**LAPORAN PRATIKUM 2**

**PRAKTIKUM PEMROGRAMAN BERGERAK**

**“Instalasi dan Kotlin Dasar”**

**Description: Logo

Description automatically generated**

**Disusun Oleh:**

**Ariyandra Putera Caisar**

**21076038**

**Dosen Pengampu :**

**Geovanne Farell, S.P.d.,M.Pd.T.**

**PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA**

**JURUSAN TEKNIK ELEKTRONIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS NEGERI PADANG**

**2023**

1. LEARNING OUTCOMES PRAKTIKUM

Mampu menggunakan dan menjelaskan Lateinit pada Kotlin

1. ALAT dan BAHAN

1. Personal Computer

2. Android studio

3. Koneksi Internet

1. TEORI SINGKAT

**Kotlin**

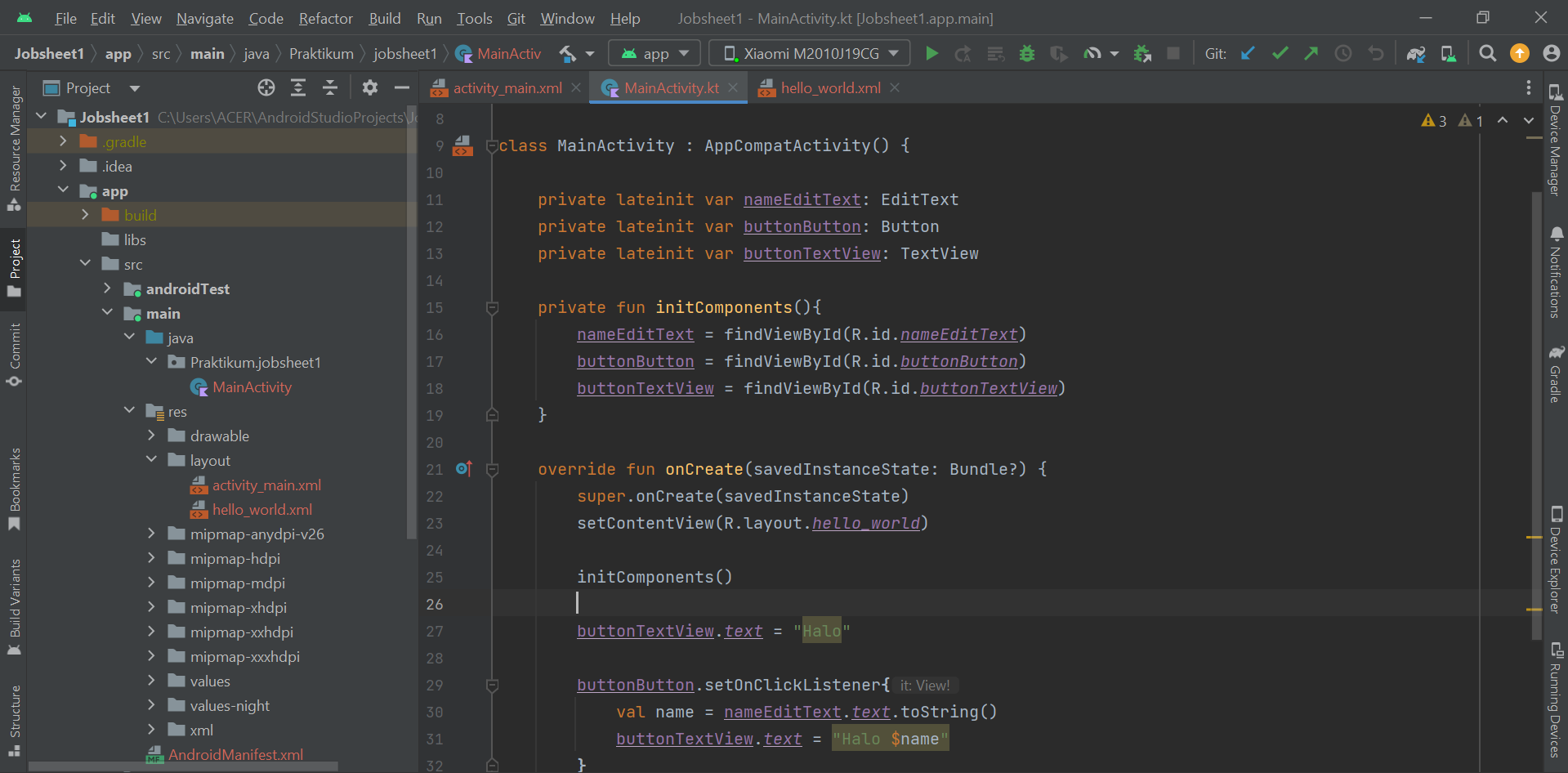
Dalam Kotlin, lateinit digunakan untuk mendeklarasikan properti yang dapat diinisialisasi nanti, meskipun pada awalnya tidak diberikan nilai. Ini sering digunakan dalam kasus di mana Anda ingin menunda inisialisasi properti hingga kemudian dalam kode, tetapi Anda ingin menghindari pengecekan null atau inisialisasi awal yang tidak perlu. Namun, ada beberapa hal yang perlu diperhatikan ketika menggunakan lateinit:

* Hanya Digunakan Untuk Properti Tipe Non-Nullable: Anda hanya dapat menggunakan lateinit untuk properti yang memiliki tipe non-nullable. Properti dengan tipe nullable masih harus diinisialisasi dengan nilai awal atau nilai null.
* Tidak Bisa Digunakan dengan Tipe Dasar (Primitive Types): lateinit tidak bisa digunakan dengan tipe data primitif seperti Int, Boolean, dll. Namun, Anda dapat menggunakan kelas kotliln yang membungkus tipe dasar, seperti Int diganti dengan Int? atau java.lang.Integer.
* Inisialisasi Sebelum Digunakan: Anda harus memastikan bahwa properti yang dinyatakan dengan lateinit diinisialisasi sebelum digunakan. Jika Anda mencoba mengakses properti lateinit sebelum diinisialisasi, itu akan menyebabkan UninitializedPropertyAccessException.

1. D. LANGKAH KERJA

**Lateinit**

* 1. Untuk membuat lateinit pada Main Activity Tambahkan codingan seperti berikut:



* 1. Jalankan Virtual Devicenya, maka akan tampil hasil dari aplikasi seperti berikut:



* 1. Mengambil text berdasarkan resources

pada Main Activity Tambahkan codingan seperti berikut:

Code Sebelumnya :

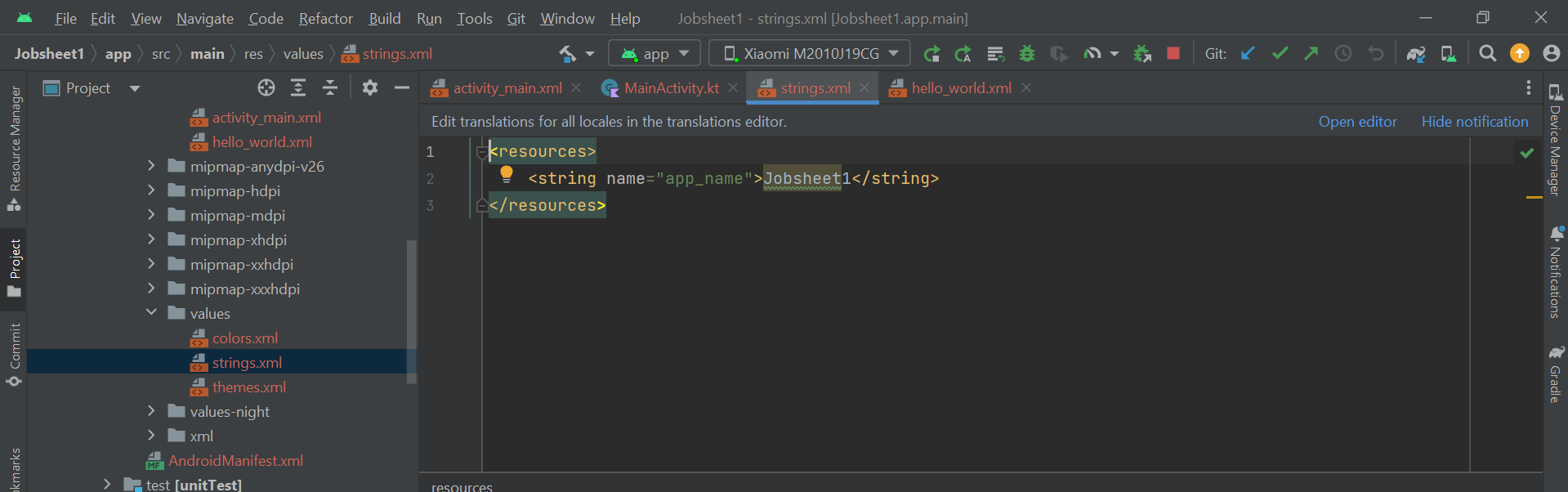
ButtonTextView.text = “Halo”

