

WORKSHOP APLIKASI MOBILE
LAPORAN PRAKTIKUM MINGGU 5
‘FRAGMENT’



DISUSUN OLEH :

1. Firgo Bhaktiar Hamsah (E41191039)
2. Yulinar Nur Rahmawati (E41191151)
3. Muhammad Lutfi Fadilah (E41190889)
4. M.Ainul Yaqin (E41190793)

Teknik Informatika Golongan B

PROGRAM STUDI D4 TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI JEMBER
2021

TUGAS PRAKTIKUM 5

Fragment merupakan komponen view yang fleksibel dan dapat disisipkan pada sebuah tampilan dari *activity*, di samping itu fragment juga merupakan komponen yang memiliki fungsi untuk menampilkan antarmuka ke pengguna melalui *activity* dengan memiliki *layout XML* sendiri.

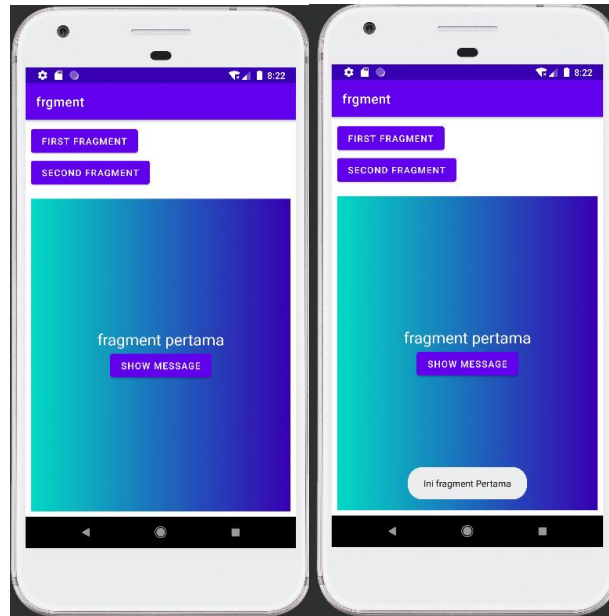
Berikut ini penjelasan mengenai tugas praktikum 5 :

- Tugas Fragment

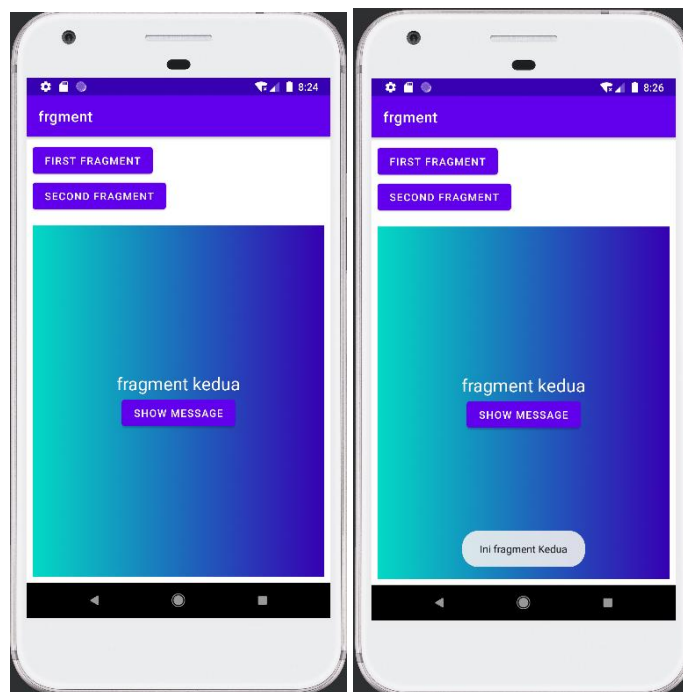
Berikut ini hasil *screen capture* hasil ketika dijalankan :



