

Tahun Ajar 2023/2024

# PEMROGRAMAN

# PERANGKAT BERGERAK

## Praktikum 5: Navigation dan Routing

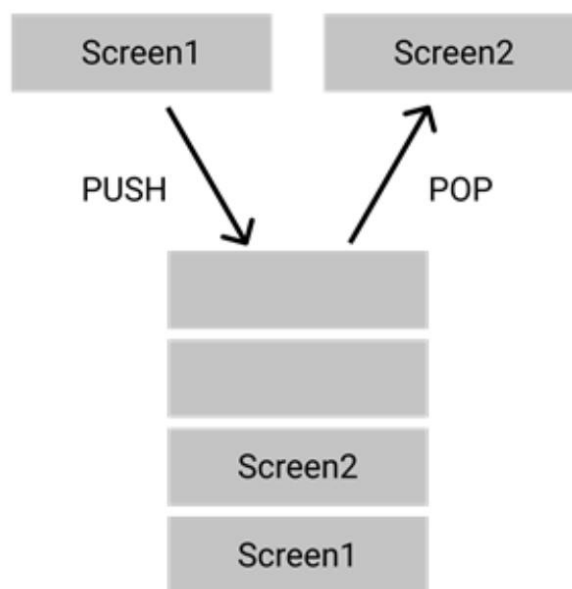
Pengembang Modul :

Novian Adi Prasetyo, S.Kom., M.Kom.  
Muhammad Lulu Latif Usman, S.Pd., M.Han.  
Muhamad Azrino Gustalika, S.Kom., M. Tr. T  
Akhmad Nur Alamsyah (Asisten Praktikum)  
Puspita Kartika Sari (Asisten Praktikum)

## Navigation dan Router

Pada saat membangun sebuah aplikasi kita akan membuat banyak screen/halaman dan akan berpindah dari satu halaman ke halaman lainnya. Dalam pemrograman android kita mengenal Intent untuk berpindah halaman. Pada flutter kita akan menggunakan sebuah class bernama Navigator. Dengan Navigator kita bisa berpindah dari satu halaman ke halaman lainnya.

Konsep navigasi pada Flutter mirip dengan pemrograman Android, yaitu ketika berpindah dari satu halaman ke halaman lain akan menjadi tumpukan/stack. Jadi halaman pertama akan ditumpuk dengan halaman kedua dan seterusnya. Dan apabila kembali ke halaman sebelumnya, maka halaman tersebut akan dihapus (pop).



### Navigator.push

Navigator.push digunakan untuk berpindah dari satu halaman ke halaman lain.

Navigator.push memiliki dua parameter. Parameter pertama adalah context dan yang kedua adalah route. Parameter route digunakan untuk menentukan tujuan kemana aplikasi akan berpindah halaman.

### Navigator.pop

Pada Navigator.pop cukup hanya menambahkan parameter context saja. Navigator.pop akan mengembalikan aplikasi ke halaman sebelumnya dengan cara menghapus halaman saat ini.

## Guided

### Langkah pertama:

Buat project flutter bernama praktikum5

### Langkah kedua:

Pada file main.dart buatlah StatelessWidget bernama MyApp yang me-return MaterialApp.

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

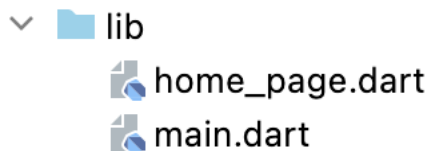
void main() {
  runApp(const MyApp());
}

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

  @override
  Widget build(BuildContext context) {
    return const MaterialApp();
  }
}
```

### Langkah ketiga:

Pada folder lib buat file baru bernama home\_page.dart



dan tambahkan StatelessWidget bernama HomePage:

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

class HomePage extends StatelessWidget {
  const HomePage({Key? key}) : super(key: key);

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

### Langkah keempat:

Pada class MyApp isi parameter home dengan class HomePage (halaman Home).

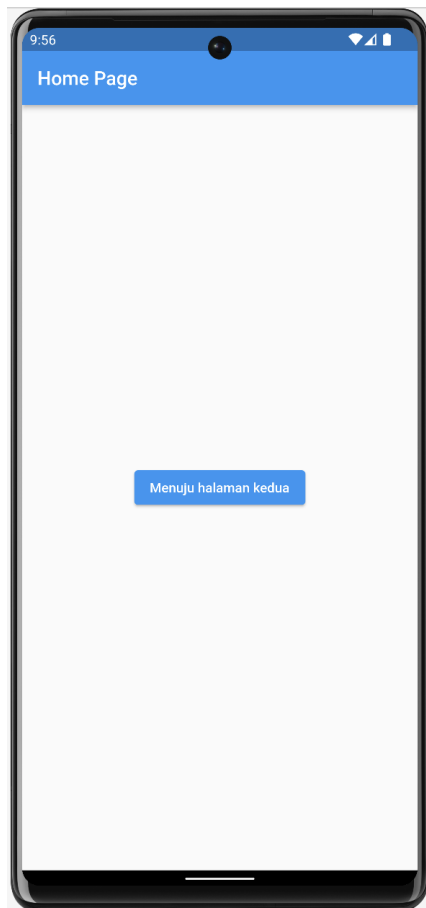
```
class MyApp extends StatelessWidget {  
  const MyApp({super.key});  
  
