

**Mata Kuliah:**Pemrograman Mobile  
**Dosen Pengampu:**Muhayat, M.IT  
**Praktikum:**P4 - “Navigation: Explicit Intent · Implicit Intent · Fragment”  
**Nama:**Ruqayah  
**NIM:**230104040208  
**Kelas:**TI23A  
**Tanggal Praktikum:**15 Oktober 2025

---

## 1. Tujuan

1. Memahami konsep *navigation* dalam Android Compose, termasuk *activity navigation*, *data passing*, dan *fragment-like tab navigation*.
2. Mengimplementasikan *explicit intent* dan *implicit intent* untuk berpindah antar Activity.
3. Mempelajari mekanisme *back stack* dalam Compose Navigation.
4. Mengintegrasikan *nested navigation* untuk simulasi navigasi antar “fragment” menggunakan *bottom navigation*.

## 2. Dasar Teori

### a. Navigation Component

Navigation Component adalah library Android Jetpack yang memudahkan pengembang dalam mengatur alur perpindahan antar layar (destination) dalam aplikasi. Dalam Jetpack Compose, navigasi diatur dengan `NavController` dan `NavHost`.

### b. Explicit Intent

Explicit Intent digunakan untuk berpindah dari satu Activity ke Activity lain dengan menyebutkan target Activity secara langsung.

Contoh:

```
val intent = Intent(this, ActivityB::class.java)
startActivity(intent)
```

### c. Implicit Intent

Implicit Intent digunakan untuk memanggil komponen lain di luar aplikasi (misalnya membuka kamera, galeri, atau mengirim pesan), tanpa menyebutkan Activity tujuan secara eksplisit.

### d. Fragment dan Nested Navigation

Fragment merupakan bagian dari Activity yang dapat digunakan untuk membuat antarmuka modular. Pada Jetpack Compose, konsep ini disimulasikan menggunakan *nested navigation graph* dan *bottom navigation bar* yang merepresentasikan tab fragment.

### e. Back Stack

Back stack adalah struktur tumpukan (*stack*) yang menyimpan urutan layar yang telah dikunjungi. Saat pengguna menekan tombol “Back”, layar paling atas dihapus dari stack, dan sistem menampilkan layar sebelumnya.

### 3. Alat dan Bahan

- **Software:** Android Studio (Giraffe / Jellyfish), Kotlin, Jetpack Compose
- **Library utama:**
  - androidx.navigation:navigation-compose
  - androidx.compose.material3
  - androidx.compose.runtime
- **Perangkat uji:** Emulator Android API 33 atau perangkat fisik

### 4. Langkah-Langkah Praktikum

#### Langkah 1 — Membuat Proyek

Membuat proyek baru dengan nama *p4appnavigation\_230104040208* menggunakan template *Empty Compose Activity*.

#### Langkah 2 — Menyiapkan Struktur Folder

Membuat folder utama:

- nav/ → berisi file Route.kt dan NavGraph.kt
- screens/ → berisi file untuk setiap Activity (Home, ActivityA, ActivityB, ActivityC, ActivityD, StepScreen, HubScreen)

#### Langkah 3 — HomeScreen (Menu Utama)

Mendesain tampilan utama berisi 4 kartu menu:

- Start Activity (Explicit Intent)
- Send Data (Intent Extras)
- Back Stack
- Activity + Fragment (Nested Navigation)

#### Langkah 4 — Implementasi Explicit Intent

Menambahkan ActivityAScreen dan ActivityBScreen. Saat tombol *Open Detail* ditekan, aplikasi berpindah ke Activity B menggunakan `nav.navigate(Route.ActivityB.path)`.

#### Langkah 5 — Implementasi Passing Data (Intent Extras)

ActivityCScreen menerima input Nama dan NIM, lalu mengirim data ke ActivityDScreen. Data dikirim menggunakan argumen rute (`Route.ActivityD.make(name, nim)`).

#### Langkah 6 — Back Stack Simulation

Menambahkan tiga layar Step1, Step2, dan Step3 untuk memperlihatkan bagaimana layar ditumpuk dalam *back stack*. Tombol *Clear to Home* digunakan untuk kembali ke awal dengan menghapus stack.

#### Langkah 7 — Nested Navigation (Fragment Simulation)

Menambahkan HubScreen dengan tiga tab: Dashboard, Messages, dan Profile. Tab diatur

menggunakan `NavigationBar` dan setiap tab memiliki konten tersendiri. `MessageDetailScreen` dapat dibuka dari tab Messages untuk meniru navigasi antar fragment.

