**INTENT**

**Teori**

Intent adalah mekanisme untuk melakukan sebuah *action* dan komunikasi antar komponen aplikasi misal activity, services, dan broadcast receiver. Ada tiga penggunaan umum intentdalam aplikasi Android yaitu:

* Memindahkan satu activityke activitylain dengan atau tidak membawa data.
* Menjalankan background service,misalnya melakukan sinkronisasi ke server dan menjalankan proses berulang (*periodic*/*scheduler* *task*).
* Mengirimkan obyek broadcast ke aplikasi yang membutuhkan. Misal, ketika aplikasi membutuhkan proses menjalankan sebuah background service setiap kali aplikasi selesai melakukan *booting*. Aplikasi harus bisa menerima obyek *broadcast* yang dikirimkan oleh sistem Android untuk *event* *booting* tersebut.

Intent memiliki dua bentuk yaitu:

1. **Explicit Intent**   
   Adalah tipe Intentyang digunakan untuk menjalankan komponen dari dalam sebuah aplikasi. Explicitintentbekerja dengan menggunakan nama kelas yang dituju misal : com.dicoding.activity.DetailActivity. Umumnya intent ini digunakan untuk mengaktifkan komponen pada satu aplikasi.
2. **Implicit Intent**   
   Adalah tipe intentyang tidak memerlukan detail nama kelas yang ingin diaktifkan. Model ini memungkinkan komponen dari aplikasi lain bisa merespon *request* intent yang dijalankan.  
   Penggunaan tipe intentini umumnya diperuntukkan untuk menjalankan fitur/fungsi dari komponen aplikasi lain. Contohnya ketika kita membutuhkan fitur untuk mengambil foto. Daripada membuat sendiri fungsi kamera, lebih baik kita menyerahkan proses tersebut pada aplikasi kamera bawaan dari peranti atau aplikasi kamera lain yang telah terinstal sebelumnya di peranti.  
   Hal yang sama misalnya ketika kita membutuhkan fungsi berbagi konten. Kita bisa memanfaatkan intent untuk menampilkan aplikasi mana saja yang bisa menangani fitur tersebut.  
   Implementasi implicitintentini akan sangat memudahkan bagi pengembang agar tetap fokus pada proses bisnis inti dari aplikasi yang dikembangkan.

**Latihan Praktikum Intent**

Praktikum ini menitikberatkan pada implementasi intentuntuk melakukan perpindahan dari activityke activity lain, dengan atau tidak membawa data. Beberapa bagian dari praktikum ini akan menjawab beberapa pertanyaan umum dalam pengembangan aplikasi Android sebagai berikut:

* Bagaimana berpindah dari satu activityke activitylain?
* Bagaimana berpindah dari satu activityke activitylain dengan membawa data?
  + Single value dari suatu variabel.
  + Obyek model Plain Old Java Object (POJO).
* Menjalankan komponen di aplikasi lain untuk keperluan membuka *browser* atau melakukan pemanggilan melalui aplikasi telepon bawaan?
* Mengirimkan hasil nilai balik melalui Intent.

Praktikum selanjutnya adalah dengan membuat aplikasi yang dapat menerapakan kegunaan intent. Seperti ini tampilannya.



**Logika Dasar**

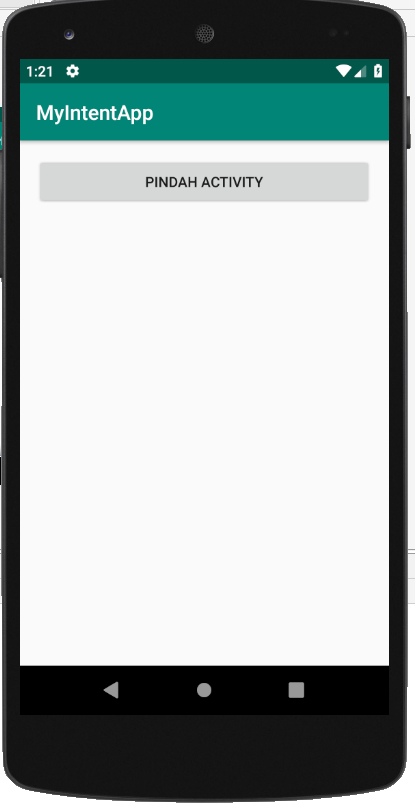
Berpindah dari satu **Activity**ke **Activity**lain dengan membawa data. **Activity**asal akan mengirimkan data melalui Intent dan **Activity**tujuan akan menerima data yang dikirimkan.

