



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

Kelas : SIB 3E
NIM : 2241760025
Nama : Hertin Nurhayati
Jobsheet Ke- : 6

Laporan Jobsheet

Praktikum Ke-1

Langkah	Jawaban/Deskripsi
1.	Buatlah sebuah project flutter baru dengan nama layout_flutter. Atau sesuaikan style laporan praktikum yang Anda buat.
2.	Buka file main.dart lalu ganti dengan kode berikut. Isi nama dan NIM Anda di text title <pre>lib > main.dart > ... 1 import 'package:flutter/material.dart'; 2 3 void main() => runApp(const MyApp()); 4 5 class MyApp extends StatelessWidget { 6 const MyApp({super.key}); 7 8 @override 9 Widget build(BuildContext context) { 10 return MaterialApp(11 title: 'Flutter layout: Hertin Nurhayati dan 2241760025', 12 home: Scaffold(13 appBar: AppBar(14 title: const Text('Flutter layout demo'), 15), // AppBar 16 body: const Center(17 child: Text('Hello World'), 18), // Center 19), // Scaffold 20); // MaterialApp 21 } 22</pre>
3.	Identifikasi layout diagram Langkah pertama adalah memecah tata letak menjadi elemen dasarnya: <ul style="list-style-type: none">• Identifikasi baris dan kolom.



	<ul style="list-style-type: none">• 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.
4.	<p>Pertama, Anda akan membuat kolom bagian kiri pada judul. Tambahkan kode berikut di bagian atas metode build() di dalam kelas MyApp:</p> <pre>import 'package:flutter/material.dart'; void main() => runApp(const MyApp()); class MyApp extends StatelessWidget { const MyApp({super.key}); @override Widget build(BuildContext context) { // Implementasi title row Widget titleSection = Container(padding: const EdgeInsets.all(32), // Padding di sepanjang setiap tepi sebesar 32 piksel child: Row(children: [Expanded(/* soal 1 */ child: Column(crossAxisAlignment: CrossAxisAlignment.start, // Mengatur posisi kolom berada di awal baris</pre>



```
children: [  
    /* soal 2 */  
    Container(  
        padding: const EdgeInsets.only(bottom: 8), // Padding bawah sebesar 8  
        child: const Text(  
            'Wisata Gunung di Batu',  
            style: TextStyle(  
                fontWeight: FontWeight.bold, // Teks tebal  
            ),  
        ),  
        Text(  
            'Batu, Malang, Indonesia',  
            style: TextStyle(  
                color: Colors.grey[500], // Warna teks menjadi abu-abu  
            ),  
        ),  
    ],  
),  
),  
/* soal 3 */  
Icon(  
    Icons.star, // Menggunakan ikon bintang  
    color: Colors.red[500], // Mengatur warna ikon menjadi merah  
,  
    const Text('41'), // Menampilkan teks "41"  
],
```



```
),  
);  
  
return MaterialApp(  
  title: 'Flutter layout: Hertin Nurhayati dan 2241760025',  
  home: Scaffold(  
    appBar: AppBar(  
      title: const Text('Flutter layout demo'),  
    ),  
    body: Column(  
      children: [  
        titleSection, // Menambahkan titleSection ke dalam Column  
      ],  
    ),  
  ),  
);  
}
```

Praktikum Ke-2

Langkah	Jawaban/Deskripsi
1.	Buat method Column _buildButtonColumn lib/main.dart (_buildButtonColumn)



