

**LAPORAN PRAKTIKUM**  
**PRAKTIKUM 9 :**  
**EXPLICIT INTENT DAN IMPLICIT INTENT**



**Disusun oleh:**  
**Emerio Kevin Aryaputra**  
**24060121120012**

**PRAKTIKUM PEMROGRAMAN BERBASIS PLATFORM**  
**LAB A2**

**DEPARTEMEN ILMU KOMPUTER/INFORMATIKA**  
**FAKULTAS SAINS DAN MATEMATIKA**  
**UNIVERSITAS DIPONEGORO**  
**2023**

## **KATA PENGANTAR**

Segala puji syukur kepada Yesus Kristus atas kasih dan karunia-Nya yang diberikan, sehingga penulis dapat menyelesaikan laporan praktikum dengan judul “Explicit Intent dan Implicit Intent” dengan baik. Adapun laporan ini disusun sebagai bagian dari penugasan praktikum Pemrograman Berbasis Platform.

Adapun maksud dan tujuan adanya laporan praktikum ini adalah dapat memahami perbedaan antara Explicit Intent dan Implicit Intent serta mampu membuat aplikasi Android sederhana dengan memanfaatkan Explicit Intent dan Implicit Intent.

Dalam penyusunan laporan ini, penulis mengucapkan terima kasih kepada Kak Jason Al-Hilal Sabda Dewa yang telah membantu penulis dalam pemahaman materi dan dasar-dasar praktikum.

Dengan ini, penulis menyadari bahwa laporan praktikum ini masih jauh dari kata sempurna. Untuk itu, penulis dengan sangat terbuka menerima kritik dan saran dari para pembaca. Semoga laporan praktikum ini bermanfaat untuk para pembaca dan penulis, khususnya dalam usaha meningkatkan kemampuan tentang pemrograman berbasis platform.

Semarang, 18 November 2023

Emerio Kevin Aryaputra

# **BAB I**

## **PENDAHULUAN**

### **1.1. Rumusan Masalah**

- 1.1.1. Buat sebuah MainActivity yang dapat berpindah ke RegistrationActivity.
- 1.1.2. Buat RegistrationActivity yang dapat menerima isian nama, NIM, nomor telepon, dan email.
- 1.1.3. Buat sebuah RegistrationActivity yang dapat berpindah ke DetailActivity.
- 1.1.4. Buat sebuah Implicit Intent pada halaman DetailActivity untuk membuka aplikasi yang mendukung fungsionalitas untuk menelepon. Nomor telepon sudah terisi otomatis dengan data yang dikirimkan dari RegistrationActivity.
- 1.1.5. Buat sebuah Implicit Intent pada halaman DetailActivity untuk membuka aplikasi yang mendukung fungsionalitas untuk mengirim email. Penerima email dan subjek email sudah terisi otomatis dengan data yang dikirimkan dari RegistrationActivity.

### **1.2. Tujuan**

- 1.2.1. Mengetahui cara untuk membuat aplikasi Android yang memanfaatkan Explicit Intent.
- 1.2.2. Mengetahui cara untuk membuat aplikasi Android yang memanfaatkan Implicit Intent.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Intent**

Intent merupakan sebuah 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, salah satunya memulai sebuah Activity.

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

#### **2.2. Explicit Intent**

Explicit Intent merupakan Intent yang memiliki tujuan yang jelas. Hal tersebut ditandai dengan didefinisikannya tujuan dari Intent seperti nama suatu komponen. Explicit Intent inilah yang biasanya digunakan untuk memulai suatu komponen seperti berpindah Activity.

#### **2.3. Implicit Intent**

Implicit Intent merupakan Intent yang tidak menyebutkan komponen yang ingin dituju. Implicit Intent hanya mendeklarasikan sebuah tugas umum yang dapat dilakukan oleh komponen pada aplikasi lain. Contohnya, ketika ingin mengirimkan suatu email, Implicit Intent dapat digunakan untuk meminta aplikasi lain yang memiliki fungsi untuk mengirimkan email agar dapat melaksanakan tugas tersebut.

