



Mobile Application

Catatan Kuliah #14

Alauddin Maulana Hirzan, M. Kom

0607069401

Flutter Navigation



Flutter Navigation

Mekanisme Perpindahan Halaman

Dalam aplikasi mobile, setiap aplikasi pastinya memiliki fungsi untuk berpindah halaman. Karena interaksi pengguna dengan aplikasi, tidak hanya terjadi di satu halaman saja.

Oleh karena itu, setiap pembuatan aplikasi pasti memiliki mekanisme untuk berpindah halaman sehingga memperlua interaksi pengguna terhadap pengguna.

Dengan bertambahnya halaman interaksi, juga dapat memperluas fungsi daripada aplikasi itu sendiri.

Flutter Navigation

Mekanisme Perpindahan Halaman





Flutter Navigation

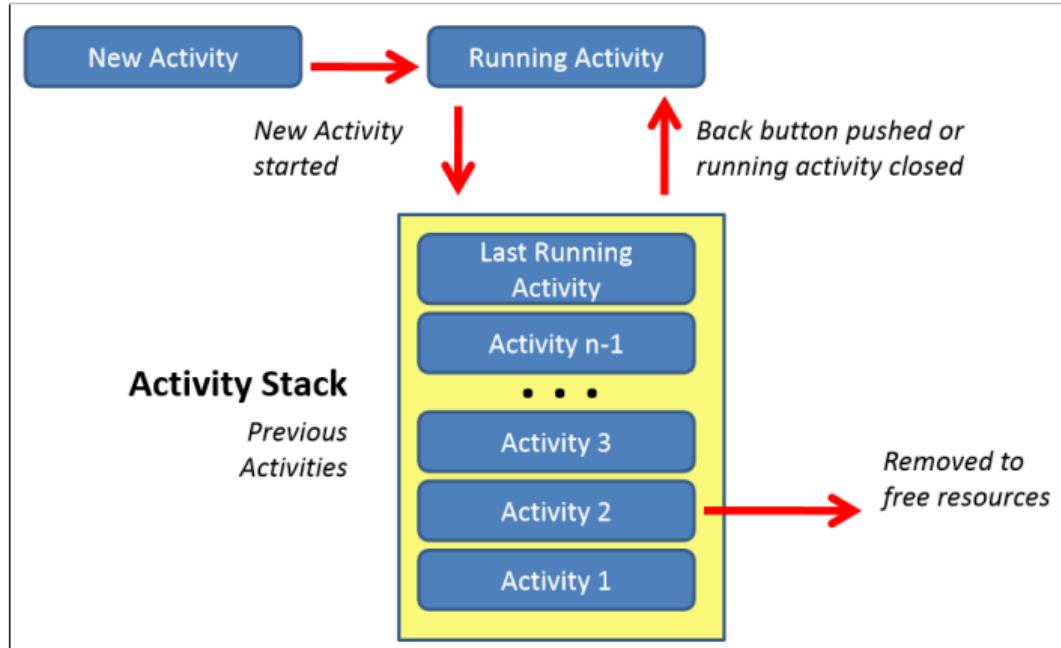
Mekanisme Tradisional Android

Android memiliki mekanisme dalam perpindahan halaman (Activity), Android menggunakan mekanisme **Stack** dalam menentukan halaman mana yang akan ditampilkan.

Activity Stack, juga dikenal sebagai "Back Stack", adalah konsep dasar untuk mengelola aliran navigasi aktivitas dalam aplikasi. **Activity Stack** ini merepresentasikan riwayat aktivitas yang telah dibuka, dengan aktivitas yang paling baru dibuka berada di bagian atas tumpukan. Memahami tumpukan aktivitas sangat penting untuk mengontrol perilaku tombol kembali dan mengelola siklus hidup aktivitas.

Flutter Navigation

Activity Stack





Flutter Navigation

Mekanisme Activity Stack

Berikut ini adalah mekanisme **Activity Stack**

- ▶ Memulai aktivitas dan mendorongnya ke tumpukan
- ▶ Menavigasi kembali dan memunculkan aktivitas dari tumpukan
- ▶ Mengelola tumpukan belakang secara manual
- ▶ Mode peluncuran dan afinitas tugas



Flutter Navigation

Mekanisme Activity Stack

- ▶ **Memulai aktivitas dan mendorongnya ke tumpukan**
 - ▶ Ketika aktivitas baru dimulai, aktivitas tersebut didorong ke bagian atas tumpukan aktivitas. Aktivitas sebelumnya tetap berada di bawahnya dalam tumpukan, mempertahankan statusnya dan memungkinkan pengguna untuk menavigasi kembali ke sana.
- ▶ **Menavigasi kembali dan memunculkan aktivitas dari tumpukan**
 - ▶ Ketika pengguna menekan tombol kembali, aktivitas saat ini akan dikeluarkan dari tumpukan, dan aktivitas sebelumnya dalam tumpukan menjadi aktivitas yang aktif.



