

BAB II

FRAGMENT

2.1. Tujuan Pembelajaran :

- a. Mahasiswa mengetahui konsep fragment.
- b. Mahasiswa dapat membuat fragment ke dalam aplikasi.
- c. Mahasiswa dapat mengirimkan data antar fragment.
- d. Mahasiswa dapat menambahkan fragment sebagai dialog

2.2. Software yang dibutuhkan :

- a. Java JDK
- b. Android Studio 4.2
- c. SDK API 26

2.3. Fragment

Fragment merupakan salah satu komponen pada Android Studio dengan fungsi yang hampir sama seperti *activity* tetapi memiliki “*lifecycle*” yang berbeda. *Fragment* merupakan bagian dari sebuah *activity* yang mana sebuah *fragment* tidak akan ada bila tidak ada sebuah *activity* karena *fragment* membutuhkan akses dari *activity* untuk dapat dijalankan. Kelebihan yang didapatkan bila menggunakan *fragment* adalah sebagai berikut :

- a. Fragment memiliki daur hidup sendiri dan bergantung penuh pada daur hidup Activity dimana ia ditanamkan.
- b. Penggunaan fragment lebih kepada pemecahan komponen tampilan
- c. aplikasi untuk menjadi fleksibel dan dapat digunakan kembali (reusable).
- d. Fragment adalah sebuah reuseable class yang mengimplement beberapa fitur sebuah Activity.
- e. Satu activity bisa ditemplei lebih dari satu fragment.
- f. Tidak seperti Activity, fragment tidak perlu didaftarkan ke dalam file AndroidManifest.xml.
- g. Satu kelas java dinyatakan sebuah fragment ketika kelas tersebut mengextends (inherit) kelas fragment.
- h. Melalui Android support library, fragment dapat kompatibel sampai Android API level 10 Gingerbread.

Ada beberapa *state* yang perlu diketahui sebelum menggunakan *fragment* :

a. Resumed

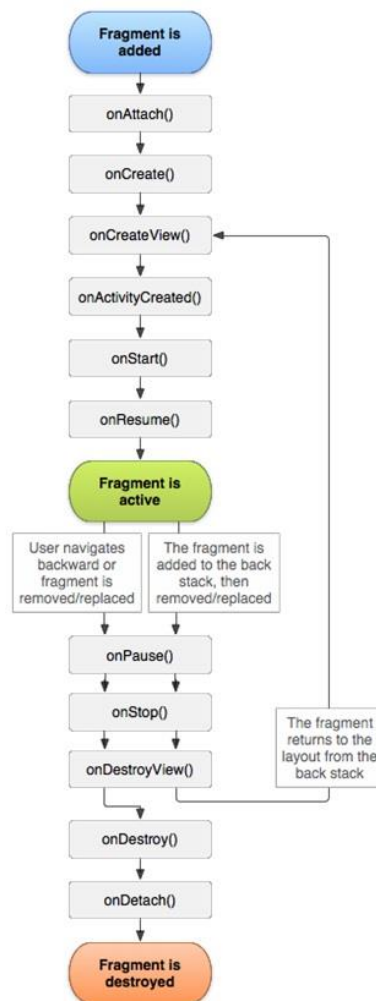
Fragment dapat dilihat ketika Activity sedang berjalan.

b. Paused

Ketika ada Activity lain yang menutupi sebagian dari Activity dimana fragment ditambahkan. Yang dimaksud menutupi sebagian adalah ketika Activitynya tidak tertutup sepenuhnya oleh Activity lain, jadi masih ada bagian dari Activity yang masih bisa kelihatan di layar.

