



UNIVERSITAS
BUDI LUHUR

Intents, Fragment and Navigation

Presented By Putri Hayati, S.ST, M.Kom

[Read More](#)



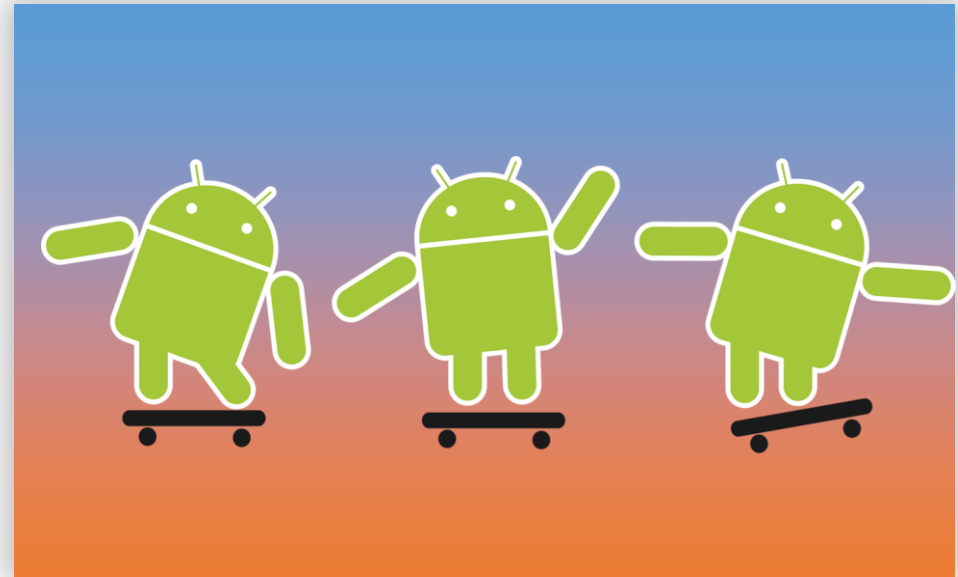
Kategori

3.0 Intents

3.1 Fragment

3.2 Navigation

3.3 Praktikum

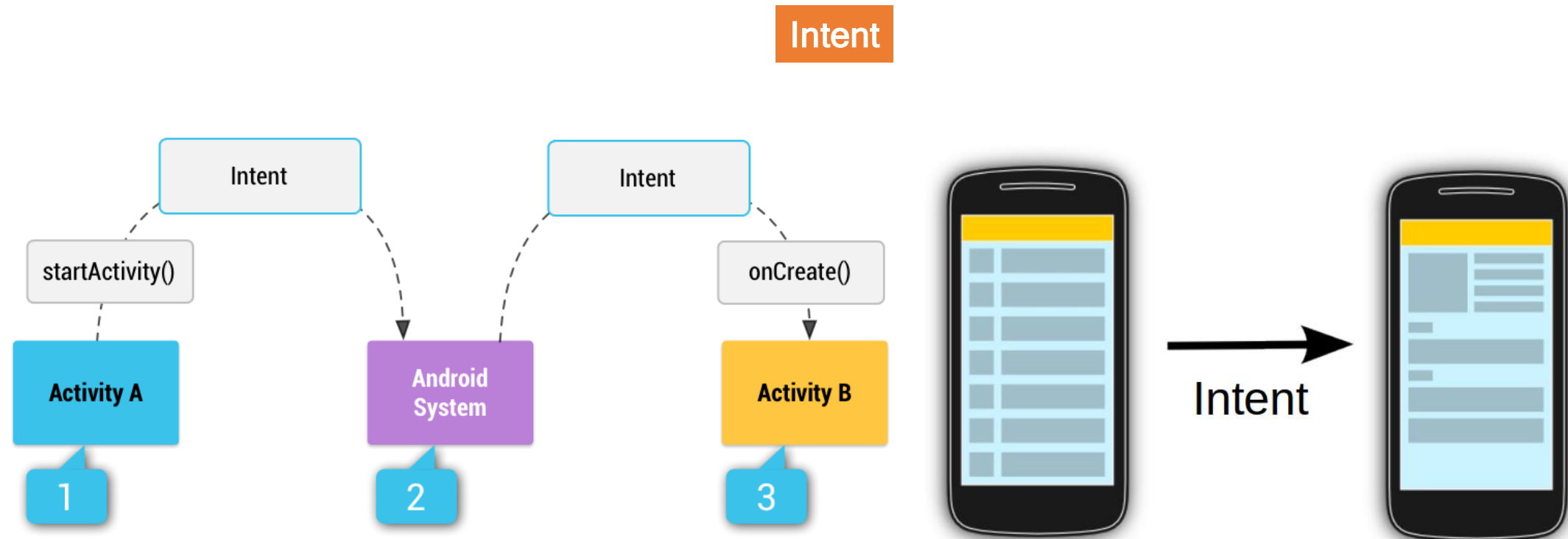




Intent dan Filter Intent



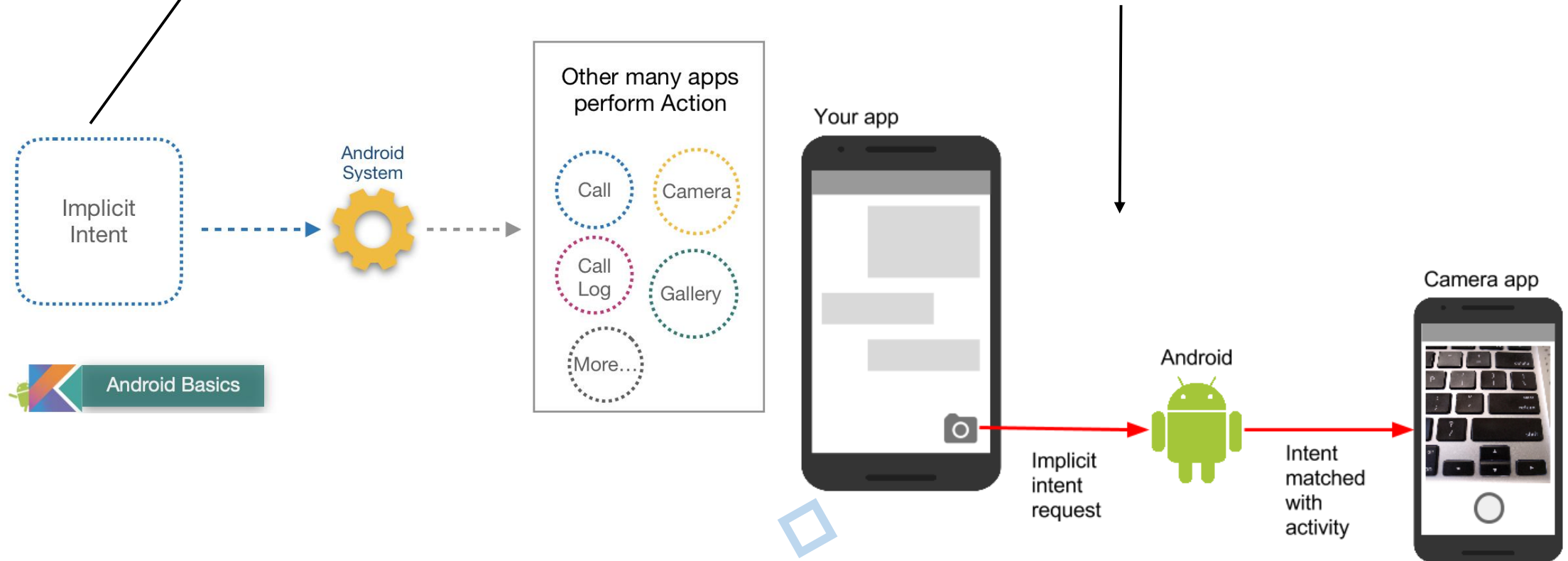
3.0 Intent dan Filter Intent



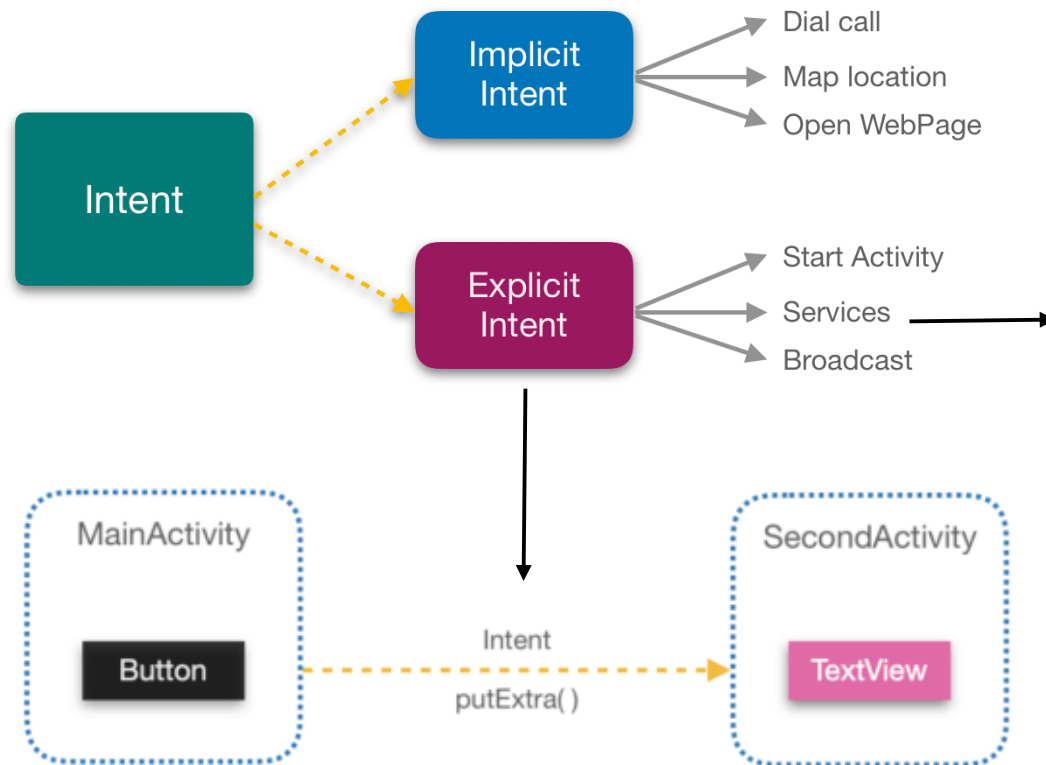
Intent merupakan mekanisme untuk melakukan sebuah *action* dan komunikasi antar komponen aplikasi. Contoh, memiliki sebuah halaman Activity yang terdapat tombol di dalamnya. Lalu tekan tombol tersebut untuk membuka peta, kamera, atau halaman lainnya.

Intent Implicit dan Intent Explicit

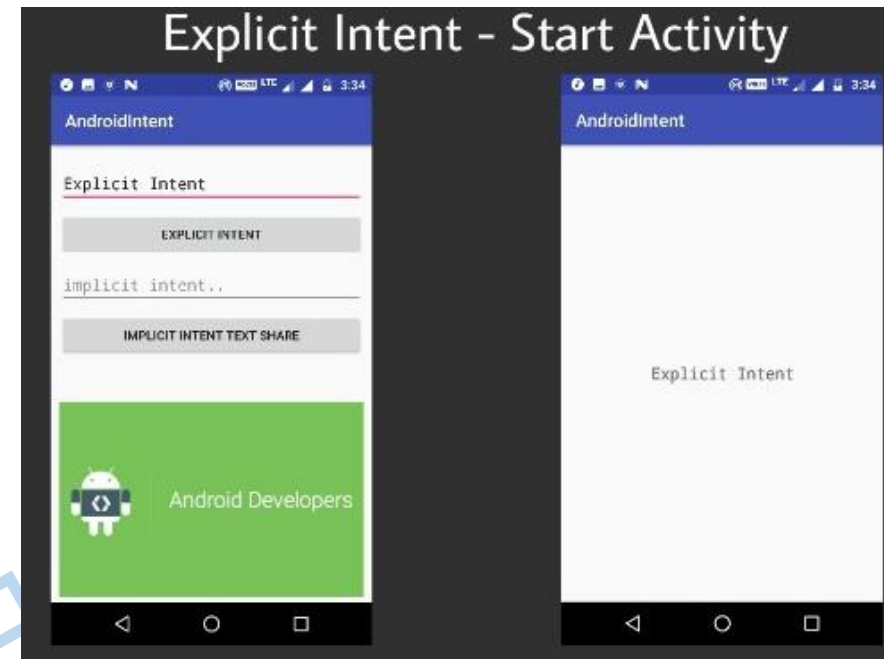
Berfungsi melakukan perpindahan activity (halaman) menuju ke aplikasi internal smartphone.
Contohnya ketika user hendak membuka sebuah kamera.



Intent Implicit dan Intent Explicit



Berfungsi melakukan perpindahan activity (halaman) ke activity (halaman) lainnya. Explicit intent bekerja dengan menggunakan nama kelas yang dituju, misal `com.dicoding.activity.DetailActivity`. Umumnya intent ini digunakan untuk mengaktifkan komponen pada satu aplikasi.



3.0 Intent dan Filter Intent

Intent Object

Component Name

Kolom Intent ini adalah objek **ComponentName**, yang bisa Anda tetapkan menggunakan nama class yang sepenuhnya memenuhi syarat dari komponen target, termasuk nama paket aplikasi, misalnya, **com.example.ExampleActivity**. Anda bisa menyetel nama komponen dengan **setComponent()**, **setClass()**, **setClassName()**, atau dengan konstruktor Intent.

