

## Tujuan Praktikum

1. Mahasiswa mampu memahami dan menerapkan penggunaan View dan ViewGroup pada Aplikasi Android.
2. Mahasiswa mampu memahami dan menerapkan penggunaan Style dan Theme pada Aplikasi Android.

## Tools

1. Android Studio

Panduan ini menggunakan versi Android Studio: Giraffe - 2022.3.1

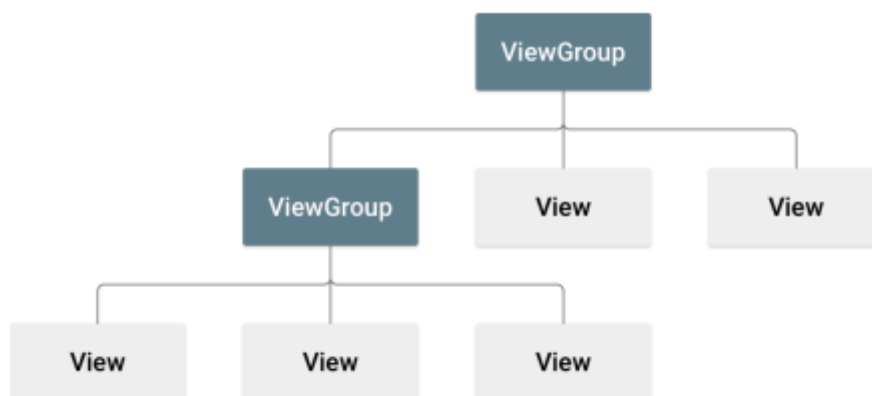
Android API: Level 34

2. Figma

Resource file: assets.zip

## View dan ViewGroup

Sebuah layout bertugas untuk mendefinisikan sebuah user interface pada aplikasi, seperti pada sebuah activity. Seluruh elemen pada layout dibangun dengan menggunakan hierarki dari objek View dan ViewGroup. **View** bertugas untuk menggambarkan sesuatu yang dapat dilihat dan berinteraksi dengan pengguna. Sementara **ViewGroup** merupakan sebuah wadah/kontainer yang digunakan untuk menyusun objek View dan ViewGroup lainnya, seperti pada gambar berikut.



View memiliki banyak subclass seperti:

- **TextView**, berfungsi untuk menampilkan teks kepada pengguna.
- **Button**, berfungsi untuk menampilkan sebuah tombol yang dapat ditekan oleh pengguna.
- **ImageView**, berfungsi untuk menampilkan gambar kepada pengguna.
- **RecyclerView**, berfungsi untuk menampilkan informasi dalam bentuk list.
- **GridView**, berfungsi untuk menampilkan informasi dalam bentuk grid.
- **RadioButton**, berfungsi untuk memungkinkan pengguna untuk memilih satu pilihan dari berbagai pilihan yang ada.
- **Checkbox**, berfungsi untuk memungkinkan pengguna untuk memilih banyak pilihan yang tersedia.

Berikut adalah beberapa contoh ViewGroup yang tersedia pada Android:

- **ConstraintLayout**, memungkinkan untuk membangun sebuah interface yang kompleks dengan flat hierarki (tidak ada ViewGroup bersarang).
- **LinearLayout**, memungkinkan untuk menyusun komponen View dan ViewGroup lain dengan orientasi tertentu, seperti vertikal atau horizontal.
- **RelativeLayout**, memungkinkan untuk menyusun komponen dengan memanfaatkan posisi relatif terhadap view lain.

## Style & Theme

Penggunaan Style dan Theme pada Android dapat memungkinkan kita untuk memisahkan detail implementasi desain dari deklarasi layout pada file xml, seperti penggunaan CSS pada pengembangan web. Deklarasi Style dan Theme tersimpan pada folder `res/values/` di file `styles.xml`.

**Style** merupakan sebuah koleksi atribut yang dapat digunakan untuk menentukan tampilan pada sebuah View. Sebuah Style dapat memiliki beberapa atribut seperti font color, font size, background color, dan masih banyak lagi.

**Theme** merupakan sebuah koleksi atribut yang dapat diterapkan ke seluruh aplikasi, activity, atau hierarki view tertentu—tidak hanya untuk sebuah View saja.

## Contoh Penerapan Style

Sebagai contoh, pada kode berikut terdapat beberapa deklarasi TextView dengan fungsi untuk menampilkan informasi berupa teks kepada pengguna. Beberapa TextView tersebut memiliki spesifikasi tampilan yang sama.

```
<?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:padding="24dp"
    tools:context=".DetailActivity">

    ...

    <TextView
        android:id="@+id/tv_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:textSize="18sp"
        tools:text="Nama: " />
```

```
<TextView
    android:id="@+id/tv_nim"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:textSize="18sp"
    tools:text="NIM: " />

<TextView
    android:id="@+id/tv_phone"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:textSize="18sp"
    tools:text="No. HP: " />

<TextView
    android:id="@+id/tv_email"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:textSize="18sp"
    tools:text="Email: " />

...
</LinearLayout>
```

Sangat tidak efektif jika kita harus melakukan copy-paste ketika mendeklarasikan sebuah TextView baru. Jika spesifikasi yang diberikan oleh UI Designer berubah, tentu saja kita harus mengubah value atributnya satu persatu. Untuk mengatasi permasalahan tersebut, kita dapat membuat style pada file /res/layout/styles.xml seperti berikut:

```
<resources>
    ...
    <style name="TextHeadline1">
        <item name="android:layout_width">match_parent</item>
        <item name="android:layout_height">wrap_content</item>
        <item name="android:layout_marginTop">8dp</item>
        <item name="android:textSize">18sp</item>
    </style>
</resources>
```

