

LAPORAN PRAKTIKUM LAB D1
PENGEMBANGAN BERBASIS PLATFORM
“Android Intent 2”



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 implicit intent untuk membuka aplikasi email

BAB II

DASAR TEORI

2.1 Intent

Intent merupakan suatu objek pemesanan 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

Pada kasus sebelumnya kita telah membuat explicit intent untuk berpindah dari satu activity ke activity lain dengan membawa data ataupun tidak. Untuk pada kasus kali ini akan diimplementasikan pembuatan implicit intent untuk membuka aplikasi email dan dial up dengan membawa beberapa data yang akan langsung ditambahkan ke aplikasi yang akan dibuka tersebut. Untuk tampilan main atau utama pada aplikasi yang akan dibuat akan seperti pada kode XML dibawah ini:

File activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
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"
tools:context=".MainActivity"
>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_marginHorizontal="20dp"
    android:layout_gravity="center">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="D-ONE Email Sender"
        android:textSize="24sp"
        android:textStyle="bold"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="By: Resma Adi Nugroho - 24060121120021"
        android:layout_marginBottom="10dp"/>

    <TextView
        android:text="To:"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

    <EditText
        android:id="@+id/et_to"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

    <TextView
        android:text="Subject:"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
```

```

<EditText
    android:id="@+id/et_subject"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

<TextView
    android:text="Body:"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>

<EditText
    android:id="@+id/et_body"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

<Button
    android:id="@+id/btn_send"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Send Email"
    android:layout_marginTop="10dp"/>

<Button
    android:id="@+id/btn_call"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Wanna Call Someone?"
    android:layout_marginTop="10dp"/>
</LinearLayout>
</androidx.coordinatorlayout.widget.CoordinatorLayout>

```

Karena pada kali ini, activity tersebut ingin mengirimkan email maka akan terdapat tiga EditText yang masing-masingnya akan berguna untuk pertama sebagai input email tujuan, subject dan isi dari email atau body email. Kemudian pada bagian bawah akan terdapat dua button yakni untuk mengirimkan email dan untuk berpindah ke activity lain untuk melakukan dialing nomor ponsel. Selanjutnya untuk kode XML untuk tampilan activity selanjutnya yakni activity_call.xml akan memiliki kode seperti berikut:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    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"
    tools:context=".CallActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_marginHorizontal="20dp"
        android:layout_gravity="center">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="D-ONE Dialing"
            android:textSize="24sp"
            android:textStyle="bold"
            />

```

```

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="By: Resma Adi Nugroho - 24060121120021"
            android:layout_marginBottom="10dp"/>

        <TextView
            android:text="Phone Number:"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>

        <EditText
            android:id="@+id/et_phone"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"/>

        <Button
            android:id="@+id/btn_call"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Call"/>

    </LinearLayout>

</androidx.coordinatorlayout.widget.CoordinatorLayout>

```

Untuk kode XML tampilan selanjutnya yakni untuk CallActivity kita hanya memerlukan satu input yakni untuk input nomor ponsel. Kemudian bagian bawah dari input tersebut akan terdapat button yang akan membuka intent dial up dengan data sesuai input dari nomor ponsel yang telah dimasukkan.

Selanjutnya setelah membuat tampilan activity dalam kode XML, langkah selanjutnya adalah membuat fungsi untuk melakukan pemanggilan intent dan mendapatkan data dari input untuk masing-masing activity. Untuk activity yang pertama yakni MainActivity.kt dapat dilihat pada kode dibawah ini:

```

package id.newtdev.praktikum9

import android.content.Intent
import android.content.pm.PackageManager
import android.net.Uri
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import androidx.appcompat.app.AlertDialog
import id.newtdev.praktikum9.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding
    private lateinit var etTo: EditText
    private lateinit var etSubject: EditText
    private lateinit var etBody: EditText
    private lateinit var btnSend: Button
    private lateinit var btnCall: Button