Action

ACTION_VIEW Gunakan tindakan ini dalam intent dengan **startActivity()** misalnya foto yang bisa dilihat dalam aplikasi galeri, atau alamat yang bisa dilihat dalam aplikasi peta. **ACTION_SEND** Juga dikenal dengan intent berbagi, Anda harus menggunakannya dalam intent bersama **startActivity()** jika Anda memiliki data yang bisa digunakan pengguna untuk berbagi melalui aplikasi lain, seperti aplikasi email atau aplikasi berbagi jejaring sosial.

Data

URI (objek Url) yang merujuk data untuk diolah dan/atau tipe MIME dari data tersebut. Tipe data yang disediakan umumnya didikte oleh tindakan intent. Misalnya, jika tindakan merupakan **ACTION_EDIT**, data harus berisi **URI** dari dokumen untuk diedit. Untuk menyetel hanya **URI** data, panggil **setData()**. Untuk menyetel hanya tipe MIME, panggil **setType()**. Jika perlu, Anda bisa menyetel keduanya secara eksplisit dengan **setDataAndType()**.

3.0 Intent dan Filter Intent

Intent Object

Category

String yang berisi informasi tambahan tentang jenis komponen yang harus menangani intent. Keterangan kategori dalam jumlah berapa pun bisa dimasukkan dalam intent, namun sebagian besar intent tidak memerlukan kategori.

`CATEGORY_BROWSABLE` dan `CATEGORY_LAUNCHER`

Extras

opsional dan hanya digunakan jika ada kebutuhan untuk memberikan beberapa informasi tambahan untuk komponen.

Flag

Objek Intent untuk berbagai jenis tugas. Ada berbagai jenis flag yang dapat disetel menggunakan metode `addFlags()` dan metode `setFlags()`.

