

Tujuan Praktikum

1. Mahasiswa mampu mengoperasikan Android Studio.
2. Mahasiswa mampu membuat sebuah aplikasi penghitung volume balok.

Tools

1. Android Studio

Panduan ini menggunakan versi Android Studio: Giraffe - 2022.3.1

Android API: Level 34

Introduction to Android Studio

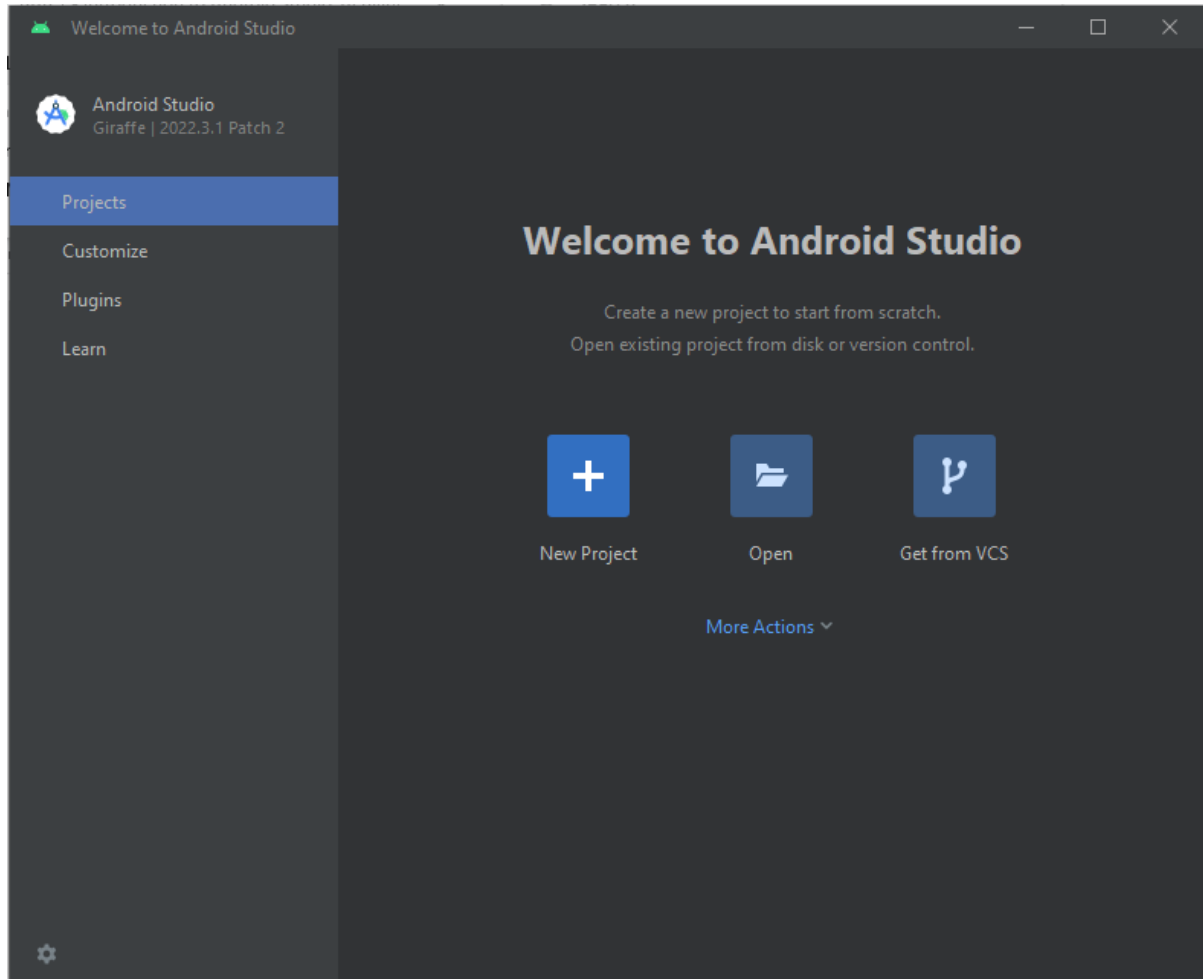
Android Studio merupakan sebuah Integrated Development Environment (IDE) resmi untuk pengembangan aplikasi pada sistem operasi Android milik Google. Android Studio dibangun di atas IntelliJ IDEA milik JetBrains yang didesain khusus untuk



pengembangan aplikasi berbasis Android. Android Studio saat ini tersedia pada sistem operasi Windows, macOS, dan Linux. Android Studio merupakan IDE utama untuk pengembangan aplikasi Android native menggantikan Eclipse Android Development Tools (E-ADT).

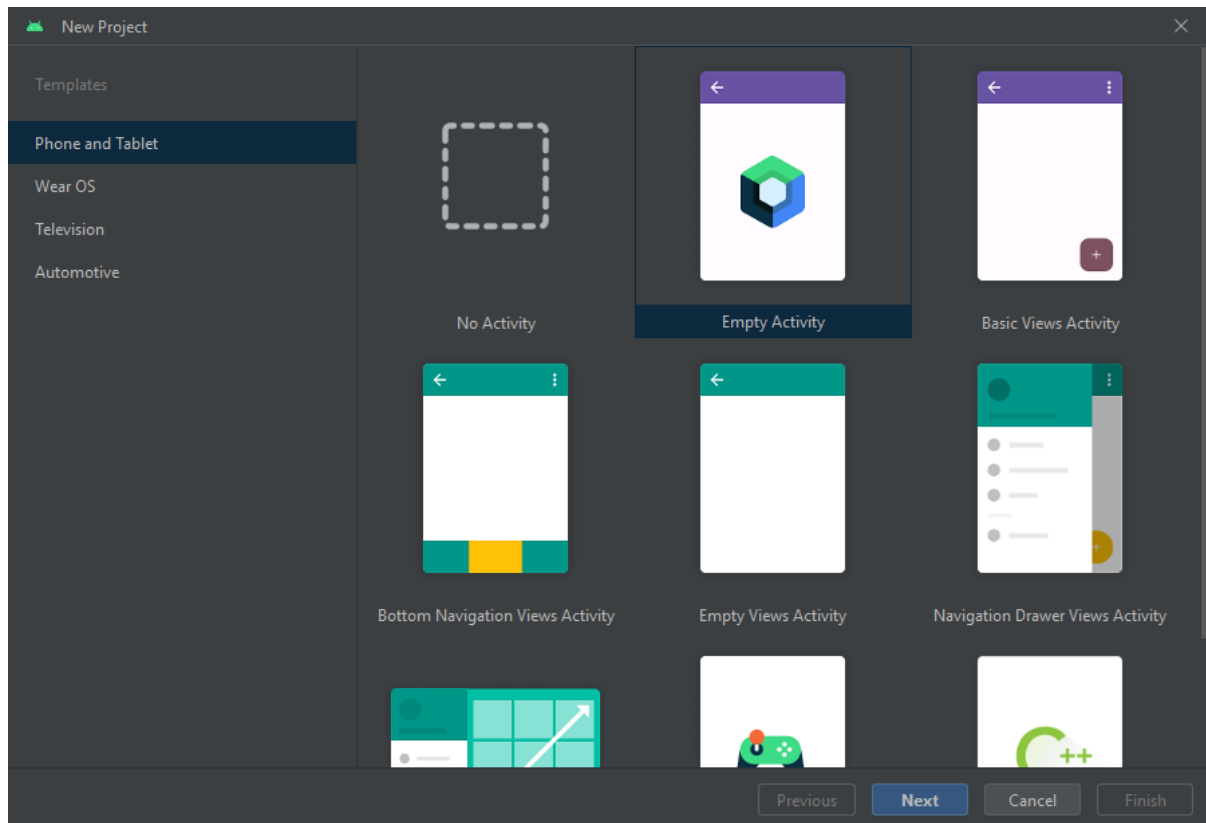
Membuat Project Aplikasi Android

Buka Android Studio, pada saat pertama kali Android Studio dijalankan, akan tampil halaman seperti ini:

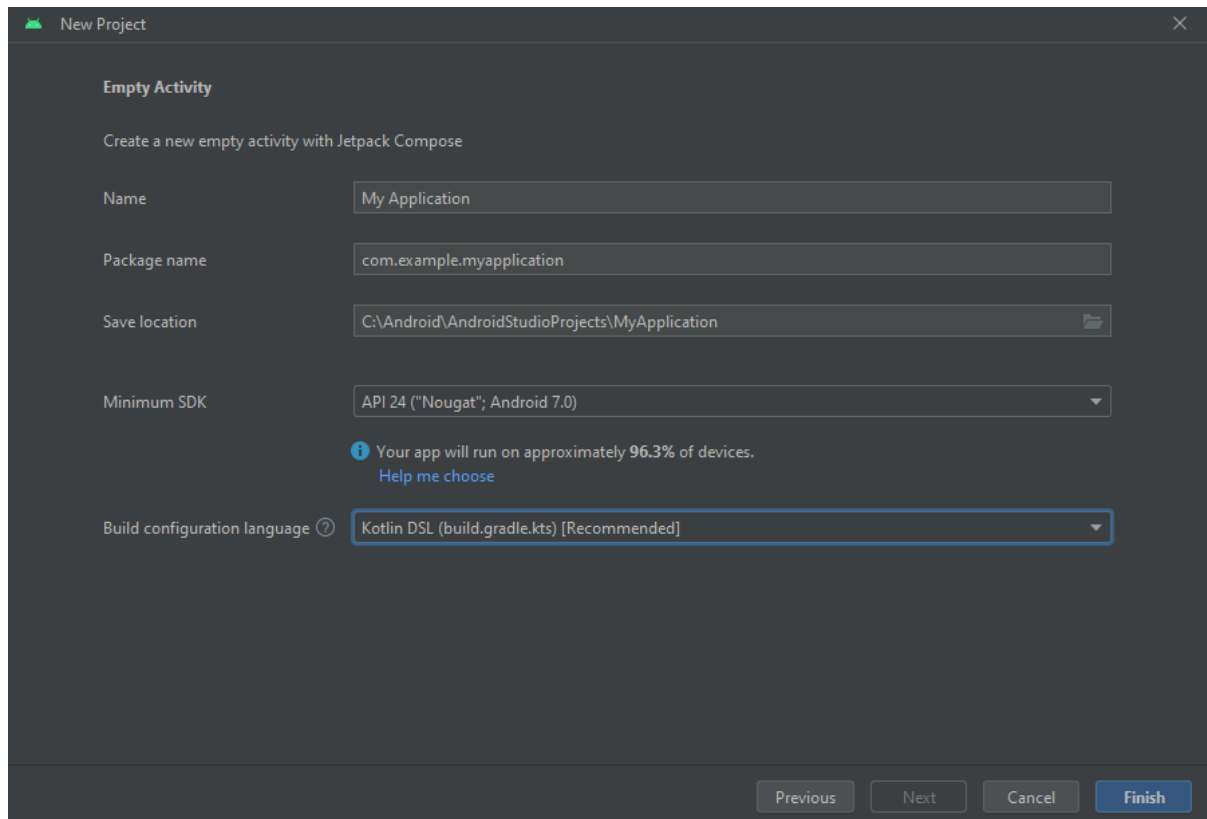


Untuk membuat sebuah project baru, pilih menu **New Project**. Setelah itu, akan tampil sebuah halaman **project wizard** tempat anda melakukan konfigurasi awal project yang akan dibuat. Ikutilah langkah berikut untuk melakukan konfigurasi awal project aplikasi Android.

1. Setelah menekan menu **New Project**, akan muncul sebuah jendela baru yang akan membantu kita untuk mengatur konfigurasi awal dari project kita. Pertama, kita akan diberikan beberapa template yang dapat kita gunakan pada halaman awal aplikasi kita. Pada pembahasan kali ini, kita akan memilih **Empty Activity**. Setelah itu, klik tombol **Next** untuk menuju pengaturan selanjutnya.



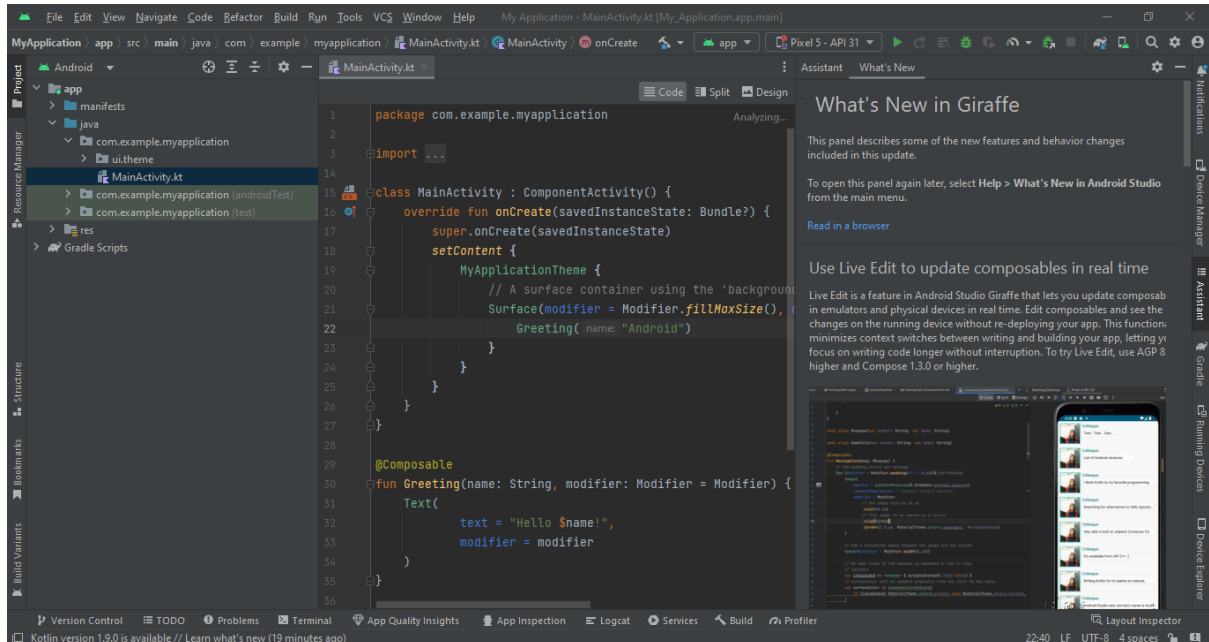
2. Pada halaman selanjutnya, kita akan diberikan beberapa isian seperti **Name**, **Package name**, **Save location**, **Minimum SDK**, dan **Build configuration language**.
 - a. **Name**: merupakan nama aplikasi yang akan dikembangkan.
 - b. **Package name**: digunakan untuk mengidentifikasi suatu aplikasi ketika akan dipublish, biasanya disesuaikan dengan domain suatu perusahaan yang mengembangkan aplikasi tersebut.
 - c. **Save location**: tempat source code project aplikasi disimpan.
 - d. **Minimum SDK**: versi Android minimum yang diperlukan suatu perangkat untuk menjalankan aplikasi yang akan dikembangkan.
 - e. **Build configuration language**: merupakan bahasa pemrograman yang digunakan dalam pengembangan aplikasi Android.



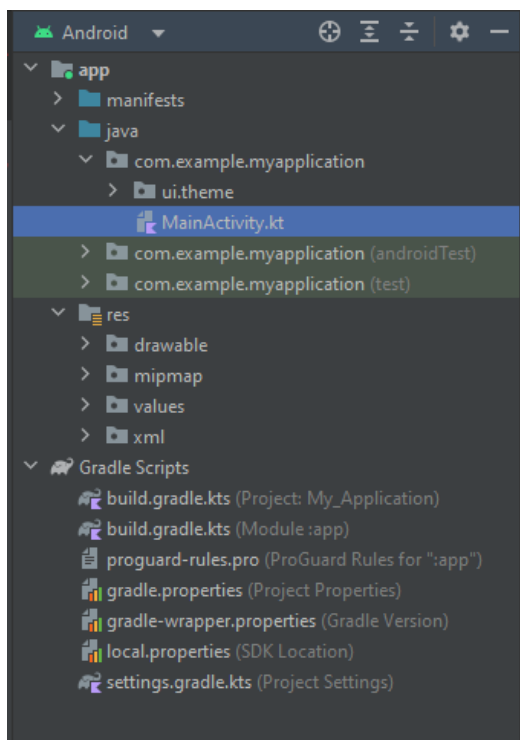
3. Sesuaikan **name**, **package name**, dan **save location** pada konfigurasi. Kemudian untuk menyelesaikan proses konfigurasi awal, tekan tombol **Finish**. Setelah itu, anda akan diarahkan ke jendela Android Studio.

