

MODUL 12

Aplikasi dengan Navigasi Dinamis



CAPAIAN PEMBELAJARAN

Mahasiswa dapat membuat aplikasi sederhana menggunakan **state** untuk mengelola navigasi antar layar tanpa menggunakan Grafik Navigasi.



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Android Studio Koala 2024.1.2 atau yang lebih tinggi
2. Handphone Android Terbaru
3. Kabel data USB.
4. Driver ADB



DASAR TEORI

Dalam Jetpack Compose, **NavHostController** dan **NavHost** biasanya digunakan untuk mengatur navigasi antarlayar secara deklaratif. Namun, jika Anda ingin membuat navigasi antarlayar **tanpa menggunakan NavHostController**, Anda bisa menggunakan logika manual dengan memanfaatkan **State** dan **Composable Functions**. Pada pendekatan ini, Anda secara eksplisit menentukan tampilan berdasarkan kondisi logis tertentu, seperti penggunaan **State** untuk menentukan layar mana yang sedang ditampilkan.

Pendekatan ini melibatkan:

1. **State Management**: Menggunakan state untuk menyimpan status layar mana yang sedang ditampilkan.
2. **Composable Function**: Menampilkan layar sesuai dengan state yang sedang aktif.

3. **Manual Screen Switching:** Menggunakan logika seperti `if` atau `when` untuk berpindah antar layar.

Kita akan menggunakan **MutableState** untuk mengatur layar mana yang sedang ditampilkan. Di sini, **currentScreen** akan menyimpan informasi tentang layar mana yang aktif. Perhatikan cuplikan kode berikut.

```
// State untuk menentukan layar mana yang aktif
var currentScreen by remember { mutableStateOf("home") }

// Menampilkan layar berdasarkan state
when (currentScreen) {
    "home" -> HomeScreen(onNavigate = { currentScreen = "detail" })
    "detail" -> DetailScreen(onNavigateBack = { currentScreen = "home" })
}
```

Penjelasan:

- **State (currentScreen)** digunakan untuk menyimpan informasi tentang layar mana yang sedang ditampilkan.
- **remember { mutableStateOf("home") }** digunakan untuk menginisialisasi state dengan layar awal, yaitu `home`.
- **when (currentScreen)** digunakan untuk memutuskan layar mana yang akan ditampilkan. Jika **currentScreen** adalah `"home"`, maka akan menampilkan **HomeScreen**. Jika **currentScreen** adalah `"detail"`, maka akan menampilkan **DetailScreen**.
- Parameter **onNavigate** dan **onNavigateBack** digunakan untuk mengganti layar secara manual.



PRAKTIK

1. Membuat Proyek Android Studio Baru dengan Jetpack Compose

1. **Buka Android Studio.**
2. Klik **"Start a new Android Studio project"**.
3. Pilih template **"Empty Compose Activity"**, lalu klik **"Next"**.

4. Isi detail proyek:
 - **Name:** NavigasiSederhana
 - **Package name:** com.example.navigasisederhana
 - **Save location:** Pilih lokasi yang diinginkan.
 - **Language:** Kotlin
 - **Minimum SDK:** Pilih sesuai kebutuhan.
5. Klik **"Finish"** dan tunggu hingga proyek selesai dibuat.

2. Membuat Layar (Screens) dengan Jetpack Compose

Kita akan membuat dua layar sederhana: **Main Screen** dan **Second Screen**.

a. Membuat File Kotlin untuk Layar

1. **Buat Package Baru (Opsional):**
 - Klik kanan pada direktori com.example.navigasisederhana.
 - Pilih **New > Package**.
 - Beri nama screens.
2. **Buat File Kotlin untuk Main Screen:**
 - Klik kanan pada package screens.
 - Pilih **New > Kotlin File/Class**.
 - Beri nama mainScreen.
 - Beri pilihan **File**

```
// MainScreen.kt
package com.example.navigasisederhana.screens

import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp

@Composable
fun MainScreen(onNavigate: () -> Unit) {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(16.dp),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text(
            text = "Ini adalah Main Screen",

