

## Tujuan Praktikum

1. Mahasiswa mampu memahami Intent.
2. Mahasiswa mampu membuat aplikasi sederhana dengan menerapkan Intent.

## Tools

1. Android Studio

Panduan ini menggunakan versi Android Studio: Giraffe - 2022.3.1

Android API: Level 34

# Intent

Intent merupakan suatu objek perpesanan yang dapat digunakan untuk meminta sebuah aksi dari komponen aplikasi yang lain. Terdapat beberapa cara untuk berkomunikasi antar komponen yang dapat dilakukan oleh Intent. Sebagai contoh, berikut adalah beberapa kasus penggunaan Intent yang sering terjadi:

## 1. Memulai sebuah activity

Activity merupakan representasi dari sebuah halaman pada aplikasi. Kita dapat memanggil sebuah Activity dengan mengirimkan object Intent sebagai parameter pada method `startActivity()`. Intent digunakan untuk mendeskripsikan activity yang akan dituju dengan atau tidak membawa data yang diperlukan.

Contoh implementasi Intent untuk berpindah dari MainActivity ke DetailActivity:

```
val detailIntent = Intent(MainActivity.this, DetailActivity::class.java);  
startActivity(detailIntent);
```

## 2. Memulai sebuah service

Service merupakan sebuah komponen yang melakukan sebuah proses di latar belakang tanpa membutuhkan sebuah user interface. Kita dapat memulai sebuah service (contohnya mengunduh file dari server) dengan mengirimkan object Intent sebagai parameter pada method `startService()`.

## 3. Mengirimkan sebuah broadcast

Broadcast merupakan sebuah pesan yang dapat diterima oleh aplikasi lain. Contoh pengiriman broadcast adalah ketika kita mengisi daya perangkat, sistem Android akan mengirimkan pesan broadcast bahwa perangkat sedang diisi dayanya. Untuk mengirimkan broadcast melalui aplikasi, kita dapat mengirimkan object Intent sebagai parameter pada method `sendBroadcast()` atau `sendOrderedBroadcast()`.

## Jenis Intent

Terdapat dua jenis Intent:

1. **Explicit Intent**, merupakan sebuah Intent yang memiliki tujuan yang jelas. Hal ini ditandai dengan didefinisikannya tujuan dari Intent tersebut seperti nama package suatu aplikasi atau nama suatu komponen. Explicit Intent inilah yang biasanya digunakan untuk memulai suatu komponen (seperti berpindah activity) dari aplikasi yang kita kembangkan.

Contoh penggunaan Explicit Intent:

```
val downloadIntent = Intent(this, DownloadService::class.java).apply {  
    data = Uri.parse(fileUrl)  
}  
startService(downloadIntent)
```

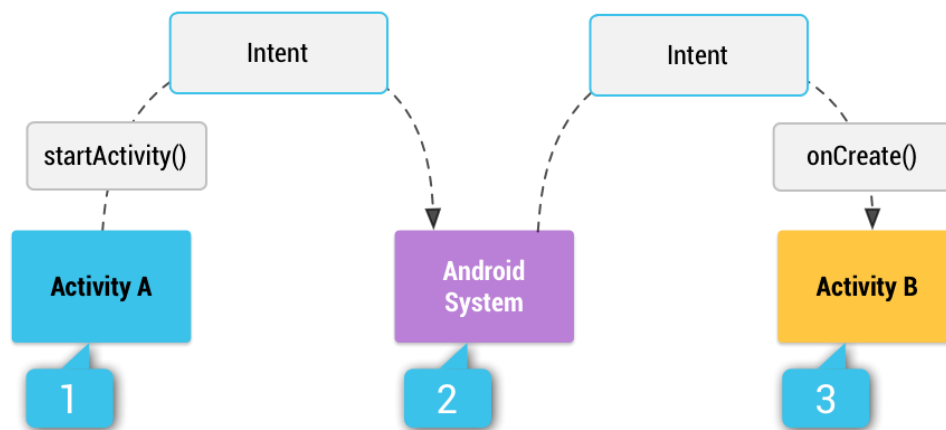
2. **Implicit Intent**, merupakan sebuah Intent yang tidak menyebutkan komponen yang ingin dituju. Implicit Intent hanya mendeklarasikan sebuah tugas general yang dapat dilakukan oleh komponen pada aplikasi lain. Contohnya, Ketika kita ingin mengirimkan email, kita dapat menggunakan Implicit Intent untuk meminta aplikasi lain yang memiliki fungsi untuk mengirimkan email agar dapat melaksanakan tugas tersebut.