```

```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = ActivityMainBinding.inflate(layoutInflater)
    setContentView(binding.root)
    // BINDING
    etTo = binding.etTo
    etSubject = binding.etSubject
    etBody = binding.etBody
    btnSend = binding.btnSend
    btnCall = binding.btnCall

    // MAIL INTENT FUNC
    fun sendEmail() {
        val email = etTo.text.toString()
        val subject = etSubject.text.toString()
        val body = etBody.text.toString()

        val gmailPackage = "com.google.android.gm"
        val isGmailInstalled = isAppInstalled(gmailPackage)
        val intent = Intent(Intent.ACTION_SEND)
        intent.putExtra(Intent.EXTRA_EMAIL, arrayOf(email))
        intent.putExtra(Intent.EXTRA_SUBJECT, subject)
        intent.putExtra(Intent.EXTRA_TEXT, body)
        // CHECK IF GMAIL INSTALLED
        if (isGmailInstalled) {
            intent.type = "text/html"
            intent.setPackage(gmailPackage)
            startActivity(intent)
        } else {
            // IF ITS NOT THEN OPEN OTHER APP
            intent.type = "message/rfc822"
            startActivity(Intent.createChooser(intent, "Pilih Aplikasi
Email"))
        }

    }

    fun showConfirm() {
        val builder: AlertDialog.Builder = AlertDialog.Builder(this)
        builder
            .setMessage("To: ${etTo.text} \nSubject: ${etSubject.text}
\nMessage: ${etBody.text}")
            .setTitle("Lets Recheck!")
            .setPositiveButton("Send") { dialog, which ->
                sendEmail()
            }
            .setNegativeButton("Cancel") { dialog, which ->
                dialog.cancel()
            }
        val dialog: AlertDialog = builder.create()
        dialog.show()
    }

    // SUBMIT AND VALIDATION
    btnSend.setOnClickListener {
        if (etTo.text.isEmpty()) {
            etTo.error = "Email tidak boleh kosong"
            etTo.requestFocus()
        } else if (etSubject.text.isEmpty()) {
            etSubject.error = "Subject tidak boleh kosong"
            etSubject.requestFocus()
        } else if (etBody.text.isEmpty()) {
            etBody.error = "Body tidak boleh kosong"
            etBody.requestFocus()
        } else {
            showConfirm()
        }
    }
}

```



```

        // OTHERS BUTTON
        btnCall.setOnClickListener {
            val intent = Intent(this, CallActivity::class.java)
            startActivity(intent)
        }

    }
    private fun isAppInstalled(packageName: String): Boolean {
        return try {
            packageManager.getApplicationInfo(packageName, 0)
            true
        } catch (e: PackageManager.NameNotFoundException) {
            false
        }
    }
}

```

Blok kode untuk binding dan inisialisasi data akan seperti pada umumnya untuk komponen atau object yang telah dibuat dalam kode XML kemudian dilakukan binding sesuai dengan id yang dimiliki oleh object atau elemen tersebut. Setelah binding dilakukan maka dapat dibuat beberapa fungsi yang akan digunakan untuk melakukan pemanggilan intent pengiriman email. Namun sebelum intent email dipanggil sebelum itu untuk masing-masing input yang ada, akan dilakukan beberapa validasi input agar input tidak boleh kosong.

```

        btnSend.setOnClickListener {
            if (etTo.text.isEmpty()) {
                etTo.error = "Email tidak boleh kosong"
                etTo.requestFocus()
            } else if (etSubject.text.isEmpty()) {
                etSubject.error = "Subject tidak boleh kosong"
                etSubject.requestFocus()
            } else if (etBody.text.isEmpty()) {
                etBody.error = "Body tidak boleh kosong"
                etBody.requestFocus()
            } else {
                showConfirm()
            }
        }
    }

