

PEMROGRAMAN BERBASIS MOBILE

POLITEKNIK NEGERI LAMPUNG

PMI 1517

KULIAH MINGGU KE- 7

Dani Rofianto
danirofianto@polinela.ac.id



NAVIGATION

PEMROGRAMAN BERBASIS *MOBILE*

Pengenalan Navigation

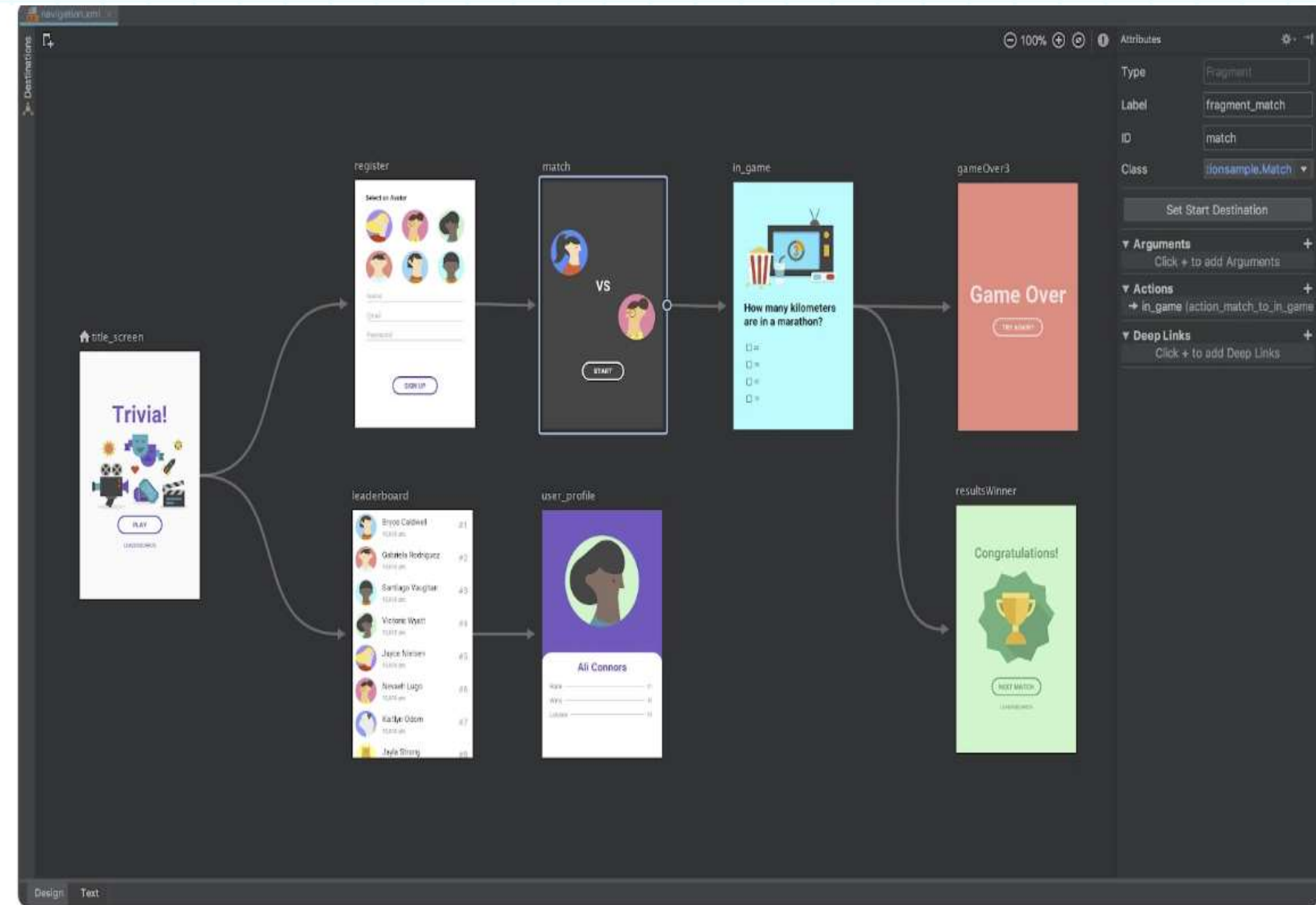
- Dalam mengembangkan sebuah aplikasi, navigation atau navigasi antar screen merupakan aspek penting dalam membangun pengalaman pengguna (user experience) yang baik.

