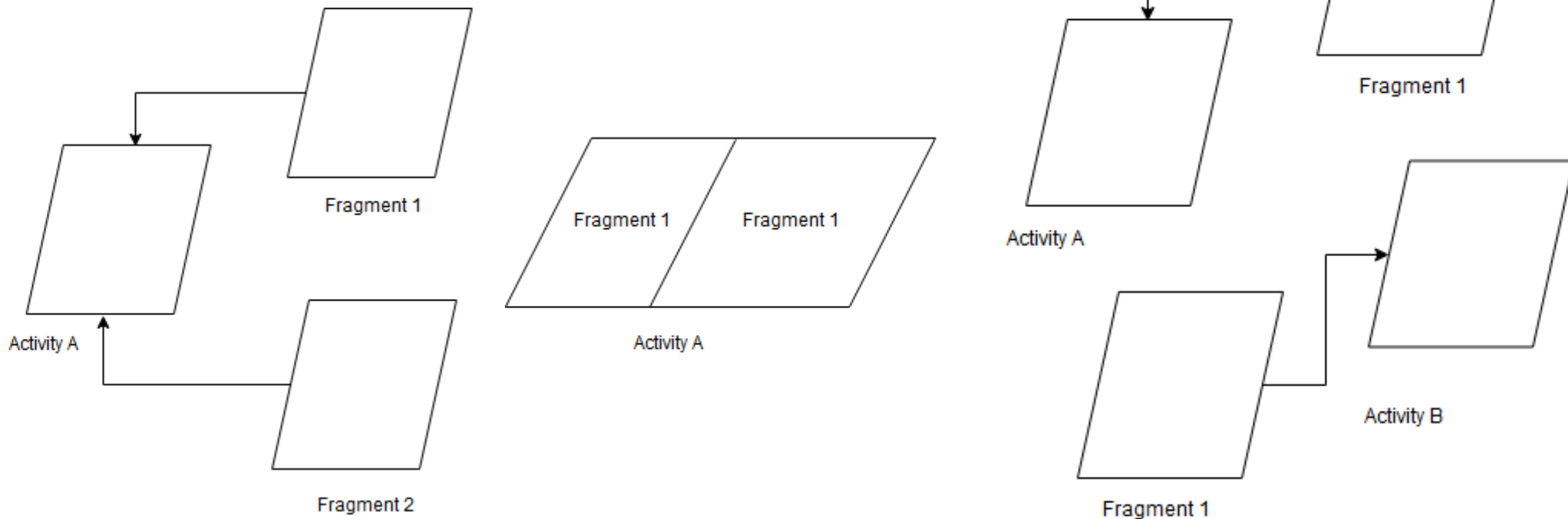




# Fragment

# Flexibilitas Fragment

- Untuk membuat fragment memerlukan activity tempat fragment tersebut berada dimana satu activity dapat menampilkan satu atau lebih fragment. Fragment juga bersifat reusable, fragment yang sama dapat digunakan pada activity yang berbeda





# FragmentManager dan FragmentTransaction

- Ada dua objek penting yang berkaitan dengan fragment yaitu FragmentManager dan FragmentTransaction.
- FragmentManager bertugas mengorganisir fragment sedangkan FragmentTransaction bertugas memanipulasi fragment. Semisal ketika kita ingin menambahkan fragment ke suatu activity, menghapus fragment dari suatu activity maka kita menggunakan FragmentTransaction