```

Apabila sudah isi maka akan dipanggil fungsi showConfirm yang merupakan dialog/modal yang merupakan konfirmasi pengisian data apakah sudah sesuai untuk data yang akan dikirimkan. Apabila sudah sesuai atau benar maka dapat ditekan tombol send yang akan memanggil intent untuk membuka aplikasi Gmail secara default. Namun apabila terdapat device yang tidak memiliki aplikasi tersebut atau gagal mengidentifikasi package untuk Gmail maka aplikasi akan menampilkan opsi untuk memilih aplikasi untuk melakukan pengiriman email.

```

fun sendEmail() {
    val email = etTo.text.toString()
    val subject = etSubject.text.toString()
    val body = etBody.text.toString()

    val gmailPackage = "com.google.android.gm"
    val isGmailInstalled = isAppInstalled(gmailPackage)
    val intent = Intent(Intent.ACTION_SEND)
    intent.putExtra(Intent.EXTRA_EMAIL, arrayOf(email))
    intent.putExtra(Intent.EXTRA_SUBJECT, subject)
    intent.putExtra(Intent.EXTRA_TEXT, body)
    // CHECK IF GMAIL INSTALLED
    if (isGmailInstalled) {
        intent.type = "text/html"
        intent.setPackage(gmailPackage)
        startActivity(intent)
    } else {
        // IF ITS NOT THEN OPEN OTHER APP
        intent.type = "message/rfc822"
        startActivity(Intent.createChooser(intent, "Pilih Aplikasi
Email"))
    }

}

fun showConfirm() {
    val builder: AlertDialog.Builder = AlertDialog.Builder(this)
    builder
        .setMessage("To: ${etTo.text} \nSubject: ${etSubject.text}
\nMessage: ${etBody.text}")
        .setTitle("Lets Recheck!")
        .setPositiveButton("Send") { dialog, which ->
            sendEmail()
        }
        .setNegativeButton("Cancel") { dialog, which ->
            dialog.cancel()
        }
    val dialog: AlertDialog = builder.create()
    dialog.show()
}

```

Setelah memilih aplikasi atau aplikasi Gmail terbuka maka data yang telah dimasukkan sebelumnya akan langsung terisi pada bagian masing-masing pada komponen email. Karena kita telah mengirimkan data hasil input dengan putExtra untuk masing-masing key yakni, EXTRA_EMAIL untuk tujuan email yang akan dikirim, EXTRA_SUBJECT untuk subject dari email kemudian EXTRA_TEXT untuk body atau message yang akan dikirimkan.