Beberapa komponen navigasi yang akan dibahas dalam modul ini adalah:

1. Navigation Component
2. ActionBar
3. Toolbar
4. Navigation Drawer
5. Bottom Navigation
6. TabLayout

1. Navigation Component

- Navigation Component adalah Komponen yang memiliki fungsi utama yaitu sebagai navigasi antar **Activity** maupun **Fragment** dalam sebuah aplikasi. Bedanya, navigation membuatnya jadi jauh lebih mudah.
- Jika sebelumnya Kita telah mengenal **Intent** untuk navigasi antar activity dan **FragmentManager** untuk bernavigasi antar fragment, sekarang **Navigation** bisa menangani itu semua hanya dalam “beberapa baris kode saja”, termasuk dalam hal pengiriman data, bahkan lebih dari itu.



Manfaat Navigation Component

1. Bisa melihat alur navigasi dengan lebih mudah dengan adanya visualisasi dari Android Studio
2. Menangani perpindahan Fragmen dengan cepat.
3. Menangani navigasi up dan back dengan tepat secara default.
4. Mengatur animasi saat transisi tanpa ribet.
5. Membantu proses saat ingin menerapkan DeepLink.
6. Terdapat NavigationUI, seperti navigation drawer dan bottom navigation.
7. Safe Args, untuk mengirim data tanpa takut adanya error karena data null.

Prinsip Navigation

Ada 5 prinsip Navigation yang perlu ketahui:

1. Setiap aplikasi harus memiliki start destination yang tetap. Sehingga saat menjalankan aplikasi pertama kali selalu menampilkan halaman yang sama.
2. Stack harus menyimpan state saat itu.
3. Tombol Up dan Back harus sesuai saat navigasi.
4. Tombol Up tidak seharusnya menyebabkan keluar dari aplikasi.
5. Seharusnya memiliki alur backstack yang sama baik memakai DeepLink maupun manual.