Gambar 1. Tampilan Utama



Gambar 2. Fragment_1

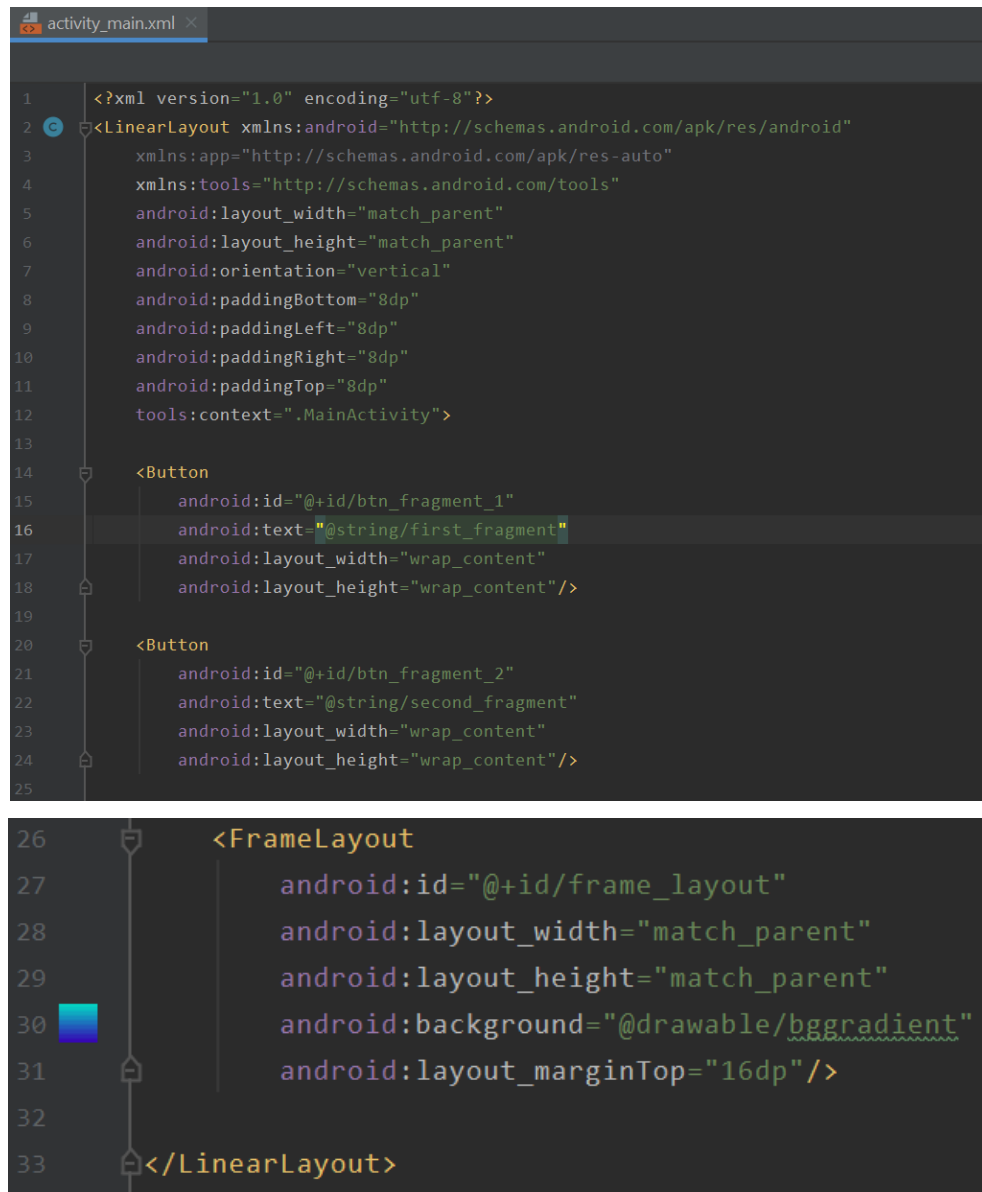


Gambar 3. Fragment_2

Pada gambar diatas merupakan hasil *screen capture* tugas fragment ketika dijalankan menggunakan emulator. Pada gambar diatas menggunakan *layout* berupa *LinearLayout* dengan orientasi vertikal atau mendatar.

Berikut ini *source code* dari tampilan (*Gambar 1. Tampilan Utama*) :

Activity_main.xml :



```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout 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      android:orientation="vertical"
8      android:paddingBottom="8dp"
9      android:paddingLeft="8dp"
10     android:paddingRight="8dp"
11     android:paddingTop="8dp"
12     tools:context=".MainActivity">
13
14     <Button
15         android:id="@+id/btn_fragment_1"
16         android:text="@string/first_fragment"
17         android:layout_width="wrap_content"
18         android:layout_height="wrap_content"/>
19
20     <Button
21         android:id="@+id/btn_fragment_2"
22         android:text="@string/second_fragment"
23         android:layout_width="wrap_content"
24         android:layout_height="wrap_content"/>
25
26     <FrameLayout
27         android:id="@+id/frame_layout"
28         android:layout_width="match_parent"
29         android:layout_height="match_parent"
30         android:background="@drawable/bggradient"
31         android:layout_marginTop="16dp"/>
32
33 </LinearLayout>
```

