



MODUL 4

Intent

A. TUJUAN

Pada modul 4, akan membahas tentang komponen-komponen yang terdapat dalam Intent, dengan tujuan agar mahasiswa:

- Mahasiswa dapat memahami fungsi intent
- Mahasiswa dapat menggunakan fungsi intent pada Android Studio

B. TEORI

1. Intent

Semua aktivitas Android dimulai atau diaktifkan dengan intent. Intent adalah objek pesan yang membuat permintaan yang akan digunakan oleh waktu proses Android untuk memulai aktivitas atau komponen aplikasi lainnya di aplikasi kita atau di beberapa aplikasi lainnya. Kita tidak bisa memulai aktivitas itu sendiri. Bila aplikasi pertama kali dimulai dari layar utama perangkat, waktu proses Android akan mengirimkan intent ke aplikasi kita untuk memulai aktivitas utama aplikasi (yang didefinisikan dengan aksi MAIN dan kategori LAUNCHER di Manifes Android). Untuk memulai aktivitas lain di aplikasi kita, atau meminta tindakan untuk dilakukan oleh beberapa aktivitas lain yang tersedia di perangkat, bangunlah intent sendiri dengan kelas Intent dan panggil metode *startActivity()* untuk mengirim intent itu.

Selain untuk memulai aktivitas, intent juga digunakan untuk meneruskan data di antara aktivitas. Bila membuat intent untuk memulai aktivitas baru, kita bisa menyertakan informasi tentang data yang diinginkan untuk mengoperasikan aktivitas baru itu. Jadi, misalnya, aktivitas email yang menampilkan daftar pesan bisa mengirim intent ke aktivitas yang menampilkan pesan itu. Aktivitas tampilan memerlukan data tentang pesan yang akan ditampilkan, dan kita bisa menyertakan data itu di intent. Dalam bab ini, kita akan mempelajari tentang penggunaan intent bersama aktivitas, namun intent juga digunakan untuk memulai layanan dan penerima.

2. Tipe Intent

a. Explicit Intent

menetapkan aktivitas penerima (atau komponen lainnya) melalui nama kelas yang benar-benar memenuhi syarat di aktivitas tersebut. Gunakan explicit intent untuk memulai komponen di aplikasi milik sendiri (misalnya, untuk beralih layar di antarmuka pengguna), karena kita sudah mengetahui paket dan nama kelas komponen itu.

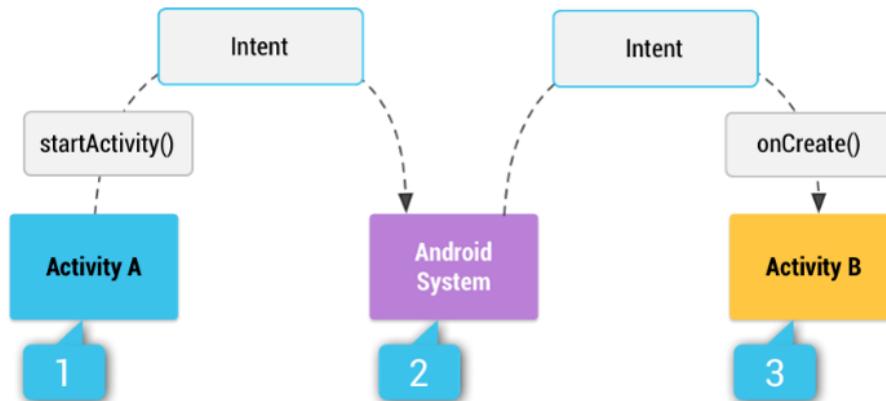
b. Implicit Intent

tidak menetapkan aktivitas tertentu atau komponen lainnya untuk menerima intent. Sebagai gantinya, kita mendeklarasikan aksi umum yang dilakukan di intent tersebut. Sistem Android mencocokkan permintaan kita dengan aktivitas atau komponen lainnya yang bisa menangani aksi permintaan tersebut.

3. Objek dan Bidang Intent

Objek Intent adalah instance kelas Intent. Untuk explicit intent, bidang kunci suatu intent menyertakan yang berikut ini:

- Kelas aktivitas (untuk explicit intent). Ini adalah nama kelas aktivitas atau komponen lainnya yang akan menerima intent, misalnya, `com.example.SampleActivity.class`. Gunakan konstruktor intent atau metode `setComponent()`, `setComponentName()` atau `setClassName()` intent untuk menetapkan kelas.
- Data intent. Bidang data intent berisi referensi ke data yang kita inginkan untuk mengoperasikan aktivitas penerima, sebagai objek Uri.
- Ekstra intent. Ini adalah pasangan nilai-kunci yang membawa informasi yang diperlukan aktivitas penerima untuk melakukan aksi yang diminta.
- Flag intent. Ini adalah bit metadata tambahan, yang didefinisikan oleh kelas Intent. Flag dapat menginstruksikan sistem Android tentang cara meluncurkan aktivitas atau cara memperlakukan aktivitas setelah diluncurkan.



Gambar ilustrasi Intent mengaktifkan Activity pada aplikasi

Keterangan:

1. Activity A membuat Intent untuk melakukan sebuah Action dengan data object yang ada.
2. Sistem Android akan mencari aplikasi yang cocok Berdasarkan Intent filter
3. Activity pada aplikasi yang cocok akan ditampilkan.