Bagian-Bagian Navigation

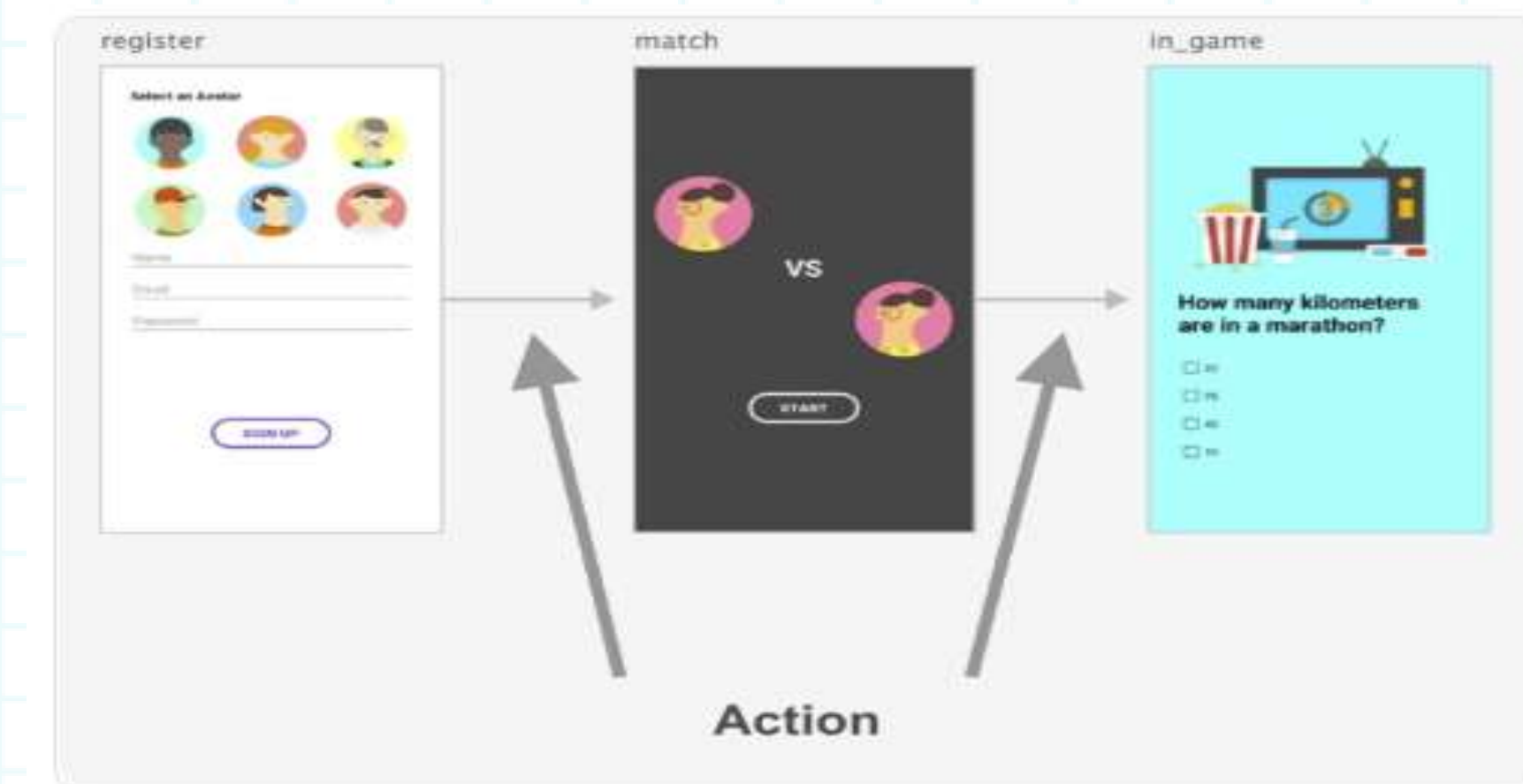
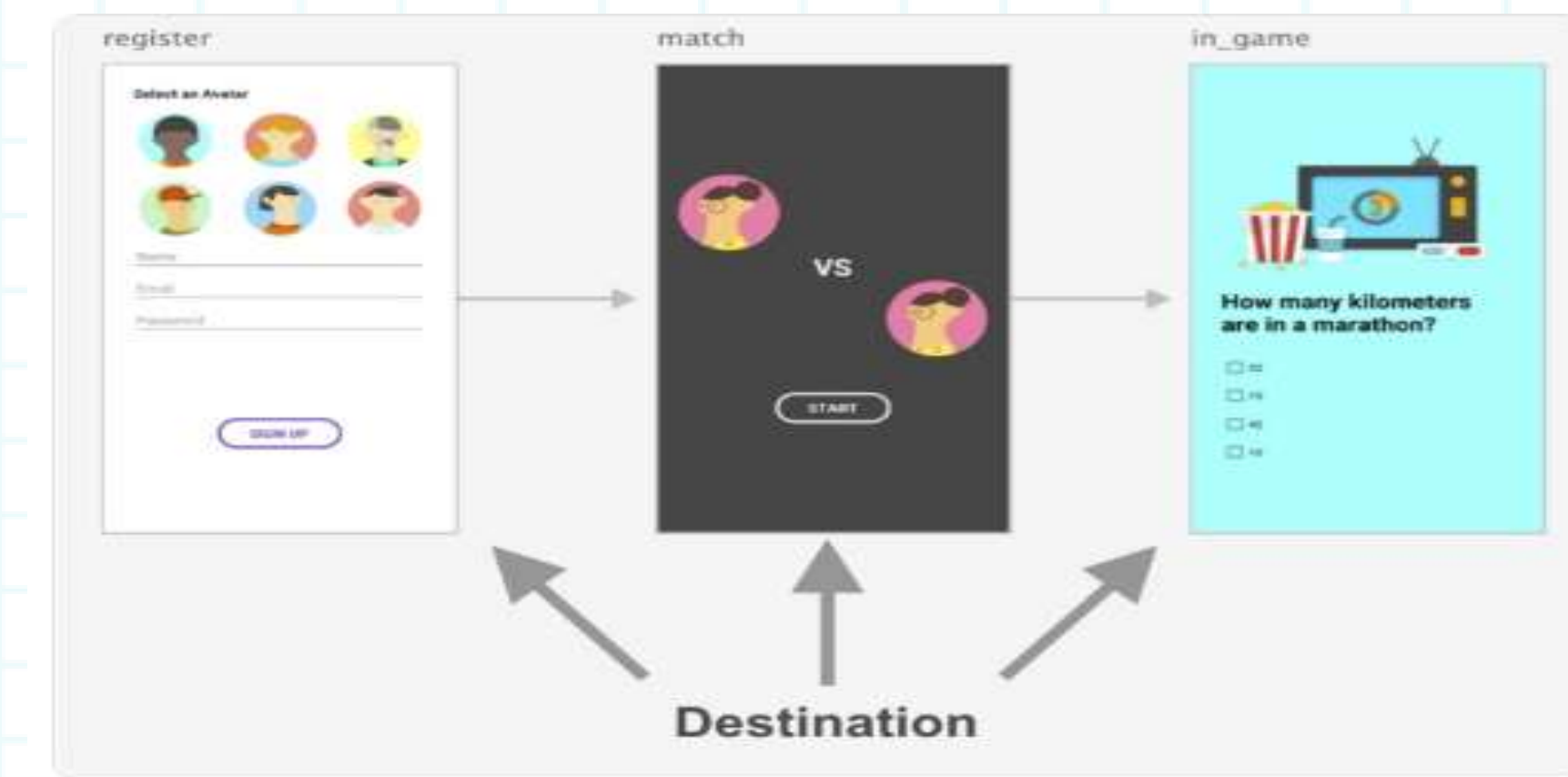
Ada 3 bagian dari Navigation Component yang harus diketahui:

1. Navigation Graph

sebuah kelas resource XML yang letaknya ada di folder res > navigation.

Berkas ini berisi semua data navigasi, yaitu seperti:

- **Destination**, berupa Fragment atau activity yang saling tersambung.
- **Action**, berupa navigasi dengan destination (tujuan) ke fragment mana. Di sini kita juga bisa mengatur animation (menerapkan animasi dari folder anim), pop behaviour (aksi ketika tombol back) dan juga launch option.
- **Argument** yang digunakan untuk mendefinisikan data yang akan dikirim antar fragment.
- Dan juga **DeepLink** untuk bisa membuka suatu fragment dengan menggunakan sebuah URL.



Bagian-Bagian Navigation

2. **NavHostFragment** adalah layout xml dasar yang berfungsi sebagai host (dasar) dari semua halaman yang lain, biasanya terletak di Activity dengan layout yang berisi fragment. Anda cukup menambahkan tiga baris kode ini supaya fragment tersebut menjadi NavHostFragment:

```
<fragment
    android:id="@+id/container"
    android:name="androidx.navigation.fragment.NavHostFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:defaultNavHost="true"
    app:navGraph="@navigation/main_navigation" />
```

3. **NavController** adalah kelas yang akan dipakai untuk melakukan eksekusi navigasi yang sudah dibuat. Berikut adalah contohnya:

```
btnCategory.setOnClickListener(
    Navigation.createNavigateOnClickListener(R.id.action_homeFragment_to_categoryFragment)
);
```


Mengirim data dengan SafeArgs

- Di dalam **Navigation Component**, selain menggunakan bundle, kita juga bisa menggunakan **SafeArgs** untuk mengirim data. SafeArgs adalah salah satu fitur yang disediakan di dalam Navigation Component, di mana kita bisa mengirimkan data tanpa takut adanya nilai null yang menyebabkan aplikasi force close. Hal ini karena SafeArgs bersifat type-safety.
- Untuk memulainya kita perlu menambahkan kode berikut di build.gradle(Project)

```
buildscript {  
    dependencies {  
        classpath "androidx.navigation:navigation-safe-args-gradle-plugin:2.5.1"  
    }  
}
```

- dan di build.gradle(Module:app)

```
plugins {  
    ...  
    id 'androidx.navigation.safeargs'  
}
```

Mengirim data dengan SafeArgs

- Setelah menambahkan plugin SafeArgs, maka Android Studio akan membuat kelas khusus yang berhubungan dengan fragment yang terdaftar di dalam Navigation. Misal di dalam Navigation ada aksi dengan nama `action_to_detail` untuk mengirim data nama dari `HomeFragment` ke `DetailFragment`, maka akan tergenerate dua kelas lagi, yaitu `HomeDirections` dan `DetailFragmentArgs`.
- `HomeDirections` berisi data action dan setter dari fragment tersebut, sehingga kita bisa memasukkan data dengan cara seperti ini:

```
1.HomeDirections.ActionToDetail toDetail =  
HomeDirections.actionToDetail();  
2.toDetail.setName("Habibi");  
3.Navigation.findNavController(view).navigate(toDetail);
```

- Sedangkan `DetailFragmentArgs` berisi bundle yang dapat kita ambil datanya, sehingga kita bisa mengambil data yang dikirimkan dengan cara seperti ini:

```
1.DetailFragmentArgs.fromBundle(getArguments()).getName();
```


DeepLink

DeepLink sering digunakan di aplikasi-aplikasi besar, terutama *marketplace*. Pernahkah Anda mendapatkan broadcast yang berisi tautan, namun ketika dibuka tidak menuju ke browser, namun justru ke sebuah aplikasi, bahkan di detail produknya. Itulah fungsi dari DeepLink, yaitu kita bisa membuat navigasi ke sebuah halaman di dalam aplikasi kita dengan menggunakan URL. Berikut contohnya:



2. ActionBar

Action bar merupakan rancangan yang dianjurkan dalam membuat aplikasi. Bila kita menggunakannya dengan baik, maka kita dapat mempertahankan konsistensi aplikasi. Tentunya ini akan berimbas positif pada user experience.