Antarmuka Android Studio

Setelah menyelesaikan konfigurasi awal pada project pertama kita, Android Studio akan mengarahkan kita ke jendela utama tempat kita akan menghabiskan waktu untuk melakukan pengembangan aplikasi.

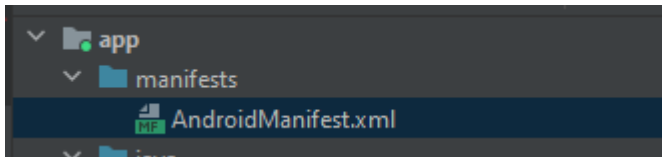


Struktur Project Aplikasi



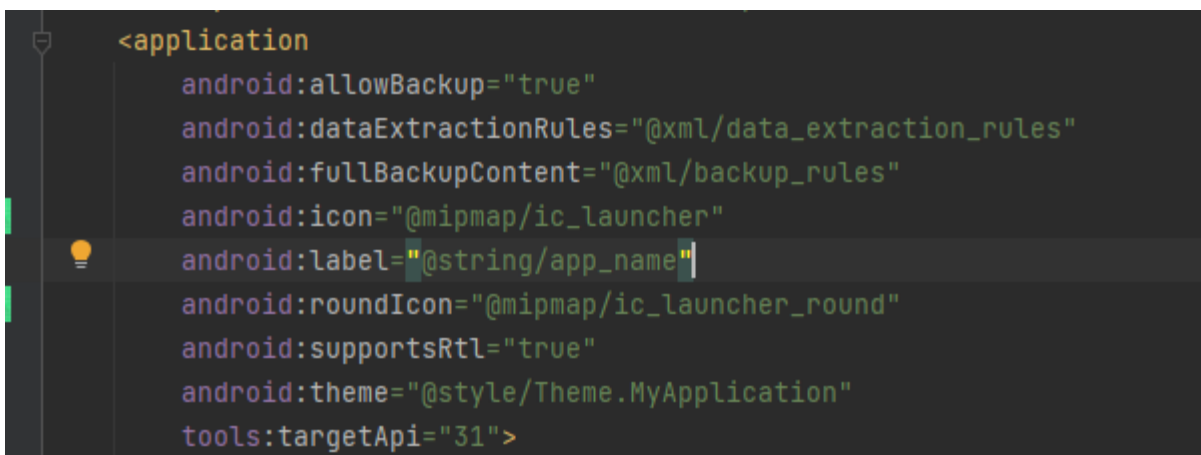
Pada setiap project aplikasi di Android Studio akan berisi satu atau beberapa modul dengan berkas **source code** dan **resource** (res). Android Studio akan menampilkan project kita dalam struktur default yaitu Android seperti pada gambar di samping. Mari kita bahas lebih rinci struktur project aplikasi pada Android Studio.

Android Manifest

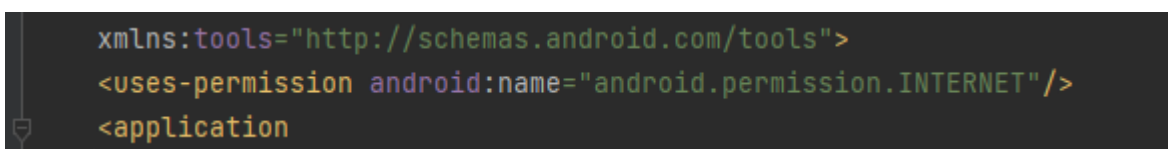


Android Manifest berisi informasi-informasi tentang aplikasi yang sedang kita kembangkan, seperti nama package dan ID aplikasi, permission atau izin yang dibutuhkan, komponen aplikasi seperti activity, dan masih banyak lagi.

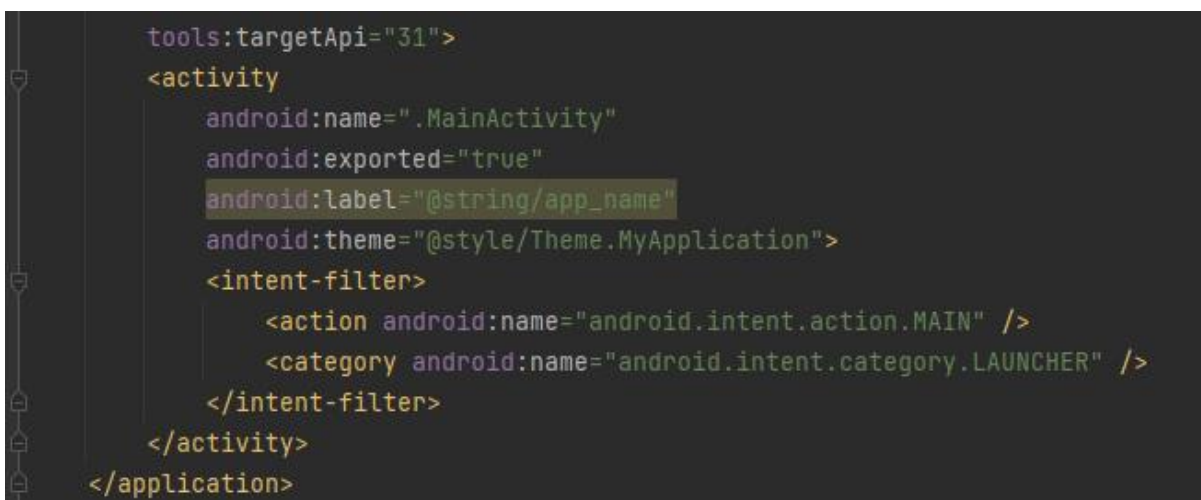
Nama package dan ID aplikasi



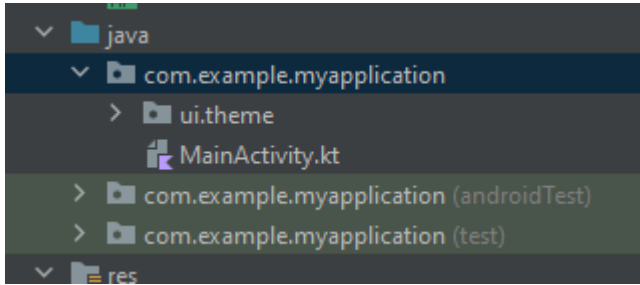
Permission yang dibutuhkan



Komponen Aplikasi



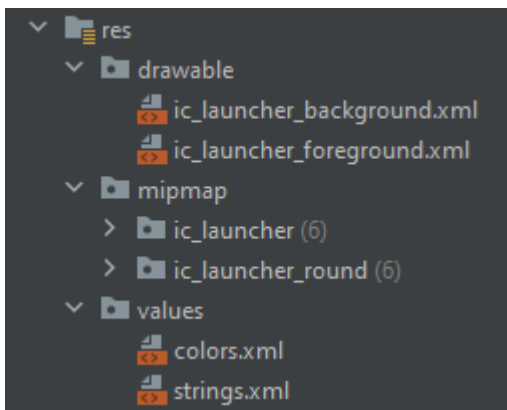
Java



Merupakan folder tempat kita menuliskan kode yang ditulis dalam bahasa Java/Kotlin. Folder dengan keterangan **androidTest** merupakan tempat untuk menulis kode

Instrumentation Test, dan folder dengan keterangan **test** merupakan tempat untuk menulis kode Unit Test.

