LAPORAN RESMI

MODUL V

NAVIGATION

PEMROGRAMAN BERGERAK

NAMA	**: AGUSTIN QURROTA A'YUN**
N.R.P	**: 210441100089**
DOSEN	**: ACHMAD ZAIN NUR, s.Kom.,M.T.**
ASISTEN	**: ALFIAN MAHENDRA IFANDIA**
TGL PRAKTIKUM	**: 15 Mei 2023**

Disetujui : 22 Mei 2023
Asisten

ALFIAN MAHENDRA IFANDIA
19.04.411.00158

LABORATORIUM BISNIS INTELIJEN SISTEM

PRODI SISTEM INFORMASI

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS TRUNOJOYO MADURA

BAB I
PENDAHULUAN

1.1 Latar Belakang

Navigasi dalam pengembangan aplikasi Android dengan Kotlin mengacu pada kemampuan untuk berpindah antara layar atau fragmen yang berbeda dalam aplikasi. Ini melibatkan penggunaan tindakan atau peristiwa tertentu yang memicu perubahan tampilan atau fragmen yang ditampilkan kepada pengguna. Navigasi menggunakan konsep grafik yang terdiri dari tujuan (destinations) dan aksi (actions) antara tujuan-tujuan tersebut.

Berikut adalah beberapa istilah penting terkait navigasi di Android Studio; Destinations, actions, NavGraph, NavController, dan NavHostAda beberapa keuntungan ketika menggunakan komponen Navigation pada Kotlin dalam pengembangan aplikasi Android, diantaranya yaitu, Pemisahan logika navigasi, navigasi tedefinisi secara deklaratif, mengelola back stack, dan animasi transisi. Dalam Android Studio, Anda dapat membuat dan mengelola grafik navigasi dengan menggunakan Navigation Editor. Ini adalah editor visual yang memungkinkan Anda untuk menambahkan destinasi, menghubungkannya dengan aksi, dan mengatur alur navigasi dengan mudah.

NavGraph adalah komponen penting dalam komponen Navigation pada Kotlin. Ini adalah representasi grafis dari struktur navigasi aplikasi Anda. NavGraph mendefinisikan tujuan (destinasi) dan tindakan (action) yang tersedia dalam aplikasi Anda. Ini memungkinkan Anda untuk mendefinisikan alur navigasi dan hubungan antara itu. Komponen Navigation Architecture Component di Android Studio memungkinkan pengembang untuk membuat dan mengelola alur navigasi dengan mudah.

1.2 Tujuan

- Mahasiswa dapat membuat aplikasi dengan menggunakan Naigation.
- Mahasiswa dapat membuat aplikasi dengan NavGraph.

BAB II
DASAR TEORI

2.1 Navigation

Navigation pada Android Studio merujuk pada pengelolaan transisi antara tampilan (fragment atau aktivitas) dalam aplikasi Android. Komponen Navigation Architecture Component di Android Studio memungkinkan pengembang untuk membuat dan mengelola alur navigasi dengan mudah.

Navigasi menggunakan konsep grafik yang terdiri dari tujuan (destinations) dan aksi (actions) antara tujuan-tujuan tersebut. Berikut adalah beberapa istilah penting terkait navigasi di Android Studio:

1. Destinations: Destinasi adalah tampilan yang ditampilkan kepada pengguna dalam aplikasi, seperti fragment atau aktivitas. Setiap destinasi memiliki ID unik dan dapat memiliki atribut seperti label, ikon, atau argumen yang diperlukan.

2. Actions: Aksi adalah penghubung antara destinasi-destinasi dalam grafik navigasi. Aksi menggambarkan bagaimana pengguna dapat berpindah dari satu destinasi ke destinasi lainnya. Misalnya, dapat ada aksi untuk mengarahkan pengguna dari halaman A ke halaman B ketika suatu tombol ditekan.

3. NavGraph: NavGraph (Grafik Navigasi) adalah representasi visual dari alur navigasi dalam aplikasi. Ini berisi daftar destinasi dan aksi yang menghubungkannya. Anda dapat membuat NavGraph dengan menggunakan editor visual di Android Studio atau dengan menulis kode XML.

4. NavController: NavController adalah objek yang mengelola navigasi antara destinasi-destinasi dalam aplikasi. Anda dapat menggunakan NavController untuk melakukan navigasi antara destinasi dengan menggunakan metode seperti **navigate()** atau **popBackStack()**.

5. NavHost: NavHost adalah wadah tempat destinasi-destinasi ditampilkan kepada pengguna. Ini biasanya merupakan bagian dari tampilan (seperti fragment atau tampilan aktivitas) yang menampilkan grafik navigasi.

Dalam Android Studio, Anda dapat membuat dan mengelola grafik navigasi dengan menggunakan Navigation Editor. Ini adalah editor visual yang memungkinkan Anda untuk menambahkan destinasi, menghubungkannya dengan aksi, dan mengatur alur navigasi dengan mudah. Anda juga dapat mengkonfigurasi tindakan seperti animasi transisi antara destinasi dan membaca argumen yang dikirim saat navigasi.