## 5. Hasil dan Tampilan

### Berhasil ditampilkan:

1. `HomeScreen` dengan empat menu navigasi.
2. Activity A → B: berpindah layar menggunakan *explicit intent*.
3. Activity C → D: pengiriman dan penerimaan data antar layar.
4. Step 1–3: simulasi *back stack*.
5. Hub (Fragment Navigation): navigasi tab bawah dengan Dashboard, Messages, dan Profile.
6. Message Detail: layar detail muncul setelah menekan pesan, meniru *fragment transaction*.

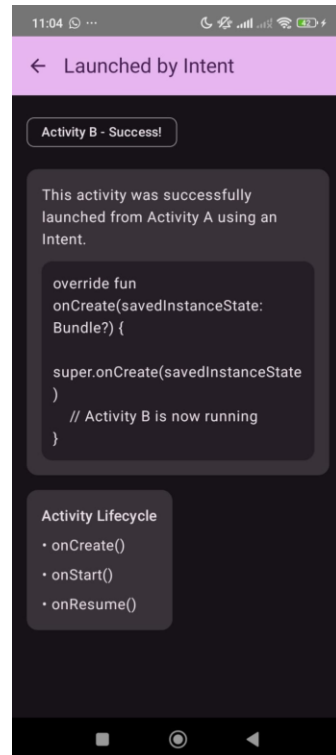
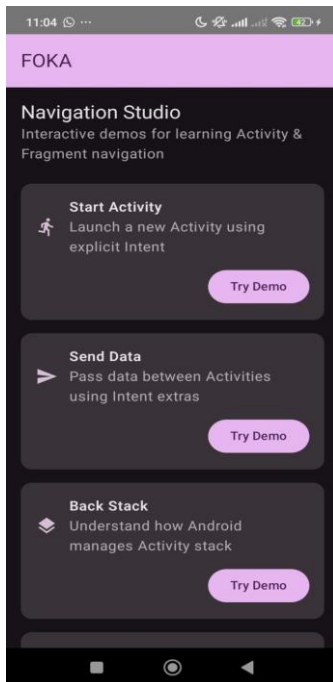
Seluruh fitur berjalan tanpa error, termasuk *TopBar* dengan tombol *Back* yang aktif sesuai kondisi stack.

## 6. Kesimpulan

1. Navigasi antar layar pada Jetpack Compose dapat dikelola secara efisien menggunakan *NavHostController*.
2. *Explicit Intent* digunakan untuk berpindah antar Activity secara langsung, sedangkan *Implicit Intent* untuk komponen eksternal.
3. *Back stack* membantu pengguna menavigasi kembali secara hierarkis sesuai urutan layar yang dikunjungi.
4. *Nested Navigation* memungkinkan satu Activity mengelola beberapa “fragment” sekaligus menggunakan Bottom Navigation.
5. Praktikum ini memperkuat pemahaman tentang bagaimana sistem navigasi modern di Compose menggantikan konsep lama berbasis XML dan Fragment tradisional

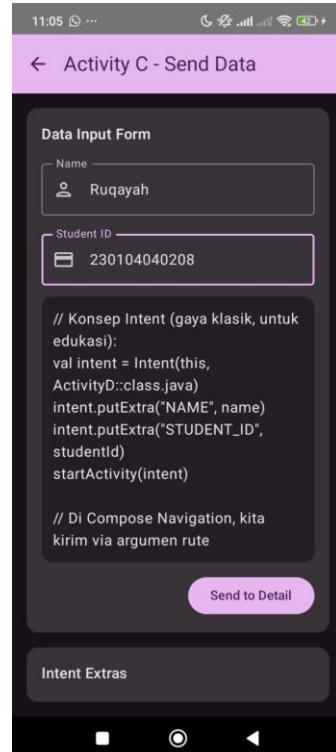
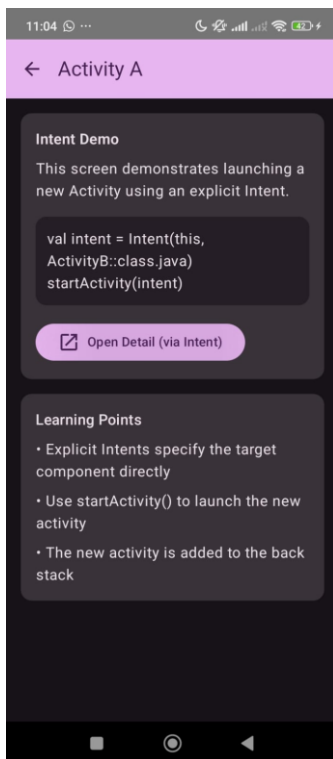
## Lampiran (Cuplikan Hasil):