	<pre>// Langkah 1: Method _buildButtonColumn Column _buildButtonColumn(Color color, IconData icon, String label) { return Column(mainAxisAlignment: MainAxisAlignment.min, mainAxisAlignment: MainAxisAlignment.center, children: <Widget>[Icon(icon: icon, color: color), Container(margin: const EdgeInsets.only(top: 8), // Margin atas 8 piksel child: Text(data: label, style: TextStyle(fontSize: 12, fontWeight: FontWeight.w400, color: color,), // TextStyle), // Text), // Container],); // Column }</pre>
2.	<p>Buat widget buttonSection</p> <p>lib/main.dart (buttonSection)</p> <pre>// Langkah 2: Buat widget buttonSection Color color = Theme.of(context).primaryColor; Widget buttonSection = Row(mainAxisAlignment: MainAxisAlignment.spaceEvenly, // Mengatur ruang kosong secara children: <Widget>[_buildButtonColumn(color: color, icon: Icons.call, label: 'CALL'), // Tombol "CAL _buildButtonColumn(color: color, icon: Icons.near_me, label: 'ROUTE'), // Tombol _buildButtonColumn(color: color, icon: Icons.share, label: 'SHARE'), // Tombol "S],);</pre>
3.	<p>Tambah button section ke body</p> <p>Tambahkan variabel buttonSection ke dalam body seperti berikut:</p> <pre>return MaterialApp(title: 'Flutter layout: Hertin Nurhayati dan 2241760025', home: Scaffold(appBar: AppBar(title: const Text(data: 'Flutter layout demo'),), // AppBar body: Column(children: <Widget>[titleSection, // Menambahkan titleSection ke dalam Column buttonSection, // Menambahkan buttonSection ke dalam Column],),),);</pre>

Praktikum Ke-3 Implementasi text section

Langkah	Jawaban/Deskripsi
1.	Buat widget textSection



	<pre>// Langkah 1 Praktikum 3: Buat widget textSection Widget textSection = Container(padding: const EdgeInsets.all(value: 32), // Padding di sepanjang setiap tepi seb child: const Text(data: 'Carilah teks di internet yang sesuai ' 'dengan foto atau tempat wisata yang ingin ' 'Anda tampilkan. ' 'Tambahkan nama dan NIM Anda sebagai ' 'identitas hasil pekerjaan Anda. ' 'Selamat mengerjakan 😊.', softWrap: true, // Mengatur teks agar bisa membungkus jika terlalu panjang), // Text); // Container</pre>
2.	<p>Tambahkan variabel text section ke body</p> <p>Tambahkan widget variabel textSection ke dalam body seperti berikut:</p> <pre>titleSection, // Menambahkan titleSection ke dalam Column buttonSection, // Menambahkan buttonSection ke dalam Column textSection,</pre>

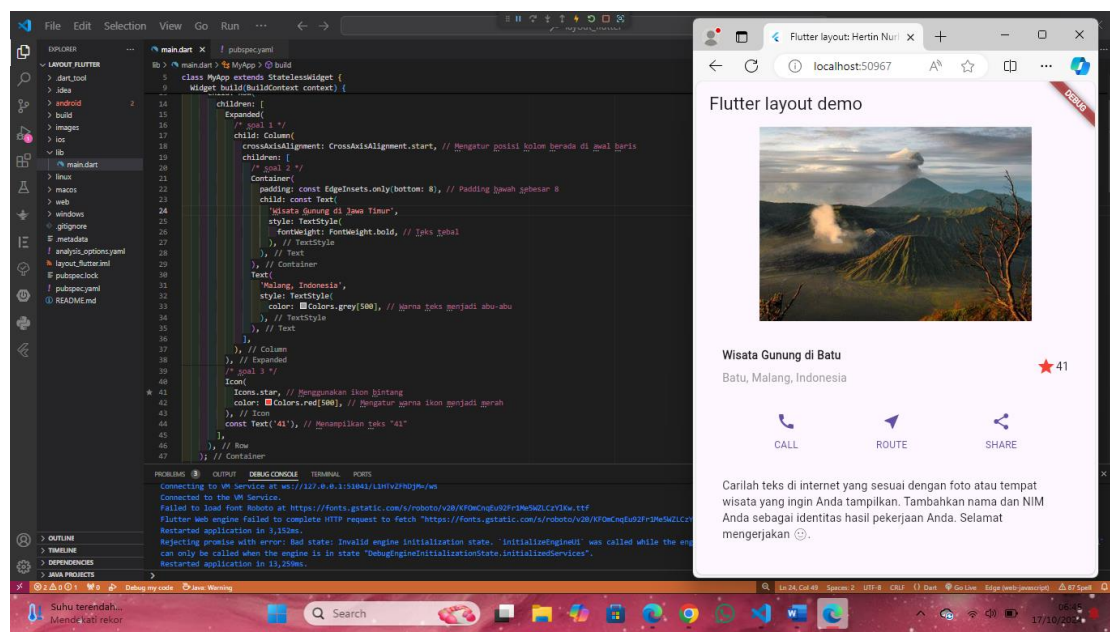
Praktikum Ke-4 Implementasi image section