Res / Resource

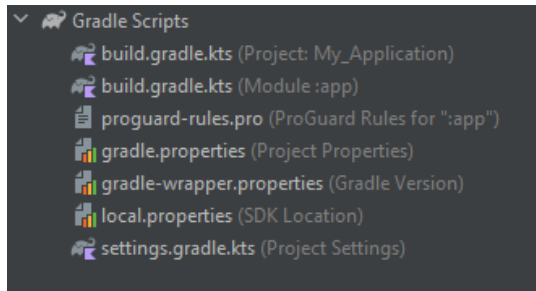


Folder ini berisikan berbagai resource yang dapat digunakan pada aplikasi. Beberapa resource pada aplikasi ini diantaranya adalah layout suatu halaman pada aplikasi, icons, style, dan lain-lain.

Berikut adalah penjelasan dari beberapa folder pada direktori /res

| Direktori | Keterangan |
|-----------|---|
| /drawable | Berisikan file-file bitmap (.png, .jpg, .gif) atau suatu file XML yang dapat dikompilasi menjadi sebuah gambar. |
| /layout | Berisikan file-file XML yang mendefinisikan layout dari suatu user interface. |
| /mipmap | Berisikan drawable files dalam beberapa ukuran. |
| /values | <p>Berisikan file-file XML yang mengandung beberapa value, seperti string, integer, colors, dll.</p> <p>Contoh file XML yang berada pada folder /values:</p> <ul style="list-style-type: none">• arrays.xml (berisikan arrays of resource)• colors.xml (berisikan nilai-nilai warna)• dimens.xml (berisikan nilai-nilai ukuran)• strings.xml (berisikan nilai-nilai string)• styles.xml (berisikan nilai-nilai tema/gaya) |

Gradle Scripts



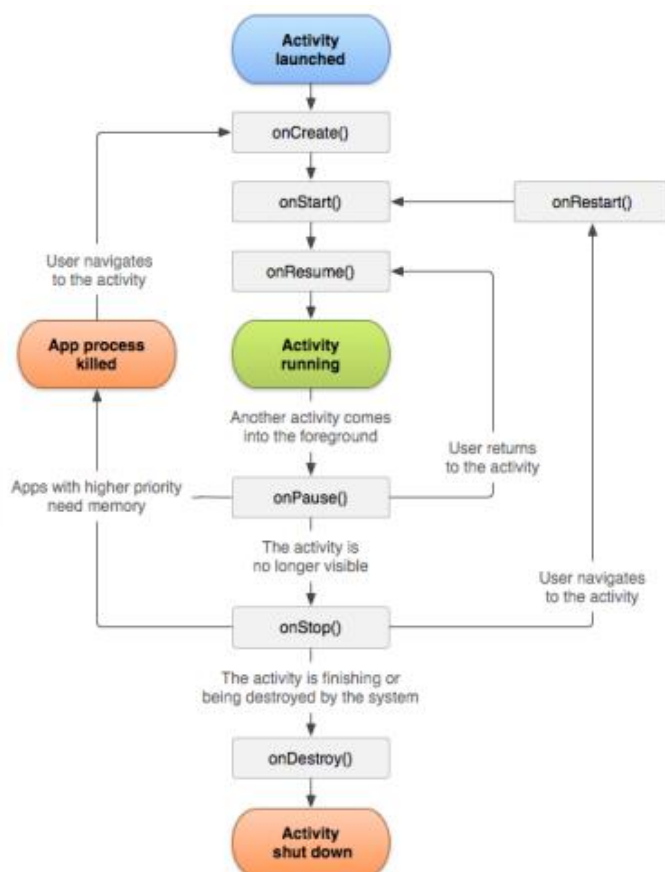
Android Studio menggunakan Gradle untuk mengotomasi proses dari software build dan proses-proses lainnya. Segala hal yang berkaitan dengan konfigurasi build akan dituliskan pada folder ini.

Activity

Activity merupakan suatu komponen yang penting pada aplikasi Android. Activity ini bertugas untuk menampilkan user interface ke layar pengguna. Maka dari itu, hampir semua activity akan berinteraksi dengan pengguna secara langsung.

Sebagai contoh, ketika kita membuat sebuah project aplikasi baru, kita akan mendapatkan dua berkas yaitu MainActivity dan activity_main.xml. Kelas MainActivity inilah yang akan bertanggung jawab untuk menampilkan view yang telah didefinisikan pada file activity_main.xml serta mengelola interaksi yang terjadi.

Activity Lifecycle



Activity pada aplikasi Android akan dikelola menggunakan *activity stacks*. Ketika suatu activity baru dimulai, ia akan berada pada stack paling atas menggantikan activity yang aktif sebelumnya. Lalu, apa yang akan terjadi pada activity sebelumnya? Ia akan memasuki state/lifecycle baru yaitu `onPause` dan `onStop` seperti gambar di samping. Proses-proses yang terjadi inilah yang disebut dengan **activity lifecycle**.

Ketika kita ingin mengembangkan aplikasi yang baik, kita harus memperhatikan lifecycle ini.

Contohnya: ketika suatu activity akan dimulai, kita bisa melakukan inisiasi koneksi ke database pada state `onCreate`, kemudian ketika activity memasuki state `onDestroy`, kita dapat memutuskan koneksi dengan database.

Secara umum, berikut adalah penjelasan dan pergantian dari lifecycle suatu activity:

| Method | Deskripsi | Next |
|-----------|---|--------------------------|
| onCreate | Dipanggil ketika activity dibuat untuk pertama kalinya. Pada method ini, kita dapat melakukan beberapa hal seperti membuat views, parsing data terhadap lists, dll. | onStart |
| onRestart | Dipanggil setelah activity dihentikan (onStop), sebelum dimulai kembali. | onStart |
| onStart | Dipanggil ketika activity akan ditampilkan kepada pengguna. | onResume atau onStop |
| onResume | Dipanggil ketika activity akan mulai berinteraksi dengan pengguna. Pada saat ini, activity telah berada pada posisi teratas dari activity stacks. | onPause |
| onPause | Dipanggil ketika activity tidak dapat berinteraksi dengan pengguna, tetapi masih terlihat oleh pengguna. Contohnya ketika membuka sebuah dialog. | onResume atau onStop |
| onStop | Dipanggil ketika activity sudah tidak terlihat oleh pengguna. Contohnya ketika pengguna membuka activity lain. | onRestart atau onDestroy |
| onDestroy | Merupakan method yang akan dipanggil sebelum activity benar-benar dihancurkan. | - |

Saving Activity State

Activity state merupakan suatu keadaan/state yang dimiliki oleh activity saat ini. Activity state biasanya meliputi informasi yang sedang ditampilkan kepada pengguna, masukan yang diberikan oleh pengguna, dll.

Ketika suatu activity memasuki lifecycle onDestroy, maka activity tersebut akan dihancurkan, termasuk juga state yang ada di dalamnya. Untuk menyimpan state yang ada pada suatu activity, kita dapat memanfaatkan metode onSaveInstanceState(). Pada metode tersebut, kita dapat memanfaatkan sebuah bundle untuk menyimpan suatu data yang diinginkan. Berikut adalah contoh penggunaan method onSaveInstanceState().

```
override fun onSaveInstanceState(outState: Bundle) {  
    super.onSaveInstanceState(outState)  
  
    val message: String = tvMessage.text.toString()  
    outState.putString(KEY_MESSAGE, message)  
}
```

Untuk menggunakan kembali data yang telah disimpan, kita dapat memanggil kembali bundle tersebut pada metode onCreate().