**Praktikum Intent Sederhana**

1. Buat Project baru di Android Studio dengan kriteria sebagai berikut:

|  |  |
| --- | --- |
| Nama Project | **MyIntentApp** |
| Target & Minimum Target SDK | **Phone and Tablet, Api level 21** |
| Tipe Activity | **Empty Activity** |
| Activity Name | **MainActivity** |

**Catatan:** Untuk versi **3.4.1** ada kecendurangan penggunaan **AndroidX** menjadi **default**. Informasi lebih detail mengenai **AndroidX** bisa Anda lihat di [sini](https://developer.android.com/jetpack/androidx), dan untuk melihat perubahan antara sebelum dan sesudah **AndroidX**, atau cara migrasi menggunakan AndroidX bisa lihat di [sini](https://developer.android.com/jetpack/androidx/migrate). Perlu Anda ketahui juga, ketika Anda sudah menggunakan **AndroidX** maka Anda tidak bisa menambahkan library lama atau sebelum **AndroidX**. Di dalam kelas ini masih menggunakan versi lama, maka Anda perlu menyesuaikan jika menggunakan **AndroidX**.

1. Selanjutnya kita akan membangun antarmuka (*interface*) seperti ini:  
   
2. Kondisikan activity\_main.xmlmenjadi seperti ini :

|  |
| --- |
| 1. <?xml version="1.0" encoding="utf-8"?> 2. <**LinearLayout** 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. **android:paddingBottom="@dimen/activity\_vertical\_margin"** 8. **android:paddingLeft="@dimen/activity\_horizontal\_margin"** 9. **android:paddingRight="@dimen/activity\_horizontal\_margin"** 10. **android:paddingTop="@dimen/activity\_vertical\_margin"**> 12. **<Button** 13. **android:id="@+id/btn\_move\_activity"** 14. **android:layout\_width="match\_parent"** 15. **android:layout\_height="wrap\_content"** 16. **android:layout\_marginBottom="@dimen/activity\_vertical\_margin"** 17. **android:text="@string/move\_activity" />** 18. </**LinearLayout**> |

Jangan lupa untuk menambahkan berkas dimens.xml secara manual di dalam **res → values**. Isikan berkas dimens.xml seperti berikut.

|  |
| --- |
| 1. <resources> 2. <!-- Default screen margins, per the Android Design guidelines. --> 3. **<dimen name="activity\_horizontal\_margin">16dp</dimen>** 4. **<dimen name="activity\_vertical\_margin">16dp</dimen>** 5. </resources> |

Tambahkan juga resource string-nya. Tambahkan semua string yang akan digunakan di project ini. Buka berkas strings.xml dan tambahkan kode berikut ini.

|  |
| --- |
| 1. <resources> 2. <string name="app\_name">MyIntentApp</string> 3. **<string name="move\_activity">Pindah Activity</string>** 4. **<string name="move\_with\_data">Pindah Activity dengan Data</string>** 5. **<string name="move\_with\_object">Pindah Activity dengan Object</string>** 6. **<string name="dial\_number">Dial a Number</string>** 7. **<string name="move\_with\_result">Pindah Activity untuk Result</string>** 8. **<string name="result\_from\_activity">Hasil dari Activity</string>** 9. **<string name="choose\_number">Pilih angka yang kamu suka</string>** 10. **<string name="\_50">50</string>** 11. **<string name="\_100">100</string>** 12. **<string name="\_150">150</string>** 13. **<string name="\_200">200</string>** 14. **<string name="choose">Pilih</string>** 15. **<string name="data\_received">Data Diterima</string>** 16. **<string name="object\_received">Object Diterima</string>** 17. **<string name="this\_is\_moveactivity">Ini MoveActivity</string>** 18. </resources> |

