

Tutorial 3

Android Intent

Intent merupakan objek perpesanan yang digunakan untuk meminta aksi dari komponen aplikasi lain seperti activities, services, broadcast receiver, dan konten providers.

Intent akan membantu kita menjaga komunikasi antara komponen aplikasi dari aplikasi yang sama serta dengan komponen aplikasi lain.

a. Implicit Intent

Di android, Intent Implicit tidak akan menentukan nama komponen apa pun untuk memulai, sebaliknya menyatakan tindakan yang dilakukan dan memungkinkan komponen dari aplikasi lain untuk menanganinya.

Misalnya, dengan menggunakan Intent Implisit, kita dapat meminta aplikasi lain untuk menampilkan detail lokasi pengguna, halaman url atau dll.

Contoh:

Buatlah sebuah Activity baru dengan nama **ActivitySatu.java** lalu isikan baris kodingan berikut:

```
package com.example.tutorial3;

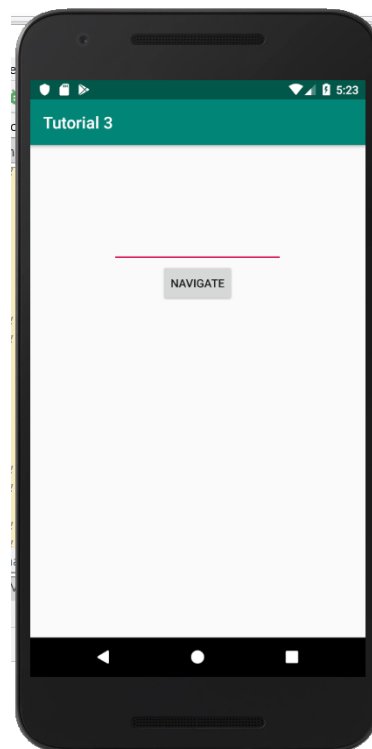
import ...

public class ActivitySatu extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_satu);
        final EditText editText = (EditText) findViewById(R.id.urlText);
        Button btn = (Button) findViewById(R.id.btnSatu);
        btn.setOnClickListener((v) → {
            String url = editText.getText().toString();
            Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));
            startActivity(intent);
        });
    }
}
```

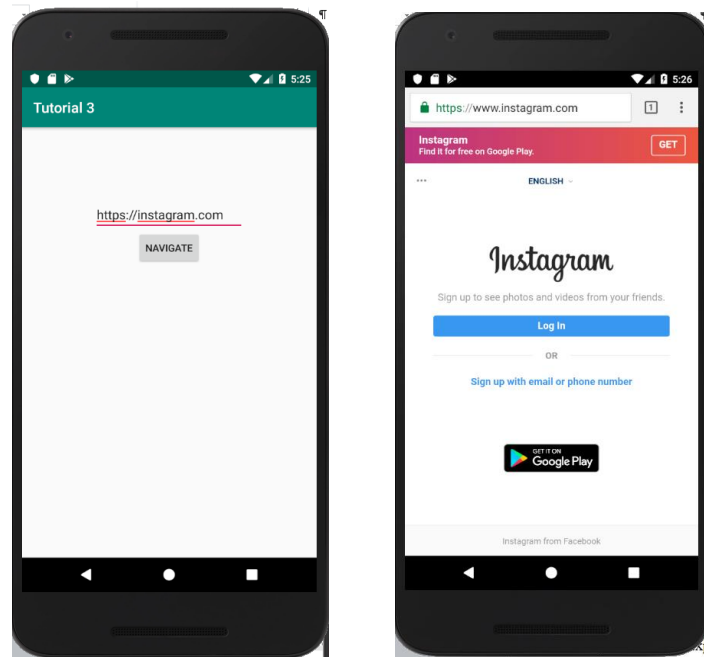
Selanjutnya buat sebuah layout baru dengan nama **activity_satu.xml** dengan komponen sebagai berikut:

```
<?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=".ActivitySatu">
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/urlText"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"
        android:ems="10" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btnSatu"
        android:layout_below="@+id/urlText"
        android:text="Navigate"
        android:layout_centerHorizontal="true" />
</RelativeLayout>
```

Setelah anda mendaftarkan ActivitySatu ke Manifest, jalankan program, sehingga akan muncul tampilan seperti berikut:



Masukkan sebuah alamat URL semisal <https://instagram.com> lalu klik tombol **navigate**, maka akan muncul tampilan berikut:



b. Explicit Intent

Intent Explicit secara eksplisit menentukan nama komponen yang akan dipanggil oleh suatu Activity dan kita menggunakan Intent Explicit untuk memulai komponen di aplikasi kita sendiri.

Misalnya, kita dapat memulai Activity baru sebagai respons terhadap aksi pengguna menggunakan Intent Explicit.