4. Memulai aktivitas dengan explicit intent

Untuk memulai aktivitas tertentu dari aktivitas lain, gunakan explicit intent dan metode `startActivity()`. Explicit intent menyertakan nama kelas yang benar-benar memenuhi syarat untuk aktivitas atau komponen lainnya di objek Intent. Semua bidang intent lainnya bersifat opsional, dan nol secara default.

Misalnya, jika kita ingin memulai `ShowMessageActivity` untuk menampilkan pesan tertentu di aplikasi email, gunakan kode seperti ini.



```
Intent messageIntent = new Intent(this, ShowMessageActivity.class);
startActivity(messageIntent);
```

Konstruktor Intent menggunakan dua argumen untuk explicit intent.

- Konteks aplikasi. Dalam contoh ini, kelas aktivitas menyediakan materi (di sini, `this`).
- Komponen tertentu untuk dimulai (`ShowMessageActivity.class`).

Gunakan metode `startActivity()` bersama objek intent baru sebagai satu-satunya argumen. Metode `startActivity()` mengirim intent ke sistem Android, yang meluncurkan kelas `ShowMessageActivity` atas nama aplikasi kita. Aktivitas baru muncul pada layar dan aktivitas pembuatnya dihentikan sementara.

Aktivitas yang dimulai tetap pada layar hingga pengguna mengetuk tombol kembali pada perangkat, pada saat itu aktivitas ditutup dan diklaim kembali oleh sistem, dan aktivitas yang menghasilkannya akan dilanjutkan. kita juga bisa menutup aktivitas yang dimulai secara manual sebagai respons terhadap aksi pengguna (seperti klik tombol) dengan metode `finish()`:

```
public void closeActivity (View view) {
    finish();
}
```

5. Meneruskan data di antara aktivitas dengan intent

Selain untuk memulai satu aktivitas dari aktivitas lain, kita juga menggunakan intent untuk meneruskan informasi di antara aktivitas. Objek intent yang kita gunakan untuk memulai aktivitas bisa menyertakan data intent (URI objek untuk bertindak), atau ekstra intent, yang merupakan bit data tambahan yang mungkin diperlukan aktivitas.

Di aktivitas (pengirim) pertama:

- 1.Buat objek Intent.
- 2.Masukkan data atau ekstra ke dalam intent itu.
- 3.Mulailah aktivitas baru dengan `startActivity()`.

Di aktivitas (penerima) kedua:

- 1.Dapatkan objek intent yang digunakan memulai aktivitas.
- 2.Ambil data atau ekstra dari objek Intent.

6. Tambahkan Data ke Intent

Untuk menambahkan data ke explicit intent dari aktivitas pembuatnya, buat objek intent seperti yang kita lakukan sebelumnya:

```
Intent messageIntent = new Intent(this, ShowMessageActivity.class);
```

Gunakan metode `setData()` bersama objek Uri untuk menambahkan URL itu ke intent. Beberapa contoh penggunaan `setData()` bersama URI:



```
// A web page URL
messageIntent.setData(Uri.parse("http://www.google.com"));
// a Sample file URI
messageIntent.setData(Uri.fromFile(new File("/sdcard/sample.jpg")));
// A sample content: URI for your app's data model
messageIntent.setData(Uri.parse("content://mysample.provider/data"));
// Custom URI
messageIntent.setData(Uri.parse("custom:" + dataID + buttonId));
```

Perlu diingat bahwa bidang data hanya bisa berisi URI tunggal; jika kita memanggil *setData()* beberapa kali, hanya nilai terakhir yang akan digunakan. Gunakan ekstra intent untuk menyertakan informasi tambahan (termasuk URI.). Setelah menambahkan data, kita bisa memulai aktivitas dengan intent seperti biasanya.

```
startActivity(messageIntent);
```

7. Tambahkan Ekstra ke Intent

Untuk menambahkan ekstra intent ke explicit intent dari aktivitas pembuatnya:

- Tentukan kunci yang akan digunakan untuk informasi yang ingin dimasukkan ke dalam ekstra, atau definisikan sendiri. Setiap bagian informasi memerlukan kunci unik.
- Gunakan metode *putExtra()* untuk menambahkan pasangan kunci/nilai ke ekstra intent. Secara opsional kita bisa membuat objek Bundle, menambahkan data ke bundel, kemudian menambahkan bundel ke intent.

Kelas Intent menyertakan sejumlah kunci ekstra intent yang bisa digunakan, didefinisikan sebagai konstanta yang dimulai dengan kata EXTRA_. Misalnya, kita bisa menggunakan Intent.EXTRA_EMAIL untuk menunjukkan larik alamat email (sebagai string), atau Intent.EXTRA_REFERRER untuk menetapkan informasi tentang aktivitas pembuat yang mengirim intent tersebut.

kita juga bisa mendefinisikan kunci ekstra intent milik sendiri. Secara konvensional kita mendefinisikan kunci ekstra intent sebagai variabel-variabel statis dengan nama yang dimulai kata EXTRA_. Untuk menjamin kunci tersebut unik, nilai string kunci itu sendiri harus diawali dengan nama kelas yang benar-benar memenuhi syarat aplikasi. Misalnya:

```
public final static String EXTRA_MESSAGE = "com.example.mysampleapp.MESSAGE";
public final static String EXTRA_POSITION_X = "com.example.mysampleapp.X";
public final static String EXTRA_POSITION_Y = "com.example.mysampleapp.Y";
```