#### **2.4. Cara Kerja Intent**

Ketika Intent sudah memiliki komponen yang ingin dipanggil seperti Explicit Intent, sistem akan langsung memulai komponen yang akan digunakan. Sedangkan jika menggunakan Implicit Intent, perlu dilakukan pencarian pada aplikasi yang dapat mendukung aksi yang akan dilakukan. Cara kerjanya secara lengkap :

- 2.4.1. Sebuah Activity membuat sebuah Intent yang berisikan action serta komponen yang ingin dipanggil, lalu mengirimkannya ke method startActivity().
- 2.4.2. Sistem Android akan mencari seluruh Intent-filters pada aplikasi yang terpasang di perangkat yang cocok dengan Intent yang dikirimkan.
- 2.4.3. Jika aplikasi yang cocok ditemukan, sistem akan memulai Activity yang cocok dengan mengeksekusi method onCreate() serta mengirimkan objek Intent-nya.

## **BAB III**

### **PEMBAHASAN**

#### **3.1. Soal**

- 3.1.1. Buat sebuah MainActivity yang dapat berpindah ke RegistrationActivity.
- 3.1.2. Buat RegistrationActivity yang dapat menerima isian nama, NIM, nomor telepon, dan email.
- 3.1.3. Buat sebuah RegistrationActivity yang dapat berpindah ke DetailActivity.
- 3.1.4. Buat sebuah Implicit Intent pada halaman DetailActivity untuk membuka aplikasi yang mendukung fungsionalitas untuk menelepon. Nomor telepon sudah terisi otomatis dengan data yang dikirimkan dari RegistrationActivity.
- 3.1.5. Buat sebuah Implicit Intent pada halaman DetailActivity untuk membuka aplikasi yang mendukung fungsionalitas untuk mengirim email. Penerima email dan subjek email sudah terisi otomatis dengan data yang dikirimkan dari RegistrationActivity.

#### **3.2. Source Code**

Terdapat tujuh file penting untuk membuat aplikasi ini, yaitu activity\_main.xml, MainActivity.kt, activity\_registration.xml, RegistrationActivity.kt, activity\_detail.xml, DetailActivity.kt, serta strings.xml.

##### **3.2.1. activity\_main.xml**

```
<?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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```

        android:text="@string/selamat_datang"
        android:textSize="28sp"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="4dp"
        android:text="@string/perintah"
    />

    <Button
        android:id="@+id/btn_start"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:text="@string/mulai"
    />

</LinearLayout>

```

### 3.2.2. MainActivity.kt

```

package com.android.intentapp2

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val btnStart: Button = findViewById(R.id.btn_start)
        btnStart.setOnClickListener {
            run {
                val registrationIntent = Intent(applicationContext,
RegistrationActivity::class.java)
                startActivity(registrationIntent)
            }
        }
    }
}

```

### 3.2.3. activity\_registration.xml

```
<?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_horizontal"
    android:orientation="vertical"
    android:padding="24dp"
    tools:context=".RegistrationActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/identitas_mahasiswa"
        android:textSize="24sp"
        />

    <EditText
        android:id="@+id/et_name"
        android:layout_width="match_parent"
        android:layout_height="48dp"
        android:layout_marginTop="64dp"
        android:autofillHints="name"
        android:hint="@string/et_name_hint"
        android:inputType="textPersonName"
        android:textColor="@color/black"
        tools:ignore="TextContrastCheck"
        />

    <EditText
        android:id="@+id/et_nim"
        android:layout_width="match_parent"
        android:layout_height="48dp"
        android:layout_marginTop="8dp"
        android:autofillHints="nim"
        android:hint="@string/et_nim_hint"
        android:inputType="number"
        android:textColor="@color/black"
        tools:ignore="TextContrastCheck"
        />
```



```

<EditText
    android:id="@+id/et_phone_number"
    android:layout_width="match_parent"
    android:layout_height="48dp"
    android:layout_marginTop="8dp"
    android:autofillHints="phone"
    android:hint="@string/et_phone_number_hint"
    android:inputType="phone"
    android:textColor="@color/black"
    tools:ignore="TextContrastCheck"
/>

<EditText
    android:id="@+id/et_email"
    android:layout_width="match_parent"
    android:layout_height="48dp"
    android:layout_marginTop="8dp"
    android:autofillHints="phone"
    android:hint="@string/et_email_hint"
    android:inputType="textEmailAddress"
    android:textColor="@color/black"
    tools:ignore="TextContrastCheck"
/>

<Button
    android:id="@+id/btn_next"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:text="@string/btn_next"
/>

</LinearLayout>