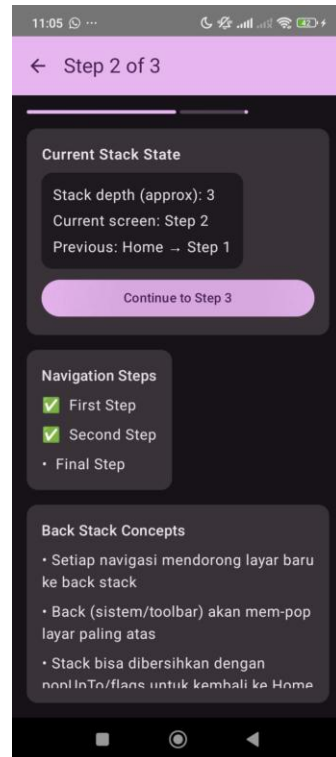
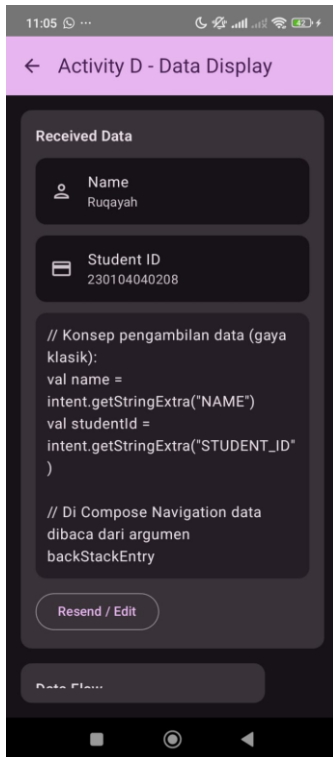
- Home Menu



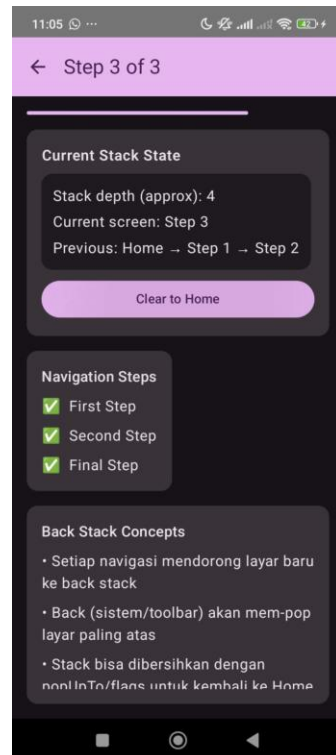
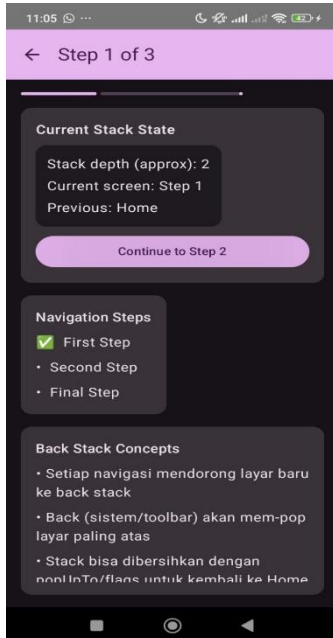
- Activity C & D (Data Passing)

- Activity A & B (Explicit Intent)

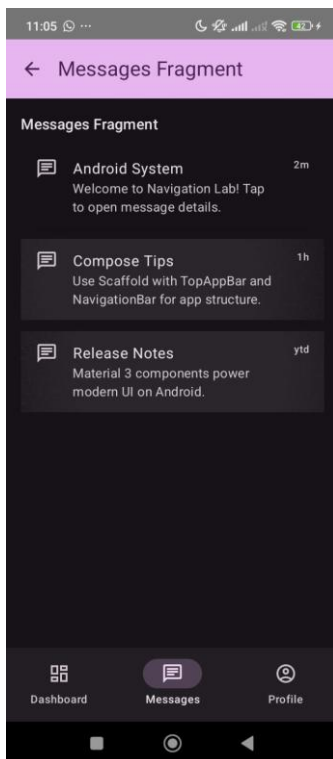
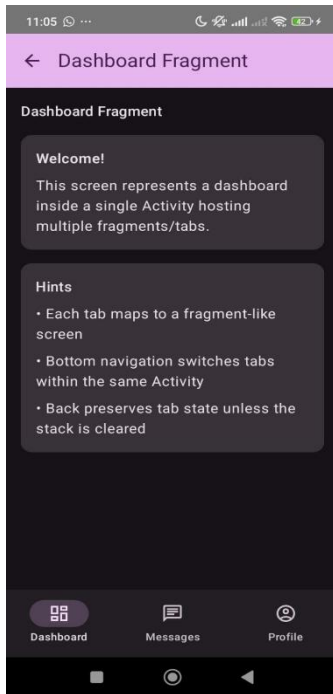




• Step 1–3 (Back Stack)



- Hub Dashboard / Messages / Profile



- Message Detail