Ganti menjadi seperti berikut:

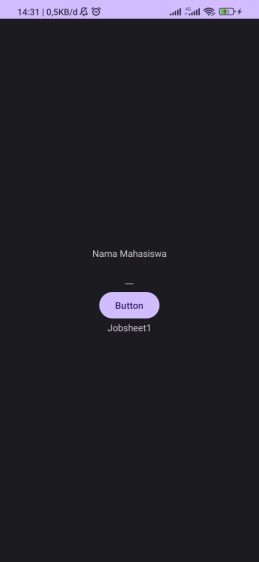


* 1. Darimana dipanggil string.app\_name?

Itu berasal dari folder berikut

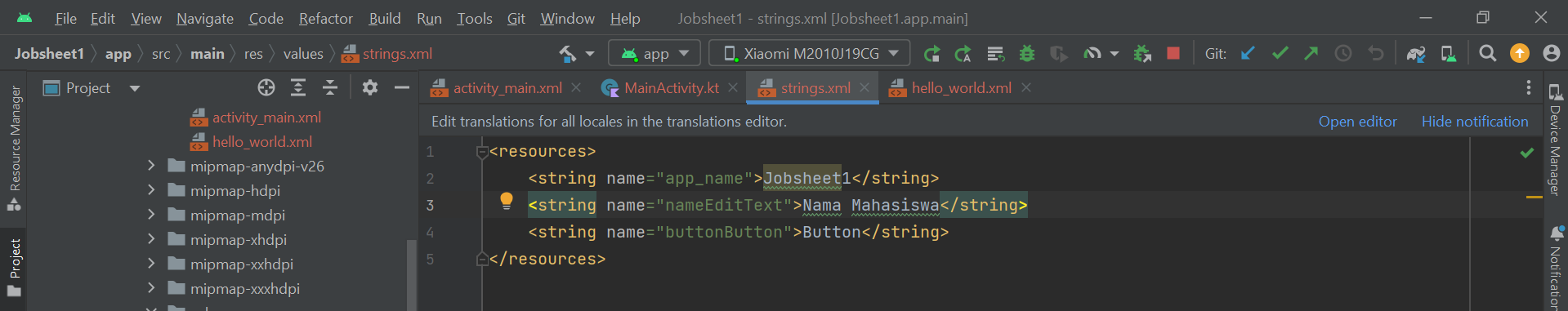


* 1. Jalankan Virtual Devicenya, maka akan tampil hasil dari aplikasi seperti berikut:



* 1. String Resources

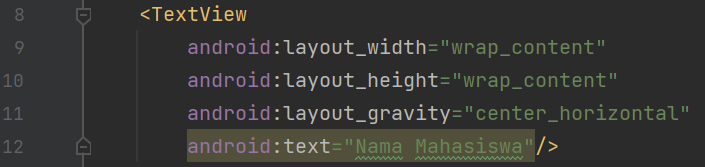
Buka folder res-values-strings.xml Tambahkan codingan seperti berikut:



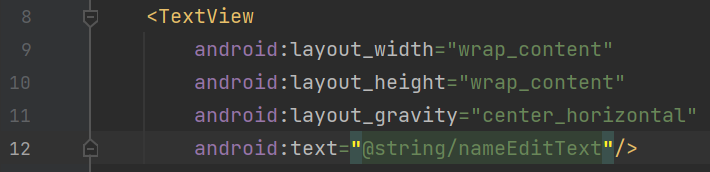
Kita bisa juga menggunakan string resource pada tampilan Layout,