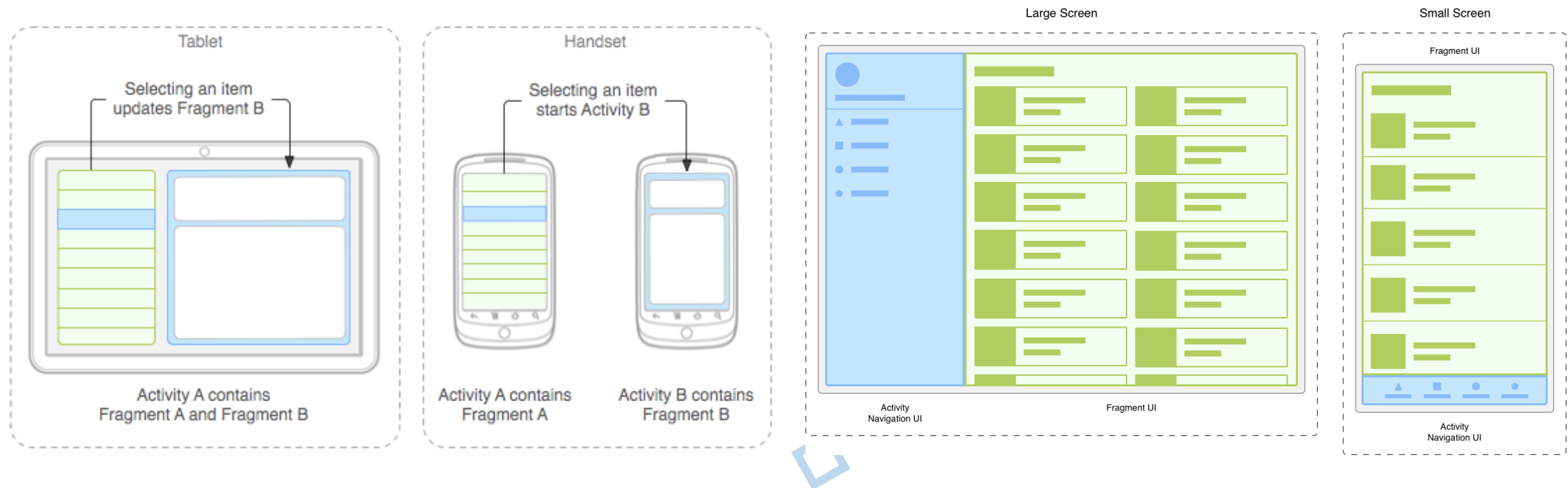


Fragment

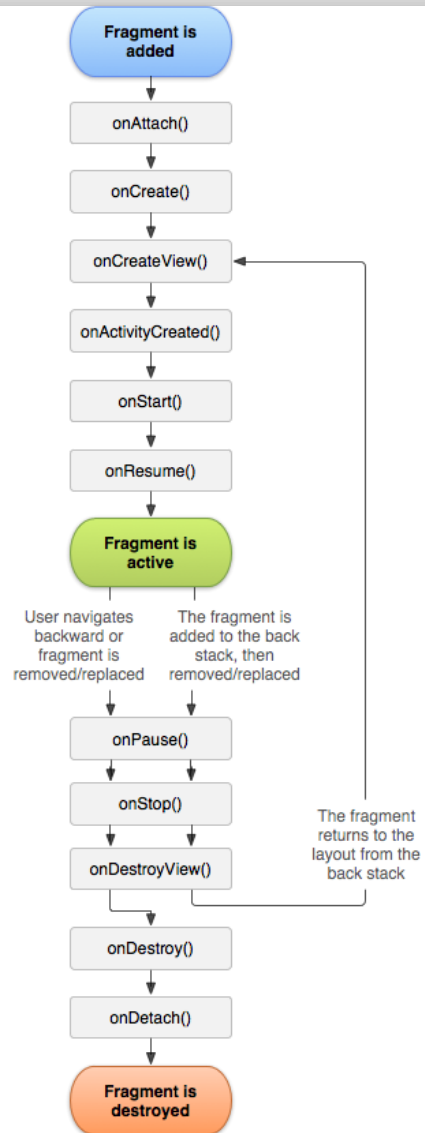


Fragment

Fragment adalah bagian UI aplikasi Anda yang dapat digunakan kembali. Sebuah fragmen menentukan dan mengelola tata letaknya sendiri, memiliki siklus proses sendiri, serta dapat menangani peristiwa inputnya sendiri. Fragmen tidak dapat berjalan sendiri, mereka harus dihosting oleh aktivitas atau fragmen lain. Hierarki tampilan fragmen menjadi bagian dari, atau dilampirkan ke, hierarki tampilan host.



3.1 Fragment



Untuk membuat fragmen, Anda harus membuat subclass Fragment (atau subclass-nya yang ada). Class Fragment memiliki kode yang mirip seperti **Activity**. Class ini memiliki metode callback yang serupa dengan aktivitas, seperti `onCreate()`, `onStart()`, `onPause()`, dan `onStop()`. Sebenarnya, jika Anda mengonversi aplikasi Android saat ini untuk menggunakan fragmen, Anda mungkin cukup memindahkan kode dari metode callback aktivitas ke masing-masing metode callback fragmen.

DialogFragment

- Menampilkan dialog mengambang. Penggunaan class ini untuk membuat dialog merupakan alternatif yang baik dari penggunaan metode helper dialog di class Activity, karena Anda bisa menyatukan dialog fragmen ke dalam back-stack fragmen yang dikelola oleh aktivitas, sehingga pengguna bisa menutup ke fragmen yang ditinggalkan.

ListFragment

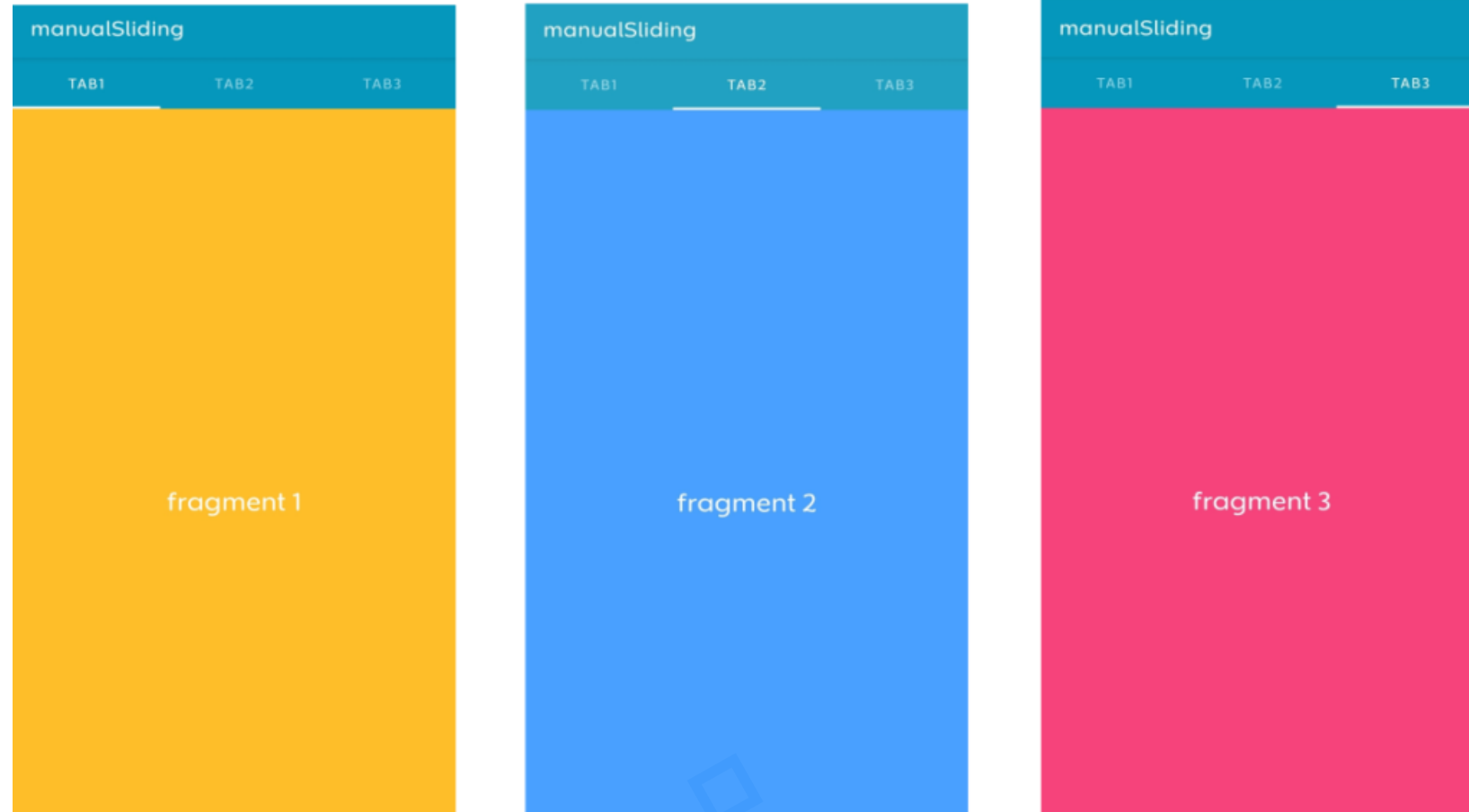
- Menampilkan daftar item yang dikelola oleh adaptor (misalnya **SimpleCursorAdapter**), serupa dengan **ListActivity**. Menampilkan beberapa metode pengelolaan tampilan daftar, seperti callback `onListItemClick()` untuk menangani kejadian klik.

PreferenceFragmentCompat

- Menampilkan hierarki objek Preference sebagai daftar. Proses ini digunakan untuk membuat layar setelan untuk aplikasi Anda.

