



# **Style & App Themes**

Pertemuan ke-3



# Android Styling

Gaya dan tema di Android memungkinkan Kita memisahkan detail desain aplikasi dari struktur dan perilaku UI, mirip dengan stylesheet di desain web. Gaya adalah kumpulan atribut yang menentukan tampilan untuk View tunggal. Gaya dapat menentukan atribut seperti warna font, ukuran font, warna latar belakang, dan banyak lagi.


Tema adalah kumpulan atribut yang diterapkan ke seluruh aplikasi, aktivitas, atau hierarki tampilan, bukan hanya tampilan individual. Saat Kita menerapkan tema, setiap tampilan di aplikasi atau aktivitas akan menerapkan setiap atribut tema yang didukungnya. Tema juga dapat menerapkan gaya ke elemen nontampilan, seperti status bar dan latar belakang jendela.

Ada 3 cara untuk mengatur gaya dan tema aplikasi, yaitu :

- Theme
- Style
- Atribut View

# Themes

Gimana sih caranya? Kita bisa mendefinisikan sebuah theme di aplikasi dengan menggunakan file **res/values/themes.xml**. Jangan lupa gunakan tag `<styles>`.



```
<style name="Theme.MyApp" parent="Theme.MaterialComponents.Light">
    <item name="colorPrimary">@color/orange_500</item>
    <item name="colorPrimaryVariant">@color/orange_700</item>
    <item name="colorSecondary">@color/pink_200</item>
    <item name="colorSecondaryVariant">@color/pink_700</item>
    ...
</style>
```

# Themes

## Menggunakan Sebuah Theme

Selain itu, kita bisa menentukan theme pada komponen yang memiliki sebuah **Context** seperti **Activity** atau **Views/ViewGroups**.

Ketika kita membuat project dari sebuah template di Android Studio, biasanya sudah ada custom theme yang didefinisikan untuk aplikasi pada file **themes.xml**.

Kita akan melihat theme tersebut dipakai pada tag **<application>** atau **<activity>** pada file **AndroidManifest.xml**.

Alternatifnya, kamu bisa menerapkan theme ke **View/ViewGroup** seperti di **ConstraintLayout** di samping.



Di AndroidManifest.xml

```
<manifest ... >
  <application ... >
    <activity android:theme="@style/Theme.MyApp" ... >
    </activity>
  </application>
</manifest>
```



Di file layout

```
<ConstraintLayout ...
  android:theme="@style/Theme.MyApp">
```

# Themes

## Menggunakan Atribut Theme

Kita juga bisa menggunakan atribut theme untuk membuat layout lebih fleksibel dengan menggunakan sintaks **?attr/<theme attribute>**.


Pada contoh di samping, LinearLayout menggunakan atribut colorSurface daripada menentukan latarnya dengan warna tertentu.

Dengan demikian, LinearLayout bisa memiliki warna latar yang berbeda tergantung theme yang digunakan.



# Styles

Untuk membuat style, buka file styles.xml pada project **res/values/styles.xml**.



```
<style name="DescriptionStyle">
    <item name="android:textColor">#00FF00</item>
    <item name="android:textSize">16sp</item>
    ...
</style>
```



# Styles

## Menggunakan styles

Di dalam file layout, terapkan style pada sebuah View dengan menentukan atribut stylenya seperti contoh di samping.

Seperti yang telah diberitahukan sebelumnya, kita juga bisa menggunakan atribut theme di dalam sebuah style.

Contohnya, textColor menggunakan atribut theme **?attr/colorOnSurface**.



```
<TextView
    style="@style/DescriptionStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/description_text" />
```



```
<style name="DescriptionStyle">
    <item name="android:textColor">?attr/colorOnSurface</item>
    <item name="android:textSize">16sp</item>
    ...
</style>
```