Beberapa fungsi penting dari **ActionBar** adalah :

1. Memberikan ruangan kepada identitas app dan juga mengindikasikan lokasi pengguna pada saat menjalankan aplikasi.
2. Akses ke beberapa action penting dengan lebih mudah seperti action pencarian.
3. Sebagai navigation dan switch antar view (tampilan).

Anatomi dari ActionBar



1. **Container**, merupakan komponen utama yang mewadahi elemen-elemen di dalamnya, biasanya memiliki ukuran
2. **Navigation Icon**, merupakan ikon yang tampil di sebelah kiri bar. Ia bisa berupa hamburger menu icon untuk membuka Navigation Drawer. Ia juga bisa berupa tombol up arrow untuk kembali ke parent jika ia tidak berada di top-level screen atau back arrow untuk kembali halaman sebelumnya.
3. **Title**
Digunakan untuk menampilkan nama aplikasi ataupun nama halaman.
4. **Action Items**
Tempat untuk meletakkan beberapa action penting dalam sebuah aplikasi.
5. **Overflow Menu**
Tempat untuk meletakkan action yang jarang digunakan.

3.Toolbar

Toolbar merupakan pengganti dari action bar yang mempunyai kontrol dan fleksibilitas lebih. **Action bar** dan **Toolbar** sebenarnya termasuk dalam komponen AppBar. Yang membedakan adalah **ActionBar** akan menampilkan bar menyesuaikan dengan versi Android-nya, jika kalian ingat pada Android versi lama, **action bar** berwarna abu-abu, sedangkan pada Android versi terbaru menggunakan warna dari theme. Sedangkan **Toolbar**, memiliki tampilan yang konsisten di semua versi.

Untuk menggunakan **Toolbar** Anda harus mengatur theme aplikasi menjadi **NoActionBar** terlebih dahulu, jika tidak, akan terdapat dua app bar dalam satu halaman.

Hal menarik lainnya jika menggunakan **Toolbar** yaitu Kita bisa membuat App Bar yang lebih fleksibel dan dapat dimodifikasi sesuai kebutuhan. Berikut adalah contoh modifikasi Toolbar dengan menambahkan logo:



```
1.<androidx.appcompat.widget.Toolbar
2.android:id="@+id/toolbar_main"
3.android:layout_width="match_parent"
4.android:layout_height="wrap_content"
5.android:background="@color/colorPrimary"
6.android:minHeight="?attr/actionBarSize">
7.<ImageView
8.android:id="@+id/toolbar_title"
9.android:layout_width="100dp"
10.android:layout_height="40dp"
11.android:src="@drawable/dicoding_logo"
12.android:layout_gravity="center"/>
13.</androidx.appcompat.widget.Toolbar>
```

