**PRAKTIKUM INTENT IMPLICIT**

Anda sudah belajar bagaimana menggunakan intent dengan tipe eksplisit. Sekarang saatnya Anda melanjutkan ke intent dengan tipe implicit.

1. Buka kembali activity\_main.xmltambahkan satu tombol lagi sebagai berikut :

|  |
| --- |
| 1. **<Button** 2. **android:id="@+id/btn\_dial\_number"** 3. **android:layout\_width="match\_parent"** 4. **android:layout\_height="wrap\_content"** 5. **android:layout\_marginBottom="@dimen/activity\_vertical\_margin"** 6. **android:text="@string/dial\_number" />** |

Sehingga file activity\_main.xml kita sekarang menjadi:

|  |
| --- |
| 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" /> 19. <Button 20. android:id="@+id/btn\_move\_activity\_data" 21. android:layout\_width="match\_parent" 22. android:layout\_height="wrap\_content" 23. android:layout\_marginBottom="@dimen/activity\_vertical\_margin" 24. android:text="@string/move\_with\_data" /> 26. **<Button** 27. **android:id="@+id/btn\_dial\_number"** 28. **android:layout\_width="match\_parent"** 29. **android:layout\_height="wrap\_content"** 30. **android:layout\_marginBottom="@dimen/activity\_vertical\_margin"** 31. **android:text="@string/dial\_number" />** 32. </LinearLayout> |

1. Sekarang, buka kembali MainActivity.java dan lanjutkan kenalkan button yang sudah ditambahkan:

|  |
| --- |
| Java |
| 1. @Override 2. protected void onCreate(Bundle savedInstanceState) { 3. super.onCreate(savedInstanceState); 4. setContentView(R.layout.activity\_main); 6. ... 8. **Button btnDialPhone = findViewById(R.id.btn\_dial\_number);** 9. **btnDialPhone.setOnClickListener(this);** 10. } |

|  |
| --- |
| Kotlin |
| 1. override fun onCreate(savedInstanceState: Bundle?) { 2. super.onCreate(savedInstanceState) 3. setContentView(R.layout.activity\_main) 5. ... 7. **val btnDialPhone: Button = findViewById(R.id.btn\_dial\_number)** 8. **btnDialPhone.setOnClickListener(this)** 9. } |

1. Kemudian dengan melakukan casting tombol Dial Number, set listener dan menambahkan action ketika tombol diklik.

|  |
| --- |
| Java |
| 1. @Override 2. public void onClick(View v) { 3. switch (v.getId()){ 5. ... 7. **case R.id.*btn\_dial\_number*:** 8. **String phoneNumber = "081210841382";** 9. **Intent dialPhoneIntent = new Intent(Intent.*ACTION\_DIAL*, Uri.*parse*("tel:"+phoneNumber));** 10. **startActivity(dialPhoneIntent);** 11. **break;** 12. } 13. } |

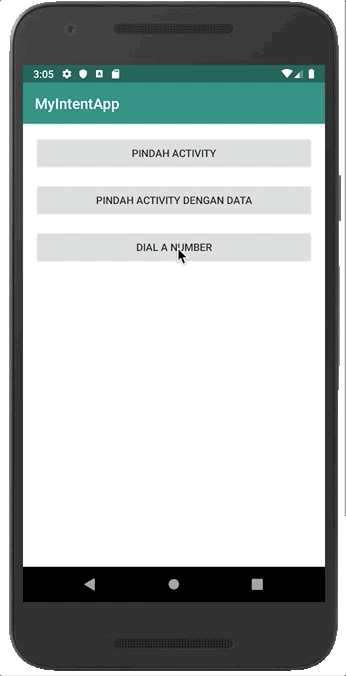
|  |
| --- |
| Kotlin |
| 1. override fun onClick(v: View) { 2. when (v.id) { 3. ... 5. **R.id.btn\_dial\_number -> {** 6. **val phoneNumber = "081210841382"** 7. **val dialPhoneIntent = Intent(Intent.ACTION\_DIAL, Uri.parse("tel:$phoneNumber"))** 8. **startActivity(dialPhoneIntent)** 9. **}** 10. } 11. } |

1. Sehingga kode keseluruhan untuk kelas MainActivity.java saat ini adalah.

|  |
| --- |
| 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); 14. **Button** **btnDialPhone = findViewById(R.id.btn\_dial\_number);** 15. **btnDialPhone.setOnClickListener(this);** 16. } 18. @Override 19. public void onClick(View v) { 20. switch (v.getId()){ 21. case R.id.btn\_move\_activity: 22. Intent moveIntent = new Intent(MainActivity.this, MoveActivity.class); 23. startActivity(moveIntent); 24. break; 26. case R.id.btn\_move\_activity\_data: 27. Intent moveWithDataIntent = new Intent(MainActivity.this, MoveWithDataActivity.class); 28. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_NAME, "DicodingAcademy Boy"); 29. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_AGE, 5); 30. startActivity(moveWithDataIntent); 31. break; 33. **case R.id.btn\_dial\_number:** 34. **String phoneNumber = "081210841382";** 35. **Intent dialPhoneIntent = new Intent(Intent.ACTION\_DIAL, Uri.parse("tel:"+phoneNumber));** 36. **startActivity(dialPhoneIntent);** 37. **break;** 38. } 39. } 40. } |