Buat objek intent (jika belum ada):

```
Intent messageIntent = new Intent(this, ShowMessageActivity.class);
```

Gunakan metode *putExtra()* bersama kunci untuk memasukkan data ke dalam ekstra intent. Kelas Intent mendefinisikan banyak metode *putExtra()* untuk jenis data yang berbeda:

```
messageIntent.putExtra(EXTRA_MESSAGE, "this is my message");
messageIntent.putExtra(EXTRA_POSITION_X, 100);
messageIntent.putExtra(EXTRA_POSITION_Y, 500);
```



Atau, kita bisa membuat bundel baru dan mengisinya dengan ekstra intent. Bundel mendefinisikan banyak metode "put" untuk jenis data primitif yang berbeda serta objek yang mengimplementasikan antarmuka Parcelable Android atau Serializable Java.

```
Bundle extras = new Bundle();
extras.putString(EXTRA_MESSAGE, "this is my message");
extras.putInt(EXTRA_POSITION_X, 100);
extras.putInt(EXTRA_POSITION_Y, 500);
```

Setelah kita mengisi bundel tersebut, tambahkan ke intent dengan metode *putExtras()* (perhatikan "s" di Extras):

```
messageIntent.putExtras(extras);
```

Mulai aktivitas dengan intent seperti biasa:

```
startActivity(messageIntent);
```

8. Ambil data dari intent di aktivitas yang dimulai

Bila kita memulai aktivitas bersama intent, aktivitas yang telah dimulai akan memiliki akses ke intent dan data yang dimuatnya. Untuk mengambil intent yang digunakan untuk memulai aktivitas (atau komponen lain), gunakan metode *getIntent()*:

```
Intent intent = getIntent();
```

Gunakan *getData()* untuk mendapatkan URI dari intent itu:

```
Uri locationUri = getData();
```

Untuk mendapatkan ekstra dari intent, kita perlu mengetahui kunci untuk pasangan kunci/nilai. kita bisa menggunakan ekstra Intent standart jika telah menggunakannya, atau bisa menggunakan kunci yang didefinisikan di aktivitas pembuatnya (jika didefinisikan sebagai publik). Gunakan salah satu metode *getExtra()* untuk mengekstrak data ekstra dari objek intent:

```
String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
int positionX = intent.getIntExtra(MainActivity.EXTRA_POSITION_X);
int positionY = intent.getIntExtra(MainActivity.EXTRA_POSITION_Y);
```

Atau kita bisa mendapatkan seluruh bundel ekstra dari intent dan mengekstrak nilai dengan beragam metode Bundle:

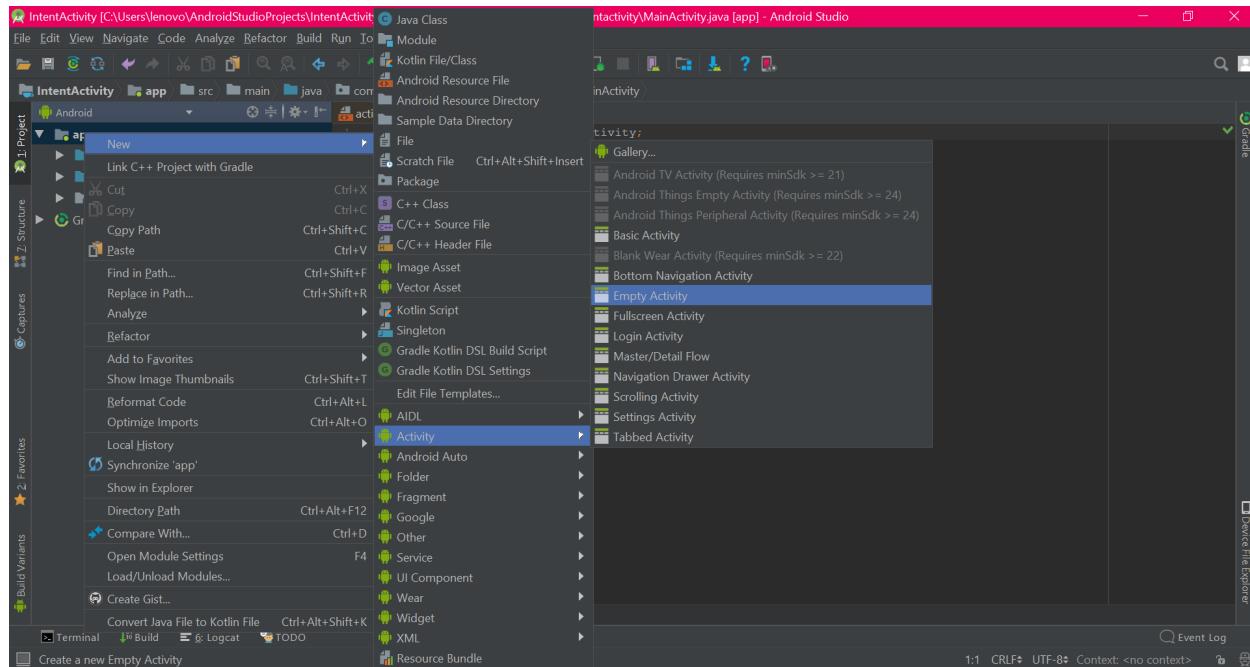
```
Bundle extras = intent.getExtras();
String message = extras.getString(MainActivity.EXTRA_MESSAGE);
```