Langkah	Jawaban/Deskripsi
1.	<p>Siapkan aset gambar</p> <pre>flutter: uses-material-design: true assets: - images/bromo.jpeg</pre>
2.	<p>Tambahkan gambar ke body</p> <p>Tambahkan aset gambar ke dalam body seperti berikut:</p> <pre>), // Appbar body: Column(children: [Image.asset('images/bromo.jpeg', width: 00, height: 240,), // Image.asset titleSection, // Menambahkan titleSection ke dalam Column buttonSection, // Menambahkan buttonSection ke dalam Column textSection,</pre>
3.	<p>ubah menjadi ListView</p>

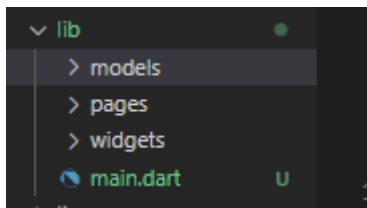


```
return MaterialApp(  
  title: 'Flutter layout: Hertin Nurhayati dan 2241760025',  
  home: Scaffold(  
    appBar: AppBar(  
      title: const Text('Flutter layout demo'),  
    ), // AppBar  
    body: ListView(  
      children: [  
        Image.asset(  
          'images/bromo.jpeg',  
          width: 80,  
          height: 240,  
        ), // Image.asset  
        titleSection, // Menambahkan titleSection ke dalam Column  
        buttonSection, // Menambahkan buttonSection ke dalam Column  
        textSection,  
      ],  
    ),  
  ),  
);
```

HASIL TUGAS PRAKTIKUM 1:



Praktikum Ke-5 Membangun Navigasi di Flutter

Langkah	Jawaban/Deskripsi
1.	Siapkan project baru 
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

```
lib > pages > item_page.dart > ItemPage > build
1 import 'package:flutter/material.dart';
2 import 'package:js6_praktikum5_belanja/models/item.dart'; // Pastikan model Item diimport
3 class ItemPage extends StatelessWidget {
4   final Item item; // Parameter untuk menerima data item
5   // Constructor untuk menerima data item
6   ItemPage({required this.item});
7   @override
8   Widget build(BuildContext context) {
9     return Scaffold(
10      appBar: AppBar(
11        title: Text('Shopping List'), // Menampilkan nama item di AppBar
12      ), // AppBar
13      body: Center(
14        child: Column(
15          mainAxisAlignment: MainAxisAlignment.center,
16          children: [
17            Text('${item.name} with ${item.price}'),
18            //Text('Item: ${item.name} with Price: Rp${item.price}'), // Menampilkan nama item
19            //Text('Price: Rp${item.price}'), // Menampilkan harga item
20          ],
21        ), // Column
22      ), // Center
23    ); // Scaffold
24  }
25 }
```

```
lib > pages > home_page.dart > ...
1 import 'package:flutter/material.dart';
2 import 'package:js6_praktikum5_belanja/models/item.dart';
3 import 'package:js6_praktikum5_belanja/pages/item_page.dart';
4 class HomePage extends StatelessWidget {
5   final List<Item> items = [
6     Item(name: 'Sugar', price: 5000),
7     Item(name: 'Salt', price: 2000),
8   ];
9   Widget build(BuildContext context) {
10     return Scaffold(
11       appBar: AppBar(
12         title: Text('Shopping List'),
13       ), // AppBar
14       body: Container(
15         margin: EdgeInsets.all(8), // Memberikan margin di sekitar ListView
16         child: ListView.builder(
17           padding: EdgeInsets.all(8), // Padding untuk item di dalam ListView
18           itemCount: items.length,
19           itemBuilder: (context, index) {
20             final item = items[index];
21             return InkWell(
22               onTap: () {
23                 // Navigasi ke halaman ItemPage dengan mengirim data item
24                 Navigator.pushNamed(
25                   context,
26                   '/item',
27                   arguments: item, // Kirim data item sebagai argument
28                 );
29               },
30             );
31           },
32         ),
33       ),
34     );
35   }
36 }
```