* 1. Tambahkan Code Berikut pada hello\_world.xml

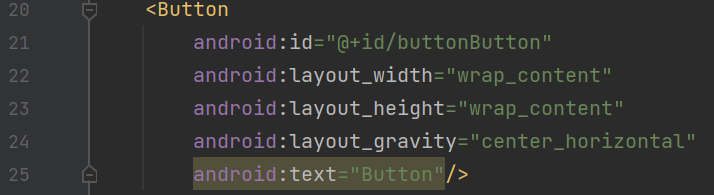
Code Sebelumnya :



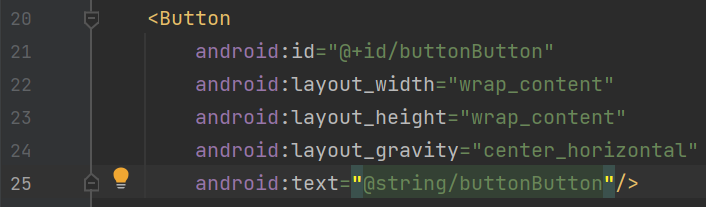
Ganti menjadi seperti berikut:



Code Sebelumnya :



Ganti menjadi seperti berikut:

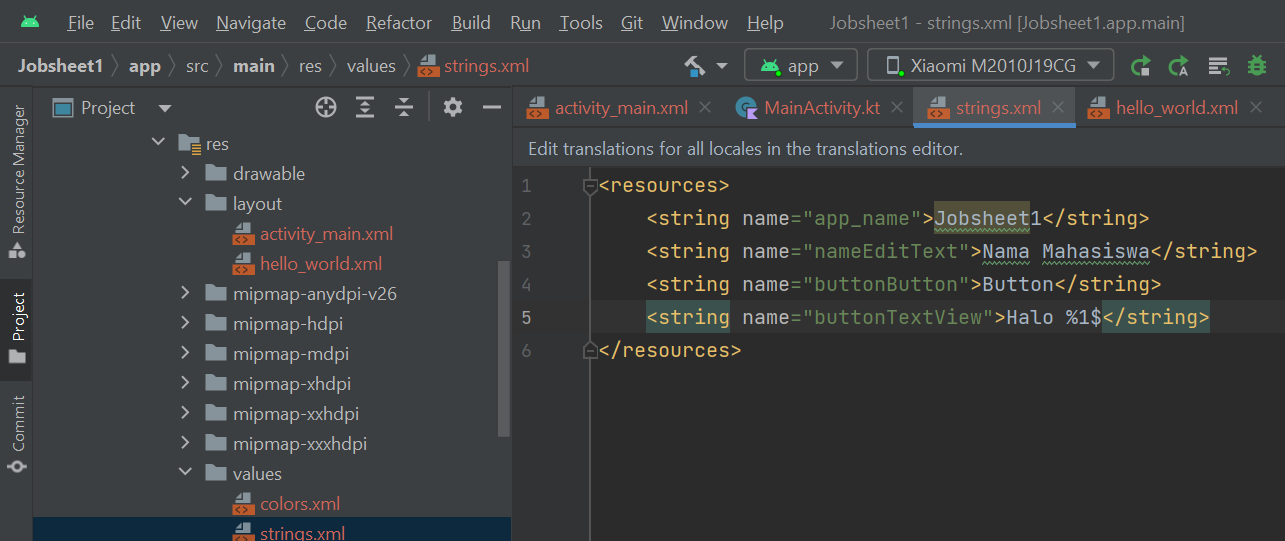


* 1. Jalankan Virtual Devicenya, maka akan tampil hasil dari aplikasi seperti berikut:

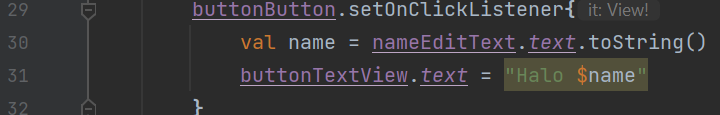
Note : Seharusnya hasil ouput tampilan sama dengan yang sebelumnya