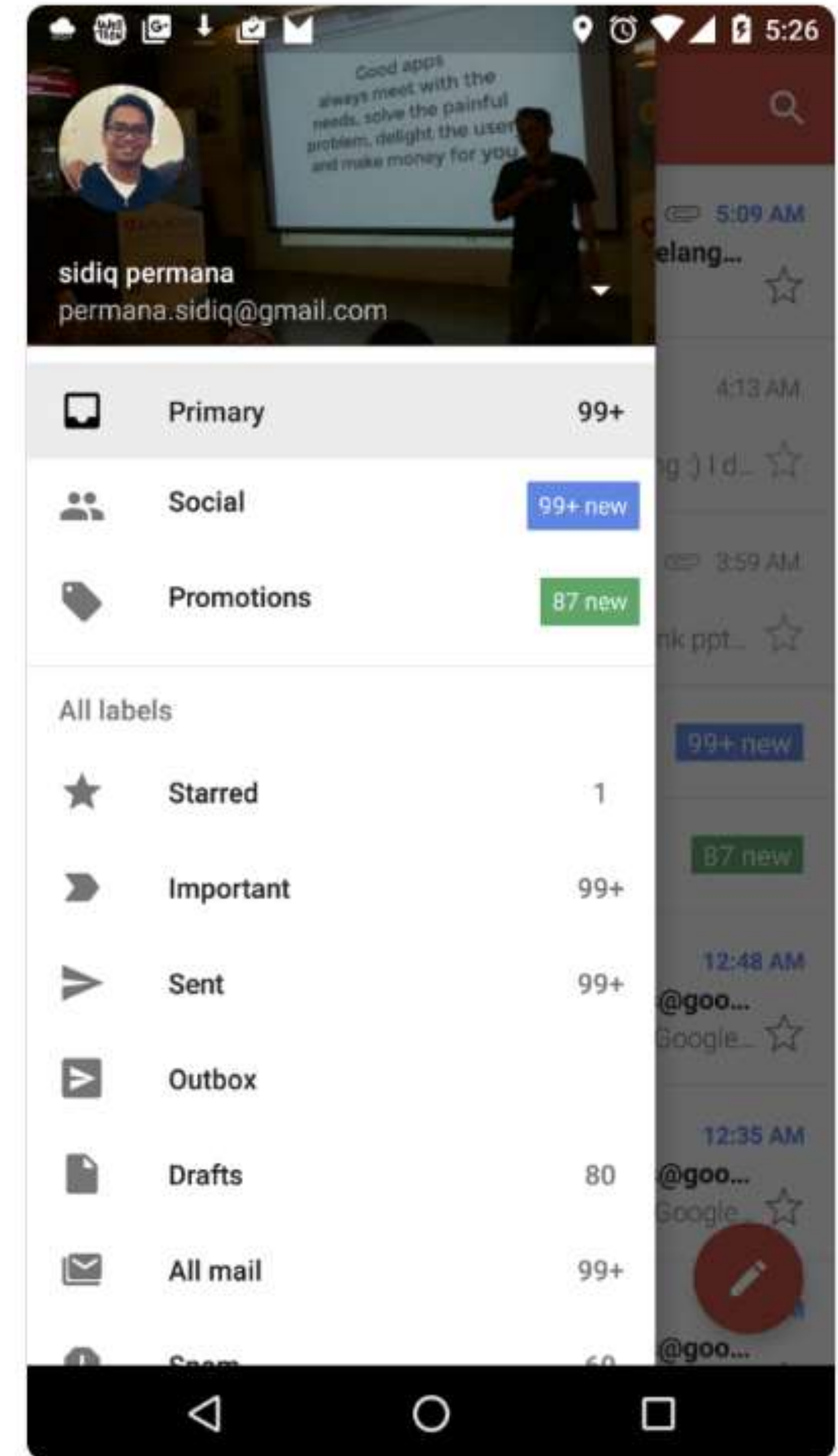
Apakah kita harus meninggalkan **Action Bar** bawaan dari **appbar** dan menggantinya dengan **Toolbar**? Jawabannya adalah tergantung dari kebutuhan Anda sebagai developer. Jika memang fitur dari aplikasi yang Anda buat tidak terpenuhi ketika menggunakan actionbar, maka toolbar adalah pilihan terbaik.

4. Navigation Drawer

- **Navigation drawer** adalah sebuah panel yang berada di sisi kiri layar perangkat Android. Navigation drawer juga akan muncul ketika pengguna menekan ikon hamburger yang berada di sudut kiri panel action bar.
- **Navigation drawer** berisi menu-menu utama dalam sebuah aplikasi. Pada prinsipnya penggunaan navigation drawer sesuai dengan pendekatan yang terdapat pada material design, di mana navigasi utama aplikasi akan diletakkan.

Navigation drawer digunakan bila:

- Aplikasi Anda memiliki top level views, dalam artian beberapa menu utama yang menjadi parent dari beberapa submenu di dalamnya.
- Jika kamu ingin memberikan kecepatan kepada pengguna untuk mengakses menu yang tidak terhubung secara langsung.
- Bila Anda ingin kembali ke halaman utama dengan cepat.
- Jika Anda ingin menempatkan menu yang jarang diakses. Contohnya, term and conditions, about app, legal dan lain sebagainya.