c. Stopped

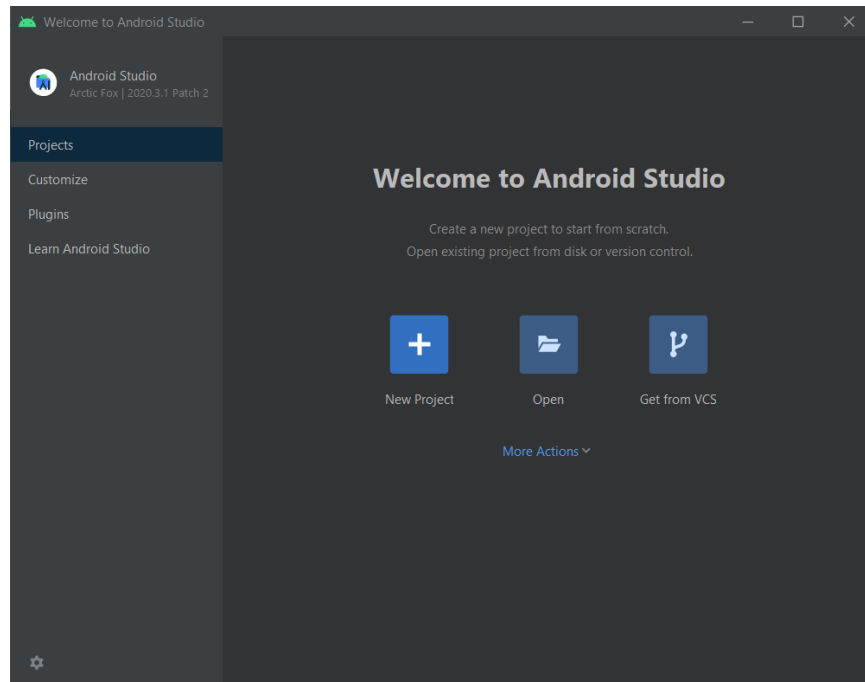
Ketika fragment tidak kelihatan di layar. Bisa jadi karena Activity dimana fragment itu ditambahkan, mengalami state stopped atau bahkan fragment itu sendiri sudah di remove dari Activity dan dilemparkan ke Back Stack. Pada kondisi ini fragment masih hidup dengan semua informasinya, akan tetapi sudah tidak kelihatan di layar dan akan di destroy atau kill ketika Activitynya di destroy.



