

PRAKTIKUM APLIKASI PROFIL DIRI

Pada Praktikum ini akan dibahas kembali cara penggunaan dari komponen-komponen antar muka, Activity, dan intent lewat pembuatan aplikasi profil diri. Aplikasi akan melakukan perintah untuk pindah halaman ke halaman profil diri dengan menampilkan data dari halaman registrasi, ketika pengguna selesai melakukan pengisian data. **Praktikum ini menggunakan bahasa Kotlin.**

◆ Layout Aplikasi Profil Diri

Aplikasi profil diri yang akan dibuat memiliki dua halaman, yaitu registrasi dan profil diri. Pada halaman-halaman tersebut akan digunakan beberapa komponen antar muka seperti LinearLayout, RelativeLayout, ImageView, TextView, EditText, Button dan komponen antar muka lain yang dibutuhkan.

View Group dan View

ViewGroup atau disebut juga parent, adalah komponen antar muka yang digunakan untuk meletakkan view (child) atau view group lain dalam membentuk layout dari sebuah halaman. Sebuah layout biasanya memiliki minimal satu ViewGroup. ViewGroup merupakan komponen antar muka berbasis layout, maka dapat diketahui dengan mudah bahwa komponen antar muka yang disebut termasuk ViewGroup saat ini di antaranya adalah LinearLayout, RelativeLayout, ConstraintLayout, CoordinatorLayout, dan FrameLayout. Sedangkan, view adalah komponen antar muka berbasis widgets seperti, TextView, ImageView, EditText, Button, dan yang lain.

Baik ViewGroup ataupun View merupakan komponen antar muka yang dapat diperlakukan sama dalam hal diberikan pengaturan properti, status visibility, dan listener yang akan dibahas lebih lanjut pada pembahasan selanjutnya.

◆ Membuat Layout Halaman Registrasi

Sebelum membuat layout halaman registrasi aplikasi profil diri, silahkan lakukan pembuatan project baru dengan nama aplikasi "My Profile App". Jangan lupa untuk memilih Kotlin pada pilihan Language dan checklist Use AndroidX artifacts.

Halaman registrasi akan memiliki layout berisi form isian dengan posisi form isian yang disusun mendatar ke bawah. ViewGroup yang paling mudah digunakan untuk membuat layout dengan karakter posisi mendatar ke bawah (vertikal) atau ke samping (horizontal) adalah LinearLayout. Untuk form isian yang berupa input text akan digunakan EditText. Kemudian, Spinner akan digunakan untuk form input berupa pilihan. Dalam hal ini, pilihan jenis kelamin.



Gambar Design Antar Muka Halaman Registrasi

Berikut ini adalah kode file layout (main_activity.xml) untuk membuat layout seperti pada gambar di atas dan akan dibahas lebih detail pada keterangan setelahnya.

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:orientation="vertical"
    tools:context=".MainActivity"
    android:background="#FFFFFF">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Registrasi"
        android:textSize="19sp"
        android:layout_margin="16dp"
        android:textColor="#000000"
        android:fontFamily="casual"
        android:layout_gravity="center_horizontal"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="16dp"
        android:orientation="vertical">

        <EditText
            android:id="@+id/edtName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Nama"
            android:textColorHint="#CCCCCC"
            android:textColor="#000000"
            android:inputType="text"
            android:imeOptions="actionNext"/>

        <Spinner
            android:layout_marginTop="16dp"
            android:id="@+id/spinnerGender"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColor="#000000"/>

        <EditText
            android:id="@+id/edtEmail"
            android:layout_marginTop="16dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Email"
            android:textColorHint="#CCCCCC"
            android:textColor="#000000"
            android:inputType="textEmailAddress"
            android:imeOptions="actionNext"/>

        <EditText
            android:id="@+id/edtTelp"
            android:layout_marginTop="16dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Telp."/>
```

```

        android:textColorHint="#CCCCCC"
        android:textColor="#000000"
        android:inputType="number"
        android:imeOptions="actionNext"/>

