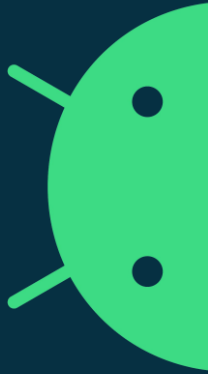


Lab.

Pemrograman Mobile

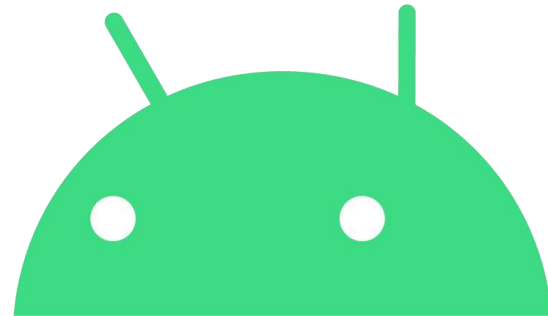


Pertemuan 3



Learning Objectives

- o Explicit Intent
 - o Sending Data Without Parcelable
- o Implicit Intent



android

Pert. 3

Intent



Secara singkat, Intent adalah komunikasi antar komponen aplikasi

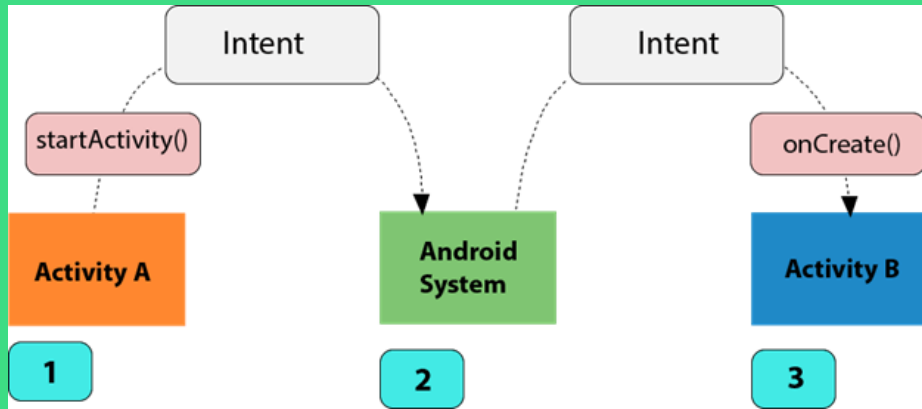


android

Intent

As the name says Intent is something that's used to perform some action with respect to the flow of the android application. Intents can be used to:

- Starting a new activity and passing some data.
- Starting Fragments/Communicating between fragments.
- Start/End service.
- Launch activities from a broadcast receiver



1. Explicit Intent

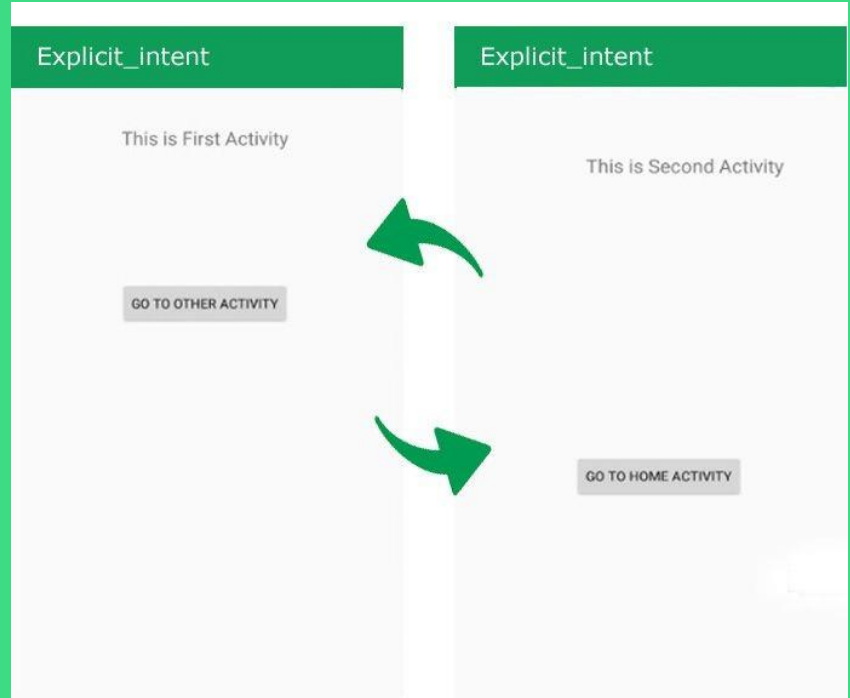
Intent yang digunakan untuk menjalankan komponen dari dalam sebuah aplikasi.