Gambar 1 : Life Cycle dari Fragment

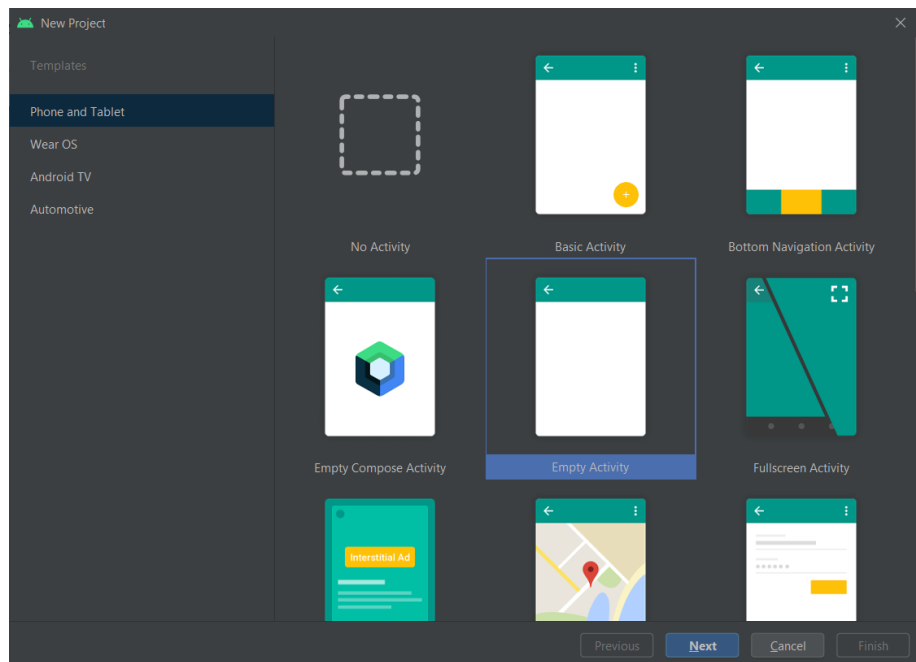
2.4. Langkah – langkah praktikum

1. Membuat project baru pada Android Studio dengan memilih New Project seperti pada gambar dibawah



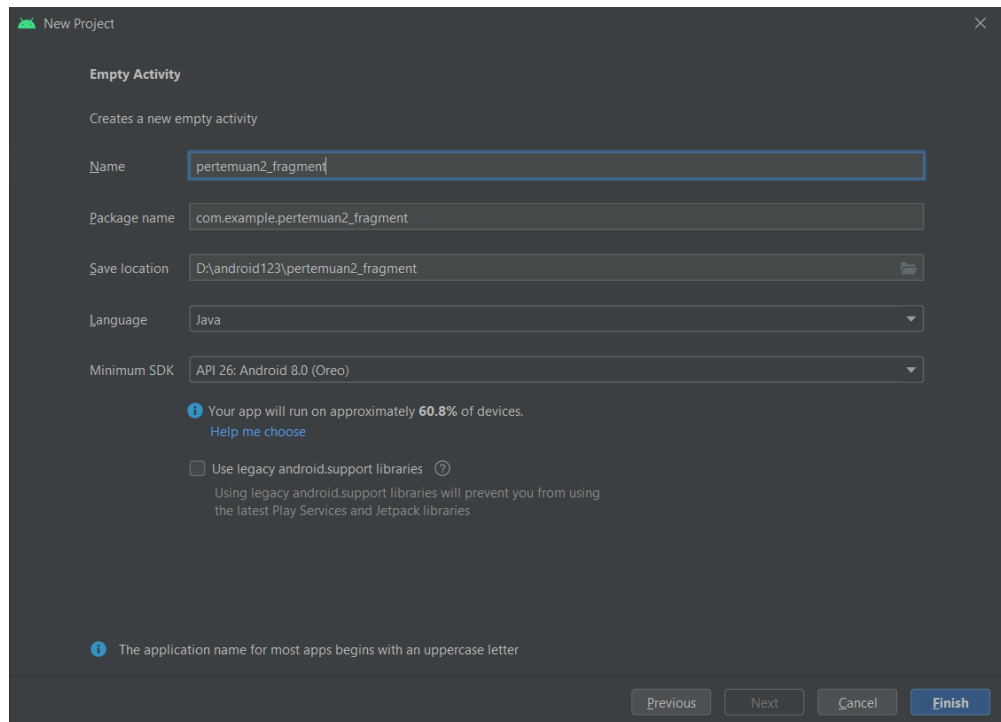