```

### 3.2.4. RegistrationActivity.kt

```

package com.android.intentapp2

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText

```

```

class RegistrationActivity : AppCompatActivity() {
    private var etName: EditText? = null
    private var etNIM: EditText? = null
    private var etPhone: EditText? = null
    private var etEmail: EditText? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_registration)

        etName = findViewById(R.id.et_name)
        etNIM = findViewById(R.id.et_nim)
        etPhone = findViewById(R.id.et_phone_number)
        etEmail = findViewById(R.id.et_email)

        val btnNext: Button = findViewById(R.id.btn_next)
        btnNext.setOnClickListener {
            run {
                val name = etName!!.text.toString().trim { it <= ' ' }
                val nim = etNIM!!.text.toString().trim { it <= ' ' }
                val phone = etPhone!!.text.toString().trim { it <= ' ' }
                val email = etEmail!!.text.toString().trim { it <= ' ' }

                var isEmptyField = false
                if (name.isEmpty()) {
                    etName!!.error = "Field harus diisi!"
                    isEmptyField = true
                }
                if (nim.isEmpty()) {
                    etNIM!!.error = "Field harus diisi!"
                    isEmptyField = true
                }
                if (phone.isEmpty()) {
                    etPhone!!.error = "Field harus diisi!"
                    isEmptyField = true
                }
                if (email.isEmpty()) {
                    etEmail!!.error = "Field harus diisi!"
                    isEmptyField = true
                }

                if (!isEmptyField) {

```

```

        val detailIntent = Intent(applicationContext,
DetailActivity::class.java)
        detailIntent.putExtra(DetailActivity.KEY_NAME, name)
        detailIntent.putExtra(DetailActivity.KEY_NIM, nim)
        detailIntent.putExtra(DetailActivity.KEY_PHONE,
phone)
        detailIntent.putExtra(DetailActivity.KEY_EMAIL,
email)
        startActivity(detailIntent)
    }
}
}
}
}
}

```

### 3.2.5. activity\_detail.xml

```

<?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:orientation="vertical"
    android:padding="24dp"
    tools:context=".DetailActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/identitas"
        android:textSize="24sp"/>

    <TextView
        android:id="@+id/tv_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:textSize="18sp"
        tools:text="Nama : "/>

    <TextView
        android:id="@+id/tv_nim"
        android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:textSize="18sp"
        tools:text="NIM : "/>

<TextView
    android:id="@+id/tv_phone"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:textSize="18sp"
    tools:text="No. HP : "/>

<TextView
    android:id="@+id/tv_email"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:textSize="18sp"
    tools:text="Email : "/>

<Button
    android:id="@+id/btn_phone_call"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:text="@string/lakukan_panggilan"/>

<Button
    android:id="@+id/btn_send_email"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text="@string/kirim_email"/>

</LinearLayout>

```

### 3.2.6. DetailActivity.kt

```

package com.android.intentapp2

import android.content.Intent
import android.net.Uri
import androidx.appcompat.app.AppCompatActivity