5. Bottom Navigation

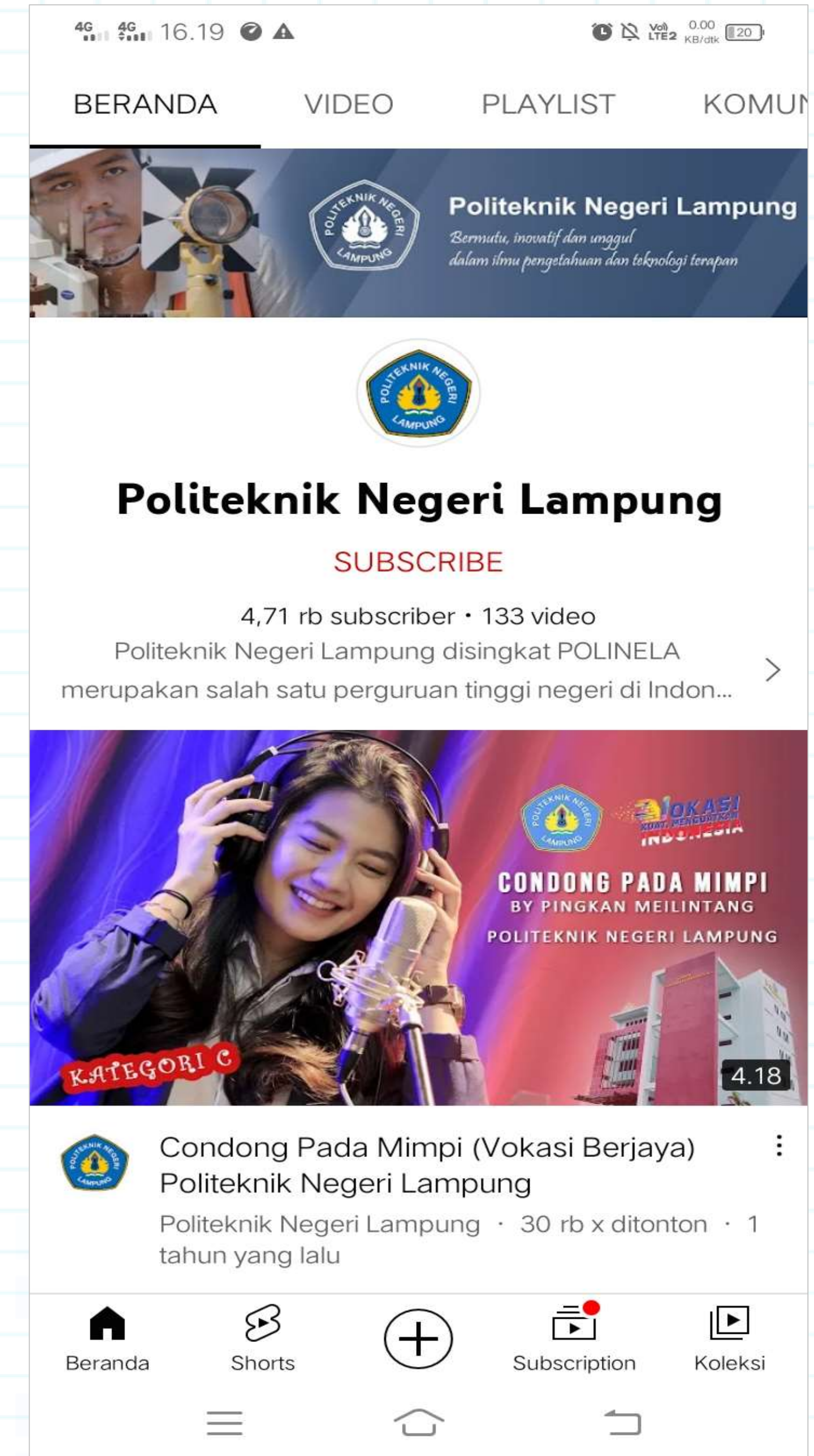
Bottom navigation adalah sebuah baris navigasi yang memungkinkan untuk melakukan perpindahan antar tujuan navigasi dalam suatu aplikasi. Bottom navigation menampilkan tiga hingga lima aksi di bagian bawah layar. Setiap aksi diwakili oleh ikon dan label dengan teks opsional. Ketika ikon pada bottom navigation diketuk, pengguna akan dibawa ke tujuan navigasi atas yang terkait dengan ikon itu.

Bottom navigation digunakan bila:

1. Fitur menu utama harus dapat diakses dari mana saja di aplikasi.
2. Hanya terdiri dari tiga hingga lima menu.
3. Device yang digunakan untuk ponsel atau tablet.

Bottom navigation tidak disarankan penggunaannya apabila :

1. Hanya mempunyai tugas tunggal atau satu menu saja, seperti melihat satu email.
2. Menu yang digunakan berfungsi untuk preferensi atau pengaturan pengguna.