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**.

```
lib > main.dart x item.dart home_page.dart 1 item_page.dart
1 import 'package:flutter/material.dart';
2 import 'package:js6_praktikum5_belanja/pages/home_page.dart';
3 import 'package:js6_praktikum5_belanja/pages/item_page.dart';
4 import 'package:js6_praktikum5_belanja/models/item.dart';
5
Run | Debug | Profile
6 void main() {
7   runApp(MaterialApp(
8     initialRoute: '/',
9     onGenerateRoute: (settings) {
10       if (settings.name == '/item') {
11         final item = settings.arguments as Item; // Ambil argumen item
12         return MaterialPageRoute(
13           builder: (context) => ItemPage(item: item), // Kirim item ke ItemPage
14         ); // MaterialPageRoute
15       }
16       return null;
17     },
18     routes: {
19       '/': (context) => HomePage(),
20     },
21   )); // MaterialApp
22 }
```

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.

```
lib > models > item.dart > Item
1 class Item {
2   String name; // Nama item
3   double price; // Harga item
4   Item({required this.name, required this.price});
5 }
```



5.	<p>Lengkapi kode di class HomePage</p> <p>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.</p> <pre>class HomePage extends StatelessWidget { final List<Item> items = [Item(name: 'Sugar', price: 5000), Item(name: 'Salt', price: 2000),]; }</pre>
6.	<p>Membuat ListView dan itemBuilder</p> <p>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.</p>



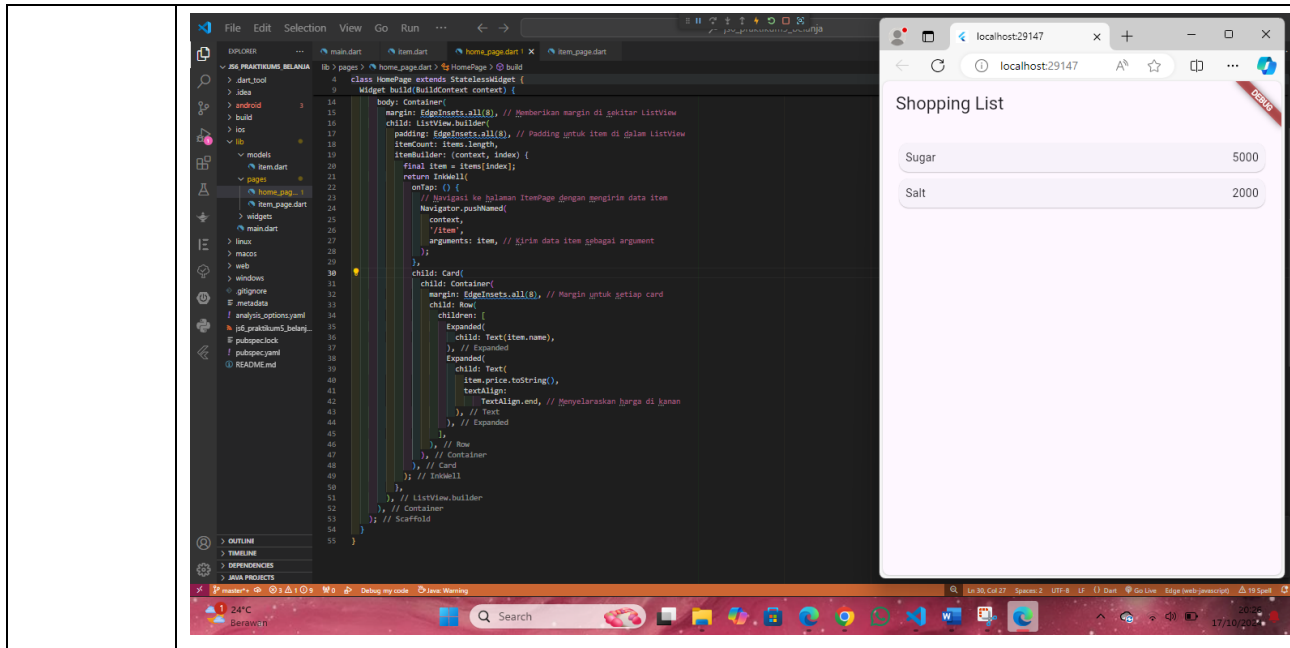
KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141