```

```

import android.os.Bundle
import android.widget.Button
import android.widget.TextView

class DetailActivity : AppCompatActivity() {
    companion object {
        const val KEY_NAME = "key_name"
        const val KEY_NIM = "key_nim"
        const val KEY_PHONE = "key_phone"
        const val KEY_EMAIL = "key_email"
    }
    private var phoneNumber: String? = null
    private var emailAddress: String? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_detail)

        val tvName: TextView = findViewById(R.id.tv_name)
        val tvNIM: TextView = findViewById(R.id.tv_nim)
        val tvPhone: TextView = findViewById(R.id.tv_phone)
        val tvEmail: TextView = findViewById(R.id.tv_email)
        val btnPhoneCall: Button = findViewById(R.id.btn_phone_call)
        val btnSendEmail: Button = findViewById(R.id.btn_send_email)

        val name = intent.getStringExtra(KEY_NAME)
        val nim = intent.getStringExtra(KEY_NIM)
        val phone = intent.getStringExtra(KEY_PHONE)
        val email = intent.getStringExtra(KEY_EMAIL)

        tvName.text = String.format("Nama : %s", name)
        tvNIM.text = String.format("NIM : %s", nim)
        tvPhone.text = String.format("No. HP : %s", phone)
        tvEmail.text = String.format("Email : %s", email)

        phoneNumber = phone
        btnPhoneCall.setOnClickListener {
            run {
                val dialIntent = Intent(Intent.ACTION_DIAL).apply {
                    data = Uri.parse("tel:$phoneNumber")
                }
                if (intent?.resolveActivity(packageManager) != null) {
                    startActivity(dialIntent)
                }
            }
        }
    }
}

```

```

    }
}

emailAddress = email
btnSendEmail.setOnClickListener {
    run {
        val sendEmailIntent = Intent(Intent.ACTION_SENDTO).apply {
            data = Uri.parse("mailto:")
            putExtra(Intent.EXTRA_EMAIL, arrayOf(emailAddress))
            putExtra(Intent.EXTRA_SUBJECT, "Belajar Intent")
        }
        if (intent?.resolveActivity(packageManager) != null) {
            startActivity(sendEmailIntent)
        }
    }
}
}
}

```

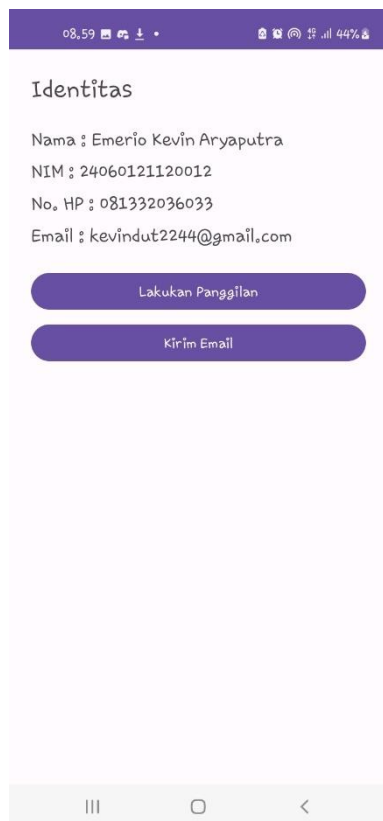
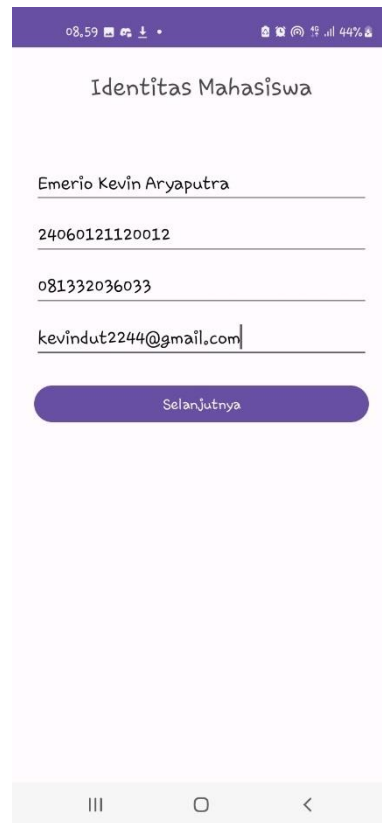
### 3.2.7. strings.xml

