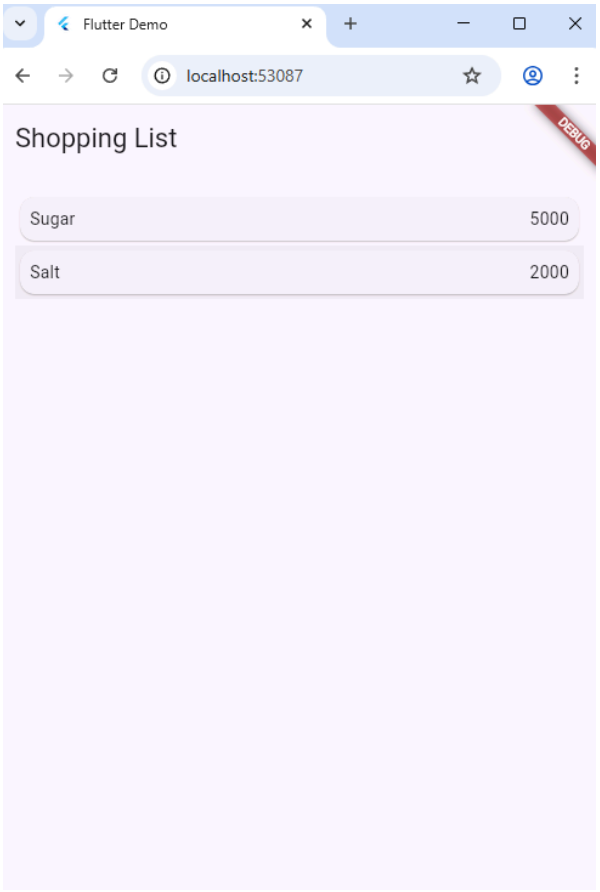
Item pada ListView saat ini ketika ditekan masih belum memberikan aksi tertentu. Untuk menambahkan aksi pada ListView dapat digunakan widget InkWell atau GestureDetector. Perbedaan utamanya InkWell merupakan material widget yang memberikan efek ketika ditekan. Sedangkan GestureDetector bersifat umum 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.

Output

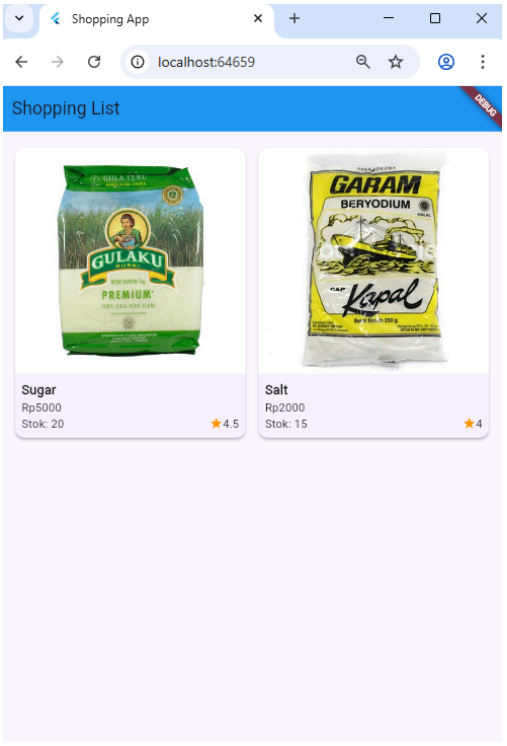
Tugas Praktikum 2

Langkah	Jawaban/Deskripsi
1	Untuk melakukan pengiriman data ke halaman berikutnya, cukup menambahkan informasi arguments pada penggunaan Navigator. Perbarui kode pada bagian Navigator menjadi seperti berikut. 
2	Pembacaan nilai yang dikirimkan pada halaman sebelumnya dapat dilakukan menggunakan ModalRoute. Tambahkan kode berikut pada blok fungsi build dalam