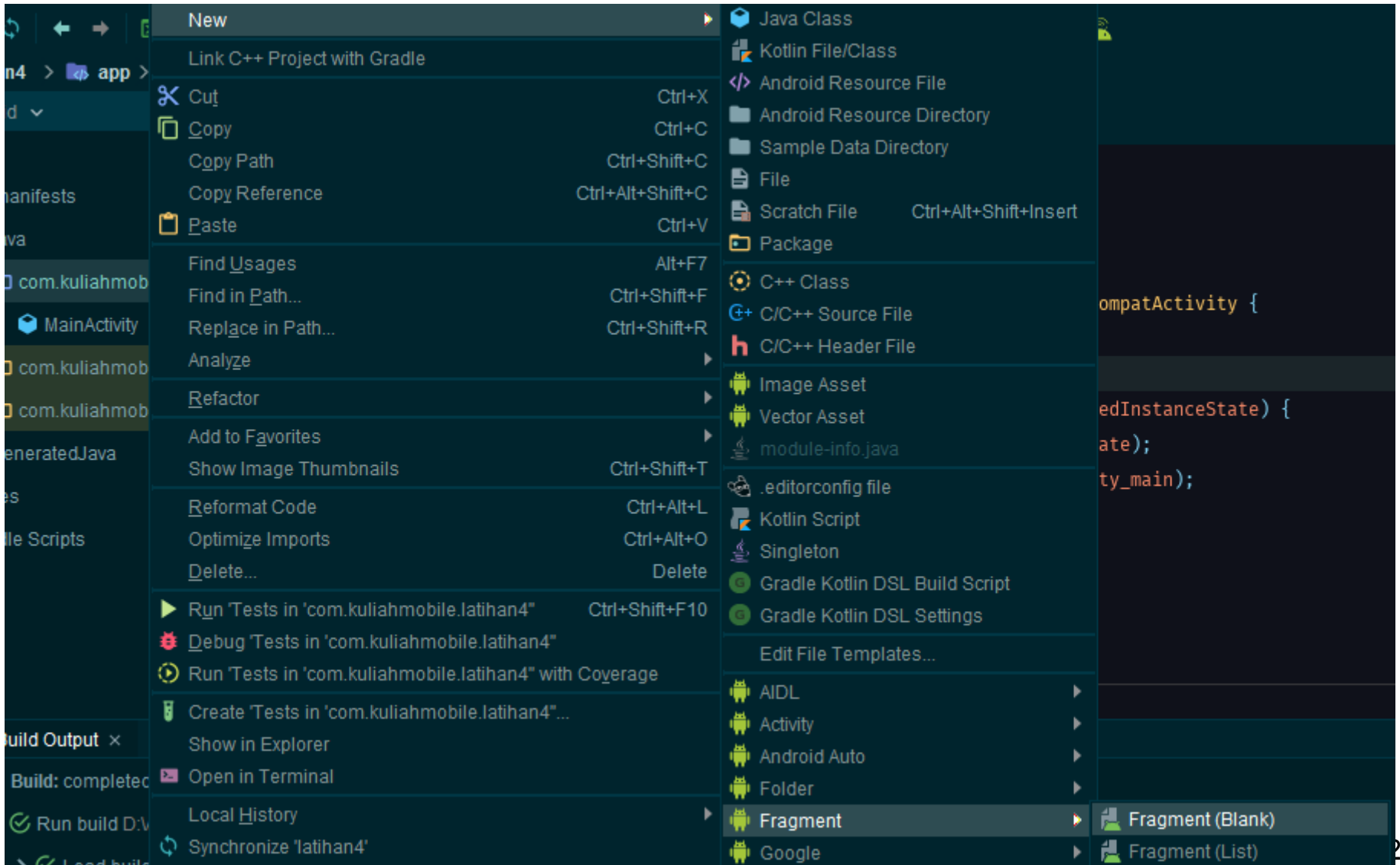
# Fragment dan Activity

- Fragment dipengaruhi secara langsung activity tempat fragment berada, apabila activity berada pada kondisi *pause* maka fragmentnya juga kondisinya *pause*, apabila kondisi activitinya *destroyed* maka fragment pada activity tersebut juga akan *destroyed*
- Tetapi ketika kondisi activity *resume* maka fragment di dalam activity tersebut akan memiliki kondisi sesuai daur hidup dari fragment