1. Kemudian untuk MainActivity.java tambahkan beberapa baris yang berfungsi untuk meperkenalkan button yang sudah ditambahkan di layout activity\_main.xml seperti ini:

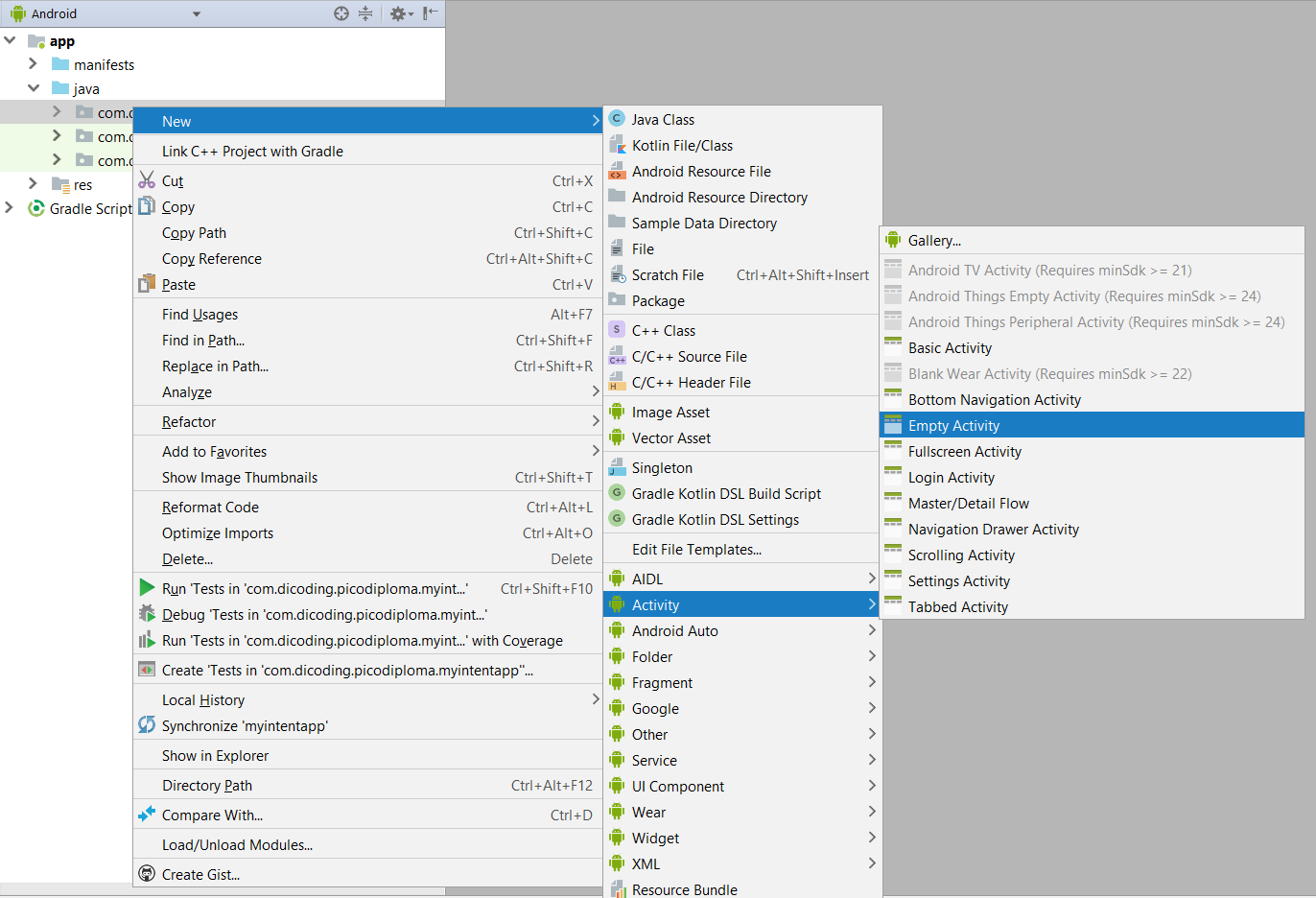
|  |
| --- |
| Java |
| 1. @Override 2. protected void onCreate(Bundle savedInstanceState) { 3. super.onCreate(savedInstanceState); 4. setContentView(R.layout.activity\_main); 6. **Button btnMoveActivity = findViewById(R.id.btn\_move\_activity);** 7. } |