# Attribute View

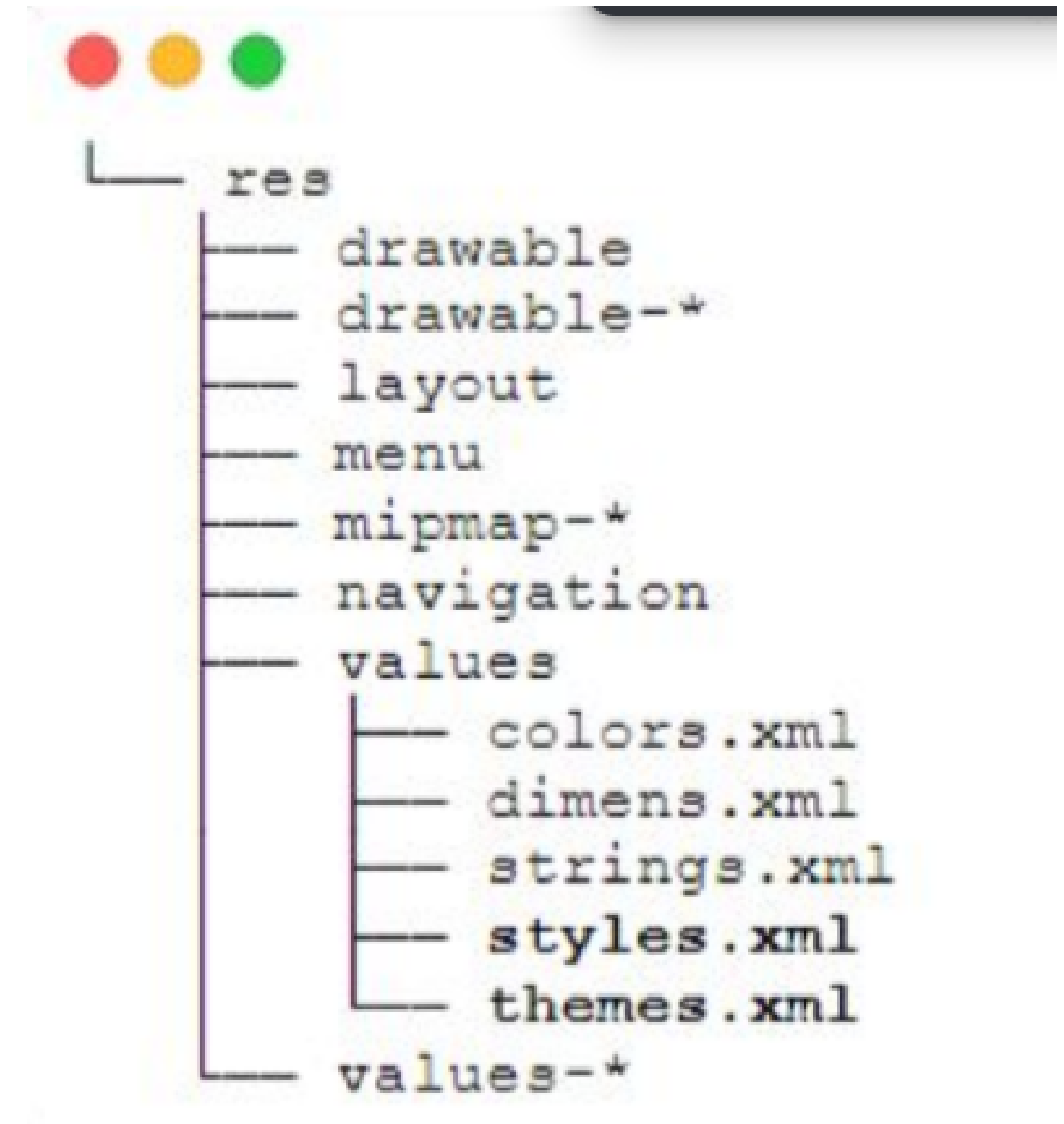
Untuk menerapkan Attribute View, ikuti langkah di bawah ini.

- Gunakan atribut view untuk mengubah satu view secara spesifik
- Kamu bisa menggunakan semua properti yang bisa diatur via styles atau themes
- Gunakanlah untuk desain sebuah view yang berbeda daripada umumnya seperti margins, padding, atau constraints



# Directory Resource

Kita bisa melakukan konfigurasi aplikasi kita dengan menggunakan file directory berikut.