# Contoh penerapan fragment sederhana

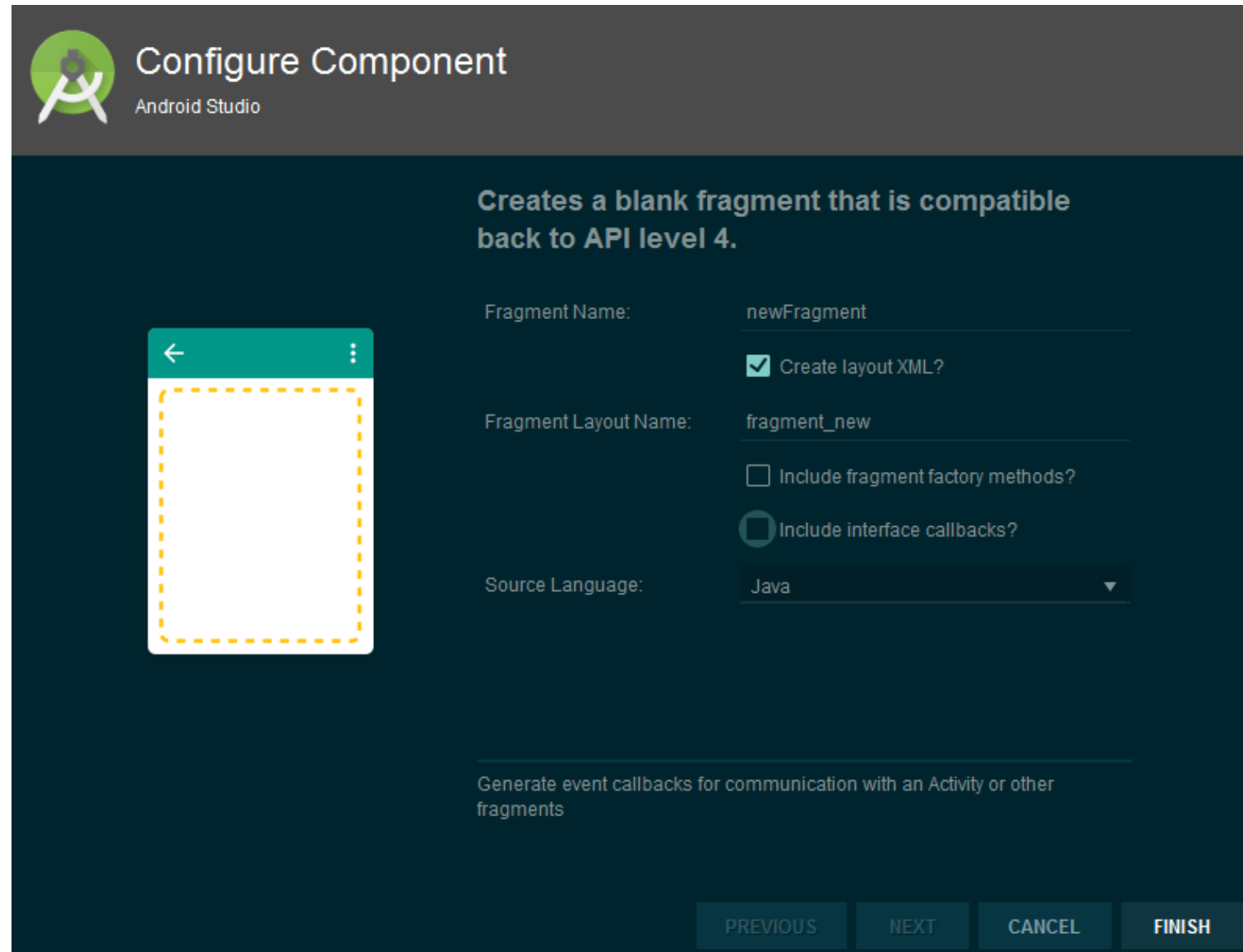
- Buat projek baru atau gunakan projek sebelumnya yang akan ditambahkan fragment
- Klik kanan pada projek tersebut pilih *new* kemudian *fragment*. Pilih blank fragment
- Pada projek ini diasumsikan pembaca sudah memahami maksud dari membuat objek baru, penggunaan method objek dan *overriding*, mekanisme berpindah antar activity serta perbedaan konseptual activity dengan fragment

# Membuat fragment baru



# Mengkonfigurasi fragment

- Pada Configure Component beri nama fragment newFragment
- Hilangkan centang pada include fragment dan include interface



**Configure Component**  
Android Studio

Creates a blank fragment that is compatible back to API level 4.

Fragment Name: newFragment

☒ Create layout XML?

Fragment Layout Name: fragment\_new

☐ Include fragment factory methods?

☐ Include interface callbacks?