  @override  
  Widget build(BuildContext context) {  
    return const MaterialApp(  
      debugShowCheckedModeBanner: false,  
      home: HomePage(),  
    );  
  }  
}
```



## Langkah kelima

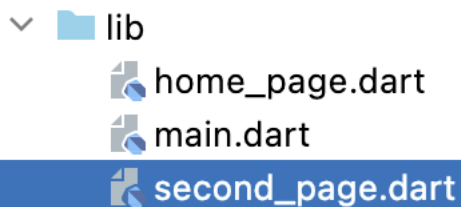
Tambahkan tombol pada body halaman HomePage.

```
class HomePage extends StatelessWidget {  
  const HomePage({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('Home Page'),  
      ),  
      body: Center(  
        child: ElevatedButton(  
          onPressed: () {},  
          child: const Text('Menuju halaman kedua'),  
        ),  
      ),  
    );  
  }  
}
```



## Langkah keenam:

Buat file baru bernama `second_page.dart` dan tambahkan StatelessWidget bernama `SecondPage`.



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

class SecondPage extends StatelessWidget {
  const SecondPage({Key? key}) : super(key: key);

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

### Langkah ketujuh:

Pada class HomePage tambahkan Navigator pada fungsi onPressed untuk menuju halaman Second Page.

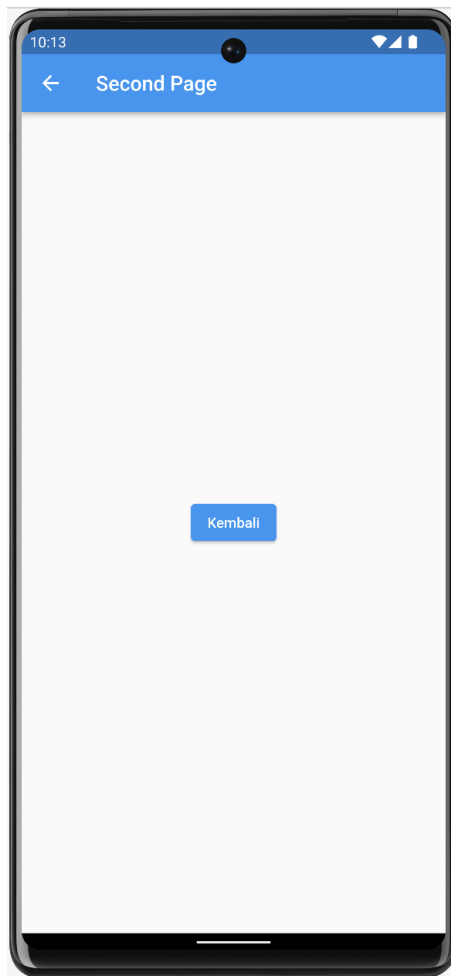
```
body: Center(  
  child: ElevatedButton(  
    onPressed: () {  
      Navigator.push(  
        context,  
        MaterialPageRoute(  
          builder: (context) => const SecondPage(),  
        ),  
      );  
    },  
    child: const Text('Menuju halaman kedua'),  
  ),  
,
```

Jalankan aplikasi maka ketika tombol ditekan halaman aplikasi akan berpindah menampilkan halaman Second Page.

### Langkah kedelapan:

Menambahkan tombol kembali pada halaman Second Page.

```
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    appBar: AppBar(  
      title: const Text('Second Page'),  
    ),  
    body: Center(  
      child: ElevatedButton(  
        onPressed: () {},  
        child: const Text('Kembali'),  
      ),  
    ),  
  );  
}
```



### Langkah kesembilan:

Tambahkan Navigator pop pada tombol kembali agar ketika tombol ditekan akan kembali ke halaman sebelumnya.

