

**LAPORAN PRAKTIKUM LAB D1**  
**PENGEMBANGAN BERBASIS PLATFORM**  
“Android Intent”



**DISUSUN OLEH:**  
RESMA ADI NUGROHO  
24060121120021

**DEPARTEMEN INFORMATIKA**  
**FAKULTAS SAINS DAN MATEMATIKA**  
**UNIVERSITAS DIPONEGORO**  
**2023**

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Tujuan**

- a. Mahasiswa mampu memahami Intent.
- b. Mahasiswa mampu membuat aplikasi sederhana dengan menerapkan Intent.

#### **1.2 Rumusan Masalah**

- a. Membuat aplikasi android dengan menerapkan intent pada main activity untuk berpindah ke activity lain dengan data dan tidak dengan data.

## BAB II

### DASAR TEORI

#### 2.1 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)
}
```

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.
}
```

## BAB III

### PEMBAHASAN

#### 3.1 Implementasi Implicit Intent

Untuk membuat implementasi implicit intent pada aplikasi android maka diperlukan dua activity untuk dapat terjadinya perpindahan halaman atau activity. Pada permasalahan ini akan dibuat MainActivity sebagai activity utama yang pertama akan dijalankan kemudian RegistrationActivity merupakan activity yang akan dituju. Sebelum masuk ke dalam logic aplikasi, perlu dibuat terlebih dahulu tampilan aplikasi melalui kode XML seperti kode dibawah ini:

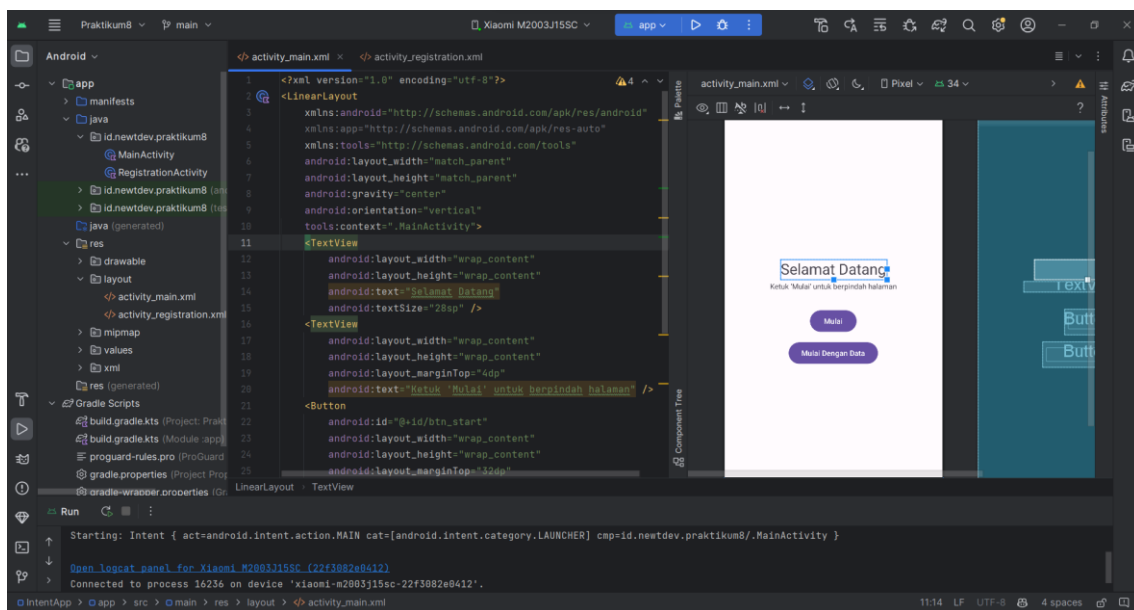
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    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" />
```

```

<Button
    android:id="@+id/btn_with_data"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Mulai Dengan Data"
    android:layout_marginTop="12dp"/>
</LinearLayout>

```

Desain tampilan dari halaman pada aplikasi tersebut akan seperti pada gambar dibawah ini:



Gambar 1 Tampilan Antarmuka Aplikasi

Sedangkan untuk RegistrationActivity kode XML dapat dilihat dibawah ini:

```

<?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:orientation="vertical"
    android:layout_margin="20dp"
    android:gravity="center"
    tools:context=".RegistrationActivity">

```