Gambar 4. Source Code *activity_main.xml*

Pada baris 2 gambar kode program diatas menunjukkan bahwa *layout* yang digunakan yaitu *LinearLayout* dengan orientasi vertikal. Panjang dan Lebar pada *LinearLayout* tersebut diatur dengan perintah seperti pada baris 5 dan 6 yang memiliki nilai *match_parent*.

- *Match_parent* digunakan untuk mengatur ukuran lebar dan tinggi dengan ukuran mengikuti layar atau *screen* pada perangkat android.

Pada kode program baris 8 – 11, *padding* digunakan untuk mengatur atau memberi jarak antara objek satu dengan objek lainnya.

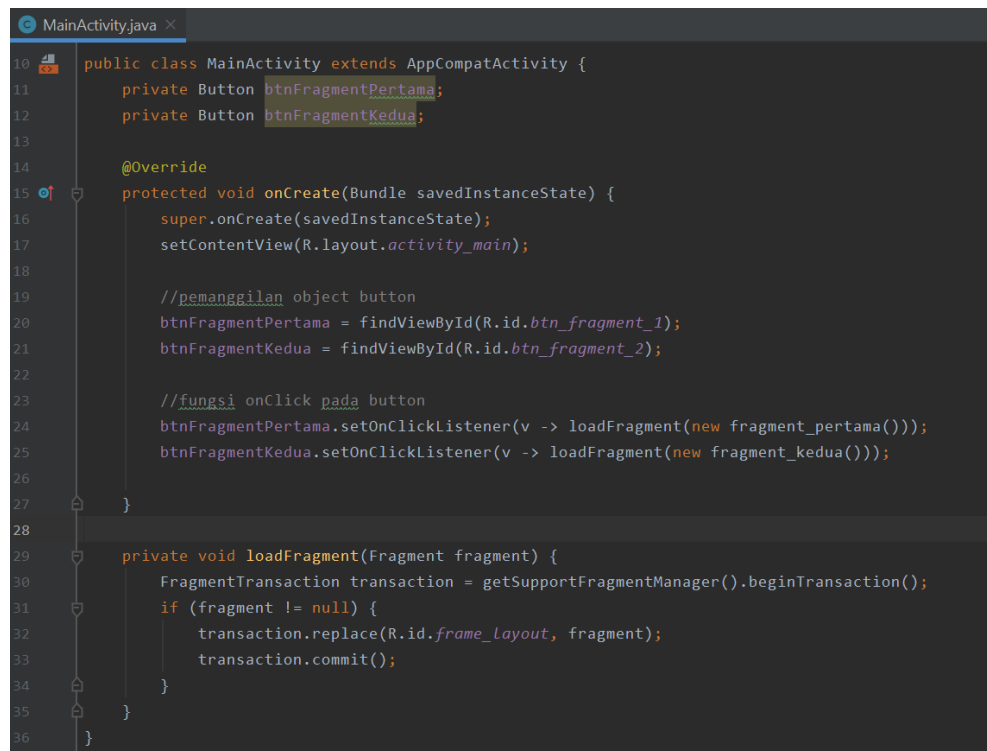
Pada kode program baris 14 – 24 merupakan kode program untuk mengatur *Button*, *button* biasa digunakan sebagai tombol. Di dalam *script button* tersebut terdapat beberapa *script* diantaranya :

- a. *Android:id*, digunakan untuk memberikan identifikasi terhadap tombol.
- b. *Android:text*, digunakan untuk memberikan keterangan berupa text pada tombol.
- c. *Android:layout_width*, digunakan untuk mengatur ukuran lebar pada tombol.
- d. *Android:layout_height*, digunakan untuk mengatur ukuran tinggi pada tombol.

Pada kode program baris 26 – 31 merupakan kode program *FrameLayout*, yang biasa digunakan untuk membuat objek yang saling bertindihan dengan beberapa *script* yang sama dengan *script Button*, berikut ini penjelasan *script* yang berbeda dengan *script* yang ada pada *Button* :

- a. *Android:background*, digunakan untuk memberikan variasi pada bagian latar belakang *FrameLayout*.
- b. *Android:layout_marginTop*, digunakan untuk mengatur ukuran jarak antara *FrameLayout* dengan bagian atasnya sebesar 16 dp.