* 1. Formating String Resource

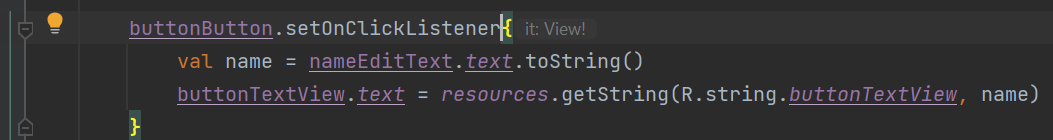
Buka folder res-values-strings.xml Tambahkan codingan seperti berikut:



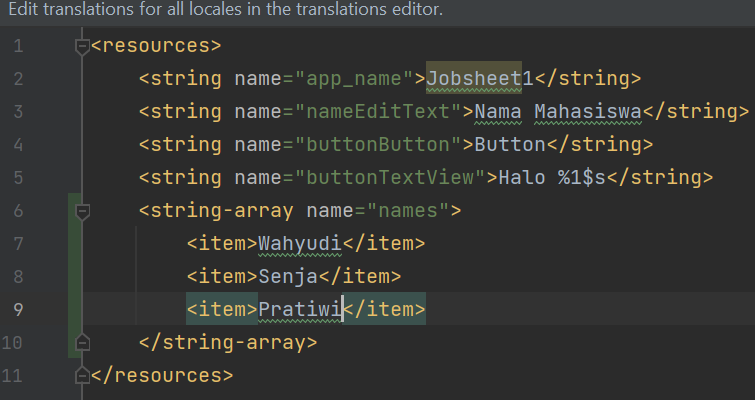
Code Sebelumnya :



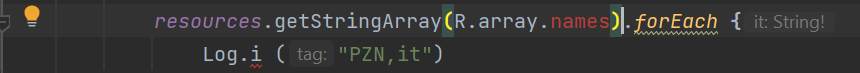
Ganti menjadi seperti berikut:



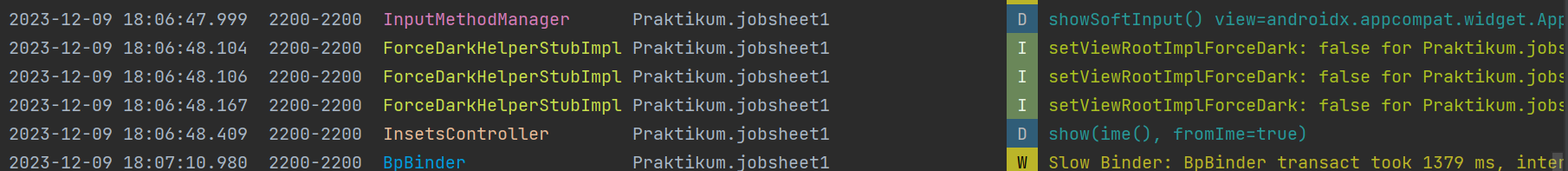
* 1. Jalankan Virtual Devicenya, maka akan tampil hasil dari aplikasi seperti berikut:
  2. String Array Resources
  3. Buka folder res-values-strings.xml Tambahkan codingan seperti berikut:



* 1. pada Main Activity Tambahkan codingan seperti berikut



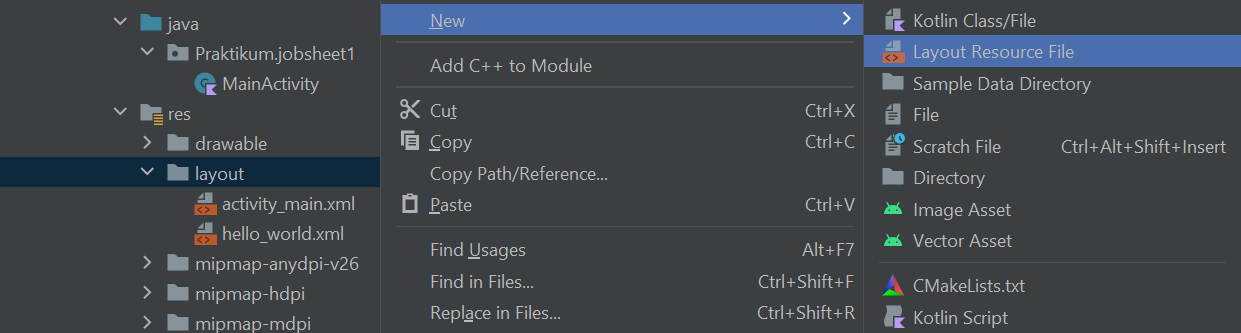
* 1. Jalankan Virtual Devicenya, maka akan tampil hasil dari aplikasi seperti berikut:
  2. Setelah aplikasi dijalankan, pilih Logcat untuk melihat list array sebelumnya