Kemudian, kita dapat menggunakannya pada tiap-tiap TextView yang membutuhkannya.

```
<?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:padding="24dp"
        tools:context=".DetailActivity">

        ...

        <TextView
            android:id="@+id/tv_name"
            style="@style/TextHeadline1"
            tools:text="Nama: " />

        <TextView
            android:id="@+id/tv_nim"
            style="@style/TextHeadline1"
            tools:text="NIM: " />

        <TextView
            android:id="@+id/tv_phone"
            style="@style/TextHeadline1"
            tools:text="No. HP: " />

        <TextView
            android:id="@+id/tv_email"
            style="@style/TextHeadline1"
            tools:text="Email: " />

        ...

    </LinearLayout>
```

# Latihan

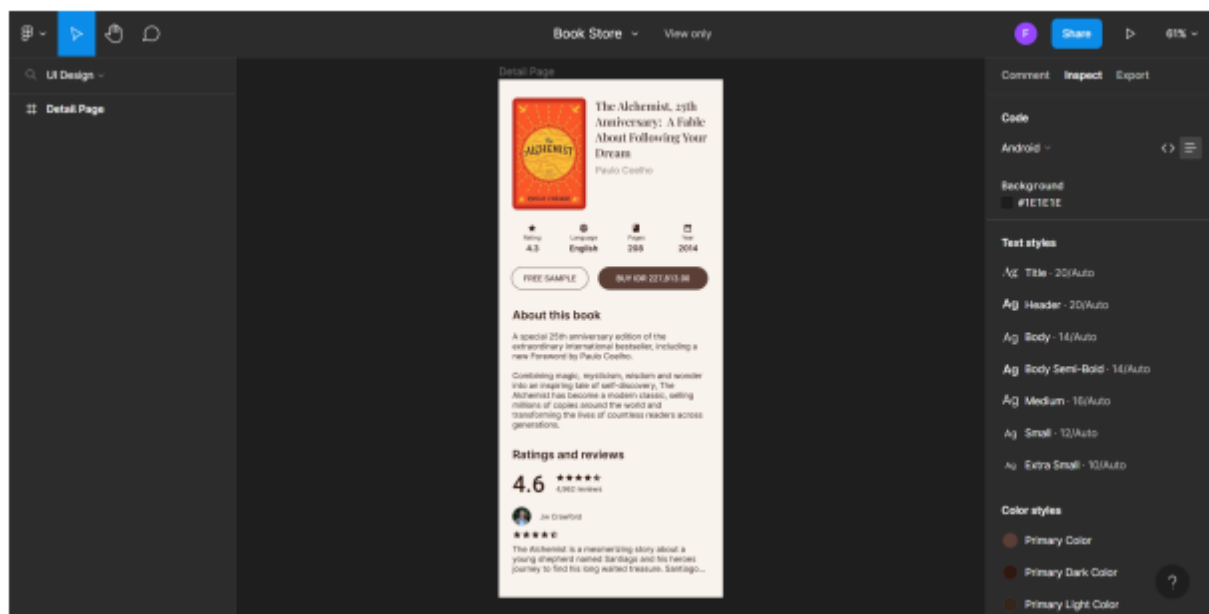
## 1. Persiapan Project

Buatlah sebuah project baru dengan konfigurasi berikut:

Nama Project	Book Store
Bahasa Pemrograman	Kotlin
Minimum SDK	API 24 (“Nougat”; Android 7.0)
Nama Package	com.android.bookstore
Tipe Activity	Empty Views Activity