<EditText
    android:id="@+id/edtAddress"
    android:layout_marginTop="16dp"
    android:layout_width="match_parent"
    android:layout_height="120dp"
    android:hint="Alamat"
    android:textColorHint="#CCCCCC"
    android:textColor="#000000"
    android:inputType="textMultiLine"
    android:gravity="top"
    android:imeOptions="actionDone"/>

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

</LinearLayout>

</LinearLayout>

```

Pembahasan komponen antar muka pada **activity_main.xml** 1. LinearLayout.

LinearLayout adalah komponen ViewGroup yang biasanya digunakan untuk membuat layout dengan susunan view mendatar ke bawah (vertical) atau ke samping (horizontal). Hal itu, dikarenakan LinearLayout memiliki property **android:orientation="vertical"** yang dapat diisi value **vertical** dan **horizontal**. Property ini harus selalu digunakan.

- TextView

TextView adalah komponen View yang digunakan untuk menampilkan informasi berupa text. Berikut adalah kegunaan dari beberapa property yang umumnya digunakan untuk TextView:

Property	Fungsi
android:text=""	Mengatur tulisan yang tampil pada TextView
android:textSize=""	Mengatur ukuran font TextView dengan satuan "sp"

<code>android:textColor=""</code>	Mengatur warna font TextView. Diisi dengan kode warna hexadesimal yang diawali dengan tanda #, misal "#000000"
<code>android:fontFamily=""</code>	Mengatur jenis font TextView

- Spinner

Spinner adalah komponen View yang digunakan untuk kebutuhan menerima input dalam bentuk pilihan. Misal, pilihan kelamin (Laki-laki atau Perempuan). Data pilihan yang ditampilkan akan diatur pada file .kt yang menggunakan layout activity_main.xml, dalam kasus ini adalah MainActivity.kt

- EditText

EditText adalah komponen View yang digunakan untuk kebutuhan menerima input dalam bentuk text. Berikut adalah beberapa property yang banyak digunakan pada EditText:

Property	Fungsi
<code>android:hint=""</code>	Mengatur tooltip yang muncul pada EditText sebelum disentuh
<code>android:textColorHint=""</code>	Mengatur warna tooltip EditText. Diisi dengan kode warna hexadesimal yang diawali dengan tanda #, misal "#CCCCCC"
<code>android:textColor=""</code>	Mengatur warna font pada EditText. Diisi dengan kode warna hexadesimal yang diawali dengan tanda #, misal "#000000"
<code>android:inputType=""</code>	Mengatur tipe soft input keyboard ketika EditText disentuh. Ada beberapa, tipe input di antaranya, text, number, textPassword, dll.
