

LAPORAN PRAKTIKUM
PENGEMBANGAN BERBASIS PLATFORM



**PENGEMBANGANBERBASIS
PLATFORM**

Lab D2

DISUSUN OLEH :

NAMA : Muhamad Ridwan Ash'shidqi

NIM 24060121130075

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO

2023

KATA PENGANTAR

Segala puji dan syukur kami panjatkan atas kesempatan dan berkah yang telah Allah SWT anugerahkan kepada kami dalam perjalanan praktikum ini. Laporan praktikum "Pengembangan Berbasis Platform ke-7" ini kami susun sebagai bagian dari eksplorasi dan pemahaman kami tentang dasar- dasar pembelajaran mesin yang telah diajarkan selama praktikum di laboratorium D2.

Adapun maksud dan tujuan adanya laporan praktikum ini adalah dapat membuat aplikasi sederhana dengna menggunakan intent.

Dalam pembuatan laporan ini, saya mengucapkan terima kasih kepada Kak Danang Wisnu Prayoga dan Kak Agung Ramadhani yang telah membantu kami dalam pemahaman materi dari praktikum yang elah dilaksanakan.

Dengan ini, saya menyadari bahwa laporan praktikum ini masih jauh dari kata sempurna. Untuk itu, saya dengan sangat terbuka menerima kritik dan saran dari para pembaca. Semoga laporan praktikum ini bermanfaat untuk para pembaca dan saya khususnya dalam usaha meningkatkan pemahaman mengenai aplikasi Android Studio dalam proses pembuatan aplikasi berbasis Android.

Semarang, 14 November
2023

Muhamad Ridwan Ash'shidqi

BAB I

PENDAHULUAN

1.1 Tujuan

Setelah mengikuti Setelah mengikuti praktikum ini mahasiswa diharapkan dapat memahami cara untuk:

1.1.1 Mahasiswa mampu memahami Intent.

1.2 Pertanyaan

1.2.1 Mahasiswa mampu membuat aplikasi sederhana dengan menerapkan Intent.

BAB II

DASAR TEORI

2.1 Intent

Intent adalah sebuah kelas dalam programming Android yang berfungsi untuk perpindahan halaman. Intent juga merupakan suatu objek yang terdapat dalam suatu activity dimana objek tersebut dapat komunikasi dengan activity yang lain, baik activity pada fungsi internal android misal seperti memanggil activity dalam satu package atau beda package yang masih berada dalam satu project.

- **Explicit Intent**

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. Intent ini berfungsi untuk mengaktifkan komponen-komponen dalam satu aplikasi yang sama. Misalnya seperti: Berpindah Activity.

```
val downloadIntent = Intent(this, DownloadService::class.java).apply {  
    data = Uri.parse(fileUrl)  
}  
startService(downloadIntent)
```

- **Implicit Intent**

sebuah Intent yang tidak menyebutkan komponen yang ingin dituju. mendeklarasikan sebuah tugas general yang dapat dilakukan oleh komponen pada aplikasi lain. Contohnya, Ketika mengirimkan email, dapat menggunakan Implicit Intent untuk meminta aplikasi lain yang memiliki fungsi untuk mengirimkan email agar dapat melaksanakan tugas tersebut.

```
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.
    }

```

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()`.