Explicit intent bekerja dengan menyebutkan nama kelas yang dituju

E.g.

`com.example.activity.ActivityExample.`

Biasanya digunakan untuk mengaktifkan komponen pada aplikasi itu sendiri.



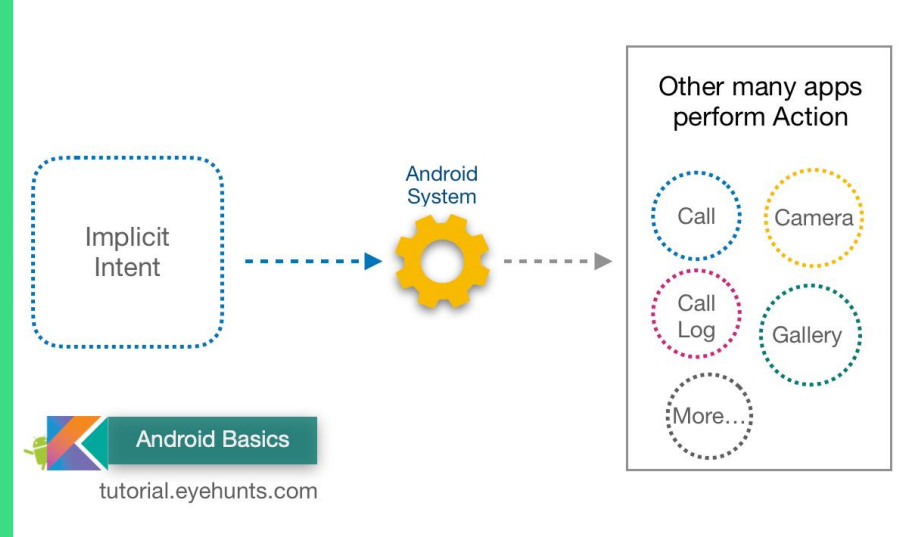
Explicit Intent

```
val moveIntent = Intent(this@MainActivity, AnotherActivity::class.java)  
startActivity(moveIntent)
```


2. Implicit Intent

Merupakan intent yang digunakan untuk berkomunikasi atau menjalankan komponen dari aplikasi lain.

Contohnya adalah membuka camera atau membuka browser dengan tidak menyebutkan nama class secara spesifik.