Gambar 2 : Halaman Awal Android Studio

2. Praktikan memilih “Empty Activity” pada “Phone and Tablet” seperti pada gambar dibawah



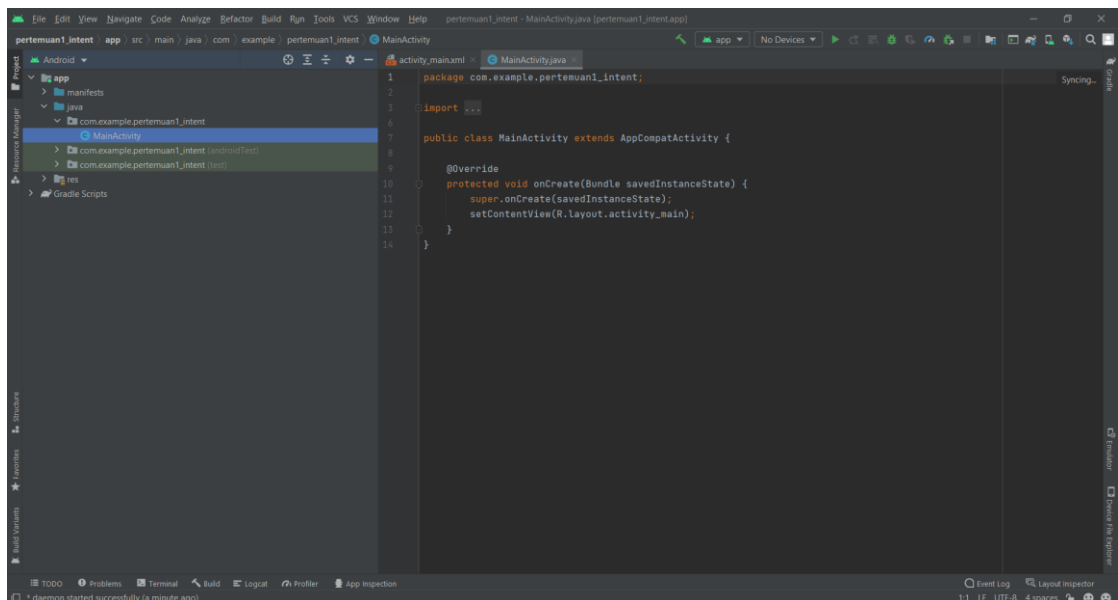
Gambar 3 : Empty Activity pada Phone and Tablet

