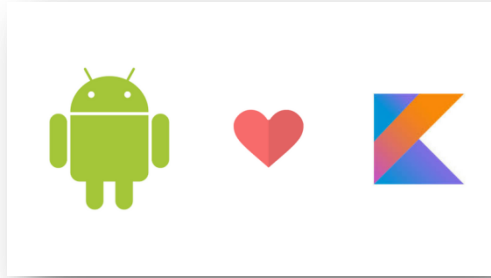


Android App Development with Kotlin



Praktikum 4

Tujuan :

Mahasiswa mengenal dan mengimplementasikan :

- Android App with Kotlin
- Android Extension
- Event Handling
- Intents
- Menu
- Fragments
- Background Task
- Android Coroutine

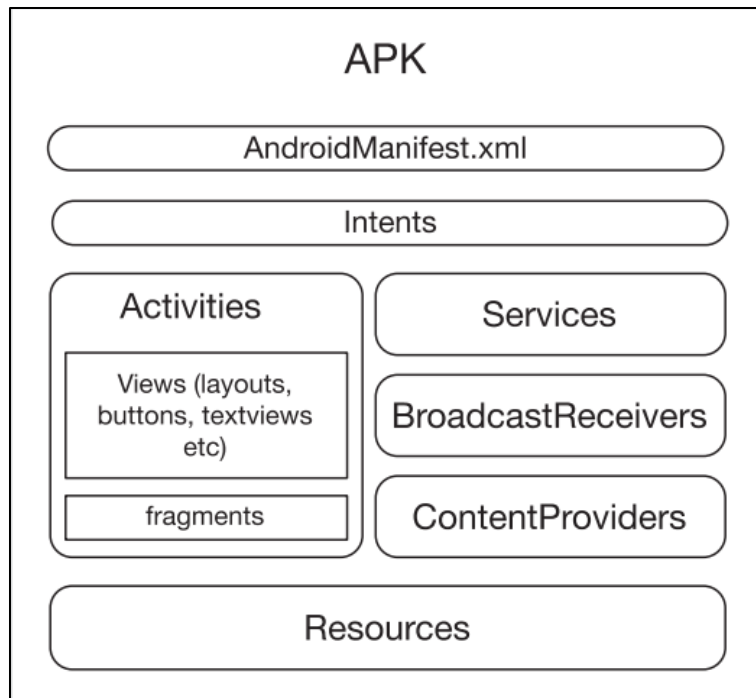
Hasil Praktikum :

- Contoh program praktikum.

Android App with Kotlin

Struktur Aplikasi Android

Sebuah aplikasi Android bukanlah berbentuk paket monolitik seperti sebuah file EXE di Windows. Ia adalah sebuah gabungan komponen secara bebas yang dipasang dengan sumber daya lain dan mereka diikat bersama didalam sebuah *file* Android Package atau APK. Berikut adalah struktur logika sebuah aplikasi berbasis Android yang dibangun.



Activities, *Services*, *BroadcastReceivers*, dan *ContentProviders* dipanggil sebagai *Android components*. Mereka adalah kunci untuk membangun sebuah aplikasi. Mereka adalah suatu abstraksi tingkat tinggi yang berguna untuk menunjukkan tampilan kepada pengguna, menjalankan tugas tertentu di latar belakang, menyiarkan *event* dari suatu aplikasi yang tertarik untuk merespon aplikasi tersebut, dan lain-lain. Komponen adalah kelas *pre-coded* atau *pre-built* dengan tingkah laku (*behavior*) yang spesifik, dan kita menggunakannya didalam aplikasi kita dengan meng-*extends* mereka, jadi kita dapat menambahkan *behavior* yang akan menjadi keunikan aplikasi kita.

Activity : adalah suatu wadah yang disediakan untuk meletakkan sesuatu yang dapat dilihat oleh pengguna. Ia berfokus kepada apa yang dapat pengguna lakukan. Sebagai contoh: sebuah *activity* harus memungkinkan pengguna untuk dapat menampilkan sebuah *form* isian *email*. Didalam *activity* ada Views dan Fragments. View adalah kelas yang digunakan ntuk menggambarkan konten kedalam layar. Contoh View adalah TextViews, Button dan lain-lain. Sebuah *fragment* sama dengan sebuah *activity* ia juga sebuah unit komposisi tetapi ia adalah bagian kecilnya.

Services : adalah kelas yang mengizinkan kita untuk menjalankan logika program tanpa menghentikan atau membekukan antarmuka yang sedang berjalan. *Service* adalah kode yang berjalan dilatar belakang; ia sangat berguna ketika aplikasi kita harus mengunduh *file* dari sebuah web, atau mungkin menjalankan musik dilatar belakang.

BroadcastReceiver : mengizinkan aplikasi kita untuk menerima pesan tertentu dari salah satu sistem Android atau aplikasi lainnya. Aplikasi kita dapat mengirim dan menyiarkan pesan ke sistem yang lebih luas. Contoh penerapannya ialah menampilkan pesan peringatan ketika daya *battery* dibawah 10%.

ContentProviders : mengizinkan kita membuat aplikasi yang mungkin bisa membagikan data ke aplikasi lain. Ia mengatur akses beberapa urutan dari repositori pusat data. Beberapa *ContentProviders* memiliki UI yang dimilikinya tetapi beberapa yang lain ada yang tidak. Contoh aplikasi *pre-built Contacts* didalam Android.

Android Application Entry Point

Sebuah aplikasi sederhana yang menampilkan layar ke pengguna paling tidak memiliki tiga unsur yaitu, pertama ia memerlukan sebuah kelas *activity* yang berperan sebagai *file* program utama, kedua sebuah *file layout* yang berisi semua UI yang didefenisikan, dan ketiga sebuah *file Manifest*, yang menghubungkan semua konten *project* secara bersamaan.

Ketika sebuah aplikasi dijalankan, Android *runtime* membuat objek *Intent* dan memeriksa file *Manifest*. Ia melihat nilai spesifik dari suatu node yang mendeklarasikan *intent-filter*; Android *runtime* mencoba melihat jika aplikasi memiliki sebuah *entry point* yang didefenisikan, seperti halnya fungsi utama “*main function*” pada kebanyakan program. Berikut adalah contoh kutipan dari *file* Android Manifest.

```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

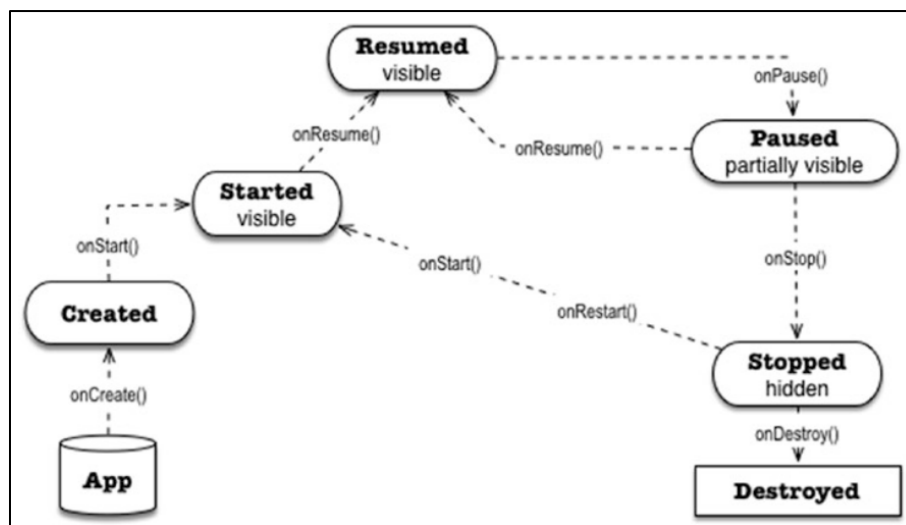
Baris kode diatas bertujuan untuk mendefenisikan sebuah *activity* dengan dengan nama MainActivity dimana didalam pendefenisian *activity* MainActivity tersebut terdapat deklarasi *intent-filter* yang menunjukkan bahwa *activity* tersebut sebagai *entry point* dari aplikasi. Artinya ketika aplikasi pertama kali dijalankan maka *activity* tersebut akan tampil untuk pertama kalinya.

Activity Class

Kelas MainActivity bertanggung jawab sebagai inisiasi berinteraksi dengan pengguna. Ia adalah sebuah kelas di Kotlin, dan didalamnya kita dapat melakukan :

- Memilih UI (layout) mana yang digunakan. Ketika kita memanggil fungsi `setContentView(R.layout.activity_main)` didalam suatu *activity*, ia akan mengikat sebuah *activity* dengan sebuah *layout* dalam file xml. Proses ini disebut dengan “**Layout binding**”. Ketika *activity* mengikat *layout*, maka layar akan mengisi elemen antarmuka yang dapat disentuh atau digeser oleh pengguna.
- Memperoleh referensi untuk objek *View*. Objek *View* sering juga disebut *widgets* atau *control*. Ketika kita memiliki suatu referensi *programmatic* ke objek *view*, kita dapat memanipulasinya, merubah propertinya, atau menghubungkannya dengan *event*. Proses ini dipanggil “**View binding**”.

Kelas *activity* adalah turunan dari `androidx.appcompat.app.AppCompatActivity`. Ia adalah anak dari `FragmentActivity` yang diturunkan dari kelas `androidx.app.Activity`. Ketika Android *runtime* menjalankan aplikasi yang pada akhirnya menjalankan sebuah *activity*, ia membuat rekaman apa saja yang terjadi pada suatu *activity*. Setiap *activity* memiliki sebuah siklus hidup, dan setiap kejadian pada suatu siklus hidup memiliki sebuah fungsi yang berhubungan yang dapat kita gunakan untuk mengkustomisasi tingkah laku dari aplikasi. Berikut adalah siklus hidup sebuah *activity*.



Sumber : Hagos T, 2018

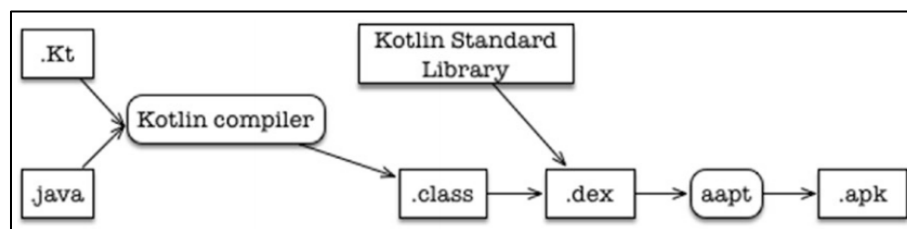
Ketika aplikasi berjalan saat *runtime*, ia memanggil fungsi `onCreate()` di kelas `MainActivity`, yang membawa status *activity* menjadi **Created**. Kita dapat menggunakan fungsi ini untuk menentukan inisialisasi *routines* seperti mempersiapkan kode *event-handling*, dan lain-lain.

Proses Activity selanjutnya yaitu **Started**. Activity tampak ke pengguna pada titik ini, tetapi belum siap untuk berinteraksi. Status selanjutnya ialah **Resumed**; status ini dimana aplikasi sedang berinteraksi dengan pengguna.

Jika pengguna meng-klik apapun yang mungkin menjalankan *activity* yang lain, *Android runtime* akan menjeda (*pause*) *activity* saat ini dan ia akan masuk kedalam status **Paused**. Dari sini, jika pengguna kembali ke *activity* semula, maka fungsi `onResume()` dipanggil dan *activity* akan dijalankan kembali. Disisi lain, jika pengguna memutuskan untuk membuka aplikasi yang berbeda, maka *Android runtime* mungkin merubah status menjadi berhenti atau **Stopped** dan pada akhirnya menghancurkan “*destroy*” aplikasi.

Proses Kompilasi Android

Gambar berikut menunjukkan representasi logika dari proses kompilasi Android.



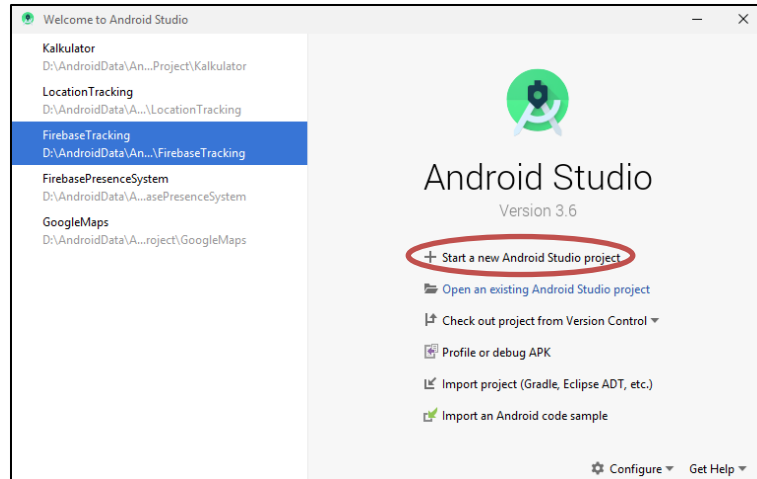
Sumber : Hagos T, 2018

Kompiler Kotlin mentransformasikan *file* sumber program kedalam Java *byte codes*. Hasil dari *byte code* dikombinasikan dengan *library standard* Kotlin kedalam bentuk *file* DEX. Sebuah *file* DEX berjenis *Executable Dalvik*, ia dalam format yang bisa dimengerti dan dieksekusi oleh Android Runtime (ART). Sebelum file DEX dan sumber daya lainnya dibungkus kedalam sebuah Android Package (APK), efek khususnya yaitu juga menghasilkan sebuah *file* dengan nama “`R.class`”. Kita menggunakan `R.class` untuk memperoleh referensi elemen UI yang didefinisikan didalam *file layout*.

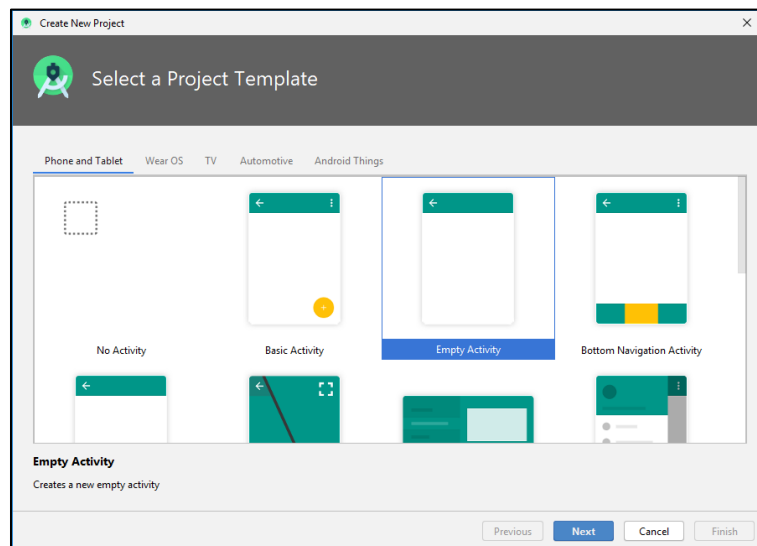
Membuat Aplikasi Android Pertama Menggunakan Kotlin

Pastikan anda telah menggunakan Android Studio versi 3 keatas. Langkah-langkah untuk membuat aplikasi `AndroidKotlinApp` adalah sebagai berikut :

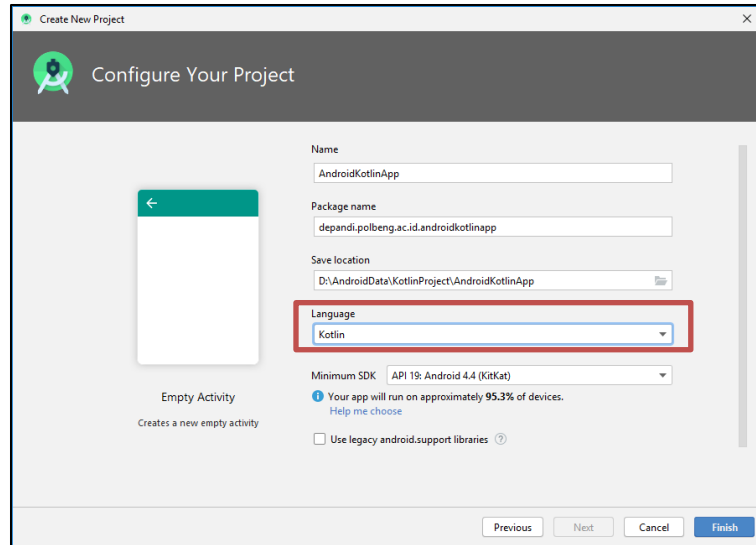
1. Klik + Start a new Android Studio Project pada form Welcome to Android Studio



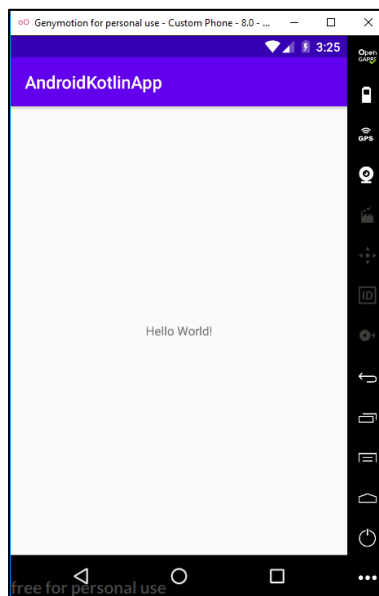
2. Klik **Empty Activity** pada form *Select a Project Template* dan klik **Next**.



3. Pada form konfigurasi *project* masukkan nama aplikasi seperti pengaturan atau konfigurasi berikut, nama paket dan lokasi penyimpanan silahkan disesuaikan dengan keinginan anda. Pada pilihan bahasa pemrograman pilihlah Kotlin. Minimum SDK yaitu API 19 : Android 4.4 (Kitkat). Klik **Finish** untuk melanjutkan pembuatan *project* anda.



4. Tunggu proses pembuatan *project* selesai. Jika proses pembuatan *project* telah selesai, silahkan jalankan *project* aplikasi anda di *emulator* Android.



Android Extension

Android *extension* mengizinkan kita untuk menyederhanakan kode *view-binding* dengan menunjukkan *property* dan fungsi dari elemen-elemen *view*. Artinya, kita tidak perlu menggunakan fungsi `findViewById()` untuk melakukan proses *view-binding*.

Tahukah anda fungsi `findViewById()`. Fungsi ini tugasnya untuk memilih komponen yang ada di *layout* untuk dijadikan objek pada Java. Mirip seperti fungsi `getElementById()` pada Javascript.

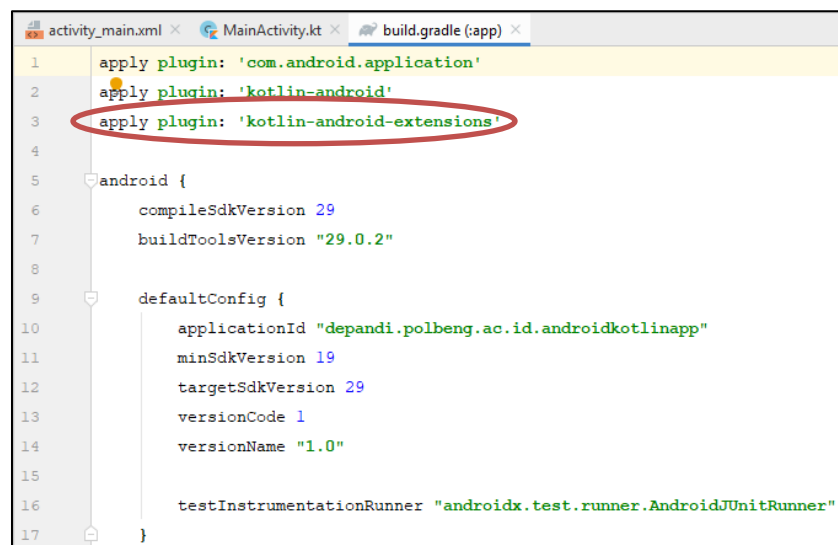
Pada Kotlin, kita tidak lagi menggunakan fungsi `findViewById()`. Karena menurut mereka (Dok. Kotlin): ***Fungsi tersebut berpotensi menimbulkan bugs dan membuat kode jadi susah dibaca.***

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    edtWidth = findViewById(R.id.edt_width);
    edtHeight = findViewById(R.id.edt_height);
    edtLength = findViewById(R.id.edt_length);
    btnCalculate = findViewById(R.id.btn_calculate);
    tvResult = findViewById(R.id.tv_result);
    btnHapus = findViewById(R.id.btn_hapus);

    edtLength.setText("50");
}
```

Kotlin memberikan terobosan dengan *plugin* *Android Extensions*, sehingga pengembang tidak lagi menghabiskan energi dan waktunya untuk mengetik fungsi `findViewById()`. Pada Android Studio versi 2 *plugin* *Android Extensions* ini perlu di tambahkan di *project* Android Kotlin anda, akan tetapi jika menggunakan Android Studio versi 3 keatas *plugin* *Android Extensions* sudah otomatis ditambahkan pada saat pembuatan project.



Implementasi Android Extension pada View Binding

1. Editlah file *strings.xml* pada folder *values* di Project *AndroidKotlinApp*, tambahkan beberapa variabel dan nilai string sebagai berikut:

```
<resources>
    <string name="app_name">AndroidKotlinApp</string>
    <string name="number">0</string>
</resources>
```



```
<string name="add_number">Add Number</string>
</resources>
```

2. Editlah file *activity_main.xml* pada folder *layout*. Sehingga kodenya menjadi seperti berikut :

```
<?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:gravity="center_vertical|center_horizontal"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tvCounter"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/number"
        android:textAlignment="center"
        android:textSize="30sp"
        android:textStyle="bold" />

    <Button
        android:id="@+id/btnAddNumber"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="@string/add_number"
        android:onClick="addNumber"/>

</LinearLayout>
```

3. Bukalah file sumber *MainActivity.kt*, editlah kode program hingga menjadi seperti berikut :

```
package depandi.polbeng.ac.id.androidkotlinapp

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import kotlinx.android.synthetic.main.activity_main.*

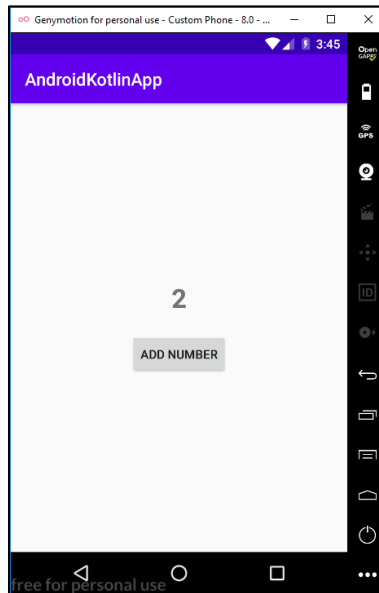
class MainActivity : AppCompatActivity() {

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

        tvCounter.text = "1"
    }

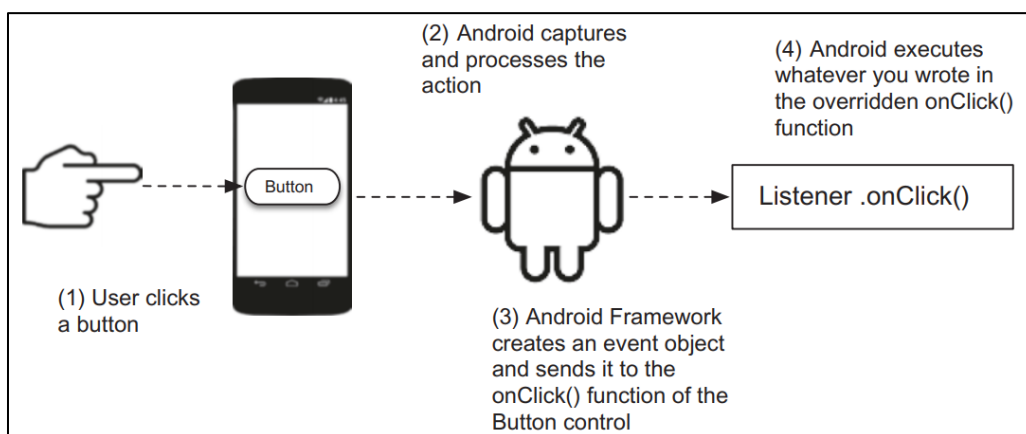
    fun addNumber(v: View) {
        val currVal = tvCounter.text.toString().toInt()
        val nextVal = currVal + 1
        tvCounter.text = nextVal.toString()
    }
}
```

4. Jalankan program di Emulator Android. Perhatikan hasilnya dan lakukan analisa terhadap kode layout *activity_main.xml* dan kode logika program *ActivityMain.kt* dilaporan praktikum.



Event Handling

Pengguna berinteraksi dengan aplikasi melalui sentuhan, klik, menyapu layar, atau mengetik sesuatu. Kerangka kerja Android menangkap, menyimpan, memproses, dan mengirim aksi tersebut ke aplikasi sebagai objek *event*. Kita dapat merespon kejadian dengan menulis fungsi khusus yang dirancang untuk menangani hal tersebut. Fungsi yang menangani kejadian ditulis didalam objek *listener*. Berikut adalah model sederhana bagaimana interaksi penggunaa ditangani oleh kerangka kerja Android.



Ketika pengguna melakukan suatu aksi sentuhan dengan cara meng-klik sebuah *button*, kerangka kerja Android menangkap aksi tersebut dan memasukkannya kedalam sebuah objek *event*.

Sebuah objek *event* berisi data tentang aksi pengguna (seperti yang mana *button* di klik, dimana lokasi *button* yang diklik, dan lain-lain) Android mengirim objek *event* ini ke aplikasi dan ia akan memanggil fungsi khusus yang sesuai dengan aksi pengguna. Jika pengguna meng-klik *button*, Android akan memanggil fungsi `onClick()` pada objek *Button*, jika pengguna meng-klik *button* yang sama tetapi di pegang dengan waktu yang sedikit lama, maka akan memanggil fungsi `onLongClick()`. Objek *View* seperti *Button*, dapat berhubungan ke jangkauan lebih besar dari *events* seperti klik, *keypresses*, *touch* atau *swipe* dan lainnya.