|  |
| --- |
| 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) 13. **val btnDialPhone: Button = findViewById(R.id.btn\_dial\_number)** 14. **btnDialPhone.setOnClickListener(this)** 15. } 17. override fun onClick(v: View) { 18. when (v.id) { 19. R.id.btn\_move\_activity -> { 20. val moveIntent = Intent(this@MainActivity, MoveActivity::class.java) 21. startActivity(moveIntent) 22. } 24. R.id.btn\_move\_activity\_data -> { 25. val moveWithDataIntent = Intent(this@MainActivity, MoveWithDataActivity::class.java) 26. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_NAME, "DicodingAcademy Boy") 27. moveWithDataIntent.putExtra(MoveWithDataActivity.EXTRA\_AGE, 5) 28. startActivity(moveWithDataIntent) 29. } 31. **R.id.btn\_dial\_number -> {** 32. **val phoneNumber = "081210841382"** 33. **val dialPhoneIntent = Intent(Intent.ACTION\_DIAL, Uri.parse("tel:$phoneNumber"))** 34. **startActivity(dialPhoneIntent)** 35. **}** 36. } 37. } 38. } |

1. Setelah selesai, silakan jalankan lagi aplikasi anda. Yeay, Anda sudah bisa men-dial sebuah nomor telepon melalui aplikasi bawaan dari peranti anda. Selamat!



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

#### **Intent Implicit**

|  |
| --- |
| Java |
| 1. String phoneNumber = "081210841382"; 2. Intent dialPhoneIntent = new Intent(Intent.ACTION\_DIAL, Uri.parse("tel:" + phoneNumber)); 3. startActivity(dialPhoneIntent); |

|  |
| --- |
| Kotlin |
| 1. val phoneNumber = "081210841382" 2. val dialPhoneIntent = Intent(Intent.ACTION\_DIAL, Uri.parse("tel:$phoneNumber")) 3. startActivity(dialPhoneIntent) |

#### **Apa itu URI (Uniform Resource Identifier)**

Buat yang belum tau apa itu Uri, berikut penjelasan singkatnya :

Uri adalah sebuah untaian karakter yang digunakan untuk mengidentifikasi nama, sumber, atau layanan di internet sesuai dengan RFC 2396. Pada Uri.parse("tel:"+phoneNumber)kita melakukan parsing Uri dari bentuk teks string menjadi sebuah obyek uri dengan menggunakan metode static parse(String)**.**Secara struktur, dibagi menjadi :

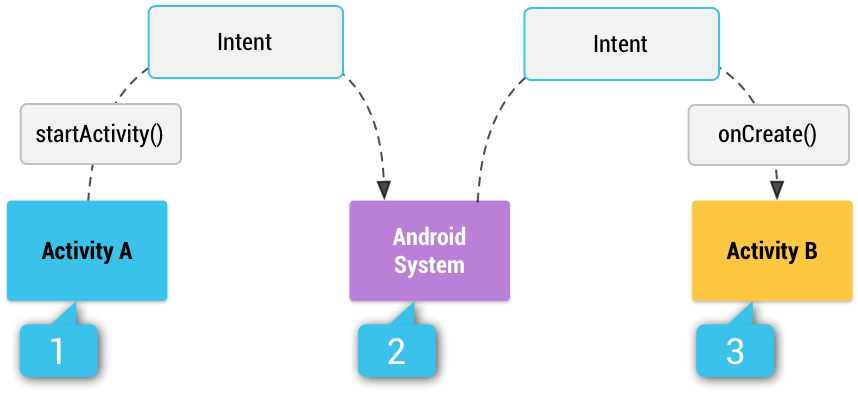


Dimana “**tel**” adalah sebuah skema yang disepakati untuk sumber daya telepon dan phoneNumber adalah variabel string yang bernilai “081210841382”. Skema lain dari Uri seperti “geo” untuk peta, “http” untuk browser sisanya bisa dilihat di halaman ini :

* [Common Intents](https://developer.android.com/guide/components/intents-common.html)

#### **Proses Intent Implicit**

Pada prosesnya, pemanggilan intent secara implicit akan berjalan sesuai dengan diagram di bawah ini.



1. Aplikasi kita menjalankan intent untuk ACTION\_DIAL melalui startActivity()**.**
2. Sistem Android akan melakukan seleksi terhadap semua aplikasi yang memiliki kemampuan untuk menangani action tersebut. Sistem Android akan menentukan aplikasi mana saja yang bisa memproses action berdasarkan intent filter yang telah ditentukan di berkas AndroidManifest.xml pada masing-masing aplikasi.  
   Sistem Android akan menampilkan opsi aplikasi-aplikasi mana saja yang bisa menangani action tersebut ke pengguna.  
   Pengguna memilih salah satu opsi aplikasi dan kemudian sistem Android akan me-routing ke activity pada aplikasi yang dipilih yang memiliki intent-filter untuk aksi ACTION\_DIAL.
3. Aplikasi yang dipilih pun muncul di layar dengan nomor telepon yang sudah diatur.

Anda sudah mempelajari 3 buah intent secara eksplisit dan 1 buah intent secara implicit. Selamat!