3.1 Fragment

Fragment

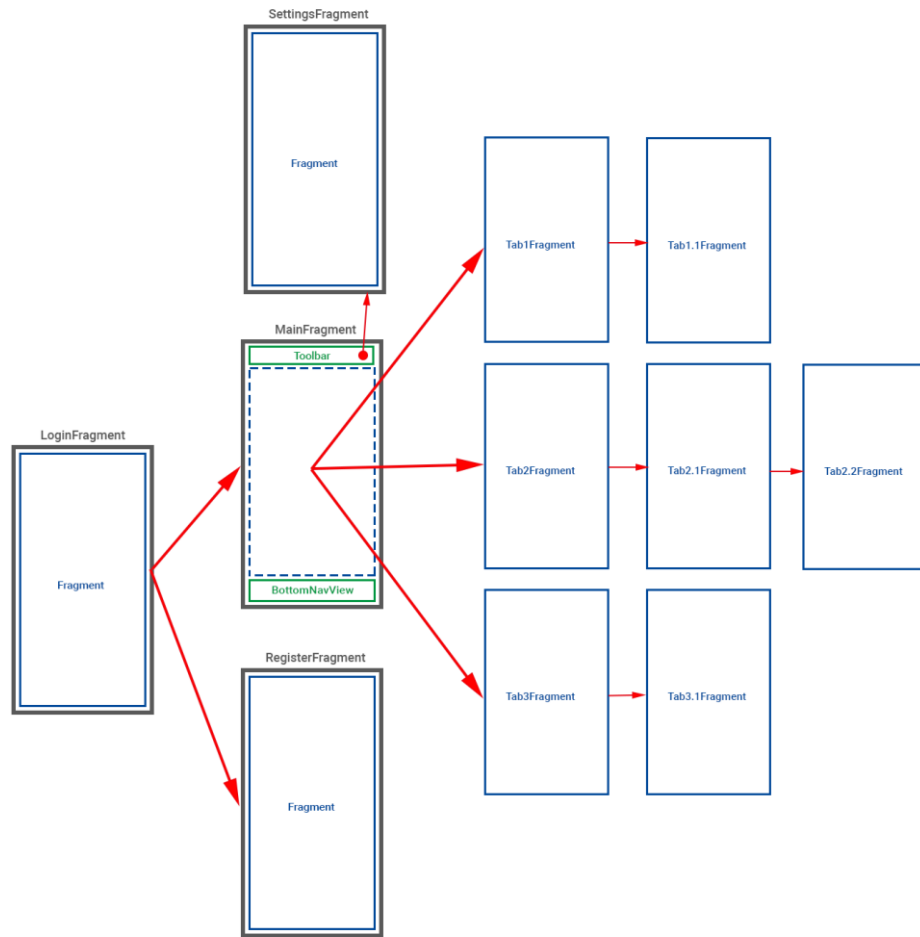




Navigation

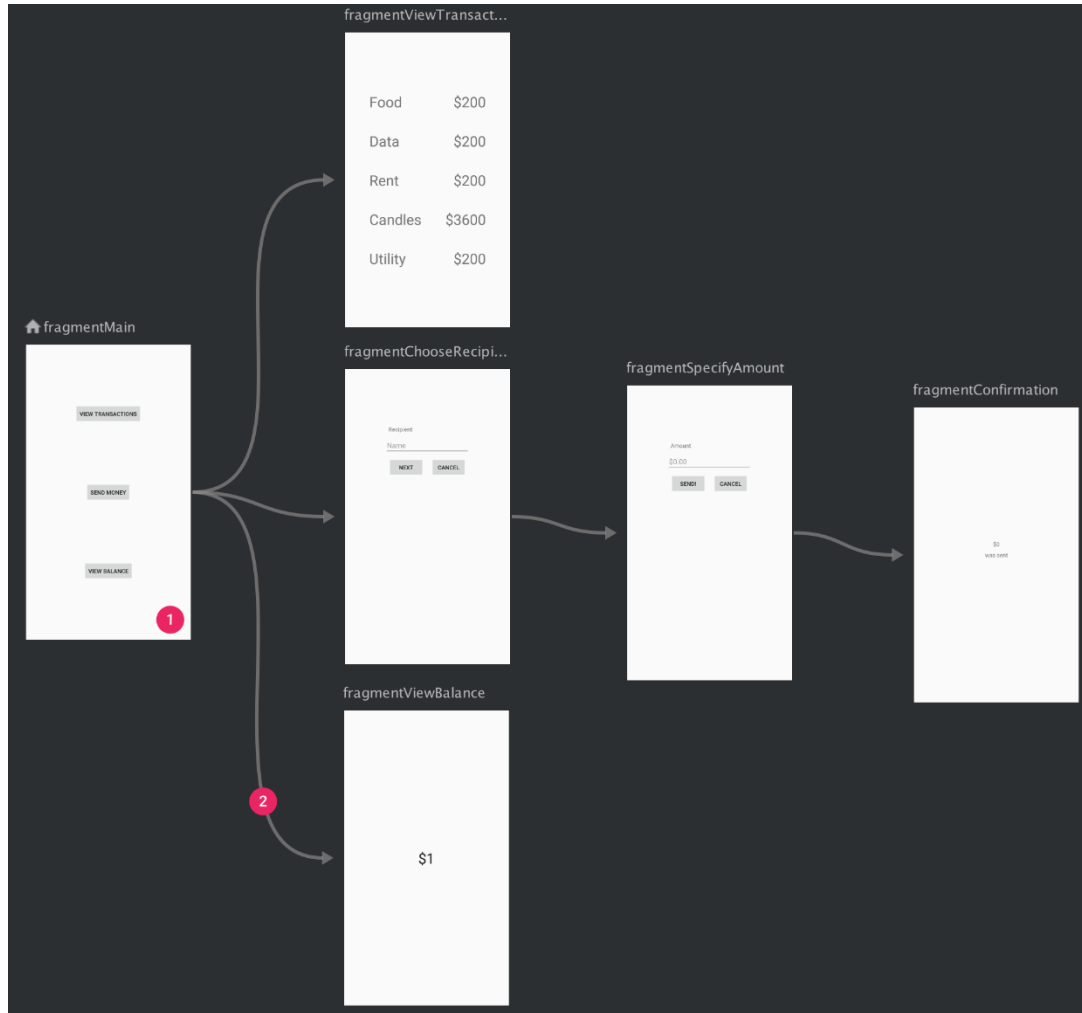


Navigasi



Navigasi adalah interaksi yang memungkinkan pengguna melihat-lihat, masuk, dan keluar dari berbagai konten dalam aplikasi Anda. Komponen Navigasi Android Jetpack membantu Anda menerapkan navigasi, mulai dari klik tombol sederhana hingga pola yang lebih kompleks, seperti bilah aplikasi dan panel samping navigasi. Komponen Navigasi juga memastikan pengalaman pengguna yang konsisten dan dapat diprediksi dengan mendasarkan pada serangkaian prinsip-prinsip yang telah ditetapkan.

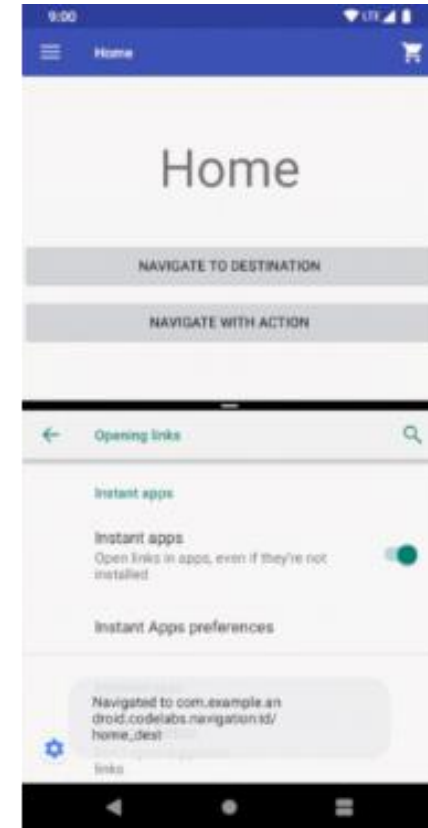
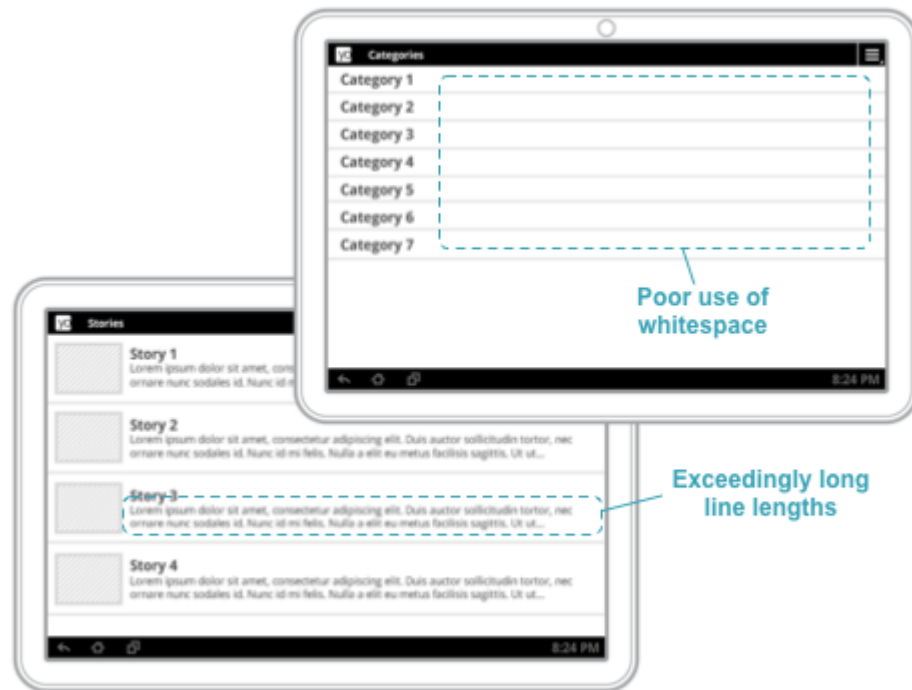
Komponen Navigasi



Komponen Navigasi terdiri dari tiga bagian utama yang dijelaskan di bawah:

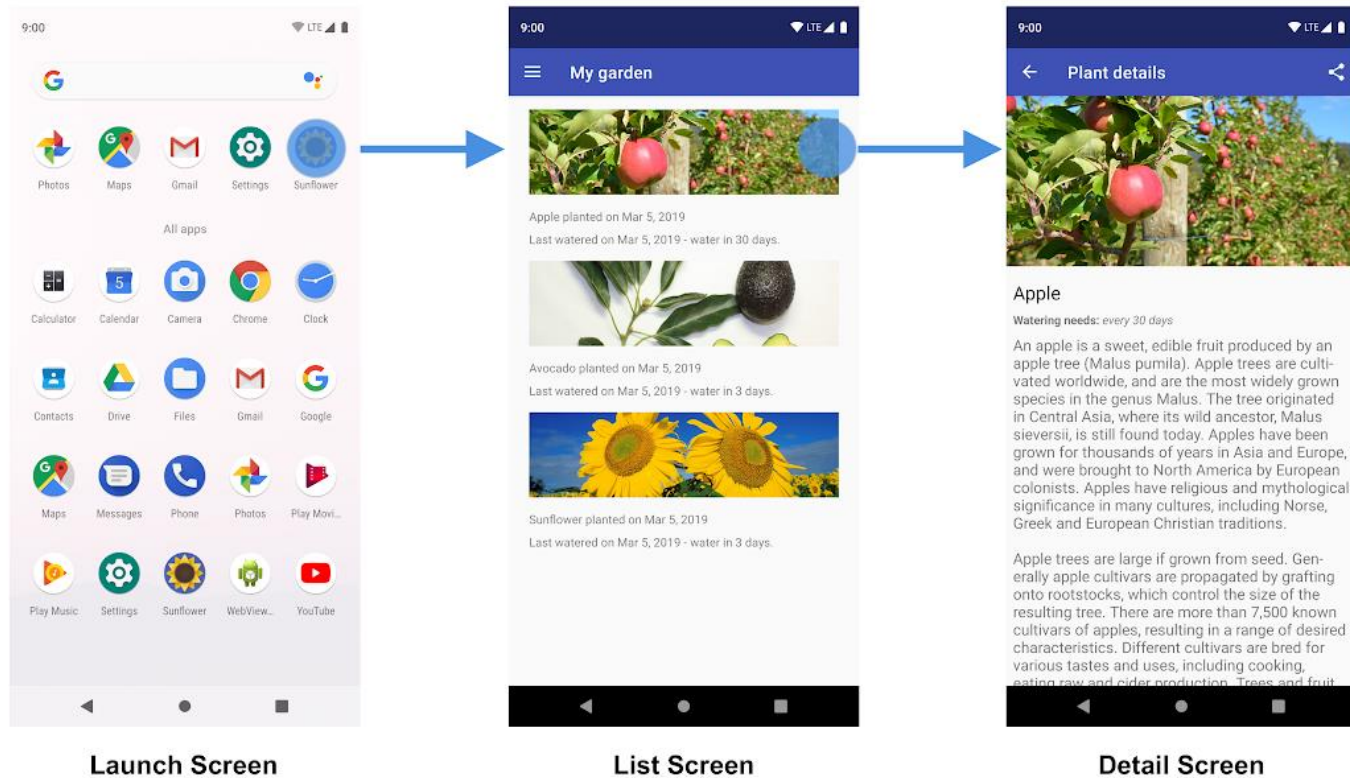
1. Grafik navigasi: Sumber daya XML yang berisi semua informasi terkait navigasi di satu lokasi terpusat. Ini mencakup semua area konten individual dalam aplikasi Anda, yang disebut tujuan, serta jalur yang mungkin dilalui oleh pengguna pada aplikasi Anda.
2. NavHost: Container kosong yang menampilkan tujuan dari grafik navigasi Anda. Komponen Navigasi berisi implementasi NavHost default, NavHostFragment, yang menampilkan tujuan fragmen.
3. NavController: Objek yang mengelola navigasi aplikasi dalam sebuah NavHost. NavController mengatur pertukaran konten tujuan di NavHost saat pengguna menggunakan aplikasi Anda.

Navigasi untuk berbagai bentuk faktor



Navigasi - Contoh

Organic Navigation through Sunflower (Browsing to apple details)



Resulting Sunflower Task Back Stack





Praktikum-03 (Intent)



Intnet

1. Pertama - tama silahkan kalian buat sebuah project baru di Android Studio atau IDE
2. Project name : Nim_Intent
3. Pilih "Empty Activity"
4. Nantinya kita akan membuat 2 buah button yang diletakan di activity utama, dari 2 button itu salah satunya untuk "Intent Explicit" yang akan berpindah ke Activity kedua ketika button itu diklik dan satunya lagi untuk "Intent Implicit" untuk menginstruksikan membuka sebuah alamat web di browser android.
5. **Jika sudah berhasil lalu di screenrecord dikirim ke Group Telegram, dengan subject : Nim_Nama_Intent**
6. Kemudian buat 2 layout activity diantaranya `Activity_main.xml` dan `Activity_second.xml` tambahkan baris program ke layout xml seperti intruksi dibawah ini :

Activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8
9      <Button
10         android:id="@+id/explicit"
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:text="Explicit Intent"
14         android:layout_centerHorizontal="true"
15         android:layout_marginTop="300dp"/>
16
17     <Button
18         android:id="@+id/implicit"
19         android:layout_width="match_parent"
20         android:layout_height="wrap_content"
21         android:text="Implicit Intent"
22         android:layout_centerHorizontal="true"
23         android:layout_marginTop="350dp"/>
24
25  </RelativeLayout>
```

Activity_second.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".SecondActivity">
8
9      <TextView
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:text="Ini adalah Activity kedua"
13         android:textStyle="bold"
14         android:textSize="40dp"
15         android:textAlignment="center"
16         android:layout_marginTop="150dp"
17         android:layout_alignParentStart="true" />
18
19  </RelativeLayout>
```

Intnet

Setelah layout xml beres, langkah selanjutnya adalah memberikan kode program ke class .java Saya membuat 2 class java diantaranya **MainActivity.java** dan **SecondActivity.java**.

MainActivity.java

```
1  import androidx.appcompat.app.AppCompatActivity;
2  import android.content.Intent;
3  import android.net.Uri;
4  import android.os.Bundle;
5  import android.view.View;
6  import android.widget.Button;
7
8  public class MainActivity extends AppCompatActivity implements View.OnClickListener
9
10     Button explicitintent;
11     Button implicitintent;
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_main);
17         explicitintent = (Button)findViewById(R.id.explicit);
18         explicitintent.setOnClickListener(this);
19         implicitintent = (Button)findViewById(R.id.implicit);
20         implicitintent.setOnClickListener(this);
21
22     }
```

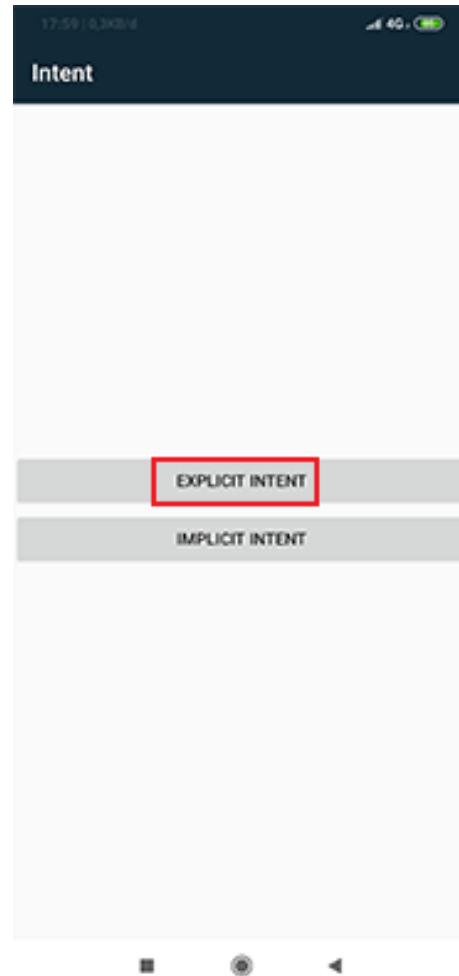

Intnet

```
@Override
public void onClick(View v) {
    switch (v.getId()){
        case R.id.explicitintent:
            Intent explicit = new Intent(MainActivity.this, SecondActivity.class);
            startActivity(explicit);
            break;
        case R.id.implicitintent:
            Intent implicit = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www
            startActivity(implicit);
            break;
        default:
            break;
    }
}
```