Implementasi *onClick listener* di Java

```
button.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        System.out.println("Hello click");  
    }  
});
```

Implementasi *onClick listener* di Kotlin

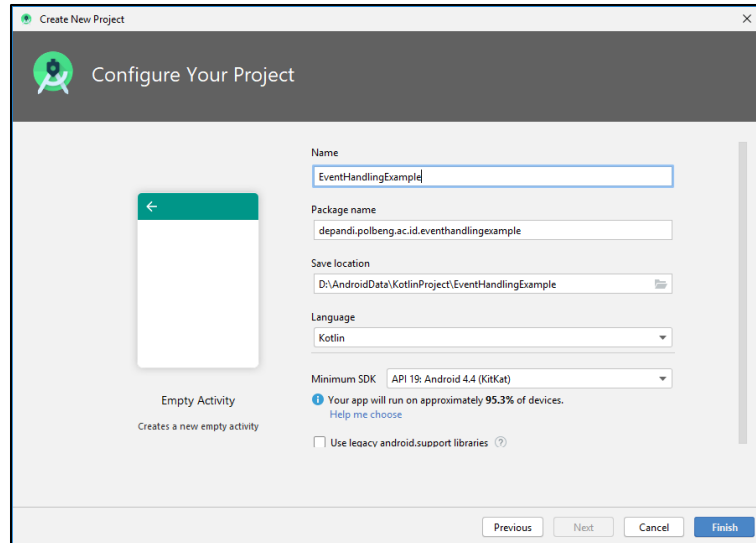
```
button.setOnClickListener(object: View.OnClickListener {  
    override fun onClick(v: View?) {  
        println("Hello click")  
    }  
})
```

Kotlin mendukung ekspresi lambda untuk menyederhanakan fungsi *onClick listener*, sehingga kode program diatas dapat disederhanakan menjadi seperti berikut :

```
button.setOnClickListener {  
    println("Hello click ")  
}
```

Implementasi Fungsi onClick Listener

1. Buatlah *project* baru dengan konfigurasi seperti berikut :



- Setelah selesai membuat project baru, selanjutnya editlah file *strings.xml* pada folder *values*, tambahkan beberapa variabel dan nilai *string* sebagai berikut:

```
<resources>
  <string name="app_name">EventHandlingExample</string>
  <string name="welcome_message">Welcome Message</string>
  <string name="display_message">Display Message</string>
</resources>
```

- Editlah file *activity_main.xml* sehingga kode *layout* menjadi seperti berikut :

```
<?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:id="@+id/root_layout"
  android:gravity="center_vertical|center_horizontal"
  android:orientation="vertical"
  tools:context=".MainActivity">

  <TextView
    android:id="@+id/tvMessage"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/welcome_message"
    android:textAlignment="center"
    android:textSize="30sp"
    android:textStyle="bold" />

  <Button
    android:id="@+id/btnDisplayMessage"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:text="@string/display_message" />
</LinearLayout>
```

- Bukalah file sumber *MainActivity.kt* editlah kode program hingga menjadi seperti berikut :

```

package depandi.polbeng.ac.id.eventhandlingexample

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Toast
import kotlinx.android.synthetic.main.activity_main.*

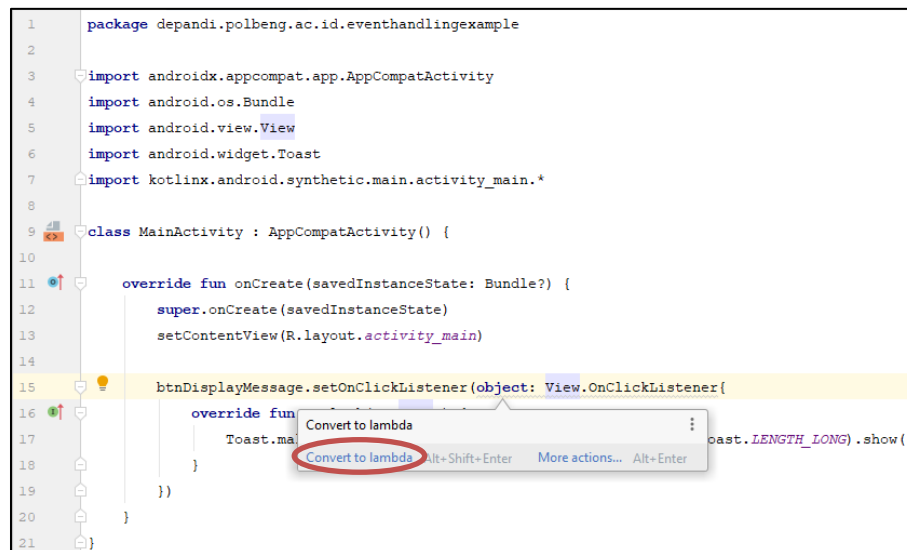
class MainActivity : AppCompatActivity() {

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

        btnDisplayMessage.setOnClickListener(object: View.OnClickListener{
            override fun onClick(v: View?) {
                Toast.makeText(applicationContext, "Hello World",
                Toast.LENGTH_LONG).show()
            }
        })
    }
}

```

5. Langkah berikutnya ialah cobalah untuk menyederhanakan fungsi `setOnClickListener` dengan mengarahkan *mouse* pada bagian kode dengan latar belakang berwarna kuning perhatikan *popup* yang muncul dan klik pada *Convert to Lambda*.



6. Jika berhasil maka fungsi `setOnClickListener` akan berubah secara otomatis menjadi fungsi *lambda* seperti berikut:

```

package depandi.polbeng.ac.id.eventhandlingexample

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Toast
import kotlinx.android.synthetic.main.activity_main.*

```

```

class MainActivity : AppCompatActivity() {

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

        btnDisplayMessage.setOnClickListener {
            Toast.makeText(applicationContext, "Hello World",
                Toast.LENGTH_LONG).show() }
        }
    }
}

```

7. Selanjutnya tambahkan satu buah fungsi lagi yang menangani *event* onLongClickListener sehingga kode program menjadi seperti berikut :

```

package depandi.polbeng.ac.id.eventhandlingexample

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Toast
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

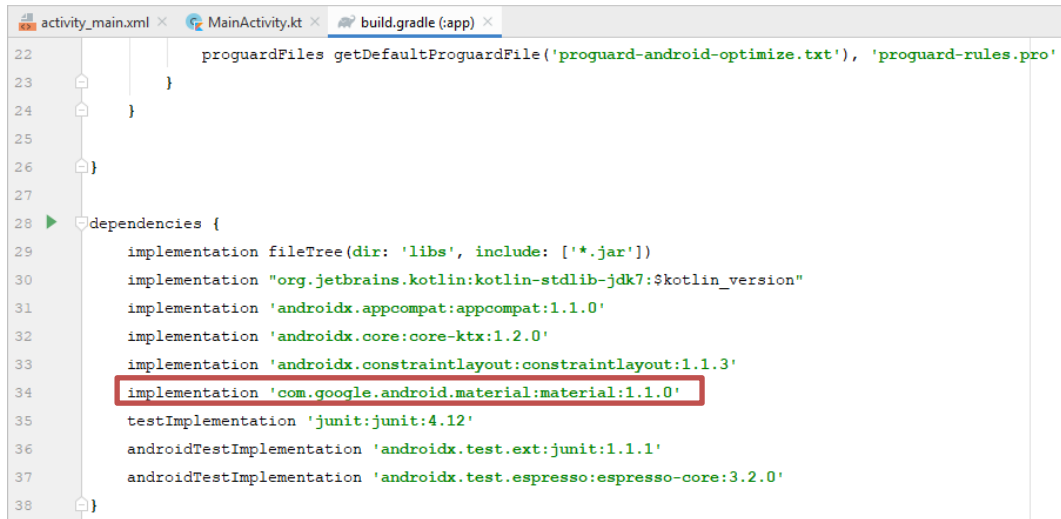
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        btnDisplayMessage.setOnClickListener {
            Toast.makeText(applicationContext, "Hello World",
                Toast.LENGTH_LONG).show() }

        btnDisplayMessage.setOnLongClickListener(object:
            View.OnLongClickListener {
                override fun onLongClick(v: View?): Boolean {
                    return true
                }
            })
    }
}

```

8. Silahkan rubah sendiri fungsi setOnLongClickListener menjadi ekspresi *lambda*.
9. Tambahkan library material design (*com.google.android.material:material:1.1.0*) yang diberi kotak berwarna merah berikut pada file **build.gradle(Module: app)**. Klik **Sync Now** untuk menerapkan penambahan *library* dalam *project*.



10. Edit kembali file *MainActivity.kt* sehingga kode program akhir menjadi seperti berikut :

```

package depandi.polbeng.ac.id.eventhandlingexample

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import com.google.android.material.snackbar.Snackbar
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

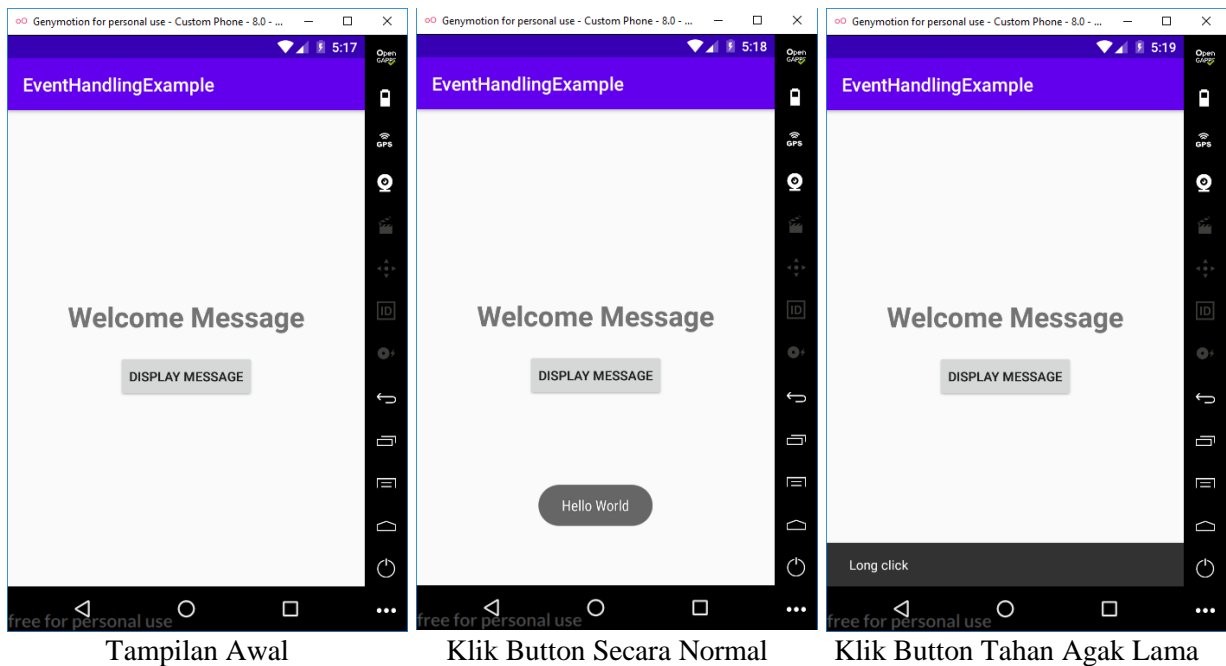
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        btnDisplayMessage.setOnClickListener {
            Toast.makeText(applicationContext, "Hello World",
                Toast.LENGTH_LONG).show()
        }

        btnDisplayMessage.setOnLongClickListener {
            Snackbar.make(root_layout, "Long click",
                Snackbar.LENGTH_LONG).show()
            true
        }
    }
}

```

11. Jalankan program di Emulator Android. Perhatikan hasilnya dan lakukan analisa terhadap kode layout *activity_main.xml* dan kode logika program *ActivityMain.kt* dilaporan praktikum.



Intents

Android menggunakan *Intens* sebagai cara untuk komponen-komponennya berkomunikasi, ia digunakan untuk mengirim pesan diantara komponen (*activity*). *Intent* adalah sebuah deskripsi abstrak dari operasi yang dilakukan. *Activity* bukanlah sebuah objek yang sederhana, ia adalah sebuah komponen. Kita tidak bisa mengaktifkan sebuah komponen hanya dengan menginstansiasinya seperti kita membuat objek seperti biasanya. Untuk menjalankan sebuah *activity*, kita perlu membuat objek *Intent* dan menjalankannya menggunakan fungsi **startActivity()**. Berikut adalah potongan kode program untuk mengaktifkan *activity* lain pada suatu *activity*.

```
button.setOnClickListener {
    val intent = Intent(this@MainActivity, SecondActivity::class.java)
    startActivity(intent)
}
```

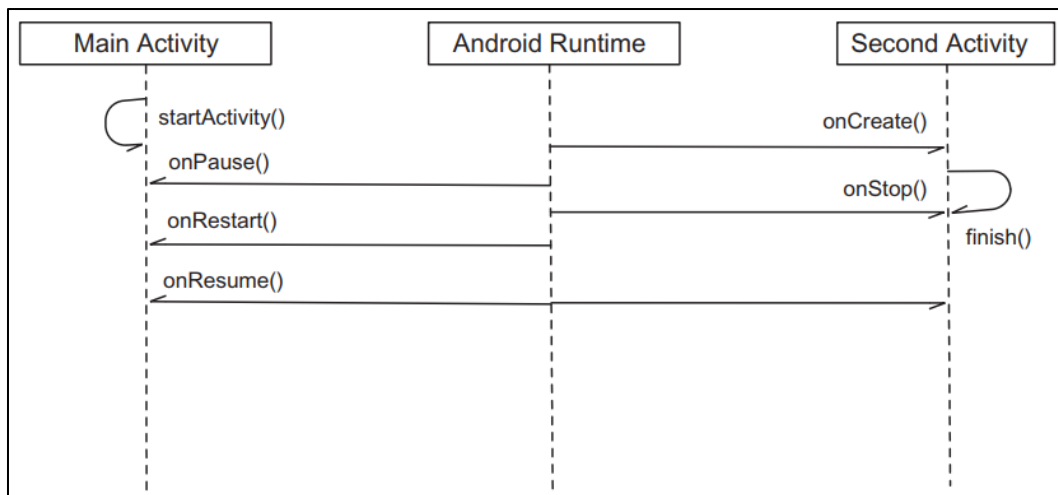
this@MainActivity : adalah parameter pertama dari konstruktor *Intent* adalah konteks dari objek. Kita mengirim *this@MainActivity* karena kelas *activity* adalah *subclass* dari *context*, jadi kita bisa menggunakannya. Alternatif lain, kita juga bisa menggunakan fungsi **getApplicationContext()**.

SecondActivity::class.java : adalah parameter kedua dari objek kelas. Ia adalah komponen kelas yang akan kita kirim pesan. Kita menjalankan sebuah *Activity* dengan memanggil fungsi **startActivity()** dan mengirim objek *Intent* kepadanya.

Terdapat dua jenis *Intent* yaitu implisit dan eksplisit. Implisit *Intent* sangat berguna karena memungkinkan aplikasi kita untuk mengambil keuntungan dari aplikasi lainnya. Aplikasi kita dapat menggunakan fungsioanalitas *Intent* tanpa harus menulis kodenya. Sebagai contoh : membuat sebuah *Intent* untuk membuka kamera, mengambil gambar, dan menyimpan gambar tanpa harus menulis kode kamera secara spesifik.

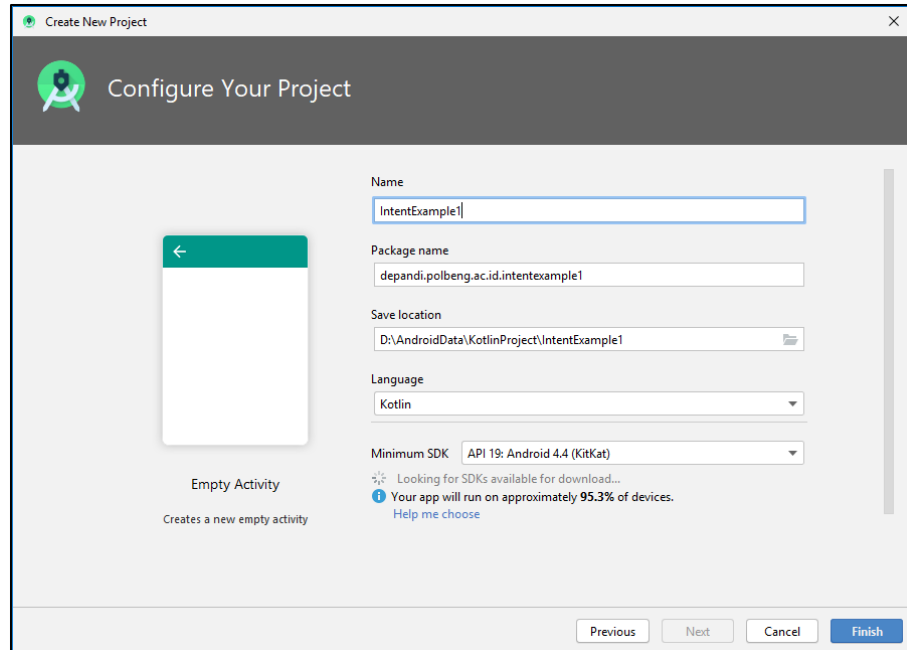
Implementasi berpindah dari satu activity ke activity yang lain

Berikut adalah diagram urutan dari aplikasi yang dirancang:

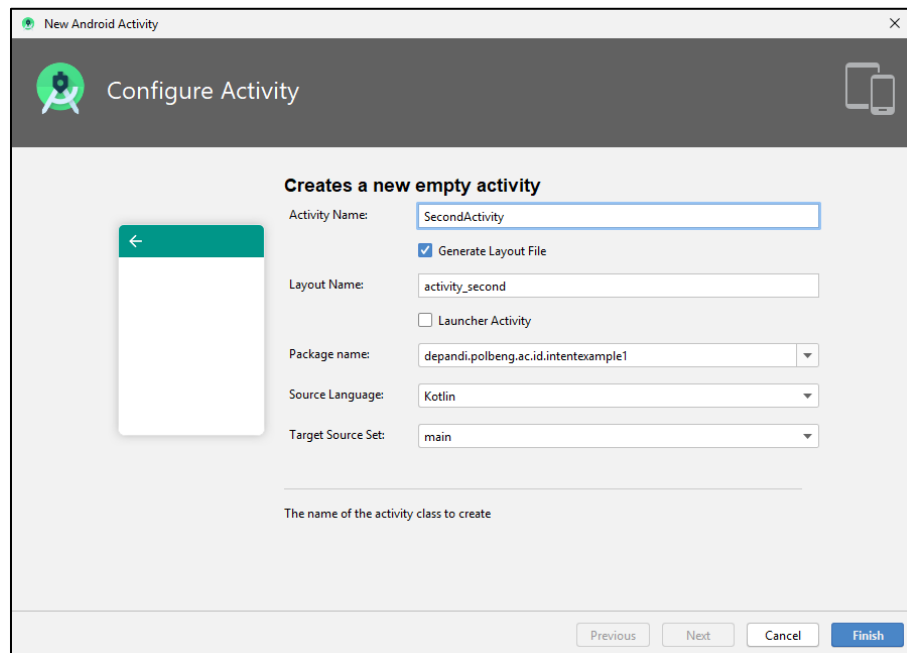


Aplikasi ini terdiri dari dua Activity yaitu MainActivity dan SecondActivity. Pada MainActivity terdapat sebuah button yang berguna untuk berpindah ke SecondActivity (dengan menggunakan fungsi startActivity). Pada SecondActivity terdapat sebuah button yang berguna untuk menghentikan *activity* tersebut dengan fungsi finish(). Terdapat beberapa method yang di-*override* pada MainActivity yaitu onPause(), onRestart() dan onResume(). Pada SecondActivity juga terdapat satu method yang di-*override* yaitu onStop().

1. Buatlah *project* baru dengan konfigurasi seperti berikut :



- Setelah selesai membuat *project* baru, buatlah *activity* baru dengan nama SecondActivity dan tipe **Empty Activity**.



- Selanjutnya editlah file *strings.xml* pada folder *values*, tambahkan beberapa variabel dan nilai *string* sebagai berikut:

```
<resources>
  <string name="app_name">IntentExample1</string>
  <string name="second_activity">Second Activity</string>
  <string name="exit_activity">Exit Activity</string>
</resources>
```

4. Editlah file *activity_main.xml* sehingga kode *layout* menjadi seperti berikut :

```
<?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"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btnToSecondActivity"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/second_activity"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

5. Bukalah file sumber *MainActivity.kt* editlah kode program hingga menjadi seperti berikut :

```
package depandi.polbeng.ac.id.intentexample1

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_main.*
import java.util.logging.Logger

class MainActivity : AppCompatActivity() {

    val Log = Logger.getLogger(MainActivity::class.java.name)

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

        Log.info("onCreate")
        btnToSecondActivity.setOnClickListener {
            val m_intent = Intent(this@MainActivity,
                SecondActivity::class.java)
            startActivity(m_intent)
        }
    }

    override fun onPause() {
        super.onPause()
        Log.info("onPause")
    }

    override fun onRestart() {
        super.onRestart()
        Log.info("onRestart")
    }
}
```

```

        override fun onResume() {
            super.onResume()
            Log.info("onResume")
        }
    }
}

```

6. Editlah file *activity_second.xml* sehingga kode *layout* menjadi seperti berikut :

```

<?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"
    tools:context=".SecondActivity">

    <Button
        android:id="@+id/btnExitActivity"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/exit_activity"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

7. Bukalah file sumber *SecondActivity.kt* editlah kode program hingga menjadi seperti berikut :

```

package depandi.polbeng.ac.id.intentexample1

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_second.*
import java.util.logging.Logger

class SecondActivity : AppCompatActivity() {

    val Log = Logger.getLogger(SecondActivity::class.java.name)

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

        Log.info("onCreate")
        btnExitActivity.setOnClickListener {
            finish()
        }
    }

    override fun onStart() {
        super.onStart()
        Log.info("onStart")
    }

    override fun onStop() {

```

```

        super.onStop()
        Log.info("onStop")
    }
}

```

8. Jalankan program di Emulator Android. Perhatikan hasilnya dan lakukan analisa terhadap kode sumber *activity_main.xml*, *activity_second.xml* dan *ActivityMain.kt*, *ActivitySecond.kt* dilaporan praktikum.



9. Berikut hasil di terminal *Logcat*:

```

logcat
2020-03-13 10:02:25.937 363-401/? D/AudioFlinger: mixer(0xefe83680) throttle end: throttle time(2)
2020-03-13 10:02:25.960 3646-3646/depandi.polbeng.ac.id.intentexample I/MainActivity: onPause
2020-03-13 10:02:26.065 3646-3651/depandi.polbeng.ac.id.intentexample I/zygote: Do partial code cache collection, code=21KB, data=28KB
2020-03-13 10:02:26.069 3646-3651/depandi.polbeng.ac.id.intentexample I/zygote: After code cache collection, code=21KB, data=28KB
2020-03-13 10:02:26.069 3646-3651/depandi.polbeng.ac.id.intentexample I/zygote: Increasing code cache capacity to 128KB
2020-03-13 10:02:26.072 3646-3646/depandi.polbeng.ac.id.intentexample I/SecondActivity: onCreate
2020-03-13 10:02:26.082 3646-3646/depandi.polbeng.ac.id.intentexample I/SecondActivity: onStart

```

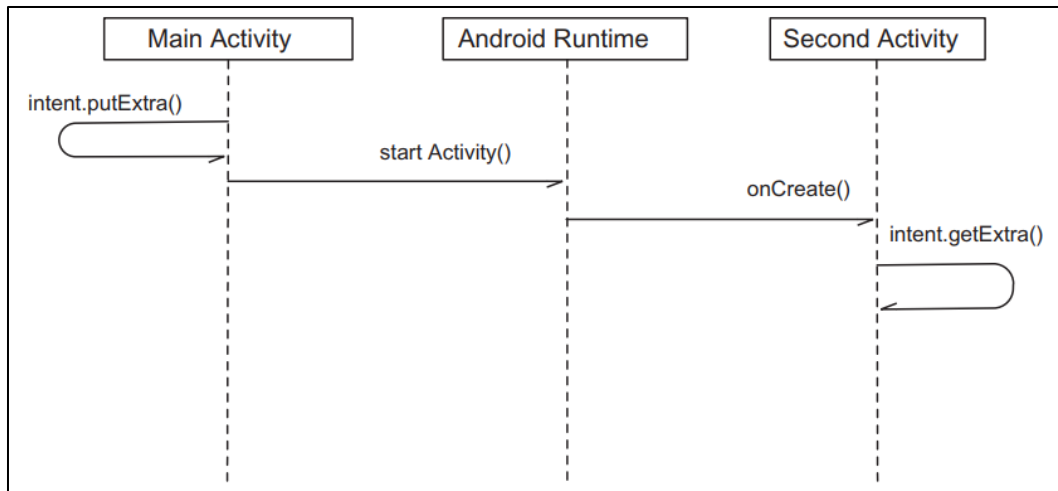
Mengirim Data Melalui Intent

Intent dapat melakukan lebih daripada hanya sekedar menjalankan *activity*; tetapi ia juga bisa mengirim dan menerima data melaluinya. Sebagai contoh kita memiliki dua *activity* dengan nama *MainActivity* dan *SecondActivity* dan ketika sebuah objek *Button View* di klik didalam *MainActivity* untuk menjalankan dan mengirim beberapa data ke *SecondActivity* maka kita perlu untuk :

1. Membuat sebuah *Intent*; untuk maksud dari contoh ini adalah eksplisit *Intent*.
2. Menambahkan data ke *Intent* menggunakan method **putExtra**.

- Menjalankan activity lain dengan memanggil method **startActivity**. Pada titik ini, Android *runtime* akan menjalankan SecondActivity.
- Dalam method **onCreate** di SecondActivity, kita dapat mengekstrak data dari Intent dengan menggunakan method **getStringExtra**.

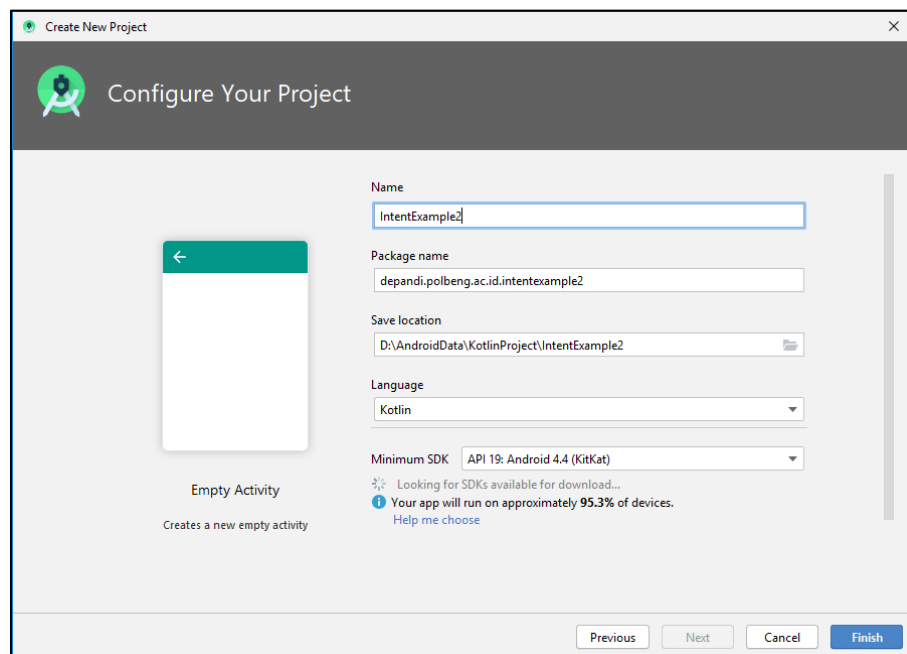
Berikut adalah diagram urutan bagaimana mengirim data dari *MainActivity* ke *SecondActivity*.