Contoh penggunaan Implicit Intent:

```
val sendIntent = Intent().apply {  
    action = Intent.ACTION_SEND  
    putExtra(Intent.EXTRA_TEXT, textMessage)  
    type = "text/plain"  
}  
try {  
    startActivity(sendIntent)  
} catch (e: ActivityNotFoundException) {  
    //Dieksekusi ketika tidak ada activity yang dapat dipilih.  
}
```

## Cara Kerja Intent

Ketika Intent yang digunakan sudah terdapat sebuah komponen yang ingin dipanggil (Explicit Intent), maka sistem akan langsung memulai komponen tersebut. Sementara ketika kita menggunakan Implicit Intent, sistem Android akan mencari terlebih dahulu komponen-komponen yang cocok untuk melaksanakan suatu aksi yang diminta. Sistem akan membandingkan intent-filters yang dideklarasikan pada AndroidManifest yang ada di aplikasi yang terpasang pada perangkat. Jika terdapat sebuah intent-filters yang cocok, maka komponen tersebut akan dipanggil. Jika terdapat lebih dari satu intent-filters yang cocok, maka akan ditampilkan sebuah dialog sehingga pengguna dapat memilih aplikasi mana yang akan digunakan.



**[1]** Activity A membuat sebuah Intent yang berisikan action dan mengirimkannya ke method `startActivity()`.

**[2]** Sistem Android mencari seluruh intent-filters pada aplikasi yang terinstall di perangkat yang cocok dengan intent yang dikirimkan.

**[3]** Jika ditemukan, sistem akan memulai activity yang cocok (Activity B) dengan mengeksekusi method `onCreate()` serta mengirimkan object Intent-nya.

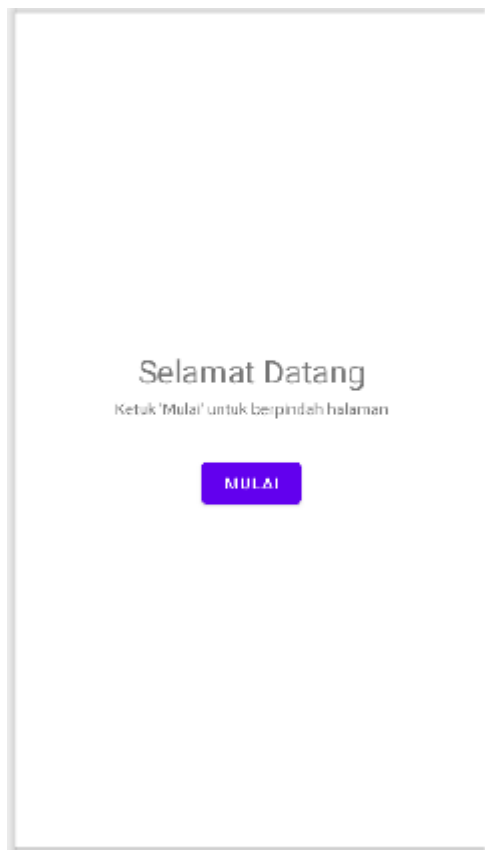
## Latihan – Explicit Intent

### 1. Persiapan Project

Buatlah sebuah project baru dengan konfigurasi berikut:

Nama Project	Intent Application
Bahasa Pemrograman	Kotlin DSL
Minimum SDK	API 24: Android 7.0 (Nougat)
Tipe Activity	Empty Views Activity
Nama Package	com.android.intentapp

### 2. Layouting



Kondisikan halaman MainActivity seperti pada gambar disamping. Silakan buat atau buka file `activity_main.xml` pada folder `res/layout/` dan tuliskan kode berikut:

**Penting: Hindari copy-paste, tulislah kembali kode sambil mengamati perubahan yang terjadi pada preview.**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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:gravity="center"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Selamat Datang"
        android:textSize="28sp" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="4dp"
        android:text="Ketuk 'Mulai' untuk berpindah halaman" />
    <Button
        android:id="@+id/btn_start"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:text="Mulai" />
</LinearLayout>
```

### 3. Daftarkan View ke MainActivity.kt

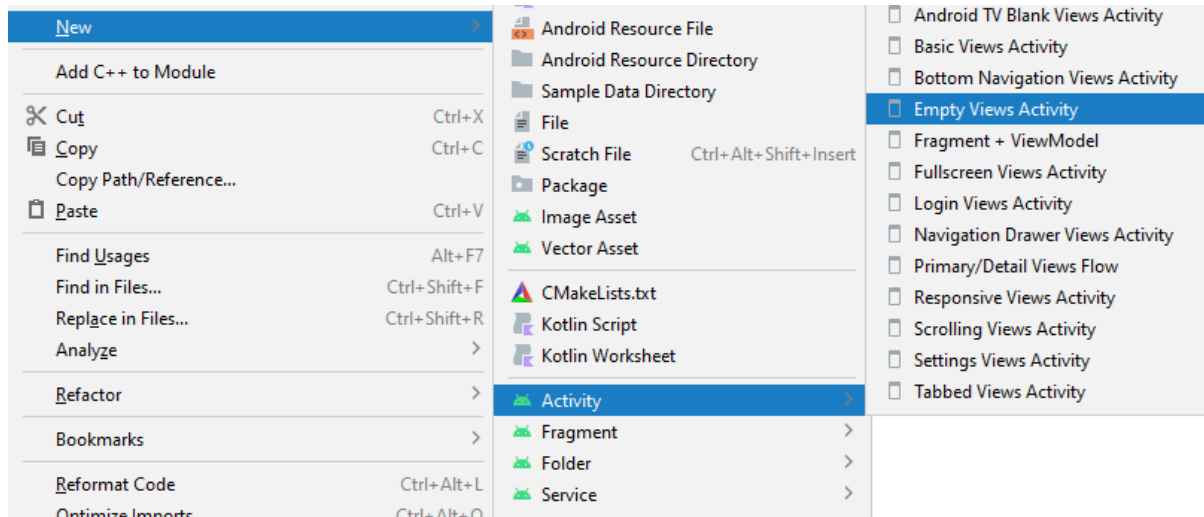
Pada halaman MainActivity.kt, deklarasikan sebuah variabel btnStart dan tambahkan sebuah event handler onClick pada view tersebut.

```
class MainActivity : AppCompatActivity(), View.OnClickListener {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView {
            MyApplicationTheme {
                setContentView(R.layout.activity_main)
                val btnStart: Button = findViewById(R.id.btn_start)
                btnStart.setOnClickListener(this)
            }
        }
    }