Contoh:

Lakukan modifikasi pada MainActivity sehingga kodingan didalam class ini akan menjadi seperti berikut:

```
public class MainActivity extends AppCompatActivity {
    private TextView header;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        header = (TextView) findViewById(R.id.txtHeader);
    }

    public void launchSecondActivity(View view) {
        // Membuat objek Intent
        Intent intent = new Intent( packageContext: this, SecondActivity.class);
        // Menjalankan Activity dengan parameter Intent
        startActivity(intent);
    }
}
```

Setelahnya pada layout **activity_main.xml** isikan komponen berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:paddingBottom="30dp"
    android:paddingLeft="30dp"
    android:paddingRight="30dp"
    android:paddingTop="30dp"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/txtHeader"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="30dp"
        android:text="Activity 1"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textStyle="bold" />
    <Button
        android:id="@+id/btnSatu"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="137dp"
        android:layout_marginRight="137dp"
        android:layout_marginBottom="364dp"
        android:onClick="launchSecondActivity"
        android:text="Ke Halaman 2"
        tools:ignore="OnClick" />
</RelativeLayout>
```

Langkah berikutnya kita akan membuat Activity baru dengan nama **SecondActivity.java** yang akan menjadi komponen Activity sebagai aksi/ respon ketika pengguna menekan tombol di MainActivity. Berikut baris kodingan untuk **SecondActivity.java**:

```
package com.example.tutorial3;

import ...

public class SecondActivity extends AppCompatActivity {
    private TextView header;
    Button btn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        header = (TextView) findViewById(R.id.txtHeader);
        btn = (Button) findViewById(R.id.btnDua);

        btn.setOnClickListener((v) -> { returnReply(v); });

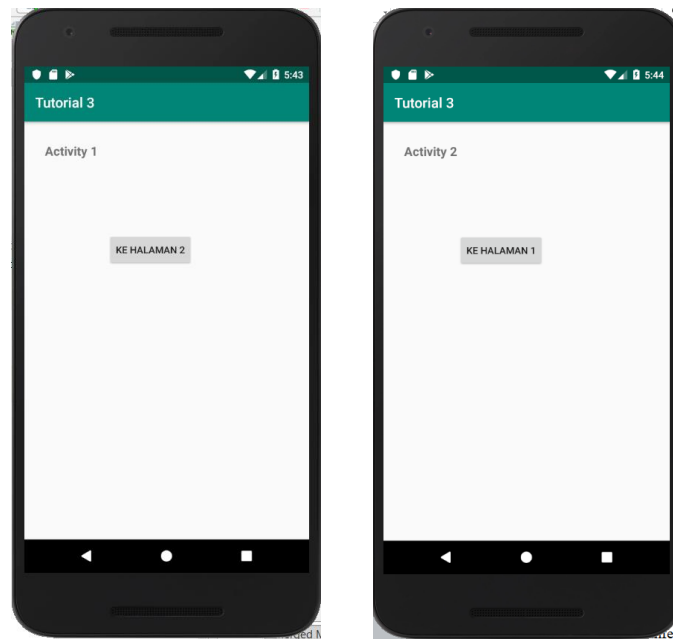
        header = (TextView) findViewById(R.id.txtHeader);
    }

    public void returnReply(View view) {
        // Membuat Intent
        Intent intent = new Intent( packageContext: this, MainActivity.class);
        // Menjalankan Activity dengan parameter Intent
        startActivity(intent);
    }
}
```

Untuk layout yang akan dipanggil oleh SecondActivity, kita buat sebuah layout dengan nama **activity_second.xml** dengan baris kodingan komponen sebagai berikut:

```
<?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:paddingBottom="30dp"
    android:paddingLeft="30dp"
    android:paddingRight="30dp"
    android:paddingTop="30dp"
    tools:context=".SecondActivity">
    <TextView
        android:id="@+id/txtHeader"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="30dp"
        android:text="Activity 2"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textStyle="bold" />
    <Button
        android:id="@+id/btnDua"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="150dp"
        android:layout_marginRight="300dp"
        android:layout_marginLeft="200dp"
        android:layout_marginBottom="364dp"
        android:onClick="returnReply"
        android:text="Ke Halaman 1"
        tools:ignore="OnClick" />
</RelativeLayout>
```

Jangan lupa untuk mendaftarkan SecondActivity di Manifest, lalu jalankan program. Coba lakukan aksi klik pada tombol **Ke Halaman 2**, maka halaman akan berpindah ke **activity_second.xml** yang dipanggil oleh **SecondActivity.java**.



c. Intent Filter

Di android, Intent Filter adalah ekspresi dalam file manifest aplikasi (ActivityManifest.xml) dan digunakan untuk menentukan jenis Intent yang ingin diterima oleh komponen.

Jika kita membuat Intent Filter untuk suatu Activity, ada kemungkinan bagi aplikasi lain untuk memulai Activity yang kita buat dengan mengirimkan jenis Intent tertentu, jika tidak, Activity tersebut dapat dimulai hanya dengan Intent Explicit.

Secara umum, Intent Filter (<intent-filter>) apa pun yang kita definisikan dalam file manifest dapat diletakkan di dalam komponen aplikasi yang sesuai dan kita dapat menentukan jenis Intent menggunakan tiga elemen ini.

- **<action>**

Mendefinisikan nama aksi Intent untuk diterima dan harus bernilai string dari suatu aksi, bukan konstanta kelas.

- **<kategori>**

Mendefinisikan nama kategori Intent untuk diterima dan harus berupa nilai string literal dari suatu aksi, bukan konstanta kelas.

- **<data>**

Mendefinisikan jenis data yang akan diterima dan dengan menggunakan satu atau lebih atribut kita dapat menentukan berbagai aspek data URI (skema, host, port, path) dan tipe MIME.

MIME / Multi-Purpose Internet Mail Extensions merupakan fitur untuk menangani beberapa tipe file di browser. Ketika file dikirim ke browser pengunjung sebuah website, MIME Type juga akan dikirim. MIME Type akan memberitahukan browser tipe file dan bagaimana file tersebut akan dikembalikan ke user. Contohnya, jika file tipe .js dikirim ke browser lalu MIME type juga akan mengirim ke browser untuk memberitahu bahwa file tipe .js adalah file JavaScript, dan browser akan menangani file yang sesuai.

Berikut contoh Intent Filter yang ada di AndroidManifest.xml

```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Contoh:

Buatlah sebuah layout dengan nama **activity_intentfilter.xml**, lalu isikan baris kodingan berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:id="@+id/sendMail"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="150dp"
        android:layout_marginTop="230dp"
        android:text="Send Mail" />
</LinearLayout>
```

Selanjutnya buat sebuah Activity baru dengan nama **ActivityIntentFilter.java** untuk memanggil layout yang tela, lalu isikan baris kodingan berikut:


```
package com.example.tutorial3;

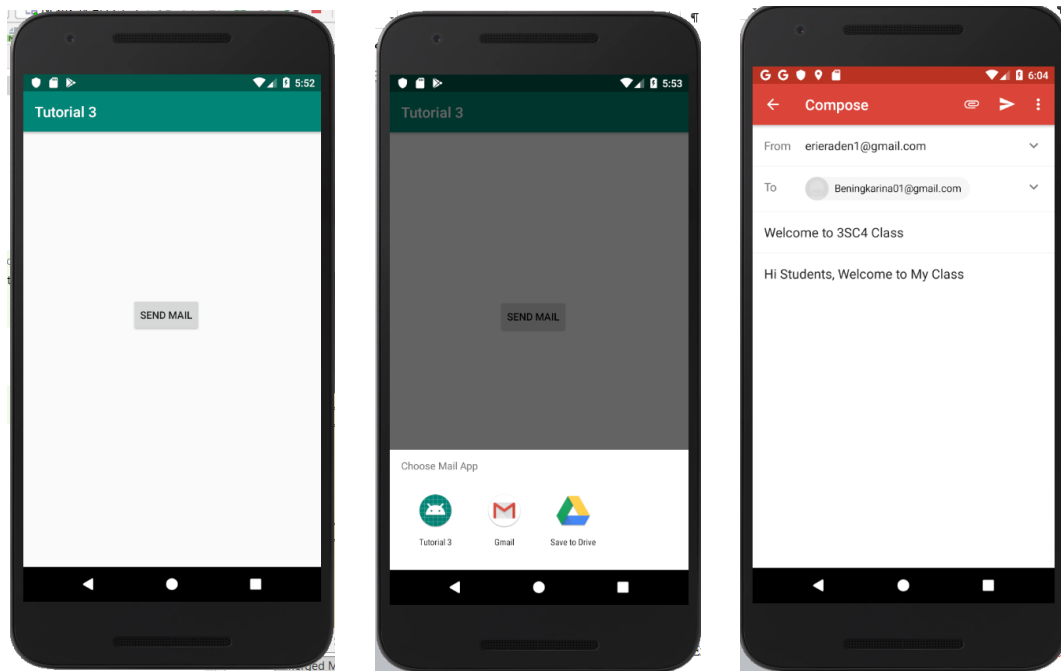
import ...