2.3 NavGraph

NavGraph adalah representasi visual dari alur navigasi dalam aplikasi Android. NavGraph berisi daftar destinasi dan aksi yang menghubungkannya. Dalam komponen Navigation Architecture Component di Android Studio, NavGraph dapat dibuat menggunakan editor visual atau dengan menulis kode XML.NavGraph menggambarkan bagaimana pengguna dapat berpindah dari satu destinasi ke destinasi lainnya dalam aplikasi. Ini mencakup semua destinasi yang ada dalam aplikasi, seperti fragment atau aktivitas, dan hubungan antara destinasi-destinasi tersebut melalui aksi.

Di dalam NavGraph, setiap destinasi memiliki ID unik dan dapat memiliki atribut seperti label, ikon, atau argumen yang diperlukan. Aksi dalam NavGraph menggambarkan interaksi yang memungkinkan pengguna untuk berpindah antara destinasi-destinasi tersebut. Misalnya, aksi dapat berupa tombol yang mengarahkan pengguna dari halaman A ke halaman B.NavGraph memudahkan pengembang dalam mengatur alur navigasi dalam aplikasi. Dengan menggunakan NavGraph, pengembang dapat dengan mudah menentukan tampilan mana yang harus ditampilkan kepada pengguna berdasarkan tindakan atau keadaan tertentu. Ini membantu dalam memisahkan logika navigasi dari tampilan individu, sehingga membuat pengembangan dan pemeliharaan aplikasi lebih terorganisir dan mudah dikelola.

BAB III

TUGAS PENDAHULUAN

3.1 Soal

1. Apa itu navigasi dalam mengembangkan aplikasi android dengan kotlin?
2. Apa keuntungan menggunakan komponen navigation pada kotlin?
3. Bagaimana cara melakukan navigasi antara (destinasi) menggunakan komponen navigation pada kotlin?
4. Bagaimana cara mengimplementasikan button navigat dg komponen nav pd kotlin?
5. Apa peran NewGraph dalam komponen Navgraph pada kotlin?

3.2 Jawab

1. mengacu pada kemampuan u/ berpindah antara fragment yg berbeda dalam aplikasi. melibatkan peristiwa tertentu yang memicu perubahan tampilan.
2. pemisahan logika navigasi, terdefinisi secara deklaratif, mengelola, back stack, dan animasi transisi.
3. - Tambakan NavHost Fragment ketampilan utama aplikasi pada file layout.
   - Buat object NavController untuk melakukan navigasi dengan memanggil metode seperti navigate() dg menyediakan ID tujuanga
   - dapat menggunakan tindakan (action) yg didefinisi dlm gravif navigasi untuk navigasi yang kompleks
4. Dalam file xml grafik navigasi, tambahkan tujuan untuk setiap tab di bottom nav, tentukan nav, default untuk tampilan awal ketika apk dimulai, dalam kelas Aktivitas, tambahkan (listener) ke bottom nav. menangani perubahan item yang terpilih.
5. komponen penting dalam komponen Navigation pada kotlin yang mana navgraph sendiri mendefinisikan tujuan dan tindakan (action) yang tersedia dalam aplikasi dan ini memungkinkan untuk mendefinisikan alur navigasi dan hubungan antara apk tersebut.

BAB IV

IMPLEMENTASI

4.1 SoureCode

a. Main Activity

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val fragment =
HomeFragment.newInstance("test1","test2")

bottom_navigation_view.setOnNavigationItemSelectedListener(menu
ItemSelected)
        addFragment(fragment)
    }

    private val menuItemSelected =
BottomNavigationView.OnNavigationItemSelectedListener { item ->
        when (item.itemId) {
            R.id.itemhome ->{
                val fragment =
HomeFragment.newInstance("test1","test2")
                addFragment(fragment)
                return@OnNavigationItemSelectedListener true
            }
            R.id.ItemProfile ->{
                val fragment =
ProfileFragment.newInstance("test1","test2")
                addFragment(fragment)
                return@OnNavigationItemSelectedListener true
            }
        }
        false
    }

    private fun addFragment(fragment: Fragment){
        supportFragmentManager
            .beginTransaction()

.setCustomAnimations(com.google.android.material.R.anim.design_
bottom_sheet_slide_in,
com.google.android.material.R.anim.design_bottom_sheet_slide_ou
t)
            .replace(R.id.content,fragment,
fragment.javaClass.getSimpleName())
            .commit()
    }
```

b. HomeFragmen

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
    tools:context=".HomeFragment">

    <!-- TODO: Update blank fragment layout -->
    <ImageView
        android:id="@+id/profile_image"
        android:layout_width="match_parent"
        android:layout_height="170dp"
        android:layout_marginTop="280dp"
        android:src="@drawable/gambar">
    </ImageView>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="Home"
        android:textAlignment="center"
        android:layout_marginTop="200dp"
        android:textColor="@color/black"
        android:textSize="24sp"/>

</FrameLayout>
```

