BAB IV
NAVIGATION BAR

1. Install Library

Pada materi kali ini membutuhkan beberapa library yang perlu diinstall untuk melakukan proses perpindahan halaman dari satu komponen ke komponen lain. Library dibawah ini merupakan library dasar untuk melakukan proses pindah komponen, silakan anda tambahkan library tersebut kedalam project react expo cli anda:

```
npm install @react-navigation/native @react-navigation/native-stack

npx expo install react-native-gesture-handler

npx expo install react-native-screens react-native-safe-area-context
```

Setelah berhasil nemambahkan library tersebut silakan anda jalankan Expo Server anda dengan benar.

2. Stack

Stack navigasi menyediakan cara bagi aplikasi anda untuk melakukan transisi antar layar, setiap komponen yang didaftarkan oleh Stack sifatnya akan bertumpuk layar. Jika memanggil komponen satu ke komponen lain maka komponen berikutnya akan berada di atas layar komponen sebelumnya. Berikut ini adalah contoh penggunaan STACK dengan React Function Component.
Bukalah file App.js anda lalu tambahkan script diatas untuk mendaftarkan komponen-komponen yang menggunakan STACK:

App.js

```jsx
import { NavigationContainer } from "@react-navigation/native";
import {createNativeStackNavigator} from '@react-navigation/native-stack';
import { Text, TouchableOpacity, View } from "react-native";

export default function App() {

  const Stack = createNativeStackNavigator();

  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen
          name="Page1"
          component={Page1}
          options={{ title: "Welcome" }}
        />
        <Stack.Screen name="Page2" component={Page2} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

Pada file App.js diatas pertama anda harus memanggil file library `@react-navigation/native` dan `@react-navigation/native-stack` . Buatlah sebuah variable bernama Stack dimana variable ini memiliki value `createNativeStackNavigator()`. Variable inilah yang nantinya digunakan untuk mendaftarkan setiap Komponen React yang telah dibuat.

Note:
`<NavigationContainer>` → pembungkus untuk mendefinisikan sebuah tumpukan layer navigasi yang berasal berbagai komponen react.
`<Stack.Navigator>` → pembungkus untuk menginisialisasi komponen-komponen layar apa saja yang dapat digunakan. Ini sama halnya dengan mendefiniskan sebuah root end-poin pada sebuah website.
`<Stack.Screen>` → inisialisasi komponen react (RFC/RCC) sebagai bagian dari layar yang akan dibuat tumpukan antar muka.

Pada `Stack.Screen` memiliki beberapa properties umum yang dapat digunakan, yaitu:

Properties	Value
name	Diperuntukan untuk penaman root pada layar
component	Memanggil file komponen react (RCC/RFC)
options	Berisi configurasi untuk title layer atau untuk aksi gesture dan masih banyak lagi. `{title:"", gestureEnabled :boolean}`

Berikut ini adalah contoh react function component untuk Page1 dan Page2 untuk menerapkan proses transisi antar layar:

Parameter **navigation** ditambahkan jika anda ingin melakukan transisi antar layar

```
const Page1 = ({navigation}) => {
  return (
    <TouchableOpacity onPress={() => navigation.navigate('Page2', {name: 'Febry'})} >
      <View style={{ backgroundColor:"purple", padding:10, borderRadius:10, margin:10 }}>
        <Text style={{ color:"white", textAlign:"center" }}>Click here to see me</Text>
      </View>
    </TouchableOpacity>
  );
};
```

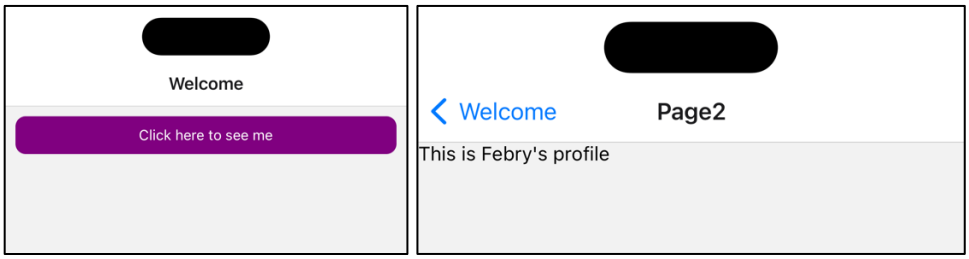
Pada RFC Page1 terdapat sebuah JSX untuk membuat sebuah tombol, jika tombol tersebut ditekan (onPress) maka akan mengeksekusi layar Stack.Screen bernama Page2. Jika anda ingin mengirimkan sebuah parameter kedalam navigasi anda dapat menginisialisasinya dengan format sebuah object:

```
navigation.navigate('name_stack', {object_passing_parameter});
```

Sedangkan pada RFC Page2 dibawah ini hanya menampilkan JSX dalam bentuk texting dan mencoba untuk mengambil nilai parameter yang telah dikirim di Page1 dengan menggunakan parameter bernama route.

```
const Page2 = ({navigation, route}) => {
  return <Text>This is {route.params.name}'s profile</Text>;
};
```

Parameter route digunakan jika anda ingin mengambil passing parameter yang telah dikirimkan, biasanya parameter tersebut akan disimpan kedalam key bernama `params` dan diiringi dengan nama key yang telah dikirim.



Gambar 2. *Contoh transisi antar layar*

3. Bottom Tabs Navigations

Jika anda ingin menggunakan tab bottom navigasi anda perlu menginstall library dibawah ini:

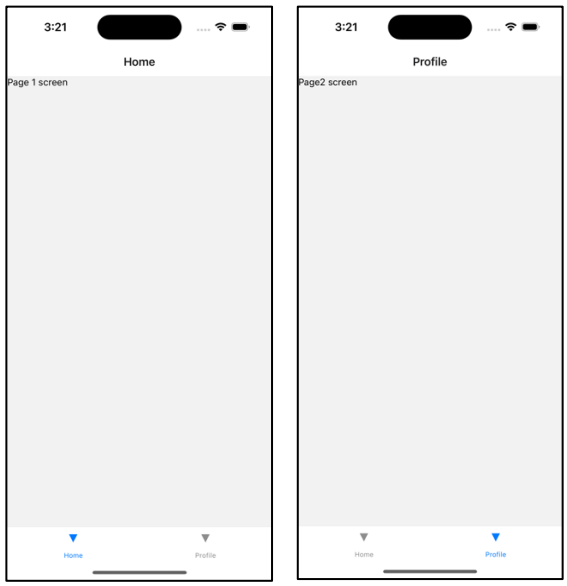
```
npm install @react-navigation/bottom-tabs
```

Berikut ini ialah contoh perpindahan antar layar menggunakan Bottom Tabs Navigations:

```
import React from "react";
import { Text, View } from "react-native";
import { createBottomTabNavigator } from "@react-navigation/bottom-tabs";