Flutter Navigation

Mekanisme Activity Stack

- ▶ **Mengelola tumpukan belakang secara manual**
 - ▶ Meskipun perilaku default dari tombol kembali sering kali sudah cukup, programmer juga dapat mengelola tumpukan aktivitas secara manual untuk kontrol yang lebih besar atas alur navigasi.
- ▶ **Mode peluncuran dan afinitas tugas**
 - ▶ Mode peluncuran menentukan bagaimana contoh baru aktivitas dibuat dan ditempatkan di tumpukan. Afinitas tugas memungkinkan programmer menentukan bagaimana aktivitas dari aplikasi yang berbeda dikelompokkan bersama. Aktivitas dengan afinitas tugas yang sama ditempatkan di tugas yang sama, yang dapat memiliki tumpukan aktivitas terpisah.



Flutter Navigation

Manfaat Activity Stack

Dengan mengelola activity stack, programmer dapat mengontrol alur navigasi, menangani perilaku tombol kembali, dan menciptakan pengalaman multi-layar yang kompleks dalam aplikasi Android programmer. Penting untuk memahami cara kerja activity stack untuk mendesain antarmuka pengguna yang intuitif dan mulus.

Cara melakukan mekanisme ini:

- ▶ Intent → Tradisional
- ▶ Navigator → Flutter



Flutter Navigation

Metode Intent

Kelas **Intent** menyediakan berbagai metode untuk menavigasi di antara berbagai aktivitas di dalam aplikasi. Metode-metode ini memungkinkan programmer untuk memulai aktivitas baru, mengoper data di antara aktivitas, dan mengontrol aliran navigasi.

```
Intent intent = new Intent(CurrentActivity.this, TargetActivity.class);
startActivity(intent);
```

```
// Inside the activity class
finish();
```



Flutter Navigation

Metode Navigator

Kelas Navigator menyediakan metode untuk menavigasi di antara layar atau rute yang berbeda di dalam aplikasi. Kelas ini mengelola tumpukan rute dan memungkinkan programmer untuk mendorong rute baru ke dalam tumpukan, mengeluarkan rute dari tumpukan, dan mengganti rute yang sudah ada.

Cara ini menggunakan konsep yang sama namun metode yang berbeda dengan mode tradisional. Flutter menggunakan istilah **Route** untuk menavigasikan layar aplikasi



Flutter Navigation

Metode Navigator

Memulai Page

```
Navigator.push(  
    context,  
    MaterialPageRoute(builder: (context) => NewScreen()),  
);
```

Mengakhiri Page

```
Navigator.pop(context);
```



Flutter Navigation

Metode Navigator - Arah Navigasi

► **Lateral Navigation**

- ▶ Perpindahan antar layar pada tingkat hierarki yang sama. Komponen navigasi utama aplikasi harus menyediakan akses ke semua tujuan di tingkat teratas hierarkinya.

► **Forward Navigation**

- ▶ Perpindahan antar layar pada tingkat hierarki yang berurutan, langkah-langkah dalam alur, atau di seluruh aplikasi.