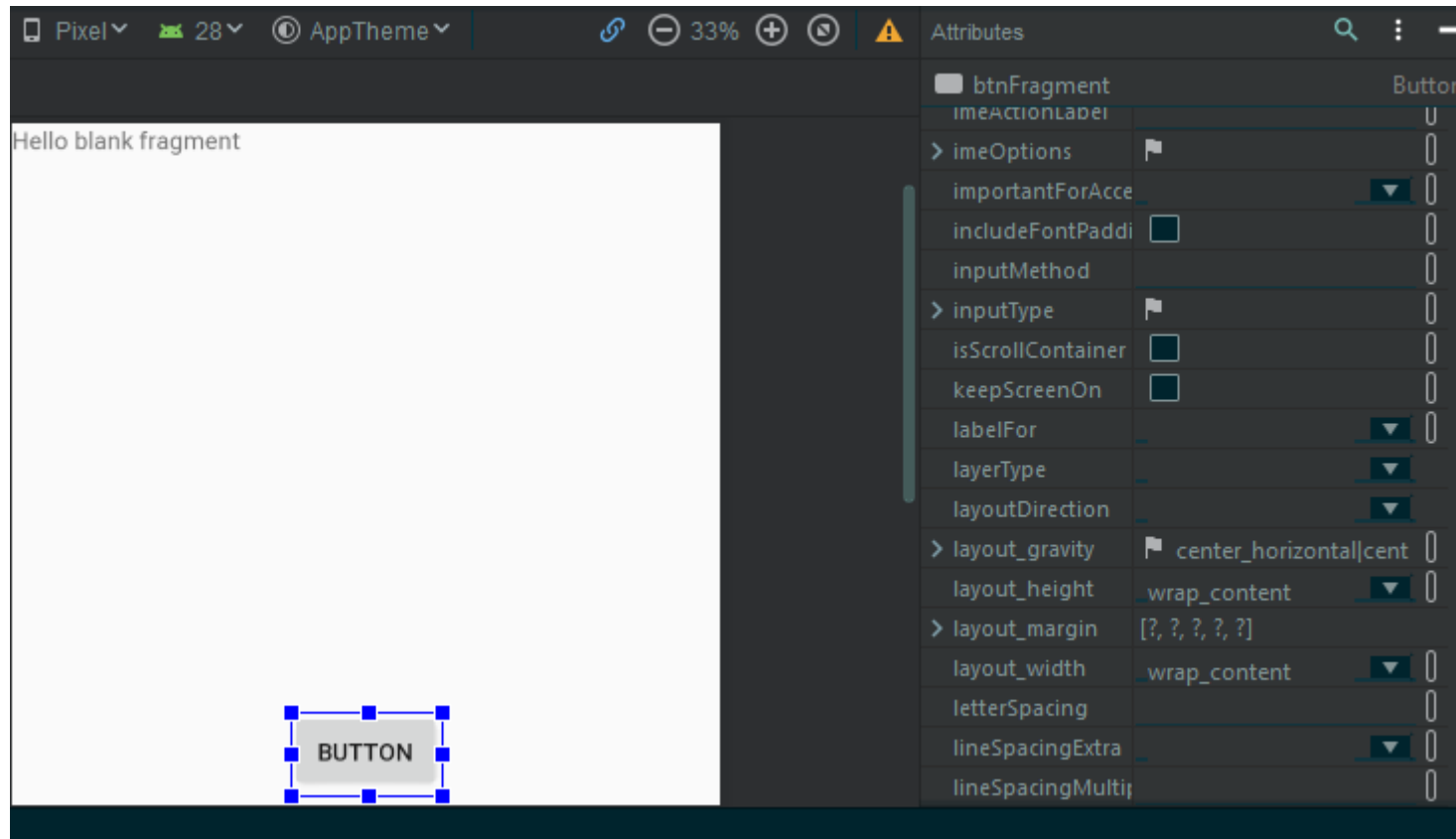
Source Language: Java

Generate event callbacks for communication with an Activity or other fragments

PREVIOUS NEXT CANCEL FINISH

# Menambahkan objek view pada fragment\_new.xml

- Tambahkan komponen view pada fragment\_new.xml seperti TextView dan button, beri nama id untuk button btnFragment