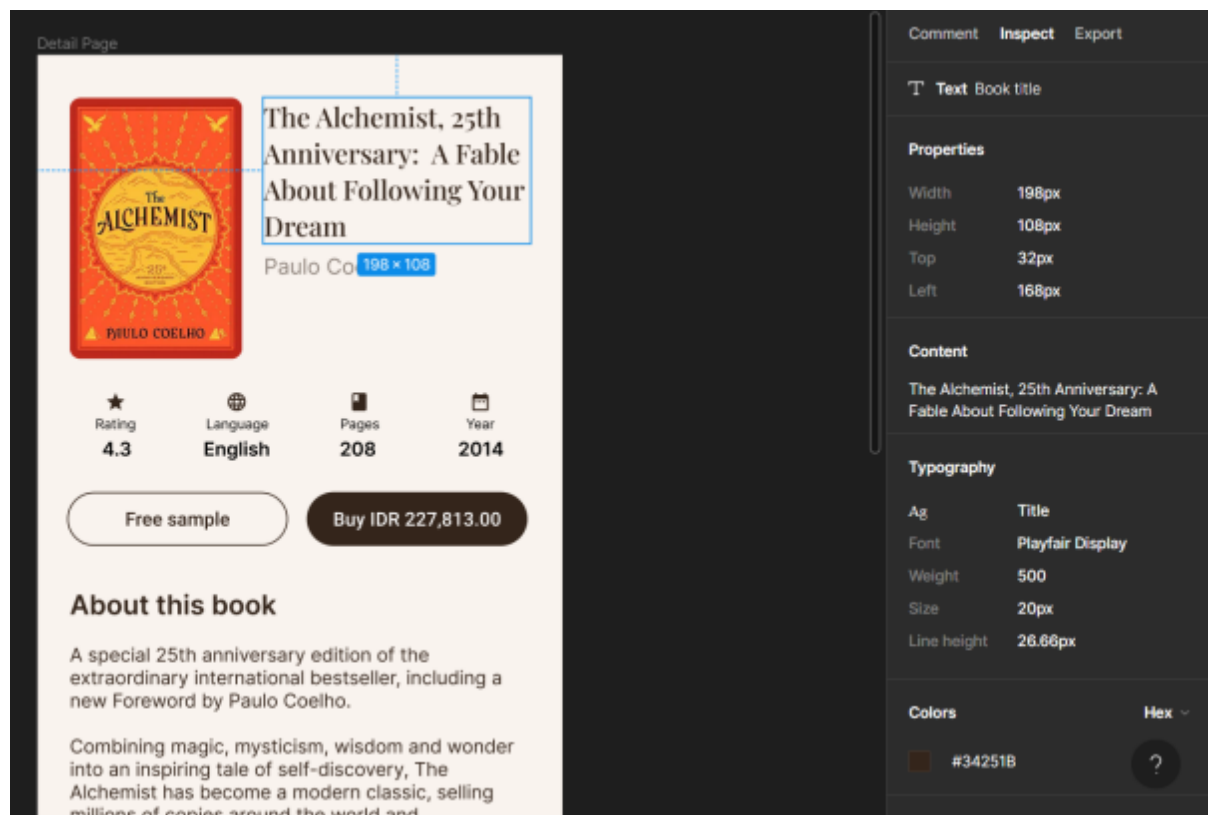
## 2. Pengenalan Project Figma

Pada latihan kali ini, kita akan berfokus pada implementasi user interface dengan menerapkan sebuah desain yang sudah ditentukan. Silakan buka project Figma pada link [berikut](#). Silakan login dengan akun Figma masing-masing untuk melihat detail dari desain yang diberikan.



Perhatikan pada sidebar sebelah kanan, terdapat informasi seperti text styles dan color styles. Kita akan menerapkan properti dari masing-masing text styles tersebut untuk kemudian diterapkan pada masing-masing view.

Cobalah untuk menekan salah satu komponen yang ada pada desain. Perhatikan bahwa detail properti yang digunakan pada suatu komponen akan ditampilkan pada sidebar sebelah kanan.



### 3. Implementasi Desain - Color styles

Pindahkan spesifikasi warna pada Figma ke dalam file colors.xml pada folder res/values. Kita dapat memindahkannya secara manual, atau menggunakan tools pada website Material Design dengan mengunjungi link [berikut](#). Klik Export pada pojok kanan atas, kemudian pilih Android. Secara otomatis akan terdapat sebuah file bernama colors.xml yang akan didownload. Pindahkan kode yang ada pada file tersebut ke dalam project Android Studio.



Kemudian tambahkan sebuah warna dengan nama backgroundColor seperti kode berikut:

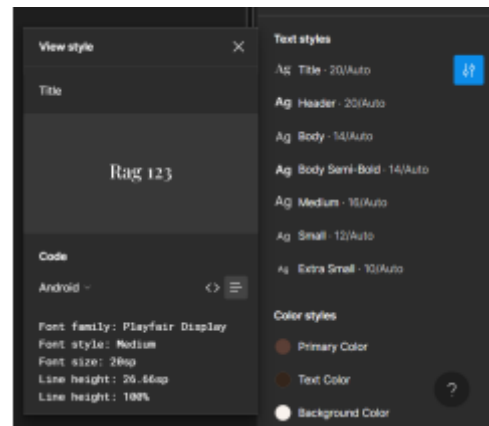
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="primaryColor">#5b3f36</color>
    <color name="primaryLightColor">#896a60</color>
    <color name="primaryDarkColor">#301810</color>
```