BAB III

PEMBAHASAN

3.1 Menjawab Pertanyaan 1.2.1

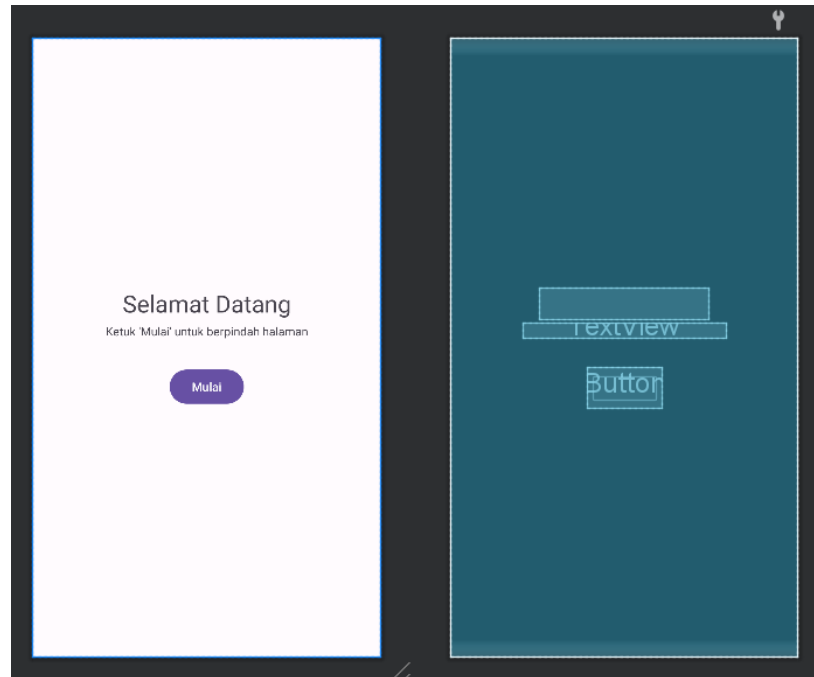
1.2.1.1 Activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:gravity="center"
9      android:orientation="vertical"
10     tools:context=".MainActivity">
11     <TextView
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:text="Selamat Datang"
15         android:textSize="28sp" />
16     <TextView
17         android:layout_width="wrap_content"
18         android:layout_height="wrap_content"
19         android:layout_marginTop="4dp"
20         android:text="Ketuk 'Mulai' untuk berpindah halaman" />
21     <Button
22         android:id="@+id/btn_start"
23         android:layout_width="wrap_content"
24         android:layout_height="wrap_content"
25         android:layout_marginTop="32dp"
26         android:text="Mulai" />
27 </LinearLayout>
```

File .xml dengan nama activity_main digunakan mengatur tampilan aplikasi yang akan dibuat. Berikut penjelasan code pada file tersebut :

- a. xmlns:android digunakan untuk mendefinisikan namespace untuk elemen-elemen Android.
- b. xmlns:app digunakan untuk mendefinisikan namespace tambahan untuk elemen-elemen dari library atau dependensi tertentu.
- c. xmlns:tools mendefinisikan namespace untuk atribut-atribut yang hanya digunakan selama waktu kompilasi dan build dan tidak mempengaruhi runtime aplikasi.
- d. android:gravity dengan nilai atribut center akan menentukan posisi konten di dalam LinearLayout agar berada di Tengah.
- e. tools:context dengan nilai atribut MainActivity akan menunjukkan bahwa layout terkait dengan kelas MainActivity.

Pada tag TextView berisi text yang ingin di isikan pada halaman dan pada text tersebut dapat diatur style sesuai yang di inginkan, seperti mengatur lebar dengan menggunakan layout_width dan mengatur tinggi menggunakan layout_height. Kemudian terdapat tag button untuk menampilkan tombol yang akan dibuat



Gambar 1. Tampilan activity_main.xml

1.2.1.2 MainActivity.kt

```
class MainActivity : AppCompatActivity(), View.OnClickListener {

    private lateinit var btnStart: Button

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

        btnStart = findViewById(R.id.btn_start)
        btnStart.setOnClickListener(this)
    }

    override fun onClick(view: View?) {
        if (view != null) {
            if (view.id == R.id.btn_start) {
                val registrationIntent = Intent(
                    packageContext, this,
                    RegistrationActivity::class.java
                )
                startActivity(registrationIntent)
            }
        }
    }
}
```

Terdapat class bernama MainActivity dengan ekstensi kelas AppCompatActivity, View.OnClickListener yang berarti bahwa kelas memiliki metode onClick yang harus diimplementasikan.

Function private lateinit var btnStart digunakan untuk button agar penginisialisasinya saat di click atau di akhir. Selanjutnya pada Fungsi onCreate terdapat super.onCreate(savedInstanceState) yang digunakan untuk memanggil implementasi metode 'onCreate' dari kelas induk. Function setContentView digunakan untuk menampilkan XML activity_main ke halaman konten. Selanjutnya pada variabel btnStart berisi elemen dengan id btn_start pada file XML dan kemudian Menetapkan fungsi OnClickListener pada variabel tersebut.

Setelah itu function `onClick` , Jika view nya tidak null , maka akan diimplementasikan dari interface `View.OnClickListener.`, dan jika yang di klik adalah elemen dengan id `btn_start`, maka akan membuat objek intent dari activity saat ini ke activity `RegistrationActivity` (aktivitas baru yang akan dibuat sebagai transisi halaman saat tombol `btn_start` di klik).

1.2.1.3 build.gradle