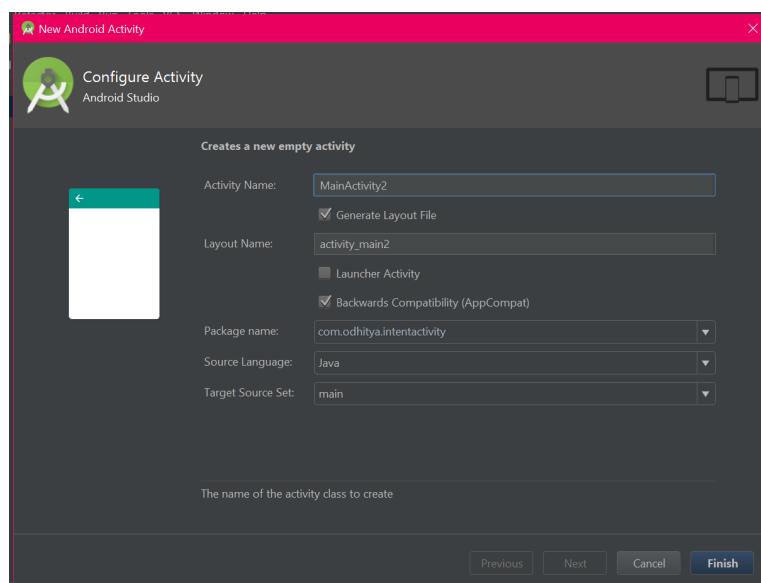
C. Latihan Praktikum

1) Intent Activity

1. Pertama buka project **Android Studio** kalian.
2. Disini langkah pertama adalah membuat Activity baru yang terdiri dari file .java dan file .xml.
3. Pada folder java atau res kalian bisa klik kanan pilih **New**, Pilih **Activity**, lalu **Empty Activity**.



4. Pada bagian **New Android Activity** kalian isikan nama activity atau file java dan nama file layout. Sebagai contoh membuat file class Activity dengan nama *MainActivity2.java* dan *activity_main2.xml*





5. Lalu buatlah button pada *activity_main.xml* seperti dibawah ini.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingTop="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:paddingBottom="10dp"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/ButtonPindah"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Klik Disini Untuk Pindah"/>
</RelativeLayout>

```

6. Buatlah atribut button pada *MainActivity.java* seperti dibawah ini.

```

package com.odhitya.intentactivity;

import ...

public class MainActivity extends AppCompatActivity {
    private Button myButton;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        myButton = (Button) findViewById(R.id.ButtonPindah);
        myButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(getApplicationContext(),MainActivity2.class);
                startActivity(i);
            }
        });
    }
}

```

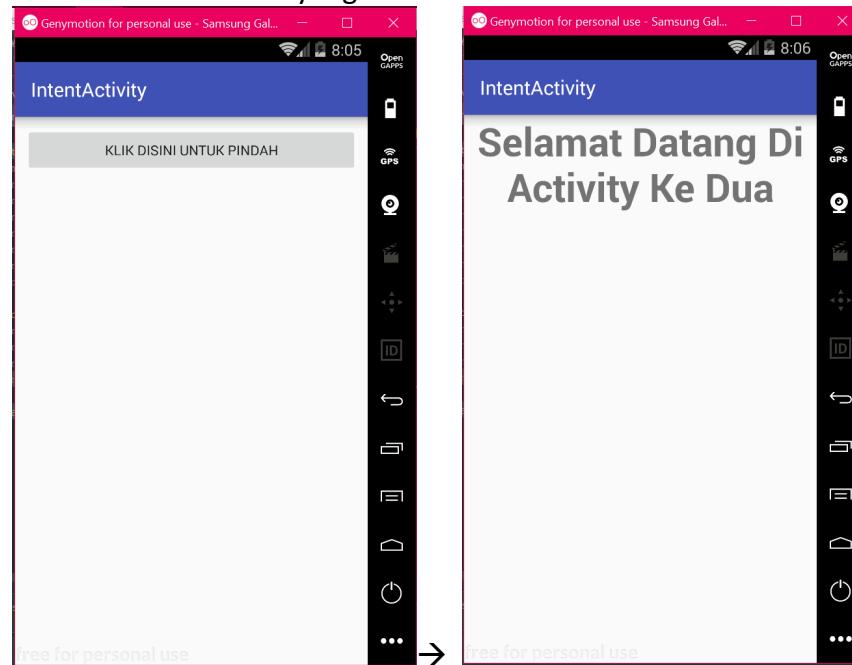
7. Kemudian buatlah sebuah text pada *activity_main2.xml* sebagai tanda bahwa telah berpindah activity.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity2">
    <TextView
        android:id="@+id/welcome"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Selamat Datang Di Activity Ke Dua"
        android:textSize="40dp"
        android:textAlignment="center"
        android:textStyle="bold"/>
</RelativeLayout>

```

8. Lakukan tes program apakah sudah berjalan dengan baik, dibawah adalah hasil perpindahan activity jika kita menekan button yang tersedia.