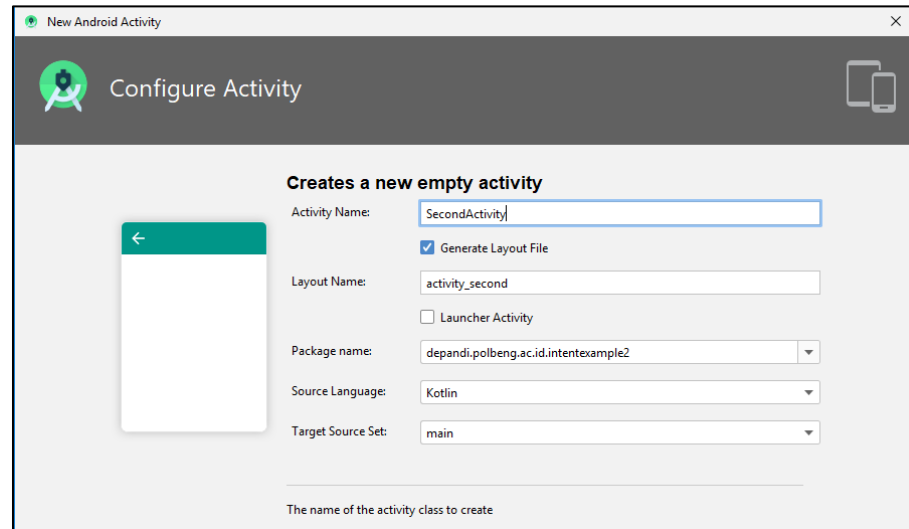
Sumber : Hagos T, 2018

Implementasi mengirim data melalui Intent

- Buatlah *project* baru dengan konfigurasi seperti berikut :



- Setelah selesai membuat project baru, buatlah *activity* baru dengan nama SecondActivity dan tipe **Empty Activity**.



- Selanjutnya editlah file *strings.xml* pada folder *values*, tambahkan beberapa variabel dan nilai *string* sebagai berikut:

```
<resources>
    <string name="app_name">IntentExample2</string>
    <string name="button">Button</string>
    <string name="name">Name</string>
    <string name="textview">TextView</string>
</resources>
```

- Editlah file *activity_main.xml* sehingga kode *layout* menjadi seperti berikut :

```
<?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"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="31dp"
        android:text="@string/button"
        app:layout_constraintEnd_toEndOf="@+id/editText"
        app:layout_constraintStart_toStartOf="@+id/editText"
        app:layout_constraintTop_toBottomOf="@+id/editText" />

    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```

        android:layout_marginTop="49dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:hint="@string/name"
        android:textAlignment="center"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:autoFillHints="" />
    </androidx.constraintlayout.widget.ConstraintLayout>

```

5. Bukalah file sumber *MainActivity.kt* editlah kode program hingga menjadi seperti berikut :

```

package depandi.polbeng.ac.id.intentexample2

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

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

        button.setOnClickListener {
            val m_data = editText.text.toString()
            val m_intent = Intent(this@MainActivity,
SecondActivity::class.java)
            m_intent.putExtra("main_activity_data", m_data)
            startActivity(m_intent)
        }
    }
}

```

6. Editlah file *activity_second.xml* sehingga kode *layout* menjadi seperti berikut :

```

<?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:gravity="center_horizontal|center_vertical"
    tools:context=".SecondActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="324dp"
        android:layout_height="wrap_content"
        android:text="@string/textview"
        android:textAlignment="center"
        android:textSize="36sp"
        tools:layout_editor_absoluteX="35dp"
        tools:layout_editor_absoluteY="78dp"
        tools:ignore="MissingConstraints" />

</LinearLayout>

```


7. Bukalah file sumber *SecondActivity.kt* editlah kode program hingga menjadi seperti berikut :

```
package depandi.polbeng.ac.id.intentexample2

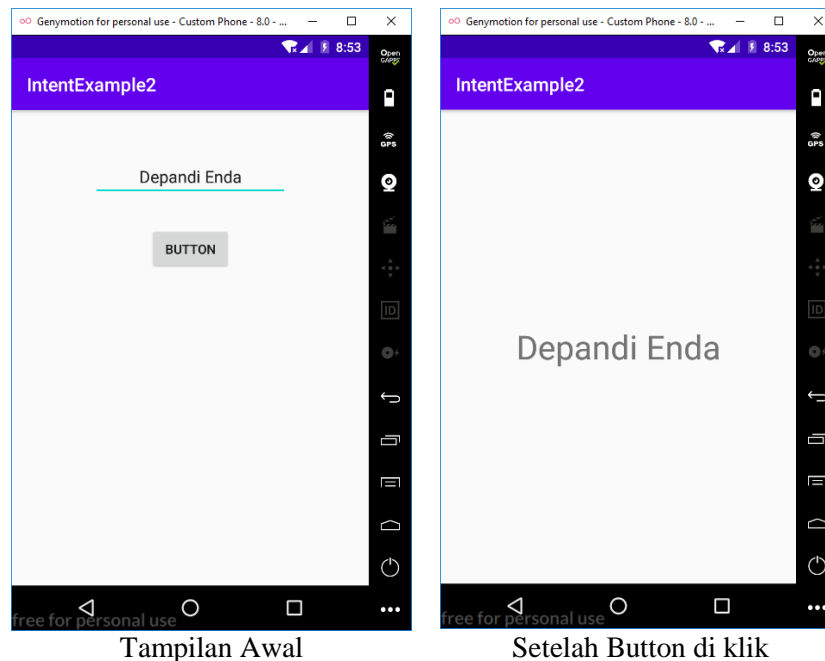
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_second.*

class SecondActivity : AppCompatActivity() {

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

        val m_data = intent.getStringExtra("main_activity_data")
        textView.setText(m_data)
    }
}
```

8. Jalankan program di Emulator Android. Perhatikan hasilnya dan lakukan analisa terhadap kode sumber *activity_main.xml*, *activity_second.xml* dan *ActivityMain.kt*, *ActivitySecond.kt* dilaporan praktikum.



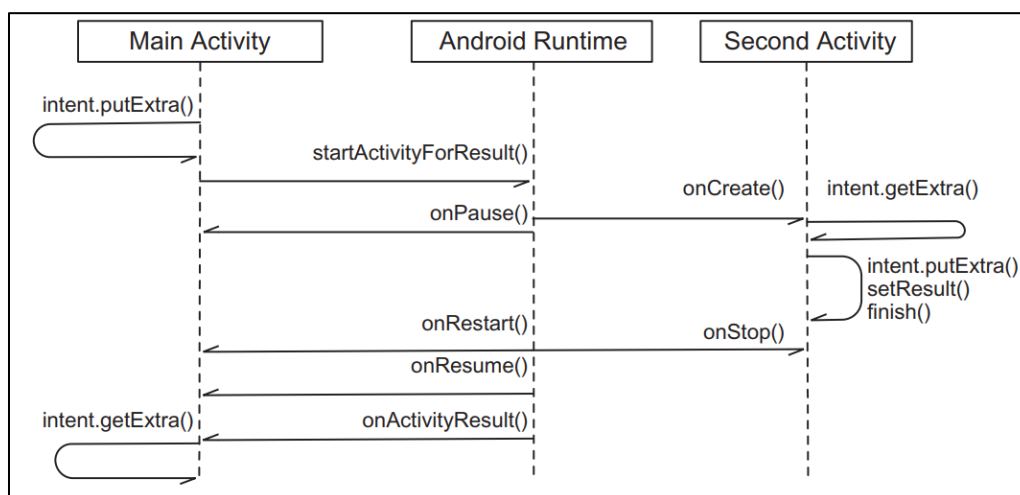
Mengirim dan Mengembalikan data dari Activity lain

Kita juga bisa mengirim dan mengembalikan beberapa data ke *MainActivity*, kita perlu melakukan beberapa hal berikut :

1. Membuat sebuah eksplisit *Intent*.
2. Menambahkan data ke *Intent* menggunakan method **putExtra**.

3. Menjalankan *activity* dengan memanggil method *startActivityForResult*. Seperti method *startActivity*, kita perlu melewati objek *Intent* sebagai parameter method ini. Sebagai tambahan, kita juga perlu melewati *request code* kedalamnya. Sebuah *request code* berperan sebagai beberapa urutan token. Ketika kita memulai sebuah *activity* dan kita berharap mengembalikan beberapa hasil kembali, apapun *activity* lain dapat mengirim kembali apapun hasil. Jika kita memiliki dua *activity* dalam sebuah *project*, ia mungkin membingungkan ketika kita mengembalikan hasil. Kita memerlukan cara untuk merekam siapa yang mengirim kembali hasil tersebut, dan *request code* akan membantu kita melakukannya.
4. Dalam method *onCreate* di *SecondActivity*, kita dapat mengekstrak data dari *Intent* dengan menggunakan method **getStringExtra**.
5. Kita dapat melakukan beberapa komputasi dalam *SecondActivity*. Ketika kita siap untuk mengirim data kembali, kita akan melakukan hal berikut:
 - Memperoleh referensi objek **Intent**
 - Menambahkan data ke *Intent* menggunakan method **putExtra**
 - Memanggil method **setResult** di *SecondActivity*.
6. Kembali ke *MainActivity*, apapun hasil yang kita harapkan kembali dari *SecondActivity* atau *activity* lain, dapat diterima melalui fungsi **callback** *onActivityResult*. Method ini memiliki tiga unsur parameter yaitu; *request code*, *result code*, dan objek *Intent* yang dikirim kembali oleh *SecondActivity*.

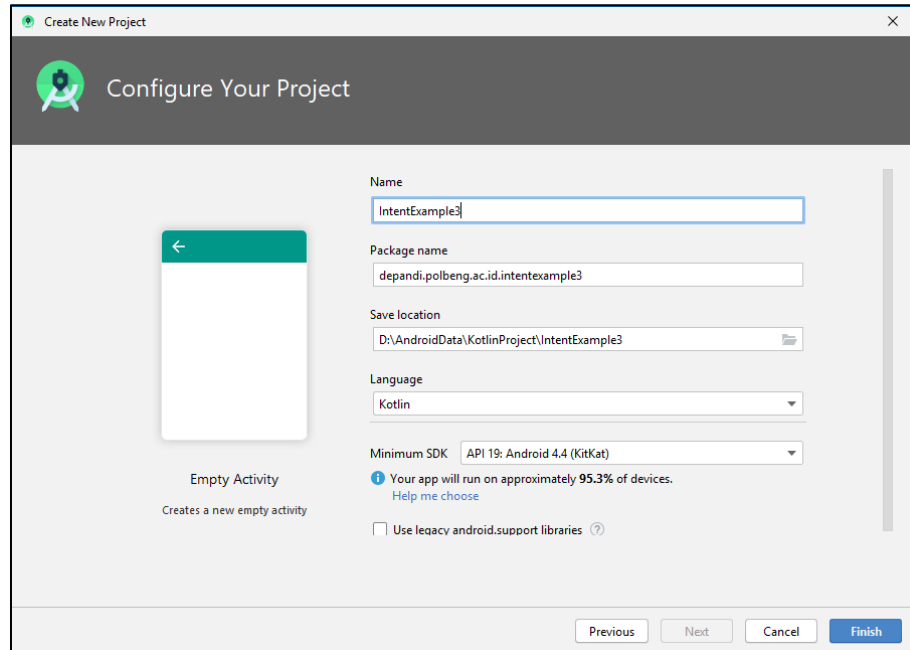
Berikut menunjukkan diagram urutan bagaimana mengirim dan menerima kembali hasil data dari *activity* lain.



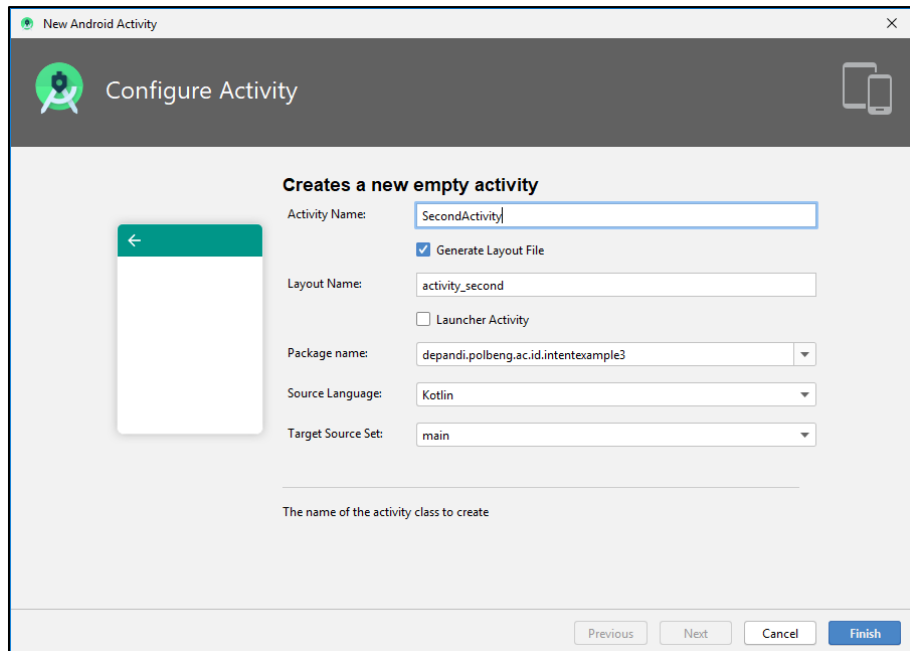
Sumber : Hagos T, 2018

Implementasi program mengirim dan mengembalikan data dari Activity lain

1. Buatlah *project* baru dengan konfigurasi seperti berikut :



2. Setelah selesai membuat project baru, buatlah *activity* baru dengan nama SecondActivity dan tipe **Empty Activity**.



3. Selanjutnya editlah file *strings.xml* pada folder *values*, tambahkan beberapa variabel dan nilai *string* sebagai berikut:

```
<resources>
  <string name="app_name">IntentExample3</string>
  <string name="calculate_bmi">calculate BMI</string>
  <string name="height">Height</string>
  <string name="weight">Weight</string>
  <string name="textview">TextView</string>
  <string name="calc_bmi">calc bmi</string>
</resources>
```

4. Editlah file *activity_main.xml* sehingga kode *layout* menjadi seperti berikut :

```
<?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"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btn_send_data"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="21dp"
        android:text="@string/calculate_bmi"
        app:layout_constraintEnd_toEndOf="@+id/txt_bmi"
        app:layout_constraintStart_toStartOf="@+id/txt_bmi"
        app:layout_constraintTop_toBottomOf="@+id/input_height" />

    <EditText
        android:id="@+id/input_height"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="23dp"
        android:ems="10"
        android:inputType="numberDecimal"
        android:hint="@string/height"
        app:layout_constraintEnd_toEndOf="@+id/btn_send_data"
        app:layout_constraintStart_toStartOf="@+id/btn_send_data"
        app:layout_constraintTop_toBottomOf="@+id/input_weight"
        android:autofillHints="" />

    <EditText
        android:id="@+id/input_weight"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="68dp"
        android:ems="10"
        android:inputType="numberDecimal"
        android:hint="@string/weight"
        app:layout_constraintEnd_toEndOf="@+id/input_height"
        app:layout_constraintStart_toStartOf="@+id/input_height"
        app:layout_constraintTop_toTopOf="parent"
```

```

        android:autofillHints="" />

<TextView
    android:id="@+id/txt_bmi"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="33dp"
    android:text="@string/textview"
    android:textSize="36sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/btn_send_data" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

5. Bukalah file sumber *MainActivity.kt* editlah kode program hingga menjadi seperti berikut :

```

package depandi.polbeng.ac.id.intentexample3

import android.app.Activity
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_main.*

val SECOND_ACTIVITY = 1000

class MainActivity : AppCompatActivity() {

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

        input_weight.setHint("weight (lbs)")
        input_height.setHint("height (inches)")
        btn_send_data.setOnClickListener {
            val m_intent = Intent(this@MainActivity,
SecondActivity::class.java)
            val m_bundle = Bundle()
            m_bundle.putFloat("weight",
input_weight.text.toString().toFloat())
            m_bundle.putFloat("height",
input_height.text.toString().toFloat())
            m_intent.putExtra("main_activity_data", m_bundle)
            startActivityForResult(m_intent, SECOND_ACTIVITY)
        }

        override fun onResume() {
            super.onResume()
            clearInputs()
        }

        override fun onActivityResult(requestCode: Int, resultCode: Int, data:
Intent?) {
            super.onActivityResult(requestCode, resultCode, data)
            if ((requestCode == SECOND_ACTIVITY) and (resultCode ==
Activity.RESULT_OK)) {

```

```

        val bmi = data!!.getFloatExtra("second_activity_data", 1.0F)
        val bmiString = "%.2f".format(bmi)
        input_height.setText("")
        input_weight.setText("")
        txt_bmi.setText("BMI : $bmiString ${getBMIDescription(bmi)}")
    }
}

private fun getBMIDescription(bmi: Float): String {
    return when (bmi) {
        in 1.0..18.5 -> "Underweight"
        in 18.6..24.9 -> "Normal weight"
        in 25.0..29.9 -> "Overweight"
        else -> "Obese"
    }
}

private fun clearInputs() {
    input_weight.setText("")
    input_height.setText("")
}
}

```

6. Editlah file *activity_second.xml* sehingga kode *layout* menjadi seperti berikut :

```

<?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"
    tools:context=".SecondActivity">

    <TextView
        android:id="@+id/txt_intentdata"
        android:layout_width="346dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="109dp"
        android:text="@string/textview"
        android:textAlignment="center"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/btn_calculate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="29dp"
        android:text="@string/calc_bmi"
        app:layout_constraintEnd_toEndOf="@+id/txt_intentdata"
        app:layout_constraintStart_toStartOf="@+id/txt_intentdata"
        app:layout_constraintTop_toBottomOf="@+id/txt_intentdata" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

7. Bukalah file sumber *SecondActivity.kt* editlah kode program hingga menjadi seperti berikut :

```
package depandi.polbeng.ac.id.intentexample3

import android.app.Activity
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_second.*

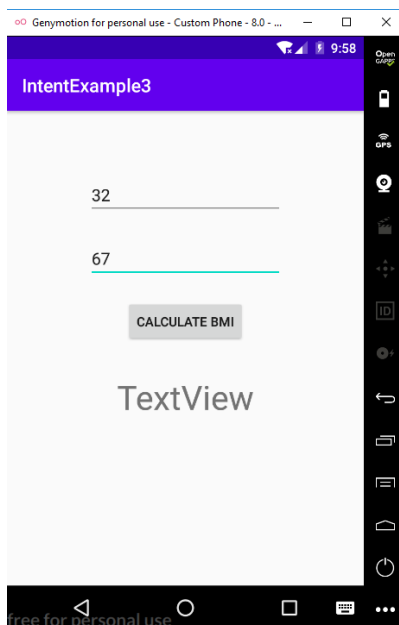
class SecondActivity : AppCompatActivity() {

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

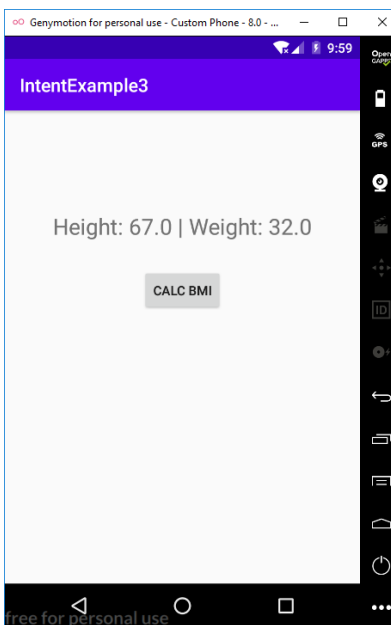
        val bundle = intent.getBundleExtra("main_activity_data")

        val height = bundle.getFloat("height")
        val weight = bundle.getFloat("weight")
        txt_intentdata.text = "Height: $height | Weight: $weight"
        btn_calculate.setOnClickListener {
            val m_intent = Intent()
            val m_bmi = 703 * (weight / (height * height))
            m_intent.putExtra("second_activity_data", m_bmi)
            setResult(Activity.RESULT_OK, m_intent)
            finish()
        }
    }
}
```

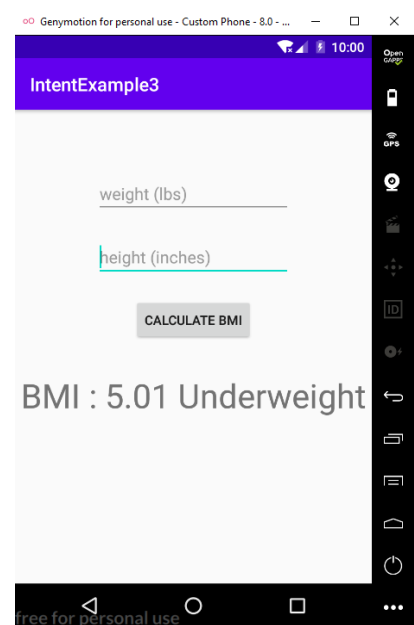
8. Jalankan program di Emulator Android. Perhatikan hasilnya dan lakukan analisa terhadap kode sumber *activity_main.xml*, *activity_second.xml* dan *ActivityMain.kt*, *ActivitySecond.kt* dilaporan praktikum.



Tampilan Awal



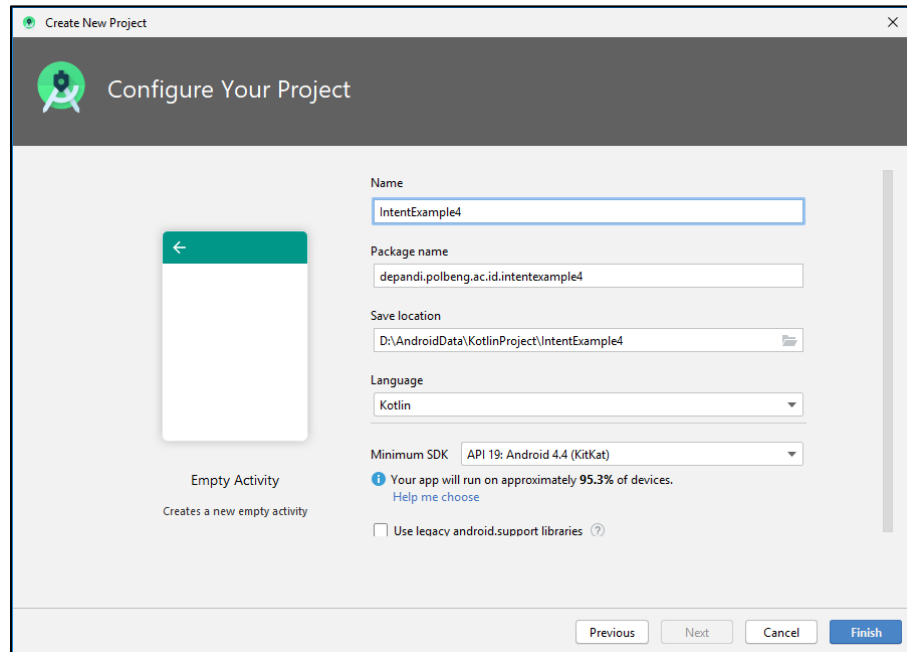
Setelah Button Calculate BMI di klik



Setelah Button Calc BMI di klik

Implementasi Implisit Intent

1. Buatlah *project* baru dengan konfigurasi seperti berikut :



2. Setelah selesai membuat *project* baru, selanjutnya bukalah file sumber *MainActivity.kt*, editlah kode program hingga menjadi seperti berikut :

```
package depandi.polbeng.ac.id.intentexample4

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.content.Intent
import android.net.Uri
import android.view.Menu
import android.view.MenuItem

class MainActivity : AppCompatActivity() {

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

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menu?.add("Web")
        menu?.add("Map")
        menu?.add("Phone number")
        return super.onCreateOptionsMenu(menu)
    }

    override fun onOptionsItemSelected(item: MenuItem?): Boolean {
        var m_uri: Uri
        var m_intent: Intent = Intent()
        when (item?.toString()) {
```

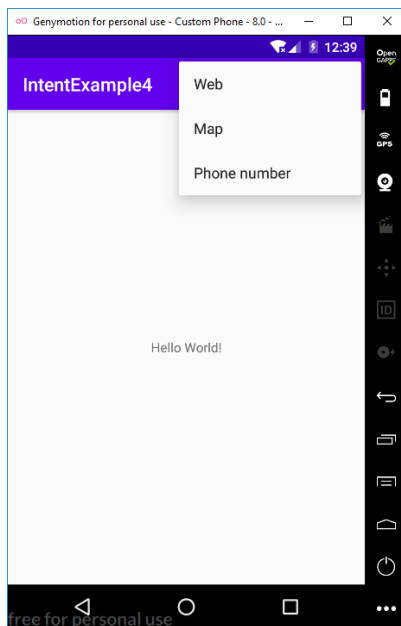


```

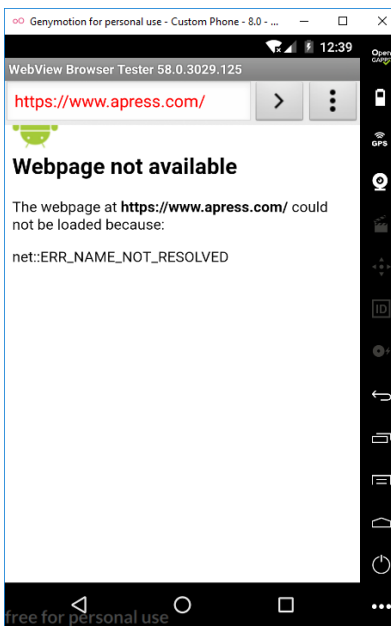
        "Web" -> {
            m_uri = Uri.parse("https://www.apress.com")
            m_intent = Intent(Intent.ACTION_VIEW, m_uri)
        }
        "Map" -> {
            m_uri = Uri.parse("geo:40.7113399,-74.0263469")
            // This would have worked as well
            // m_uri = Uri.parse("https://maps.google.com/maps?q =
40.7113399, -74.0263469")
            m_intent = Intent(Intent.ACTION_VIEW, m_uri)
        }
        "Phone number" -> {
            m_uri = Uri.parse("tel:639285083333")
            m_intent = Intent(Intent.ACTION_DIAL, m_uri)
        }
    }
    startActivity(m_intent)
    return true
}
}

```

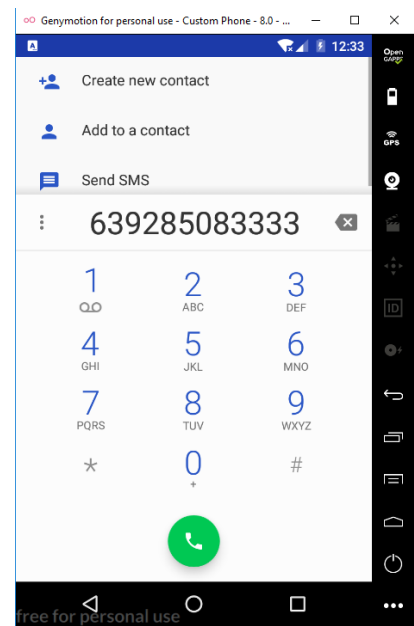
3. Jalankan program di Emulator Android. Perhatikan hasilnya dan lakukan analisa terhadap kode logika program *ActivityMain.kt* dilaporan praktikum.