```
buildFeatures { this: ApplicationBuildFeatures
    viewBinding = true
}
}
```

Untuk menerapkan intent explicit dibutuhkan binding maka `viewBinding` perlu diset True pada `build.gradle`

1.2.1.4 activity_registration.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="25dp"
    tools:context=".RegistrationActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="136dp"
        android:text="Registration Activity"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/et_nama"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="36dp"
        android:hint="Nama"
        android:textSize="15sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

    <EditText
        android:id="@+id/et_nim"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:hint="NIM"
        android:textSize="15sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/et_nama" />

    <EditText
        android:id="@+id/et_jurusan"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:hint="Jurusan"
        android:textSize="15sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/et_nim" />


```

```

<Button
    android:id="@+id/btn_implicit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="44dp"
    android:text="Implicit"
    app:layout_constraintStart_toStartOf="@+id/et_jurusan"
    app:layout_constraintTop_toBottomOf="@+id/et_jurusan" />

<Button
    android:id="@+id/btn_explicit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="44dp"
    android:text="Explicit with data"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.973"
    app:layout_constraintStart_toEndOf="@+id/btn_implicit"
    app:layout_constraintTop_toBottomOf="@+id/et_jurusan" />

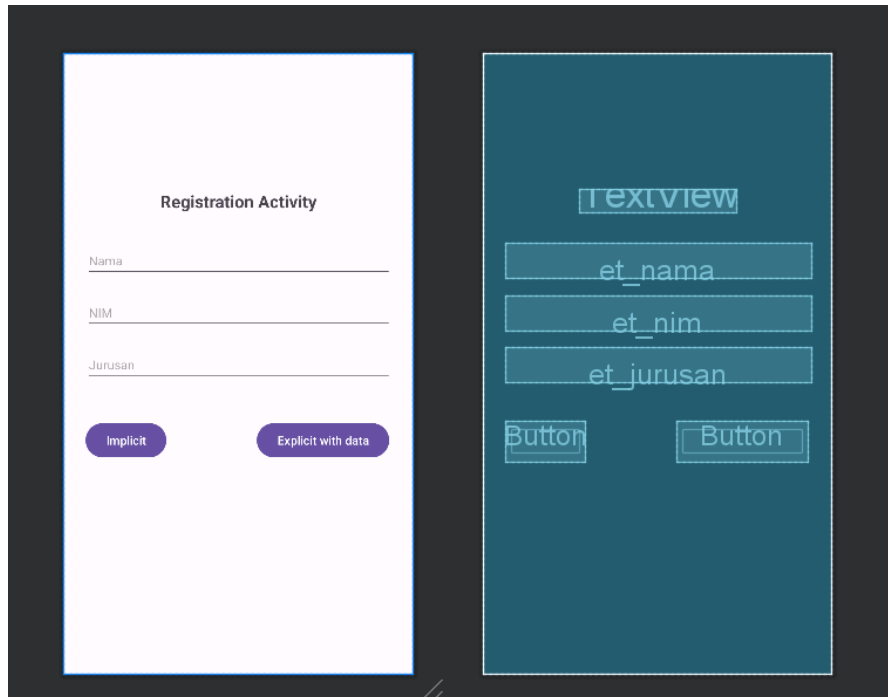
</androidx.constraintlayout.widget.ConstraintLayout>

```

File `activity_registration.xml` berfungsi untuk mengatur tampilan registration. Pada file ini dibuat komponen bernama Constraint Layout untuk membuat layout dari tampilan aplikasi. Berikut penjelasan code pada file tersebut :

