**INTENT EXPLICIT**

**Praktikum Intent Explicit**

Selanjutnya kita akan membuat sebuah intent yang di dalamnya akan membawa data ke Activity tujuan.

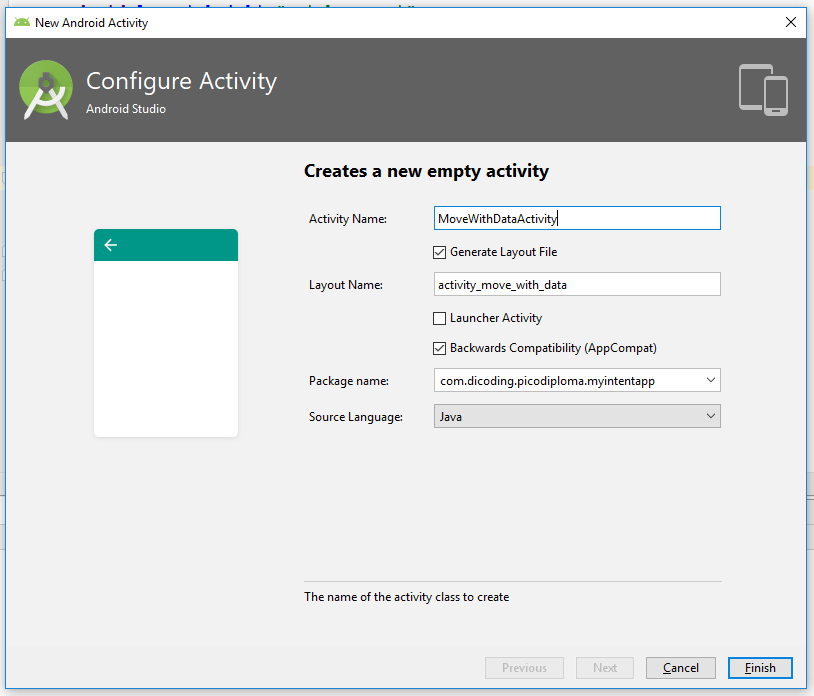
1. Bukaactivity\_main.xml, kemudian tambahkan satu tombol lagi di bawah tombol sebelumnya. Kode activity\_main.xml kita  pun, jadi seperti ini.

|  |
| --- |
| 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. ... 14. <Button 15. android:id="@+id/btn\_move\_activity\_data" 16. android:layout\_width="match\_parent" 17. android:layout\_height="wrap\_content" 18. android:layout\_marginBottom="@dimen/activity\_vertical\_margin" 19. android:text="@string/move\_with\_data" /> 20. </LinearLayout> |

1. Setelah selesai dengan penambahan pada berkas activity\_main.xml, maka lanjutkan dengan menambahkan beberapa baris berikut di MainActivity.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. .... 9. **Button btnMoveWithDataActivity = findViewById(R.id.*btn\_move\_activity\_data*);** 10. **btnMoveWithDataActivity.setOnClickListener(this);** 11. } 13. @Override 14. public void onClick(View v) { 15. switch (v.getId()){ 16. case R.id.*btn\_move\_activity*: 17. Intent moveIntent = new Intent(MainActivity.this, MoveActivity.class); 18. startActivity(moveIntent); 19. break; 20. **case R.id.*btn\_move\_activity\_data*:** 21. **break;** 22. } 23. } 24. } |

1. Selanjutnya, buat Activity baru lagi seperti cara sebelumnya dan beri nama MoveWithDataActivity**.**



1. Lalu pada layout activity\_move\_with\_data.xml kita tambahkan sebuah TextView ber-ID untuk menampilkan data yang dikirimkan dari Activity asal.

|  |
| --- |
| 1. <?xml version="1.0" encoding="utf-8"?> 2. <RelativeLayout 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:paddingBottom="@dimen/activity\_vertical\_margin" 7. android:paddingLeft="@dimen/activity\_horizontal\_margin" 8. android:paddingRight="@dimen/activity\_horizontal\_margin" 9. android:paddingTop="@dimen/activity\_vertical\_margin"> 11. **<TextView** 12. **android:id="@+id/tv\_data\_received"** 13. **android:layout\_width="match\_parent"** 14. **android:layout\_height="wrap\_content"** 15. **android:text="@string/data\_received" />** 16. </RelativeLayout> |