2) Pindah Data Intent

1. Lanjutkan memakai project sebelumnya.
2. Tambahkan data yang akan dikirimkan ke *MainActivity2.java* pada *MainActivity.java*

```

1 package com.odhitya.intentactivity;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity {
6     private Button myButton;
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main);
11        myButton = (Button) findViewById(R.id.ButtonPindah);
12        myButton.setOnClickListener(new View.OnClickListener() {
13            @Override
14            public void onClick(View view) {
15                Intent i = new Intent(getApplicationContext(),MainActivity2.class);
16                i.putExtra("matkul", "Pemrograman Perangkat Mobile");
17                startActivity(i);
18            }
19        });
20    }
21 }
22
23
24
25
26

```

PRAKTIKUM PEMROGAMAN MOBILE



3. Buatlah tempat berupa TextView pada layout *activity_main2.xml* untuk meletakkan data yang dikirimkan dari *MainActivity.java*

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity2">

    <TextView
        android:id="@+id/welcome"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Selamat Datang Di Activity Ke Dua"
        android:textSize="40dp"
        android:textAlignment="center"
        android:textStyle="bold"
    />

    <TextView
        android:id="@+id/matkul"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/welcome"
        android:text="Nama Matkul Disini"
        android:textSize="20dp"
        android:paddingTop="40dp"
        android:layout_centerHorizontal="true"
        android:textStyle="italic"
    />

</RelativeLayout>
```

4. Kemudian kita melakukan pemanggilan data tersebut dari *MainActivity2.java* dengan membuat atribut TextView.

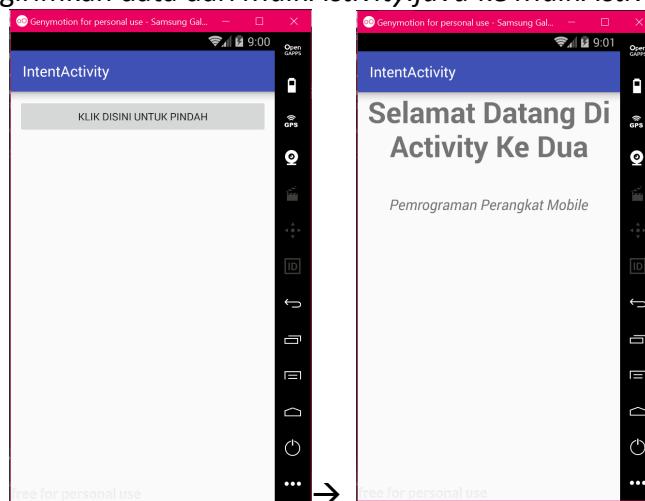
```
package com.odhitya.intentactivity;

import ...

public class MainActivity2 extends AppCompatActivity {

    private TextView myTextView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        myTextView = (TextView) findViewById(R.id.matkul);
        Intent i = getIntent();
        String matkul = i.getStringExtra("matkul");
        myTextView.setText(matkul);
    }
}
```

5. Lakukan tes program apakah sudah berjalan dengan baik, dibawah ini adalah hasil jika kita telah berhasil mengirimkan data dari *MainActivity.java* ke *MainActivity2.java*





3) Pindah Data Intent dengan menggunakan Inputan

1. Lanjutkan menggunakan project sebelumnya
2. Tambahkan data yang akan dikirimkan ke *MainActivity2.java* pada *MainActivity.java* yang menggunakan inputan data text dengan menambahkan atribut dari *EditText*.

```

activity_main.xml x MainActivity.java x activity_main2.xml x MainActivity2.java x
1 package com.odhitya.intentactivity;
2
3 import ...
4
5
6 public class MainActivity extends AppCompatActivity {
7
8     private Button myButton;
9     private EditText myKampus;
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15         myButton = (Button) findViewById(R.id.buttonPindah);
16         myKampus = (EditText) findViewById(R.id.NamaKampus);
17         myButton.setOnClickListener(new View.OnClickListener() {
18             @Override
19             public void onClick(View view) {
20                 Intent i = new Intent(getApplicationContext(),MainActivity2.class);
21                 i.putExtra("matkul", "Pemrograman Perangkat Mobile");
22                 i.putExtra("kampus", myKampus.getText().toString());
23                 startActivity(i);
24             }
25         });
26     }
27 }

```

3. Buatlah *TextView* dan *EditText* pada layout *activity_main.xml* sebagai inputan data yang akan dikirimkan ke *MainActivity2.java*

```

activity_main.xml x MainActivity.java x activity_main2.xml x MainActivity2.java x
1 <Button
2     android:id="@+id/ButtonPindah"
3     android:layout_width="match_parent"
4     android:layout_height="wrap_content"
5     android:text="Pindah ke Matkul Disini"
6     android:layout_below="@+id/NamaKampusText"
7     />
8
9 <TextView
10    android:id="@+id/NamaKampusText"
11    android:layout_width="wrap_content"
12    android:layout_height="wrap_content"
13    android:text="Nama Kampus :"
14    android:layout_marginTop="20dp"
15    android:layout_marginBottom="20dp"
16    android:textStyle="bold"
17    />
18
19 <EditText
20    android:id="@+id>NamaKampus"
21    android:layout_width="wrap_content"
22    android:layout_height="wrap_content"
23    android:layout_toRightOf="@+id>NamaKampusText"
24    android:ems="10"
25    android:inputType="text"
26    />
27
28 </RelativeLayout>

```

4. Buatlah tempat berupa *TextView* pada layout *activity_main2.xml* untuk meletakkan data berupa inputan yang dikirimkan dari *MainActivity.java*