<code>android:imeOptions=""</code>	Mengatur tipe OK button pada soft input keyboard yang muncul ketika EditText disentuh.

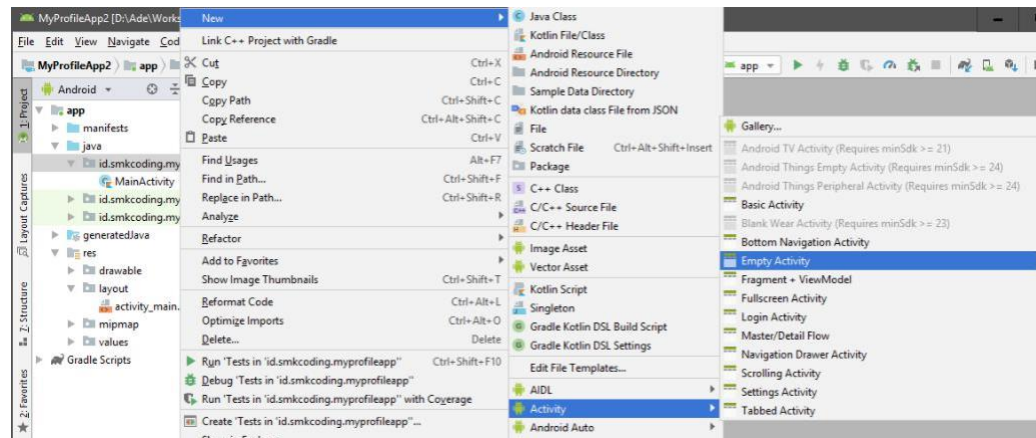
- Button

Button adalah komponen View berupa button. Button dapat diberi label dengan menggunakan property `android:text=""`

◆ Membuat Layout Halaman Profil Diri

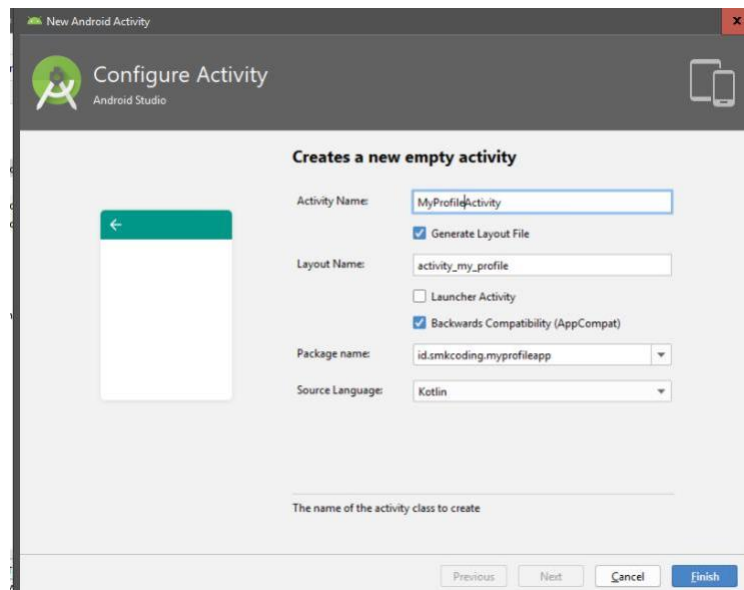
Halaman profil diri adalah halaman (activity) baru. Setiap halaman umumnya berpasangan antara file layout .xml dengan file class .kt. Android Studio sudah memberikan kemudahan untuk membuat activity baru, yaitu dengan melakukan langkah berikut:

1. Klik kanan pada direktori **app\java\<app package>**, pilih **New**, kemudian pilih **Activity**, lalu pilih **Empty Activity**



Gambar Membuat Activity Baru

2. Beri nama pada Activity Baru (MyProfileActivity)



Gambar Memberi Nama Activity Baru

activity_my_profile.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    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"
    tools:context=".MyProfileActivity"
    android:background="#FFFFFF">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="16dp"
        android:orientation="vertical">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <ImageView
                android:id="@+id/imgProfile"
                android:layout_width="100dp"
                android:layout_height="100dp"
                android:background="#EEEEEE"/>

            <LinearLayout
                android:layout_marginLeft="16dp"
                android:layout_toRightOf="@id/imgProfile"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical">

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:orientation="horizontal">

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="Name: "
                        android:textSize="15sp"
                        android:textColor="#000000"
                        android:fontFamily="sans-serif"/>

                    <TextView
                        android:id="@+id/txtName"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text=""
                        android:textSize="15sp"
                        android:textColor="#000000"
                        android:fontFamily="sans-serif"/>

                </LinearLayout>

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal">