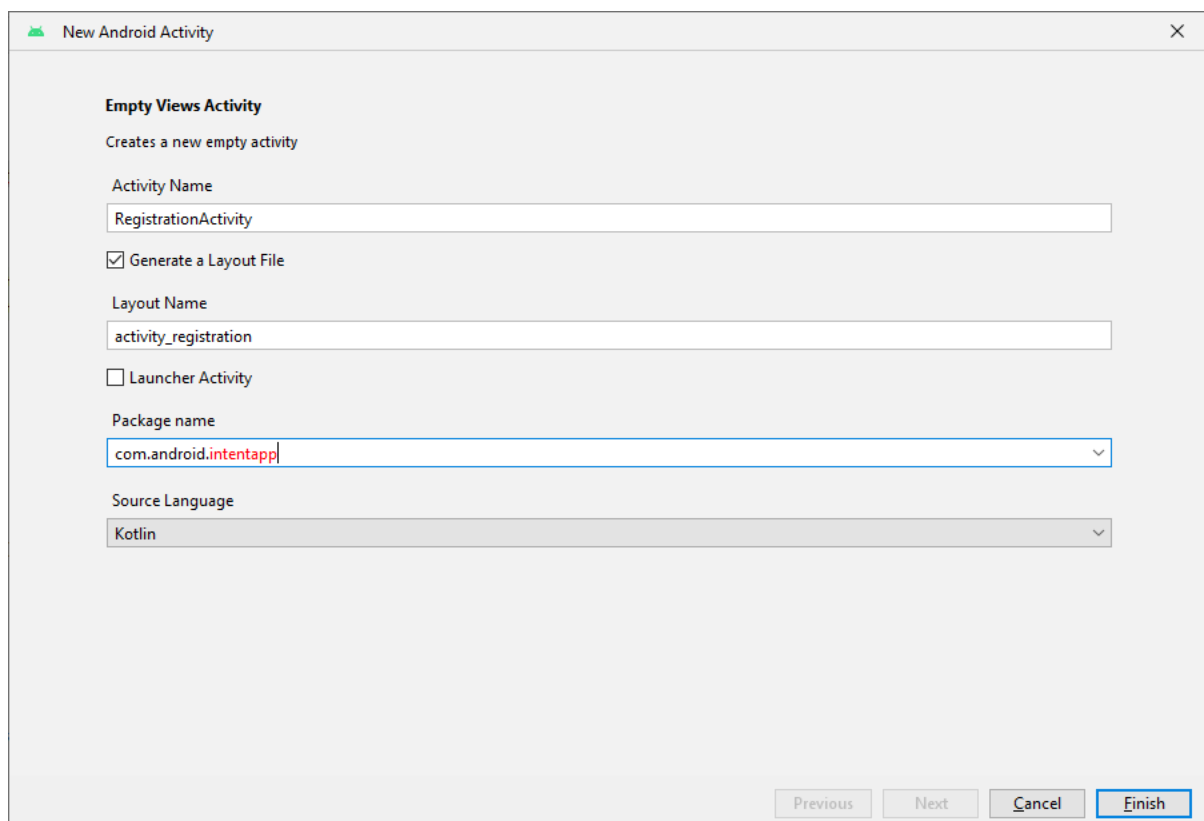
    override fun onClick(view: View?) {
        if (view != null) {
            if (view.id == R.id.btn_start) {
            }
        }
    }
}
```

#### 4. Membuat Activity Baru

Button `btnStart` akan digunakan untuk berpindah Activity pada aplikasi. Sebelum mengimplementasikan Intent untuk memanggil komponen Activity yang dituju, kita buat terlebih dahulu sebuah Activity baru. Untuk membuat Activity baru yang kosong, kita dapat menuju menu `File > New > Activity > Empty Views Activity`.



Setelah itu akan tampil sebuah dialog. Isikan `RegistrationActivity` sebagai Activity Name pada dialog tersebut, lalu tekan tombol `Finish`.



Activity baru bernama RegistrationActivity sudah berhasil kita buat. Ketika membuat sebuah Activity, kita juga harus mendaftarkannya pada AndroidManifest, tetapi jangan khawatir karena proses ini sudah dilakukan secara otomatis oleh Android Studio.

Silakan tambahkan kode berikut pada file activity\_registration.xml untuk menginformasikan nama activity ketika nanti ditampilkan.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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:gravity="center"
    tools:context=".RegistrationActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Registration Activity" />
</LinearLayout>
```

## 5. Berpindah Activity dengan Intent

Buka kembali MainActivity.kt lalu buatlah object Intent pada callback onClick dengan menulis kode berikut:

```
class MainActivity : AppCompatActivity(), View.OnClickListener {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView {
            MyApplicationTheme {
                setContentView(R.layout.activity_main)
                val btnStart: Button = findViewById(R.id.btn_start)
                btnStart.setOnClickListener(this)
            }
        }
    }

    override fun onClick(view: View?) {
        if (view != null) {
            if (view.id == R.id.btn_start) {
                val registrationIntent = Intent(
                    this,
                    RegistrationActivity::class.java
                )
                startActivity(registrationIntent)
            }
        }
    }
}
```



Pastikan untuk selalu mengimpor kelas-kelas yang dibutuhkan pada MainActivity.kt

```
package android.intentapp

import android.content.Intent
import android.os.Bundle
import android.view.View
import android.widget.Button
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import com.android.intentapp.RegistrationActivity
import com.android.intentapp.ui.theme.MyApplicationTheme

class MainActivity : ComponentActivity(), View.OnClickListener {
    override fun onCreate(savedInstanceState: Bundle?) {
        ...
    }

    override fun onClick(view: View?) {
        ...
    }
}
```

Cobalah untuk melakukan build aplikasi dan jalankan pada emulator atau device Android. Ketuk tombol Mulai, lalu akan terbuka sebuah Activity baru yaitu RegistrationActivity. Selamat, kita sudah berhasil melakukan implementasi Intent untuk berpindah activity.