```
<color name="secondaryColor">#ffa726</color>
<color name="secondaryLightColor">#ffd95b</color>
<color name="secondaryDarkColor">#c77800</color>
<color name="backgroundColor">#f9f3ee</color>
<color name="primaryTextColor">#ffffff</color>
<color name="secondaryTextColor">#000000</color>
</resources>
```

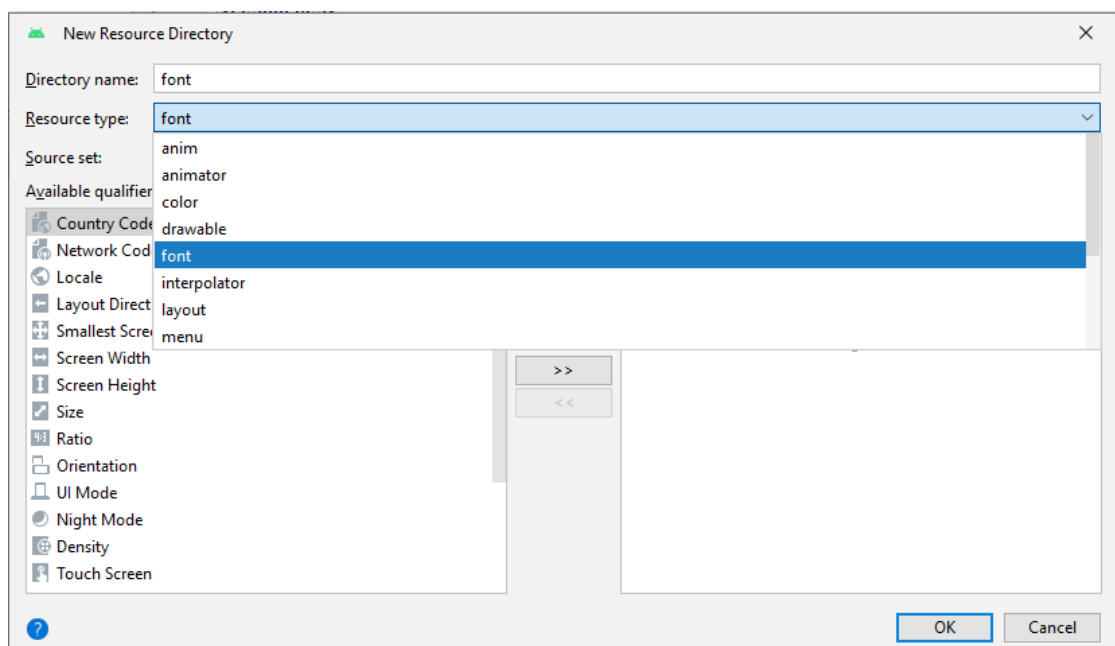
## 4. Implementasi Desain - Theme & Text Style

Pada tahap ini, kita akan menyalin spesifikasi text style ke dalam tema aplikasi Android yang akan kita kerjakan. Untuk melihat spesifikasi dari suatu text style, kita dapat menekan icon setting pada salah satu text style yang ada. Kita akan dapat melihat beberapa spesifikasi seperti font family, font style, font size, dan lain-lain.

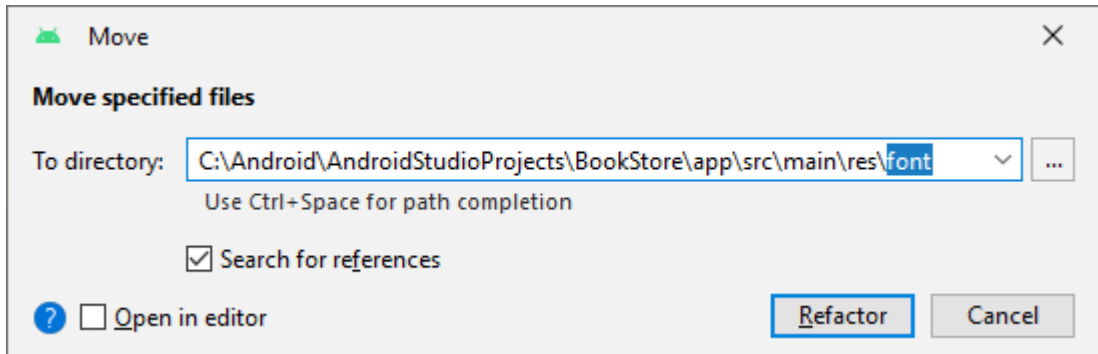


Sebelum melakukan implementasi text styles, kita akan import terlebih dahulu font yang telah disediakan ke dalam project Android Studio. Ikutilah langkah-langkah berikut:

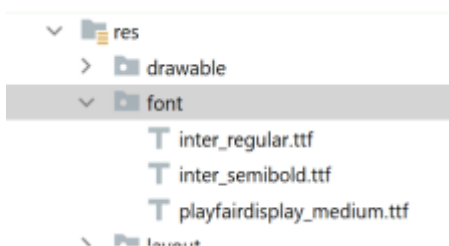
1. Klik kanan pada folder res, lalu pilih menu New > Android Resource Directory
2. Akan tampil sebuah jendela New Resource Directory. Pada jendela tersebut, ganti Resource Type menjadi font, lalu klik OK.



3. Drag & drop file font yang telah disediakan ke folder res/font/. Jika muncul sebuah pop-up, klik Refactor.



4. Pastikan seluruh font telah berhasil ditambahkan ke dalam folder res/font.



Selanjutnya, kita akan buat sebuah style baru pada file themes.xml di folder res/values/. Tambahkan dan sesuaikan kode seperti berikut:

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.BookStore" parent="Theme.MaterialComponents.Light.DarkActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/primaryColor</item>
        <item name="colorPrimaryVariant">@color/primaryDarkColor</item>
        <item name="colorOnPrimary">@color/primaryTextColor</item>
        <!-- Secondary brand color. -->
        <item name="colorSecondary">@color/secondaryColor</item>
        <item name="colorSecondaryVariant">@color/secondaryDarkColor</item>
        <item name="colorOnSecondary">@color/secondaryTextColor</item>
        <!-- Status bar color. -->
        <item name="android:statusBarColor">?attr/colorPrimaryVariant</item>
        <!-- Customize your theme here. -->
        <item name="android:windowBackground">@color/backgroundColor</item>
    </style>
    <style name="Text">
        <item name="fontFamily">@font/inter_regular</item>
        <item name="android:textColor">@color/primaryDarkColor</item>
    </style>
    <style name="Text.Title">
        <item name="fontFamily">@font/playfairdisplay_medium</item>
        <item name="android:textSize">20sp</item>
    </style>
    <style name="Text.Header">
```

```
<item name="fontFamily">@font/inter_semibold</item>
<item name="android:textSize">20sp</item>
</style>
<style name="Text.Body">
    <item name="android:textSize">14sp</item>
</style>
<style name="Text.Body.SemiBold">
    <item name="fontFamily">@font/inter_semibold</item>
</style>
<style name="Text.Medium">
    <item name="android:textSize">16sp</item>
</style>
<style name="Text.Small">
    <item name="android:textSize">12sp</item>
</style>
<style name="Text.ExtraSmall">
    <item name="android:textSize">10sp</item>
</style>
</resources>
```