# Color Resources

Colour Resources digunakan untuk menamai dan menstandarisasi warna-warna pada aplikasi. Kita dapat mendeklarasikannya pada **res/values/colors.xml**



```
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    ...
</resources>
```

# Dimension Resources

Cara ini digunakan untuk menamai dan menstandarisasi nilai dimensi pada aplikasi. Kita dapat mendeklarasikannya pada **res/values/dimens.xml**



Declare your dimension values in res/values/dimens.xml:

```
<resources>
    <dimen name="top_margin">16dp</dimen>
</resources>
```



Refer to them as @dimen/<name> in layouts or R.dimen.<name> in code:

```
<TextView ...
    android:layout_marginTop="@dimen/top_margin" />
```

# Typography

Typography ada 3 yaitu :

- Scalable Pixel (sp)
- Tipe scale
- TextAppearance

# Typography

## Scalable Pixels

Scalable Pixels (sp) adalah unit pengukuran yang harus kita gunakan untuk setiap hal yang berhubungan dengan teks.

# Typography

## Type Scale

Type Scale merujuk pada sebuah rangkaian styles yang didesain untuk bekerja bersama secara kohesif buat aplikasi dan konten kita, guys!

Type Scale berisi kategori teks yang dapat digunakan kembali dengan tujuan yang berbeda pada setiap kategorinya. Contoh Headline, subtitle, atau caption.

# Typography

## TextAppearance

TextAppearance adalah sebuah style yang telah ditentukan sebelumnya (predefined style) dalam type scale yang bisa diakses sebagai TextAppearance. Sebuah style TextAppearance sering mengubah satu atau lebih atribut di bawah ini.

- typeface (android:fontFamily)
- weight (android:textStyle)
- text size (android:textSize)
- capitalization (android:textAllCaps)
- letter spacing (android:letterSpacing)



# Material Design

Material Design adalah sistem pedoman, komponen, dan tools yang dapat disesuaikan yang mendukung best practices untuk desain UI.

Tools yang disediakan oleh Material Design memungkinkan kita untuk membuat keputusan desain tertentu (misalnya warna atau gaya tipografi) dan menerapkannya di seluruh UI aplikasi.

# Material Design

## Material Component

Material Component adalah blok interaktif untuk membuat antarmuka (user interface).

Selain itu, Material Component dilengkapi dengan cara bawaan (built-in) untuk mengkomunikasikan focus, selection, dan status lainnya.