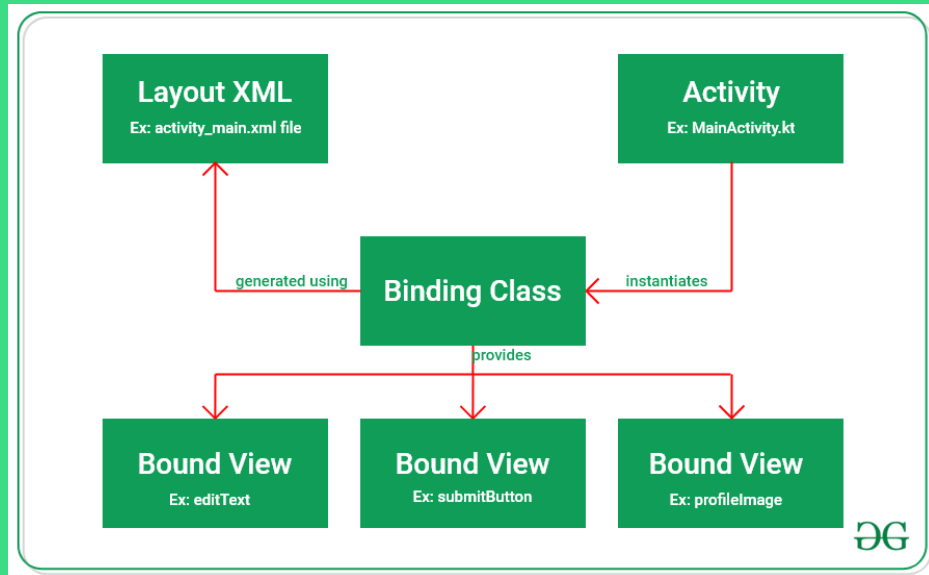
Implicit Intent

```
val intent = Intent(Intent.ACTION_VIEW, Uri.parse("https://developer.android.com"))  
startActivity(intent)
```

View Binding

View binding is a feature that makes it easier to write code that interacts with views. Once view binding is enabled in a module, it generates a binding class for each XML layout file present in that module. An instance of a binding class contains direct references to all views that have an ID in the corresponding layout.

In most cases, view binding replaces `findViewById`.



Setup View Binding

First enable view binding in build.gradle (module) file by writing these line of codes under android block.

```
buildFeatures {  
    viewBinding = true  
}
```

Setup View Binding

Then in activity file to use view binding features

Declare a global variable for binding as

```
private lateinit var binding:Activity[nama_activity]Binding
```

eg. ActivityMain:

```
private lateinit var binding:ActivityMainBinding
```

Setup View Binding

Then in activity file to use view binding features

Declare a global variable for binding as

```
private lateinit var binding:Activity[nama_activity]Binding
```

eg. ActivityMain:

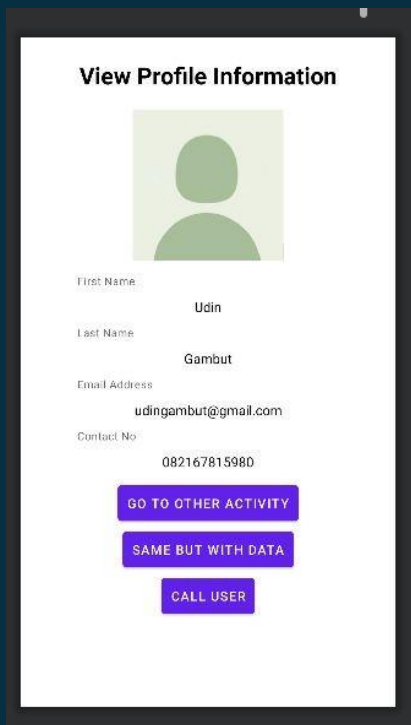
```
private lateinit var binding:ActivityMainBinding
```

Here ActivityMainBinding is an auto generated class

Setup View Binding

then write these codes in onCreate() method

```
super.onCreate(savedInstanceState)
binding = ActivityMainBinding.inflate(layoutInflater)
val view = binding.root
setContentView(view)
```



```
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

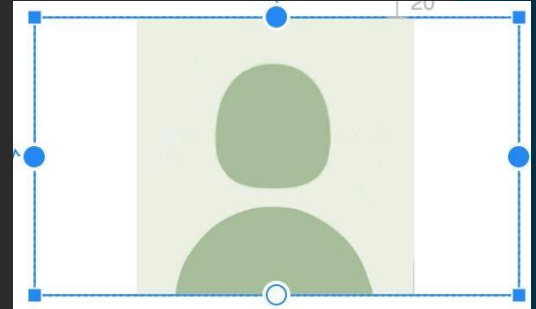
```
<TextView
    android:id="@+id/tv_heading"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/profile_info_data"
    android:textStyle="bold"
    android:textSize="26sp"
    android:textColor="#000"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginTop="24dp"
/>
```

Try To Imitate This