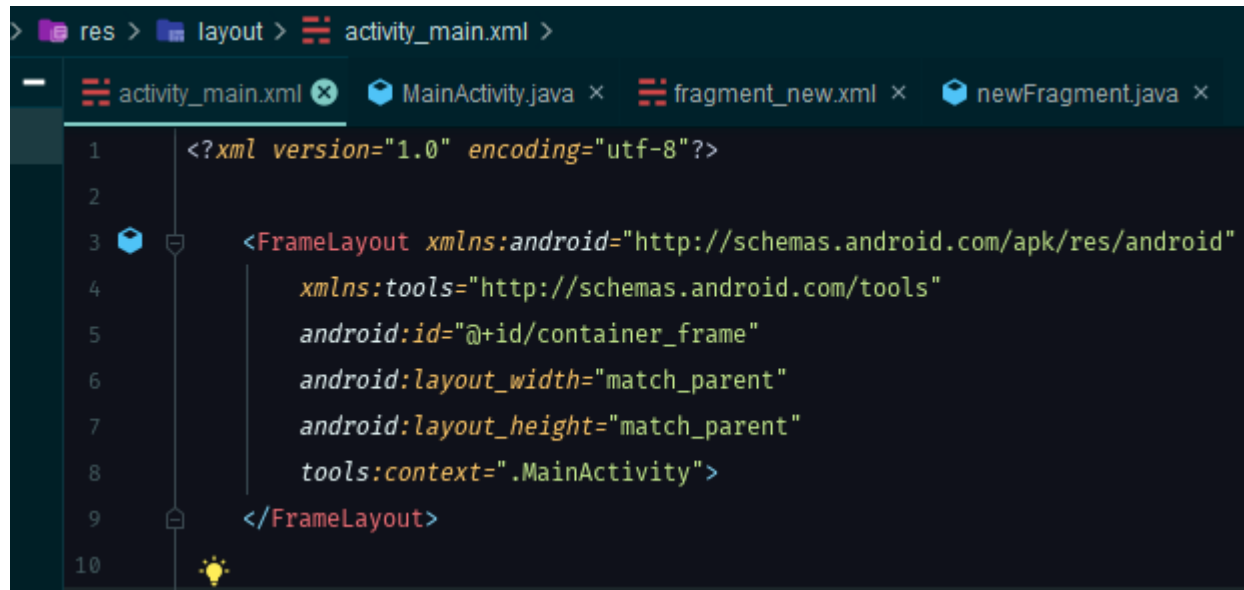


# Menambahkan objek view pada fragment\_new.xml(2)

- Kode Text fragment\_new.xml nya

```
activity_main.xml × MainActivity.java × fragment_new.xml × newFragment.java ×
1 <?xml version="1.0" encoding="utf-8"?>
2 <FrameLayout 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     tools:context=".newFragment"
8 >
9     <!-- TODO: Update blank fragment layout -->
10    <TextView
11        android:layout_width="match_parent"
12        android:layout_height="wrap_content"
13        android:text="Hello blank fragment" />
14    <Button
15        android:id="@+id/btnFragment"
16        android:layout_width="wrap_content"
17        android:layout_height="wrap_content"
18        android:layout_gravity="center_horizontal|center_vertical"
19        android:text="Button" />
```

# Membuat FrameLayout pada activity\_main.xml



```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/container_frame"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
</FrameLayout>
```