# Material Design

## Menggunakan Material Component

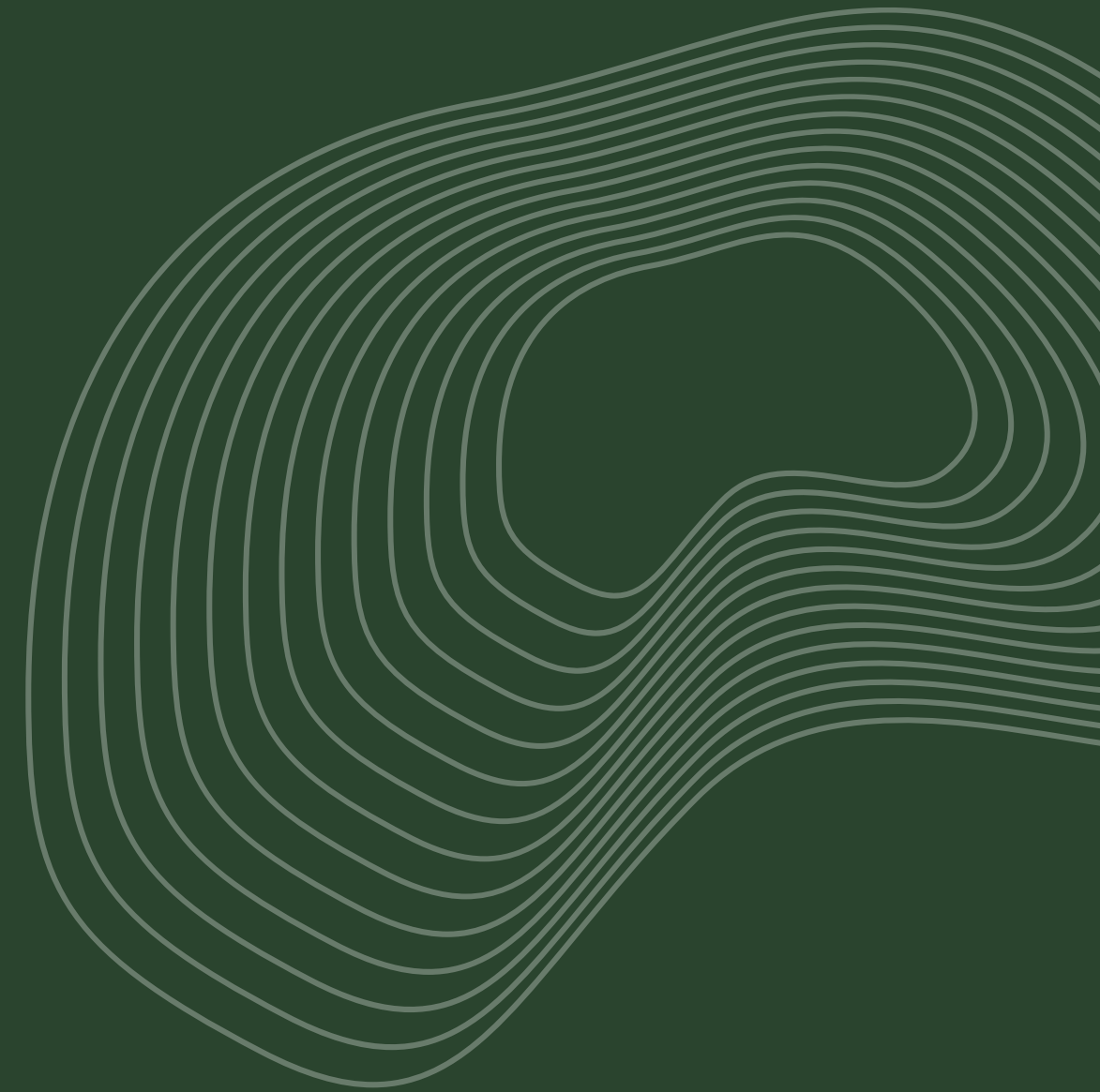
Untuk menggunakan komponen bawaan dan themes yang dijelaskan sebelumnya, kita bisa pakai Library Android Material Component.

Tinggal tambahkan dependensinya ke dalam project. Namun biasanya udah otomatis ditambahkan jika project dibuat berdasarkan template di Android Studio.



```
implementation 'com.google.android.material:material:<version>'
```