```

activity_main.xml x MainActivity.java x activity_main2.xml x MainActivity2.java x
1 <TextView
2     android:id="@+id/matkul"
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     android:layout_below="@+id/welcome"
6     android:text="Nama Matkul Disini"
7     android:textSize="20dp"
8     android:paddingTop="40dp"
9     android:layout_centerHorizontal="true"
10    android:textStyle="italic"
11    />
12
13 <TextView
14    android:id="@+id/kampus"
15    android:layout_width="wrap_content"
16    android:layout_height="wrap_content"
17    android:layout_below="@+id/matkul"
18    android:text="Nama Kampus Disini"
19    android:textSize="20dp"
20    android:paddingTop="40dp"
21    android:layout_centerHorizontal="true"
22    android:textStyle="italic"
23    android:textAppearance="?android:attr/textAppearanceMedium"
24    />
25
26 </RelativeLayout>

```

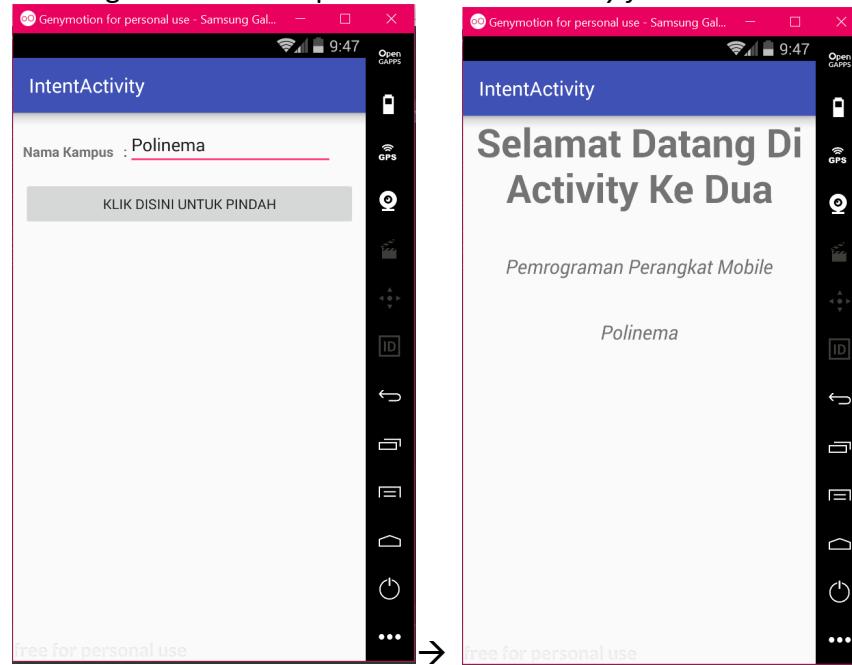
PRAKTIKUM PEMROGAMAN MOBILE



5. Kemudian kita melakukan pemanggilan data tersebut dari *MainActivity2.java* dengan membuat atribut *TextView* yang kedua.