```
child: ElevatedButton(  
  onPressed: () {  
    Navigator.pop(context) ;  
  },  
  child: const Text('Kembali'),  
) ,
```

### Mengirimkan data dari satu halaman ke halaman lain

Seringkali beberapa halaman pada aplikasi perlu berbagi dan mengirimkan data. Pada Slutter dapat memanfaatkan constructor dari sebuah class untuk mengirimkan data antar halaman.

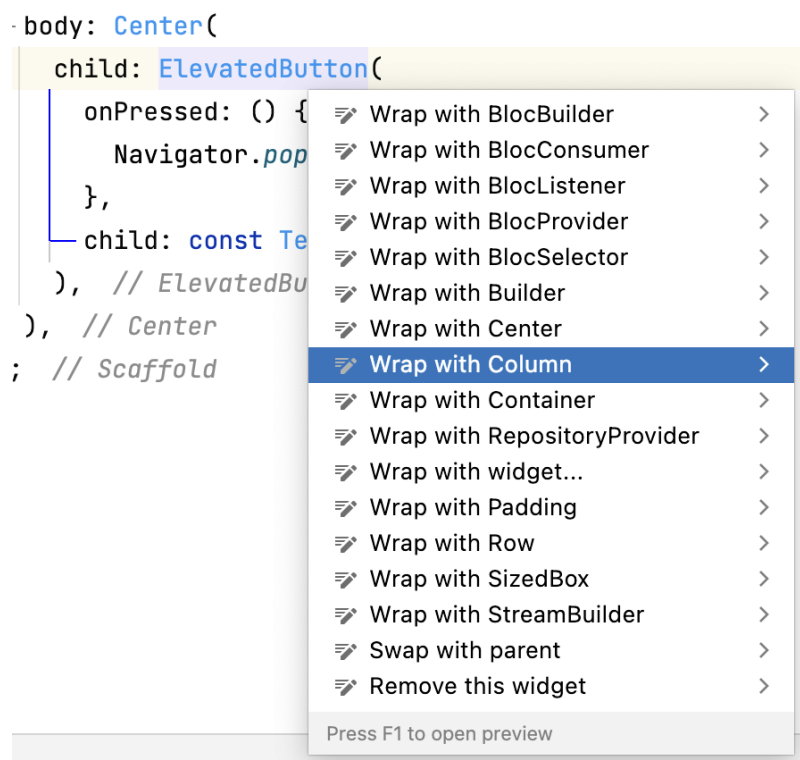
### Langkah kesepuluh:

Tambahkan variable data pada SecondPage dan ubah constructor SecondPage. Variable data ini nantinya akan menyimpan data yang dikirimkan dari HomePage.

```
class SecondPage extends StatelessWidget {  
  final String? data;  
  
  const SecondPage({Key? key, this.data}) : super(key: key);  
}
```

Bungkus tombol (ElevatedButton) dengan Column dan tambahkan widget Text yang akan menampilkan data.

Untuk membungkus widget pada Android Studio dengan menekan (alt+enter), pada vscode tekan (ctrl+.)



```
body: Center(  
  child: Column(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: [  
      Text(data ?? '', style: const TextStyle(fontSize: 20.0)),  
      ElevatedButton(  
        onPressed: () {  
          Navigator.pop(context);  
        },  
        child: const Text('Kembali'),  
      ),  
    ],  
  ),  
,  
,  
,  
)
```

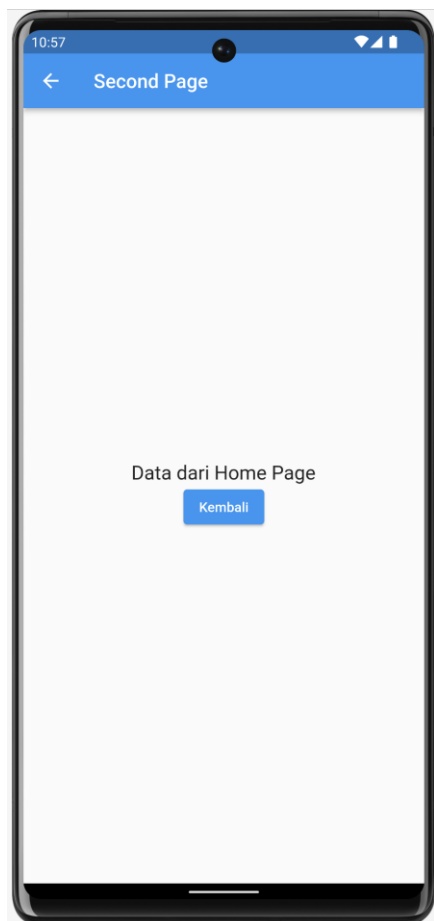


Langkah kesebelas:

Pada HomePage tambahkan argument untuk parameter data.