Tampilan Menu Awal



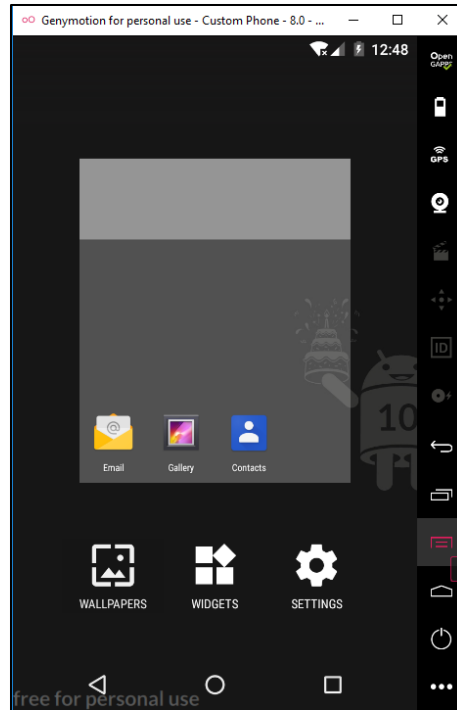
Tampilan Implisit Intent Web



Tampilan Implisit Intent Contacts

Menu

Menu adalah sesuatu yang sangat penting dalam rancangan UI. Menu mengizinkan pengguna untuk memperoleh fungsionalitas aplikasi. Secara tradisional, sistem menu diorganisasikan secara hirarki kedalam kelompok, yang bermaksud sebelum pengguna memperoleh target aksinya, pengguna perlu menjelajah hirarki daripada menu. Berikut adalah contoh tampilan menu pada Sistem Emulator Android versi 8.0 di Genymotion.



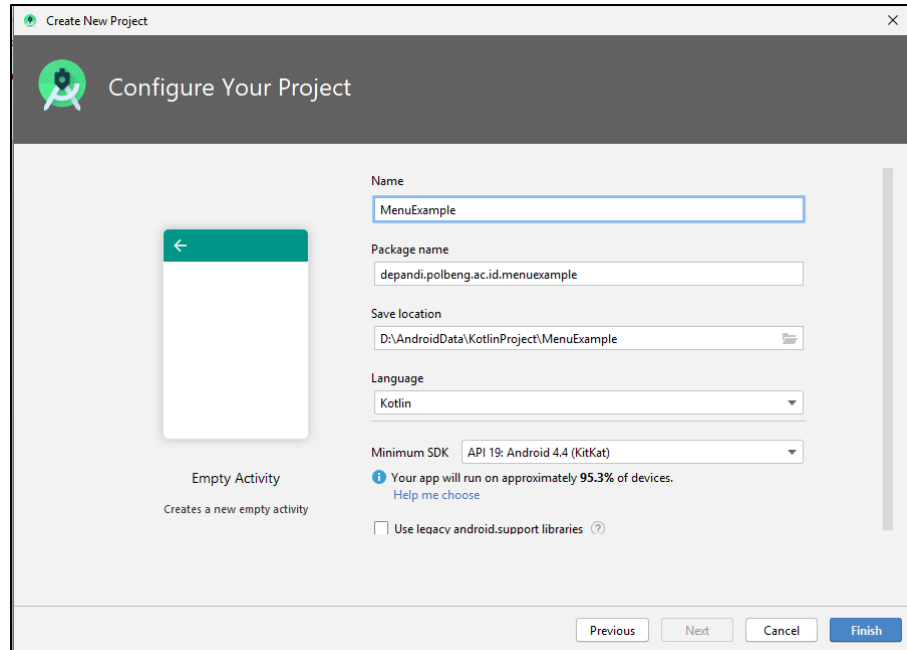
Dalam API Android versi 10 kebawah, pilihan menu akan tampil pada dasar layar ketika pengguna menekan button menu perangkat keras. Dalam API Android versi 11 keatas, item-item dari pilihan menu tersedia di app bar. Secara default, system meletakkan semua item dari pilihan aksi menu secara mengambang, dimana pengguna dapat melihat aksi icon yang mengambang pada sisi kanan app bar.

Untuk menambahkan menu kedalam aplikasi, kita perlu melakukan hal berikut :

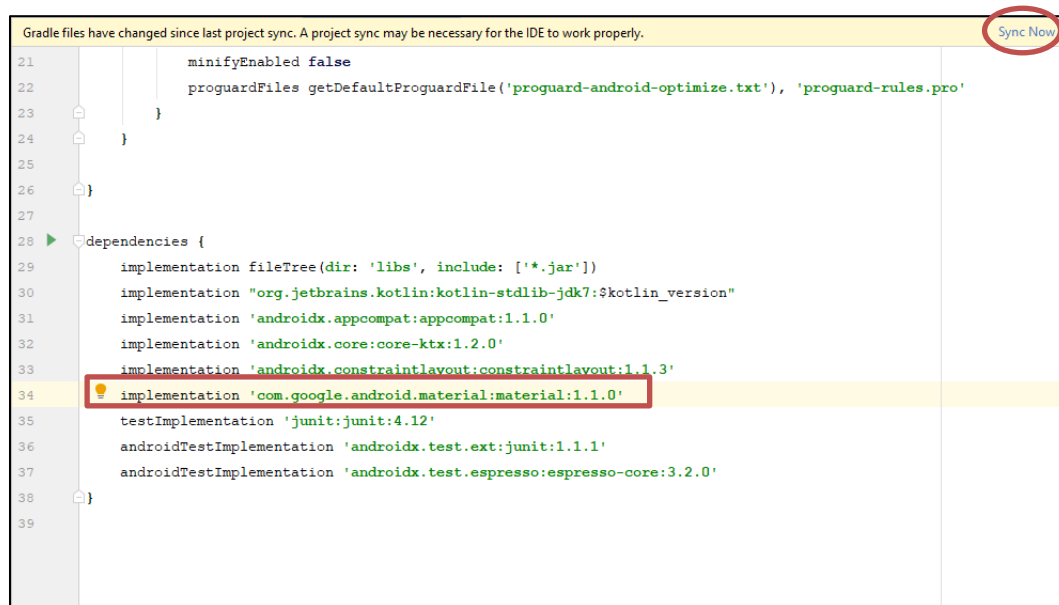
1. Membuat file sumber daya menu. Kita akan membuat sebuah folder menu didalam folder app/res. Kemudian, kita membuat file sumber daya menu didalamnya.
2. Menggunakan sumber daya menu di program utama. Kita akan meng-override fungsi `onCreateOptionsMenu` dari `MainActivity` dan memanggil fungsi `inflate` dari objek `Menu`.
3. Menambahkan penanganan event kedalam menu item. Kita akan meng-override fungsi `onOptionsItemSelected` dari `MainActivity`, dan mengarahkan aksi pengguna tergantung menu mana yang di klik.
4. Secara opsional, menambahkan vector images kedalam menu.

Implementasi Menu

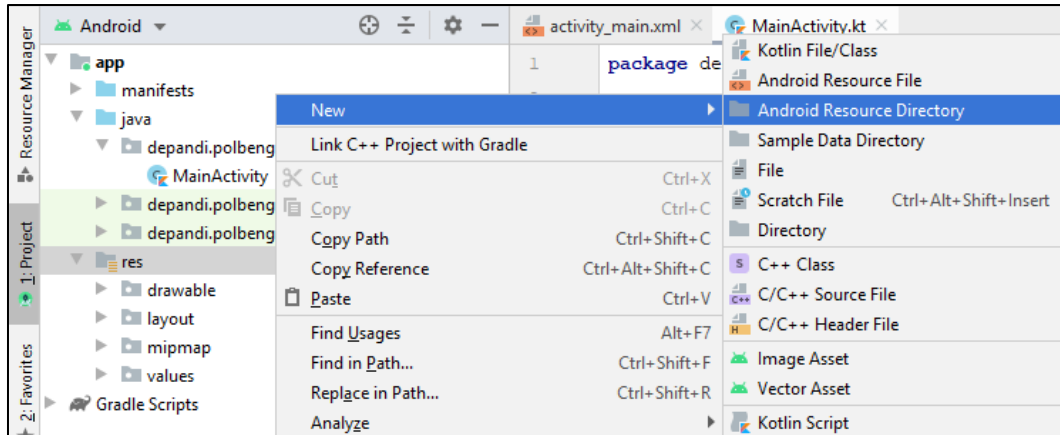
1. Buatlah *project* baru dengan konfigurasi seperti berikut :



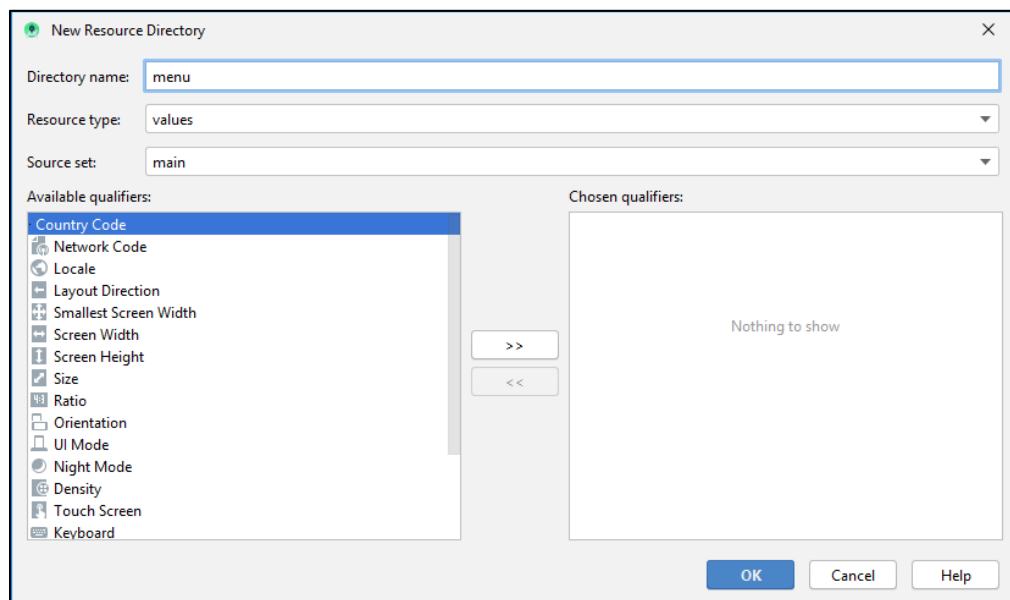
- Setelah selesai membuat *project* baru, selanjutnya bukalah file konfigurasi module **build.gradle(Module: app)**, tambahkan *library* tambahan material *design* yang diberi tanda kotak merah pada gambar berikut. Klik **Sync Now** untuk menerapkan penambahan *library* dalam *project*.



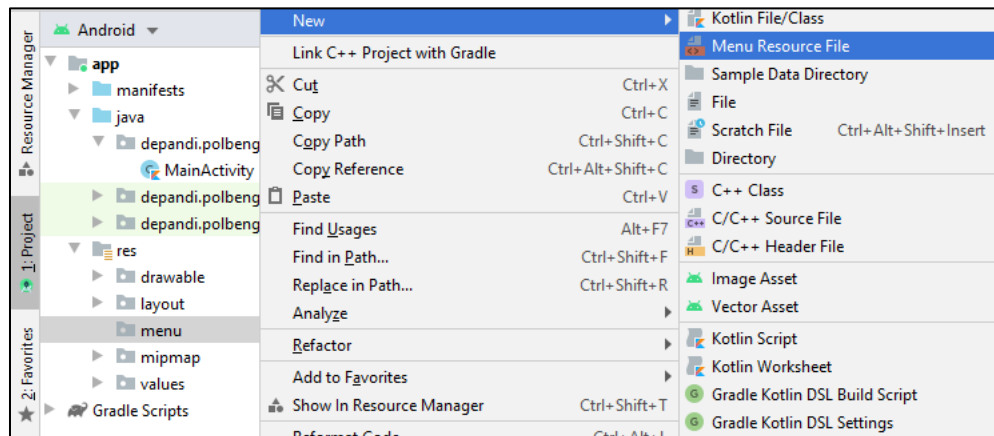
- Selanjutnya buatlah *folder* baru dengan cara klik kanan pada folder res → New → Klik Android Resource Directory.



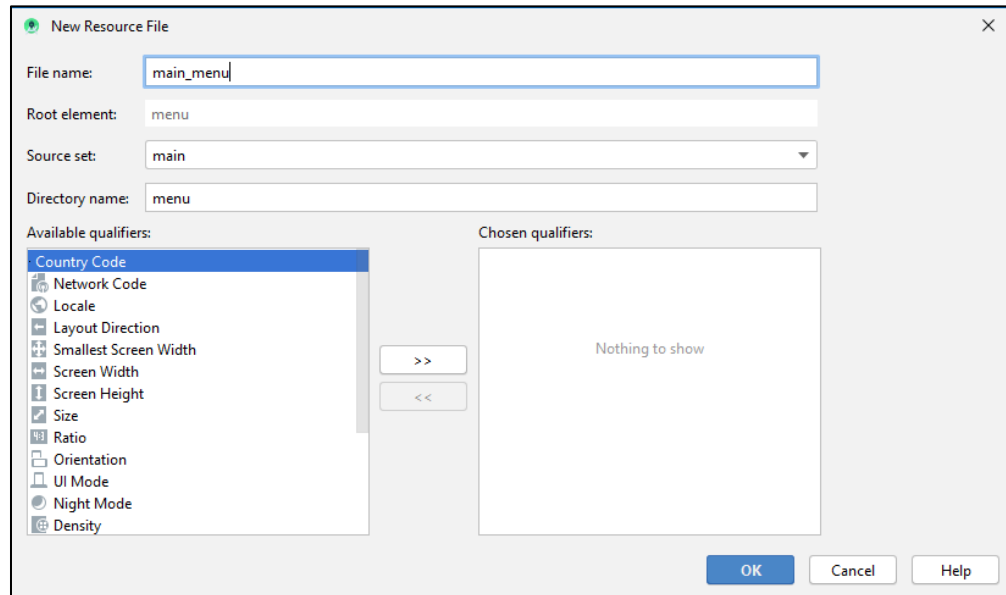
4. Pada form New Resource Directory isian Directory name: masukkan dengan nama menu. Klik OK.



5. Pada folder menu yang telah dibuat klik kanan → New → Klik Menu Resource File.



6. Pada form New Resource File isian Directory name: masukkan dengan nama menu. Klik OK.



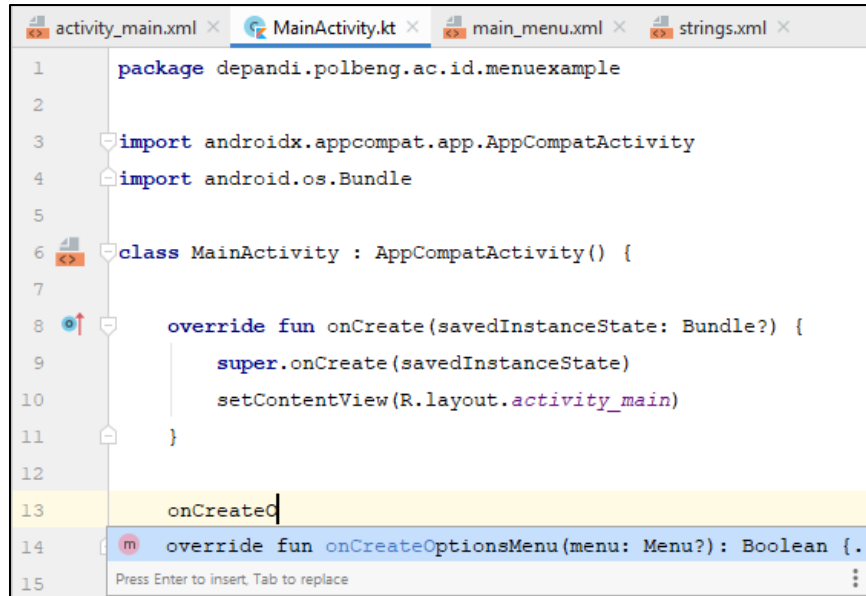
7. Setelah itu masukkan terlebih dahulu deklarasi string nama menu pada file string.xml di folder *values*. Isinya adalah sebagai berikut :

```
<resources>
    <string name="app_name">MenuExample</string>
    <string name="menuExit">Exit</string>
    <string name="menuHelp">Help</string>
    <string name="menuEdit">Edit</string>
    <string name="menuFile">File</string>
</resources>
```

8. Kemudian ubahlah file sumber daya menu di folder menu yaitu file main_menu.xml. Tambahkan deklarasi pilihan menu di layout utama berikut :

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/menuFile" android:title="@string/menuFile" />
    <item android:id="@+id/menuEdit" android:title="@string/menuEdit" />
    <item android:id="@+id/menuHelp" android:title="@string/menuHelp" />
    <item android:id="@+id/menuExit" android:title="@string/menuExit" />
</menu>
```

9. Bukalah file MainActivity.kt tambahkan override fungsi onCreateOptionsMenu didalam file tersebut. Caranya adalah ketik onCreateO setelah fungsi onCreate maka akan tampil popup rekomendasi untuk mengenerate override fungsi onCreateOptionsMenu seperti gambar dibawah ini. Jika popup tidak muncul tekan **alt + enter** pada keyboard untuk memunculkan popup rekomendasi. Pilihlah atau klik pada pilihan **override fun onCreateOptionsMenu()**.



```
1 package depandi.polbeng.ac.id.menuexample
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5
6 class MainActivity : AppCompatActivity() {
7
8     override fun onCreate(savedInstanceState: Bundle?) {
9         super.onCreate(savedInstanceState)
10        setContentView(R.layout.activity_main)
11    }
12
13    onCreateO
14    { m override fun onCreateOptionsMenu(menu: Menu?): Boolean { ..
15    Press Enter to insert, Tab to replace
```

10. Selanjutnya tambahkan fungsi `menuInflater.inflate()` dalam blok `onCreateOptionsMenu`. Sehingga kodenya menjadi seperti berikut:

```
package depandi.polbeng.ac.id.menuexample

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.Menu

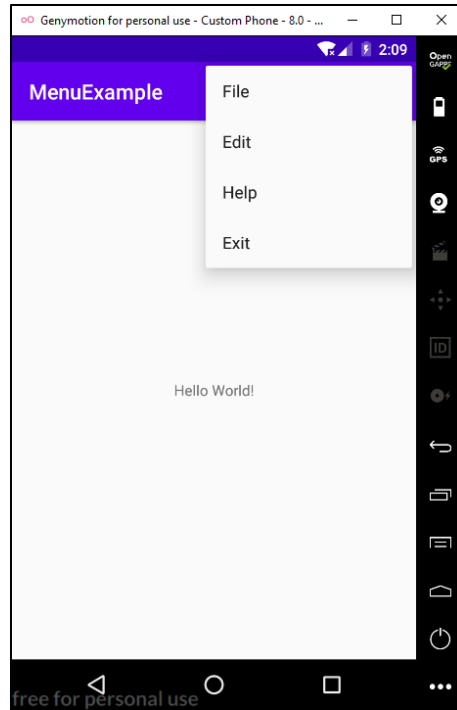
class MainActivity : AppCompatActivity() {

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

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.main_menu, menu)
        return super.onCreateOptionsMenu(menu)
    }
}
```

Fungsi `inflate()` membuat menu item menggunakan file menu XML yang telah kita buat di langkah 3 hingga 8 dan melampirkannya ke objek menu (sebagai parameter yang kedua).

11. Coba jalankan program. Perhatikan apakah menu yang dibuat sudah muncul di app bar. Jika menu sudah muncul di app bar, langkah berikutnya ialah cobalah untuk meng-klik menu yang ada, perhatikan apakah sudah ada aksi dari menu tersebut.



12. Tambahkan id pada root layout file activity_main.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:id="@+id/root_layout"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

13. Jika sudah berhasil menampilkan menu langkah berikutnya ialah memberikan aksi terhadap beberapa pilihan menu yang tersedia. Untuk menangani setiap kejadian dari setiap menu item, kita akan meng-override fungsi `onOptionsItemSelected()` dalam `MainActivity`. Berikut adalah kode akhir dari `MainActivity.kt`.

```
package depandi.polbeng.ac.id.menuexample

import androidx.appcompat.app.AppCompatActivity
```

```

import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import com.google.android.material.snackbar.Snackbar
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

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

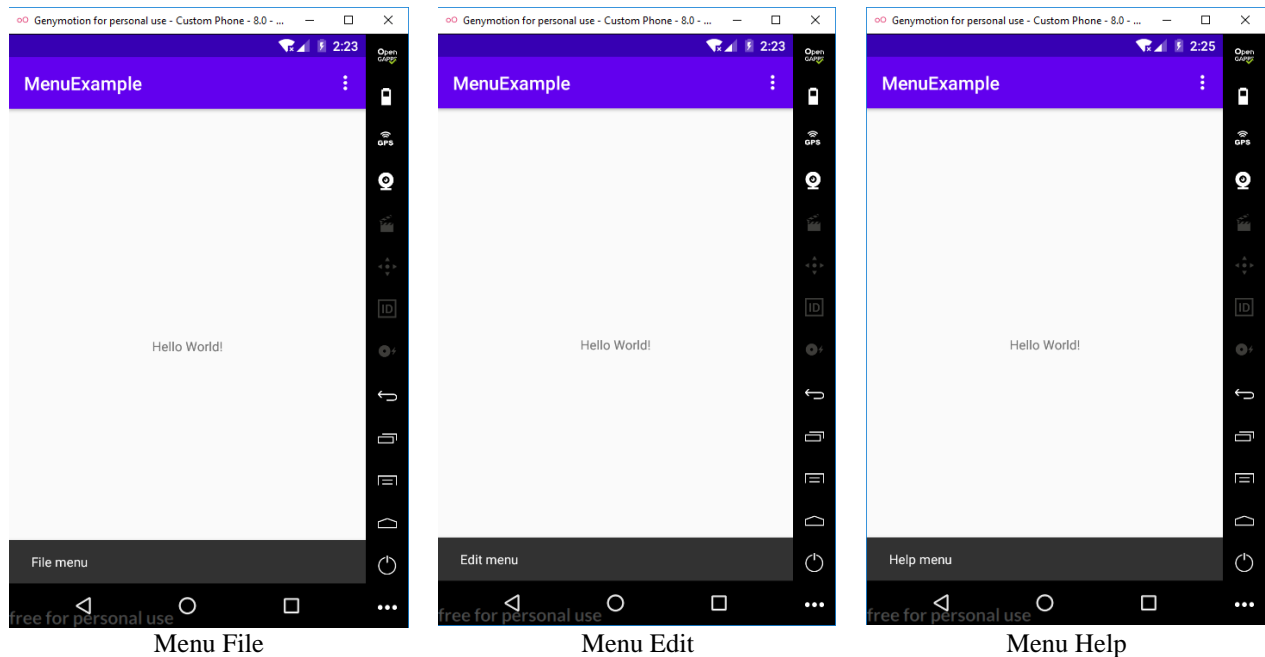
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.main_menu, menu)
        return super.onCreateOptionsMenu(menu)
    }

    override fun onOptionsItemSelected(item: MenuItem?): Boolean {
        when (item?.itemId) {
            R.id.menuFile -> {
                showMessage("File menu")
                return true
            }
            R.id.menuEdit -> {
                showMessage("Edit menu")
                return true
            }
            R.id.menuHelp -> {
                showMessage("Help menu")
                return true
            }
            R.id.menuExit -> {
                showMessage("Exit")
                return true
            }
        }
        return super.onOptionsItemSelected(item)
    }