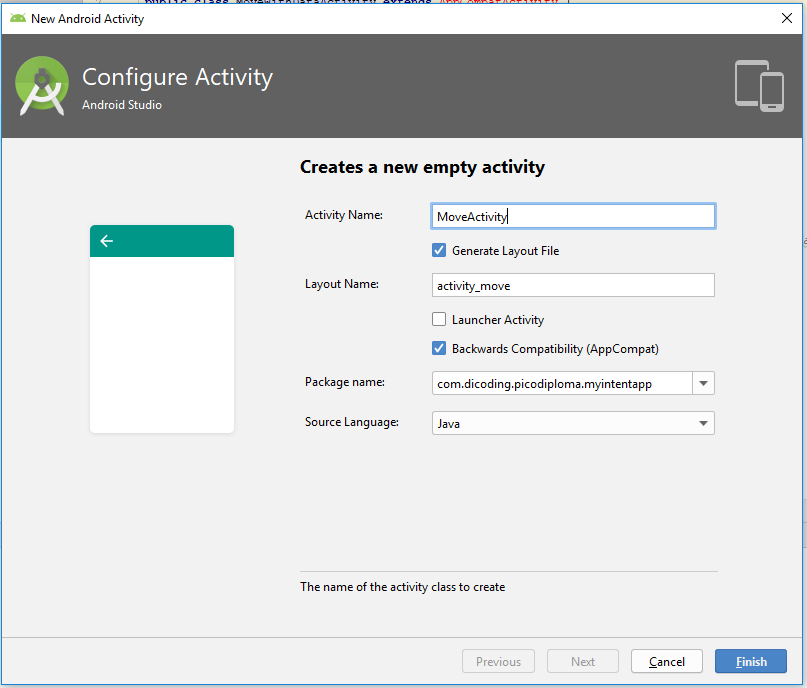
|  |
| --- |
| Kotlin |
| 1. override fun onCreate(savedInstanceState: Bundle?) { 2. super.onCreate(savedInstanceState) 3. setContentView(R.layout.activity\_main) 5. **val btnMoveActivity: Button = findViewById(R.id.btn\_move\_activity)** 6. } |

1. Lalu tambahkan beberapa baris yang berfungsi untuk menambahkan event onClick pada button btnMoveActivity seperti ini.

|  |
| --- |
| Java |
| 1. public class MainActivity extends AppCompatActivity **implements View.OnClickListener** { 3. @Override 4. protected void onCreate(Bundle savedInstanceState) { 5. super.onCreate(savedInstanceState); 6. setContentView(R.layout.activity\_main); 8. **Button btnMoveActivity = findViewById(R.id.btn\_move\_activity);** 9. **btnMoveActivity.setOnClickListener(this);** 10. } 12. **@Override** 13. **public void onClick(View v) {** 14. **switch (v.getId()) {** 15. **case R.id.btn\_move\_activity:** 16. **break;** 17. **}** 18. **}** 19. } |

|  |
| --- |
| Kotlin |
| 1. class MainActivity : AppCompatActivity(), View.OnClickListener { 3. override fun onCreate(savedInstanceState: Bundle?) { 4. super.onCreate(savedInstanceState) 5. setContentView(R.layout.activity\_main) 7. **val btnMoveActivity: Button = findViewById(R.id.btn\_move\_activity)** 8. **btnMoveActivity.setOnClickListener(this)** 9. } 11. **override fun onClick(v: View) {** 12. **when (v.id) {** 13. **R.id.btn\_move\_activity -> {** 14. **}** 15. **}** 16. **}** 17. } |

1. Button btnMoveActivity akan memiliki fungsi untuk berpindah Activityke Activitylain. Sekarang kita buat Activitybaru dengan cara sebagai berikut: **Klik kanan** di package utama aplikasi **package name → New → Activity → Empty Activity**.  
   