Selanjutnya untuk pada bagian CallActivity.kt yakni untuk melakukan dial up atau memanggil nomor sesuai dengan inputan yang telah diisi, maka kode yang dapat digunakan dapat dilihat dibawah ini:

```

package id.newtdev.praktikum9

import android.content.Intent
import android.net.Uri
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import id.newtdev.praktikum9.databinding.ActivityCallBinding

class CallActivity : AppCompatActivity() {

    private lateinit var binding: ActivityCallBinding
    private lateinit var etPhone: EditText
    private lateinit var btnCall: Button

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityCallBinding.inflate(layoutInflater)
        setContentView(binding.root)
        // BINDING
        etPhone = binding.etPhone
        btnCall = binding.btnCall

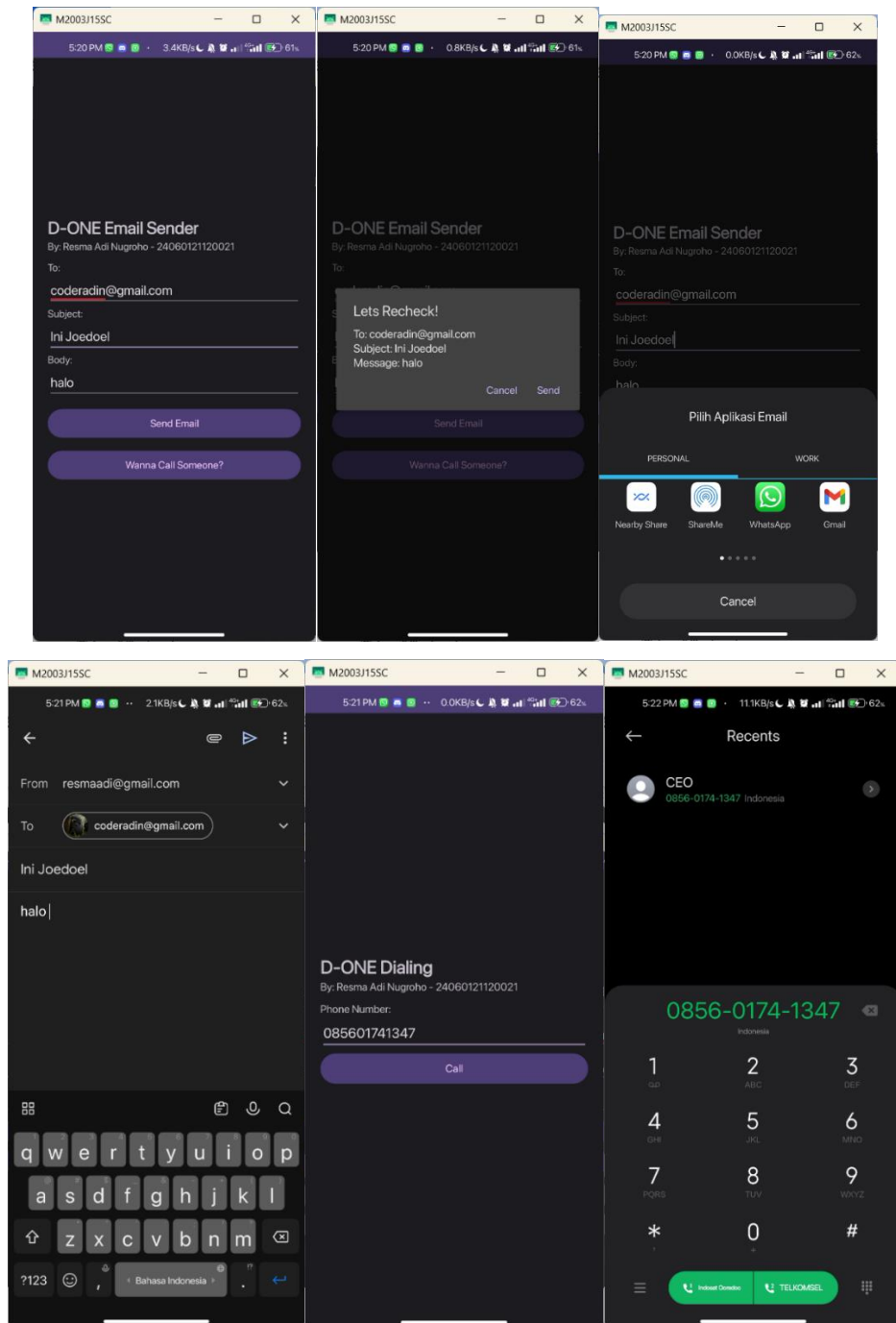
        // CALL FUNC
        fun call() {
            val phone = etPhone.text.toString()
            if(phone.isEmpty()){
                etPhone.error = "Nomor Telepon Tidak Boleh Kosong"
                return
            }
            val intent = Intent(Intent.ACTION_DIAL,
Uri.parse("tel:$phone"))
            startActivity(intent)
        }

        btnCall.setOnClickListener { call() }
    }
}

```

Kode tersebut akan memanggil intent untuk DIAL yang akan langsung membuka default aplikasi telfon yang ada pada sebuah device, dan dari input yang telah dimasukkan secara otomatis akan terisi sehingga user hanya perlu menekan tombol telfon saja.

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.