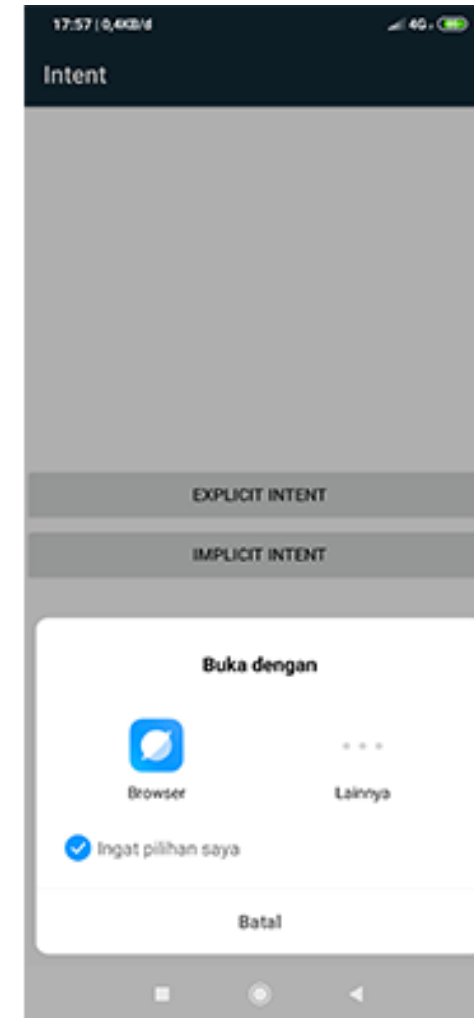
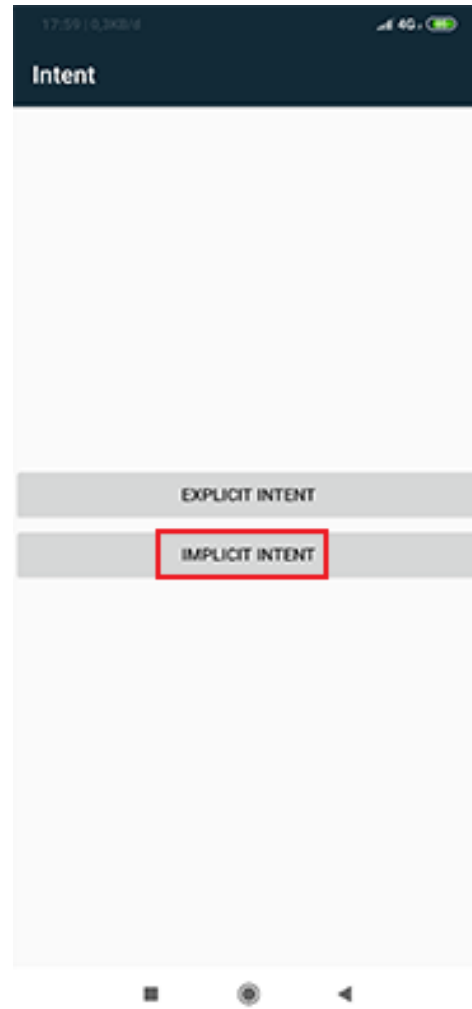
SecondActivity.java

```
1  import androidx.appcompat.app.AppCompatActivity;
2  import android.os.Bundle;
3
4  public class SecondActivity extends AppCompatActivity {
5
6      @Override
7      protected void onCreate(Bundle savedInstanceState) {
8          super.onCreate(savedInstanceState);
9          setContentView(R.layout.activity_second);
10     }
11 }
```

Intnet



Intnet





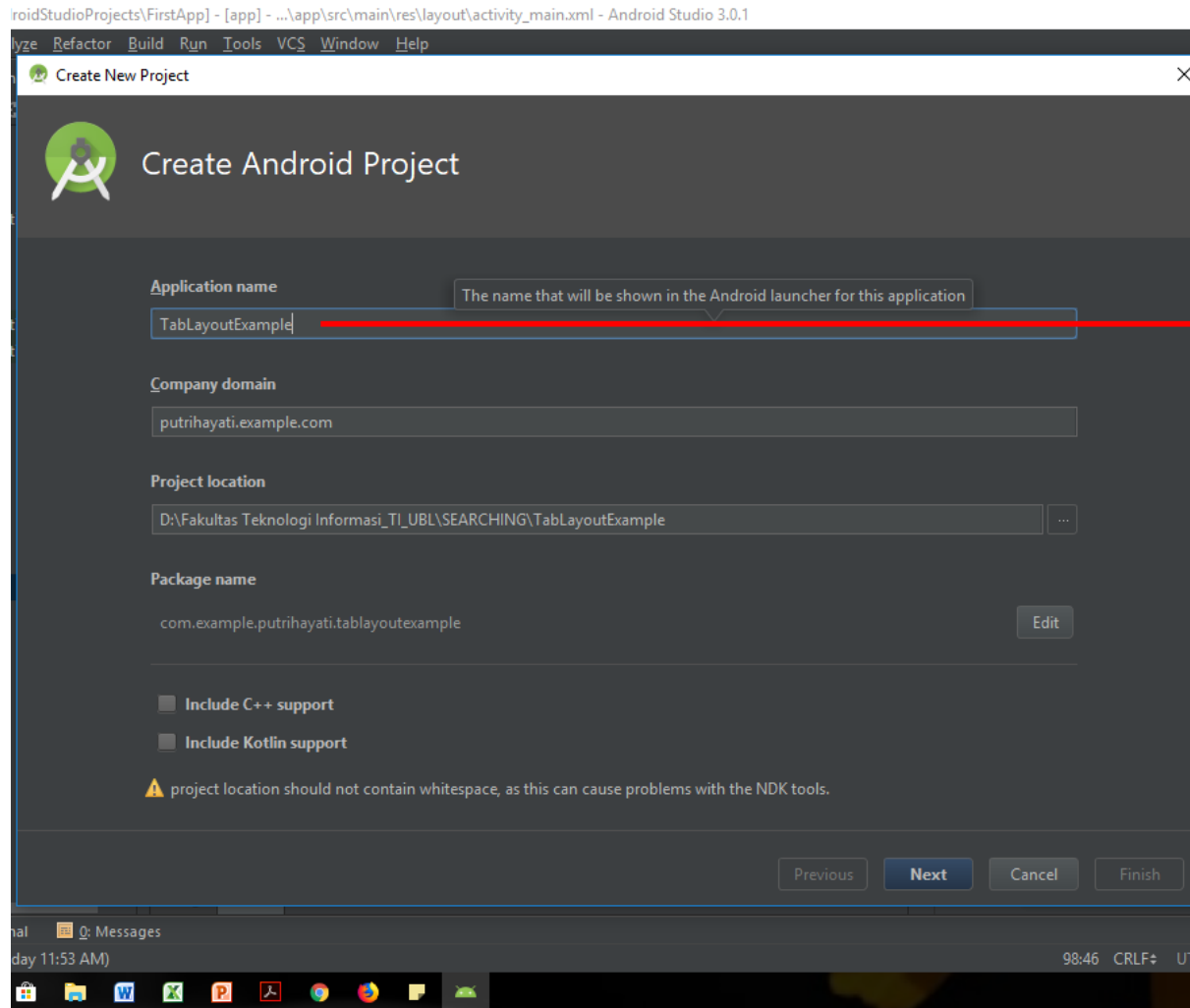
Praktikum-03 (Fragment)



Fragment

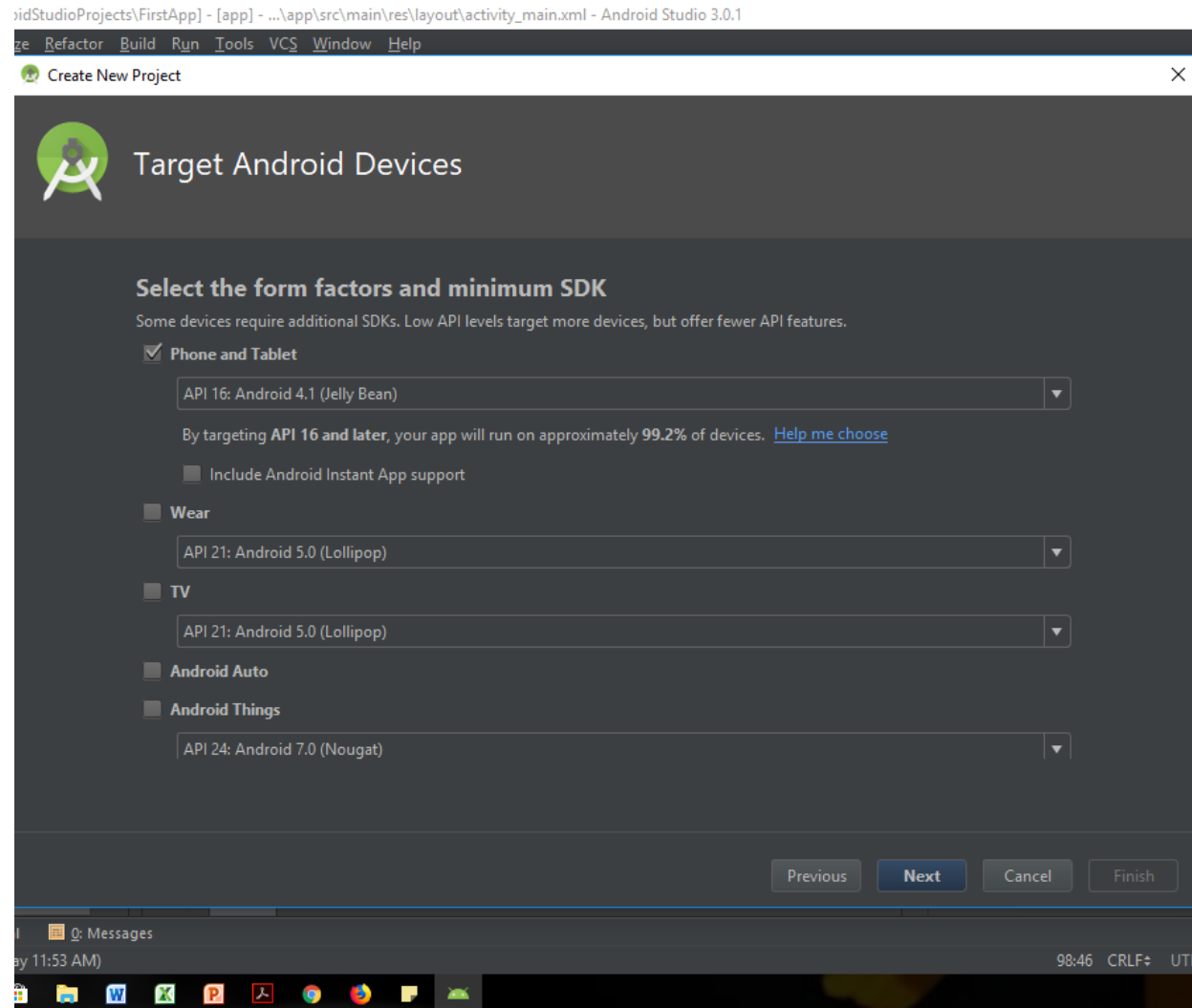
1. Pertama - tama silahkan kalian buat sebuah project baru di Android Studio atau IDE
2. Project name : **Nim_Fragment**
3. Pilih **"Tabbed Activity"**
4. **Jika sudah berhasil lalu di screenrecord dikirim ke Group Telegram, dengan subject : Nim_Nama_Fragment**