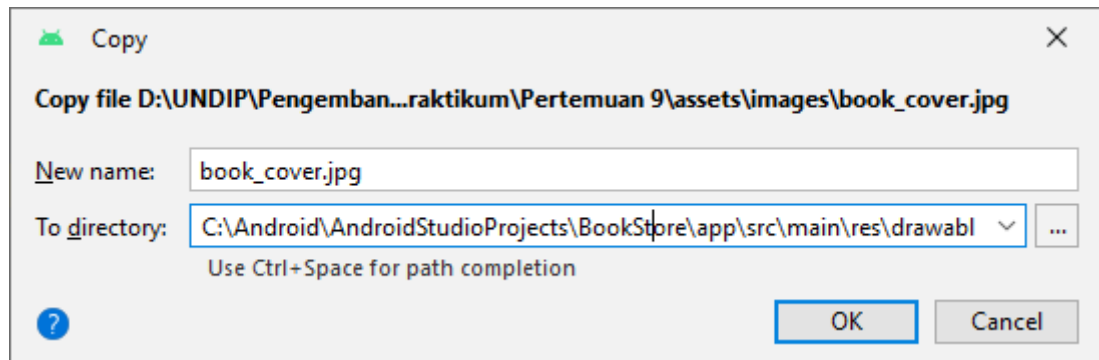
\* Pada praktikum ini, kita hanya akan menggunakan satu file themes.xml saja. Jika terdapat file themes.xml (Night) silakan hapus/abaikan file tersebut. Hal ini bertujuan agar praktikum lebih fokus pada implementasi Theme & Style.

## 5. Implementasi Desain - Book Main Information Group

Pada tahap ini, kita akan berfokus untuk melakukan implementasi terhadap sebuah grup yaitu Book main information. Pada grup ini, terdapat tiga buah view yang akan digunakan, yaitu satu buah ImageView dan dua buah TextView.



Pertama, import asset gambar cover buku yang akan digunakan. Silakan lakukan drag & drop file `book_cover.jpg` ke dalam folder `res/drawable`. Pastikan file tersebut sudah berhasil ditambahkan. Jika muncul sebuah pop-up, silakan tekan tombol Refactor. Lakukan hal yang sama untuk file `user_profile.jpg`.



Selanjutnya kita akan lakukan implementasi pada file `activity_main.xml`. Kita akan belajar untuk menggunakan Constraint Layout sebagai ViewGroup utama. Silakan implementasikan desain dan sesuaikan dengan spesifikasi yang telah ditentukan pada project Figma. Hasil kode akan seperti berikut.

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    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">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingHorizontal="24dp"
        android:paddingVertical="32dp">

        <ImageView
            android:id="@+id/imageView"
            android:layout_width="128dp"
            android:layout_height="194dp"
            android:src="@drawable/book_cover"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <TextView
            android:id="@+id/textView"
```

```
style="@style/Text.Title"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginStart="16dp"
android:text="The Alchemist, 25th Anniversary: A Fable About
Following Your Dream"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@+id/imageView"
app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/textView2"
    style="@style/Text.Medium"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:alpha="0.5"
    android:text="Paulo Coelho"
    app:layout_constraintStart_toStartOf="@+id/textView"
    app:layout_constraintTop_toBottomOf="@+id/textView" />