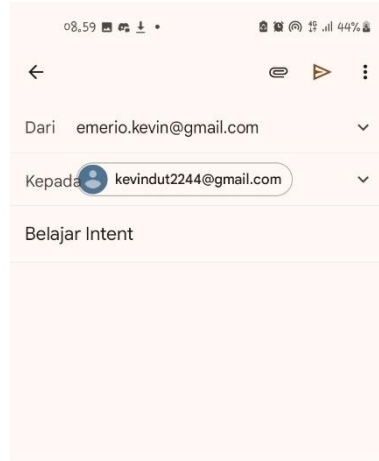
```

<resources>
    <string name="app_name">Intent Application 2</string>
    <string name="selamat_datang">Selamat datang!</string>
    <string name="perintah">Ketuk \'Mulai\' untuk berpindah halaman</string>
    <string name="mulai">Mulai</string>
    <string name="registration_activity">Registration Activity</string>
    <string name="identitas_mahasiswa">Identitas Mahasiswa</string>
    <string name="et_name_hint">Nama Lengkap</string>
    <string name="et_nim_hint">NIM</string>
    <string name="et_phone_number_hint">Nomor Handphone</string>
    <string name="btn_next">Selanjutnya</string>
    <string name="identitas">Identitas</string>
    <string name="lakukan_panggilan">Lakukan Panggilan</string>
    <string name="et_email_hint">Email</string>
    <string name="kirim_email">Kirim Email</string>
</resources>

```

### 3.3. Screenshot Program : Input dan Output





### 3.4. Penjelasan

#### 3.4.1. activity\_main.xml

File ini berfungsi untuk membuat tampilan pada Activity utama, yaitu saat aplikasi pertama kali diluncurkan. Tampilan ini akan dibuat dalam format XML.

Bagian utama dari file ini adalah LinearLayout, yaitu tampilan diatur agar elemen-elemen di dalamnya ditampilkan dalam satu baris secara vertikal ataupun horizontal. android:gravity akan menentukan bahwa elemen anak akan diposisikan di tengah induk dan android:orientation akan menentukan bahwa elemen anak akan ditampilkan secara vertikal.

Kemudian, dibentuk dua TextView dengan ukuran besar dan ukuran kecil untuk memberi sambutan dan perintah kepada pengguna. TextView ini akan memiliki isi string yang diambil dari strings.xml.

Terakhir, dibuat elemen Button dengan id btn\_start yang digunakan untuk membuat Explicit Intent untuk berpindah ke RegistrationActivity dan isi string-nya diambil dari strings.xml.



### 3.4.2. MainActivity.kt

Pertama, ditentukan terlebih dahulu package untuk file ini. Setelah itu, impor modul-modul yang akan digunakan pada program.

Kedua, dibuat class MainActivity yang merupakan turunan dari class AppCompatActivity. Setelah itu, dibuat sebuah fungsi onCreate() yang merupakan override dari onCreate() AppCompatActivity. Fungsi ini menerima parameter savedInstanceState dengan tipe Bundle, yaitu menyimpan state yang sudah ada sebelum Activity dibuat ulang. Pada awalnya fungsi dari AppCompatActivity akan dijalankan sebelum menambahkan perilaku-perilaku lainnya dengan super.onCreate. Kemudian, tampilan aplikasi diatur dengan setContentView, dengan parameter adalah layout yang ingin digunakan, yaitu layout yang berada di file activity\_main.xml.

Setelah itu, inisialisasi variabel untuk mengambil elemen UI Button, yaitu btnStart yang merupakan Button yang memiliki id btn\_start. Button ini diberi fungsi setOnClickListener untuk menetapkan listener klik ke tombol btnStart.

Di dalam fungsi setOnClickListener, dijalankan sebuah run untuk menjalankan kode secara sinkron. Di dalamnya, dibuat sebuah konstanta registrationIntent untuk membuat objek Intent baru yang akan digunakan untuk memulai Activity tujuan RegistrationActivity. Hal tersebut dapat dilakukan dengan dua parameter yang diberikan, yaitu applicationContext untuk menunjuk Activity MainActivity sebagai Activity asal dan RegistrationActivity::class.java untuk menunjuk Activity RegistrationActivity sebagai Activity tujuan. Setelah itu, dijalankan fungsi startActivity(registrationIntent) untuk memulai Activity RegistrationActivity menggunakan objek Intent yang telah dibuat.