Fragment

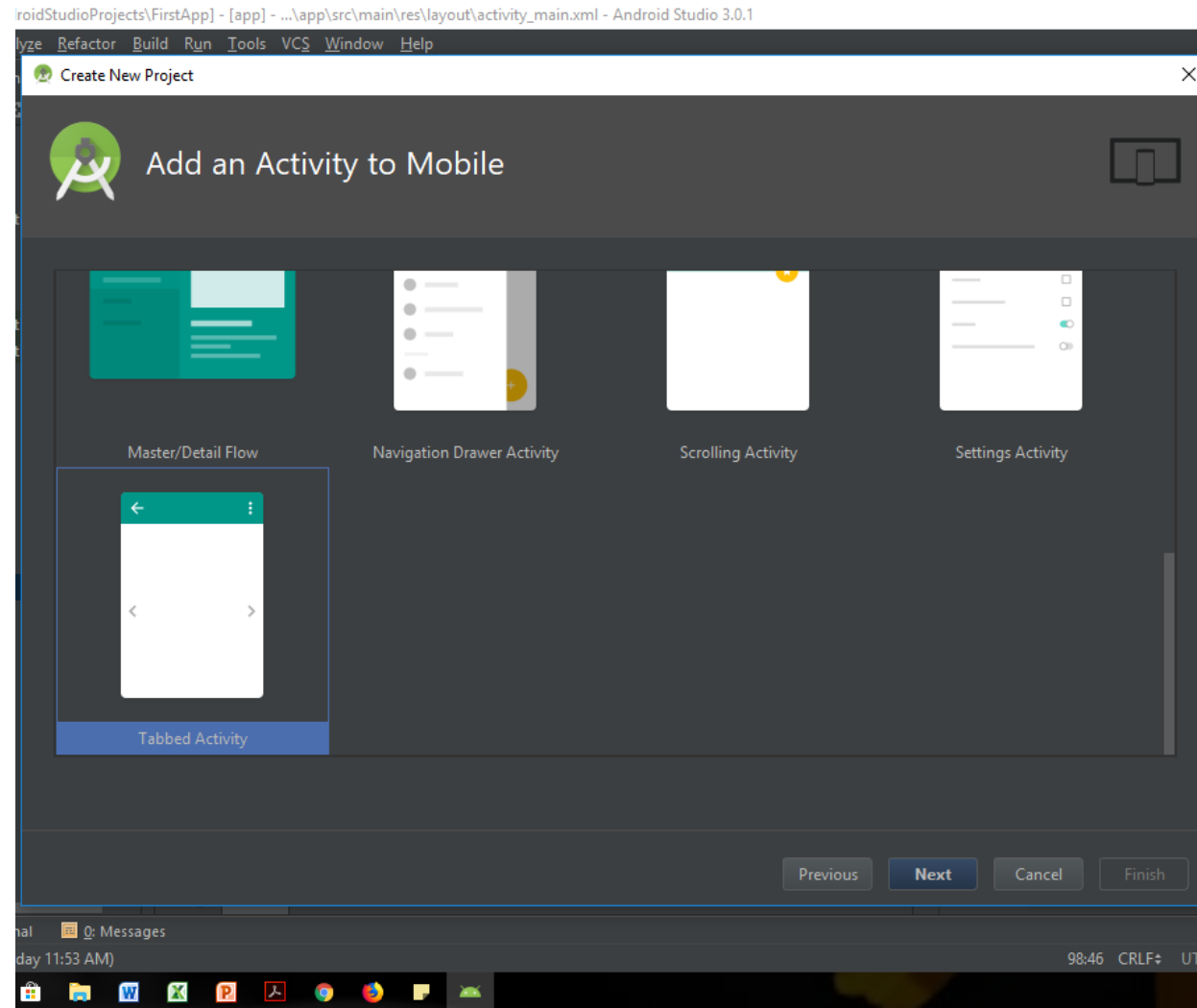


WARNING !!
Application Name:
Nim_Fragment

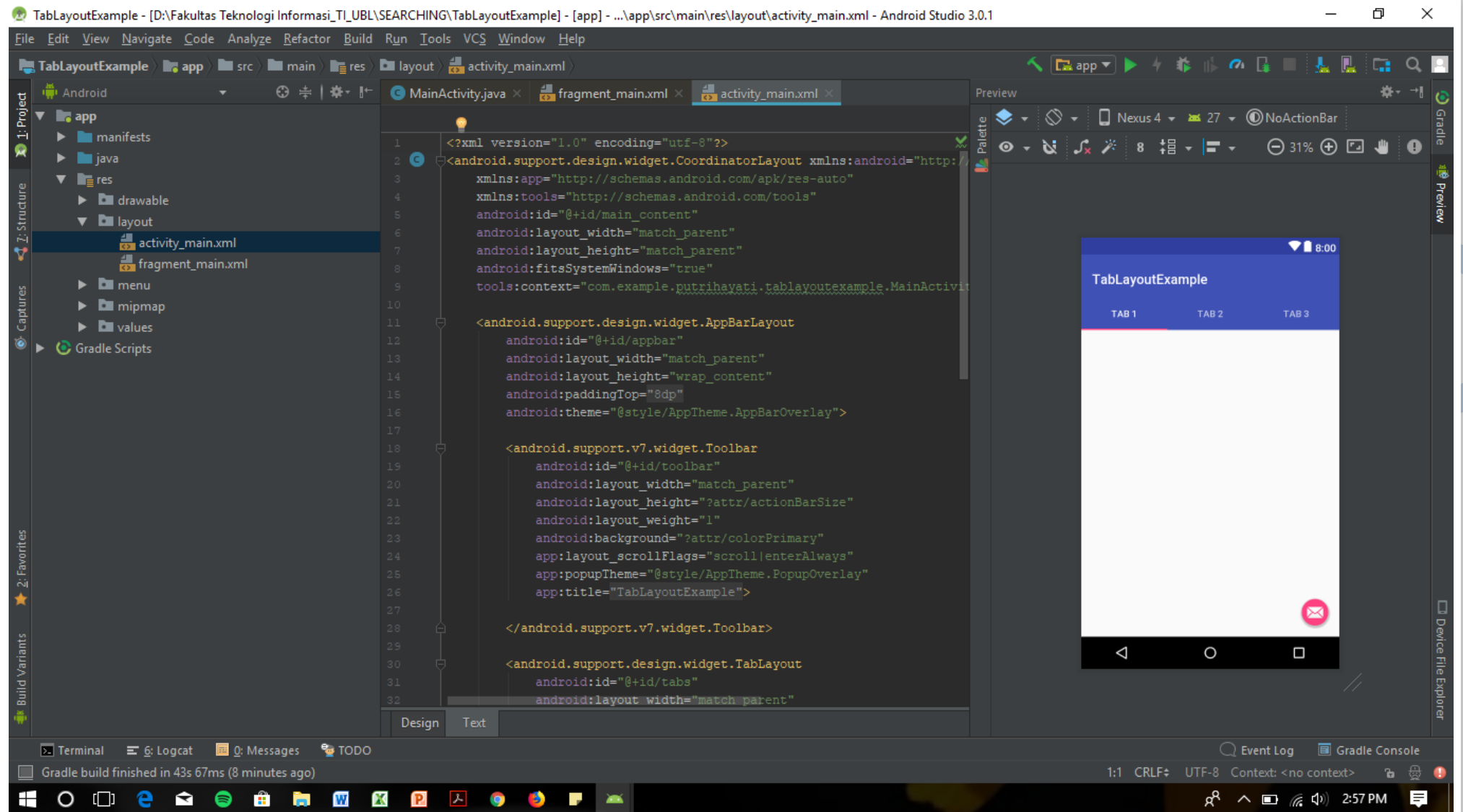
Fragment



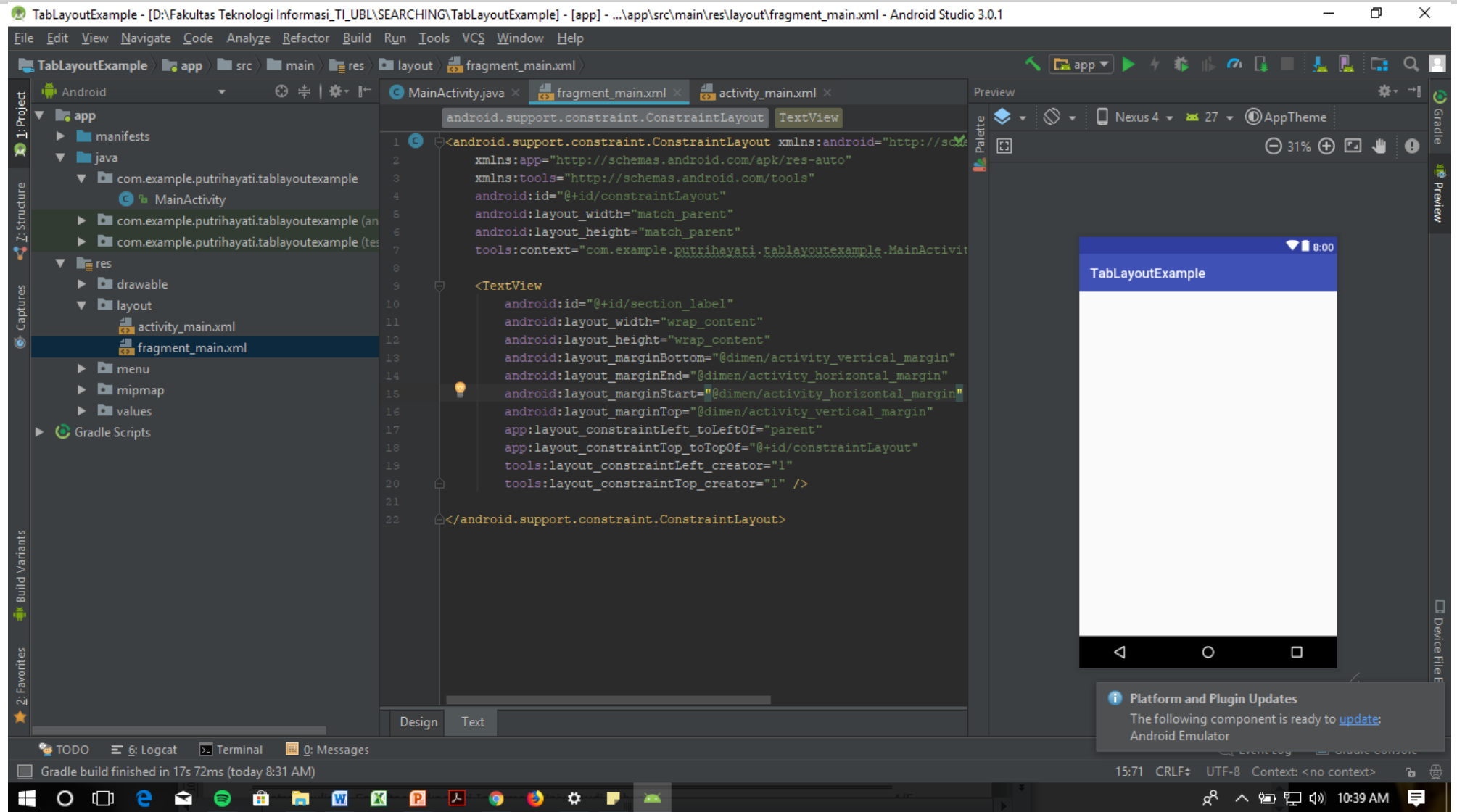