    private fun showMessage(msg:String) {
        Snackbar.make(root_layout, msg, Snackbar.LENGTH_LONG).show()
    }
}

```

14. Jalankan program di Emulator Android. Cobalah untuk mengklik beberapa pilihan item menu satu persatu dan perhatikan apa yang terjadi. Buatlah analisa dari program yang telah dibuat dan masukkan kedalam laporan praktikum.



Fragments

Jika kita berpikir sebuah activity sebagai unit komposisi dari antarmuka kita (UI), maka berpikirlah bahwa sebuah fragment adalah sebagai mini-activity, ia adalah unit komposisi yang lebih kecil. Kita bisa menampilkan dan menyembunyikan sebuah fragment selama runtime sebagai respon terhadap sesuatu yang dilakukan oleh pengguna (misalnya : mengubah orientasi perangkat dari potrait menjadi landscape).

Sebuah fragment sama seperti activity, ia terdiri dari dua bagian; sebuah kelas di Kotlin dan sebuah file layout. Identya adalah hampir sama dengan mendefenisikan elemen UI didalam file XML dan kemudian juga meng-inflate file program (sebuah kelas) semua objek view dalam XML akan menjadi sebuah objek. Setelah itu kita dapat merujuk setiap objek view dalam file XML menggunakan R.class. Untuk membuat sebuah fragment, umumnya kita melakukan hal berikut :

1. Membuat file sumber daya XML, letakkan didalam folder /app/res/layout, seperti kita meletakkan activity_main.xml.
2. Memberikan file sumber daya yang baru. Contohnya fragment_booktitles.
3. Membuat kelas fragment. Kita harus memilih satu diantara dua kelas untuk membuat fragment, dimana kelas yang harus diwariskan ialah **androidx.fragment.app.Fragment** atau **android.support.v4.app.Fragment** (tergantung jenis API yang mendukung).

4. Berikutnya, mengaitkan kelas fragment yang dibuat dengan file sumber daya layout XML. Kita dapat melakukannya dengan meng-inflate file sumber daya XML didalam method onCreate pada kelas fragment.
5. Tambahkan fragment yang baru dibuat ke layout utama.

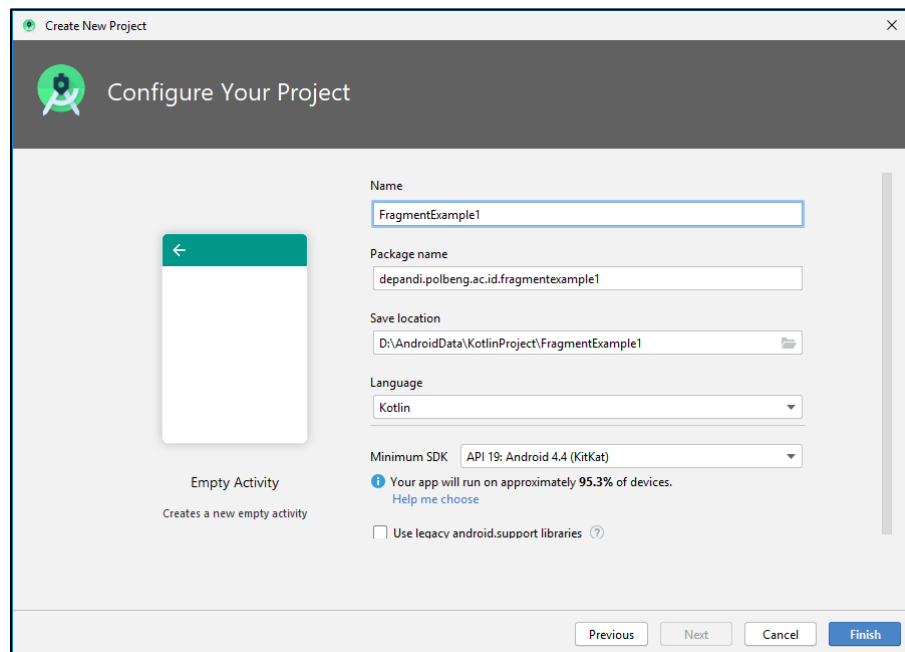
Implementasi Fragment Statis di Kotlin

Deskripsi Project

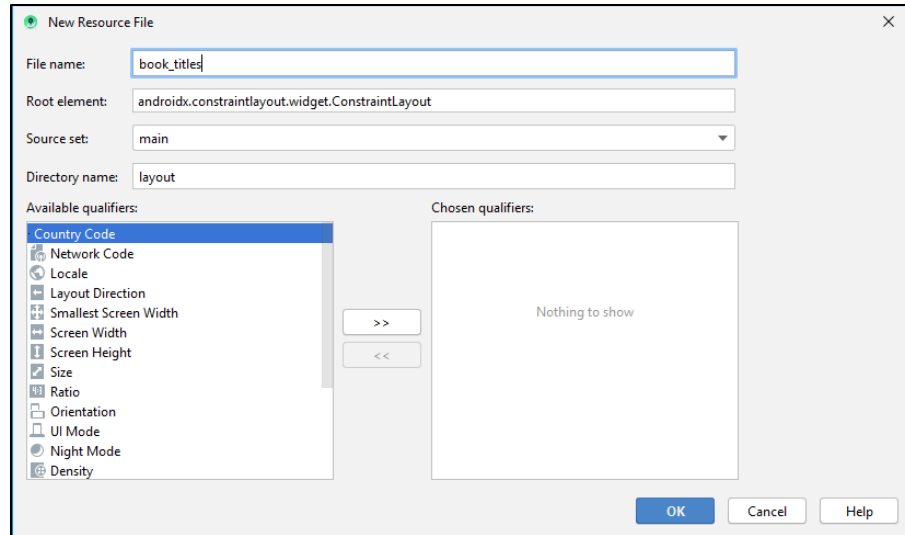
- ✎ Menggunakan 2 *fragment* dalam MainActivity.
- ✎ Fragment pertama berisi daftar buku; kita akan menyediakan pilihan kepada pengguna dengan meng-klik salah satu *radio button*
- ✎ Fragment kedua berisi deskripsi dari buku yang sedang dipilih.
- ✎ Fragment akan mengatur dirinya sendiri tergantung dari bagaimana pengguna memegang perangkat (dalam orientasi *potrait* atau *landscape*)

Langkah-langkah pembuatan project.

1. Buatlah *project* baru dengan konfigurasi seperti berikut :



2. Setelah selesai membuat *project* baru, selanjutnya buatlah layout fragment dengan cara klik kanan pada folder layout → New → Layout Resource File. Masukkan nama file book_titles.



- Setelah itu masukkan terlebih dahulu deklarasi string pada file string.xml di folder *values*.
Isinya adalah sebagai berikut :

```
<resources>
    <string name="app_name">FragmentExample1</string>
    <string name="learn_android_studio_3">Learn Android Studio 3</string>
    <string name="learn_android_studio_3_with_kotlin">Learn Android Studio
3 with Kotlin</string>
    <string name="minimum_android_programming">Minimum Android
Programming</string>
</resources>
```

- Tambahkan kode layout dari file book_titles.xml menjadi seperti berikut:

```
<?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"
tools:layout_editor_absoluteY="81dp">

    <RadioGroup
        android:id="@+id/radioGroup"
        android:layout_width="354dp"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:layout_editor_absoluteX="16dp"
        tools:layout_editor_absoluteY="75dp">

        <RadioButton
            android:id="@+id/r1as3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
```

```

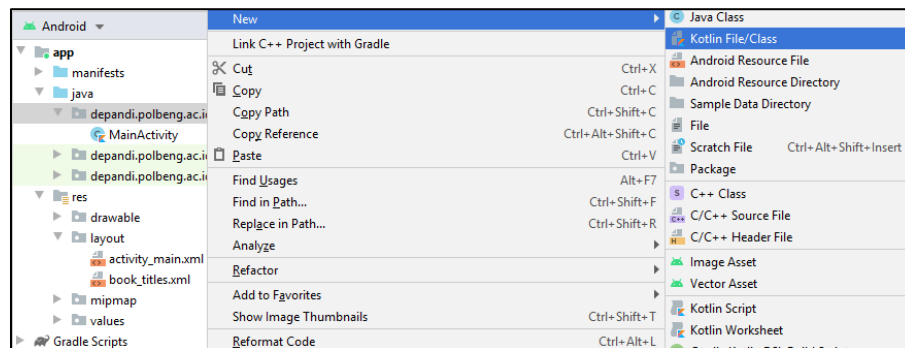
        android:text="@string/learn_android_studio_3"
        android:textSize="18sp" />

<RadioButton
    android:id="@+id/r1as3kotlin"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="@string/learn_android_studio_3_with_kotlin"
    android:textSize="18sp" />

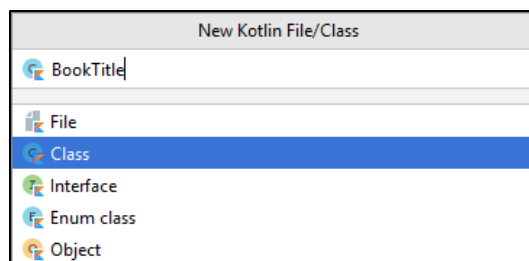
<RadioButton
    android:id="@+id/rminandroid"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="@string/minimum_android_programming"
    android:textSize="18sp" />
</RadioGroup>
</androidx.constraintlayout.widget.ConstraintLayout>

```

5. Buatlah kelas baru dengan cara klik kanan pada paket project anda → New → Klik Kotlin File Class.



6. Pada pilihan dialog mengambang pastikan anda memilih jenis Class dan masukkan nama kelas BookTitle.



7. Bukalah file BookTitle.kt yang telah dibuat, tambahkan baris kode berikut:

```

package depandi.polbeng.ac.id.fragmentexample1

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup

```

```

import androidx.fragment.app.Fragment
import kotlinx.android.synthetic.main.book_titles.*

class BookTitle : Fragment(), View.OnClickListener {

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val v = inflater.inflate(R.layout.book_titles, container, false)
        return v
    }

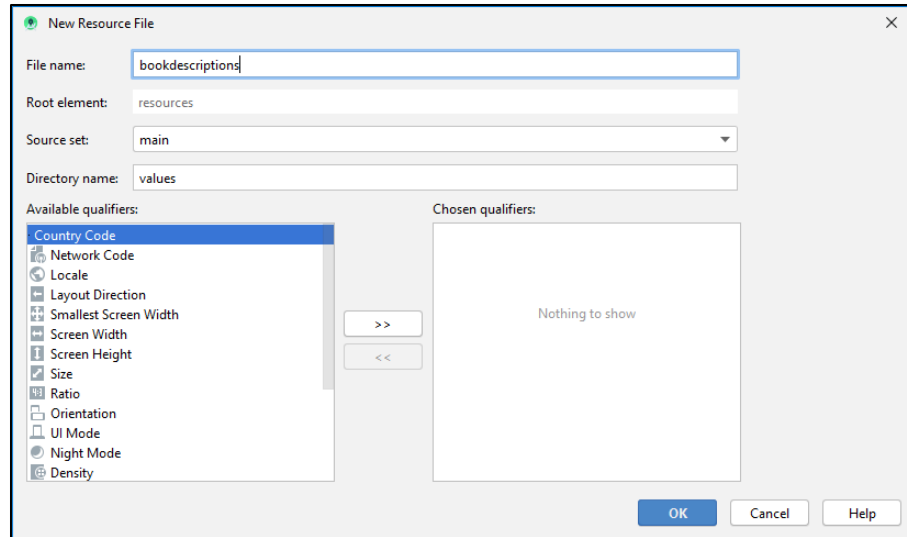
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        rlas3.setOnClickListener(this)
        rlas3kotlin.setOnClickListener(this)
        rminandroid.setOnClickListener(this)
    }

    override fun onClick(v: View?) {
        var index: Int = 0
        when(v?.id) {
            R.id.rlas3 -> {
                index = 0
            }
            R.id.rlas3kotlin -> {
                index = 1
            }
            R.id.rminandroid -> {
                index = 2
            }
        }
    }
}

```

Kelas fragment dapat dihubungkan dengan file sumber daya UI dengan cara meng-inflate file sumber daya dan mengembalikannya dari dalam fungsi *callback* **onCreateView**.

8. Buatlah array string pada sebuah file dalam folder app/res/values caranya klik kanan pada folder values → New → Values XML File. Masukkan nama file bookdescriptions.

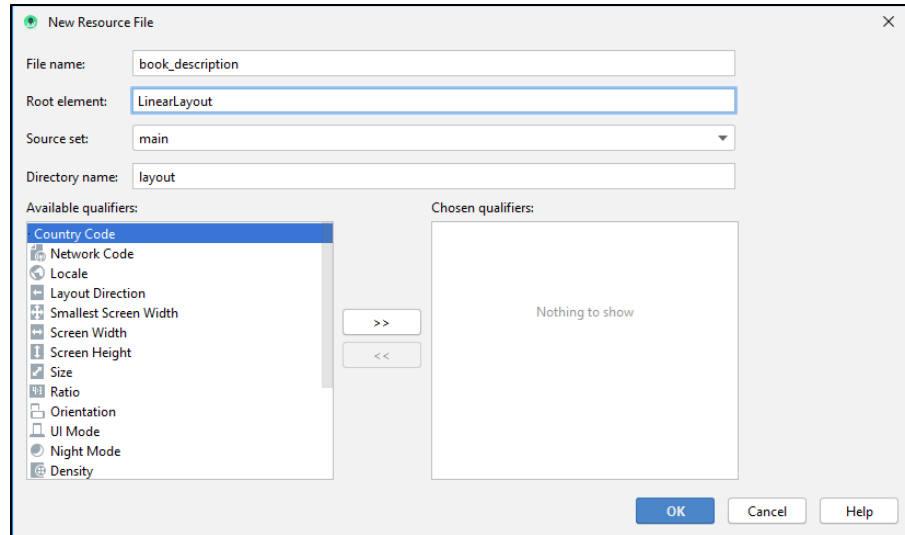


9. Tambahkan kode string array dari file bookdescriptions.xml menjadi seperti berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="bookdescriptions">
        <item>
            ""How to use Android Studio 3, but also teaches you how basic
            Android programming. And hey, in case you're also a beginner in Java,
            that's covered too.""
        </item>
        <item>
            ""This book is also about how to use Android Studio. Like the
            first one, it also teaches you the basics of the IDE and Android
            programming; but this time around, you'll use Kotlin. The newest kid in
            the JVM block""
        </item>
        <item>
            ""Minimum Android Programming is the book that got me
            started. I wrote in an age when even the Eclipse ADT doesn't exist yet.
            So, this means you'll use the Android SDK in all the glory of the CLI
            tools""
        </item>
    </string-array>
</resources>
```

Perlu diperhatikan bahwa setiap item pada string-array ditulis hanya dalam satu baris tidak menggunakan enter `<item>"" ... ""</item>`.

10. Selanjutnya buatlah layout fragment dengan cara klik kanan pada folder layout → New → Layout Resource File. Masukkan nama file book_description dan elemen root Linear Layout.



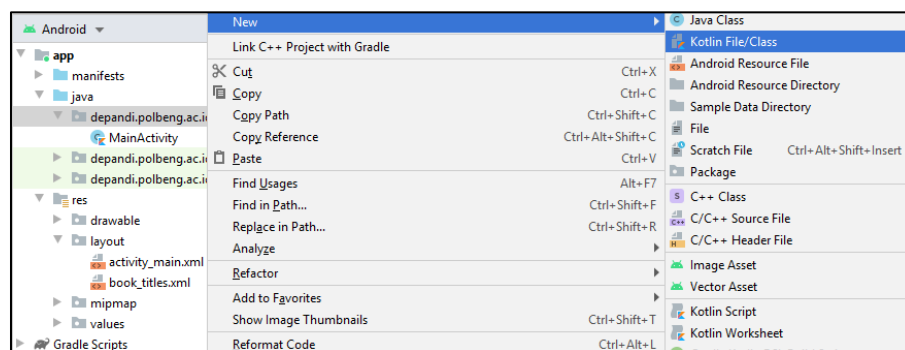
11. Tambahkan kode layout dari file book_description.xml menjadi seperti berikut:

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

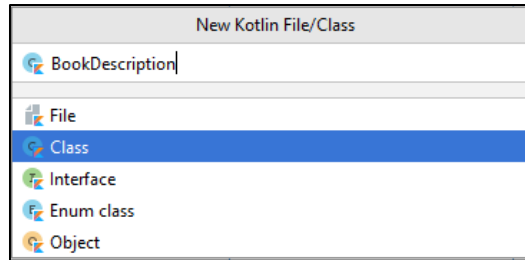
    <TextView
        android:id="@+id/txtdescription"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="TextView"
        android:textSize="24sp" />

</LinearLayout>
```

12. Buatlah kelas baru dengan cara klik kanan pada paket project anda → New → Klik Kotlin File Class.



13. Pada pilihan dialog mengambang pastikan anda memilih jenis Class dan masukkan nama kelas BookDescription.



14. Bukalah file BookDescription.kt yang telah dibuat, tambahkan baris kode berikut:

```
package depandi.polbeng.ac.id.fragmentexample1

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import kotlinx.android.synthetic.main.book_description.*

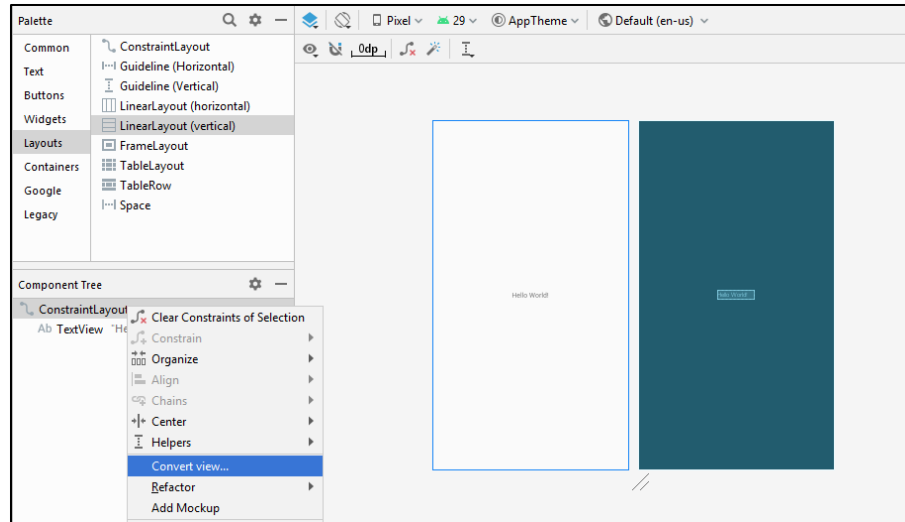
class BookDescription : Fragment() {

    lateinit var arrbookdesc: Array<String>
    var bookindex = 0
    override fun onCreateView(
        inflater: LayoutInflater, container:
        ViewGroup?, savedInstanceState: Bundle?
    ): View? {
        val v = inflater.inflate(R.layout.book_description, container,
false)
        arrbookdesc = resources.getStringArray(R.array.bookdescriptions)
        return v
    }

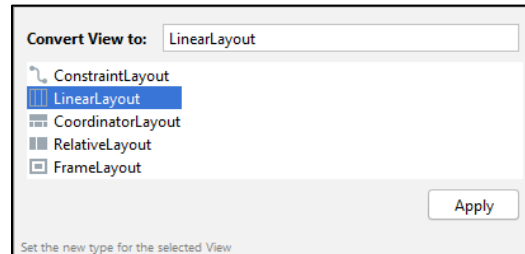
    fun changeDescription(index: Int): Unit {
        bookindex = index
        txtdescription?.setText(arrbookdesc[bookindex])
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        changeDescription(bookindex)
    }
}
```

15. Konversikan jenis root layout pada file activity_main.xml kedalam LinearLayout, caranya adalah bukalah layout activity_main.xml kedalam mode design. Bukalah side bar Pallette dan Component Tree. Klik kanan hirarki root ConstraintLayout → klik Convert view...



16. Pada kotak dialog Convert View to: Pilihlah opsi LinearLayout dan klik Apply.



17. Tambahkan kode layout dari file activity_main.xml menjadi seperti berikut:

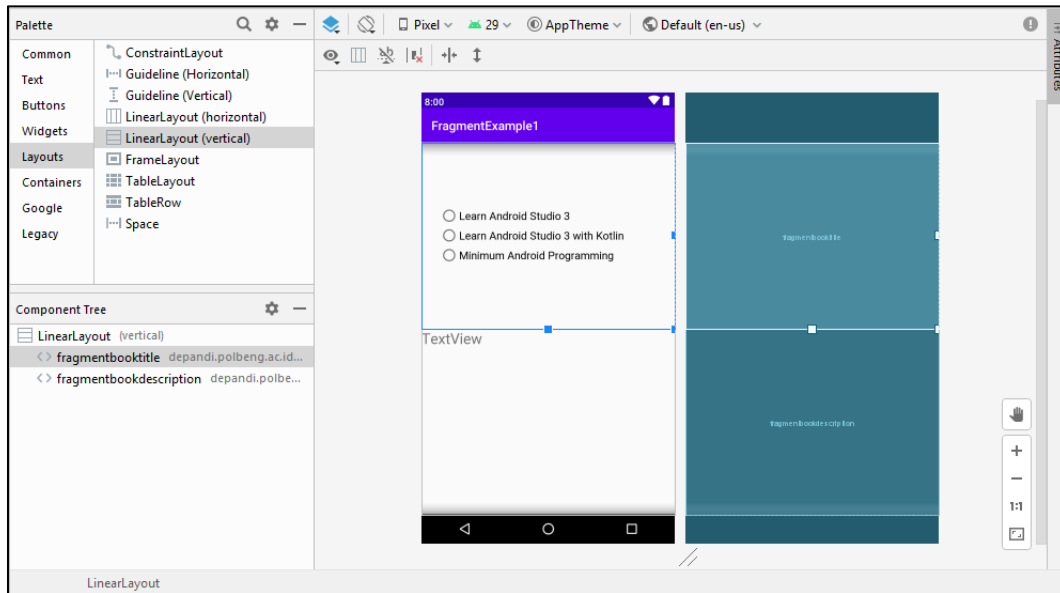
```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <fragment
        android:id="@+id/fragmentbooktitle"
        android:name="depandi.polbeng.ac.id.fragmentexample1.BookTitle"
        android:layout_width="match_parent"
        android:layout_height="0px"
        android:layout_weight="1"
        tools:layout="@layout/book_titles" />

    <fragment
        android:id="@+id/fragmentbookdescription"
        android:name="depandi.polbeng.ac.id.fragmentexample1.BookDescription"
        android:layout_width="match_parent"
        android:layout_height="0px"
        android:layout_weight="1"
        tools:layout="@layout/book_description" />

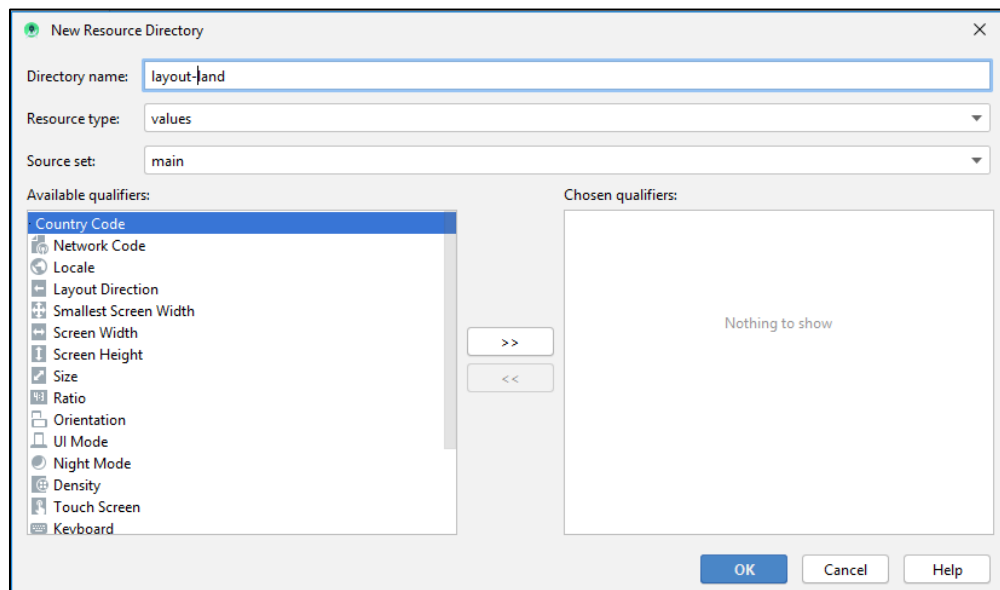
</LinearLayout>
```

Cobalah untuk melihat tampilan layout kedalam mode *design*. Hasilnya adalah seperti berikut:

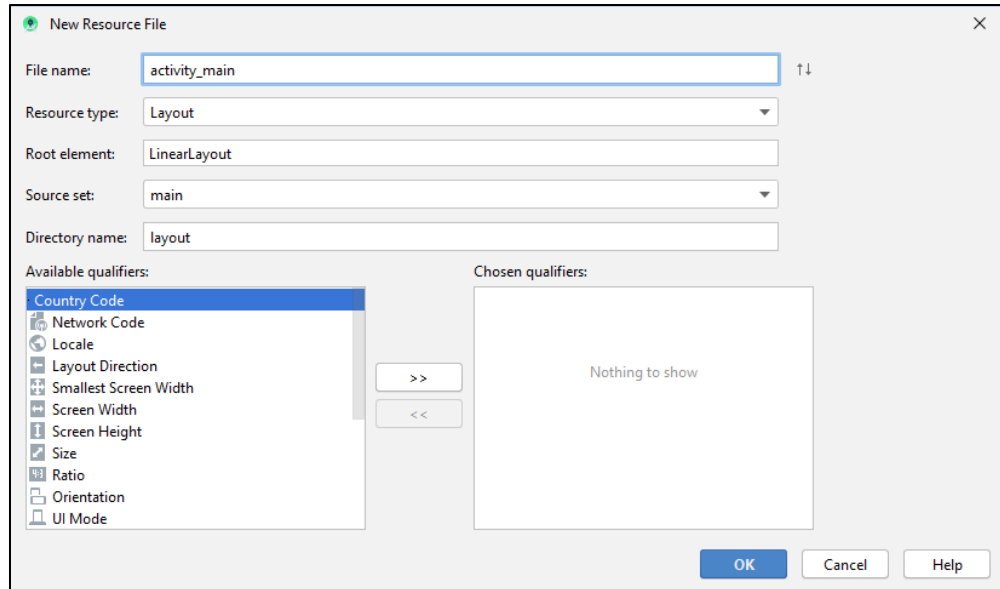


18. Secara default pengaturan orientasi pada langkah-langkah sebelumnya ialah dalam menangani dalam mode potrait. Sekarang untuk bekerja dalam mode landscape kita perlu untuk membuat tampilan tambahan seperti berikut:

1. Membuat folder app/res/layout-land



2. Membuat file sumberdaya UI lainnya didalam folder layout-land; kita juga beri namanya dengan activity_main.xml



3. Salinlah isi dari file `app/res/layout/activity_main.xml` ke `app/res/layout-land/activity_main.xml`

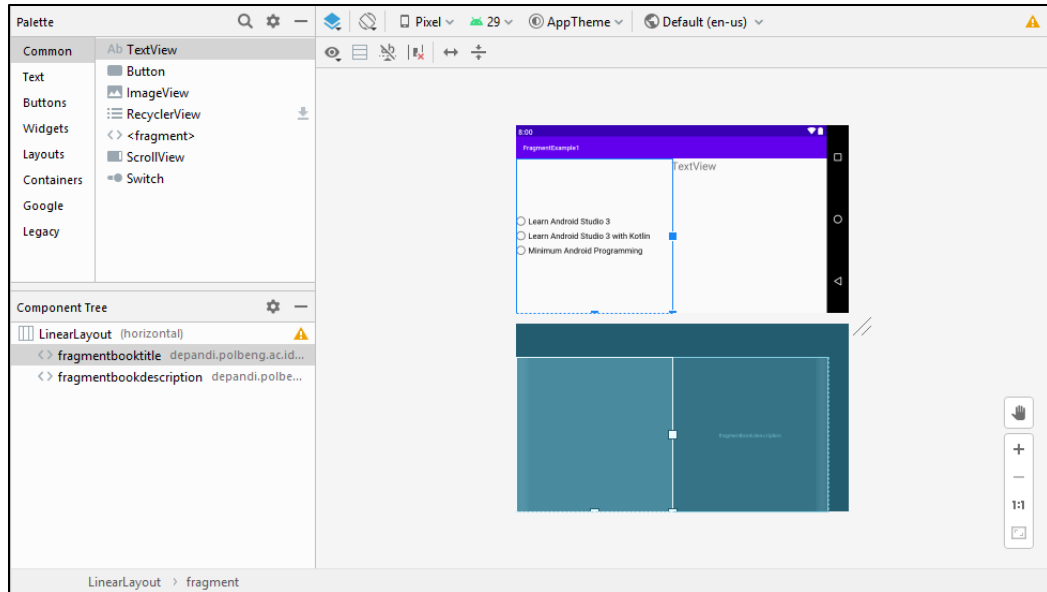
```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    tools:context=".MainActivity">