</androidx.constraintlayout.widget.ConstraintLayout>

</ScrollView>
```

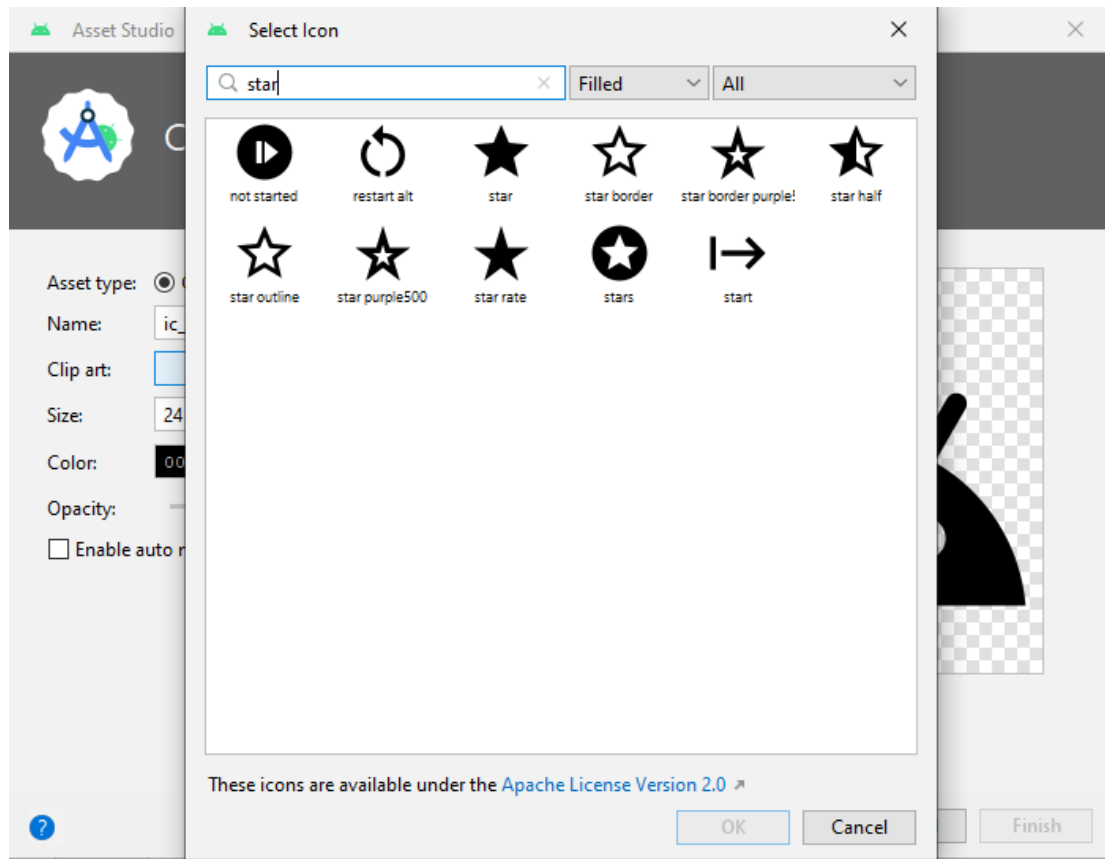
## 6. Implementasi Desain - Status Group

Pada tahap ini, kita akan melanjutkan implementasi desain pada halaman detail. Kita akan melakukan implementasi terhadap grup selanjutnya yaitu status.

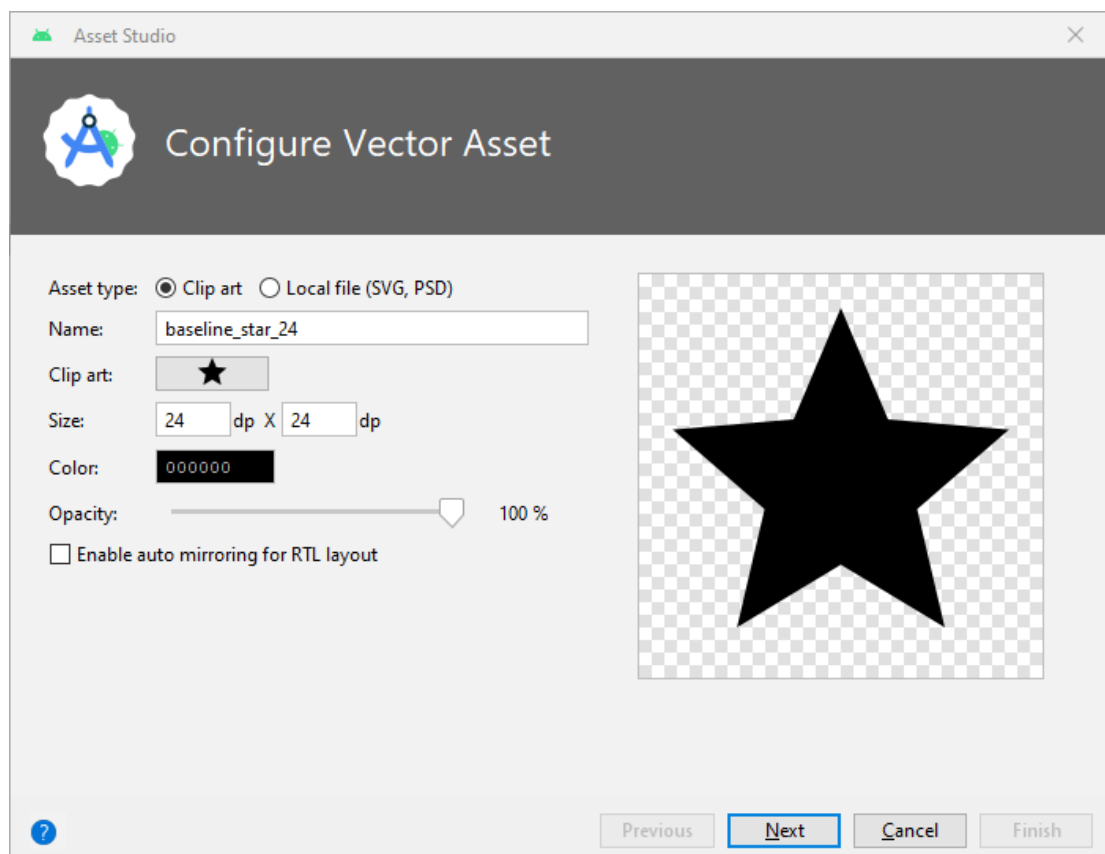


Pertama, kita akan menambahkan icon-icon yang kita butuhkan. Silakan menuju menu File > New > Vector Asset. Ganti Clip Art dengan gambar yang bersesuaian. Sebagai contoh, kita akan menambahkan icon bintang.

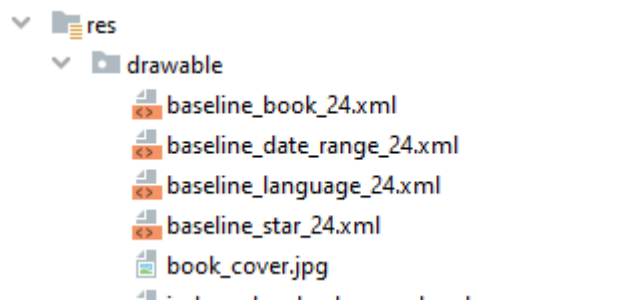
\* Manfaatkan fitur pencarian pada jendela Select Icon untuk memudahkan pencarian icon yang diinginkan



Selanjutnya klik Next dan Finish untuk menambahkan icon yang telah dipilih.



Lakukan hal yang sama untuk icon-icon lainnya, sehingga folder res/drawable akan berisi file-file berikut:



Selanjutnya, implementasikan desain dan sesuaikan dengan spesifikasi yang telah ditentukan pada project Figma. Hasil kode akan seperti berikut.

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    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">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingHorizontal="24dp"
        android:paddingVertical="32dp">

        ...

        <LinearLayout
            android:id="@+id/linearLayout"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="24dp"
            android:weightSum="4"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/imageView">

            <LinearLayout
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:gravity="center_horizontal"
                android:orientation="vertical">

                <ImageView
                    android:layout_width="16dp"
                    android:layout_height="16dp"
```

```
        android:src="@drawable/baseline_star_24" />

<TextView
    style="@style/Text.ExtraSmall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Rating" />

<TextView
    style="@style/Text.Body.SemiBold"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="4.3" />

</LinearLayout>

<LinearLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:gravity="center_horizontal"
    android:orientation="vertical">

    <ImageView
        android:layout_width="16dp"
        android:layout_height="16dp"
        android:src="@drawable/baseline_language_24" />

    <TextView
        style="@style/Text.ExtraSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Language" />

    <TextView
        style="@style/Text.Body.SemiBold"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="English" />

</LinearLayout>

<LinearLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:gravity="center_horizontal"
    android:orientation="vertical">

    <ImageView
        android:layout_width="16dp"
        android:layout_height="16dp"
        android:src="@drawable/baseline_book_24" />

    <TextView
        style="@style/Text.ExtraSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```