# LATIHAN



Slicing UI tersebut dengan  
menggunakan material component

## Login Page



 Email

 Password

SIGN UP

LOGIN

# Membuat file style.xml untuk mengatur style dari setiap view

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <style name="FontStyle">
4          <item name="android:textColor">@color/black</item>
5          <item name="android:textSize">30sp</item>
6          <item name="android:textStyle">bold</item>
7      </style>
8  </resources>
```

# Implementasi material design pada gradle module:app

47

48

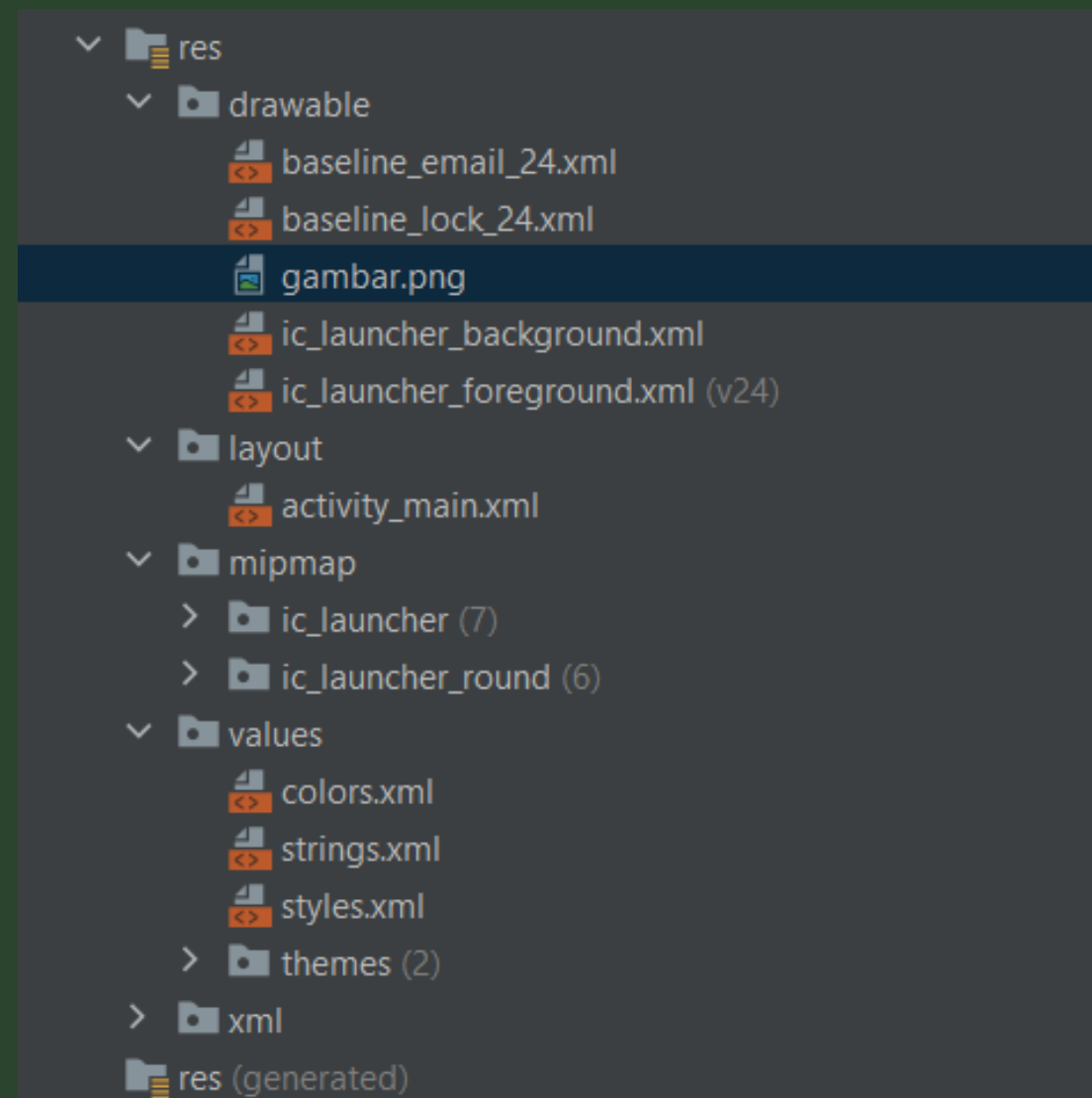
49

 implementation 'com.google.android.material:material:1.8.0'

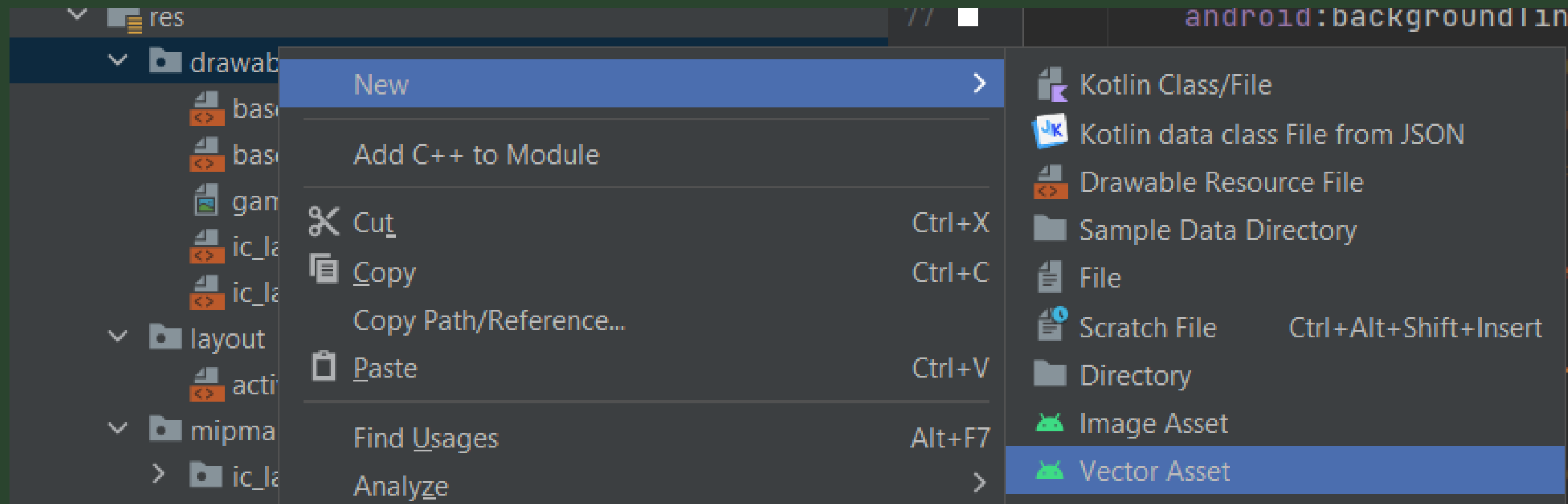
 }



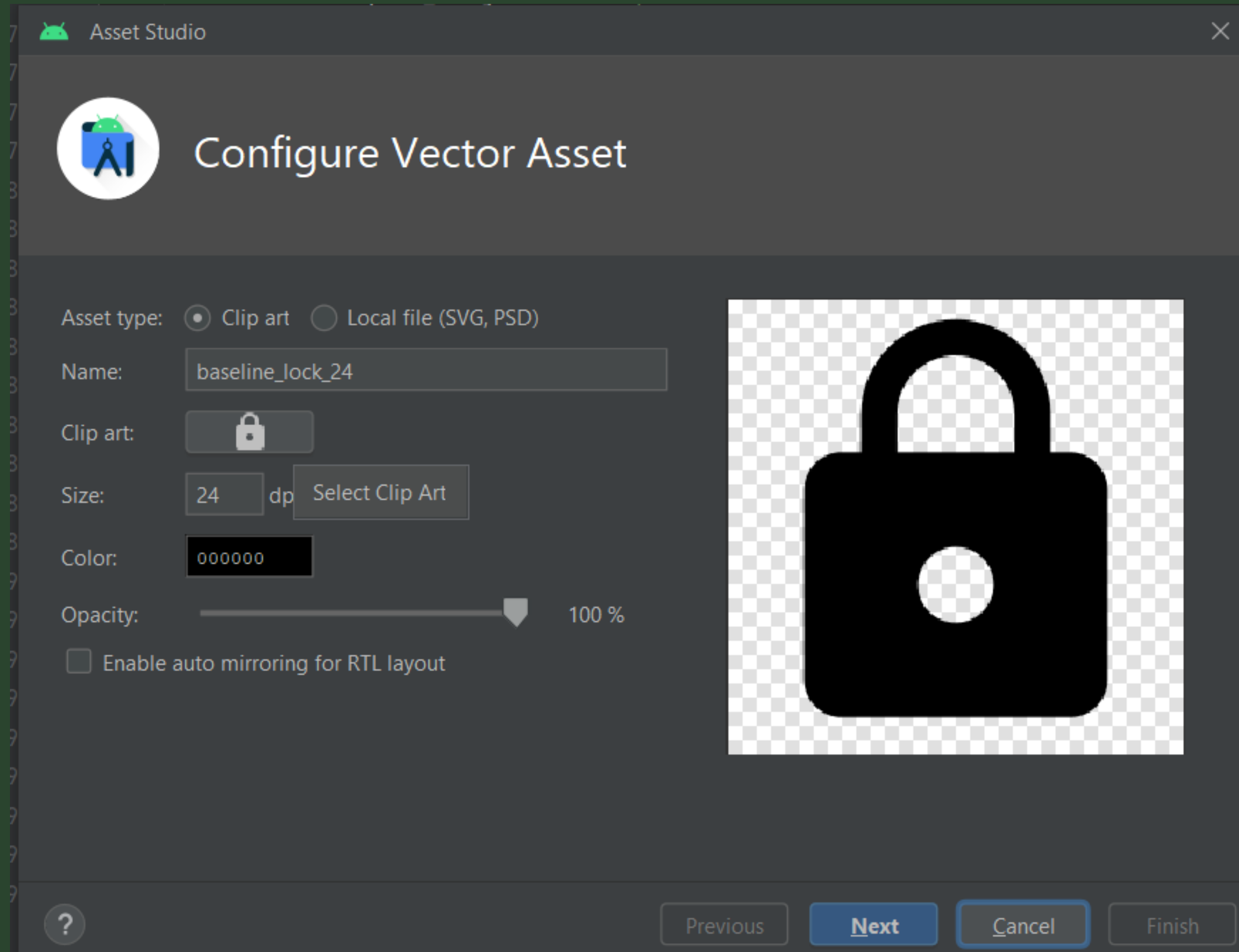