```

<TextView

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:text="Registration Activity"

/>

<TextView

    android:id="@+id/tv_name"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

/>

<TextView

    android:id="@+id/tv_nim"

    android:layout_width="wrap_content"

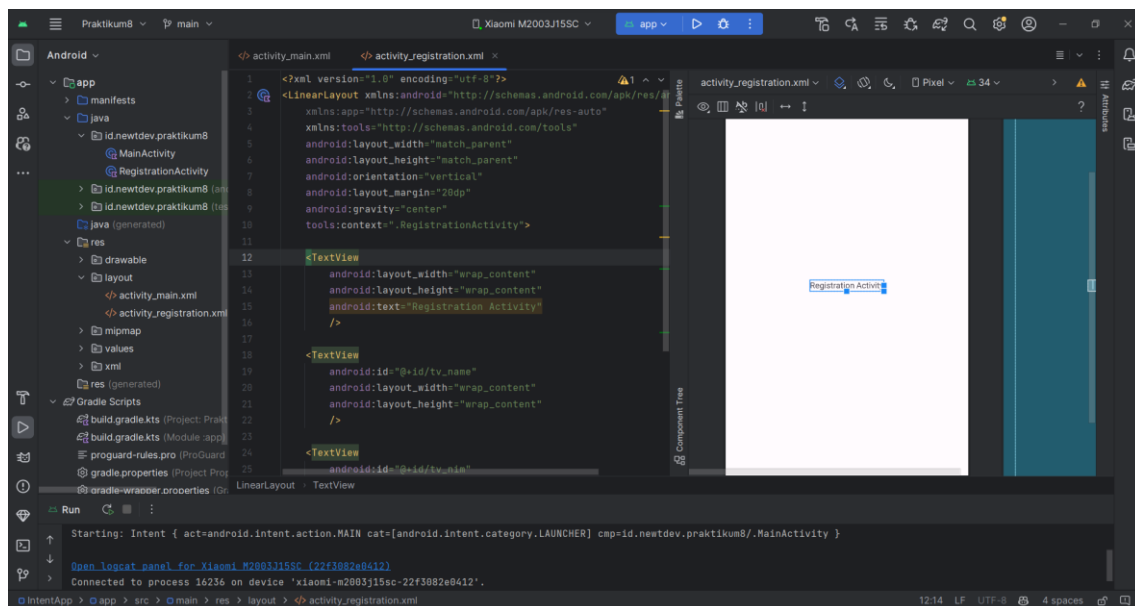
    android:layout_height="wrap_content"

/>

</LinearLayout>

```

Sehingga desain tampilan dari activity tersebut akan seperti dibawah ini:



Gambar 2 Tampilan RegistrationActivity

Setelah tampilan activity dibuat maka selanjutnya dapat dilakukan implementasi intent pada file kotlin activity tersebut.

## File MainActivity.kt

```
package id.newtdev.praktikum8

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import id.newtdev.praktikum8.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding
    private lateinit var mulaiBtn: Button
    private lateinit var withDataBtn: Button

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        var binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        // BINDING
        mulaiBtn = binding.btnStart
        withDataBtn = binding.btnWithData
        // INTENT VAR DECLARE
        val intent = Intent(this, RegistrationActivity::class.java)
        // BUTTON FUNCTION
        // MOVE WITHOUT DATA
        mulaiBtn.setOnClickListener {
            startActivity(intent)
        }
        // MOVE WITH DATA
        withDataBtn.setOnClickListener {
            intent.putExtra("name", "Resma Adi Nugroho")
            intent.putExtra("nim", "24060121120021")
            startActivity(intent)
        }
    }
}
```



Pada kode MainActivity diatas akan berisi fungsi untuk dua button yang telah dibuat pada kode XML. Selain itu sebelum button dapat memiliki fungsi ketika ditekan perlu juga mendefinisikan beberapa variable yang dibutuhkan seperti bagian yang sudah diberi komentar pada kode diatas.

```
private lateinit var binding: ActivityMainBinding
private lateinit var mulaiBtn: Button
private lateinit var withDataBtn: Button
```

Pada bagian kode diatas berisi inisialisasi awal untuk variable binding, mulaiBtn, dan withDataBtn. Ketiga variable tersebut diset lateinit yakni inisialisasi akan dilakukan nanti ketika onCreate. Variable binding digunakan untuk melakukan binding secara langsung nantinya tanpa perlu findViewById. Namun sebelum menggunakan fitur ini, kita perlu mengaktifkan buildFeatures dengan viewBinding true.