c. ProfileFragmen

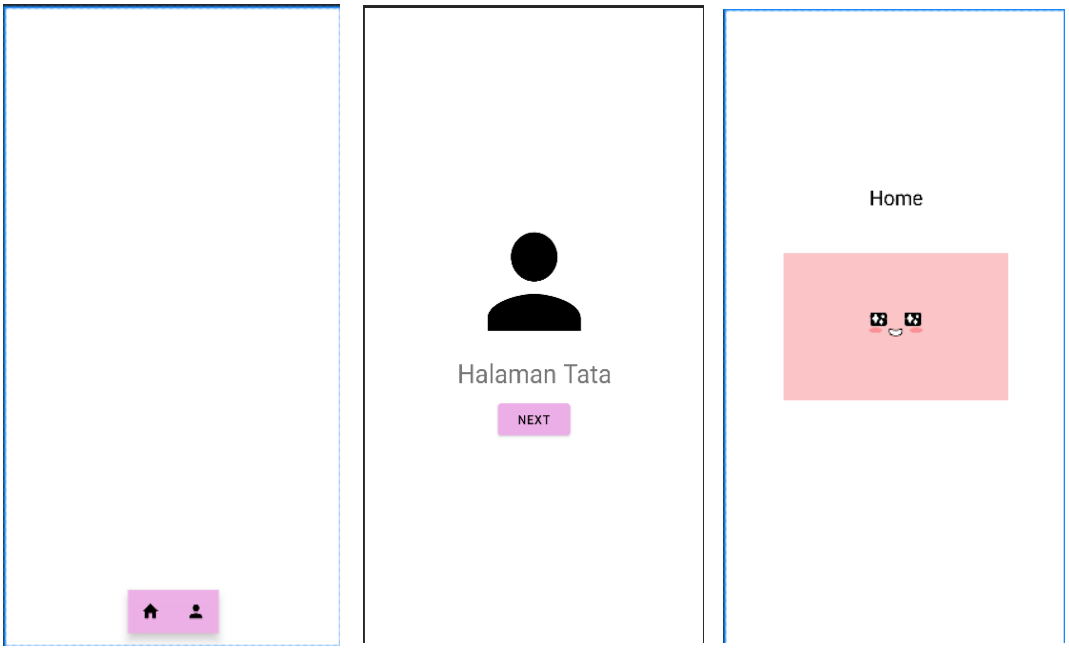
```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ProfileFragment">

    <ImageView
        android:id="@+id/profile_image"
        android:layout_width="match_parent"
        android:layout_height="170dp"
        android:layout_marginTop="230dp"
        android:src="@drawable/baseline_person_24">
    </ImageView>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="Halaman Tata"
        android:textAlignment="center"
        android:layout_marginTop="400dp"
        android:textSize="30sp"/>
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="450dp"
        android:backgroundTint="#FAECAEE6"
        android:text="next"
        android:textColor="@color/black"/>

</FrameLayout>
```

4.2 Hasil



pada gambar diatas dapat menunjukkan navigation dengan 2 Fragment

BAB V
PENUTUP

5.1 Analisa

Dari hasil praktikum, praktikan menganalisa bahwa di dalam Android Studio, praktiikan dapat membuat dan mengelola grafik navigasi dengan menggunakan Navigation Editor. Navigation Editor adalah editor visual yang memungkinkan Praktikan untuk menambahkan destinasi, menghubungkannya dengan aksi, dan mengatur alur navigasi dengan mudah. Praktikan juga dapat mengkonfigurasi tindakan seperti animasi transisi antara destinasi dan membaca argumen yang dikirim saat navigasi.

Dengan menggunakan NavGraph, pengembang dapat dengan mudah menentukan tampilan mana yang harus ditampilkan kepada pengguna berdasarkan tindakan atau keadaan tertentu. Ini membantu dalam memisahkan logika navigasi dari tampilan individu, sehingga membuat pengembangan dan pemeliharaan aplikasi lebih terorganisir dan mudah dikelola. NavGraph adalah komponen penting dalam komponen Navigation pada Kotlin. Ini adalah representasi grafis dari struktur navigasi aplikasi Anda. NavGraph mendefinisikan tujuan (destinasi) dan tindakan (action) yang tersedia dalam aplikasi Anda. Ini memungkinkan Anda untuk mendefinisikan alur navigasi dan hubungan antara itu

5.2 Kesimpulan

Dari hasil praktikum, praktikan menyimpulkan bahwa, Navigasi dalam pengembangan aplikasi Android dengan Kotlin mengacu pada kemampuan untuk berpindah antara layar atau fragmen yang berbeda dalam aplikasi. Ini melibatkan penggunaan tindakan atau peristiwa tertentu yang memicu perubahan tampilan atau fragmen yang ditampilkan kepada pengguna. Di dalam NavGraph, setiap destinasi memiliki ID unik dan dapat memiliki atribut seperti label, ikon, atau argumen yang diperlukan. Aksi dalam NavGraph menggambarkan interaksi yang memungkinkan pengguna untuk berpindah antara destinasi-destinasi tersebut