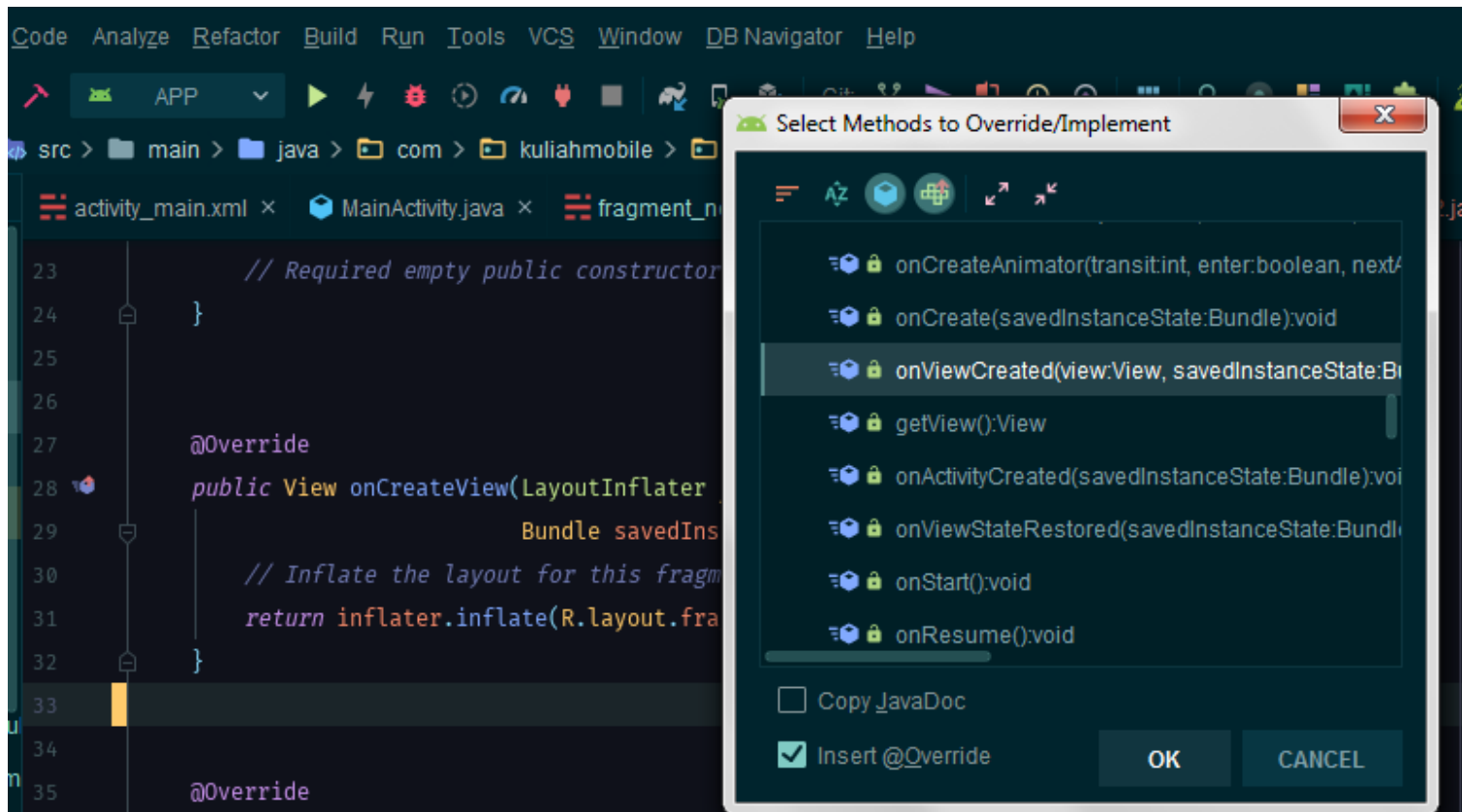
- Ubah kode xml pada activity\_main.xml dengan kode FrameLayout diatas
- Beri nama id “@+id/container\_frame

# Mendefinisikan dan mentransformasikan layout fragment

```
25      @Override
26      public View onCreateView(LayoutInflater inflater, ViewGroup container,
27                               Bundle savedInstanceState) {
28          // Inflate the layout for this fragment
29          return inflater.inflate(R.layout.fragment_new, container, attachToRoot: false);
30      }
```