```
var binding = ActivityMainBinding.inflate(layoutInflater)
setContentView(binding.root)
mulaiBtn = binding.btnStart
withDataBtn = binding.btnWithData
```

Karena kita telah mendefinisikan tiga variable sebelumnya yang belum memiliki nilai maka pada saat onCreate kita perlu untuk memberikan nilai pada variable tersebut. Untuk variable binding berisi nilai ActivityMainBinding yang merupakan generated class karena kita telah mengaktifkan fitur viewBinding. Kemudian setContentView terdapat modifikasi karena kita menggunakan binding yakni dengan binding.root. Kemudian agar dapat berpindah ke activity lain maka perlu kita perlu membuat variable intent yang berisi seperti pada kode dibawah ini:

```
val intent = Intent(this, RegistrationActivity::class.java)
```

Pada kode diatas intent akan berpindah dari activity sekarang ke RegistrationActivity. Namun karena intent baru dideklarasikan dalam sebuah variable maka kita perlu memanggil fungsi startActivity agar kita dapat berpindah ke activity yang sebelumnya sudah ditentukan pada variable intent.

```
// MOVE WITHOUT DATA
mulaiBtn.setOnClickListener {
    startActivity(intent)
}
```

```
// MOVE WITH DATA

withDataBtn.setOnClickListener {

    intent.putExtra("name", "Resma Adi Nugroho")

    intent.putExtra("nim", "24060121120021")

    startActivity(intent)

}
```

Terdapat dua button yang sama-sama akan menuju activity yang sama namun untuk button pertama akan berpindah tanpa membawa data. Sedangkan untuk activity yang kedua akan membawa data name dan nim. Untuk perpindahan activity yang membawa data kita perlu memanggil putExtra kemudian menambahkan companion object yang berisi key value yang akan dibawa ke activity lain. Pada hal ini yakni name berisi nilai Resma Adi Nugroho kemudian untuk key nim dengan value 24060121120021.

#### File RegistrationActivity.kt

```
class RegistrationActivity : AppCompatActivity() {

    private lateinit var binding: ActivityRegistrationBinding
    private lateinit var tvName: TextView
    private lateinit var tvNim: TextView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        var binding = ActivityRegistrationBinding.inflate(layoutInflater)
        setContentView(binding.root)

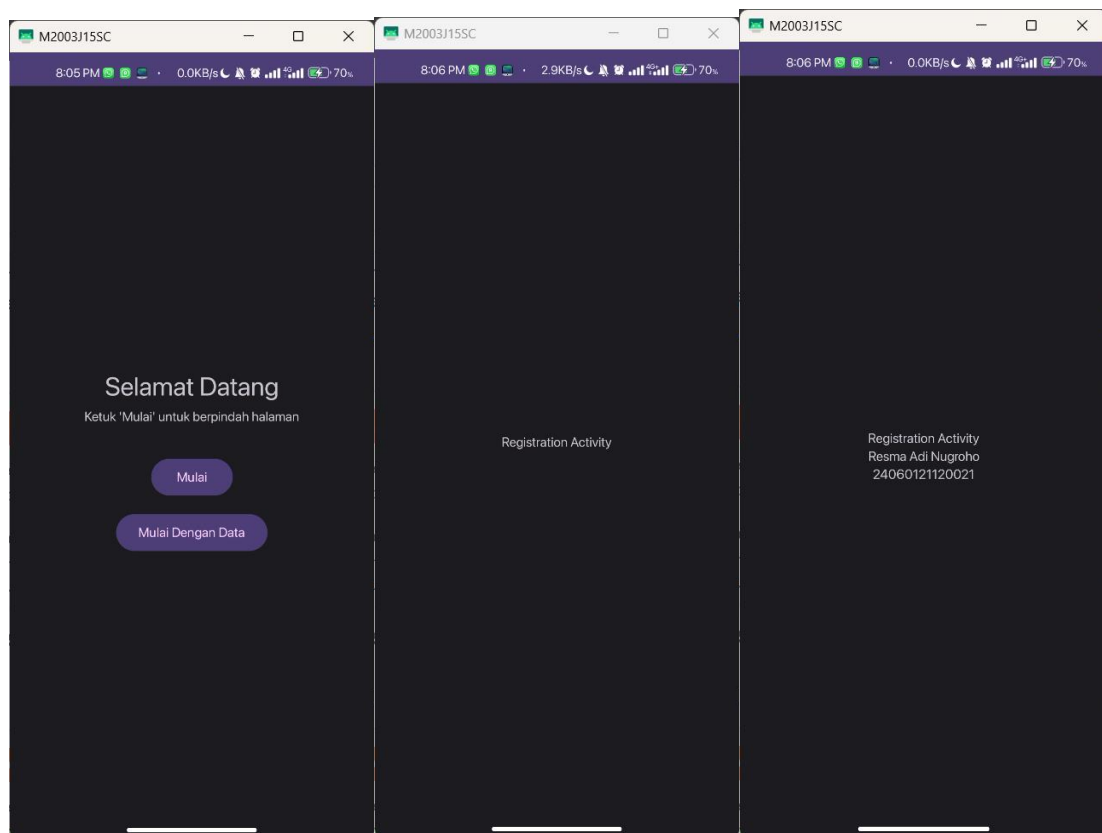
        // GET DATA FROM INTENT
        val name = intent.getStringExtra("name")
        val nim = intent.getStringExtra("nim")