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView {  
        MyApplicationTheme {  
            setContentView(R.layout.activity_main)  
  
            val tvMessage = findViewById<TextView>(R.id.text)  
            if (savedInstanceState != null) {  
                val message = savedInstanceState.getString(KEY_MESSAGE)  
                tvMessage.text = message  
            }  
        }  
    }  
}
```

Latihan - Aplikasi Penghitung Volume

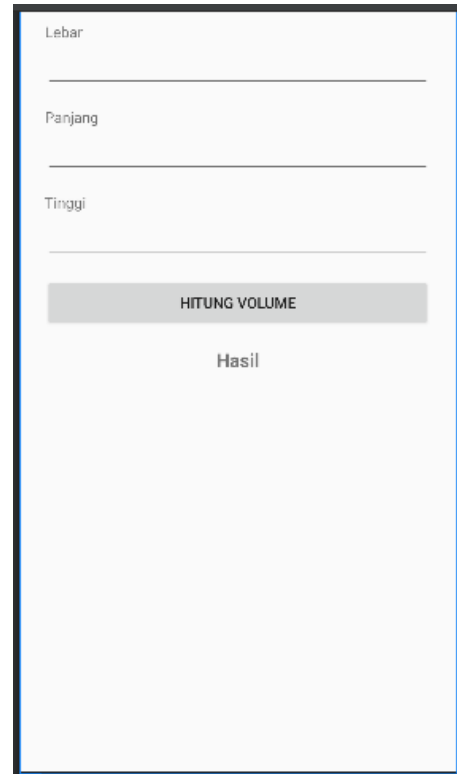
Pada latihan kali ini, kita akan membuat sebuah aplikasi penghitung volume balok sederhana. Gunakanlah project yang sudah dibuat sebelumnya dan ikutilah langkah-langkah berikut:

Layouting

Pada tahap pertama, kita akan mendesain layout halaman yang akan ditampilkan kepada pengguna. Hasil akhir yang akan didapatkan melalui latihan ini adalah seperti gambar disamping.

Terdapat beberapa komponen view yang akan kita gunakan, diantaranya adalah:

- TextView untuk menampilkan text 'Lebar', 'Panjang', 'Tinggi', dan 'Hasil'.
- EditText untuk tempat pengguna memberikan masukan.
- Button untuk melakukan proses perhitungan volume berdasarkan masukan yang diberikan pengguna.



Seluruh view tersebut akan dibungkus dengan LinearLayout sebagai viewgroup. Agar halaman dapat di-scroll, kita dapat membungkusnya lagi dengan ScrollView.

Setelah melakukan analisis terhadap desain, kita dapat melakukan implementasi desain tersebut pada file layout. Silakan buat file baru dengan nama **activity_volume.xml** pada folder /res/layout dan tuliskan kode berikut:

Penting: Hindari copy-paste, tulislah kembali kode sambil mengamati perubahan yang terjadi pada preview. Untuk mengaktifkan preview, pilih menu Split pada pojok kanan atas.

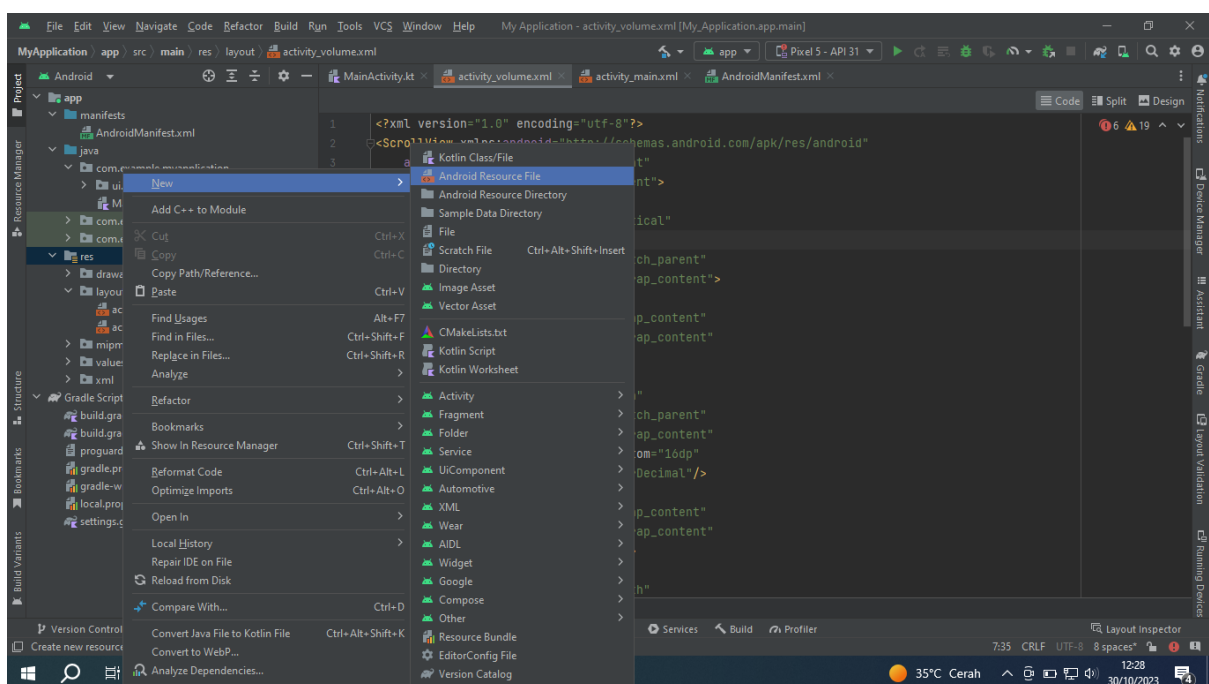
```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:orientation="vertical"
        android:padding="24dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Lebar" />
        <EditText
            android:id="@+id/et_width"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="16dp"
            android:inputType="numberDecimal"/>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Panjang" />
        <EditText
            android:id="@+id/et_length"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="16dp"
            android:inputType="numberDecimal" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Tinggi" />
        <EditText
            android:id="@+id/et_height"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="16dp"
            android:inputType="numberDecimal" />
        <Button
            android:id="@+id/btn_calculate"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="hitung volume" />

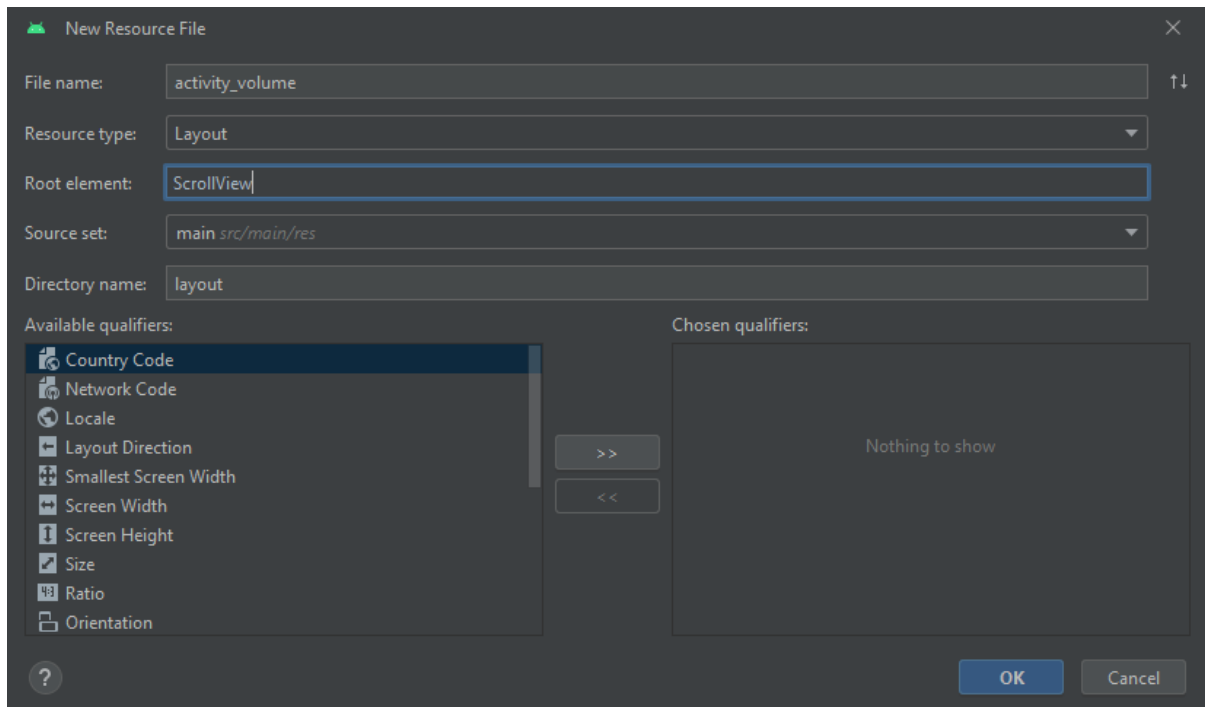
        <TextView
            android:id="@+id/tv_result"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="18dp"
            android:text="Hasil"
            android:textAlignment="center"
            android:textSize="18sp"
            android:textStyle="bold" />