3. Mengisi identitas project seperti pada gambar. Disarankan memakai minimum SDK 26 (Android 8.0 Oreo). Jika sudah klik finish



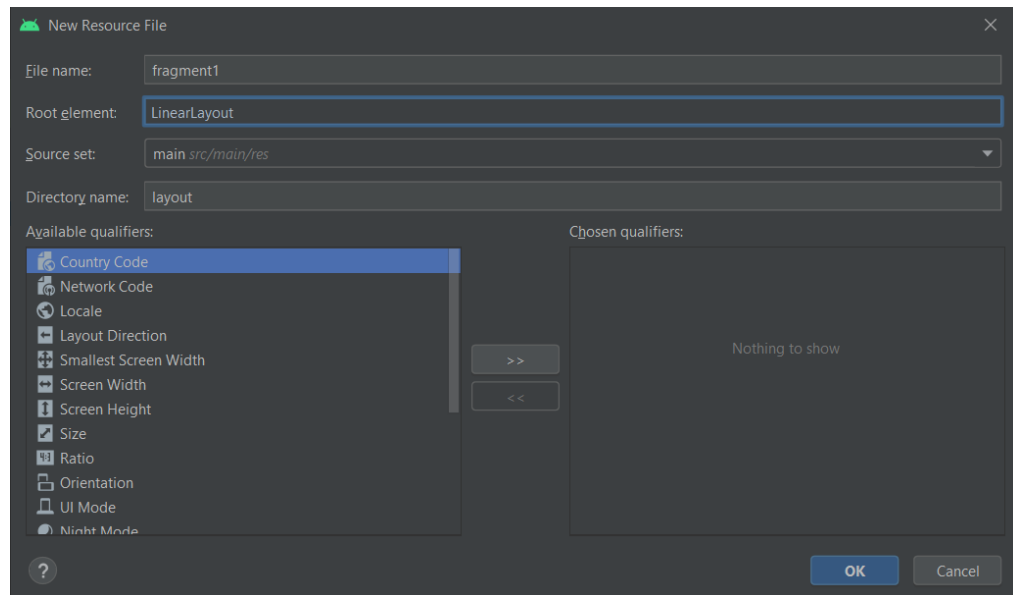
Gambar 4 : Mengisi Identitas Project

4. Jika sudah akan muncul tampilan IDE seperti dibawah



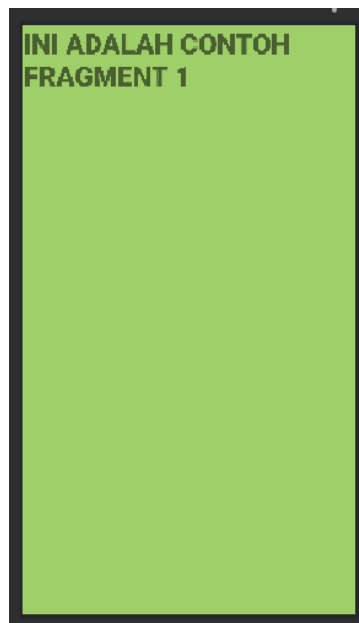
Gambar 5 : IDE Android Studio

5. Klik pada folder res > layout > klik kanan pada layout > New > Layout Resource File



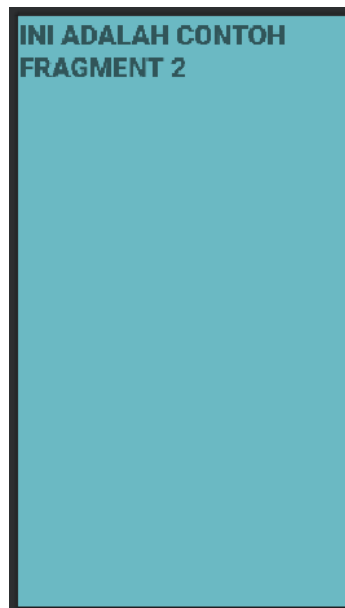
Gambar 6 : Membuat linear layout untuk fragment

6. Klik pada layout yang barusan dibuat, lalu buatlah sebuah tampilan sederhana seperti pada gambar dengan menambahkan komponen TextView