6. TabLayout

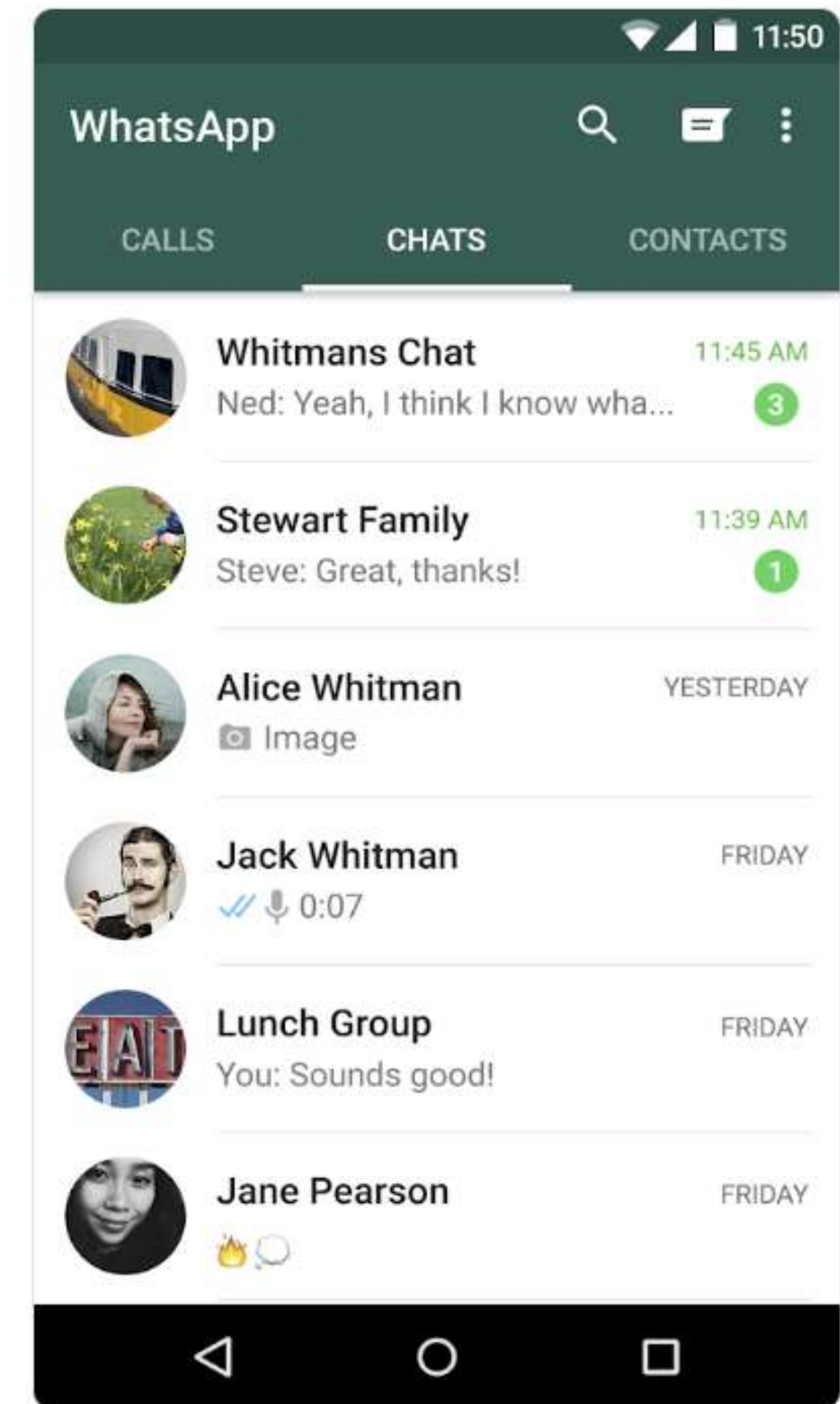
TabLayout adalah sebuah baris navigasi yang memungkinkan untuk melakukan perpindahan antar tujuan navigasi dalam suatu aplikasi. Biasanya TabLayout dikombinasikan dengan ViewPager supaya bisa berpindah antar menu dengan cara swipe ke kanan dan kiri. Bedanya dengan Bottom Navigation yaitu TabLayout letaknya biasanya di atas.

TabLayout digunakan bila:

1. Fitur antar menu memiliki kemiripan atau berhubungan dalam level yang sama.
2. Ingin berpindah antar menu dengan cepat menggunakan swipe.

TabLayout tidak disarankan penggunaannya apabila :

1. Antar menu memiliki level yang berbeda dan tak saling berhubungan.
2. Di dalam menu terdapat fitur yang menggunakan swipe juga, sehingga bertabrakan dengan swipe pada tab.





TERIMA KASIH

PEMROGRAMAN BERBASIS MOBILE