    <fragment
        android:id="@+id/fragmentbooktitle"
        android:name="depandi.polbeng.ac.id.fragmentexample1.BookTitle"
        android:layout_width="0px"
        android:layout_height="match_parent"
        android:layout_weight="1"
        tools:layout="@layout/book_titles" />

    <fragment
        android:id="@+id/fragmentbookdescription"
        android:name="depandi.polbeng.ac.id.fragmentexample1.BookDescription"
        android:layout_width="0px"
        android:layout_height="match_parent"
        android:layout_weight="1"
        tools:layout="@layout/book_description" />

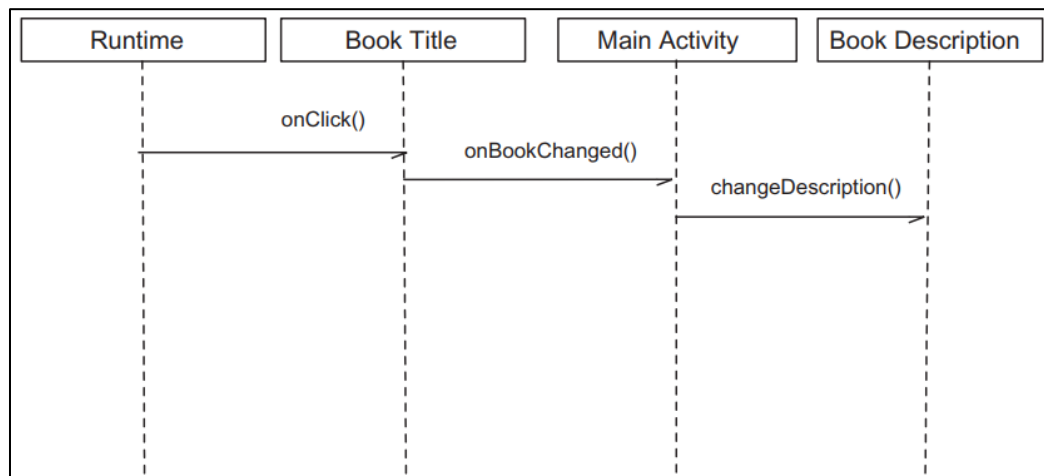
</LinearLayout>
```

4. Buatlah perubahan orientasi yang dibutuhkan dalam file `app/res/layout-land/activity_main`. Cobalah untuk melihat tampilan layout kedalam mode *design*. Hasilnya adalah seperti berikut:



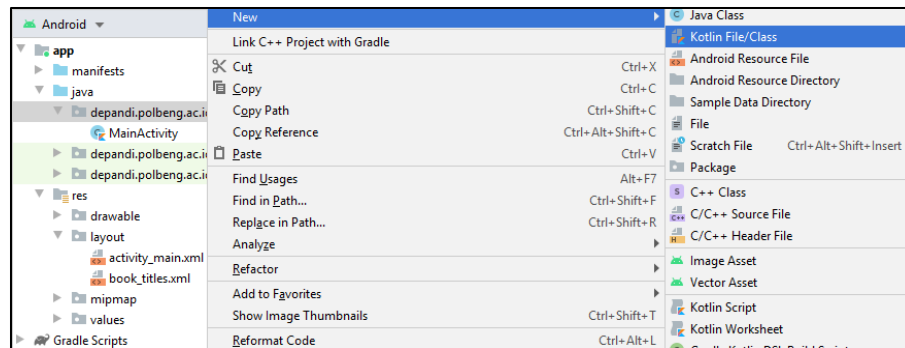
19. Langkah berikutnya adalah melakukan sinkronisasi fragment. Ketika pengguna meng-klik salah satu dari radio button dalam fragment book_titles maka fragment book_description juga harus berubah dan menampilkan deskripsi buku yang sedang dipilih.

Berikut adalah diagram urutan komunikasi antar fragment.

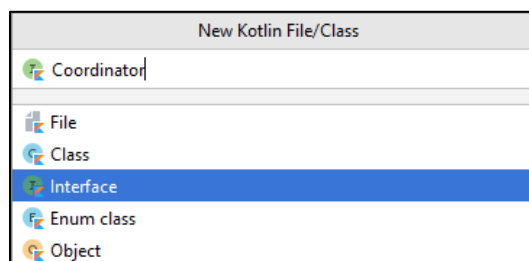


Pada diagram urutan BookTitle memanggil fungsi onBookChanged pada MainActivity, kemudian MainActivity memanggil fungsi changeDescription dalam kelas BookDescription. Maka kita membutuhkan sebuah interface agar derajat kohesi berkurang antar kelas.

20. Buatlah sebuah interface dengan nama Coordinator dengan cara klik kanan pada paket projet → New → Klik Kotlin File/Class.



21. Pada pilihan dialog mengambang pastikan anda memilih jenis Interface dan masukkan nama interface Coordinator.



22. Bukalah file Coordinator.kt yang telah dibuat, tambahkan baris kode berikut:

```
package depandi.polbeng.ac.id.fragmentexample1

interface Coordinator {
    fun onBookChanged(index: Int)
}
```

23. Selanjutnya buatlah impementasi interface yang telah dibuat pada kelas MainActivity.kt.

```
package depandi.polbeng.ac.id.fragmentexample1

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity(), Coordinator {

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

    override fun onBookChanged(index: Int) {
        val frag = fragmentbookdescription
        if (frag is BookDescription) {
            frag.changeDescription(index)
        }
    }
}
```

24. Pada kelas BookTitle berikan akses referensi ke kelas MainActivity dengan menggunakan interface Coordinator, selanjutnya memanggil metode onBookChanged. Sehingga kode BookTitle pada fungsi onClick menjadi seperti berikut:

```
package depandi.polbeng.ac.id.fragmentexample1

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import kotlinx.android.synthetic.main.book_titles.*

class BookTitle : Fragment(), View.OnClickListener {

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val v = inflater.inflate(R.layout.book_titles, container, false)
        return v
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        rlas3.setOnClickListener(this)
        rlas3kotlin.setOnClickListener(this)
        rminandroid.setOnClickListener(this)
    }

    override fun onClick(v: View?) {
        var index: Int = 0
        when (v?.id) {
            R.id.rlas3 -> {
                index = 0
            }
            R.id.rlas3kotlin -> {
                index = 1
            }
            R.id.rminandroid -> {
                index = 2
            }
        }
        val activity = getActivity()
        if (activity is Coordinator) {
            activity.onBookChanged(index)
        }
    }
}
```

25. Terakhir ubahlah kode BookDescription menjadi seperti berikut :

```
package depandi.polbeng.ac.id.fragmentexample1

import android.os.Bundle
import android.view.LayoutInflater
```

```

import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import kotlinx.android.synthetic.main.book_description.*

class BookDescription : Fragment() {

    lateinit var arrbookdesc: Array<String>
    var bookindex = 0

    override fun onCreateView(
        inflater: LayoutInflater, container:
        ViewGroup?, savedInstanceState: Bundle?
    ): View? {
        val v = inflater.inflate(R.layout.book_description, container,
false)
        arrbookdesc = resources.getStringArray(R.array.bookdescriptions)
        bookindex = if(savedInstanceState?.getInt("bookindex") == null) 0
        else { savedInstanceState.getInt("bookindex") }
        return v
    }

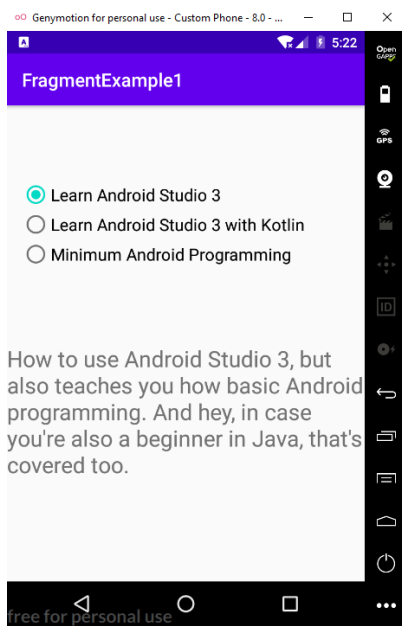
    override fun onSaveInstanceState(outState: Bundle) {
        outState.putInt("bookindex", bookindex)
    }

    fun changeDescription(index: Int) : Unit {
        bookindex = index
        println("BOOK INDEX = $bookindex")
        txtdescription?.setText(arrbookdesc[bookindex])
        println(arrbookdesc[bookindex])
    }

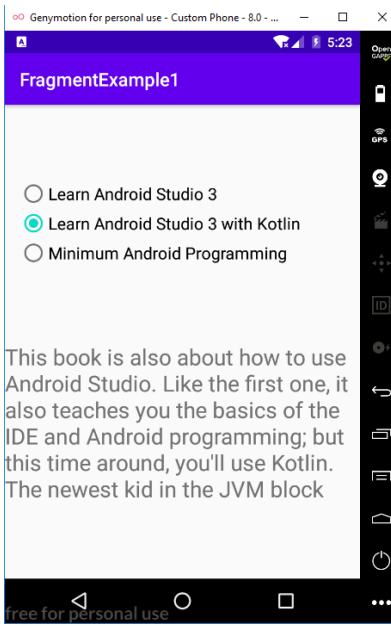
    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        changeDescription(bookindex)
    }
}

```

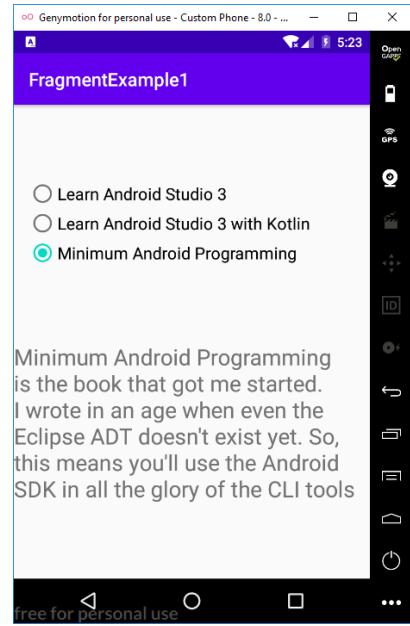
26. Jalankan program di Emulator Android dalam mode potrait. Cobalah untuk menganalisa program yang telah dibuat dan masukkan kedalam laporan praktikum.



Tampilan Pilihan 1 Potrait

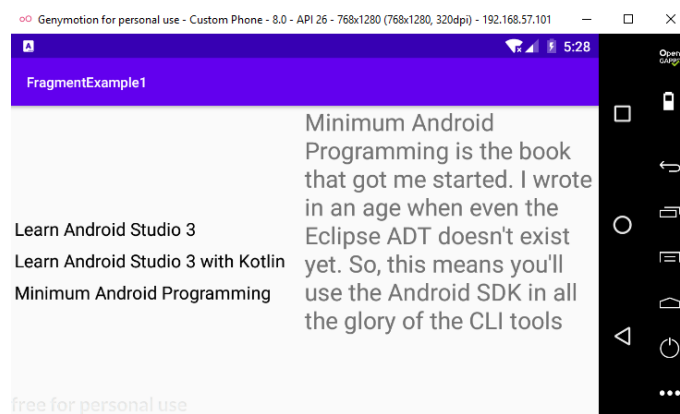


Tampilan Pilihan 2 Potrait

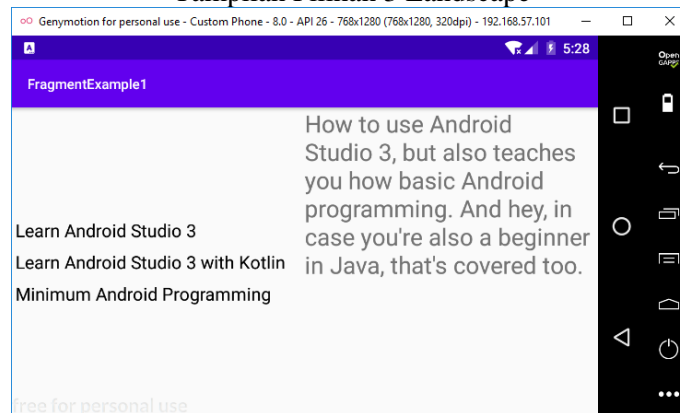


Tampilan Pilihan 3 Potrait

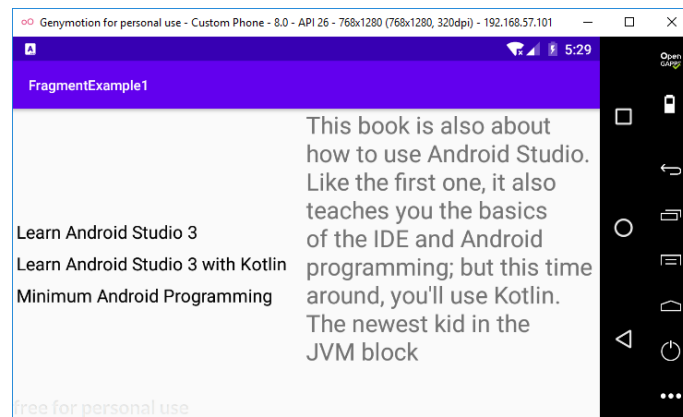
27. Cobalah untuk mengubah orientasi emulator Android menjadi Landscape. Perhatikan perubahan program, analisislah apa yang terjadi dan buat kesimpulan terhadap hasil analisa dan masukkan kedalam laporan praktikum.



Tampilan Pilihan 3 Landscape



Tampilan Pilihan 1 Landscape



Tampilan Pilihan 2 Landscape

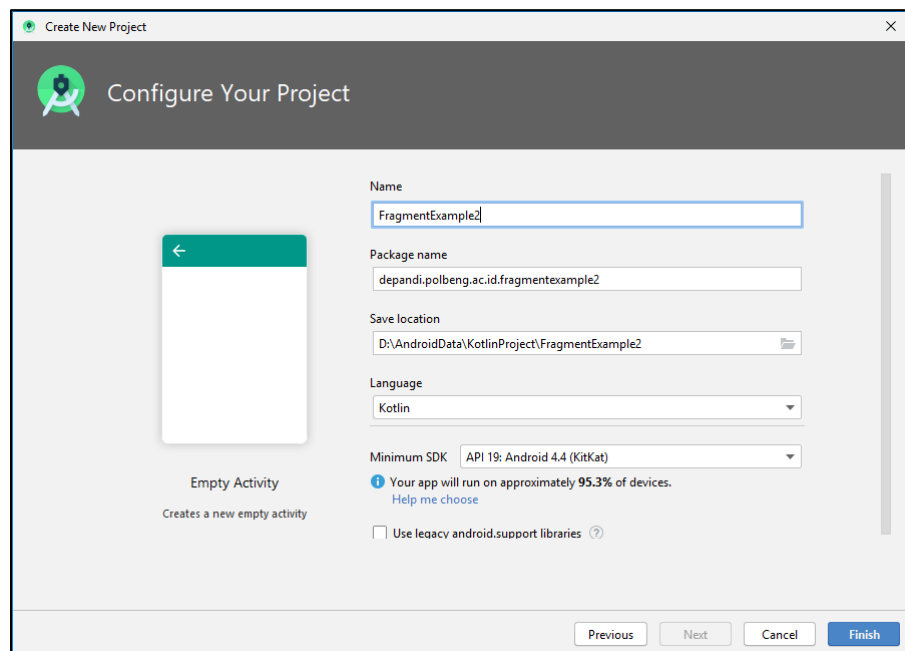
Implementasi Fragment Dinamis di Kotlin

Deskripsi Project

Merupakan project yang sama dengan project *fragment* statis pertama diatas. Tetapi terdapat sedikit perbedaan terhadap file MainActivity.kt dan activity_main.xml.

Langkah-langkah pembuatan project.

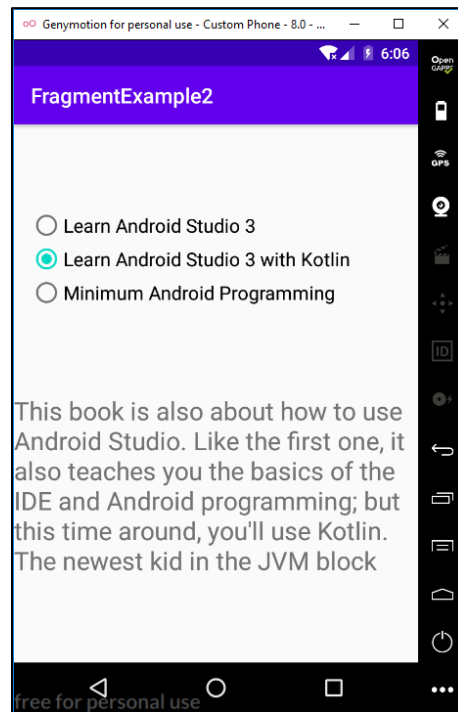
1. Buatlah *project* baru dengan konfigurasi seperti berikut :



2. Setelah selesai membuat *project* baru, salinlah seluruh file yang ada pada paket depandi.polbeng.ac.id.fragmentexample1. Isi file yang disalin terdiri dari beberapa file seperti pada tabel berikut ini.

Direktori dan Nama File
app/src/main/java/depandi.polbeng.ac.id.fragmentexample1/MainActivity.kt
app/src/main/java/depandi.polbeng.ac.id.fragmentexample1/BookDescription.kt
app/src/main/java/depandi.polbeng.ac.id.fragmentexample1/BookTitle.kt
app/src/main/java/depandi.polbeng.ac.id.fragmentexample1/Coordinator.kt
app/src/main/res/layout/activity_main.xml
app/src/main/res/layout/book_description.xml
app/src/main/res/layout/book_titles.xml
app/src/main/res/layout-land/activity_main.xml
app/src/main/res/values/bookdescriptions.xml
app/src/main/res/values/strings.xml

3. Pastikan semua file telah disalin dengan benar. Jalankan program yang telah diduplikat di Emulator Android. Pastikan program berjalan dengan baik.



4. Ubahlah isi file activity_main.xml pada folder layout. Sehingga menjadi seperti berikut:

```
<?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">

    <LinearLayout
        android:id="@+id/fragtop"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
```

```

        android:orientation="horizontal"/>

<LinearLayout
    android:id="@+id/fragbottom"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:orientation="horizontal"/>

</LinearLayout>

```

5. Ubahlah isi file activity_main.xml pada folder layout-land. Sehingga menjadi seperti berikut:

```

<?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="horizontal"
    tools:context=".MainActivity">

    <LinearLayout
        android:id="@+id/fragtop"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="vertical"/>

    <LinearLayout
        android:id="@+id/fragbottom"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="vertical"/>

</LinearLayout>

```

6. Terakhir ubahlah isi file MainActivity sehingga menjadi seperti berikut :

```

package depandi.polbeng.ac.id.fragmentexample2

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity(), Coordinator {

    lateinit var fragBookDescription: BookDescription
    lateinit var fragBookTitle: BookTitle

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

        fragBookTitle = BookTitle()
        fragBookDescription = BookDescription()
        val fragTransaction = supportFragmentManager.beginTransaction()
        fragTransaction.add(R.id.fragtop, fragBookTitle)
        fragTransaction.add(R.id.fragbottom, fragBookDescription)
        fragTransaction.commit()
    }
}

```

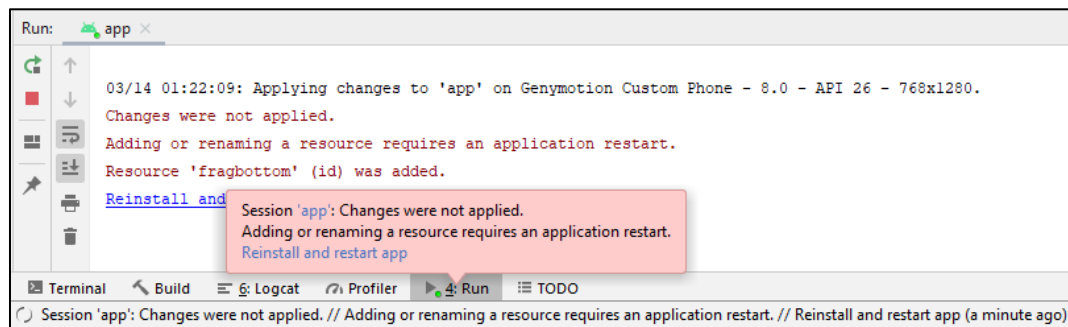
```

    }

    override fun onBookChanged(index:Int) {
        fragBookDescription.changeDescription(index)
    }
}

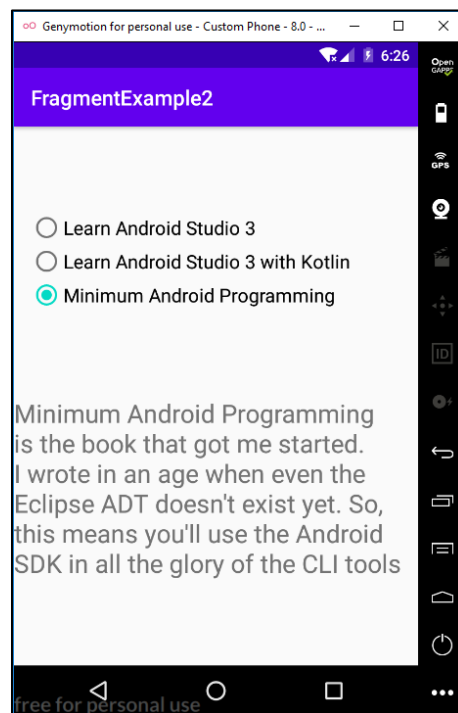
```

7. Jalankan program di Emulator Android dalam mode *potrait* dan *landscape*. Cobalah untuk menganalisa program yang telah dibuat dan masukkan kedalam laporan praktikum.

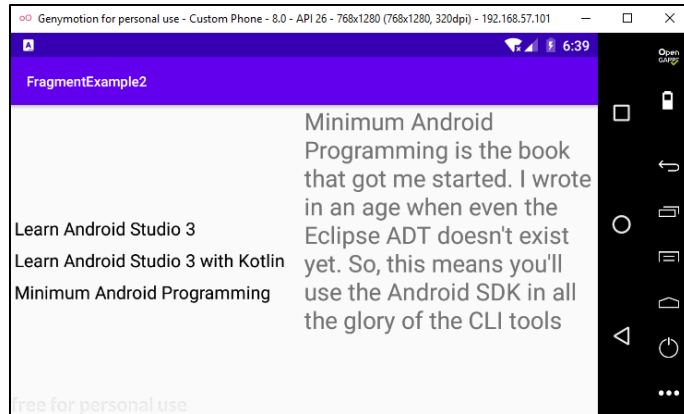


Jika terdapat pesan peringatan seperti diatas maka tutuplah program yang sedang berjalan terlebih dahulu dan klik reinstall aplikasi. Jika tidak ada pesan peringatan maka akan tampil hasil program seperti berikut:

Tampilan dalam mode *potrait* :



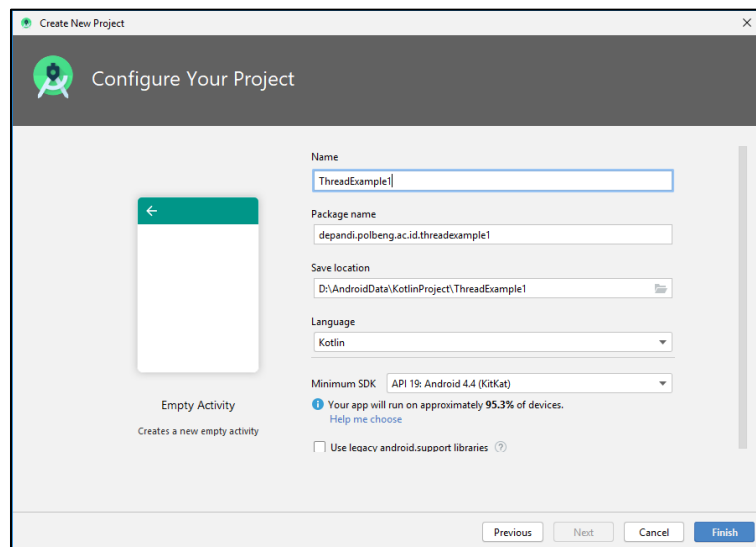
Tampilan dalam mode *landscape* :



Background Task

Main Thread (UI Thread)

1. Buatlah *project* baru dengan konfigurasi seperti berikut :



2. Setelah selesai membuat project baru, selanjutnya editlah file *strings.xml* pada folder *values*, tambahkan beberapa variabel dan nilai *string* sebagai berikut:

```
<resources>
    <string name="app_name">ThreadExample1</string>
    <string name="start">Start</string>
</resources>
```

3. Editlah file *activity_main.xml* sehingga kode *layout* menjadi seperti berikut :

```
<?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"
```

```

        android:gravity="center_vertical|center_horizontal"
        tools:context=".MainActivity">

        <Switch
            android:id="@+id/switchUI"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginBottom="16dp" />

        <Button
            android:id="@+id/btnStartThread"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="startThread"
            android:text="@string/start" />

    </LinearLayout>

```

4. Bukalah file sumber *MainActivity.kt* editlah kode program hingga menjadi seperti berikut :

```

package depandi.polbeng.ac.id.threadexample1

import android.os.Bundle
import android.util.Log
import android.view.View
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

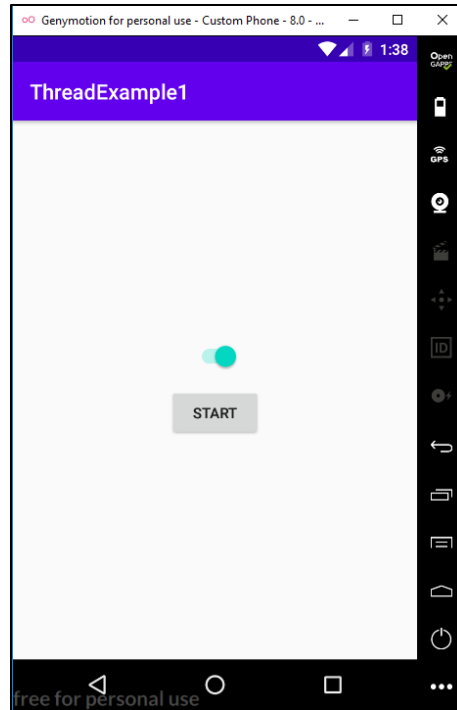
    val TAG = "MainActivity"