    </LinearLayout>
</ScrollView>
```

Tips: Membuat file layout dapat menggunakan fitur GUI, yaitu dengan klik kanan pada folder res, kemudian pilih New - Android Resource File.

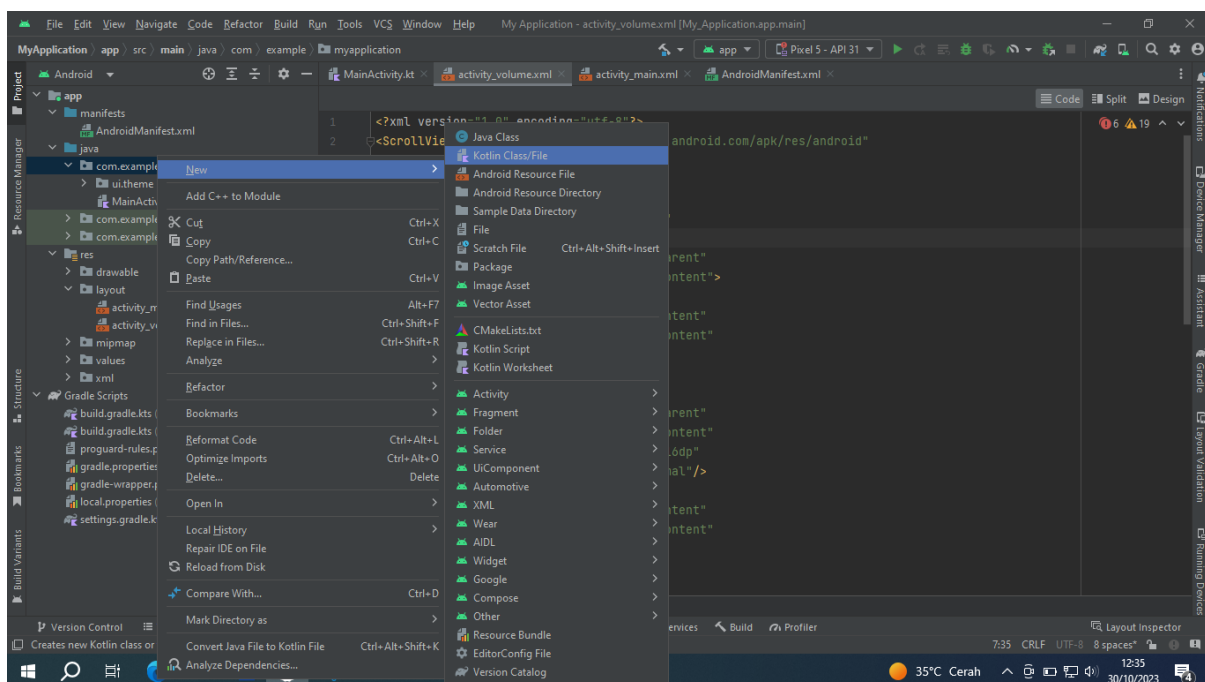


Selanjutnya sesuaikan File name, Resource type, dan Root element sebagaimana gambar berikut.



VolumeActivity

Untuk mengelola interaksi yang akan terjadi pada activity, kita dapat menuliskan program pada **VolumeActivity.kt** yang akan mengelola segala bentuk interaksi yang akan dilakukan oleh pengguna terhadap aplikasi.



Buat file baru pada package com.example.myapplication, dengan melakukan klik kanan pada package, lalu pilih New - Kotlin Class/File. Beri nama VolumeActivity dan pilih tipe File.

Pertama, kita akan melakukan deklarasi variabel yang akan menyimpan object dari view yang sudah kita definisikan pada **activity_volume.xml**. Tuliskan kode berikut di dalam kelas VolumeActivity sebagai global variable.

```
1  package com.example.myapplication
2
3  import android.os.Bundle
4  import android.widget.Button
5  import android.widget.EditText
6  import android.widget.TextView
7  import androidx.activity.ComponentActivity
8  import androidx.activity.compose.setContent
9  import com.example.myapplication.ui.theme.MyApplicationTheme
10
11 class VolumeActivity : ComponentActivity() {
12
13     private lateinit var tvResult: TextView
14     private lateinit var etWidth: EditText
15     private lateinit var etLength: EditText
16     private lateinit var etHeight: EditText
17     private lateinit var btnCalculate: Button
18
19     override fun onCreate(savedInstanceState: Bundle?) {
20         super.onCreate(savedInstanceState)
21         setContent {
22             MyApplicationTheme {
```

Kemudian, inisialisasi variabel tersebut dengan view yang bersesuaian. Inisialisasi ini dilakukan pada saat activity memasuki lifecycle onCreate.

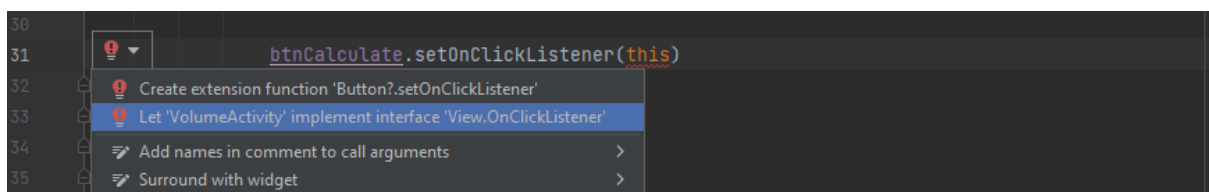
```
18
19     override fun onCreate(savedInstanceState: Bundle?) {
20         super.onCreate(savedInstanceState)
21         setContent {
22             MyApplicationTheme {
23                 setContentView(R.layout.activity_volume)
24
25                 tvResult = findViewById<TextView>(R.id.tv_result)
26                 etWidth = findViewById<EditText>(R.id.et_width)
27                 etLength = findViewById<EditText>(R.id.et_length)
28                 etHeight = findViewById<EditText>(R.id.et_height)
29                 btnCalculate = findViewById<Button>(R.id.btn_calculate)
30             }
31         }
32     }
33 }
```

Selanjutnya, tambahkan sebuah `onClickListener` pada `btnCalculate`, agar Ketika tombol tersebut ditekan, kita dapat mengeksekusi perintah yang diinginkan.

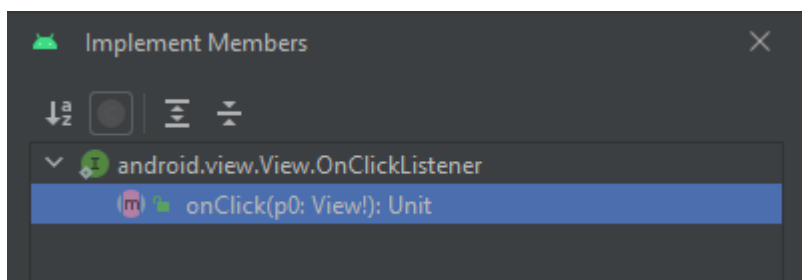
```
29         btnCalculate = findViewById<Button>(R.id.btn_calculate)
30
31         btnCalculate.setOnClickListener(this)
32     }
33 }
```

Akan terjadi error pada kode yang baru kita tambahkan. Error tersebut terjadi karena kita perlu mengimplementasi suatu interface pada kelas `VolumeActivity`. Untuk mengatasinya, kita dapat mengarahkan kursor ke arah icon lampu, kemudian pilih Let 'VolumeActivity' implement interface 'View.OnClickListener'.