### 3.4.3. activity\_registration.xml

File ini berfungsi untuk membuat tampilan pada saat RegistrationActivity dijalankan.

Bagian utama dari file ini adalah LinearLayout, yaitu tampilan diatur agar elemen-elemen di dalamnya ditampilkan dalam satu baris secara vertikal ataupun horizontal. android:gravity akan menentukan bahwa elemen anak akan

diposisikan di tengah induk secara horizontal dan `android:orientation` akan menentukan bahwa elemen anak akan ditampilkan secara vertikal.

Kemudian, dibentuk `TextView` sebagai petunjuk halaman registrasi dan dibuat empat `EditText` untuk menerima masukan nama, NIM, nomor telepon, dan email yang diberi id yang berbeda-beda.

Terakhir, dibuat elemen `Button` dengan id `btn_next` yang digunakan untuk membuat `Explicit Intent` untuk berpindah ke `DetailActivity` dan isi string-nya diambil dari `strings.xml`.

#### 3.4.4. `RegistrationActivity.kt`

Kelas ini berfungsi untuk menampilkan halaman registrasi setelah terjadi perpindahan `Activity` dari `MainActivity` ke `RegistrationActivity` dengan menggunakan `Intent`.

Pertama, ditentukan terlebih dahulu package untuk file ini. Setelah itu, impor modul-modul yang akan digunakan pada program.

Kedua, dibuat class `RegistrationActivity` yang merupakan turunan dari class `AppCompatActivity`. Di dalamnya, diinisialisasi variabel untuk mengakses elemen `EditText`, yaitu `etName`, `etNIM`, `etPhone`, dan `etEmail`.

Ketiga, dibuat sebuah fungsi `onCreate()` yang merupakan override dari `onCreate()` `AppCompatActivity`. Fungsi ini menerima parameter `savedInstanceState` dengan tipe `Bundle`, yaitu penyimpan state yang sudah ada sebelum `Activity` dibuat ulang. Pada awalnya fungsi dari `AppCompatActivity` akan dijalankan sebelum menambahkan perilaku-perilaku lainnya dengan `super.onCreate`. Kemudian, tampilan aplikasi diatur dengan `setContentView`, dengan parameter adalah layout yang ingin digunakan, yaitu layout yang berada di file `activity_registration.xml`.

Keempat, di dalam fungsi `onCreate()`, variabel pengakses `EditText`, yaitu `etName`, `etNIM`, `etPhone`, dan `etEmail` diinisialisasi dengan elemen yang memiliki ID yang sesuai, yaitu `et_name`, `et_nim`, `et_phone_number`, dan `et_email` menggunakan `findViewById()`. Setelah itu, diinisialisasi variabel untuk mengakses elemen `Button`, yaitu `btnNext`. `btnNext` ini merupakan elemen `Button` yang memiliki ID `btn_next`.

Kelima, dibuat click listener pada btnNext. Jika btnNext ditekan, maka akan diambil nilai etName, etNIM, etPhone, etEmail yang kemudian disimpan pada name, nim, phone, dan email. Setelah itu, dilakukan validasi untuk memastikan bahwa elemen EditText telah memiliki nilai yang diisi oleh pengguna. Jika tidak, maka field akan diberi error dan tidak bisa berpindah ke DetailActivity.

Keenam, setelah semua field terisi, maka akan dibuat sebuah variabel detailIntent. Variabel ini akan membuat sebuah Intent yang berisi dua parameter, yaitu applicationContext untuk menunjuk Activity saat ini, yaitu RegistrationActivity, serta DetailActivity::class.java untuk menunjuk Activity tujuan Intent, yaitu DetailActivity. Kemudian, detailIntent akan ditambahkan method putExtra() sebanyak empat kali untuk tiap masukan dengan dua parameter, yaitu kunci yang dimiliki di DetailActivity untuk sebuah masukan dan data yang ingin dikirimkan. Setelah itu, digunakan method startActivity(detailIntent) untuk memulai Activity DetailActivity dengan Intent detailIntent yang telah dibuat.