* 1. Layout Resource

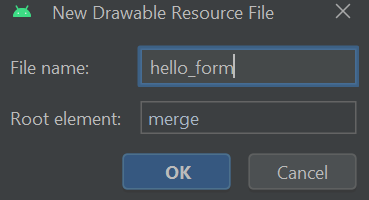
Buat layout baru

New - Layout Resource File

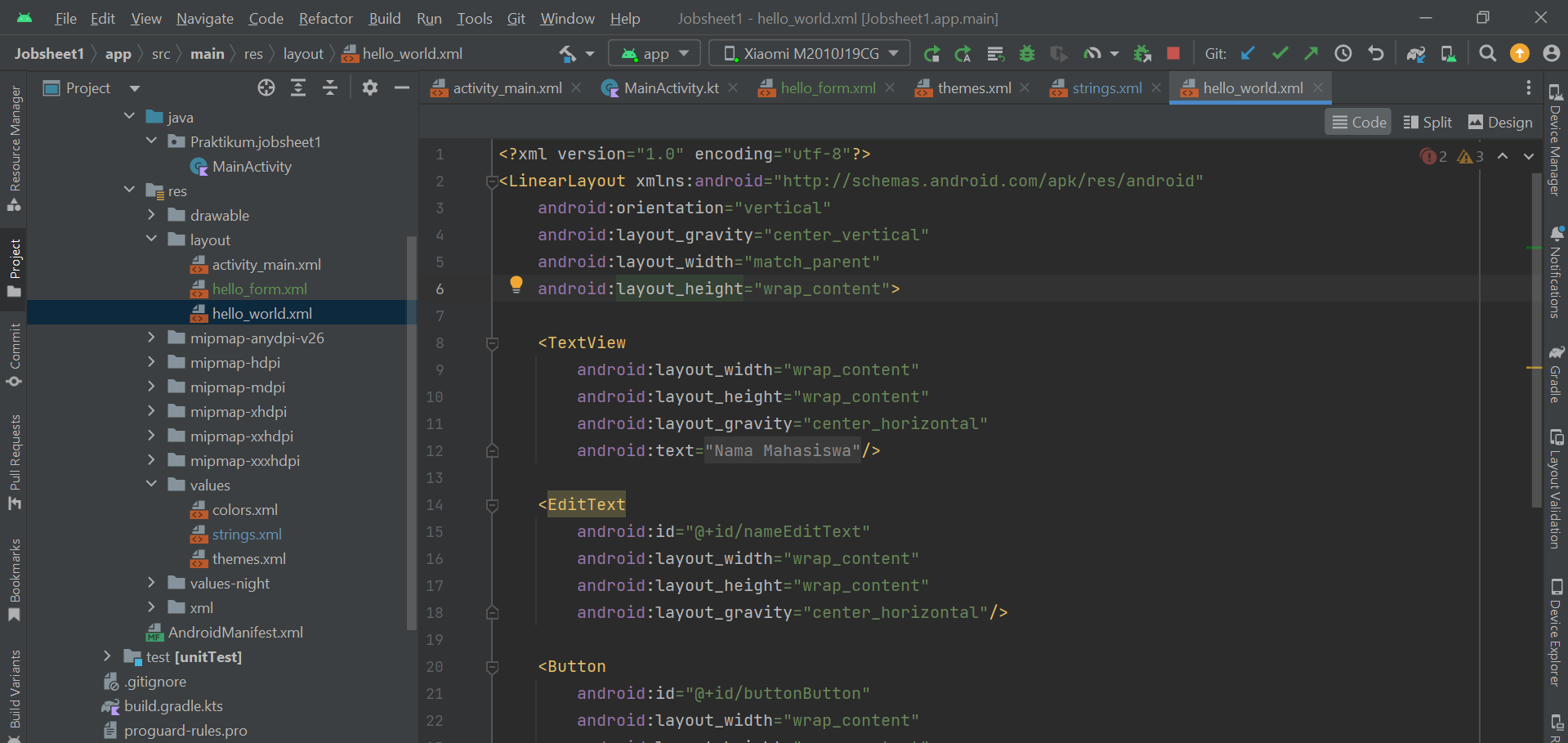


* 1. File Name “hello\_form”