# Pindahkan gambar yang telah di download ke folder drawable



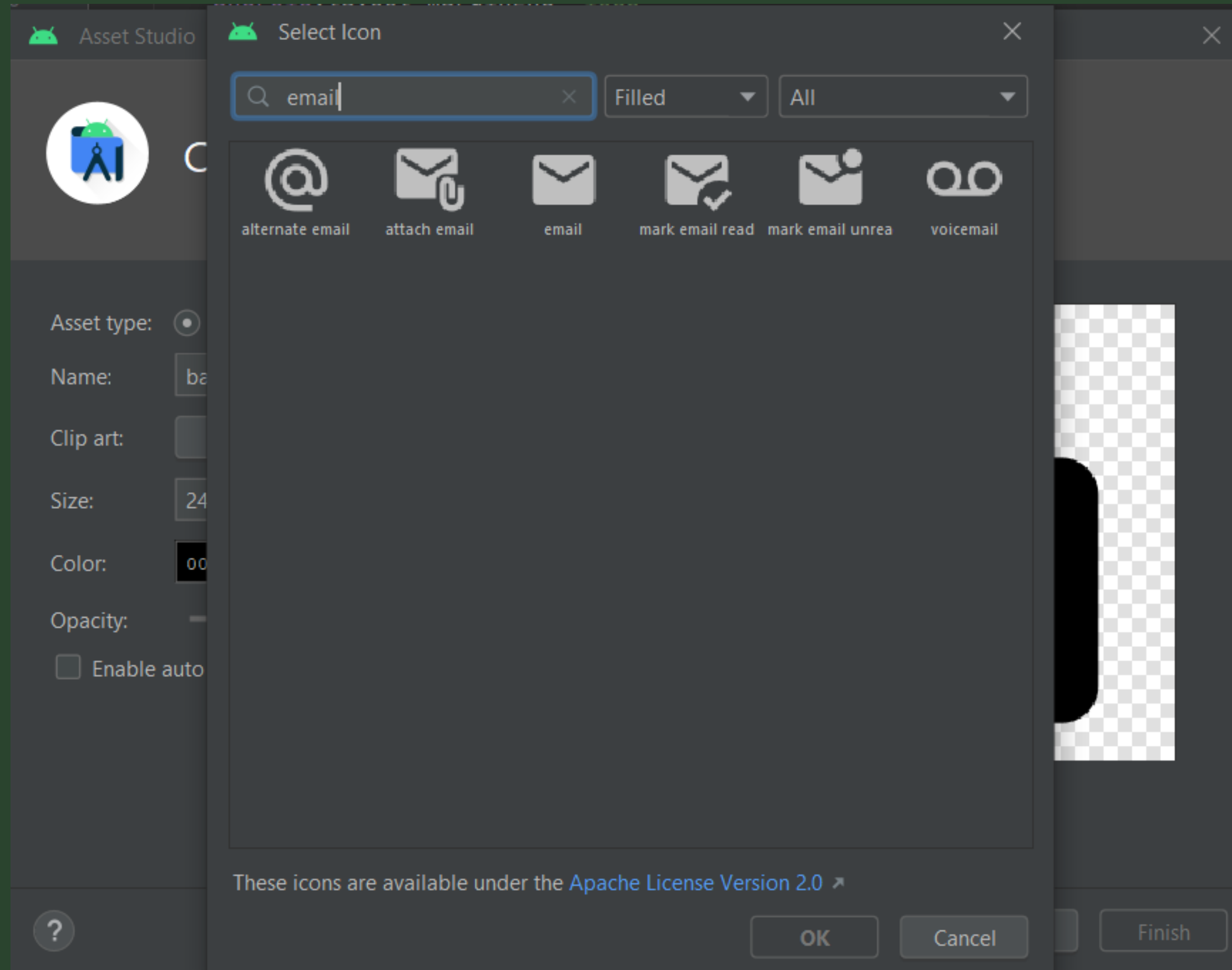
Tambahkan 2 vector baru berupa gembok dan email. Karena nanti akan berguna untuk styling di inputan email dan password



# Pilih Clip Art



Cari vectornya, dan klik ok. Setelah itu next saja dan klik finish.



Ikuti Kode Berikut ini pada  
activity\_main.xml