```
body: Center(  
  child: ElevatedButton(  
    onPressed: () {  
      Navigator.push(  
        context,  
        MaterialPageRoute(  
          builder: (context) => const SecondPage(data: 'Data dari Home Page'),  
        ),  
      );  
    },  
    child: const Text('Menuju halaman kedua'),  
  ),  
)
```

Maka ketika tombol pada halaman Home Page ditekan aplikasi akan menampilkan halaman Second Page dengan data dari halaman Home Page.



## Menggunakan Route

Langkah duabelas:

Pada class MyApp hapus home dan daftarkan route tiap halaman.

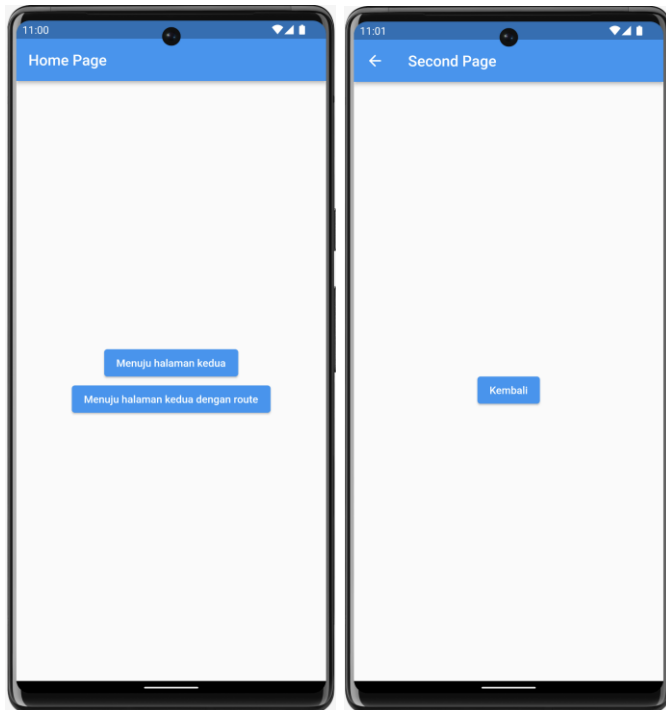
```
class MyApp extends StatelessWidget {  
  const MyApp({super.key});  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      debugShowCheckedModeBanner: false,  
      initialRoute: '/',  
      routes: {  
        '/': (context) => const HomePage(),  
        '/second_page': (context) => const SecondPage(),  
      },  
    );  
  }  
}
```

Langkah ketigabelas:

Bungkus tombol pada Home Page dengan Column dan tambahkan tombol baru.

```
class HomePage extends StatelessWidget {  
  const HomePage({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('Home Page'),  
      ),  
      body: Center(  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.center,  
          children: [  
            ElevatedButton(  
              onPressed: () {  
                Navigator.push(  
                  context,  
                  MaterialPageRoute(  
                    builder: (context) => const SecondPage(  
                      data: 'Data dari Home Page',  
                    ),  
                ),  
              },  
            ),  
            child: const Text('Menuju halaman kedua'),  
          ],  
        ),  
        // Tombol Kedua  
        ElevatedButton(  
          onPressed: () {  
            Navigator.pushNamed(context, '/second_page');  
          },  
          child: const Text('Menuju halaman kedua dengan route'),  
        ),  
      ],  
    ),  
  ),  
);
```

Ketika tombol kedua ditekan maka aplikasi akan menampilkan halaman Second Page.



Untuk membuat tombol kedua (menggunakan route) dapat mengirimkan data maka buat variable data2 dan tambahkan widget Text pada class SecondPage.

```
@override
Widget build(BuildContext context) {
  final data2 = ModalRoute.of(context)!.settings.arguments;

  return Scaffold(
    appBar: AppBar(
      title: const Text('Second Page'),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text(
            data ?? '',
            style: const TextStyle(fontSize: 20.0),
          ),
          Text(
            data2.toString() ?? '',
            style: const TextStyle(fontSize: 20.0),
          ),
          ElevatedButton(
            onPressed: () {
              Navigator.pop(context);
            },
            child: const Text('Kembali'),
          ),
        ],
      ),
    ),
  );
}
```

Langkah keempatbelas:

Pada tombol kedua ('menuju halaman kedua dengan route') tambahkan argument seperti berikut.

```
ElevatedButton(  
  onPressed: () {  
    Navigator.pushNamed(context, '/second_page',arguments:  
    'Menggunakan route');  
  },  
  child: const Text('Menuju halaman kedua dengan route'),  
)
```

Ketika tombol kedua ditekan aplikasi akan menampilkan Second Page dengan data.