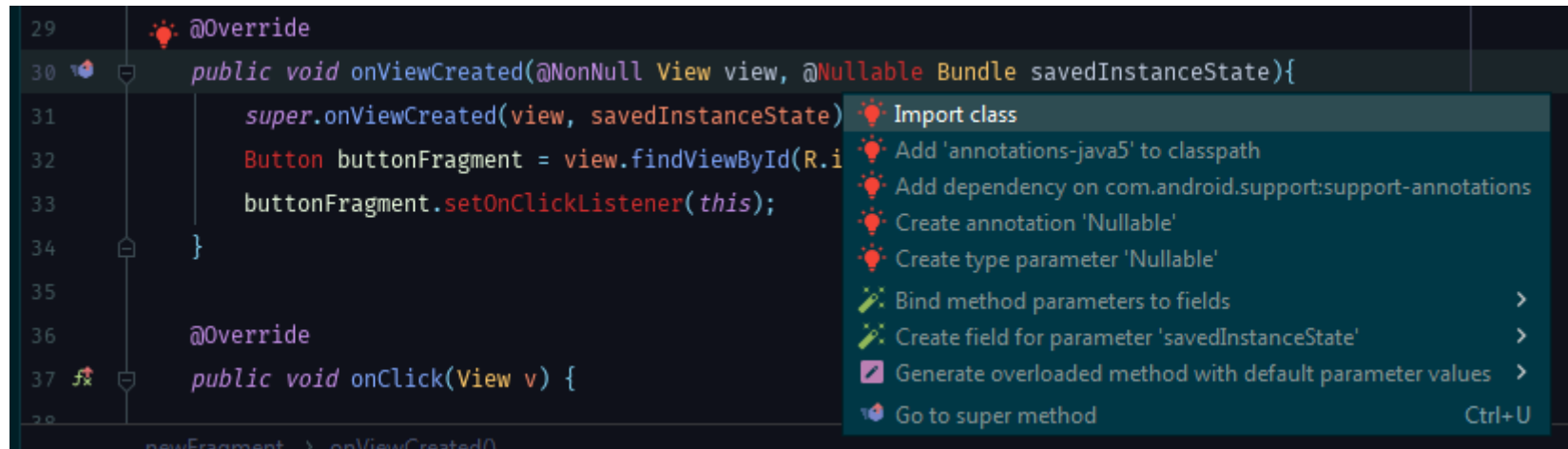
- Pada newFragment.java
  - Dengan menggunakan method inflate() definisikan dan transformasikan layout fragment yang berupa kode xml menjadi objek view atau dengan kata lain mengubah layout xml ke dalam bentuk objek viewgroup atau widget melalui pemanggilan method inflate(). Dengan cara pada method onCreateView tambahkan return inflater.inflate (R.layout.fragment\_new, container, false);

# Inisialisasi view dan action



- Buat method onCreateView() pada newFragment.java  
Untuk membuat method ini secara otomatis bisa dengan kombinasi Ctrl + o kemudian pada jendela Select Methods to Override/Implement pilih onCreateView()

# Inisialisasi view dan action(2)



```
29  @Override
30  public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState){
31      super.onViewCreated(view, savedInstanceState);
32      Button buttonFragment = view.findViewById(R.id.btnFragment);
33      buttonFragment.setOnClickListener(this);
34  }
35
36  @Override
37  public void onClick(View v) {
38      // ...
39  }
```

- Method onViewCreate ini akan bekerja setelah method onCreateView(). Setelah itu masih pada method onViewCreate kita inisialisasi view dan juga memberi action *listener*:
  - Dengan cara `Button buttonFragment = view.findViewById(R.id.btnFragment); buttonFragment.setOnClickListener(this);`

# Inisialisasi view dan action(3)

```
25      @Override
26      public View onCreateView(LayoutInflater inflater, ViewGroup container,
27                               Bundle savedInstanceState) {
28          // Inflate the layout for this fragment
29          return inflater.inflate(R.layout.fragment_new, container, attachToRoot: false);
30      }
31      @Override
32      public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState){
33          super.onViewCreated(view, savedInstanceState);
34          Button buttonFragment = view.findViewById(R.id.btnFragment);
35          buttonFragment.setOnClickListener(this);
36      }
```

- View pada Button buttonFragment = view.findViewById(R.id.btnFragment); menandakan btnFragment berada pada objek view di mana objek view berasal dari konversi fragment\_new.xml

# Inisialisasi view dan action(4)

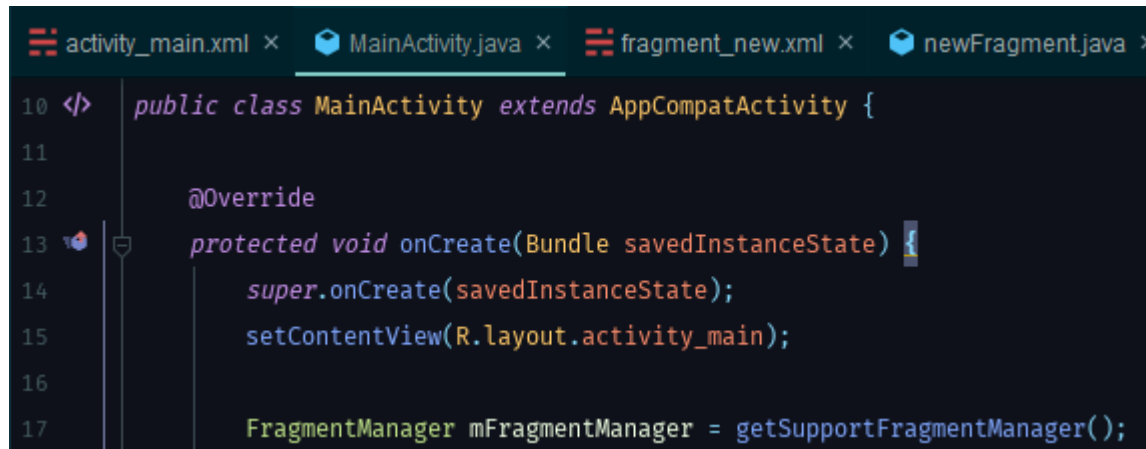


```
1 package com.kuliahmobile.latihan4;
2
3
4 import ...
5
6
7
8
9
10
11 /**
12  * A simple {@link Fragment} subclass.
13  */
14 public class newFragment extends Fragment implements View.OnClickListener {
15
16
17     public newFragment() {
18         // Required empty public constructor
19     }
```