public class ActivityIntentFilter extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_intentfilter);
        Button btnSend = (Button) findViewById(R.id.sendMail);
        btnSend.setOnClickListener((v) -> {
            // Membuat objek Intent
            Intent si = new Intent(Intent.ACTION_SEND);
            // Mengatur MIME Type
            si.setType("message/rfc822");
            // Mengatur data nilai untuk Alamat E-Mail
            si.putExtra(Intent.EXTRA_EMAIL, new String[]{"dindamlv@gmail.com"});
            // Mengatur data nilai untuk Subject E-Mail
            si.putExtra(Intent.EXTRA_SUBJECT, value: "Welcome to 3SC5 Class");
            // Mengatur data nilai untuk Badan E-Mail
            si.putExtra(Intent.EXTRA_TEXT, value: "Hi Students, Welcome to My Class");
            /* Menjalankan Activity dengan parameter fungsi createChooser dengan parameter objek Intent */
            startActivity(Intent.createChooser(si, title: "Choose Mail App"));
        });
    }
}
```

Langkah terakhir, daftarkan ActivityIntentFilter ke manifest dengan detail seperti berikut:

```
<activity android:name=".ActivityIntentFilter">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
        <action android:name="android.intent.action.SEND"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:mimeType="message/rfc822"/>
    </intent-filter>
</activity>
```

Jalankan aplikasi!



d. Menggunakan Intent untuk Mengirim Data

Dengan menggunakan Intent Explicit, kita bisa mengirim atau berbagi data / konten dari satu Activity ke Activity lain berdasarkan persyaratan yang kita tentukan. Untuk membuat Intent Explicit, kita perlu mendefinisikan nama komponen untuk objek Intent.

Contoh:

Kita ingin mengirimkan dua buah angka dari Activity pertama ke Activity kedua, kemudian kedua angka tersebut akan dikalkulasikan lalu dikembalikan ke Activity pertama oleh Activity kedua.

Untuk melakukan hal tersebut, buatlah dua buah activity dengan nama **ActivityInput.java** dan **ActivityOutput.java**. Selain itu, buatlah dua buah layout dengan nama **activity_input.xml** dan **activity_output.xml**.

activity_input.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/txtSatu"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:layout_marginTop="150dp"
        android:text="Angka Pertama"
    />
    <EditText
        android:id="@+id/angkaSatu"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:ems="10">
    </EditText>
    <TextView
        android:id="@+id/txtDua"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Angka Kedua"
        android:layout_marginLeft="100dp"
    />
    <EditText
        android:id="@+id/angkaDua"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:ems="10" />
    <Button
        android:id="@+id/btnTambah"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:text="TAMBAH" />
</LinearLayout>
```

activity_output.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/vHasil"
        android:layout_width="161dp"
        android:layout_height="34dp"
        android:layout_marginLeft="135dp"
        android:layout_marginTop="300dp"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="30sp" />
</LinearLayout>
```

ActivityInput.java

```
package com.example.tutorial3;

import ...

public class ActivityInput extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_input);
        final EditText firstNum = (EditText)findViewById(R.id.angkaSatu);
        final EditText secNum = (EditText)findViewById(R.id.angkaDua);
        Button btnAdd = (Button)findViewById(R.id.btnTambah);
        btnAdd.setOnClickListener((v) -> {
            // Memberi nilai ke dua buah variabel yakni num1 dan num2
            // Nilai diambil dari nilai yang dimasukkan pengguna di TextField
            int num1 = Integer.parseInt(firstNum.getText().toString());
            int num2 = Integer.parseInt(secNum.getText().toString());

            // Membuat objek Intent
            Intent intent = new Intent( packageContext: ActivityInput.this, ActivityOutput.class);
            /* Mendeklarasikan atribut Intent yakni jumlah dengan nilai hasil penjumlahan num1 dan num2 */
            intent.putExtra( name: "jumlah", value: num1+ " + "+num2+" = "+(num1+num2));

            // Menjalankan activity dengan parameter Intent
            startActivity(intent);
        });
    }
}
```

ActivityOutput.java

```
package com.example.tutorial3;

import ...

public class ActivityOutput extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_output);
        TextView result = (TextView)findViewById(R.id.vHasil);

        // Mengambil nilai Intent dari activity sebelumnya, yakni Activity1
        Intent intent = getIntent();

        /* Mengambil nilai atribut yang diangkut Intent dari Activity1 lalu disimpan ke
        sebuah variabel bertipe string dengan nama addition */
        String addition = (String)intent.getSerializableExtra( name: "jumlah");

        // Mengatur nilai TextField result dengan nilai didalam variabel addition
        result.setText(addition);
    }
}
```

Daftarkan kedua Activity di manifest:

```
<activity android:name=".ActivityInput">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity android:name=".ActivityOutput"
    android:label="Explicit Intent - Activity2">
</activity>
```

Jalankan program!