- `xmlns:android` digunakan untuk mendefinisikan namespace untuk elemen-elemen Android.
- `xmlns:app` digunakan untuk mendefinisikan namespace tambahan untuk elemen-elemen dari library atau dependensi tertentu.
- `xmlns:tools` mendefinisikan namespace untuk atribut-atribut yang hanya digunakan selama waktu kompilasi dan build dan tidak mempengaruhi runtime aplikasi.
- `android:gravity` dengan nilai atribut `center` akan menentukan posisi konten di dalam `LinearLayout` agar berada di Tengah.
- `tools:context` dengan nilai atribut `MainActivity` akan menunjukkan bahwa layout terkait dengan kelas `MainActivity`

Seperti pada penjelasan sebelumnya, tag `TextView` digunakan untuk menampilkan text yang akan dibuat dan dapat diatur style sesuai yang di inginkan. Kemudian, terdapat dua tag `Button` untuk menampilkan tombol pada aplikasi. Tombol bertuliskan “Implicit” yang memiliki `id` `btn_implicit` dan tombol “Explicit with data” yang memiliki `id` `btn_explicit`.



Gambar 2. Tampilan `activity_registration.xml`

1.1.1.1 RegistrationActivity.kt

```
class RegistrationActivity : AppCompatActivity(), View.OnClickListener {
    //Deklarasi binding
    private lateinit var binding: ActivityRegistrationBinding

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

        //melakukan view binding
        binding = ActivityRegistrationBinding.inflate(layoutInflater)
        setContentView(binding.root)

        //menambah onclicklistener
        binding.btnImplicit.setOnClickListener(this)
        binding.btnExplicit.setOnClickListener(this)
    }

    override fun onClick(v: View) {
        when(v.id) {
            R.id.btn_implicit -> {
                //implicit intent
                val message = "How's your day"
                val intent = Intent()
                intent.action = Intent.ACTION_SEND
                intent.putExtra(Intent.EXTRA_TEXT, message)
                intent.type = "text/plain"
                startActivity(intent)
            }

            R.id.btn_explicit -> {
                //explicit intent
                val inputNama: String = binding.etNama.text.toString().trim()
                val inputNIM: String = binding.etNim.text.toString().trim()
                val inputJurusan: String = binding.etJurusan.text.toString().trim()
                //Validasi input kosong
                var isEmptyField = false
                if (inputNama.isEmpty()) {
                    isEmptyField = true
                    binding.etNama.error = "Field nama harus diisi"
                }
                if (inputNIM.isEmpty()) {
                    isEmptyField = true
                    binding.etNim.error = "Field NIM harus diisi"
                }
                if (inputJurusan.isEmpty()) {
                    isEmptyField = true
                    binding.etJurusan.error = "Field jurusan harus diisi"
                }
            }
        }
    }
}
```

```

if (!isEmptyField){
    val bundle = Bundle()
    bundle.putString("nama", inputNama)
    bundle.putString("nim", inputNIM)
    bundle.putString("jurusan", inputJurusan)

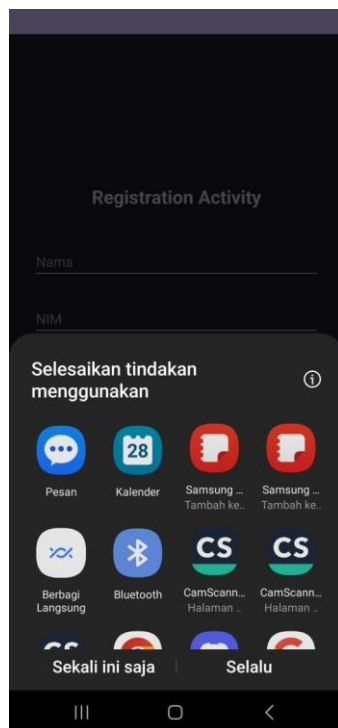
    //mengirim intent ke halaman home
    val intent = Intent( packageContext: this@RegistrationActivity, HomeActivity::class.java)
    intent.putExtras(bundle)
    startActivity(intent)
}
}
}
}

```

Terdapat class `RegistrationActivity` dengan ekstensi kelas `ComponentActivity`, `View.OnClickListener` (menunjukkan bahwa kelas memiliki metode `onClick` yang harus diimplementasikan). Function `private lateinit var binding: ActivityRegistrationBinding` digunakan untuk mendeklarasikan binding pada file `kt` ini. Variabel `binding` akan menginflate(konversi) layout XML ke dalam objek `binding`. Selanjutnya Function `onCreate` terdapat `super.onCreate(savedInstanceState)` yang akan memanggil implementasi metode 'onCreate' dari kelas induk, biasanya diperlukan untuk memastikan bahwa lifecycle aktivitas diatur dengan benar. Selanjutnya function `setContentView` akan menampilkan XML `activity_registration` ke halaman konten. Terdapat variable `Binding` ini akan menjadi sebuah variable `root` untuk button nantinya pada file `kt`. Menetapkan fungsi `OnClickListener` dan diterapkan binding pada variabel `btnImplicit` dan `btnExplicit`.