        // BINDING
        tvName = binding.tvName
        tvNim = binding.tvNim

        // SET DATA FROM INTENT
        tvName.text = name
        tvNim.text = nim
    }
}
```

Kode diatas merupakan kode pada RegistrationActivity, untuk menampilkan data dari intent yang telah kita awa maka dapat digunakan getStringExtra dari intent sekarang. Selain string, ada juga variasi lain ketika mendapatkan data dari intent contohnya seperti getBooleanExtra, getIntExtra, dan masih banyak lagi yang dapat disesuaikan dengan tipe data value yang dibawa. Kemudian karena hasil data dari intent ingin kita tampilkan maka perlu juga untuk mendefinisikan dan binding komponen TextView dengan cara yang sama seperti pada kode sebelumnya. Kemudian nilai text dari tvName dan tvNim dapat diset sesuai data yang didapatkan dari intent.

### 3.2 Hasil Aplikasi



*Gambar 3 Hasil Aplikasi*

## **BAB IV**

### **PENUTUP**

Intent dalam Android Studio adalah mekanisme yang memungkinkan komunikasi antara komponen aplikasi, seperti aktivitas, layanan, dan pengumpan balik. Ini merupakan wadah untuk mengirimkan pesan yang berisi informasi untuk memicu aktivitas tertentu, memulai layanan, atau meminta respons dari komponen lain dalam aplikasi Android. Intent berperan sebagai jembatan komunikasi yang kuat, memungkinkan aplikasi untuk berinteraksi dengan komponen internal dan eksternal, termasuk aplikasi lain di perangkat.

Salah satu kegunaan utama Intent adalah untuk memicu perpindahan antar layar atau aktivitas dalam aplikasi. Dengan menggunakan Intent, pengembang dapat menavigasi dari satu layar ke layar lain dalam aplikasi Android. Ini membantu dalam mengatur alur navigasi, memungkinkan pengguna untuk berpindah antar bagian yang berbeda dalam aplikasi dengan lancar. Selain itu, Intent juga dapat digunakan untuk memicu aksi tertentu, seperti memanggil layanan tertentu untuk mengeksekusi tugas tertentu atau meminta respon dari komponen lain, seperti hasil dari aktivitas pemilihan gambar atau data dari sensor.

Selain fungsi navigasi dan komunikasi antar-komponen, Intent juga mendukung pengiriman data antar komponen aplikasi. Dengan membawa informasi dalam bentuk data ekstra (extra) atau melalui URI, Intent memfasilitasi pertukaran data di antara aktivitas, layanan, atau komponen aplikasi lainnya. Ini memungkinkan pengiriman dan penerimaan data dalam bentuk string, bilangan bulat, objek parcelable, atau jenis data lainnya, yang memperluas kemampuan interaksi aplikasi Android secara keseluruhan. Intent di Android Studio sangat penting karena menjadi fondasi utama bagi interaksi antar-komponen dalam pengembangan aplikasi Android.