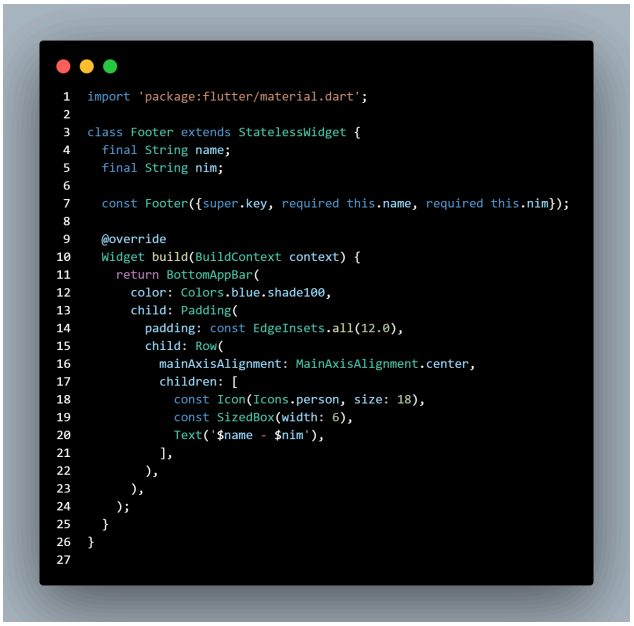
	halaman ItemPage. Setelah nilai didapatkan, anda dapat menggunakannya seperti penggunaan variabel pada umumnya. (https://docs.flutter.dev/cookbook/navigation/navigate-with-arguments) `final itemArgs = ModalRoute.of(context)!.settings.arguments as Item;`
3	Pada hasil akhir dari aplikasi belanja yang telah anda selesaikan, tambahkan atribut foto produk, stok, dan rating. Ubahlah tampilan menjadi GridView seperti di aplikasi marketplace pada umumnya. **Output**

```dart
class Item {
  String name;
  int price;
  String photo;
  int stock;
  double rating;

  Item({
    required this.name,
    required this.price,
    required this.photo,
    required this.stock,
    required this.rating,
  });
}
```

```dart
body: Padding(
  padding: const EdgeInsets.all(8.0),
  child: GridView.builder(
    padding: EdgeInsets.all(8),
    itemCount: items.length,
    gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
      crossAxisCount: 2, // tampil 2 kolom
      crossAxisSpacing: 8,
      mainAxisSpacing: 8,
      childAspectRatio: 0.8, // proporsi kartu
    ),
    itemBuilder: (context, index) {
      final item = items[index];
      return InkWell(
        onTap: () {
          Navigator.pushNamed(context, '/item', arguments: item);
        },
```