1. Selanjutnya pada MoveWithDataActivity.java kita kondisikan TextView seperti berikut ini:

|  |
| --- |
| Java |
| 1. public class MoveWithDataActivity extends AppCompatActivity { 3. @Override 4. protected void onCreate(Bundle savedInstanceState) { 5. super.onCreate(savedInstanceState); 6. setContentView(R.layout.activity\_move\_with\_data); 8. **TextView tvDataReceived = findViewById(R.id.tv\_data\_received);** 9. } 10. } |

|  |
| --- |
| Kotlin |
| 1. class MoveWithDataActivity : AppCompatActivity() { 3. override fun onCreate(savedInstanceState: Bundle?) { 4. super.onCreate(savedInstanceState) 5. setContentView(R.layout.activity\_move\_with\_data) 7. **val tvDataReceived: TextView = findViewById(R.id.tv\_data\_received)** 8. } 9. } |

1. Selanjutnya untuk menerima data dari Activity asal, kondisikan MoveWithDataActivity seperti berikut:

|  |
| --- |
| Java |
| 1. public class MoveWithDataActivity extends AppCompatActivity { 2. **public static final String EXTRA\_AGE = "extra\_age";** 3. **public static final String EXTRA\_NAME = "extra\_name";** 5. @Override 6. protected void onCreate(Bundle savedInstanceState) { 7. super.onCreate(savedInstanceState); 8. setContentView(R.layout.activity\_move\_with\_data); 10. TextView tvDataReceived = findViewById(R.id.tv\_data\_received); 11. **String name = getIntent().getStringExtra(EXTRA\_NAME);** 12. **int age = getIntent().getIntExtra(EXTRA\_AGE, 0);** 13. **String text = "Name : " + name + ",\nYour Age : " + age;** 14. **tvDataReceived.setText(text);** 15. } 16. } |

|  |
| --- |
| Kotlin |
| 1. class MoveWithDataActivity : AppCompatActivity() { 3. override fun onCreate(savedInstanceState: Bundle?) { 4. super.onCreate(savedInstanceState) 5. setContentView(R.layout.activity\_move\_with\_data) 7. val tvDataReceived: TextView = findViewById(R.id.tv\_data\_received) 9. **val name = intent.getStringExtra(EXTRA\_NAME)** 10. **val age = intent.getIntExtra(EXTRA\_AGE, 0)** 11. **val text = "Name : $name, Your Age : $age"** 12. **tvDataReceived.text = text** 13. } 15. **companion object {** 16. **const val EXTRA\_AGE = "extra\_age"** 17. **const val EXTRA\_NAME = "extra\_name"** 18. **}** 19. } |

1. Dan sekarang saatnya kita menambahkan obyek Intent pada MainActivity.java seperti baris yang ditebalkan.

|  |
| --- |
| 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. **case R.id.*btn\_move\_activity\_data*:** 9. **Intent moveWithDataIntent = new Intent(MainActivity.this, MoveWithDataActivity.class);** 10. **moveWithDataIntent.putExtra(MoveWithDataActivity.*EXTRA\_NAME*, "DicodingAcademy Boy");** 11. **moveWithDataIntent.putExtra(MoveWithDataActivity.*EXTRA\_AGE*, 5);** 12. **startActivity(moveWithDataIntent);** 13. **break;** 14. } 15. } |

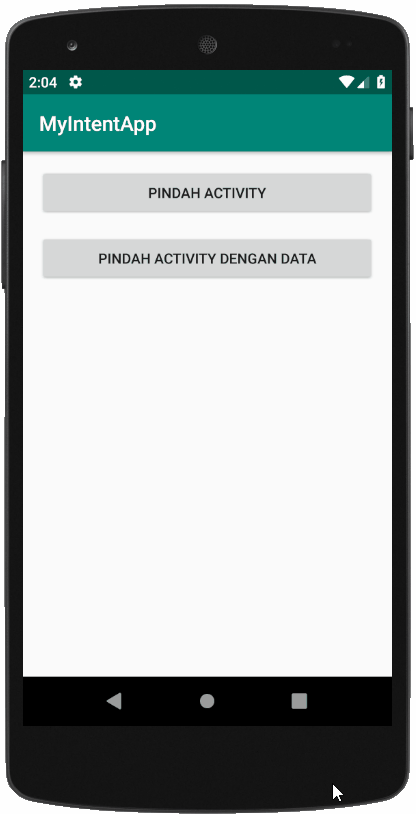
|  |
| --- |
| Kotlin |
| 1. override fun onClick(v: View) { 2. when (v.id) { 3. R.id.btn\_move\_activity -> { 4. val moveIntent = Intent(this@MainActivity, MoveActivity::class.java) 5. startActivity(moveIntent) 6. } 8. **R.id.btn\_move\_activity\_data -> {** 9. **val moveWithDataIntent = Intent(this@MainActivity, MoveWithDataActivity::class.java)** 10. **moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_NAME, "DicodingAcademy Boy")** 11. **moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_AGE, 5)** 12. **startActivity(moveWithDataIntent)** 13. **}** 14. } 15. } |