#### 3.4.5. activity\_detail.xml

File ini berfungsi untuk membuat tampilan pada saat DetailActivity dijalankan.

Bagian utama dari file ini adalah LinearLayout, yaitu tampilan diatur agar elemen-elemen di dalamnya ditampilkan dalam satu baris secara vertikal ataupun horizontal. android:orientation akan menentukan bahwa elemen anak akan ditampilkan secara vertikal. Elemen ini juga akan diberi padding sebanyak 24dp.

Kemudian, dibentuk empat TextView untuk menampilkan data nama, NIM, nomor telepon, dan email yang dikirimkan dari RegistrationActivity.

Terakhir, dibuat dua elemen Button dengan ID btn\_phone\_call yang digunakan untuk membuat Implicit Intent untuk melakukan panggilan serta dengan ID btn\_send\_email yang digunakan untuk membuat Implicit Intent untuk mengirim email. Isi string pada dua Button akan diambil dari strings.xml.

#### 3.4.6. DetailActivity.kt

Kelas ini berfungsi untuk menampilkan halaman registrasi setelah terjadi perpindahan Activity dari RegistrationActivity ke DetailActivity dengan menggunakan Intent.

Pertama, ditentukan terlebih dahulu package untuk file ini. Setelah itu, impor modul-modul yang akan digunakan pada program.

Kedua, dibuat class DetailActivity yang merupakan turunan dari class AppCompatActivity. Di dalamnya, diinisialisasi variabel untuk menyimpan nomor telepon dan alamat email untuk melakukan Intent. Setelah itu, dilakukan inisialisasi variabel statis untuk menyimpan kunci yang akan diakses oleh RegistrationActivity untuk mengirimkan data, yaitu KEY\_NAME, KEY\_NIM, KEY\_PHONE, dan KEY\_EMAIL dengan companion object.

Ketiga, dibuat sebuah fungsi onCreate() yang merupakan override dari onCreate() AppCompatActivity. Fungsi ini menerima parameter savedInstanceState dengan tipe Bundle, yaitu penyimpanan state yang sudah ada sebelum Activity dibuat ulang. Pada awalnya fungsi dari AppCompatActivity akan dijalankan sebelum menambahkan perilaku-perilaku lainnya dengan super.onCreate. Kemudian, tampilan aplikasi diatur dengan setContentView, dengan parameter adalah layout yang ingin digunakan, yaitu layout yang berada di file activity\_detail.xml.

Keempat, di dalam fungsi onCreate(), variabel pengakses TextView, yaitu tvName, tvNIM, tvPhone, dan tvEmail diinisialisasi dengan elemen yang memiliki ID yang sesuai, yaitu tv\_name, tv\_nim, tv\_phone, dan tv\_email menggunakan findViewById(). Setelah itu, diinisialisasi variabel untuk mengakses dua elemen Button, yaitu btnPhoneCall dan btnSendEmail. btnPhoneCall dan btnSendEmail ini merupakan elemen Button yang memiliki ID btn\_phone\_call dan btn\_send\_email

Kelima, diambil data nama, NIM, nomor telepon, dan email yang dikirimkan dari RegistrationActivity. Pertama-tama, dibuat variabel name, nim, phone, dan email. Di dalam variabel tersebut, digunakan intent.getStringExtra() dengan variabel kunci yang digunakan pada RegistrationActivity untuk mengambil nilai extra yang ditambahkan di RegistrationActivity. Setelah itu, ditampilkan data name, nim, phone, dan

email yang telah diambil dengan menetapkan isi text pada TextView yang sesuai dengan mengisi nilainya seperti biasa.