Lalu isikan MoveActivitypada dialog. Ketika sudah klik **Finish**.  


1. Untuk menandakan bahwa perpindahan activityberhasil, silakan tambahkan satu TextView dan kondisikan activity\_move.xmlmenjadi seperti berikut.

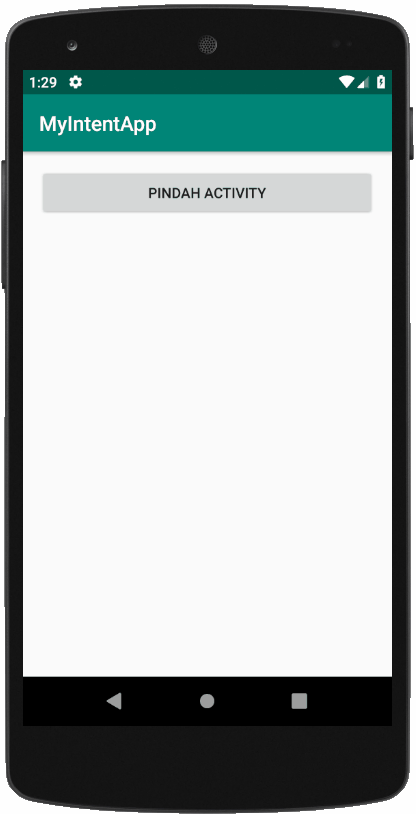
|  |
| --- |
| 1. <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android" 2. xmlns:tools="http://schemas.android.com/tools" 3. android:layout\_width="match\_parent" 4. android:layout\_height="match\_parent" 5. android:paddingBottom="@dimen/activity\_vertical\_margin" 6. android:paddingLeft="@dimen/activity\_horizontal\_margin" 7. android:paddingRight="@dimen/activity\_horizontal\_margin" 8. android:paddingTop="@dimen/activity\_vertical\_margin"> 10. **<TextView** 11. **android:layout\_width="match\_parent"** 12. **android:layout\_height="wrap\_content"** 13. **android:text="@string/this\_is\_moveactivity" />** 14. </RelativeLayout> |

1. Setelah Activitytujuan sudah berhasil diciptakan, sekarang saatnya menambahkan suatu Intentpada method onClick() di MainActivity.java menjadi sebagai berikut.

|  |
| --- |
| Java |
| 1. @Override 2. public void onClick(View v) { 3. switch (v.getId()){ 4. case R.id.*btn\_move\_activity*: 5. **Intent moveIntent = new Intent(MainActivity.this, MoveActivity.class);** 6. **startActivity(moveIntent);** 7. break; 8. } 9. } |

|  |
| --- |
| Kotlin |
| 1. override fun onClick(v: View) { 2. when (v.id) { 3. R.id.btn\_move\_activity -> { 4. **val moveIntent = Intent(this@MainActivity, MoveActivity::class.java)** 5. **startActivity(moveIntent)** 6. } 7. } 8. } |

1. Selesai! Langkah pertama untuk memindahkan satu Activityke Activitylain sudah selesai, sekarang silakan jalankan aplikasi Anda dengan mengklik tombol pada menu bar. Seharusnya sekarang anda sudah bisa memindahkan Activitydengan mengklik tombol ‘**Pindah Activity**’.



### ****Bedah Kode****

#### **Register Activity**

Kita telah belajar bagaimana membuat suatu activitybaru. Di materi sebelumnya, syarat suatu activityharuslah terdaftar pada berkas AndroidManifest.xml. Karena menggunakan Android Studio, proses pendaftaran tersebut dilakukan secara otomatis.

|  |
| --- |
| 1. <?xml version="1.0" encoding="utf-8"?> 2. <manifest xmlns:android="http://schemas.android.com/apk/res/android" 3. package="com.dicoding.picodiploma.myintentapp"> 5. <application 6. android:allowBackup="true" 7. android:icon="@mipmap/ic\_launcher" 8. android:label="@string/app\_name" 9. android:supportsRtl="true" 10. android:theme="@style/AppTheme"> 11. <activity android:name=".MainActivity"> 12. <intent-filter> 13. <action android:name="android.intent.action.MAIN" /> 15. <category android:name="android.intent.category.LAUNCHER" /> 16. </intent-filter> 17. </activity> 18. **<activity android:name=".MoveActivity" />** 19. </application> 21. </manifest> |

Perhatikan bahwa MoveActivity sudah teregistrasi di AndroidManifest.xml .Sekarang sudah aman jika kita melakukan perpindahan activitydari MainActivity ke MoveActivity.

Jika kita lupa meregistrasikan Activitybaru ke dalam berkas AndroidManifest.xml, maka akan terjadi  eror seperti ini "android.content.ActivityNotFoundException:Unable to find explicit activity class".

#### **Intent Filter**

Berikutnya, perhatikan pada kode dengan tag intent-filter yang ada di dalam MainActivity.

|  |
| --- |
| 1. <intent-filter> 2. <action android:name="android.intent.action.MAIN" /> 3. <category android:name="android.intent.category.LAUNCHER" /> 4. </intent-filter> |

Intent-filter merupakan mekanisme untuk menentukan bagaimana suatu activitydijalankan oleh Android Runtime (ART) atau Dalvik Virtual Machine (DVM).

|  |
| --- |
| 1. <action android:name="android.intent.action.MAIN" /> |

Baris di atas bermakna bahwa MainActivity menjadi entry point ke aplikasi.

|  |
| --- |
| 1. <category android:name="android.intent.category.LAUNCHER" /> |

Kode di atas menandakan bahwa MainActivity akan dikategorikan sebagai activitylauncher**.**Ini menandakan bahwa activity ini akan muncul di halaman home screen dalam bentuk launcher.

#### **Pindah Activity**

Selanjutnya, perhatikan kode berikut.

|  |
| --- |
| Java |
| 1. **Intent moveIntent = new Intent(MainActivity.this, MoveActivity.class);** 2. startActivity(moveIntent); |

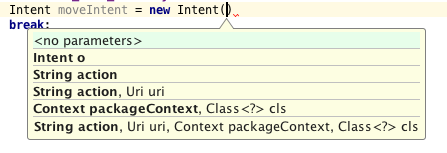
|  |
| --- |
| Kotlin |
| 1. **val moveIntent = Intent(this@MainActivity, MoveActivity::class.java)** 2. startActivity(moveIntent) |

Kita membuat suatu obyek Intentdengan cara seperti di atas dengan memberikan kelas Activityasal (MainActivity.this) dan kelas Activitytujuan (MoveActivity.class) pada konstruktor kelas Intent.

### 

Kelas asal selalu menggunakan .thisyang menandakan obyek kelas saat ini. Kelas tujuan selalu menggunakan .class**.**

Jika kita menggunakan code assistant (tekan ctrl + spasi) dari Android Studio, maka akan tampil rekomendasi (code hint) sebagai berikut:



Pada konteks di atas kita memilih Context packageContext, Class cls sebagai inputan untuk nilai constructor intent.

Kemudian perhatikan baris kode berikutnya.

|  |
| --- |
| Java |
| 1. Intent moveIntent = new Intent(MainActivity.this, MoveActivity.class); 2. **startActivity(moveIntent);** |

|  |
| --- |
| Kotlin |
| 1. val moveIntent = Intent(this@MainActivity, MoveActivity::class.java) 2. **startActivity(moveIntent)** |

startActivity(moveIntent)metodeini akan menjalankan activitybaru tanpa membawa data. Obyek intentyang diinputkan adalah obyek moveIntentyang akan ketika kode ini dijalankan maka akan membuka MoveActivity.

Pada modul ini kita telah berhasil memindahkan satu Activity ke Activity lain dengan tidak membawa data. Pada bagian selanjutnya kita akan membuat suatu Intent yang di dalamnya akan membawa data ke Activity tujuan.