Berikut ini *source code* pada *MainActivity.java* untuk menambahkan beberapa baris kode program untuk memulai tombol.



```

10 public class MainActivity extends AppCompatActivity {
11     private Button btnFragmentPertama;
12     private Button btnFragmentKedua;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18
19         //pemanggilan object button
20         btnFragmentPertama = findViewById(R.id.btn_fragment_1);
21         btnFragmentKedua = findViewById(R.id.btn_fragment_2);
22
23         //fungsi onClick pada button
24         btnFragmentPertama.setOnClickListener(v -> loadFragment(new fragment_pertama()));
25         btnFragmentKedua.setOnClickListener(v -> loadFragment(new fragment_kedua()));
26
27     }
28
29     private void loadFragment(Fragment fragment) {
30         FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();
31         if (fragment != null) {
32             transaction.replace(R.id.frame_layout, fragment);
33             transaction.commit();
34         }
35     }
36 }

```

Gambar 5. Source Code MainActivity.java

Pada baris kode program 11 – 12 digunakan untuk mendeklarasikan sebuah *Button*. Sedangkan pada baris 20 - 21 digunakan untuk memanggil objek *Button* berdasarkan dengan *id* yang telah di deklarasikan pada file *activity_main.xml* sebelumnya.

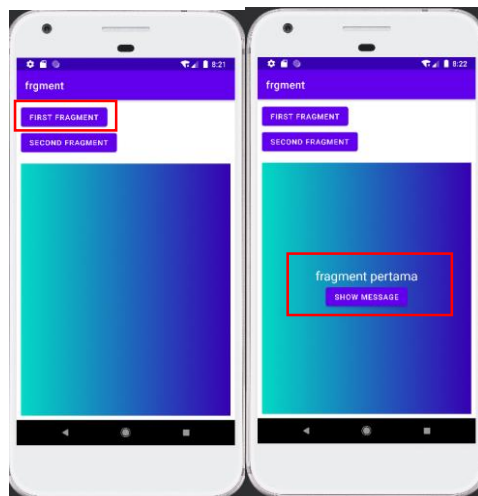
Pada baris 24 – 25 melakukan *setOnClickListener* pada kedua tombol yang digunakan untuk membuka atau menjalankan file *fragment_pertama* dan juga file *fragment_kedua* dengan tata letak *FrameLayout*. Untuk mengganti *Fragment* dengan *FrameLayout* dengan membuat *Manajer Fragmen* dan kemudian memulai transaksi menggunakan *Transaksi Fragmen* dan terakhir mengganti *Fragmen* dengan tata letak yaitu *FrameLayout*.

Berikut ini *source code* pada *activity_fragment_pertama* :

```
activity_fragment_pertama.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:orientation="vertical"
7      xmlns:app="http://schemas.android.com/apk/res-auto"
8      android:background="@drawable/bggradient"
9      android:gravity="center"
10     tools:context=".fragment_pertama">
11
12     <TextView
13         android:text="fragment pertama"
14         android:textSize="24sp"
15         android:textColor="@color/white"
16         android:layout_width="wrap_content"
17         android:layout_height="wrap_content"/>
18
19     <Button
20         android:id="@+id/btn_toast"
21         android:text="show message"
22         android:layout_width="wrap_content"
23         android:layout_height="wrap_content"/>
24 </LinearLayout>
```

Gambar 6. Source Code *activity_fragment_pertama.xml*

Kode program diatas digunakan untuk membuat tampilan yang muncul berupa teks dan tombol yang ditandai dengan kode program baris 12 dan juga baris 19, ketika mengklik tombol utama dan aplikasi dijalankan, seperti pada gambar dibawah ini :