► **Reverse Navigation**

- ▶ Bergerak mundur melalui layar baik secara kronologis atau secara hirarkis



Flutter Navigation

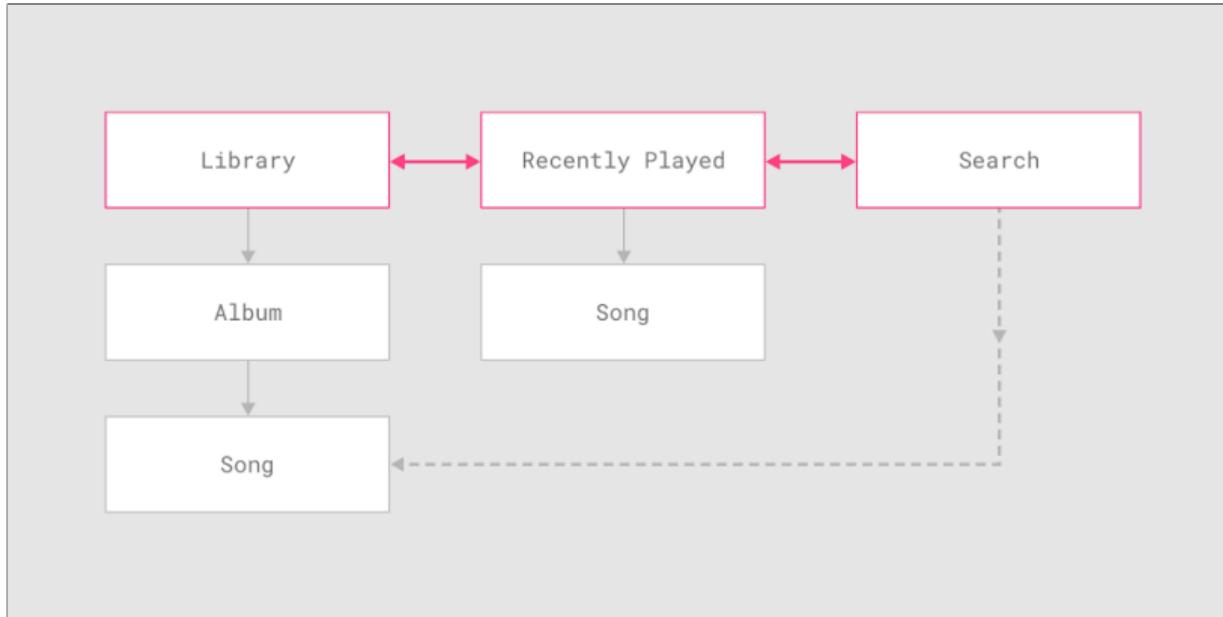
Metode Navigator - Lateral Navigation

Navigasi lateral mengacu pada pergerakan antar layar pada tingkat hierarki yang sama. Navigasi ini memungkinkan akses ke tujuan dan fitur aplikasi yang berbeda, atau berputar di antara item yang terkait dalam satu set.

Komponen navigasi utama aplikasi harus menyediakan akses ke semua tujuan di tingkat teratas hierarkinya. Aplikasi dengan dua atau lebih tujuan tingkat atas dapat menyediakan navigasi lateral melalui laci navigasi, bilah navigasi bawah, atau tab.

Flutter Navigation

Metode Navigator - Lateral Navigation





Flutter Navigation

Metode Navigator - Forward Navigation

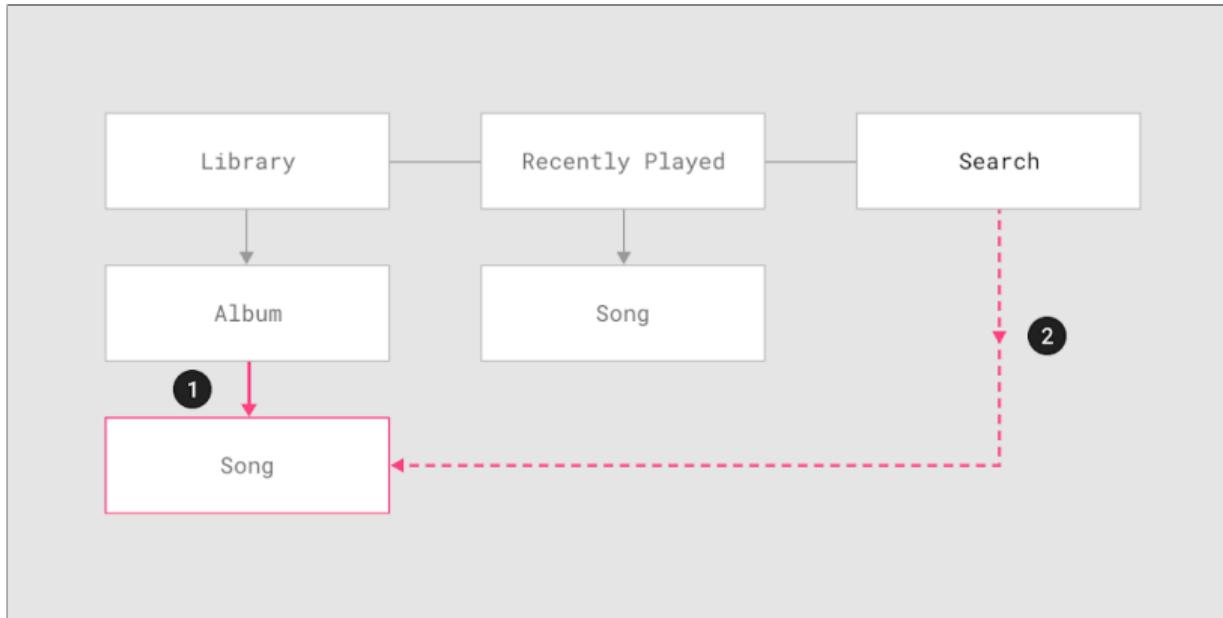
Sementara navigasi lateral menggunakan komponen navigasi khusus, navigasi maju sering kali disematkan ke dalam konten layar melalui berbagai komponen.