Function `onClick` yang akan diimplementasikan dari interface `View.OnClickListener`. pertama dilakukan pengecekan saat `v.id`, jika `R.idnya` `btnImplicit` (klik button implicit), maka membuat variable `message` yang di isikan “How’s your day” kemudian membuat objek `intent` berisi `intent()`.

Pada `intent.action = Intent.ACTION_SEND` digunakan untuk Menetapkan tindakan `Intent` ke `ACTION_SEND`, yang digunakan untuk mengirim data seperti teks. Selanjutnya `intent.putExtra(Intent.EXTRA_TEXT, message)` untuk Menambahkan data tambahan ke `Intent`. Dalam hal ini, kita menambahkan teks pesan ke `Intent` menggunakan kunci `EXTRA_TEXT`. Function `intent.type = "text/plain"`: Menetapkan tipe data `Intent` menjadi teks biasa (`text/plain`), memberi tahu sistem bahwa kita mengirim teks. Yang terakhir pada function `startActivity(intent)`: Memulai aktivitas yang sesuai dengan `Intent`. Dalam konteks ini, akan memulai aktivitas yang dapat menangani tindakan `ACTION_SEND`, seperti aplikasi pesan atau media sosial, dan memungkinkan pengguna untuk membagikan pesan yang sudah ditentukan. Berikut adalah gambar ketika mengklik implicit



Kondisi kedua, saat v.id, jika R.idnya btnExplicit(klik button explicit), melakukan pengambilan input yang dimasukkan user menggunakan **binding.etNama.text.toString().trim(),binding.etNim.text.toString().trim(),binding.etJurusan.text.toString().trim()**. Selanjutnya, melakukan validasi input pada kondisi input yang kosong. Jika ada field yang kosong, variabel isEmptyField menjadi true, dan pesan kesalahan akan muncul.

Function if (!isEmptyField) melakukan pengecekan apakah tidak ada field yang kosong. Jika tidak ada field yang kosong, maka akan membuat bundle dan diisi dengan data yang ingin dikirim ke aktivitas tujuan. Data ini disimpan dalam bentuk pasangan kunci-nilai, di mana kunci adalah nama data dan nilainya adalah nilai data tersebut.

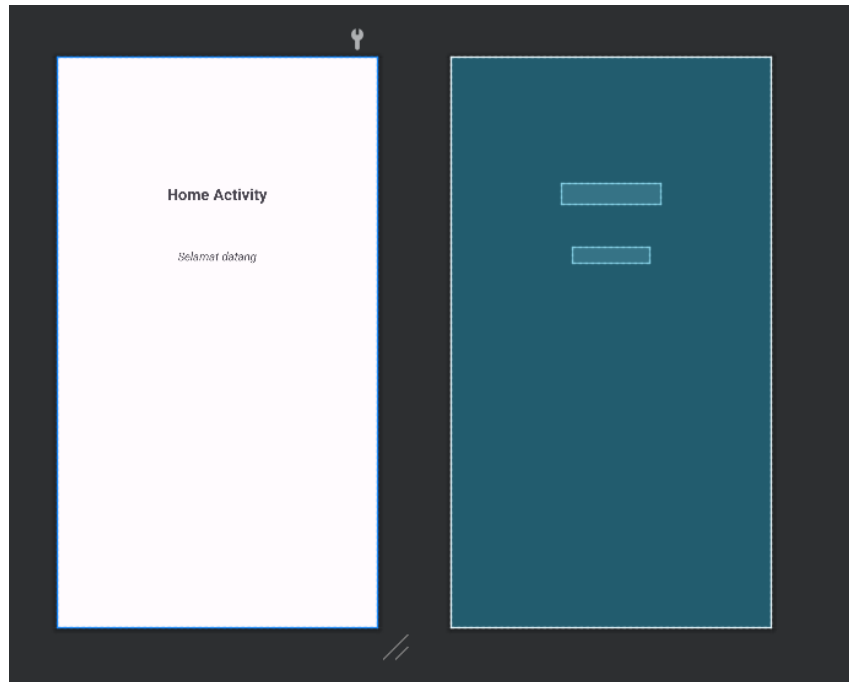
Dibuat Intent eksplisit dengan merinci aktivitas tujuan, yaitu HomeActivity::class.java. Kemudian, data dimasukkan ke dalam intent menggunakan intent.putExtras(bundle), di mana bundle adalah objek yang berisi data yang ingin dikirim. Setelah itu, intent tersebut dieksekusi dengan menggunakan startActivity(intent), sehingga pengguna akan diarahkan ke HomeActivity

1.1.1.2 activity_home.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout 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:padding="25dp"
    tools:context=".HomeActivity">
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="136dp"
        android:text="Home Activity"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tv_pesan"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="56dp"
        android:text="Selamat datang"
        android:textSize="15sp"
        android:textStyle="italic"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView2" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Seperti pada penjelasan sebelumnya, tag `TextView` digunakan untuk menampilkan text yang akan dibuat dan dapat diatur style sesuai yang di inginkan. Pada coding tersebut terdapat 2 text yang akan ditambahkan yaitu “Home Activity” dan “Selamat Datang”



Gambar 5. Tampilan Home

1.1.1.3 HomeActivity.kt

```
package com.example.intentapplication

import ...

class HomeActivity : AppCompatActivity() {

    private lateinit var binding: ActivityHomeBinding

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

        //melakukan view binding
        binding = ActivityHomeBinding.inflate(layoutInflater)
        setContentView(binding.root)

        //mengambil data dari bundle
        val bundle = intent.extras
        if (bundle != null){
            val nama = bundle.getString( key: "nama")
            val nim = bundle.getString( key: "nim")
            val jurusan = bundle.getString( key: "jurusan")
            binding.tvPesan.setText("Halo \n" + nama + "\n" + "NIM : " + nim + "\n" + "Jurusan : " + "jurusan" + "\n" + "By one dek")
        }
    }
}
```

Function `private lateinit var binding: ActivityHomeBinding` digunakan untuk mendeklarasikan `binding` pada file `kt`. Variabel `binding` nantinya akan menginflate(konversi) layout XML ke dalam objek `binding`.

Pada Fungsi `onCreate` terdapat function `super.onCreate(savedInstanceState)` yang akan memanggil implementasi metode 'onCreate' dari kelas induk. Kemudian function `setContentView` akan menampilkan XML `activity_home` ke halaman konten.

`Binding` ini akan menjadi sebuah variable `root`

Membuat variable `bundle = intent.extras` yang mengambil data dari intent yang memulai aktivitas. Kemudian Data dikirimkan dari aktivitas sebelumnya menggunakan intent yaitu dari registration.

Melakukan pengecekan bundle kosong atau tidak pada `if (bundle != null)`. Jika tidak, maka lanjut mengambil data bundle menggunakan `bundle.getString("nama")`, `bundle.getString("nim")`, dan `bundle.getString("jurusan")` sesuai dengan keynya juga. Selanjutnya `binding.tvPesan.setText("Halo \n" + nama + "\n" + "NIM : " + nim + "\n" + "Jurusan :" + jurusan + "\n" + "By one dek")` digunakan untuk mengatur teks pada TextView (`tvPesan`) menggunakan data yang diambil dari bundle.



BAB IV

PENUTUP

4.1 Kesimpulan

Setelah melakukan praktikum, dapat disimpulkan bahwa pembuatan aplikasi dengan intent dibagi menjadi implicit dan explicit yang mana jika implicit akan mengarahkan ke halaman registration activity sedangkan explicit dapat mengarahkan ke aplikasi lain.

4.2 Saran

Saat memberikan penjelasan jangan cepat cepat karna susah untuk diikuti

DAFTAR PUSTAKA

Modul Praktikum 8 Pemrograman Berbasis Platform Universitas Diponegoro, n.d. *Bab 2*
- *Intent*