export default function  App() {
  const Tab = createBottomTabNavigator();
    return (
      <Tab.Navigator>
        <Tab.Screen name="Home" component={Page1} />
        <Tab.Screen name="Profile" component={Page2} />
      </Tab.Navigator>
    );
}
```

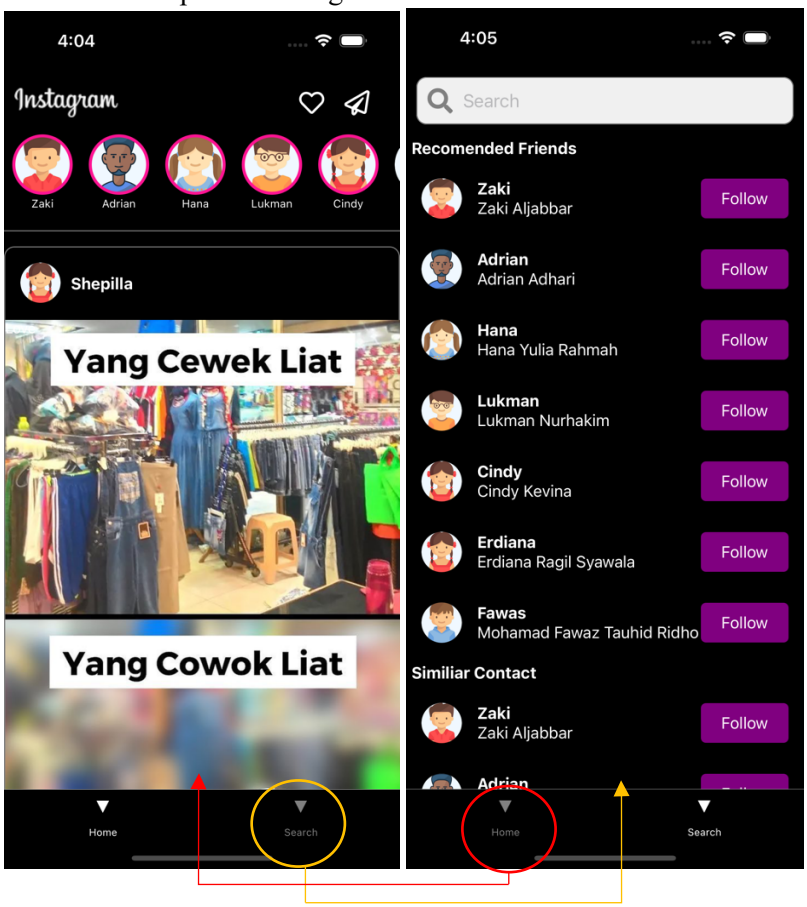
Pada penggunaan perpindahan antar layar kali ini kurang lebih memiliki skema yang sama dengan penggunaan STACK. Namun navigasi ini menggunakan TAB sebagai inisialisasi layar setiap komponennya.



Gambar 3.*Contoh transisi dengan bottom tab navigations*

4. Latihan Praktikum

1. Buatlah package baru bernama praktikum-4.
2. Pada module mengenai Listing, silakan anda masukan halaman-halaman komponen react yang telah dibuat seperti komponen HOME dan MyFriends dengan menggunakan Navigation Bottom Tab seperti contoh gambar dibawah ini:



3. Buatlah sebuah komponen react dimana berisi tampilan halaman SIGN IN, form isian berupa EMAIL dan PASSWORD. Jika mengklik tombol SIGN IN maka akan mengalihkan layer ke HOME seperti gambar pada soal Latihan 2.
   Gunakan navigation STACK untuk komponen LOGIN agar dapat melakukan transisi halaman dimana halaman komponen tersebut bukan bagian dari Bottom Tab Navigation.
   Halaman form SIGN IN hanya sebatas UI tanpa adanya validasi data yang dimasukan.

Pengumpulan tugas Latihan praktikum dikumpulkan kedalam GITHUB masing-masing mahasiswa berdasarkan repository yang telah dibuat PPB-TI-21-[PA/KA]-NPM. File source code disimpan sesuai nama project-praktikum dan masukan kedalam repositori tersebut. Buatkanlah file dokumen dalam bentuk file pdf yang berisi Screen Capture dari hasil program yang telah dikerjakan. Simpan dalam file PDF tersebut kedalam project tersebut.
Tambahkan Collaborator management access pada repository anda kepada:

@FebryFairuz dan *(@IrvanRizkyAriansyah* atau *@thesyamarcella*)

Disusun Oleh	Disetujui Oleh

Thesya Marcella **Penyusun I**	*Irvan Rizky Ariansyah* **Penyusun II**	*Febri Damatraseta Fairuz, S.T., M.Kom* **Dosen Kordinator**