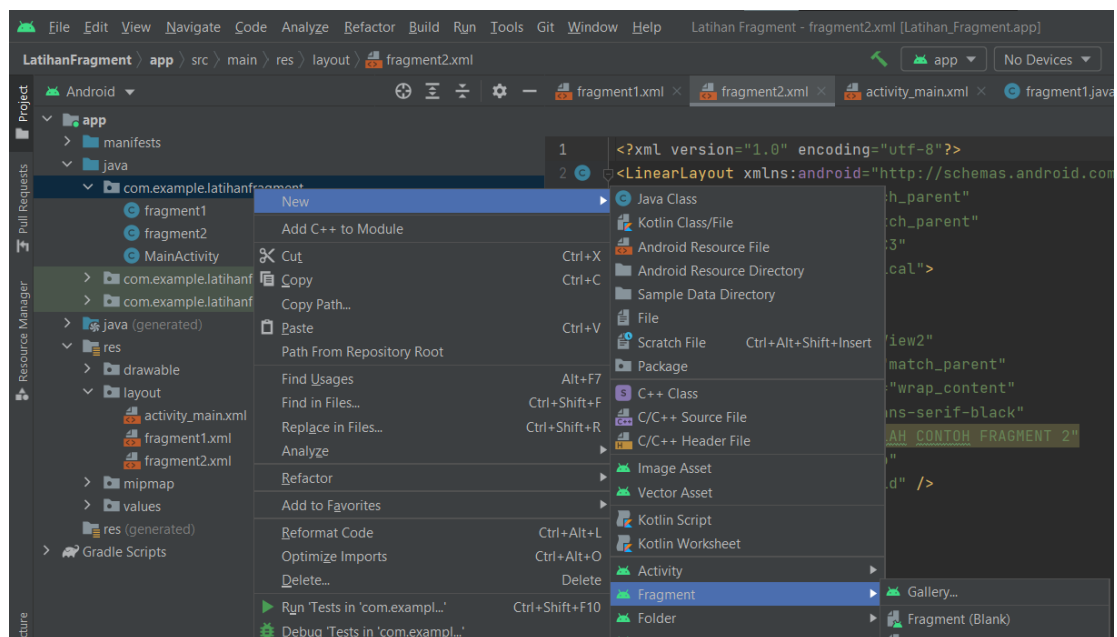
Gambar 7 : Membuat tampilan fragment1

7. Ulangi langkah diatas dengan membuat layout fragment2 dengan tampilan seperti gambar dibawah :



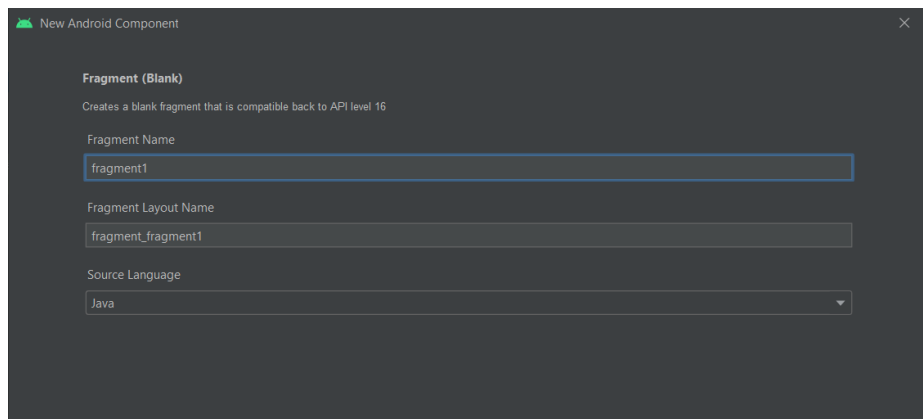
Gambar 8 : Membuat tampilan fragment2

8. Buat class java pada package dengan cara klik package utama kita > New > Fragment > Fragment Blank



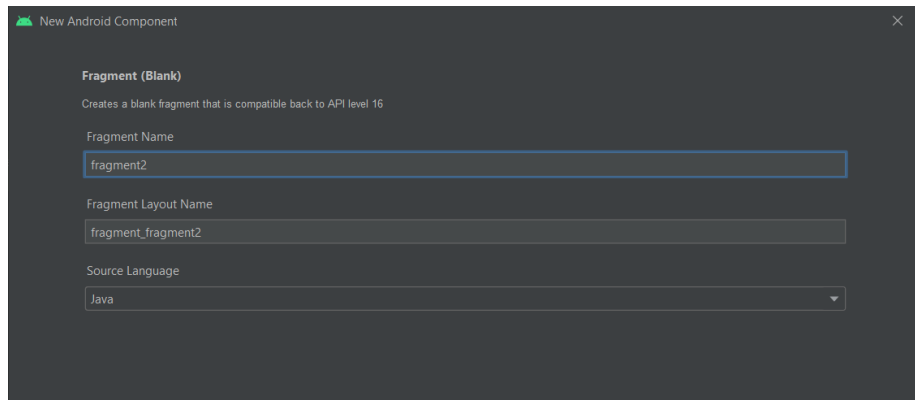
Gambar 9 : Membuat Java class dari fragment

9. Isikan nama fragment seperti pada gambar



Gambar 10 : Memberi nama class fragment

10. Ulangi langkah diatas, lalu buatlah fragment class untuk fragment2



Gambar 11 : Memberi nama class fragment

11. Pergi ke fragment1 class, ubahlah source code pada method onCreateView() seperti dibawah :

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                        Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment1, container,
false);
}
```

12. Lakukan hal yang sama pada fragment2

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                        Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment2, container,
false);
}
```

13. Langkah selanjutnya adalah menggabungkan kedua fragment menjadi 1 pada activity main. Bukalah activity_main.xml pada resource layout, lalu ubahlah dari constraint layout menjadi LinearLayout. Ubahlag tag menjadi seperti pada gambar dibawah

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    tools:context=".MainActivity">

</LinearLayout>
```

14. Buatlah tag fragment didalam activity_main.xml

```
<fragment
    android:id="@+id/fragment1"
    android:name="com.example.pertemuan2_fragment.fragment1"
    android:layout_width="0px"
    android:layout_height="match_parent"
    android:layout_weight="1" />

<fragment
    android:id="@+id/fragment2"
    android:name="com.example.pertemuan2_fragment.fragment2"
    android:layout_width="0px"
    android:layout_height="match_parent"
    android:layout_weight="1" />
```

2.5. Tugas Rumah

Buatlah project baru, lalu tampilkan 4 fragment didalam 1 Activity seperti pada contoh gambar dibawah :