```
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@id/tv_heading"
>
```

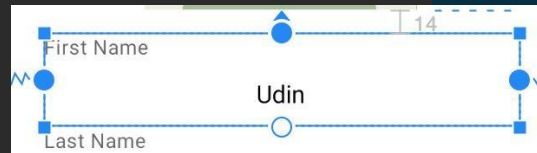
```
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/layout_image"
    android:layout_width="0dp"
    app:layout_constraintWidth_percent="0.7"
    android:layout_height="wrap_content"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="20dp"
    android:textColorHint="#808080"
    app:layout_constraintTop_toTopOf="parent"
>

<ImageView
    android:id="@+id/image_profile"
    android:layout_width="wrap_content"
    android:layout_height="165dp"
    android:src="@drawable/clipart1717870"
    android:contentDescription="@string/image_profile" />
</com.google.android.material.textfield.TextInputLayout>
```

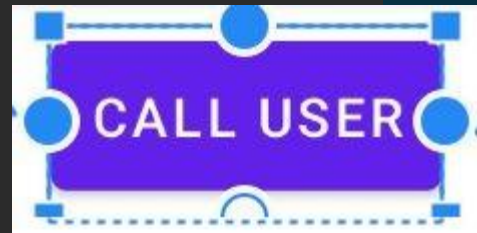


```
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/layout_first_name"
    app:helperText="First Name"
    android:layout_width="0dp"
    app:layout_constraintWidth_percent="0.7"
    android:layout_height="wrap_content"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="14dp"
    android:textColorHint="#808080"
    app:layout_constraintTop_toBottomOf="@+id/layout_image"
>
```

```
<TextView
    android:id="@+id/et_first_name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/handy"
    android:gravity="center"
    android:padding="10dp"
    android:textColor="#000000"
    android:textColorHint="#808080"
    android:background="#00000000"
    android:textSize="15sp"
/>
</com.google.android.material.textfield.TextInputLayout>
```



```
<Button
    android:id="@+id/call_user"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_3"
    android:gravity="center"
    android:padding="10dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button_2"
/>
```



Declaring Buttons for Intent Purpose

```
// Explicit intent to move to another activity without data  
binding.moveActivity.setOnClickListener {  
    val moveIntent = Intent(this@MainActivity, AnotherActivity::class.java)  
    startActivity(moveIntent)  
}
```

Declaring Buttons for Intent Purpose

```
// Explicit intent to move to another activity with data
binding.moveActivityData.setOnClickListener {
    val firstName = binding.etFirstName.text.toString()
    val lastName = binding.etLastName.text.toString()

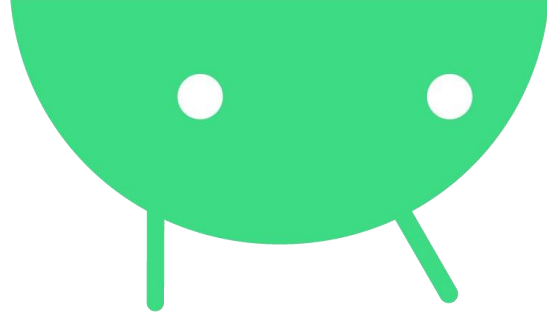
    val moveWithDataIntent = Intent(this@MainActivity, AnotherActivity::class.java)

    moveWithDataIntent.putExtra("extra_fname", firstName)
    moveWithDataIntent.putExtra("extra_lname", lastName)

    startActivity(moveWithDataIntent)
}
```

Declaring Buttons for Intent Purpose

```
// Implicit content to call a phone number  
binding.callUser.setOnClickListener {  
    val contactData = binding.etContactNumber.text  
  
    val dialPhoneIntent = Intent(Intent.ACTION_DIAL, Uri.parse("tel:$contactData"))  
    startActivity(dialPhoneIntent)  
}
```



Any Question?