```
activity_main.xml x MainActivity.java x activity_main2.xml x MainActivity2.java x
1 package com.odhitya.intentactivity;
2
3 import ...
4
5 public class MainActivity2 extends AppCompatActivity {
6
7     private TextView myTextView, myTextView2;
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main2);
13        myTextView = (TextView) findViewById(R.id.matkul);
14        myTextView2 = (TextView) findViewById(R.id.kampus);
15        Intent i = getIntent();
16        String matkul = i.getStringExtra("matkul");
17        String kampus = i.getStringExtra("kampus");
18        myTextView.setText(matkul);
19        myTextView2.setText(kampus);
20
21    }
22
23}
24
```

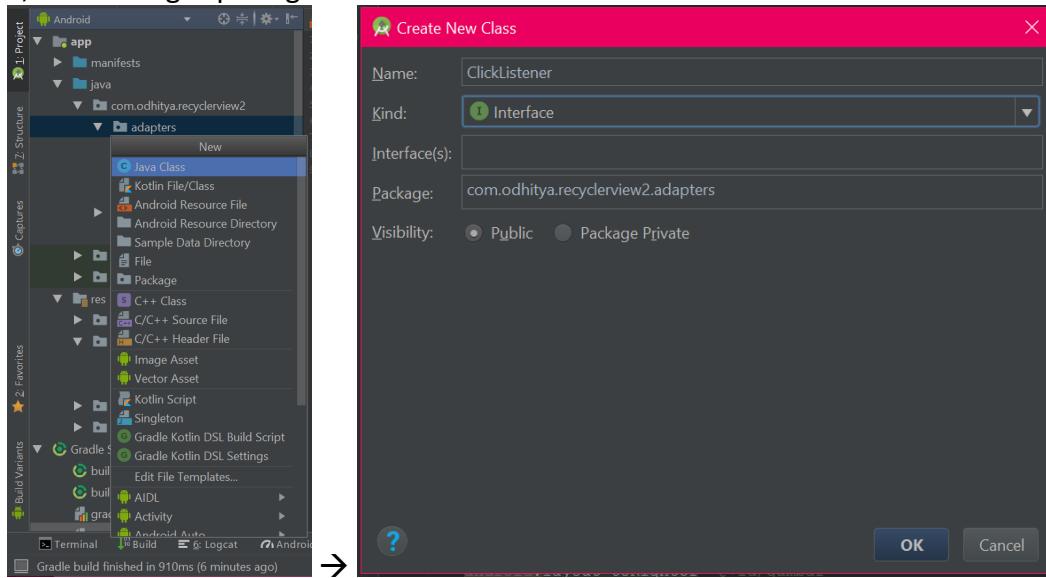
6. Lakukan tes program apakah sudah berjalan dengan baik, dibawah ini adalah hasil jika kita telah berhasil mengirimkan data inputan dari *MainActivity.java* ke *MainActivity2.java*



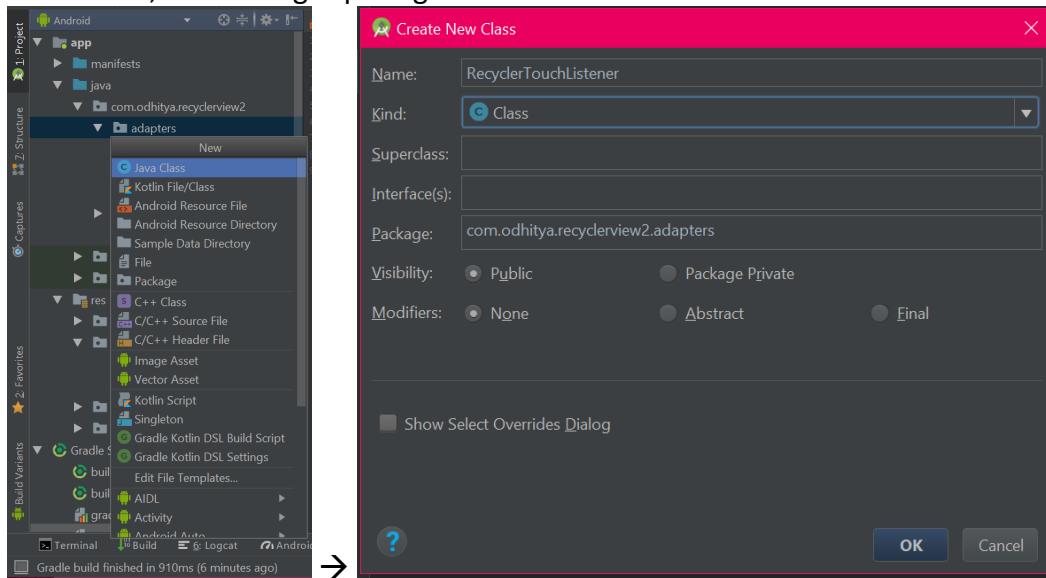


4) Membuat Event Click pada RecyclerView

1. Lanjutkan / buatlah sebuah recycler view yang ada pada modul recycler view.
2. Tambahkan sebuah interface pada package adapters dengan cara klik **Alt+Insert**, pilih **Java Class**, lalu setting seperti gambar dibawah ini.



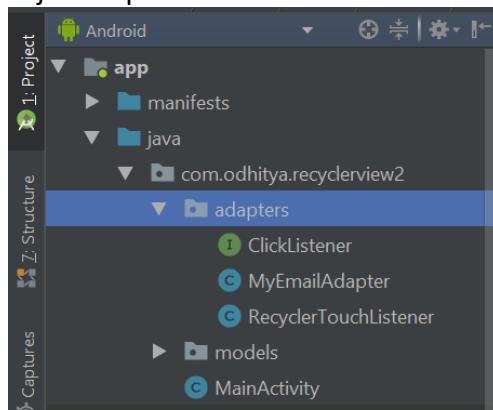
3. Kemudian buatlah sebuah Java Class pada package adapters dengan cara klik **Alt+Insert**, pilih **Java Class**, lalu setting seperti gambar dibawah ini.



PRAKTIKUM PEMROGAMAN MOBILE



4. Maka akan terbentuk 2 file java seperti berikut.



5. Pada *ClickListener.java* tambahkan script seperti gambar dibawah ini.

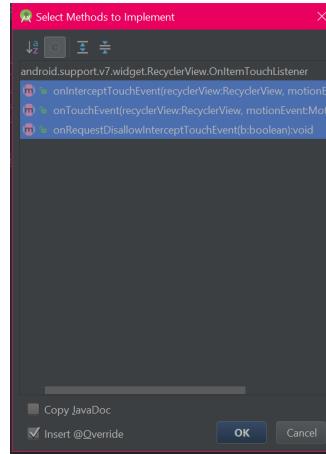
```
activity_main.xml x list_email.xml x ClickListener.java x RecyclerTouchListener.java x
1 package com.odhitya.recyclerview2.adapters;
2
3 import android.view.View;
4
5 public interface ClickListener {
6     void onClick(View view, int position);
7     void onLongClick(View view, int position);
8 }
9
```

6. Kemudian pada *RecyclerTouchListener.java* buatlah sebuah class yang dapat meng-implement class lain seperti script dibawah ini. Setelah itu tekan **Alt+Enter** untuk meng-implement methods. Akan muncul seperti dibawah ini lalu pilih semua dan klik **OK**. Selanjutnya akan muncul override hasil implement methods tersebut.