1. Maka MainActivity.java akan jadi 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); 11. Button btnMoveWithDataActivity = findViewById(R.id.btn\_move\_activity\_data); 12. btnMoveWithDataActivity.setOnClickListener(this); 13. } 15. @Override 16. public void onClick(View v) { 17. switch (v.getId()) { 18. case R.id.btn\_move\_activity: 19. Intent moveIntent = new Intent(MainActivity.this, MoveActivity.class); 20. startActivity(moveIntent); 21. break; 23. case R.id.btn\_move\_activity\_data: 24. Intent moveWithDataIntent = new Intent(MainActivity.this, MoveWithDataActivity.class); 25. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_NAME, "DicodingAcademy Boy"); 26. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_AGE, 5); 27. startActivity(moveWithDataIntent); 28. break; 29. } 30. } 31. } |

|  |
| --- |
| 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) 10. val btnMoveWithDataActivity: Button = findViewById(R.id.btn\_move\_activity\_data) 11. btnMoveWithDataActivity.setOnClickListener(this) 12. } 14. override fun onClick(v: View) { 15. when (v.id) { 16. R.id.btn\_move\_activity -> { 17. val moveIntent = Intent(this@MainActivity, MoveActivity::class.java) 18. startActivity(moveIntent) 19. } 21. R.id.btn\_move\_activity\_data -> { 22. val moveWithDataIntent = Intent(this@MainActivity, MoveWithDataActivity::class.java) 23. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_NAME, "DicodingAcademy Boy") 24. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_AGE, 5) 25. startActivity(moveWithDataIntent) 26. } 27. } 28. } 29. } |

1. Silakan jalankan aplikasi Anda, kemudian coba klik pada tombol **Pindah Activity dengan Data**. Seharusnya Anda sudah bisa memindahkan satu Activity ke Activity lain dengan membawa data.



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

Pada bagian sebelumnya Anda sudah mempelajari bagaimana memindahkan satu Activity ke Activity lain dengan membawa data. Dan itu sangat penting karena ketika kita mengembangkan suatu aplikasi Android yang kompleks, akan ada banyak Activity yang terlibat. Untuk memberikan pengalaman yang baik kepada pengguna, perpindahan Activity dengan data, sangat krusial.

#### **Put Extra**

Perhatikan kode di bawah ini.

|  |
| --- |
| Java |
| 1. Intent moveWithDataIntent = new Intent(MainActivity.this, MoveWithDataActivity.class); 2. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_NAME, "DicodingAcademy Boy"); 3. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_AGE, 5); 4. startActivity(moveWithDataIntent); |

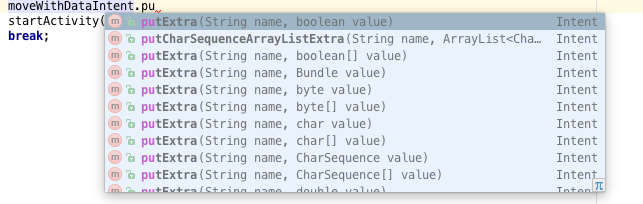
|  |
| --- |
| Kotlin |
| 1. val moveWithDataIntent = Intent(this@MainActivity, MoveWithDataActivity::class.java) 2. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_NAME, "DicodingAcademy Boy") 3. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_AGE, 5) 4. startActivity(moveWithDataIntent) |

Perbedaan mendasar antara memindahkan Activity dengan membawa data atau tidak, adalah dengan menempatkan data ke obyek Intent pada baris ini.