```
        android:text="Pages" />

        <TextView
            style="@style/Text.Body.SemiBold"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="208" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="center_horizontal"
        android:orientation="vertical">

        <ImageView
            android:layout_width="16dp"
            android:layout_height="16dp"
            android:src="@drawable/baseline_date_range_24" />

        <TextView
            style="@style/Text.ExtraSmall"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Year" />

        <TextView
            style="@style/Text.Body.SemiBold"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="2014" />

    </LinearLayout>

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

</ScrollView>
```

Selamat, sampai tahap ini, kita sudah berhasil mengimplementasikan dua grup desain. Tetap semangat ya..

## 8. Implementasi Desain - About This Book Group

Implementasi desain pada grup ini cukup sederhana, kita hanya perlu menambahkan dua buah TextView untuk menampilkan informasi buku. Silakan implementasikan desain dan sesuaikan dengan spesifikasi yang telah ditentukan pada project Figma. Hasil kode akan seperti berikut.

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    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">

    ...

    <LinearLayout
        android:id="@+id/linearLayout2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:weightSum="2"
        app:layout_constraintEnd_toEndOf="@+id/linearLayout"
        app:layout_constraintStart_toStartOf="@+id/linearLayout"
        app:layout_constraintTop_toBottomOf="@+id/linearLayout">

        <com.google.android.material.button.MaterialButton
            style="@style/Widget.MaterialComponents.Button.OutlinedButton"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="0.8"
            android:text="Free sample"
            app:cornerRadius="24dp" />

        <Space
            android:layout_width="0dp"
            android:layout_height="0dp"
            android:layout_weight="0.1" />

        <com.google.android.material.button.MaterialButton
            android:id="@+id/materialButton2"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1.1"
            android:text="Buy IDR 227,813.00"
            app:cornerRadius="24dp" />

    </LinearLayout>

    <TextView
        android:id="@+id/textView3"
```

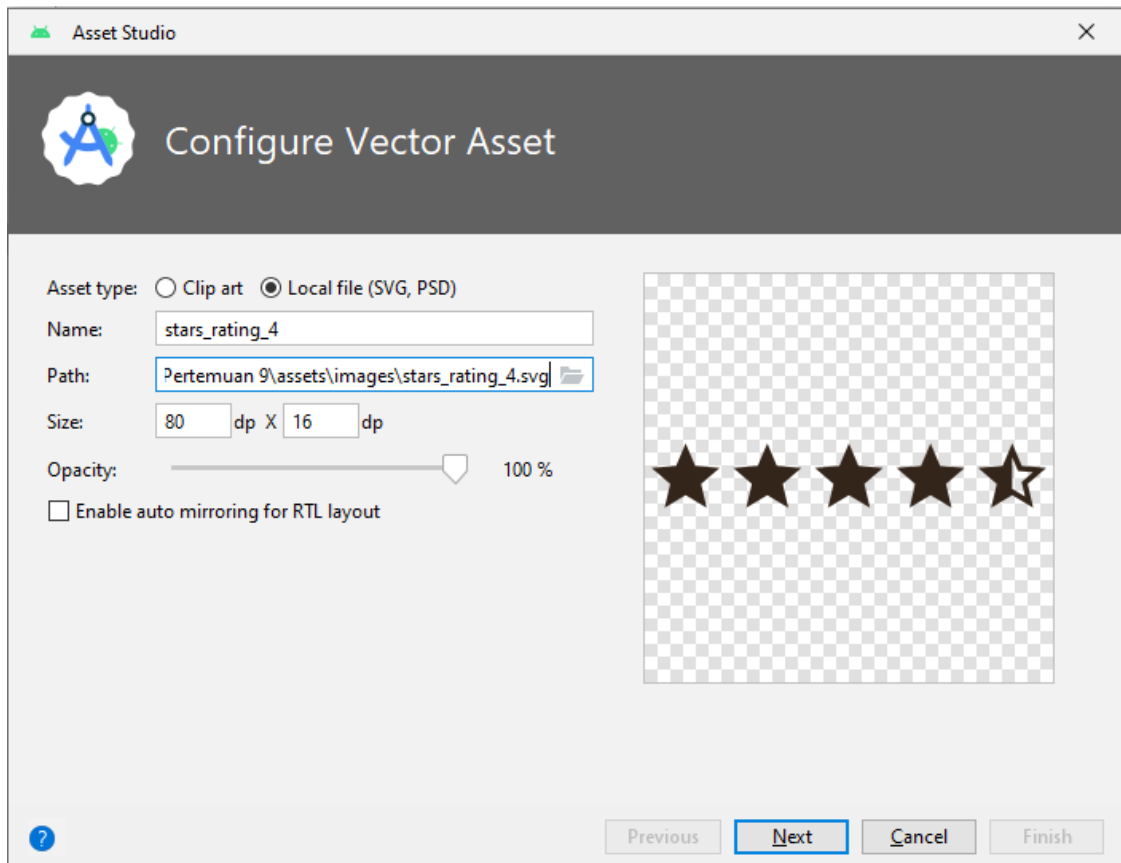
```
        style="@style/Text.Header"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:text="About this book"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/linearLayout2"/>

<TextView
    android:id="@+id/textView4"
    style="@style/Text.Body"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="A special 25th anniversary edition of the extraordinary
international bestseller, including a new Foreword by Paulo Coelho.
\n\nCombining magic, mysticism, wisdom and wonder into an inspiring
tale of self-discovery, The Alchemist has become a modern classic,
selling millions of copies around the world and transforming the lives
of countless readers across generations."
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView3" />

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

## 9. Implementasi Desain - Ratings And Reviews Group

Sebelum implementasi desain, kita akan melakukan beberapa persiapan terlebih dahulu. Silakan import gambar yang dibutuhkan ke Android Studio. Untuk gambar yang memiliki format .svg kita dapat mengimpornya melalui menu File > New > Vector Asset.

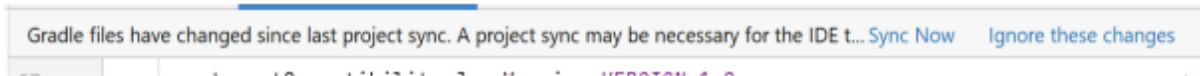


Ubah Asset Type menjadi Local file (SVG, PSD). Pada bagian Path, pilih lokasi file .svg yang ingin diimport ke dalam project.

Selanjutnya, karena kita akan menampilkan sebuah gambar berbentuk lingkaran, kita akan memanfaatkan library CircleImageView untuk melakukan tugas tersebut. Untuk informasi lebih jelas tentang library tersebut, silakan kunjungi halaman [berikut](#). Silakan tambahkan kode berikut pada file build.gradle.kts (Module :app) pada Gradle Scripts.