```
30
31         btnCalculate.setOnClickListener(this)
32
33
34
35
```



Lalu, pilih method `onClick` untuk di-implementasi pada kelas `VolumeActivity`.



Setelah itu, kelas `VolumeActivity` akan mengimplementasi interface `View.OnClickListener` ditandai dengan perubahan sintaks pada deklarasi kelas `VolumeActivity`.

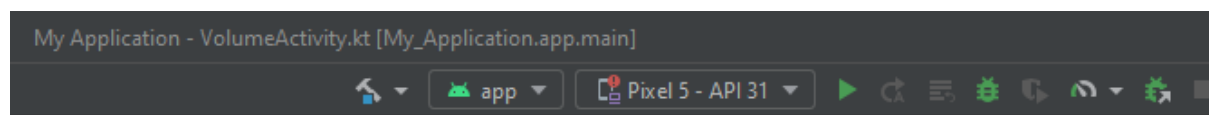
```
11
12 class VolumeActivity : AppCompatActivity(), View.OnClickListener {
13
14     private lateinit var tvResult: TextView
15     private lateinit var etWidth: EditText
16     private lateinit var etLength: EditText
17     private lateinit var etHeight: EditText
18     private lateinit var btnCalculate: Button
19
20     override fun onCreate(savedInstanceState: Bundle?) { ... }
21
22     override fun onClick(p0: View?) {
23         TODO( reason: "Not yet implemented")
24     }
25 }
```

Menuliskan Algoritma Menghitung Volume

Method `onClick` yang ditambahkan, akan dieksekusi ketika terdapat event 'click' yang dilakukan oleh pengguna terhadap suatu view. Karena kita akan menghitung volume balok ketika pengguna menekan button 'hitung' maka kita akan menuliskan algoritmanya pada fungsi `onClick` ini. Tambahkan source code berikut pada fungsi `onClick`:

```
37 override fun onClick(view: View?) {  
38     if (view != null) {  
39         // Perintah hanya akan dieksekusi jika btn_calculate ditekan  
40         if (view.id == R.id.btn_calculate) {  
41  
42             // Ambil nilai yang diberikan pengguna pada seluruh EditText  
43             val inputLength: String = etLength.text.toString().trim();  
44             val inputWidth: String = etWidth.text.toString().trim();  
45             val inputHeight: String = etHeight.text.toString().trim();  
46  
47             // Hitung volume balok  
48             val volume: Double = inputLength.toDouble() * inputWidth.toDouble() *  
49                 inputHeight.toDouble();  
50  
51             // Tampilkan hasil perhitungan ke TextView -> tvResult  
52             tvResult.text = String.format("Volume: %s", volume.toString());  
53         }  
54     }  
55 }
```

Sejauh ini, aplikasi kita sudah dapat dijalankan. Silakan hubungkan device Android atau aktifkan emulator. Kemudian klik Run 'app' pada pojok kanan atas Android Studio.



Sebelum menjalankan, ubah arah launcher menuju VolumeActivity pada Android Manifest.

```
15 <activity  
16     android:name=".MainActivity"  
17     android:label="@string/app_name"  
18     android:theme="@style/Theme.MyApplication">  
19 </activity>  
20 <activity  
21     android:name=".VolumeActivity"  
22     android:exported="true"  
23     android:label="@string/app_name"  
24     android:theme="@style/Theme.MyApplication">  
25     <intent-filter>  
26         <action android:name="android.intent.action.MAIN" />  
27         <category android:name="android.intent.category.LAUNCHER" />  
28     </intent-filter>  
29 </activity>
```

Memastikan Input Tidak Kosong

Apa yang terjadi ketika pengguna tidak memasukkan nilai apapun ke dalam EditText? Ya, aplikasi yang kita kembangkan akan mengalami force close. Untuk mengatasi hal ini, kita dapat melakukan improvement terhadap algoritma yang telah ditulis sebelumnya. Tambahkan validasi berikut untuk memastikan bahwa perhitungan hanya akan dilakukan ketika seluruh input telah diisi oleh pengguna.

```
45         val inputHeight: String = etHeight.text.toString().trim();  
46  
47         var isEmptyFields = false  
48         if (inputLength.isEmpty()) {  
49             isEmptyFields = true  
50             etLength.error = "Field ini tidak boleh kosong"  
51         }  
52         if (inputWidth.isEmpty()) {  
53             isEmptyFields = true  
54             etWidth.error = "Field ini tidak boleh kosong"  
55         }  
56         if (inputHeight.isEmpty()) {  
57             isEmptyFields = true  
58             etHeight.error = "Field ini tidak boleh kosong"  
59         }  
60  
61         if (!isEmptyFields) {  
62             // Hitung volume balok  
63             val volume: Double = inputLength.toDouble() * inputWidth.toDouble() *  
64                 inputHeight.toDouble();  
65  
66             // Tampilkan hasil perhitungan ke TextView -> tvResult  
67             tvResult.text = String.format("Volume: %s", volume.toString());  
68         }  
69     }
```

Menyimpan State dengan onSaveInstanceState

Selanjutnya, apa yang akan terjadi jika pengguna melakukan perubahan orientasi perangkat? Ketika terjadi perubahan orientasi, activity yang sedang ditampilkan akan dihancurkan terlebih dahulu, sebelum ditampilkan kembali kepada pengguna. Ketika activity dihancurkan, maka seluruh state yang ada juga akan ikut hilang, termasuk hasil perhitungan yang sudah dilakukan. Untuk mencegah hilangnya hasil perhitungan, kita dapat menggunakan method `onSaveInstanceState()`.

Tambahkan kode berikut pada kelas `VolumeActivity`. Kode ini, akan menyimpan state pada suatu Activity ke sebuah `Bundle`.

```
18     private lateinit var btnCalculate: Button
19
20     private val KEY_RESULT = "key_result"
21
22     override fun onCreate(savedInstanceState: Bundle?) {...}
23
24
25
26
27
28
29     override fun onClick(view: View?) {...}
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75     override fun onSaveInstanceState(outState: Bundle) {
76         super.onSaveInstanceState(outState)
77
78         val calculationResult = tvResult.text.toString()
79         outState.putString(KEY_RESULT, calculationResult)
80     }
81 }
```

Untuk memanggil state yang telah disimpan, kita dapat melakukannya pada method onCreate.

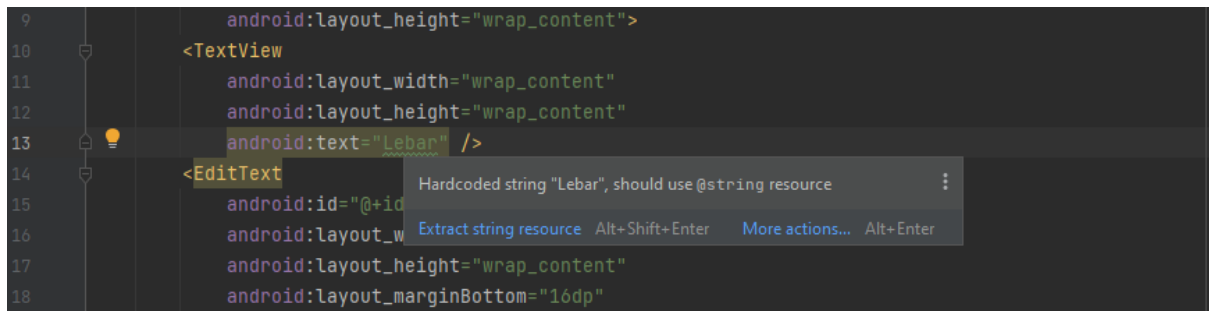
Tambahkan kode berikut:

```
22  override fun onCreate(savedInstanceState: Bundle?) {  
23      super.onCreate(savedInstanceState)  
24      setContentView {  
25          MyApplicationTheme {  
26              setContentView(R.layout.activity_volume)  
27  
28              tvResult = findViewById<TextView>(R.id.tv_result)  
29              etWidth = findViewById<EditText>(R.id.et_width)  
30              etLength = findViewById<EditText>(R.id.et_length)  
31              etHeight = findViewById<EditText>(R.id.et_height)  
32              btnCalculate = findViewById<Button>(R.id.btn_calculate)  
33  
34              btnCalculate.setOnClickListener(this)  
35  
36              if (savedInstanceState != null) {  
37                  val result = savedInstanceState.getString(KEY_RESULT)  
38                  tvResult.text = result  
39              }  
40          }  
41      }  
42  }
```

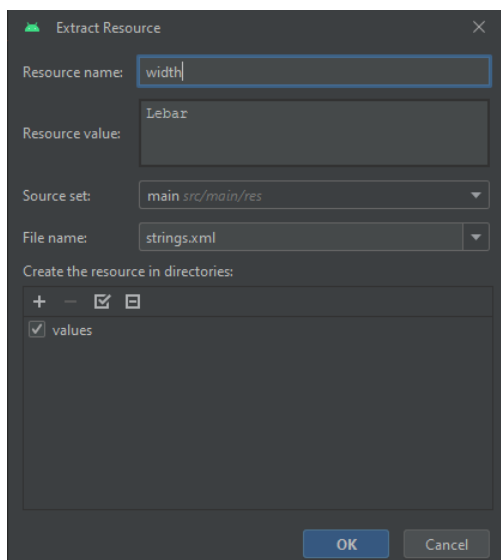
Improvement yang dapat dilakukan

Kembali ke file `activity_volume.xml`, terdapat beberapa warning yang diberikan oleh Android Studio. Kita akan memperbaiki beberapa warning tersebut untuk meningkatkan kualitas dari kode kita.

Hardcoded string “Lebar”, should use ‘@string’ resource



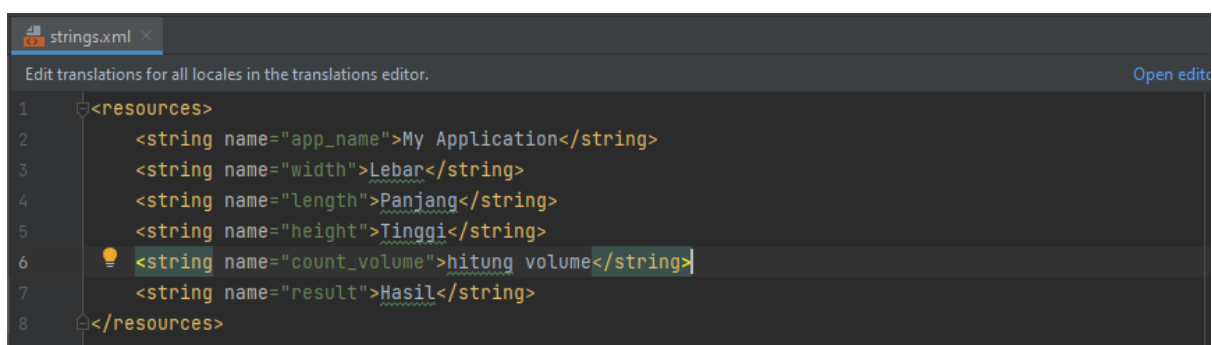
Maksud dari warning ini adalah, sebaiknya kita tidak menuliskan teks secara hardcoded pada atribut android:text. Kita diminta untuk menggunakan resource. Untuk memperbaiki hal ini, kita dapat menekan pilihan ‘Extract string resource’.



Setelah itu akan muncul sebuah jendela ‘Extract Resource’, lalu ubah nilai pada Resource name menjadi width.

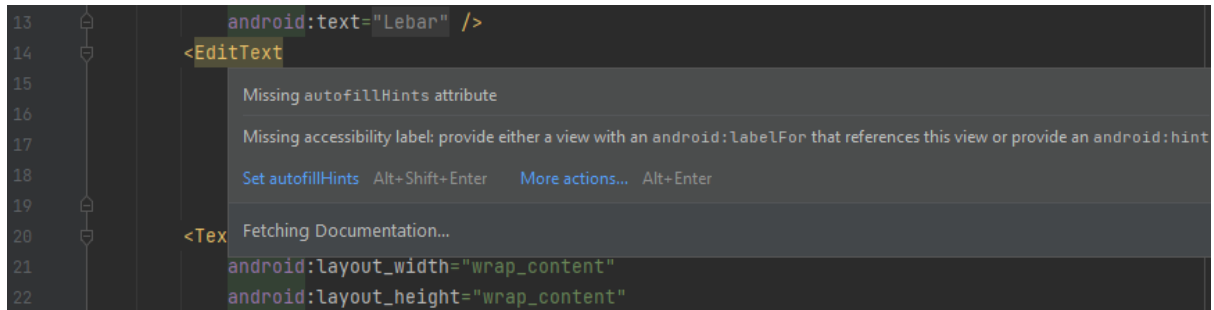
Ulangi untuk seluruh view yang masih menggunakan hardcoded string untuk atribut android:text.

Proses tersebut akan menambahkan string baru ke dalam resource. Untuk melihatnya silakan buka file strings.xml pada folder res/values.



Missing 'autofillHints' attribute

Atribut autofillHints dapat digunakan untuk mendeskripsikan nilai yang perlu dimasukkan ke dalam EditText, sehingga layanan autofill dapat bekerja secara optimal. Untuk mengatasi warning ini, kita dapat menambahkan atribut android:autofillHints dengan value yang sesuai.



Sebagai contoh, kita dapat menambahkan value 'height' pada atribut autofillHints, sehingga layanan autofill dapat mengenali bahwa EditText ini menerima nilai 'height' atau tinggi dari suatu balok.



Missing accessibility label: provide either a view with an 'android:labelFor' that reference this view or provide an 'android:hint'

Maksud dari warning ini adalah, suatu edit text membutuhkan sebuah label atau hint agar tidak membuat pengguna kebingungan harus memasukkan nilai seperti apa ke dalam EditText. Warning ini bertujuan untuk meningkatkan accessibility, sehingga pengguna yang memiliki disabilitas dapat memanfaatkan fitur accessibility dengan baik.

Karena kita menggunakan sebuah TextView untuk menandai suatu EditText, kita dapat menambahkan atribut android:labelFor pada masing-masing TextView.

```
11 <TextView
12     android:layout_width="wrap_content"
13     android:layout_height="wrap_content"
14     android:text="Lebar"
15     android:labelFor="@+id/et_width" />
16 <EditText
17     android:id="@+id/et_width"
18     android:layout_width="match_parent"
19     android:layout_height="wrap_content"
20     android:layout_marginBottom="16dp"
21     android:inputType="numberDecimal"
22     android:autofillHints="width" />
23 <TextView
24     android:layout_width="wrap_content"
25     android:layout_height="wrap_content"
26     android:text="Panjang"
27     android:labelFor="@+id/et_length" />
28 <EditText
29     android:id="@+id/et_length"
30     android:layout_width="match_parent"
31     android:layout_height="wrap_content"
32     android:layout_marginBottom="16dp"
33     android:inputType="numberDecimal"
34     android:autofillHints="length" />
```

Tugas

1. Tambahkan fitur untuk menghitung luas permukaan dan keliling balok pada project aplikasi yang sudah dikerjakan pada latihan sebelumnya. Tampilkan hasil perhitungan pada tv_result.
2. Tuliskan laporan praktikum yang berisi penjelasan source code fitur menghitung luas permukaan dan keliling balok yang diimplementasikan pada project.
3. Export project dengan menggunakan menu File > Export > Export to ZIP File, kemudian kumpulkan pada link pengumpulan yang sudah ditentukan.

The screenshot shows a mobile application interface with a light gray background. It features three input fields for dimensions: 'Lebar' (Width), 'Panjang' (Length), and 'Tinggi' (Height), each with a corresponding horizontal line for text entry. Below these fields are three blue buttons with white text: 'HITUNG VOLUME', 'HITUNG LUAS PERMUKAAN', and 'HITUNG KELILING'. At the bottom of the interface is a label 'Hasil' (Result) in a bold, dark gray font.

Referensi Tugas

-- Selamat Belajar --