|  |
| --- |
| Java |
| 1. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_NAME, "DicodingAcademy Boy"); 2. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_AGE, 5); |

|  |
| --- |
| Kotlin |
| 1. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_NAME, "DicodingAcademy Boy") 2. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_AGE, 5) |

Kita memanfaatkan metode putExtra() untuk mengirimkan data bersamaan dengan obyek Intent. Sedangkan metode putExtra() itu sendiri merupakan metode yang menampung pasangan key-value dan memiliki beberapa pilihan tipe input seperti berikut:



Hampir semua tipe data untuk input value didukung oleh metode putExtra().

|  |
| --- |
| Java |
| 1. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_NAME, "DicodingAcademy Boy"); |

|  |
| --- |
| Kotlin |
| 1. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_NAME, "DicodingAcademy Boy") |

|  |  |
| --- | --- |
| **Name** | MoveWithDataActivity.EXTRA\_NAME dimana EXTRA\_NAME adalah variabel static bertipe data string dan bernilai “extra\_name” pada MoveWithDataActivity.java. Penentuan nilai untuk key parameter untuk intent adalah bebas, di sini kami merekomendasikan format terbaik yang biasa diimplementasikan. |
| **Value** | DicodingAcademy Boy dengan tipe data string. |

#### **Get Data**

Setelah data dikirimkan, selanjutnya adalah mengambil data tersebut.

Di sini kita akan mengirimkan data bertipe string ke MoveWithDataActivity. Di dalam MoveWithdataActivity kita akan mengambil nilai data berdasarkan key yang dikirimkan dengan menggunakan metode getIntent().getStringExtra(key). Implementasinya sebagai berikut:

|  |
| --- |
| Java |
| 1. String name = getIntent().getStringExtra(*EXTRA\_NAME*); |

|  |
| --- |
| Kotlin |
| 1. val name = intent.getStringExtra(*EXTRA\_NAME*) |

**Catatan** : Key yang dikirimkan melalui putExtra() harus sama dengan key sewaktu mengambil nilai dari data yang dikirimkan melalui getStringExtra().

Dalam konteks di atas, key yang digunakan untuk mengirim dan mengambil nilai data adalah sama, yaitu EXTRA\_NAME (yang bernilai “extra\_name”). Nilai dari data yang dikirimkan melalui intent disimpan ke dalam variabel name bertipe data string.

Begitu juga dengan variable age.

|  |
| --- |
| Java |
| 1. int age = getIntent().getIntExtra(*EXTRA\_AGE*, 0); |

|  |
| --- |
| Kotlin |
| 1. val age = intent.getIntExtra(EXTRA\_AGE, 0) |

Nilai dari variabel age yang bertipe data integer berasal dari getIntent().getIntExtra(Key, nilai default)**.** Key yang digunakan untuk mengirimkan dan mengambil data adalah EXTRA\_AGE (yang bernilai “extra\_age”). Nilai default di sini merupakan nilai yang akan digunakan jika ternyata datanya kosong. Data kosong atau nilainya null bisa terjadi ketika datanya memang tidak ada, atau key-nya tidak sama.

Selamat! Anda telah mempelajari dua intent explicit dengan atau tidak membawa data.

Sebelumnya, mengirimkan data bernilai tunggal dari satu Activity ke Activity lain adalah hal yang mudah. Bernilai tunggal karena data yang dikirimkan berasal dari satu tipe data.

Misalnya pada contoh di atas, pengiriman nilai data name dan age dilakukan secara individu. Yang satu bertipe data string dan yang lainnya bertipe data integer.

Sekarang pertanyaanya bagaimana Anda bisa mengirimkan data dalam bentuk Plain Old Java Object (POJO) dari satu Activity ke Activity lain melalui intent? Caranya adalah dengan mengubah obyek POJO yang Anda miliki ke dalam bentuk obyek parcelable. Di Android Studio sudah ada fasilitas plugin bernama Android Parcelable Generator untuk memudahkan proses tersebut.

**Pada praktikum ini tidak akan dijelaskan tentang parcelable object. Untuk materi yang lebih lengkap dan komprehensif mengenai parcelable object bisa langsung daftar kelas** [**Menjadi Android Developer Expert**](https://www.dicoding.com/academies/14/)**.**