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

    fun startThread(view: View): Unit {
        for (i in 1..10) {
            Log.d(TAG, "startThread: $i")
            try {
                Thread.sleep(1000)
            } catch (ie: InterruptedException) {
                ie.printStackTrace()
            }
        }
    }
}

```

5. Jalankan program dan perhatikan hasilnya. Apakah bisa merubah switch UI pada saat thread dijalankan dengan menekan tombol Start.



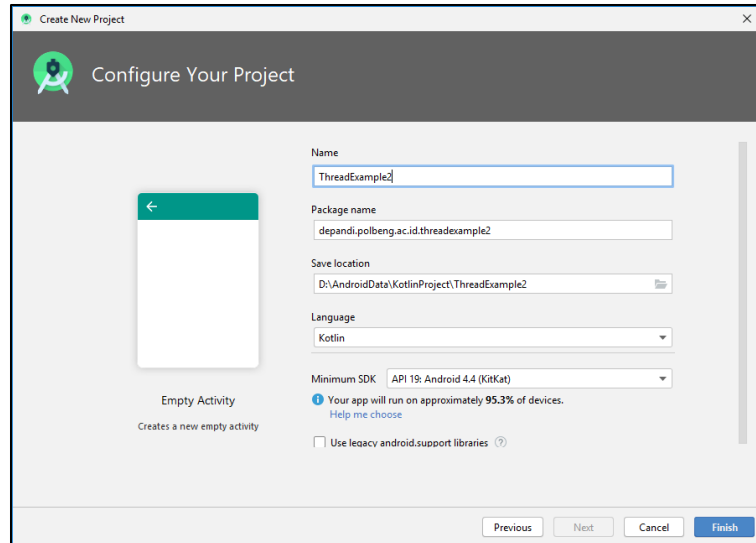
Hasil rekaman Logcat:

```

logcat
2020-03-14 20:37:54.618 3709-3709/depandi.polbeng.ac.id.threadexample1 D/MainActivity: startThread: 1
2020-03-14 20:37:55.618 3709-3709/depandi.polbeng.ac.id.threadexample1 D/MainActivity: startThread: 2
2020-03-14 20:37:56.621 3709-3709/depandi.polbeng.ac.id.threadexample1 D/MainActivity: startThread: 3
2020-03-14 20:37:57.621 3709-3709/depandi.polbeng.ac.id.threadexample1 D/MainActivity: startThread: 4
2020-03-14 20:37:58.622 3709-3709/depandi.polbeng.ac.id.threadexample1 D/MainActivity: startThread: 5
2020-03-14 20:37:59.623 3709-3709/depandi.polbeng.ac.id.threadexample1 D/MainActivity: startThread: 6
2020-03-14 20:38:00.625 3709-3709/depandi.polbeng.ac.id.threadexample1 D/MainActivity: startThread: 7
2020-03-14 20:38:01.626 3709-3709/depandi.polbeng.ac.id.threadexample1 D/MainActivity: startThread: 8
2020-03-14 20:38:02.627 3709-3709/depandi.polbeng.ac.id.threadexample1 D/MainActivity: startThread: 9
2020-03-14 20:38:03.627 3709-3709/depandi.polbeng.ac.id.threadexample1 D/MainActivity: startThread: 10
2020-03-14 20:38:04.651 3709-3709/depandi.polbeng.ac.id.threadexample1 I/Choreographer: Skipped 603 frames! The application may be doing too much work on its main thread.
  
```

Implementasi Thread

1. Buatlah *project* baru dengan nama ThreadExample2 sesuai dengan konfigurasi berikut.



- Setelah selesai membuat project baru, selanjutnya editlah file *strings.xml* pada folder *values*, tambahkan beberapa variabel dan nilai *string* sebagai berikut:

```
<resources>
    <string name="app_name">ThreadExample1</string>
    <string name="start">Start</string>
</resources>
```

- Editlah file *activity_main.xml* sehingga kode *layout* menjadi seperti berikut :

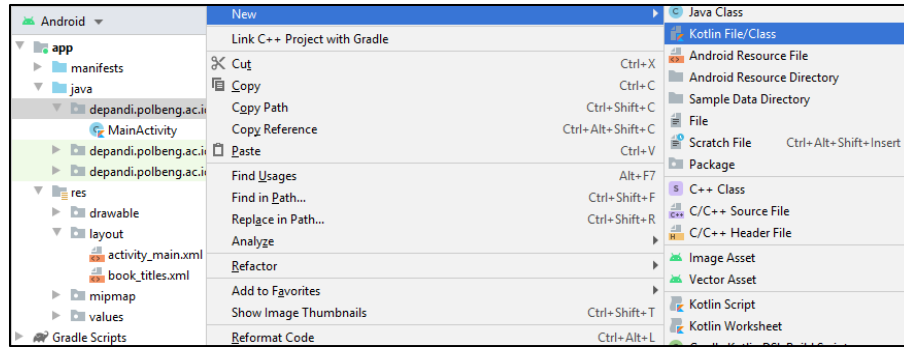
```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center_vertical|center_horizontal"
    tools:context=".MainActivity">

    <Switch
        android:id="@+id/switchUI"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp" />

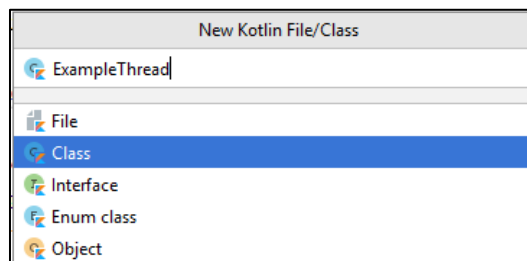
    <Button
        android:id="@+id/btnStartThread"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="startThread"
        android:text="@string/start" />

</LinearLayout>
```

- Buatlah kelas baru dengan cara klik kanan pada paket project anda → New → Klik Kotlin File Class.



5. Pada pilihan dialog mengambang pastikan anda memilih jenis Class dan masukkan nama kelas ExampleThread.



6. Bukalah file ExampleThread.kt yang telah dibuat, tambahkan baris kode berikut:

```
package depandi.polbeng.ac.id.threadexample2

import android.util.Log;

class ExampleThread : Thread{

    var seconds = 0
    val TAG = "MainActivity"

    constructor(seconds: Int){
        this.seconds = seconds
    }

    override fun run() {
        for (i in 1 until seconds) {
            Log.d(TAG, "startThread: $i")
            try {
                Thread.sleep(1000)
            } catch (ie: InterruptedException) {
                ie.printStackTrace()
            }
        }
    }
}
```

7. Bukalah file sumber MainActivity.kt editlah kode program hingga menjadi seperti berikut :

```
package depandi.polbeng.ac.id.threadexample2

import android.os.Bundle
import android.view.View
```

```

import androidx.appcompat.app.AppCompatActivity

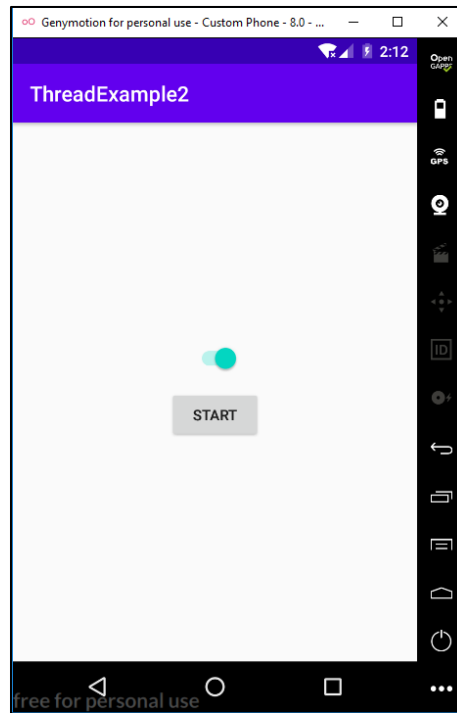
class MainActivity : AppCompatActivity() {

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

    fun startThread(view: View):Unit{
        val runnable = ExampleThread(10)
        Thread(runnable).start()
    }
}

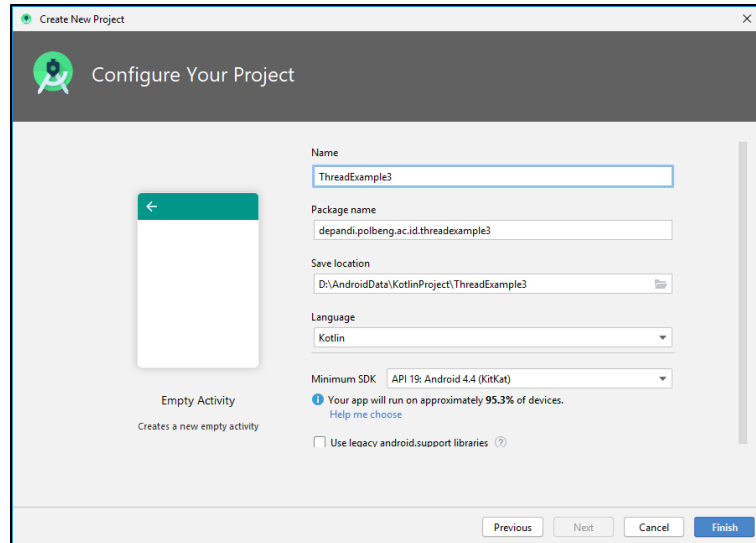
```

8. Jalankan program dan perhatikan hasilnya. Apakah bisa merubah switch UI pada saat thread dijalankan dengan menekan tombol Start.



Implementasi Runnable

1. Buatlah *project* baru dengan nama ThreadExample3 sesuai dengan konfigurasi berikut.



- Setelah selesai membuat project baru, selanjutnya editlah file *strings.xml* pada folder *values*, tambahkan beberapa variabel dan nilai *string* sebagai berikut:

```
<resources>
    <string name="app_name">ThreadExample1</string>
    <string name="start">Start</string>
</resources>
```

- Editlah file *activity_main.xml* sehingga kode *layout* menjadi seperti berikut :

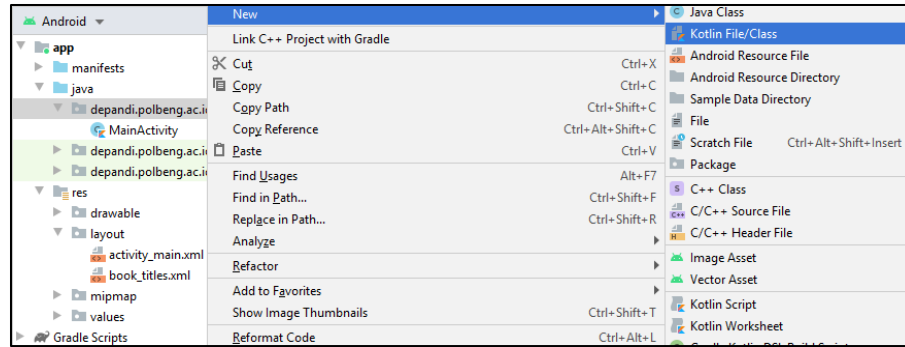
```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center_vertical|center_horizontal"
    tools:context=".MainActivity">

    <Switch
        android:id="@+id/switchUI"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp" />

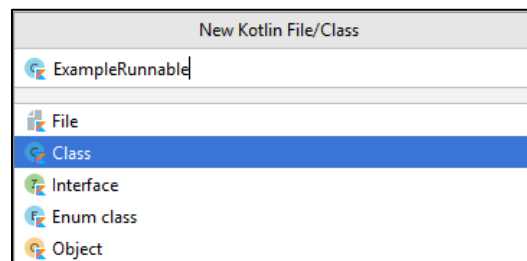
    <Button
        android:id="@+id/btnStartThread"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="startThread"
        android:text="@string/start" />

</LinearLayout>
```

- Buatlah kelas baru dengan cara klik kanan pada paket project anda → New → Klik Kotlin File Class.



5. Pada pilihan dialog mengambang pastikan anda memilih jenis Class dan masukkan nama kelas ExampleRunnable.



6. Bukalah file ExampleRunnable.kt yang telah dibuat, tambahkan baris kode berikut:

```
package depandi.polbeng.ac.id.threadexample3

import android.util.Log

class ExampleRunnable : Runnable {

    var seconds = 0
    val TAG = "MainActivity"

    constructor(seconds: Int) {
        this.seconds = seconds
    }

    override fun run() {
        for (i in 1 until seconds) {
            Log.d(TAG, "startThread: $i")
            try {
                Thread.sleep(1000)
            } catch (ie: InterruptedException) {
                ie.printStackTrace()
            }
        }
    }
}
```

7. Bukalah file sumber MainActivity.kt editlah kode program hingga menjadi seperti berikut :

```
package depandi.polbeng.ac.id.threadexample3

import android.os.Bundle
import android.view.View
```

```

import androidx.appcompat.app.AppCompatActivity

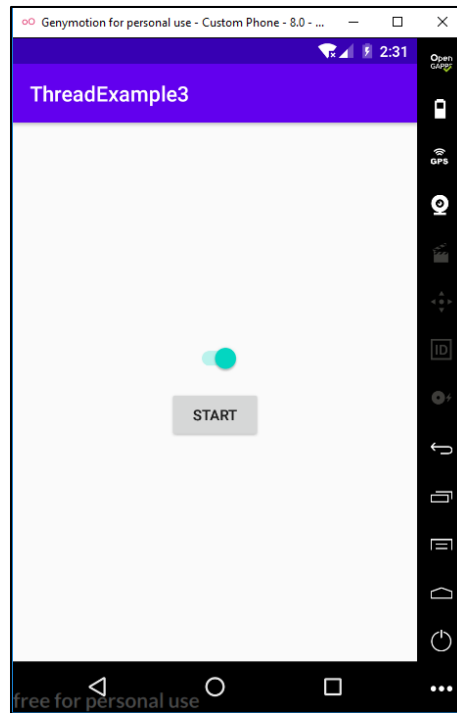
class MainActivity : AppCompatActivity() {

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

    fun startThread(view: View):Unit{
        val runnable = ExampleRunnable(10);
        Thread(runnable).start()
    }
}

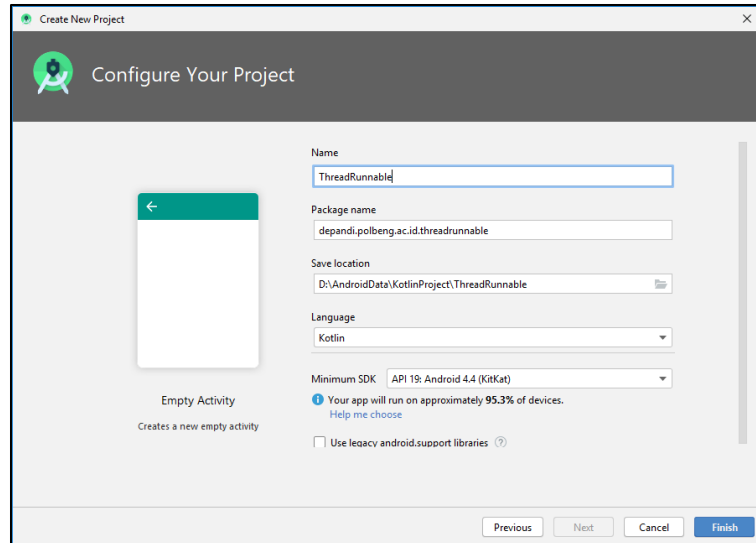
```

8. Jalankan program dan perhatikan hasilnya. Apakah bisa merubah switch UI pada saat thread dijalankan dengan menekan tombol Start.



Implementasi Thread dan Runnable (Simple Thread)

1. Buatlah *project* baru dengan konfigurasi seperti berikut :



- Setelah selesai membuat project baru, selanjutnya editlah file *strings.xml* pada folder *values*, tambahkan beberapa variabel dan nilai *string* sebagai berikut:

```
<resources>
    <string name="app_name">ThreadRunnable</string>
    <string name="textview">TextView</string>
    <string name="button">Button</string>
</resources>
```

- Editlah file *activity_main.xml* sehingga kode *layout* menjadi seperti berikut :

```
<?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"
    android:gravity="center_vertical|center_horizontal"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/textview"
        android:textSize="30sp"
        android:layout_marginBottom="30dp"/>

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button"/>

</LinearLayout>
```

- Bukalah file sumber *MainActivity.kt* editlah kode program hingga menjadi seperti berikut :

```
package depandi.polbeng.ac.id.threadrunnable
```

```

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

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

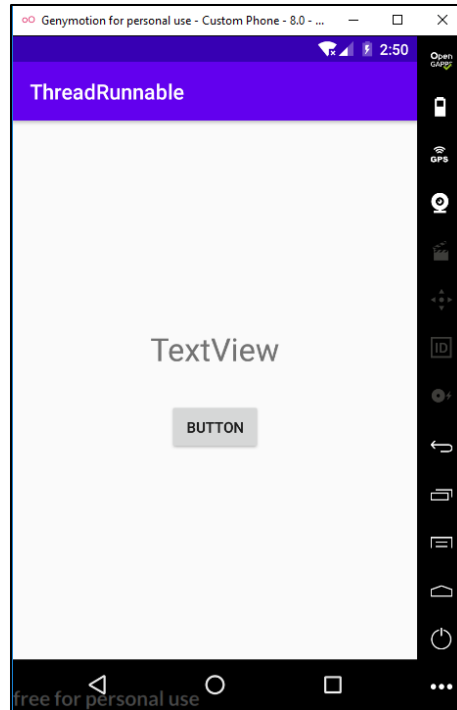
        button.setOnClickListener {
            val runnable = Worker()
            val thread = Thread(runnable)
            thread.start()
        }

        inner class Worker : Runnable {
            override fun run() {
                killSomeTime()
            }
        }

        private fun killSomeTime() {
            for (i in 1..20) {
                Thread.sleep(2000)
                println("i: $i")
            }
        }
    }
}

```

5. Jalankan program di Emulator Android. Cobalah untuk menganalisa program yang telah dibuat dan masukkan kedalam laporan praktikum.



Hasil rekaman Logcat:

```

logcat
2020-03-14 21:48:50.772 4729-4787/depandi.polbeng.ac.id.threadrunnable I/System.out: i: 5
2020-03-14 21:48:52.807 4729-4787/depandi.polbeng.ac.id.threadrunnable I/System.out: i: 6
2020-03-14 21:48:54.853 4729-4787/depandi.polbeng.ac.id.threadrunnable I/System.out: i: 7
2020-03-14 21:48:56.879 4729-4787/depandi.polbeng.ac.id.threadrunnable I/System.out: i: 8
2020-03-14 21:48:58.880 4729-4787/depandi.polbeng.ac.id.threadrunnable I/System.out: i: 9
2020-03-14 21:49:00.005 487-504/system_process E/memtrack: Couldn't load memtrack module
2020-03-14 21:49:00.006 487-504/system_process W/android.os.Debug: failed to get memory consumption info: -1
2020-03-14 21:49:00.672 1114-1213/com.genymotion.genyid W/GenydService: unknown message set_device_clipboard |
2020-03-14 21:49:00.884 4729-4787/depandi.polbeng.ac.id.threadrunnable I/System.out: i: 10
2020-03-14 21:49:02.884 4729-4787/depandi.polbeng.ac.id.threadrunnable I/System.out: i: 11
2020-03-14 21:49:04.885 4729-4787/depandi.polbeng.ac.id.threadrunnable I/System.out: i: 12

```

Mengubah UI melalui Thread

Thread latar belakang tidak mengizinkan kita untuk mengubah apapun dalam antarmuka pengguna (UI). Hanya UI Thread yang memiliki tanggung jawab untuk mengubah UI. Permasalahan yang perlu kita selesaikan pada project ThreadRunnable adalah bagaimana mengubah UI (mengupdate tampilan TextView) melalui thread latar belakang. Untuk menyelesaikan permasalahan tersebut kita gunakan metode `runOnUiThread()` dari kelas Activity untuk mengupdate TextView.

6. Ubahlah file sumber MainActivity.kt editlah kode program hingga menjadi seperti berikut :

```
package depandi.polbeng.ac.id.threadrunnable
```



```

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

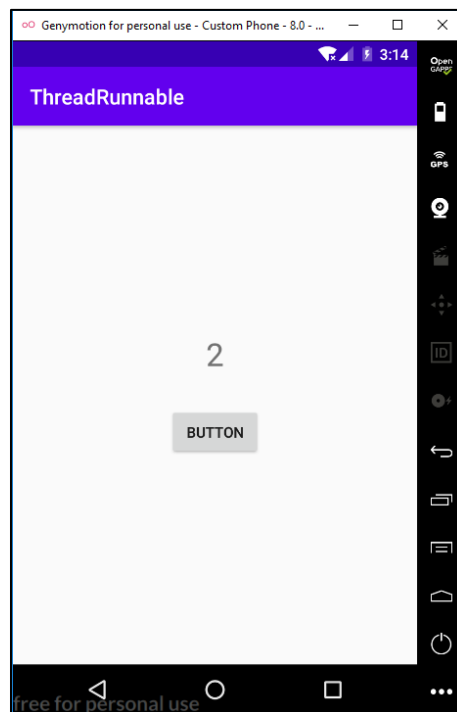
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        button.setOnClickListener {
            Thread(Runnable {
                killSomeTime()
            }).start()
        }

        private fun killSomeTime() {
            for (i in 1..20) {
                runOnUiThread(Runnable{
                    textView.text = i.toString()
                })
                println("i:$i")
                Thread.sleep(2000)
            }
        }
    }
}

```

7. Jalankan program di Emulator Android. Cobalah untuk menganalisa program yang telah dibuat dan masukkan kedalam laporan praktikum.



Hasil rekaman Logcat:

```
logcat
2020-03-14 22:14:49.056 5072-5189/depandi.polbeng.ac.id.threadrunnable I/System.out: i:10
2020-03-14 22:14:51.057 5072-5189/depandi.polbeng.ac.id.threadrunnable I/System.out: i:11
2020-03-14 22:14:53.064 5072-5189/depandi.polbeng.ac.id.threadrunnable I/System.out: i:12
2020-03-14 22:14:55.065 5072-5189/depandi.polbeng.ac.id.threadrunnable I/System.out: i:13
2020-03-14 22:14:57.065 5072-5189/depandi.polbeng.ac.id.threadrunnable I/System.out: i:14
2020-03-14 22:14:59.066 5072-5189/depandi.polbeng.ac.id.threadrunnable I/System.out: i:15
2020-03-14 22:15:01.067 5072-5189/depandi.polbeng.ac.id.threadrunnable I/System.out: i:16
2020-03-14 22:15:03.068 5072-5189/depandi.polbeng.ac.id.threadrunnable I/System.out: i:17
2020-03-14 22:15:05.068 5072-5189/depandi.polbeng.ac.id.threadrunnable I/System.out: i:18
2020-03-14 22:15:07.069 5072-5189/depandi.polbeng.ac.id.threadrunnable I/System.out: i:19
2020-03-14 22:15:07.083 5072-5080/depandi.polbeng.ac.id.threadrunnable I/zygote: Do partial
2020-03-14 22:15:07.083 5072-5080/depandi.polbeng.ac.id.threadrunnable I/zygote: After code
```

Menggunakan Kelas Handler

Kelas Handler tidak seperti Thread, ia adalah bagian dari kerangka kerja Android, bukan bagian dari Java. Objek Handler berguna untuk mengatur Thread. Ide dasarnya adalah untuk memperoleh referensi Handler dari thread utama, kemudian didalam thread latar belakang (dimana kita tidak dapat membuat perubahan apapun terhadap UI), akan mengirim sebuah **Message** ke objek Handler. Gunakan objek **Message** untuk mengirim data diantara thread latar belakang dan thread utama.

Untuk menggunakan objek Handler, kita perlu melakukan beberapa hal berikut :

1. Mengambil objek Handler yang berhubungan dengan thread UI.
 2. Dimanapun didalam kode yang mungkin menyebabkan stutter/hang ketika dijalankan, maka masukkan ia ke thread latar belakang.
 3. Pada thread latar belakang ketika mau mengupdate UI lakukan hal berikut :
 - ✎ Membuat objek **Message**, cara terbaik adalah dengan memanggil fungsi **obtain()**.
 - ✎ Mengirim pesan ke objek Handler dengan memanggil fungsi **sendMessage**.
Objek Message dapat mengirim data. Atribut data dari objek Message adalah sebuah objek Bundle, jadi kita dapat menggunakan berbagai macam method seperti **putString**, **putInt**, **putBundle**, **putFloat** dan lain-lain.
 4. Kita dapat melakukan perubahan UI dalam fungsi callback **handleMessage** dari objek Handler.
8. Ubahlah file sumber MainActivity.kt editlah kode program hingga menjadi seperti berikut :

```
package depandi.polbeng.ac.id.threadrunnable

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Handler
```

```

import android.os.Message
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

    lateinit var mHandler: Handler

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

        mHandler = object : Handler() {
            override fun handleMessage(msg: Message?) {
                textView.text = msg?.data?.getString("counter")
            }
        }

        button.setOnClickListener {
            Thread(Runnable {
                killSomeTime()
            }).start()
        }

        private fun killSomeTime() {
            for (i in 1..20) {
                var msg = Message.obtain()
                msg.data.putString("counter", i.toString())
                mHandler.sendMessage(msg)
                Thread.sleep(2000)
            }
        }
    }
}

```

9. Jalankan program di Emulator Android. Cobalah untuk menganalisa program yang telah dibuat dan masukkan kedalam laporan praktikum.

AsyncTask

Cara lain untuk menjalankan tugas dilatar belakang adalah dengan menggunakan kelas AsyncTask. AsyncTask sama seperti kelas Handler adalah bagian dari kerangka kerja Android. Seperti Handler, ia juga memiliki mekanisme untuk melakukan pekerjaan di latar belakang, ia juga menyediakan sebuah cleaner untuk mengupdate UI.

Untuk menggunakan AsyncTask, kita perlu melakukan beberapa hal berikut :

1. Meng-extends kelas **AsyncTask**.
 2. Meng-override fungsi **doInBackground**, **onProgressUpdate**, **onPostExecute**.
 3. Membuat instansiasi dari subclass AsyncTask dan memanggil metode **execute** untuk mulai menjalankan tugas latar belakang.
10. Ubahlah file sumber MainActivity.kt editlah kode program hingga menjadi seperti berikut :

```

package depandi.polbeng.ac.id.threadrunnable

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.AsyncTask
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

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

        button.setOnClickListener {
            Worker().execute()
        }

        inner class Worker : AsyncTask<Void, String, Boolean>() {
            override fun doInBackground(vararg p0: Void?): Boolean {
                for (i in 1..20) {
                    publishProgress(i.toString())
                    Thread.sleep(2000)
                }
                return true
            }

            override fun onProgressUpdate(vararg values: String?) {
                textView.text = values[0]
            }

            override fun onPostExecute(result: Boolean?) {
                println(result)
            }
        }
    }
}

```

11. Jalankan program di Emulator Android. Cobalah untuk menganalisa program yang telah dibuat dan masukkan kedalam laporan praktikum.

Anko's doAsync

Anko adalah pustaka Android yang ditulis menggunakan Kotlin oleh JetBrains. Anko's doAsync menjalankan kode secara asinkron atau didalam latar belakang.

12. Sebelum menggunakan Anko, kita perlu menambahkan dependensi tambahan pada file Gradle project kita.