- Implements View.OnClickListener

# Menampilkan objek fragment ke layar

- Mekanisme pemasangan objek `newFragment` sehingga dapat ditampilkan ke layar pengguna dilakukan dengan cara menambahkan beberapa kode berikut pada `MainActivity.java`
  1. Pada method `onCreate` setelah `setContentView` tambahkan `FragmentManager mFragmentManager = getSupportFragmentManager();` Disini kita menggunakan instance dari `FragmentManager` dimana fungsinya merupakan antarmuka untuk mengorganisir objek fragment yang terdapat didalam sebuah `Activity`.



```
activity_main.xml x MainActivity.java x fragment_new.xml x newFragment.java x
10 </> public class MainActivity extends AppCompatActivity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16
17         FragmentManager mFragmentManager = getSupportFragmentManager();
```



# Menampilkan objek fragment ke layar (2)

2. Membuat objek FragmentManager untuk melakukan operasi pada fragment seperti add(), replace(), commit().  
FragmentTransaction mFragmentTransaction = mFragmentManager.beginTransaction();

```
activity_main.xml × MainActivity.java × fragment_new.xml × newFragment.java ×
10 </> public class MainActivity extends AppCompatActivity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16
17         FragmentManager mFragmentManager = getSupportFragmentManager();
18         FragmentTransaction mFragmentTransaction = mFragmentManager.beginTransaction();
```

# Menampilkan objek fragment ke layar (3)

3. Membuat objek newFragment dengan cara newFragment mnewFragment = new newFragment(); Nantinya objek newFragment inilah yang akan digunakan oleh objek FragmentTransaction sebagai fragment yang ditambahkan ke dalam activity

```
activity_main.xml × MainActivity.java × fragment_new.xml × newFragment.java ×  
10 </> public class MainActivity extends AppCompatActivity {  
11  
12     @Override  
13     protected void onCreate(Bundle savedInstanceState) {  
14         super.onCreate(savedInstanceState);  
15         setContentView(R.layout.activity_main);  
16  
17         FragmentManager mFragmentManager = getSupportFragmentManager();  
18         FragmentTransaction mFragmentTransaction = mFragmentManager.beginTransaction();  
19         newFragment mnewFragment = new newFragment();
```

# Menampilkan objek fragment ke layar (4)

4. Manipulasi penambahan fragment kedalam activity dengan menggunakan `mFragmentManager.add`
5. Meminta objek `mFragmentManager` melakukan pemasangan objek secara asynchronous untuk ditampilkan ke antar muka pengguna (*user interface*)



```
activity_main.xml x MainActivity.java x fragment_new.xml x newFragment.java x
10 </> public class MainActivity extends AppCompatActivity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16
17         FragmentManager mFragmentManager = getSupportFragmentManager();
18         FragmentTransaction mFragmentTransaction = mFragmentManager.beginTransaction();
19         newFragment mnewFragment = new newFragment();
20         Fragment fragment = mFragmentManager.findFragmentByTag(newFragment.class.getSimpleName());
21         if(!(fragment instanceof newFragment)){
22             mFragmentTransaction.add(R.id.container_frame,mnewFragment,newFragment.class.getSimpleName());
23             mFragmentTransaction.commit();
24         }
25     }
```



# Source Code dan Slide

- [https://github.com/gamawis/pemrogramanberbasismobile\\_fragment\\_2](https://github.com/gamawis/pemrogramanberbasismobile_fragment_2)