```

```

        fontSize = 24.sp,
        modifier = Modifier.padding(bottom = 20.dp)
    )
    Button(onClick = onNavigate) {
        Text(text = "Pergi ke Second Screen")
    }
}
}

@Preview(showBackground = true)
@Composable
fun MainScreenPreview() {
    MainScreen({})
}

```

3. Buat File Kotlin untuk Second Screen:

- Klik kanan pada package screens.
- Pilih **New > Kotlin File/Class**.
- Beri nama SecondScreen.
- Beri pilihan **File**

```

// SecondScreen.kt
package com.example.navigasisederhana.screens

import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Button
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp

@Composable
fun SecondScreen(onNavigateBack: () -> Unit) {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(16.dp),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text(
            text = "Ini adalah Second Screen",
            fontSize = 24.sp,
            modifier = Modifier.padding(bottom = 20.dp)
        )
        Button(onClick = onNavigateBack) {
            Text(text = "Kembali ke Main Screen")
        }
    }
}

```

```

        }
    }
}

@Preview(showBackground = true)
@Composable
fun SecondScreenPreview() {
    SecondScreen({})
}

```

b. Menambahkan Layar ke MainActivity

Buka `MainActivity.kt` dan modifikasi agar dapat menampilkan layar sesuai navigasi yang diinginkan.

```

// MainActivity.kt
package com.example.navigasisederhana

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import com.example.navigasisederhana.screens.MainScreen
import com.example.navigasisederhana.screens.SecondScreen
import com.example.navigasisederhana.ui.theme.NavigasiSederhanaTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            NavigasiSederhanaTheme {
                Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding ->
                    MyApp()
                }
            }
        }
    }
}

@Composable
fun MyApp() {
    // Mengelola state navigasi
    var currentScreen by remember { mutableStateOf<Screen>(Screen.Main) }
}

```

```

when (currentScreen) {
    Screen.Main -> MainScreen (
        onNavigate = { currentScreen = Screen.Second }
    )
    Screen.Second -> SecondScreen (
        onNavigateBack = { currentScreen = Screen.Main }
    )
}

enum class Screen {
    Main, Second
}

```

c. Jalankan aplikasi, klik tombol yang ada. Perhatikan apa yang terjadi

3. Menambahkan Layar Tambahan

1. Buat File Kotlin untuk Third Screen:
 - a. Klik kanan pada package screens.
 - b. Pilih **New > Kotlin File/Class**.
 - c. Beri nama ThirdScreen.
 - d. Beri pilihan **File**
 - e. Buatlah fungsi composable **ThirdScreen** seperti pada file SecondScreen.kt
2. Tambahkan sebuah button lagi di SecondScreen untuk memanggil ThirdScreen.

```

@Composable
fun SecondScreen(
    onNavigate: () -> Unit,
    onNavigateBack: () -> Unit
) {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(16.dp),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text(
            text = "Ini adalah Second Screen",
            fontSize = 24.sp,
            modifier = Modifier.padding(bottom = 20.dp)
        )
        Button(onClick = onNavigate) {
            Text(text = "Pergi ke Third Screen")
        }
        Button(onClick = onNavigateBack) {
            Text(text = "Kembali ke Main Screen")
        }
    }
}

```

```

    }
}

@Preview(showBackground = true)
@Composable
fun SecondScreenPreview() {
    SecondScreen(
        onNavigate={},
        onNavigateBack={})
}

```

3. Selanjutnya dalam MainActivity di struktur when diubah menjadi seperti berikut.

```

when (currentScreen) {
    Screen.Main -> MainScreen (
        onNavigate = { currentScreen = Screen.Second }
    )
    Screen.Second -> SecondScreen (
        onNavigate = { currentScreen = Screen.Third },
        onNavigateBack = { currentScreen = Screen.Main }
    )
    Screen.Third -> ThirdScreen (
        onNavigateBack = { currentScreen = Screen.Second }
    )
}

```

4. Jalankan aplikasi, klik tombol yang ada. Perhatikan apa yang terjadi.

4. Mengirim Data antar Layar

Seringkali, kita perlu mengirim data dari satu layar ke layar lainnya. Berikut adalah cara melakukannya dengan menggunakan parameter pada composable.

a. Memodifikasi MainScreen untuk Mengirim Data

Misalkan, kita ingin mengirim pesan dari **MainScreen** ke **SecondScreen**.

