

Tugas 1

Android Kotlin Beginner

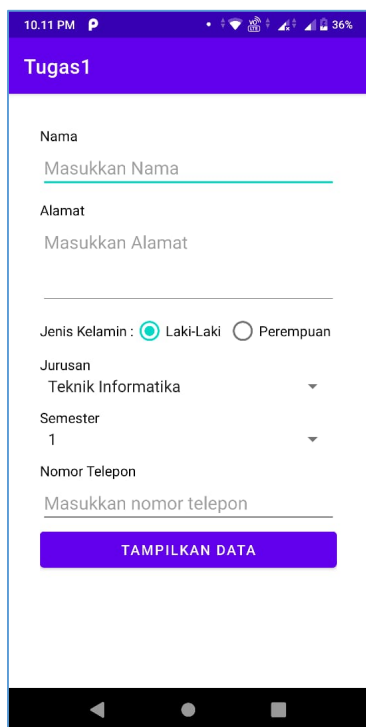
Membuat aplikasi mengirim object (buka kembali materi Intent Activity) dengan kriteria sebagai berikut :

1. Menggunakan tipe-tipe inputan, seperti EditText, RadioButton, Spinner (untuk spinner disertakan tutorial disini).
2. Membuat validasi setiap EditText
3. Membuat **data class** untuk menampung data yang akan dikirim
4. Implementasi **Parcelable** untuk mengirim data (disertakan tutorial disini)
5. Menerapkan calling view dengan **ViewBinding** (disertakan tutorial disini)
6. Untuk EditText telepon, tipe inputan harus telepon, agar keyboard yang muncul angka semua
7. Di activity kedua / detail, Button Telepon dan SMS bisa diklik, Button Telepon ke dial phone, Button SMS ke aplikasi SMS
8. Tema data bebas, **tidak boleh data Mahasiswa**
9. Desain UI boleh bebas, spt warna, gambar, dll
10. Jika ada kendala, jangan sungkan untuk bertanya di grup WhatsApp

Selamat mengerjakan...