Navigasi maju dapat diimplementasikan menggunakan:

- ▶ Wadah konten seperti kartu, daftar, atau daftar gambar
- ▶ Tombol yang maju ke layar lain
- ▶ Pencarian dalam aplikasi pada satu layar atau lebih
- ▶ Tautan di dalam konten

Flutter Navigation

Metode Navigator - Forward Navigation





Flutter Navigation

Metode Navigator - Reverse Navigation

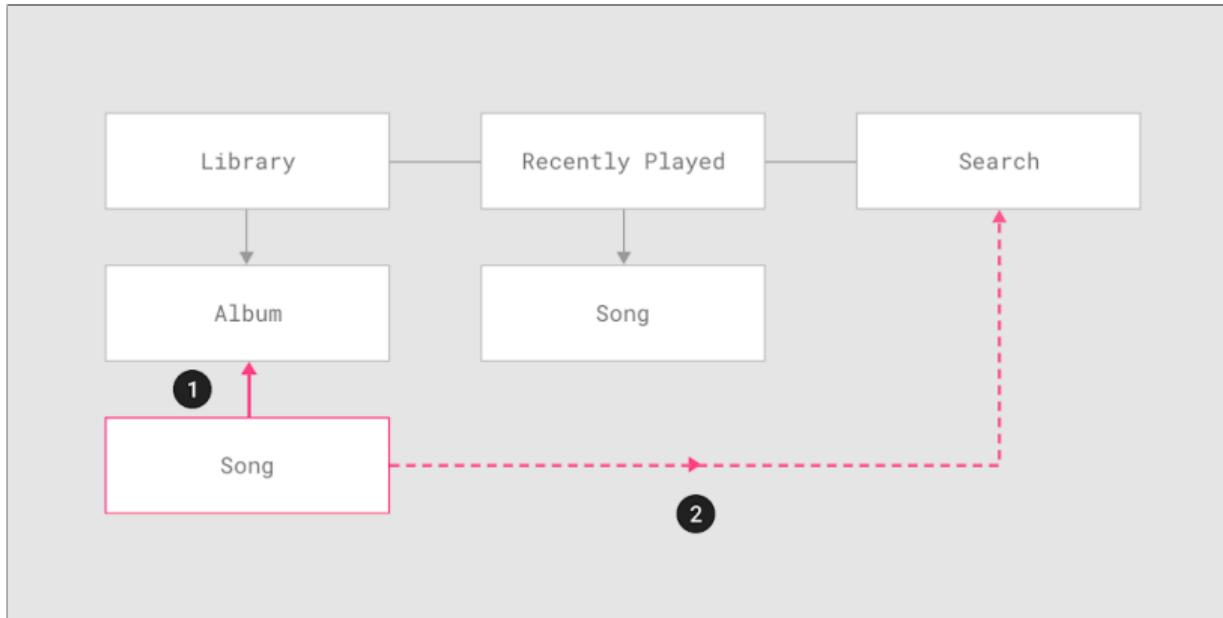
Navigasi mundur mengacu pada gerakan mundur di antara layar. Navigasi ini dapat memindahkan pengguna secara kronologis melalui riwayat layar terakhir mereka, atau ke atas melalui hierarki aplikasi.

Navigasi kronologis terbalik mengacu pada navigasi dalam urutan terbalik melalui riwayat pengguna dari layar yang terakhir dilihat.

Navigasi ke atas memungkinkan pengguna untuk menavigasi satu tingkat ke atas dalam satu hierarki aplikasi, hingga layar beranda atau layar tingkat atas aplikasi tercapai

Flutter Navigation

Metode Navigator - Reverse Navigation



THANK YOU

YOU