Fragment



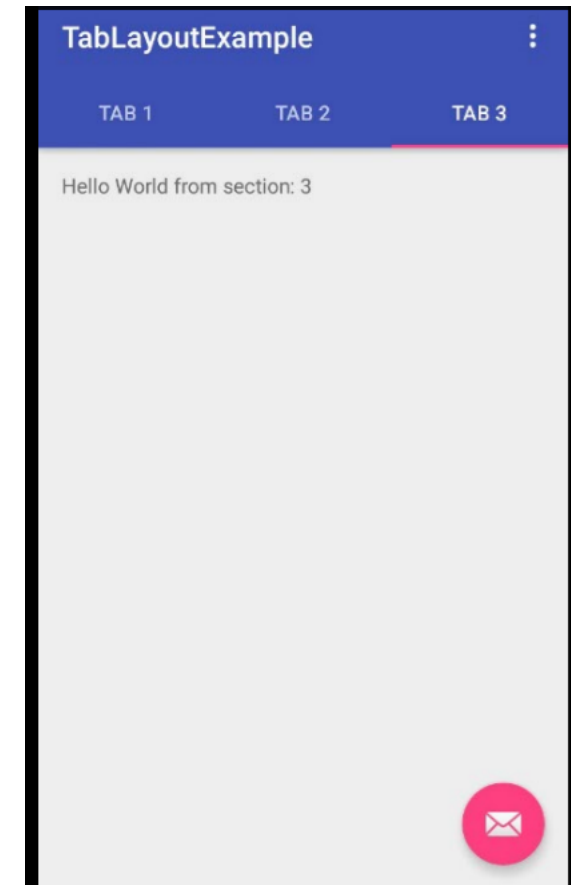
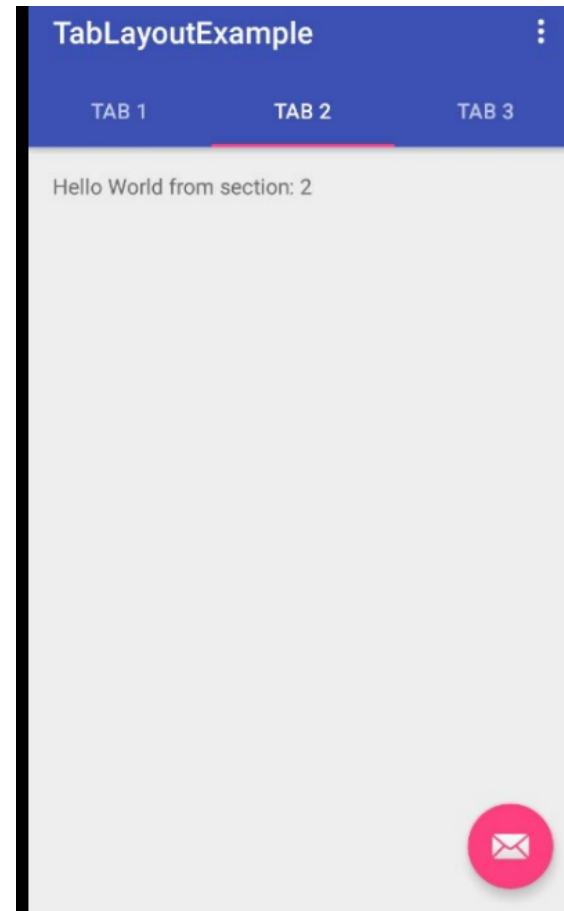
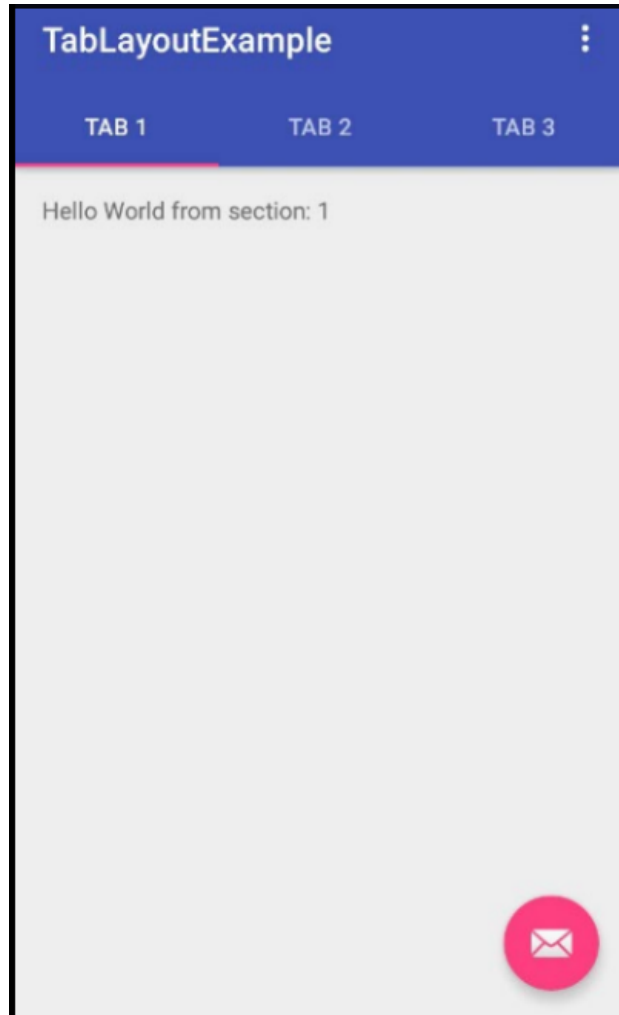
Fragment



Fragment



Fragment



Any Question ?