```
activity_main.xml x list_email.xml x ClickListener.java x RecyclerTouchListener.java x MainActivity.java x MyEmailAdapter.java x
1 package com.odhitya.recyclerview2.adapters;
2
3 import android.content.Context;
4 import android.support.v7.widget.RecyclerView;
5 import android.view.GestureDetector;
6 import android.view.MotionEvent;
7 import android.view.View;
8
9
10 public class RecyclerTouchListener implements RecyclerView.OnItemTouchListener {
```

A context menu is open over the 'implements RecyclerView.OnItemTouchListener' line. The menu items are:

- Implement methods
- Make 'RecyclerTouchListener' abstract
- Add import for 'android.support.v7.widget.RecyclerView.OnItemTouchListener'
- Create Test
- Create subclass
- Unimplement Interface
- Annotate interface 'OnItemTouchListener' as @Deprecated



7. Lanjutkan untuk menuliskan script pada *RecyclerTouchListener.java* dengan membuatkan 2 variable class dan constructornya seperti dibawah ini.

```

11  public class RecyclerTouchListener implements RecyclerView.OnItemTouchListener {
12
13     private GestureDetector gestureDetector;
14     private ClickListener clickListener;
15
16     public RecyclerTouchListener(Context context, final RecyclerView recyclerView, final ClickListener clickListener) {
17         this.clickListener = clickListener;
18         gestureDetector = new GestureDetector(context, new GestureDetector.SimpleOnGestureListener() {
19             @Override
20             public boolean onSingleTapUp(MotionEvent e) {
21                 return true;
22             }
23
24             @Override
25             public void onLongPress(MotionEvent e) {
26                 View child = recyclerView.findChildViewUnder(e.getX(), e.getY());
27                 if (child != null && clickListener != null) {
28                     clickListener.onLongClick(child, recyclerView.getChildPosition(child));
29                 }
30             }
31         });
32     }

```

8. Setelah itu lengkapi script pada 3 override yang ada di bawah hasil dari meng-implement methods (langkah no. 6)

```

34     @Override
35     public boolean onInterceptTouchEvent(RecyclerView rv, MotionEvent e) {
36         View child = rv.findChildViewUnder(e.getX(), e.getY());
37         if (child != null && clickListener != null && gestureDetector.onTouchEvent(e)) {
38             clickListener.onClick(child, rv.getChildPosition(child));
39         }
40         return false;
41     }
42
43     @Override
44     public void onTouchEvent(RecyclerView rv, MotionEvent e) {
45     }
46
47     @Override
48     public void onRequestDisallowInterceptTouchEvent(boolean disallowIntercept) {
49     }
50
51 }
52

```

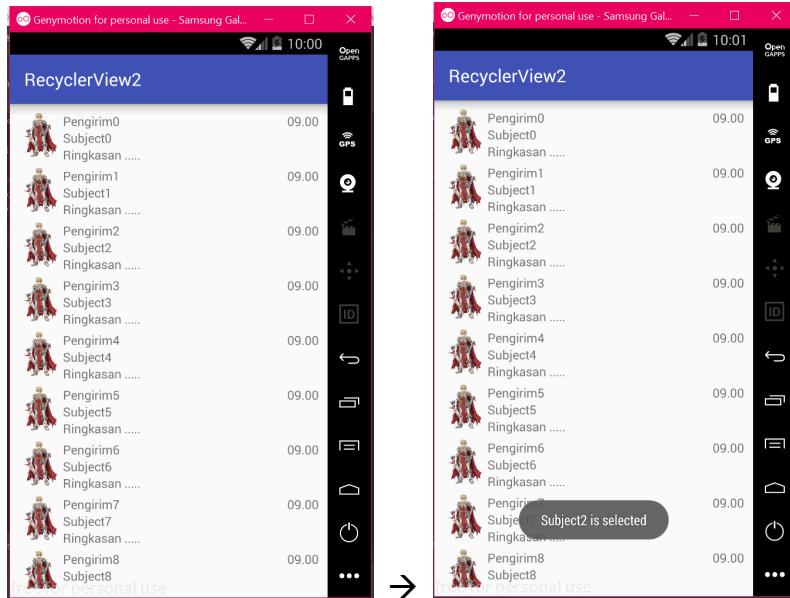
PRAKTIKUM PEMROGAMAN MOBILE



9. Pada *MainActivity.java* tambahkan script addOnItemTouchListener untuk memberikan event klik dan mengetahui list item mana yang dipilih / diklik.

```
98
99
100
101
102
103
104     mAdapter = new MyEmailAdapter(myEmails);
105     mRecyclerView.setAdapter(mAdapter);
106
107     mRecyclerView.addOnItemTouchListener(new RecyclerTouchListener(getApplicationContext(), mRecyclerView, new ClickListener() {
108         @Override
109         public void onClick(View view, int position) {
110             MyEmail email = myEmails.get(position);
111             Toast.makeText(getApplicationContext(), "text: " + email.getSubject() + " is selected", Toast.LENGTH_SHORT).show();
112         }
113     });
114
115
116
117 })
```

10. Lakukan tes program apakah sudah berjalan dengan baik, berikut adalah hasil jika kita memilih salah satu item untuk di klik.



D. Tugas Praktikum

- Buatlah 3 Activity dimana activity pertama mempunyai button untuk pindah ke activity kedua dan activity kedua mempunyai button untuk pindah ke activity ketiga.
- Buatlah 2 buah activity, pada activity pertama buatlah sebuah form yang isinya meminta biodata kita (nama, nim, alamat, tanggal lahir, jenis kelamin, agama, dll.) dan sebuah tombol simpan dimana jika tombol ini ditekan akan membuka activity ke dua yang akan menampilkan data yang kita masukkan. Gunakan widget yang sesuai seperti Edit Text, Radio Button, Spinner, dll.