```

```

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Gender: "
            android:textSize="15sp"
            android:textColor="#000000"
            android:fontFamily="sans-serif"/>

        <TextView
            android:id="@+id/txtGender"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text=""
            android:textSize="15sp"
            android:textColor="#000000"
            android:fontFamily="sans-serif"/>

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Email: "
            android:textSize="15sp"
            android:textColor="#000000"
            android:fontFamily="sans-serif"/>

        <TextView
            android:id="@+id/txtEmail"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text=""
            android:textSize="15sp"
            android:textColor="#000000"
            android:fontFamily="sans-serif"/>

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Telp.: "
            android:textSize="15sp"
            android:textColor="#000000"
            android:fontFamily="sans-serif"/>

        <TextView
            android:id="@+id/txtTelp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text=""
            android:textSize="15sp"

```



```

        android:textColor="#000000"
        android:fontFamily="sans-serif"/>

    </LinearLayout>

</LinearLayout>

</RelativeLayout>

<TextView
    android:layout_marginTop="16dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Address:"
    android:textSize="15sp"
    android:textColor="#000000"
    android:fontFamily="sans-serif"/>

    <TextView
        android:layout_marginTop="16dp"
        android:id="@+id/txtAddress"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:textSize="15sp"
        android:textColor="#000000"
        android:fontFamily="sans-serif"/>

</LinearLayout>

</ScrollView>

```

Pembahasan komponen antar muka pada **activity_my_profile.xml**

1. ScrollView

ScrollView adalah ViewGroup yang digunakan untuk halaman dengan kebutuhan layar device dapat digulirkan. Biasanya digunakan untuk halaman dengan komponen view tersusun mendatar ke bawah yang cukup panjang.

2. RelativeLayout

RelativeLayout adalah ViewGroup yang digunakan untuk halaman dengan tata letak komponen view tidak hanya mendatar ke bawah atau ke samping. Biasanya digunakan untuk tata letak yang cukup kompleks karena jika menggunakan RelativeLayout, komponen view di dalamnya dapat dibuat relatif terhadap komponen view tertentu. Misal, dengan menggunakan property `android:layout_toRightOf="@id/<id_view>"`. Berikut adalah beberapa property

yang umumnya digunakan pada komponen view yang ada di dalam ViewGroup RelativeLayout:

Property	Fungsi
<code>android:layout_toRightOf="@id/<id_view>"</code>	Mengatur posisi view ada di sebelah kanan view tertentu
<code>android:layout_toLeftOf="@id/<id_view>"</code>	Mengatur posisi view ada di sebelah kiri view tertentu
<code>android:layout_above="@id/<id_view>"</code>	Mengatur posisi view ada di atas view tertentu
<code>android:layout_below="@id/<id_view>"</code>	Mengatur posisi view ada di bawah view tertentu
<code>android:layout_alignParentRight="true"</code>	Mengatur posisi view berada di pojok kanan
<code>android:layout_alignParentLeft="true"</code>	Mengatur posisi view berada di pojok kiri
<code>android:layout_alignParentBottom="true"</code>	Mengatur posisi view berada di pojok bawah
<code>android:layout_alignParentTop="true"</code>	Mengatur posisi view berada di pojok atas
<code>android:layout_centerInParent="true"</code>	Mengatur posisi view berada di tengah-tengah secara horizontal dan vertical
<code>android:layout_centerVertical="true"</code>	Mengatur posisi view berada di tengah-tengah secara vertical
<code>android:layout_centerHorizontal="true"</code>	Mengatur posisi view berada di tengah-tengah secara horizontal

3. ImageView

ImageView adalah komponen view yang digunakan untuk menampilkan gambar, baik berupa file ataupun url gambar. Jika menggunakan file, maka property yang digunakan adalah `android:src="@<posisi_file>"`. Sedangkan jika menggunakan url, maka akan diatur pada file .kt yang menggunakan layoutnya.

Disamping beberapa property yang sudah dijelaskan, ada beberapa property yang umumnya digunakan oleh semua view:

Property	Fungsi
<code>android:layout_width=""</code>	Mengatur lebar view. <code>match_parent</code> , <code>fill_parent</code> jika ingin selebar parent dan <code>wrap_parent</code> jika ingin selebar isi viewnya
<code>android:layout_height=""</code>	Mengatur tinggi view. <code>match_parent</code> , <code>fill_parent</code> jika ingin setinggi parent dan <code>wrap_parent</code> jika ingin setinggi isi viewnya
<code>android:visibility=""</code>	Mengatur status tampil sebuah view. <code>invisible</code> jika ingin tidak terlihat. <code>visible</code> jika ingin terlihat <code>gone</code> jika ingin menghilang. Secara default, akan terisi dengan <code>visible</code> jika property tersebut tidak digunakan
<code>android:layout_margin=""</code>	Mengatur jarak luar semua sisi sebuah view. Diisi dengan satuan dp
<code>android:layout_marginTop=""</code>	Mengatur jarak luar sisi atas sebuah view
<code>android:layout_marginBottom=""</code>	Mengatur jarak luar sisi bawah sebuah view
<code>android:layout_marginRight=""</code>	Mengatur Jarak luar sisi kanan sebuah view
<code>android:layout_marginLeft=""</code>	Mengatur Jarak luar sisi kiri sebuah view
<code>android:layout_padding=""</code>	Mengatur jarak dalam semua sisi sebuah view. Diisi dengan satuan dp
<code>android:layout_paddingTop=""</code>	Mengatur jarak dalam sisi atas sebuah view
<code>android:layout_paddingBottom=""</code>	Mengatur jarak dalam sisi bawah sebuah view
<code>android:layout_paddingRight=""</code>	Mengatur Jarak dalam sisi kanan sebuah view
<code>android:layout_paddingLeft=""</code>	Mengatur Jarak dalam sisi kiri sebuah view
<code>android:background=""</code>	Mengatur warna latar sebuah view. Diisi dengan kode warna hexadesimal yang diawali dengan tanda #, misal <code>"#CCCCCC"</code>

◆ Memasang Validasi Input Pada Halaman Registrasi

Untuk memberikan sebuah perintah logika (function) pada sebuah view yang terdapat pada sebuah halaman tertentu, dalam hal ini adalah memvalidasi input dari form input halaman registrasi, diperlukan kepastian bahwa antara file .kt dan file .xml sudah saling terhubung. Untuk mengetahui hal tersebut dapat dilakukan dengan membuka file **MainActivity.kt** dan melihat pada bagian kode

```
setContentView(R.layout.activity_main)
```

Kode tersebut memiliki arti bahwa file **MainActivity.kt** menggunakan file **activity_main.xml** sebagai layoutnya.

Selanjutnya, komponen view pada **activity_main.xml** yang ingin diberikan perintah perlu dikenali pada MainActivity.kt. Ada dua cara untuk melakukan hal tersebut, yaitu dengan menggunakan metode **findViewById** dan menggunakan fitur **Kotlin Android Extension**. Kedua metode tersebut akan dibahas. Akan tetapi hanya akan digunakan metode **Kotlin Android Extension** karena metode tersebut dapat membuat kode menjadi lebih ringkas.

◆ Menggunakan Metode findViewById

Pada metode ini, hal pertama yang dilakukan adalah membuat variabel view kemudian memasang kode findViewById pada variable tersebut.

```
class MainActivity : AppCompatActivity() {  
  
    var edtName : EditText? = null  
    var spinnerGender : Spinner? = null  
    var edtEmail : EditText? = null  
    var edtTelp : EditText? = null  
    var edtAlamat : EditText? = null  
  
    override fun onCreate(savedInstanceState: Bundle?)  
    {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        edtName = findViewById(R.id.edtName)  
        spinnerGender = findViewById(R.id.spinnerGender)  
        edtEmail = findViewById(R.id.edtEmail)  
        edtTelp = findViewById(R.id.edtTelp)  
        edtAlamat = findViewById(R.id.edtAddress)  
    }  
}
```

```

    }
}

```

◆ Menggunakan Metode Kotlin Android Extension

Dengan Kotlin Android Extension, penggunaan kode seperti pada contoh di atas tidak perlu dilakukan. Fungsi kode di atas digantikan dengan hanya menambahkan satu baris kode di bawah ini:

```
import kotlinx.android.synthetic.main.activity_main.*
```

Fungsi validasi akan dijalankan ketika pengguna menekan Button Simpan. Maka, Button tersebut harus diberikan OnClickListener.

```
btnSave.setOnClickListener { }
```