1. Ubahlah MainScreen.kt:

Tambahkan TextField untuk memasukkan pesan.

```

@Composable
fun MainScreen(onNavigate: (String) -> Unit) {
    var message by remember { mutableStateOf("") }

    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(16.dp),
    )
}

```

```

        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text(
            text = "Ini adalah Main Screen",
            fontSize = 24.sp,
            modifier = Modifier.padding(bottom = 20.dp)
        )
        TextField(
            value = message,
            onChange = { message = it },
            label = { Text("Masukkan Pesan") },
            modifier = Modifier
                .fillMaxWidth()
                .padding(bottom = 20.dp)
        )
        Button(onClick = { onNavigate(message) }) {
            Text(text = "Pergi ke Second Screen")
        }
    }
}

```

Jangan lupa tambahkan import berikut.

```

import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.material3.TextField

```

2. Update fungsi MyApp:

Di file **MainActivity.kt**, fungsi **MyApp** sesuaikan callback `onNavigate` untuk menerima pesan.

```

@Composable
fun MyApp() {
    // Mengelola state navigasi
    var currentScreen by remember { mutableStateOf<Screen>(Screen.Main) }
    var messageToSend by remember { mutableStateOf("") }

    when (currentScreen) {
        Screen.Main -> MainScreen (
            onNavigate = { message ->
                messageToSend = message
                currentScreen = Screen.Second
            }
        )
        Screen.Second -> SecondScreen (
            message = messageToSend,

```



```

        onNavigate = { currentScreen = Screen.Third },
        onNavigateBack = { currentScreen = Screen.Main }
    )
    Screen.Third -> ThirdScreen (
        onNavigateBack = { currentScreen = Screen.Second }
    )
}
}

```

b. Memodifikasi SecondScreen untuk Menerima Data

1. Update SecondScreen.kt:

Tambahkan parameter message untuk menampilkan pesan yang diterima.

```

@Composable
fun SecondScreen(
    message: String,
    onNavigate: () -> Unit,
    onNavigateBack: () -> Unit
) {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(16.dp),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text(
            text = "Ini adalah Second Screen",
            fontSize = 24.sp,
            modifier = Modifier.padding(bottom = 20.dp)
        )
        Text(
            text = "Pesan: $message",
            fontSize = 18.sp,
            modifier = Modifier.padding(bottom = 20.dp)
        )
        Button(onClick = onNavigate) {
            Text(text = "Pergi ke Third Screen")
        }
        Button(onClick = onNavigateBack) {
            Text(text = "Kembali ke Main Screen")
        }
    }
}

@Preview(showBackground = true)
@Composable

```

```
fun SecondScreenPreview() {  
    SecondScreen(  
        message = "",  
        onNavigate={},  
        onNavigateBack={})  
}
```

5. Menjalankan dan Menguji Aplikasi

Sekarang jalankan aplikasi dan uji navigasi antar layar.

Langkah-Langkah:

1. **Klik tombol "Run"** (ikon segitiga hijau) di Android Studio atau tekan **Shift + F10**.
2. **Pilih perangkat emulator atau fisik** untuk menjalankan aplikasi.
3. Setelah aplikasi diluncurkan, Anda akan melihat **Main Screen** dengan TextField dan tombol **"Pergi ke Second Screen"**.
4. **Masukkan pesan** di TextField.
5. **Klik tombol "Pergi ke Second Screen"**, dan Anda akan dinavigasi ke **Second Screen** yang menampilkan pesan yang Anda masukkan.
6. **Klik tombol "Kembali ke Main Screen"**, dan Anda akan kembali ke **Main Screen**.



LATIHAN

Diberikan oleh dosen pengampu



TUGAS

Diberikan oleh dosen pengampu



REFERENSI

1. <https://developer.android.com/develop/ui/compose/navigation?hl=id>