```
dependencies {  
    ...  
    implementation("de.hdodenhof:circleimageview:3.1.0")  
}
```

Setelah itu, klik Sync Now agar library dapat diunduh dan dimasukkan ke dalam project aplikasi Android.



Silakan implementasikan desain dan sesuaikan dengan spesifikasi yang telah ditentukan pada project Figma. Hasil kode akan seperti berikut.

```
<ScrollView
    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">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingHorizontal="24dp"
        android:paddingVertical="32dp">

        ...

        <TextView
            android:id="@+id/textView5"
            style="@style/Text.Header"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="32dp"
            android:text="Ratings and reviews"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/textView4" />

        <TextView
            android:id="@+id/textView6"
            style="@style/Text"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:text="4.6"
            android:textSize="40sp"
            android:textStyle="bold"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/textView5" />

        <TextView
            style="@style/Text.Small"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="16dp"
```

```
        android:text="4,962 reviews"
        app:drawableTopCompat="@drawable/stars_rating_4"
        app:layout_constraintBottom_toBottomOf="@+id/textView6"
        app:layout_constraintStart_toEndOf="@+id/textView6"
        app:layout_constraintTop_toTopOf="@+id/textView6" />

<de.hdodenhof.circleimageview.CircleImageView
    android:id="@+id/circleImageView"
    android:layout_width="32dp"
    android:layout_height="32dp"
    android:layout_marginTop="16dp"
    android:src="@drawable/user_profile"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView6" />

<TextView
    android:id="@+id/textView7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:text="Jw Crawford"
    app:layout_constraintBottom_toBottomOf="@+id/circleImageView"
    app:layout_constraintStart_toEndOf="@+id/circleImageView"
    app:layout_constraintTop_toTopOf="@+id/circleImageView" />

<ImageView
    android:id="@+id/imageView2"
    android:layout_width="wrap_content"
    android:layout_height="16dp"
    android:layout_marginTop="8dp"
    android:src="@drawable/stars_rating_4"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/circleImageView" />

<TextView
    style="@style/Text.Body"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:maxLines="3"
    android:ellipsize="end"
    android:text="The Alchemist is a mesmerizing story about a
    young shepherd named Santiago and his heroes journey to find
    his long waited treasure. Santiago keeps on having a recurring
    dream about a hidden treasure somewhere in Egypt."
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView2" />

</androidx.constraintlayout.widget.ConstraintLayout>

</ScrollView>
```

Setelah seluruh tahapan implementasi desain dilakukan, maka hasil akhir yang didapat adalah sebagai berikut



## Tugas

1. Buatlah sebuah project aplikasi baru, lalu implementasikan sebuah desain ke dalam `activity_main.xml` dengan ketentuan:
  - a. Terdapat implementasi dari `Constraint Layout` dan `Linear Layout` sebagai `ViewGroup`.
  - b. Terdapat implementasi `ImageView`, `TextView`, `Button`, dan `CircleImageView`.
  - c. Menggunakan style untuk elemen view tertentu.
  - d. Masing-masing peserta praktikum menerapkan desain antarmuka yang berbeda.
2. Export project dengan menggunakan menu `File > Export > Export to ZIP file`, kemudian kumpulkan pada link pengumpulan yang sudah ditentukan.

-- Selamat Belajar --



## Additional Resources

<https://developer.android.com/develop/ui>

<https://developer.android.com/develop/ui/views/layout/declaring-layout>

<https://developer.android.com/reference/android/view/View>

<https://developer.android.com/develop/ui/views/layout/constraint-layout>

<https://developer.android.com/develop/ui/views/layout/linear>

<https://developer.android.com/develop/ui/views/layout/relative>