Gambar 7. Running Fragment 1

Berikut ini *source code* pada file *fragment_pertama.java* :

```
fragment_pertama.java
1 package com.example.frgment;
2
3 import ...
4
5 public class fragment_pertama extends Fragment {
6
7     private Button btnShowMessage;
8
9     @Override
10    public View onCreateView(LayoutInflater inflater, ViewGroup container,
11                             Bundle savedInstanceState) {
12        return inflater.inflate(R.layout.activity_fragment_pertama, container, attachToRoot: false);
13    }
14
15    @Override
16    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
17        super.onViewCreated(view, savedInstanceState);
18
19        btnShowMessage = view.findViewById(R.id.btn_toast);
20        btnShowMessage.setOnClickListener(
21            v -> Toast.makeText(getContext().getApplicationContext(), text: "Ini fragment Pertama", Toast.LENGTH_SHORT).show());
22    }
23 }
```

Gambar 8. Fragment_pertama.java

Pada baris kode program 28 – 30 digunakan untuk menampilkan tulisan (*Ini fragment pertama*) ketika tombol dengan *id* : *btn_toast* di klik, ketika dijalankan maka akan tampil seperti pada gambar dibawah ini :



Gambar 9. Hasil Running Fragment_1

Begitu juga dengan tampilan pada fragmen kedua, dengan *source code* sebagai berikut :

a. *Source code* activity_fragment_kedua.xml

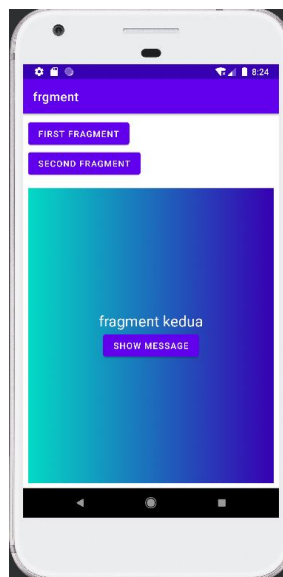

```

activity_fragment_kedua.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:orientation="vertical"
7      xmlns:app="http://schemas.android.com/apk/res-auto"
8      android:background="@drawable/bggradient"
9      android:gravity="center"
10     tools:context=".fragment_kedua">
11
12     <TextView
13         android:text="fragment kedua"
14         android:textSize="24sp"
15         android:textColor="@color/white"
16         android:layout_width="wrap_content"
17         android:layout_height="wrap_content"/>
18
19     <Button
20         android:id="@+id/btn_toast"
21         android:text="show message"
22         android:layout_width="wrap_content"
23         android:layout_height="wrap_content"/>
24
25 </LinearLayout>

```

Gambar 10. Source code activity_fragment_kedua.xml

Pada kode program diatas akan menampilkan tampilan seperti pada gambar dibawah ini ketika aplikasi dijalankan :



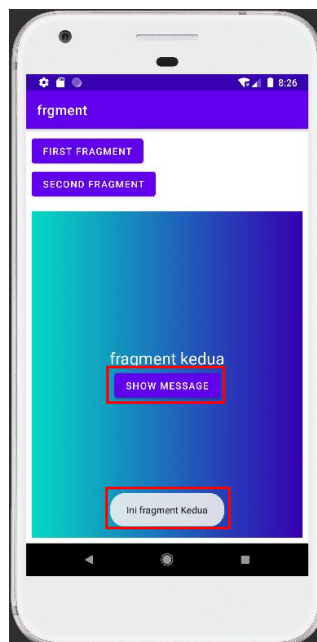
Gambar 11. Running Fragment 2

b. Source code fragment_kedua.java

```
1 package com.example.frgment;
2
3 import androidx.fragment.app.Fragment;
4
5 public class Fragment_kedua extends Fragment {
6
7     private Button btnShowMessage;
8
9     // Dijalankan saat pembuatan View
10    @Override
11    public View onCreateView(LayoutInflater inflater, ViewGroup container,
12        Bundle savedInstanceState) {
13
14        return inflater.inflate(R.layout.activity_fragment_kedua, container, attachToRoot false);
15    }
16
17    @Override
18    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
19        super.onViewCreated(view, savedInstanceState);
20
21        btnShowMessage = view.findViewById(R.id.btn_toast);
22        btnShowMessage.setOnClickListener(v -> Toast.makeText(getContext().getApplicationContext(), "Ini fragment_Kedua", Toast.LENGTH_SHORT).show());
23    }
24 }
```

Gambar 12. Source code fragment_kedua.java

Pada baris kode program 28 – 32 digunakan untuk menampilkan tulisan (*Ini fragment kedua*) saat tombol dengan *id : btn_toast* di klik, dan ketika dijalankan maka akan tampil seperti pada gambar dibawah ini :



Gambar 13. Hasil Running Fragment_2