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

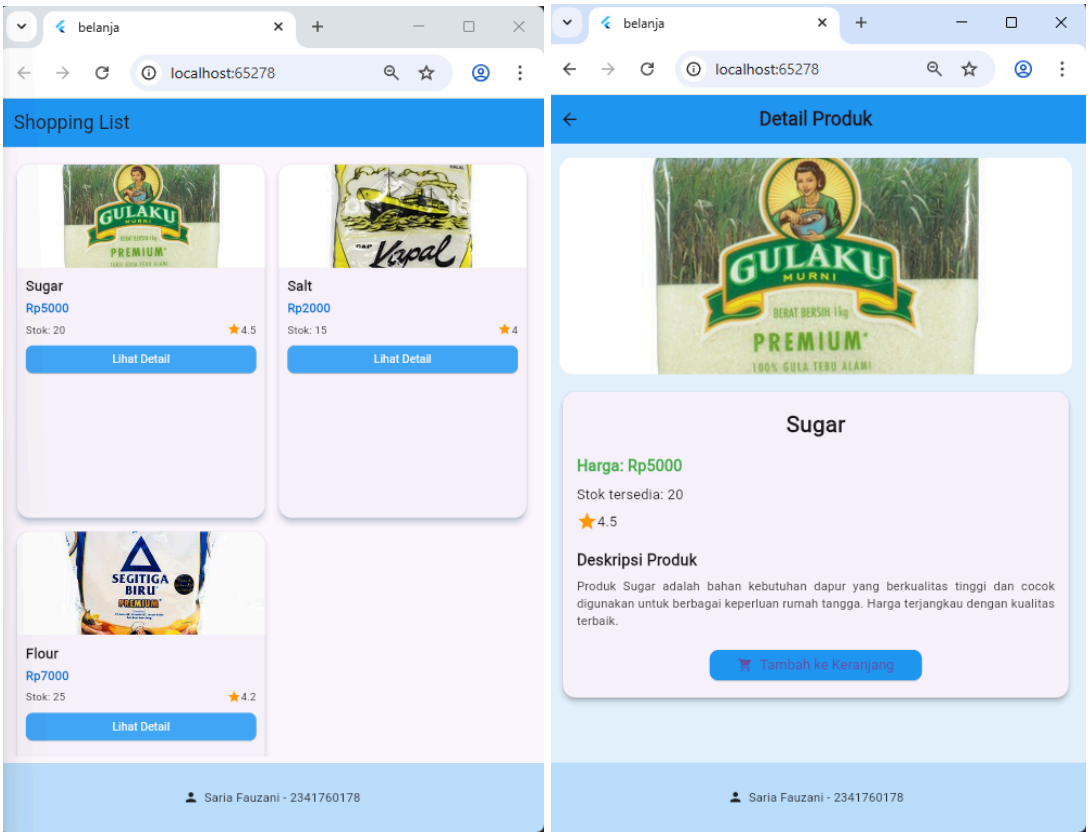

</td></tr>
<tr><td>5</td><td>Sesuaikan dan modifikasi tampilan sehingga menjadi aplikasi yang menarik. Selain itu, pecah widget menjadi kode yang lebih kecil. Tambahkan Nama dan NIM di footer aplikasi belanja Anda.</td></tr>
</table>

	
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!

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

class Footer extends StatelessWidget {
  final String name;
  final String nim;

  const Footer({super.key, required this.name, required this.nim});

  @override
  Widget build(BuildContext context) {
    return BottomAppBar(
      color: Colors.blue.shade100,
      child: Padding(
        padding: const EdgeInsets.all(12.0),
        child: Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            const Icon(Icons.person, size: 18),
            const SizedBox(width: 6),
            Text('$name - $nim'),
          ],
        ),
      ),
    );
  }
}
```

```dart
import 'package:belanja/models/item.dart';
import 'package:belanja/pages/home_page.dart';
import 'package:belanja/pages/item_page.dart';
import 'package:flutter/material.dart';
import 'package:go_router/go_router.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    final GoRouter router = GoRouter(
      routes: [
        GoRoute(
          path: '/',
          builder: (context, state) => HomePage(),
        ),
        GoRoute(
          path: '/item',
          builder: (context, state) {
            final item = state.extra as Item;
            return ItemPage(item: item);
          },
        ),
      ],
    );

    return MaterialApp.router(
      routerConfig: router,
      debugShowCheckedModeBanner: false,
    );
  }
}
```

Output


belanja
localhost:65278
Shopping List
Sugar
Rp5000
Stok: 20
4.5
Lihat Detail
Salt
Rp2000
Stok: 15
4
Lihat Detail
Flour
Rp7000
Stok: 25
4.2
Lihat Detail
Saria Fauzani - 2341760178
belanja
localhost:65278
Detail Produk
Sugar
Harga: Rp5000
Stok tersedia: 20
4.5
Deskripsi Produk
Produk Sugar adalah bahan kebutuhan dapur yang berkualitas tinggi dan cocok digunakan untuk berbagai keperluan rumah tangga. Harga terjangkau dengan kualitas terbaik.
Tambah ke Keranjang
Saria Fauzani - 2341760178