Alur aplikasi

1. Tampilan awal



10.11 PM P 36%

Tugas1

Nama
Masukkan Nama

Alamat
Masukkan Alamat

Jenis Kelamin : ☒ Laki-Laki ☐ Perempuan

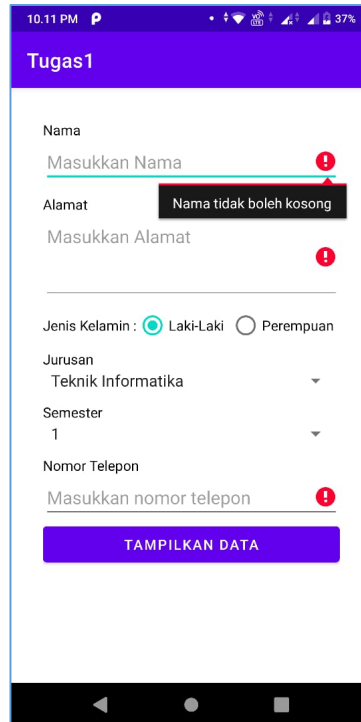
Jurusan
Teknik Informatika

Semester
1

Nomor Telepon
Masukkan nomor telepon

TAMPILKAN DATA

2. Validasi data kosong



10.11 PM P 37%

Tugas1

Nama
Masukkan Nama

Alamat
Masukkan Alamat

Jenis Kelamin : ☒ Laki-Laki ☐ Perempuan

Jurusan
Teknik Informatika

Semester
1

Nomor Telepon
Masukkan nomor telepon

TAMPILKAN DATA

3. Data diisi semua

10.19 PM

Tugas1

Nama
Asad

Alamat
BSD

Jenis Kelamin : ☒ Laki-Laki ☐ Perempuan

Jurusan
Teknik Informatika

Semester
7

Nomor Telepon
08123456789

TAMPILKAN DATA

4. Data ditampilkan

10.19 PM

Tugas1

Nama
Asad

Alamat
BSD

Jenis Kelamin
Laki-laki

Jurusan
Teknik Informatika

Semester
7

Nomor telepon
08123456789

TELEPON **SMS**

5. Ketika Button Telepon diklik

10.47 PM

+ Buat kontak baru
+ Tambahkan ke kontak
Kirim pesan

0812-3456-789

1 QW 2 ABC 3 DEF
4 GHI 5 JKL 6 MNO
7 PQRS 8 TUV 9 WXYZ
* 0 + #

TELEPON

6. Ketika Button SMS diklik

10.47 PM

← 0812-3456-789

10.47 PM

Percakapan dengan 0812-3456-789

SMS
dari TELKOMSEL

Tutorial – tutorial

Tutorial disini tidak semua, hanya beberapa saja, yang belum pernah diajarkan selama training online.

1. Konfigurasi Parcelable & ViewBinding

Buka file build.gradle level module, tambahkan syntax berikut :

A. Konfigurasi Parcelable

B. Konfigurasi ViewBinding

Jika sudah, klik Sync Now di kanan atas

```
plugins {  
    id 'com.android.application'  
    id 'kotlin-android'  
    id 'kotlin-android-extensions' A  
}  
  
android {  
    compileSdkVersion 30  
    buildToolsVersion "30.0.3"  
  
    defaultConfig {  
        applicationId "id.asad.tugas1"  
        minSdkVersion 21  
        targetSdkVersion 30  
        versionCode 1  
        versionName "1.0"  
  
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"  
    }  
  
    buildTypes {  
        release {  
            minifyEnabled false  
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'  
        }  
    }  
  
    compileOptions {  
        sourceCompatibility JavaVersion.VERSION_1_8  
        targetCompatibility JavaVersion.VERSION_1_8  
    }  
  
    kotlinOptions {  
        jvmTarget = '1.8'  
    }  
  
    buildFeatures{  
        viewBinding = true B  
    }  
}
```

2. Implementasi Parcelable

Tambahkan annotation **@Parcelize** dan supertype **Parcelable**

```
@Parcelize
data class Mahasiswa(
    var nama : String,
    var alamat : String,
    var jenisKelamin : String,
    var jurusan : String,
    var semester : Int,
    var nomorTelepon : String
) : Parcelable
```

2. Implementasi ViewBinding

Sebelum implementasi ViewBinding (findViewById)

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

    <TextView
        android:id="@+id/tv_result"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Ha"/>

</LinearLayout>
```

```
class HomeActivity : AppCompatActivity() {

    lateinit var tvResult : TextView

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

        tvResult = findViewById(R.id.tv_result)

        // usage
        tvResult.text = "Hello World"
    }
}
```

Menggunakan ViewBinding, jadi hanya ditambahkan syntax **binding** di depan view/widget

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

    <TextView
        android:id="@+id/tv_result"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Ha"/>

</LinearLayout>
```

```
class HomeActivity : AppCompatActivity() {

    lateinit var binding : ActivityHomeBinding

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

        // usage
        binding.tvResult.text = "Hello World"
    }
}
```

3. Implementasi Spinner

Syntax XML

```
<Spinner
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/spin_jurusan"/>
```

Syntax logic kotlin

```
val arrayJurusan : Array<String> = arrayOf("Teknik Informatika","Teknik Mesin","Teknik Sipil","Teknik Industri","Teknik Elektro")
val spinAdapterJurusan : ArrayAdapter<String!> = ArrayAdapter( context: this, android.R.layout.simple_spinner_item, arrayJurusan)
binding.spinJurusan.adapter = spinAdapterJurusan
```

Ambil data yang dipilih dari spinner

```
val jurusan : String = binding.spinJurusan.selectedItem.toString()
```

4. Contoh validasi jika data belum diisi oleh user

```
// validasi jika kosong / belum terisi
if (nama.isBlank()){
    binding.edtNama.error = "Nama tidak boleh kosong"
}
if(alamat.isBlank()){
    binding.edtAlamat.error = "Alamat tidak boleh kosong"
}
if(nomorTelepon.isBlank()){
    binding.edtTelepon.error = "Nomor telepon tidak boleh kosong"
}

// jika sudah terisi semua
if(!nama.isBlank() && !alamat.isBlank() && !nomorTelepon.isBlank()){
}
}
```