Keenam, `btnPhoneCall` diatur agar bisa menjalankan Implicit Intent untuk membuka aplikasi yang dapat menjalankan fungsi telepon. Pertama-tama, nilai `phoneNumber` yang telah diinisialisasi diisi dengan nilai `phone`. Setelah itu, dibuat `setOnClickListener` yang berisi blok run untuk menjalankan tugas di dalamnya saat tombol ditekan. Di dalamnya, dibuat sebuah `dialIntent` yang berisi Intent. Intent yang digunakan merupakan `ACTION_DIAL`, yaitu Intent yang berfungsi untuk membuka aplikasi yang mendukung aksi telepon. Pada Intent tersebut, digunakan `apply` untuk menambahkan data berupa nomor telepon yang telah diisi pada `phoneNumber`. Setelah itu, dilakukan pengecekan pada intent yang akan dipanggil. Jika terdapat minimal satu aplikasi yang dapat menangani permintaan Intent, maka akan dilakukan `startActivity` dengan parameter `dialIntent`, yang berarti Intent `dialIntent` akan dijalankan.

Ketujuh, `btnSendEmail` diatur agar bisa menjalankan Implicit Intent untuk membuka aplikasi yang dapat menjalankan fungsi mengirim email. Pertama-tama, nilai `emailAddress` yang telah diinisialisasi diisi dengan nilai email. Setelah itu, dibuat `setOnClickListener` yang berisi blok run untuk menjalankan tugas di dalamnya saat tombol ditekan. Di dalamnya, dibuat sebuah `sendEmailIntent` yang berisi Intent. Intent yang digunakan merupakan `ACTION_SENDTO`, yaitu Intent yang berfungsi untuk membuka aplikasi yang mendukung aksi mengirim email tanpa lampiran. Pada Intent tersebut, digunakan `apply` untuk menambahkan data-data. Pada data, diberikan isi “mailto:” untuk memberi tahu bahwa Intent hanya dapat ditangani oleh aplikasi yang dapat mengelola email. Setelah itu, diberikan dua `putExtra` untuk menambahkan informasi tambahan. Yang pertama, diberikan `putExtra` dengan parameter `Intent.EXTRA_EMAIL` dan `arrayOf(emailAddress)` yang menandakan bahwa ditambahkan penerima email sesuai dengan nilai `emailAddress`. Yang kedua, diberikan `putExtra` dengan parameter `Intent.EXTRA_SUBJECT` dan “Belajar Intent” yang menandakan bahwa ditambahkan subjek email dengan nilai “Belajar Intent”. Setelah itu, dilakukan pengecekan pada intent yang akan dipanggil. Jika terdapat minimal satu aplikasi yang dapat menangani permintaan Intent, maka akan dilakukan

startActivity dengan parameter sendEmailIntent, yang berarti Intent sendEmailIntent akan dijalankan.

#### 3.4.7. strings.xml

File ini berfungsi untuk menyimpan string yang akan digunakan pada activity\_main.xml, activity\_registration.xml, dan activity\_detail.xml. Nilai string akan disimpan pada tag <string></string> dan atribut namanya dapat digunakan berulang-ulang pada file activity\_main.xml dan activity\_registration.xml.

## **BAB IV**

### **PENUTUP**

#### **4.1. Kesimpulan**

Intent merupakan suatu objek perpesanan yang dapat digunakan untuk meminta sebuah aksi dari komponen aplikasi yang lain. Terdapat dua jenis Intent yang dapat digunakan, yaitu Explicit Intent dan Implicit Intent. Explicit Intent akan memulai sebuah komponen yang telah ditentukan secara langsung dan dapat mengirim data pada saat komponen tersebut dimulai. Sedangkan pada Implicit Intent, akan dimulai Intent dengan menspesifikasikan aksi yang dapat dilakukan beserta data yang telah ditentukan.

#### **4.2. Saran**

Dapat dilakukan eksplorasi pada Implicit Intent lain, seperti mengatur alarm, membuka kamera, membuka file, memulai musik, dan lain-lain.

## **DAFTAR PUSTAKA**

*BAB 2 - Intent.* Modul Praktikum 2 Pengembangan Berbasis Platform.