Telp. (0341) 404424 – 404425, Fax (0341) 404420

<https://jti.polinema.ac.id>

```
body: Container(  
  margin: EdgeInsets.all(8), // Memberikan margin di sekitar ListView  
  child: ListView.builder(  
    padding: EdgeInsets.all(8), // Padding untuk item di dalam ListView  
    itemCount: items.length,  
    itemBuilder: (context, index) {  
      final item = items[index];  
      return InkWell(  
        onTap: () {  
          // Navigasi ke halaman ItemPage dengan mengirim data item  
          Navigator.pushNamed(  
            context,  
            '/item',  
            arguments: item, // Kirim data item sebagai argument  
          );  
        },  
        child: Card(  
          child: Container(  
            margin: EdgeInsets.all(8), // Margin untuk setiap card  
            child: Row(  
              children: [  
                Expanded(  
                  child: Text(item.name),  
                ), // Expanded  
                Expanded(  
                  child: Text(  
                    item.price.toString(),  
                    textAlign:  
                      TextAlign.end, // Menyelaraskan harga di kanan  
                  ), // Text  
                ), // Expanded  
              ],  
            ), // Row  
          ), // Container  
        ), // Card  
      ); // InkWell  
    },  
  ), // ListView.builder  
), // Container}
```



Tugas Praktikum 2

Langkah	Jawaban/Deskripsi
1.	<p>Untuk melakukan pengiriman data ke halaman berikutnya, cukup menambahkan informasi <code>arguments</code> pada penggunaan <code>Navigator</code>. Perbarui kode pada bagian <code>Navigator</code> menjadi seperti berikut.</p> <pre>Navigator.pushNamed(context, '/item', arguments: item, // Kirim data item sebagai argument);</pre>
2.	<p>Pembacaan nilai yang dikirimkan pada halaman sebelumnya dapat dilakukan menggunakan <code>ModalRoute</code>. Tambahkan kode berikut pada blok fungsi <code>build</code> dalam halaman <code>ItemPage</code>. Setelah nilai didapatkan, anda dapat menggunakannya seperti penggunaan variabel pada umumnya</p>



	<pre>final List<Item> items = [Item(name: 'Sugar', price: 5000, imageUrl: 'assets/images/gula.jpeg', stock: 10, rating: 4.5,), Item(name: 'Salt', price: 2000, imageUrl: 'assets/images/garam.jpeg', stock: 20, rating: 4.2,),];</pre>
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.
4.	Silakan implementasikan Hero widget pada aplikasi belanja Anda dengan mempelajari dari sumber ini: https://docs.flutter.dev/cookbook/navigation/hero-animations
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.