```
1      <?xml version="1.0" encoding="utf-8"?>
2      <RelativeLayout
3          xmlns:android="http://schemas.android.com/apk/res/android"
4          xmlns:app="http://schemas.android.com/apk/res-auto"
5          xmlns:tools="http://schemas.android.com/tools"
6          android:layout_width="match_parent"
7          android:layout_height="match_parent"
8          android:background="#F7F7F7"
9          tools:context=".MainActivity">
10
11          <TextView
12              android:layout_width="match_parent"
13              android:layout_height="wrap_content"
14              style="@style/FontStyle"
15              android:text="Login Page"
16              android:textAlignment="center"
17              />
18
19          <View
20              android:id="@+id/Image"
21              android:layout_width="150dp"
22              android:layout_height="150dp"
23              android:background="@drawable/gambar"
24              android:layout_centerHorizontal="true"
25              android:layout_marginTop="156dp"
26              />
27
28          <com.google.android.material.textfield.TextInputLayout
29              android:layout_marginTop="40dp"
```

Ikuti Kode Berikut ini pada  
activity\_main.xml

```
30         android:layout_below="@id/Image"
31         android:id="@+id/editText1"
32         android:padding="8dp"
33         android:layout_width="match_parent"
34         android:layout_height="wrap_content"
35         style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
36         android:hint="Email">
37
38         <com.google.android.material.textfield.TextInputEditText
39             android:inputType="textEmailAddress"
40             android:layout_width="match_parent"
41             android:layout_height="wrap_content"
42             android:drawableStart="@drawable/baseline_email_24"
43             android:drawablePadding="10dp"
44
45             />
46     </com.google.android.material.textfield.TextInputLayout>
47
48     <com.google.android.material.textfield.TextInputLayout
49         android:id="@+id/editText2"
50         android:layout_below="@id/editText1"
51         android:padding="8dp"
52         android:layout_width="match_parent"
53         android:layout_height="wrap_content"
54         style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
55         android:hint="@string/password">
56
57         <com.google.android.material.textfield.TextInputEditText
58             android:inputType="textPassword"
```

Ikuti Kode Berikut ini pada  
activity\_main.xml

```
59         android:layout_width="match_parent"
60         android:layout_height="wrap_content"
61         android:drawableStart="@drawable/baseline_lock_24"
62         android:drawablePadding="10dp"
63
64     />
65 </com.google.android.material.textfield.TextInputLayout>
66
67 <com.google.android.material.button.MaterialButton
68     android:padding="12dp"
69     android:id="@+id/btnBuatAkun"
70     android:layout_width="match_parent"
71     android:layout_height="wrap_content"
72     android:layout_below="@id/editText2"
73     android:layout_marginStart="10dp"
74     android:layout_marginTop="10dp"
75     android:layout_marginEnd="10dp"
76     android:layout_marginBottom="10dp"
77     android:backgroundTint="@color/white"
78     android:elevation="10dp"
79     android:text="@string/SignUp"
80     android:textColor="@color/black">
81
82 </com.google.android.material.button.MaterialButton>
83
84 <com.google.android.material.button.MaterialButton
85
86     android:padding="12dp"
87     android:elevation="10dp"
```



Ikuti Kode Berikut ini pada  
activity\_main.xml

```
87         android:elevation="10dp"
88         android:layout_marginStart="10dp"
89         android:layout_marginEnd="10dp"
90         android:layout_width="match_parent"
91         android:text="@string/login"
92         android:backgroundTint="#0095F6"
93         android:layout_below="@id/btnBuatAkun"
94         android:layout_height="wrap_content">
95
96     </com.google.android.material.button.MaterialButton>
97
98 </RelativeLayout>
```



# Terima Kasih