```

dependencies {
    ....
    implementation 'org.jetbrains.anko:anko-common:0.9'
}

```

13. Ubahlah file sumber MainActivity.kt editlah kode program hingga menjadi seperti berikut :

```

package depandi.polbeng.ac.id.threadrunnable

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_main.*
import org.jetbrains.anko.activityUiThread
import org.jetbrains.anko.doAsync

class MainActivity : AppCompatActivity() {

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

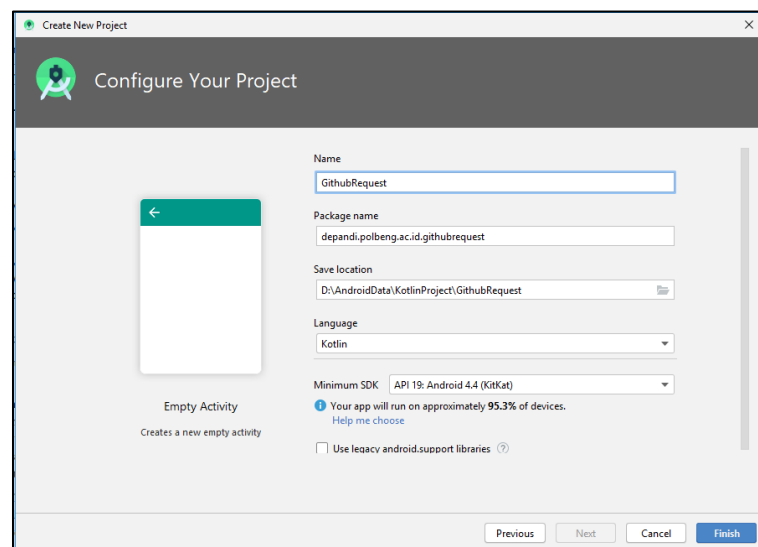
        button.setOnClickListener {
            doAsync {
                for(i in 1..15) {
                    Thread.sleep(2000)
                    activityUiThread {
                        textView.text = i.toString()
                    }
                }
            }
        }
    }
}

```

14. Jalankan program di Emulator Android. Cobalah untuk menganalisa program yang telah dibuat dan masukkan kedalam laporan praktikum.

Implementasi Tugas Latarbelakang (Request Github)

1. Buatlah *project* baru dengan konfigurasi seperti berikut :



2. Setelah selesai membuat project baru, selanjutnya editlah file *strings.xml* pada folder *values*, tambahkan beberapa variabel dan nilai *string* sebagai berikut:

```

<resources>
    <string name="app_name">GithubRequest</string>
    <string name="info_github_user">Info Github User</string>
    <string name="get_info">Get Info</string>
    <string name="username">Username</string>
</resources>

```

3. Editlah file *activity_main.xml* sehingga kode *layout* menjadi seperti berikut :

```

<?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">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:weightSum="1"
        android:padding="10dp">
        <EditText
            android:id="@+id/txtsearchuser"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:inputType="textPersonName"
            android:hint="@string/username"
            android:layout_weight="0.75"
            android:importantForAutofill="no" />
        <Button
            android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/get_info"
            android:layout_weight="0.25" />
    </LinearLayout>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="10dp">

        <TextView
            android:id="@+id/tvInfo"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/info_github_user"
            android:textSize="20sp" />

    </ScrollView>

</LinearLayout>

```

4. Tambahkan dua dependensi tambahan pada file Gradle project.

```

dependencies {
    ....
}

```

```
implementation 'com.squareup.okhttp:okhttp:2.5.0'
implementation 'org.jetbrains.anko:anko-common:0.9'
}
```

5. Bukalah file sumber *MainActivity.kt* editlah kode program hingga menjadi seperti berikut :

```
package depandi.polbeng.ac.id.githubrequest

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import com.squareup.okhttp.OkHttpClient
import com.squareup.okhttp.Request
import kotlinx.android.synthetic.main.activity_main.*
import org.jetbrains.anko.activityUiThread
import org.jetbrains.anko.doAsync
import org.json.JSONObject

class MainActivity : AppCompatActivity() {

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

        button.setOnClickListener {
            doAsync {
                val mgithubinfo =
                    fetchGitHubInfo(txtsearchuser.text.toString())
                val jsonreader = JSONObject(mgithubinfo)
                activityUiThread {
                    val id = jsonreader.getString("id")
                    val name = jsonreader.getString("name")
                    val url = jsonreader.getString("url")
                    val blog = jsonreader.getString("blog")
                    val bio = jsonreader.getString("bio")
                    tvInfo.text =
                        "${id}\n${name}\n${url}\n${blog}\n${bio}"
                }
            }
        }

        private fun fetchGitHubInfo(login_id: String): String {
            val url = "https://api.github.com/users/$login_id"
            val client = OkHttpClient()
            val request = Request.Builder().url(url).build()
            val response = client.newCall(request).execute()
            val bodystr = response.body().string() // this can be consumed
            only once
            return bodystr
        }

        override fun onResume() {
            super.onResume()
            txtsearchuser.setText("")
            txtsearchuser.setHint("Enter GitHub username")
        }
    }
}
```

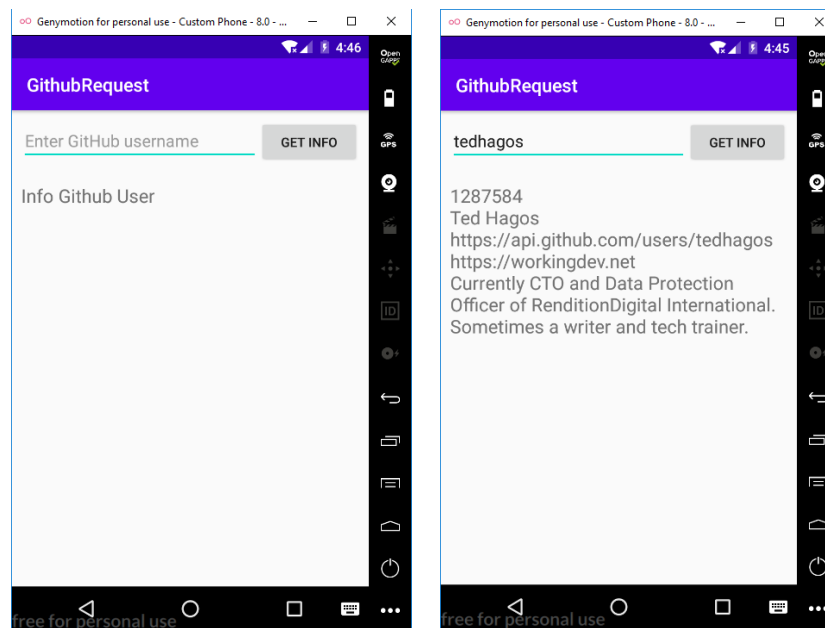
6. Tambahkan izin akses penggunaan internet pada file AndroidManifest.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="depandi.polbeng.ac.id.githubrequest">

    <uses-permission android:name="android.permission.INTERNET"/>

    ...
</manifest>
```

7. Jalankan program di Emulator Android. Cobalah untuk menganalisa program yang telah dibuat dan masukkan kedalam laporan praktikum.

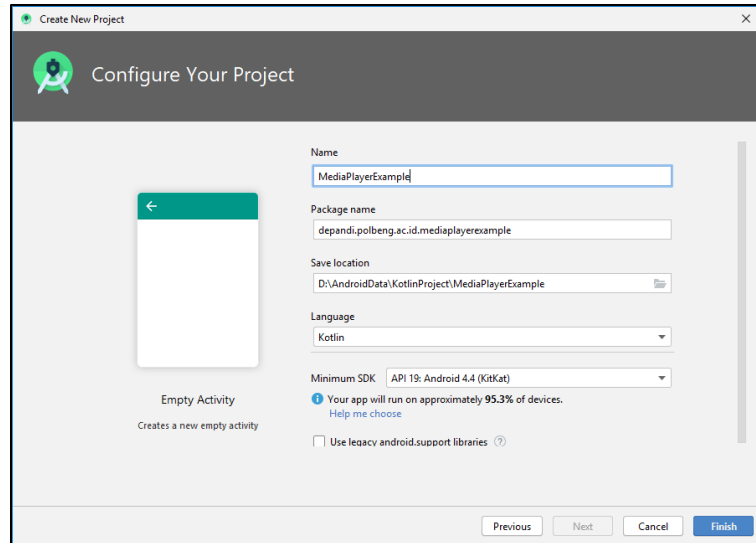


Tampilan Awal

Setelah Request Data

Media Player

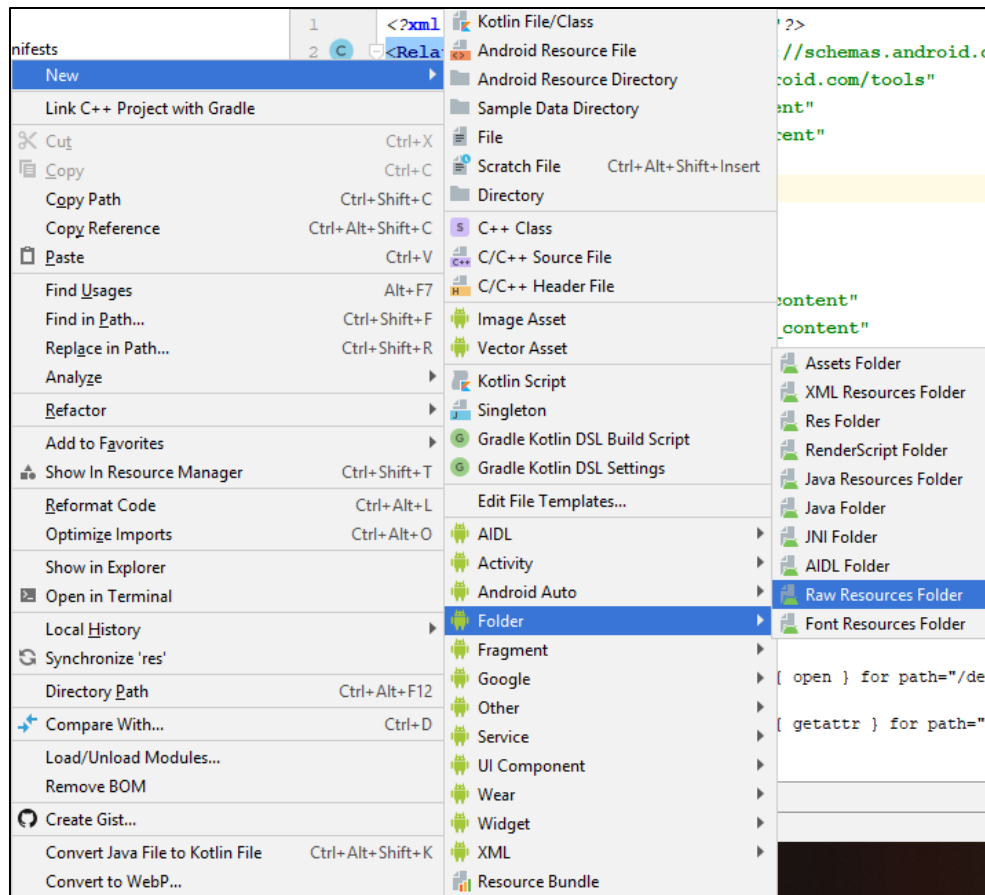
1. Buatlah *project* baru dengan konfigurasi seperti berikut :



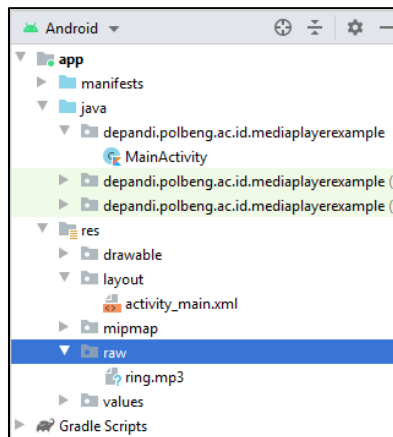
2. Setelah selesai membuat project baru, selanjutnya editlah file *strings.xml* pada folder *values*, tambahkan beberapa variabel dan nilai *string* sebagai berikut:

```
<resources>
    <string name="app_name">MediaPlayerExample</string>
    <string name="media_player">Media Player</string>
    <string name="start">Start</string>
    <string name="pause">Pause</string>
    <string name="stop">Stop</string>
</resources>
```

3. Buatlah folder raw untuk menampung file media suara dalam format mp3 dengan cara klik kanan pada folder res pilih New → Folder → Raw Resource Folder.



4. Masukkan file media suara ring.mp3 (silahkan dipilih file media suara sesuai keinginan anda) kedalam folder res→raw dengan cara *copy-paste* sehingga hasilnya adalah seperti berikut:



5. Editlah file *activity_main.xml* sehingga kode *layout* menjadi seperti berikut :

```
<?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"
    android:gravity="center_vertical">
```

```

tools:context=".MainActivity">

<TextView
    android:id="@+id/tvJudul"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/media_player"
    android:textSize="25sp"
    android:textStyle="bold"
    android:layout_marginTop="30dp"
    android:textAlignment="center"/>

<Button
    android:id="@+id/btnStart"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/start"
    android:onClick="startMedia"
    android:layout_marginTop="20dp"
    android:layout_gravity="center_horizontal"/>

<Button
    android:id="@+id/btnPause"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/pause"
    android:onClick="pauseMedia"
    android:layout_marginTop="10dp"
    android:layout_gravity="center_horizontal"/>

<Button
    android:id="@+id/btnStop"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/stop"
    android:onClick="stopMedia"
    android:layout_marginTop="10dp"
    android:layout_gravity="center_horizontal"/>

</LinearLayout>

```

6. Bukalah file sumber *MainActivity.kt* editlah kode program hingga menjadi seperti berikut :

```

package depandi.polbeng.ac.id.mediaplayerexample

import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import android.media.MediaPlayer;
import android.os.Bundle
import android.view.View

class MainActivity : AppCompatActivity() {

    private var mediaPlayer: MediaPlayer? = null
    private var pause = false
    private var position = 0

    override fun onCreate(savedInstanceState: Bundle?) {

```

```

        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

    fun startMedia(view: View?) {
        if (pause) { //initially, pause is set to false
            mediaPlayer?.seekTo(position)
            mediaPlayer?.start()
            pause = false
            //playing audio when in paused state
        } else {
            if (mediaPlayer == null) {
                mediaPlayer = MediaPlayer.create(applicationContext,
R.raw.ring)
                mediaPlayer?.setOnCompletionListener { stopPlayer() }
            }
            mediaPlayer?.start()
            //playing audio when in prepared state
        }
    }

    fun pauseMedia(view: View?) {
        if (mediaPlayer != null) {
            position = mediaPlayer!!.getCurrentPosition()
            mediaPlayer?.pause()
            pause = true
        }
    }

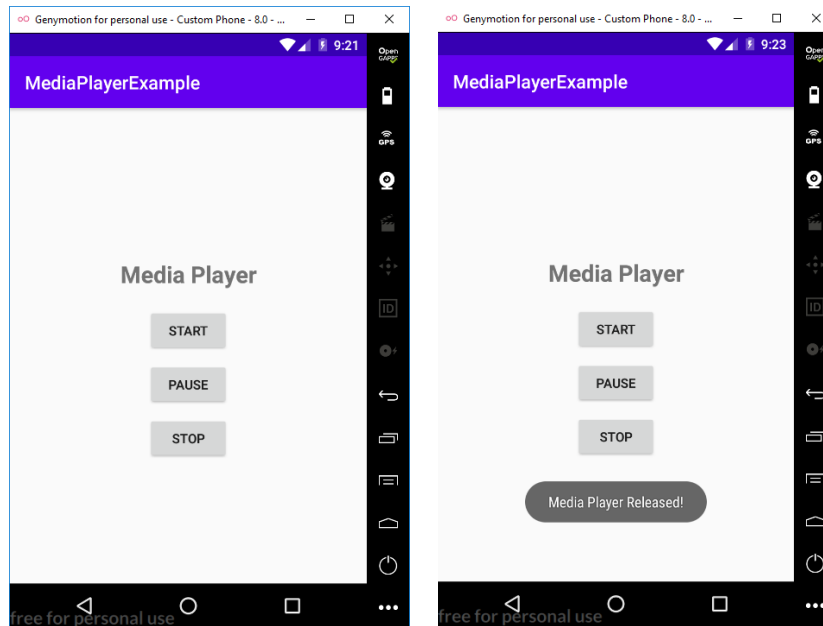
    fun stopMedia(view: View?) {
        stopPlayer()
    }

    private fun stopPlayer() {
        if (mediaPlayer != null) {
            pause = false
            position = 0
            mediaPlayer?.release()
            mediaPlayer = null;
            Toast.makeText(this, "Media Player Released!",
Toast.LENGTH_SHORT).show()
        }
    }

    override fun onStop() {
        super.onStop()
        stopPlayer()
    }
}

```

7. Jalankan program di Emulator Android. Cobalah untuk menganalisa program yang telah dibuat dan masukkan kedalam laporan praktikum.



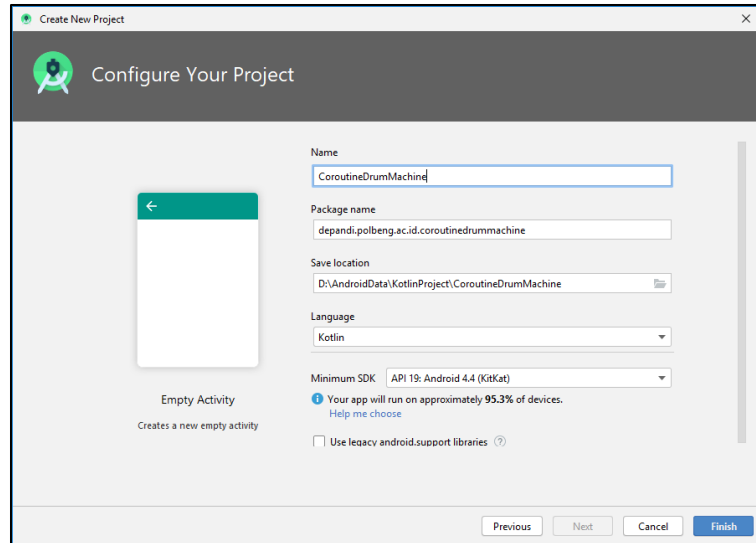
Coroutine

Coroutines = Co + Routines

Co bermaksud Cooperations atau kerjasama dan Routines bermaksud Function atau fungsi. Coroutine bisa disebut dengan kerjasama antar fungsi. Coroutine mengizinkan kita untuk menulis kode yang menjalankan tugas secara paralel di latar belakang (secara asinkron). Coroutine menjalankan setiap bagian kode secara berurutan, satu per satu di latar belakang. Contohnya adalah seperti membaca data dari server eksternal, tanpa menjeda atau menunggu suatu pekerjaan selesai sebelum melakukan pekerjaan lainnya.

Implementasi Coroutine pada aplikasi mesin Drum

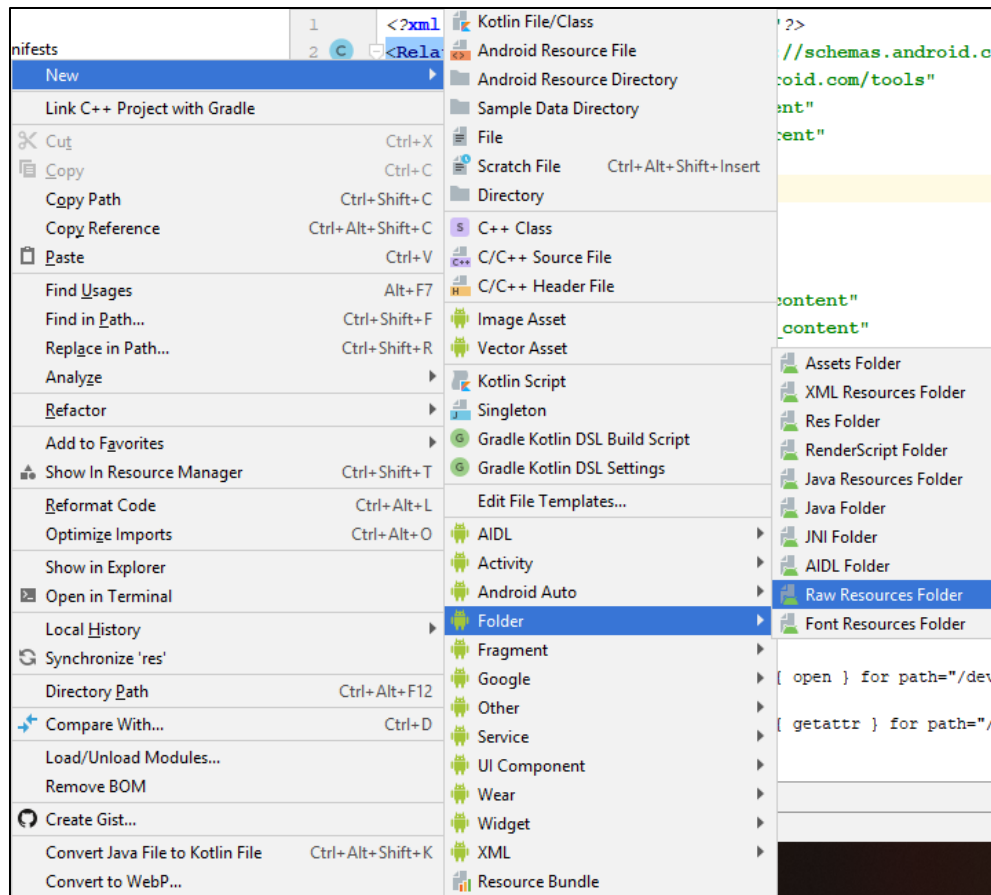
1. Buatlah *project* baru dengan konfigurasi seperti berikut :



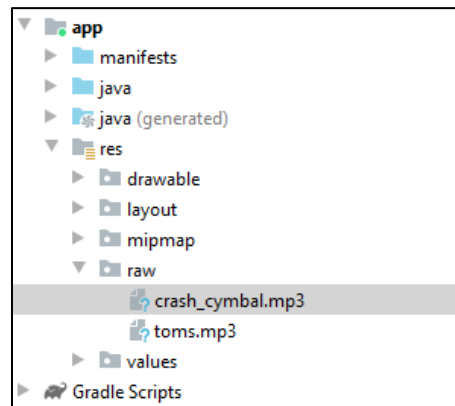
2. Setelah selesai membuat project baru, selanjutnya editlah file *strings.xml* pada folder *values*, tambahkan beberapa variabel dan nilai *string* sebagai berikut:

```
<resources>
    <string name="app_name">CoroutineDrumMachine</string>
    <string name="media_player">Media Player</string>
    <string name="start">Start</string>
</resources>
```

3. Buatlah folder raw untuk menampung file media suara dalam format mp3 dengan cara klik kanan pada folder res pilih New → Folder → Raw Resource Folder.



4. Masukkan file media suara crash_cymbal.mp3 dan toms.mp3 kedalam folder res→raw dengan cara copy-paste sehingga hasilnya adalah seperti berikut:



5. Editlah file *activity_main.xml* sehingga kode *layout* menjadi seperti berikut :

```
<?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"
    android:gravity="center_vertical"
    tools:context=".MainActivity">
```

</LinearLayout>

- dependencies {

}

- ```
package depandi.polbeng.ac.id.coroutinedrummy
```

```
package depandi.polbeng.ac.id.coroutinedrummy

import android.media.MediaPlayer
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import kotlinx.android.synthetic.main.activity_main.*
import kotlinx.coroutines.*

class MainActivity : AppCompatActivity() {

 private lateinit var cymbal: MediaPlayer
 private lateinit var tom: MediaPlayer

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

 cymbal = MediaPlayer.create(this, R.raw.crash_cymbal)
 tom = MediaPlayer.create(this, R.raw.toms)

 btnStart.setOnClickListener {
 runBlocking {
 launch { playBeats("x-x-x-x-x-x-x-x-x-x-x-x-", R.raw.toms) }

 playBeats("x-----x-----x-----x-----", R.raw.crash_cymbal)
 }
 }
 }
}
```



```

 }

 suspend fun playBeats(beats: String, fileId: Int){
 val parts = beats.split("x")
 var count = 0
 for(part in parts){
 count += part.length + 1
 if(part == ""){
 if(fileId == R.raw.crash_cymbal)
 cymbal.start()
 else
 tom.start()
 }else{
 delay(1000 * (part.length + 1L))
 if(count < beats.length){
 if(fileId == R.raw.crash_cymbal)
 cymbal.start()
 else
 tom.start()
 }
 }
 }
 }

 override fun onStop() {
 super.onStop()
 cymbal.stop()
 tom.stop()
 }
}

```

8. Jalankan program di Emulator Android. Cobalah untuk menganalisa program yang telah dibuat dan masukkan kedalam laporan praktikum.

### Tugas Praktikum

- Tuliskan pemahaman anda mengenai contoh koding program praktikum 1 dan screen shoot hasilnya.
- Buatlah laporan resmi praktikum 1.

### Daftar Referensi

<https://www.raywenderlich.com/84-kotlin-android-extensions>  
<https://kotlinlang.org/docs/tutorials/android-plugin.html>  
<https://www.petanikode.com/android-kotlin-extensions/>  
<https://antonioleiva.com/kotlin-android-extensions/>  
<https://www.codepolitan.com/belajar-android-kotlin-extension-59480152c8d5c>  
<https://medium.com/@budioktaviyans/kotlin-extensions-untuk-android-549497c042bb>  
<https://android.jlelse.eu/kotlin-extensions-for-android-part-1-32aeb276b6b2>

<https://proandroiddev.com/kotlin-android-extensions-using-view-binding-the-right-way-707cd0c9e648>

<https://www.raywenderlich.com/84-kotlin-android-extensions>

<https://www.raywenderlich.com/>

<https://blog.mindorks.com/mastering-kotlin-coroutines-in-android-step-by-step-guide>

<https://www.raywenderlich.com/2117501-kotlin-coroutines-tutorial-for-android-advanced>

\*\* 😊 Selamat Mengerjakan 😊 \*\*\*