Kemudian menambahkan fungsi `validasiInput()` dan diletakkan di dalam fungsi `btnSave.setOnClickListener { validasiInput() }`. Lalu letakkan kursor diantara huruf pada nama fungsi `validasiInput` tersebut dan klik **alt + enter**, pilih **create function 'validasiInput'** untuk membuat fungsi secara otomatis.

```
private fun validasiInput() {
    val namaInput = edtName.text.toString()
    val emailInput = edtEmail.text.toString()
    val telpInput = edtTelp.text.toString()
    val alamatInput = edtAddress.text.toString()

    when{
        namaInput.isEmpty() -> edtName.error = "Nama tidak boleh kosong"
        emailInput.isEmpty() -> edtEmail.error = "Email tidak boleh kosong"
        telpInput.isEmpty() -> edtTelp.error = "Telp tidak boleh kosong"
        alamatInput.isEmpty() -> edtAddress.error = "Alamat tidak boleh kosong"

        else -> {
            Toast.makeText(this,
                "Navigasi ke halaman profil diri", Toast.LENGTH_SHORT).show()
        }
    }
}
```

```
}
```

◆ Memasang Pilihan Data Pada Spinner

Seperti diketahui bahwa ada pilihan jenis kelamin pada halaman registrasi akan tetapi data yang ditampilkan belum diatur. Pada poin ini, akan dibahas cara menambahkan pilihan data pada komponen view berupa Spinner sebagai berikut:

1. Menambahkan data array pilihan pada file strings.xml yang terletak pada direktori `app/res/values/strings.xml`

```
<resources>
    <string name="app_name">MyProfileApp</string>

    <string-array name="gender_list">
        <item>Pilih kelamin</item>
        <item>Laki-laki</item>
        <item>Perempuan</item>
    </string-array>

</resources>
```

2. Kemudian tambahkan properti `android:entries="@array/gender_list"` pada view Spinner pada layout `activity_main.xml`
3. Menambahkan fungsi baru pada `MainActivity.kt` dan letakan di dalam fungsi `onCreate`

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

    btnSave.setOnClickListener { validasiInput() }

    setDataSpinnerGender()
}

private fun setDataSpinnerGender() {
    val adapter = ArrayAdapter.createFromResource(this,
        R.array.gender_list, android.R.layout.simple_spinner_item)

    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)

    spinnerGender.adapter = adapter
}
```

4. Menambahkan validasi SpinnerGender pada fungsi `validasiInput()`.

```
private fun validasiInput() {  
    val namaInput = edtName.text.toString()  
    val emailInput = edtEmail.text.toString()  
    val telpInput = edtTelp.text.toString()  
    val alamatInput = edtAddress.text.toString()  
    val genderInput = spinnerGender.selectedItem.toString()  
  
    when{  
        namaInput.isEmpty() -> edtName.error = "Nama tidak boleh kosong"  
        genderInput.equals("Pilih kelamin") -> tampilToast("Kelamin harus  
dipilih")  
        emailInput.isEmpty() -> edtEmail.error = "Email tidak boleh  
kosong"  
        telpInput.isEmpty() -> edtTelp.error = "Telp tidak boleh kosong"  
        alamatInput.isEmpty() -> edtAddress.error = "Alamat tidak boleh  
kosong"  
  
        else -> {  
            tampilToast("Navigasi ke halaman profil diri")  
        }  
    }  
}
```

3.5. Memasang Intent Pada Aplikasi Profil Diri

Intent adalah sebuah class yang dapat digunakan untuk perpindahan (navigasi) ke halaman (activity) lain atau halaman itu sendiri dengan atau tanpa mengirim data. Pada aplikasi profil diri, Intent akan digunakan untuk keperluan navigasi dari activity `MainActivity.kt` (Registrasi) ke activity `MyProfileActivity.kt` (Profil Diri) dengan mengirimkan data yang diisi pada halaman registrasi.

Dalam penggunaannya untuk navigasi, sebuah Intent harus diatur terlebih dahulu dari mana dan akan ke mana navigasinya dilakukan. Kemudian barulah sebuah Intent dijalankan (execute) dengan fungsi `startActivity(intent)`.

Untuk memastikan penggunaan Intent berjalan pada aplikasi, dapat dibuat sebuah fungsi navigasi tanpa mengirimkan data kemudian jalankan aplikasi. Berikut adalah kode yang digunakan:

```
private fun navigasiKeProfilDiri() {

    val intent = Intent(this, MyProfileActivity::class.java)
    startActivity(intent)

}
```

Kemudian *execute* fungsi tersebut di dalam fungsi *validasiInput* ketika kondisi semua input diisi:

```
else -> {

    tampilToast("Navigasi ke halaman profil diri")
    navigasiKeProfilDiri()

}
```

Untuk mengirimkan data pada saat navigasi secara bersamaan, dapat memanfaatkan penggunaan salah satu class yang disebut Bundle. Akan tetapi beberapa variable yang digunakan untuk menampung input harus dibuat menjadi variable global (var) terlebih dahulu. Berikut adalah contoh penggunaan Bundle pada Intent untuk kebutuhan mengirimkan data pada saat melakukan navigasi.

```
private fun navigasiKeProfilDiri() {

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

    val bundle = Bundle()
    bundle.putString("nama", namaInput)
    bundle.putString("gender", genderInput)
    bundle.putString("email", emailInput)
    bundle.putString("telp", telpInput)
    bundle.putString("alamat", alamatInput)

    intent.putExtras(bundle)

    startActivity(intent)

}
```


♦ Mengambil Intent Data Pada Halaman Profil Diri

Data yang dikirimkan dengan Intent dapat diambil pada halaman tujuan (MyProfileActivity.kt). Karena, pada saat dikirimkan menggunakan Bundle, maka untuk mengambilnya pun menggunakan Bundle dengan kode seperti di bawah ini:

```
val bundle = intent.extras
```

Setelah itu, data akan diambil dengan menggunakan nama variable dan tipe data sesuai dengan yang digunakan pada saat dikirim. Contoh, untuk mengambil data yang dikirimkan dengan tipe String dan variable "nama":

```
val nama = bundle.getString("nama")
```

Pada MyProfileActivity.kt, dibuat sebuah fungsi untuk mengambil dan menampilkan data yang dikirim sebagai berikut:

```
private fun ambilData() {  
    val bundle = intent.extras  
  
    val nama = bundle.getString("nama")  
    val gender = bundle.getString("gender")  
    val email = bundle.getString("email")  
    val telp = bundle.getString("telp")  
    val alamat = bundle.getString("alamat")  
  
    txtName.text = nama  
    txtGender.text = gender  
    txtEmail.text = email  
    txtTelp.text = telp  
    txtAddress.text = alamat  
  
}
```

Referensi

Android Studio

<https://developer.android.com/studio>

<https://developer.android.com/studio/install>

Kotlin

<https://kotlinlang.org/docs/reference/android-overview.html>

<https://kotlinlang.org/docs/tutorials/android-plugin.html>

<https://antonioleiva.com/kotlin-android-extensions/>

Struktur Project Android

<https://medium.com/mindorks/ultimate-android-app-project-structure-a-d-s-003-4e8ce8499469>

<https://developer.android.com/studio/projects/>

<https://medium.com/@mikelimantara/overview-of-android-project-structure-and-naming-conventions-b08f6d0b7291>

<https://code.tutsplus.com/tutorials/android-project-structure--mobile-12781>

Android Spinner

<https://tutorial.eyehunts.com/android/android-spinner-with-example-in-kotlin/>

AndroidX

<https://developer.android.com/jetpack/androidx/migrate>

Selamat Belajar !!

Pahami ini dulu, baru memahami dia 😊

Jika kamu merasa sulit, artinya kamu selangkah lebih maju dari yang lain 😊

Dan percayalah kamu bisa 😊

#washinghands #stayhealthy #stayathome #physichaldistancing