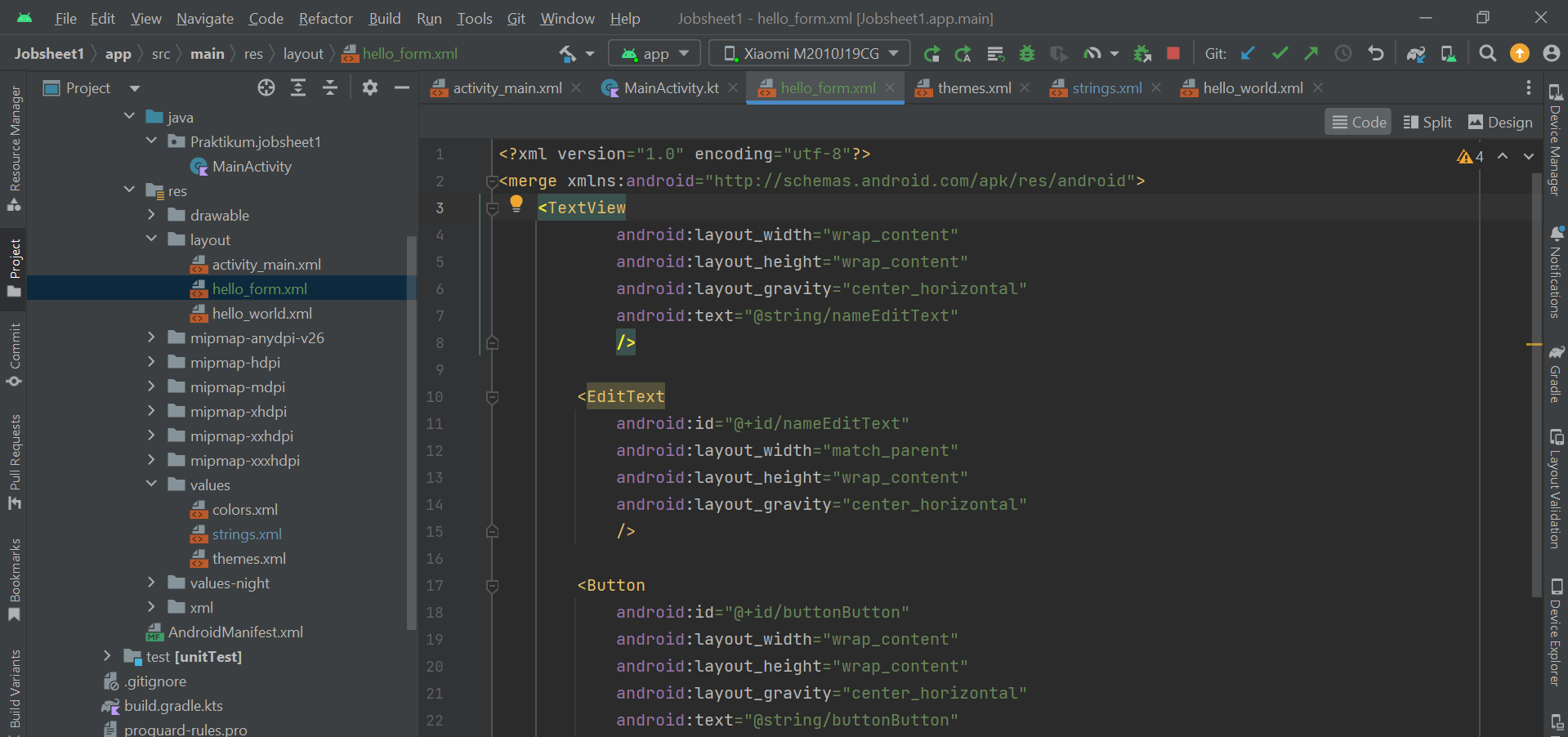
Root Elemen “merge



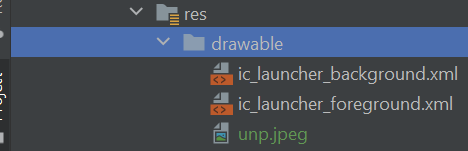
* 1. Pindahkan Code berikut dari hello\_world.xml ke hello\_form.xml



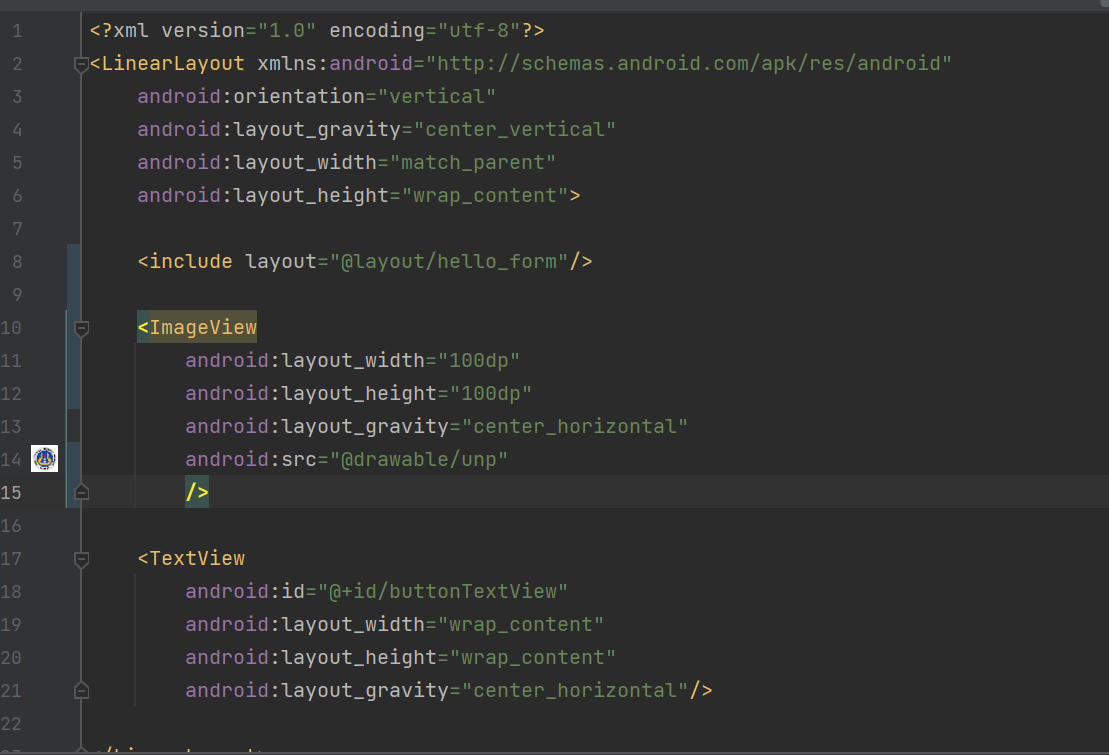
* 1. Pada hello\_world.xml tambahkan codingan seperti berikut :



* 1. Jalankan Virtual Devicenya, maka akan tampil hasil dari aplikasi akan sama seperti sebelumnya.
  2. Menambahkan Gambar Pada Kotlin
  3. Buka Folder res-drawable, copy file gambar kedalam folder tersebut



* 1. Pada hello\_world.xml tambahkan codingan seperti berikut :



* 1. Jalankan Virtual Devicenya, maka akan tampil hasil dari aplikasi seperti berikut:

