

—
Log



Log

- Saat kita membuat kode program Kotlin atau Java, kita sering melakukan println atau menulis tulisan ke console
- Di Android, kita juga bisa melakukan hal tersebut dengan menggunakan class Log
- Lebih tepatnya Log ini digunakan untuk melakukan logging di Android
- <https://developer.android.com/reference/android/util/Log>



Log Level

- Berbeda dengan println biasanya, pada Log, terdapat yang namanya Level, yaitu level informasi yang ingin kita tampilkan
- Android mendukung beberapa jenis Log Level



Jenis Log Level

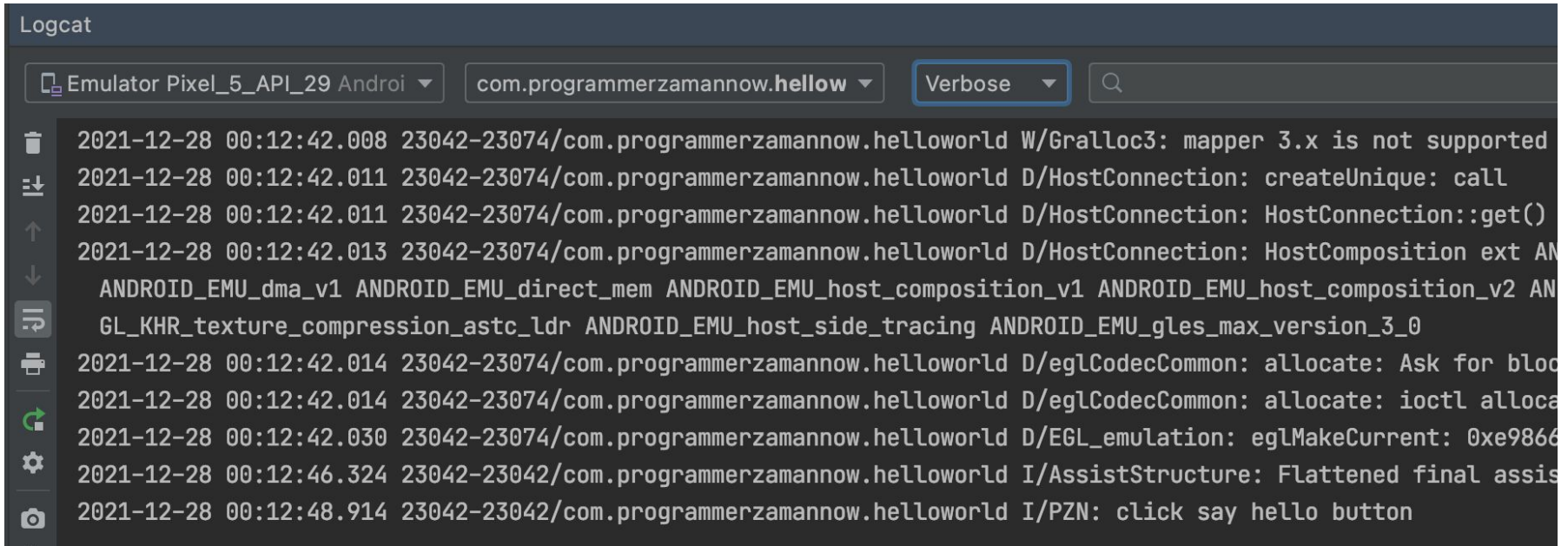
Log Level	Function
VERBOSE	Log.v(tag, message?, throwable?)
DEBUG	Log.d(tag, message?, throwable?)
INFO	Log.i(tag, message?, throwable?)
WARN	Log.w(tag, message?, throwable?)
ERROR	Log.e(tag, message?, throwable?)



Kode : Log

```
sayHelloButton.setOnClickListener { it: View!  
    Log.i( tag: "PZN", msg: "click say hello button")  
    val name = nameEditText.text.toString()  
    sayHelloTextView.text = "Hi $name"  
}  
}
```

Hasil Log



The screenshot shows the Logcat window in Android Studio. The top bar indicates the device is 'Emulator Pixel_5_API_29 Android'. The package name is 'com.programmerzamannow.helloworld' and the log level is set to 'Verbose'. The log messages are as follows:

```
2021-12-28 00:12:42.008 23042-23074/com.programmerzamannow.helloworld W/Gralloc3: mapper 3.x is not supported
2021-12-28 00:12:42.011 23042-23074/com.programmerzamannow.helloworld D/HostConnection: createUnique: call
2021-12-28 00:12:42.011 23042-23074/com.programmerzamannow.helloworld D/HostConnection: HostConnection::get()
2021-12-28 00:12:42.013 23042-23074/com.programmerzamannow.helloworld D/HostConnection: HostComposition ext AN
    ANDROID_EMU_dma_v1 ANDROID_EMU_direct_mem ANDROID_EMU_host_composition_v1 ANDROID_EMU_host_composition_v2 AN
    GL_KHR_texture_compression_astc_ldr ANDROID_EMU_host_side_tracing ANDROID_EMU_gles_max_version_3_0
2021-12-28 00:12:42.014 23042-23074/com.programmerzamannow.helloworld D/eglCodecCommon: allocate: Ask for bloc
2021-12-28 00:12:42.014 23042-23074/com.programmerzamannow.helloworld D/eglCodecCommon: allocate: ioctl allocat
2021-12-28 00:12:42.030 23042-23074/com.programmerzamannow.helloworld D/EGL_emulation: eglMakeCurrent: 0xe9866
2021-12-28 00:12:46.324 23042-23042/com.programmerzamannow.helloworld I/AssistStructure: Flattened final assis
2021-12-28 00:12:48.914 23042-23042/com.programmerzamannow.helloworld I/PZN: click say hello button
```



Resource



Resource

- Resource adalah file tambahan atau konten statis yang biasanya digunakan di kode program kita
- Seperti contohnya text, gambar, layout, animasi, dan lain-lain
- Android sendiri mendukung banyak sekali resource, kita akan bahas secara bertahap
- Pertanyaannya, bagaimana untuk mengambil data resource nya?
- Untuk mengambil data resource nya, kita bisa menggunakan class Resources
- <https://developer.android.com/reference/android/content/res/Resources>



Jenis-Jenis Resource (1)

Resource Directory	Keterangan
animator	XML file definisi property animations
anim	XML file definisi tween animations
color	XML file definisi warna
drawable	Bitmap files (png, jpg, gif) atau XML file drawable resource
mipmap	Drawable file untuk icon launcher



Jenis-Jenis Resource (2)

Resource Directory	Keterangan
layout	XML file definisi layout user interface
menu	XML file definisi app menu
raw	File yang utuh lainnya
values	XML file yang berisi value seperti string, integer dan lain-lain
xml	XML file yang bisa dibaca menggunakan Resources.getXML()
font	Font files



Pembahasan Resources

- Pada kelas ini, kita tidak akan membahas semua Resources yang ada di Android, karena tiap jenis resource perlu topik masing-masing
- Contohnya seperti animator atau anim, kita perlu mengenal penggunaan graphics terlebih dahulu di Android, sebelum menggunakan resource tersebut
- Pada kelas ini, kita akan fokus membahas jenis resource dasar yang memang bisa langsung digunakan seperti values atau layout



Resources di Activity

- Class Resources secara otomatis akan dibuatkan di dalam Android Context, dimana Activity adalah turunan dari Android Context
- Untuk mendapatkan object Resources, kita bisa langsung menggunakan function `getResources()`, atau jika menggunakan Kotlin, kita bisa langsung panggil property `resources`



Mengakses Resource

- Setiap kita membuat Resource, secara otomatis resource tersebut akan memiliki id yang secara otomatis ditambahkan ke class R
- Dengan demikian, untuk mengakses resource id nya di kode program kita, kita bisa langsung menggunakan class R
- Namun resource juga bisa dipanggil dari resource lainnya, pada kasus ini, kita bisa menggunakan id resource diawali dengan @ dan jenis resource, lalu diikuti dengan id resource, misal @string/id_string
- Contohnya pada layout, ketika kita membuat text di button, kita ingin mengisi text nya dari resource string misalnya



Kode : Resources

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.hello_world)  
  
    initComponents()  
  
    sayHelloTextView.text = resources.getText(R.string.app_name)
```

String Resource



Values Resource

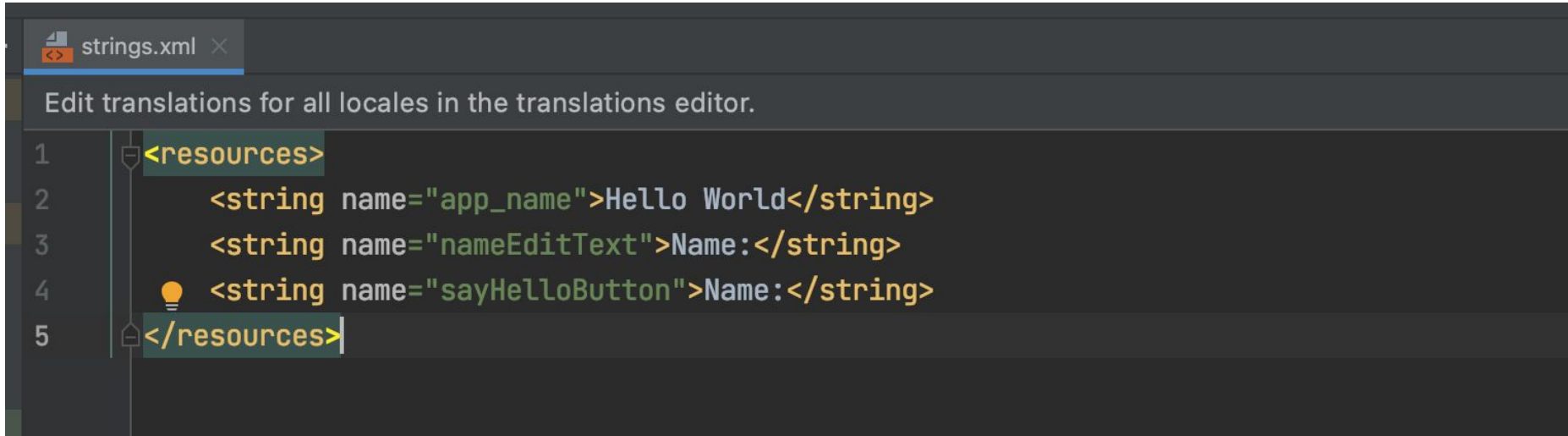
- Values resource merupakan jenis resource yang biasanya digunakan untuk menyimpan data-data statis yang digunakan di kode program kita, misal string, integer, boolean, color dan lain-lain



String Resource

- String merupakan resource yang berisi teks
- Rekomendasinya ketika kita membuat tulisan untuk kita tampilkan di halaman UI aplikasi Android kita, disarankan jangan meng-hardcode pada kode program, lebih baik menggunakan String resource
- Hal ini karena jika kita ingin mengubah text nya, tidak perlu mengubah kode program, selain itu data text nya bisa digunakan ulang di halaman UI berbeda

Kode : String Resource



The screenshot shows an IDE window titled "strings.xml" with a close button. Below the title bar, a message reads "Edit translations for all locales in the translations editor." The main editor area displays the following XML code:

```
1 <resources>
2     <string name="app_name">Hello World</string>
3     <string name="nameEditText">Name:</string>
4     <string name="sayHelloButton">Name:</string>
5 </resources>
```

The code is color-coded: the opening and closing tags of the root element are yellow, the opening and closing tags of the child elements are green, and the attribute values are blue. A lightbulb icon is visible next to the third line of code.



Kode : String Resource di Layout

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="@string/nameEditText" />

<EditText
    android:id="@+id/nameEditText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" />

<Button
    android:id="@+id/sayHelloButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="@string/sayHelloButton" />
```

Containers

Helpers

Google

Legacy

Switch





Formatting String

- Kadang saat menggunakan String resource, kita butuh membuat String yang datanya dinamis dan memiliki parameter, contoh pada kasus kita adalah tulisan Hello \$name
- Artinya \$name tersebut bisa berubah-ubah
- Untung nya String resource mendukung hal tersebut, kita bisa menggunakan formatting string pada string resource, caranya cukup gunakan format %index\$s untuk parameter string, atau %index\$d untuk angka desimal

Kode : String Resource

strings.xml

Edit translations for all locales in the translations editor.

```
1 <resources>
2     <string name="app_name">Hello World</string>
3     <string name="nameEditText">Name:</string>
4     <string name="sayHelloButton">Name:</string>
5     <string name="sayHelloTextView">Hello %1$s</string>
6 </resources>
```



Kode : Formatting String

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.hello_world)  
  
    initComponents()  
  
    sayHelloButton.setOnClickListener { it: View!  
        Log.i( tag: "PZN", msg: "click say hello button")  
        val name = nameEditText.text.toString()  
        sayHelloTextView.text = resources.getString(R.string.sayHelloTextView, name)  
    }  
}
```



String Array Resource

- Values resource juga bisa kita tambahkan tipe resource berupa String Array
- Kita bisa menggunakan tag `<string-array>` untuk menambahkan tipe String Array Resource
- Dan di dalamnya untuk menambahkan tiap datanya, kita bisa gunakan tag `<item>`
- String Array Resource secara otomatis akan terdapat di property array di class R
- Untuk mengambil String Array Resource, kita bisa gunakan function `getStringArray(resourceId)`

Kode : String Array Resource

```
<string name="sayHelloTextView">Hello %1$s</string>
<string-array name="names">
    <item>Eko</item>
    <item>Kurniawan</item>
    <item>Khannedy</item>
</string-array>
<resources>
```




Kode : Mengambil String Array Resource

```
sayHelloButton.setOnClickListener { it: View!  
    Log.i( tag: "PZN", msg: "click say hello button")  
    val name = nameEditText.text.toString()  
  
    resources.getStringArray(R.array.names).forEach { it: String!  
        Log.i( tag: "StringArray", it)  
    }  
}
```

Value Resource Lainnya



Value Resource Lainnya

- Selain String Resource dan String Array Resource, di dalam Value Resource, kita bisa menambahkan banyak jenis Resource Lainnya
- Misalnya Integer, Integer Array, Boolean, Color dan lain-lain
- <https://developer.android.com/guide/topics/resources/more-resources?hl=id>

Kode : Value Resource Lainnya

```
other.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <integer name="maxPaging">100</integer>
4      <integer-array name="numbers">
5          <item>100</item>
6          <item>200</item>
7          <item>300</item>
8      </integer-array>
9      <bool name="isProductionMode">true</bool>
10     <color name="background">#FF0000</color>
11 </resources>
```



Kode : Menggunakan Value Resource Lainnya

```
sayHelloButton.setOnClickListener { it: View!  
  
    Log.i( tag: "VaLueResource", resources.getBoolean(R.bool.isProductionMode).toString())  
    Log.i( tag: "VaLueResource", resources.getInteger(R.integer.maxPaging).toString())  
    Log.i( tag: "VaLueResource", resources.getIntArray(R.array.numbers).joinToString( separator: ", "))  
    Log.i( tag: "VaLueResource", resources.getColor(R.color.background, theme).toString())  
  
    sayHelloButton.setBackgroundColor(resources.getColor(R.color.background, theme))
```

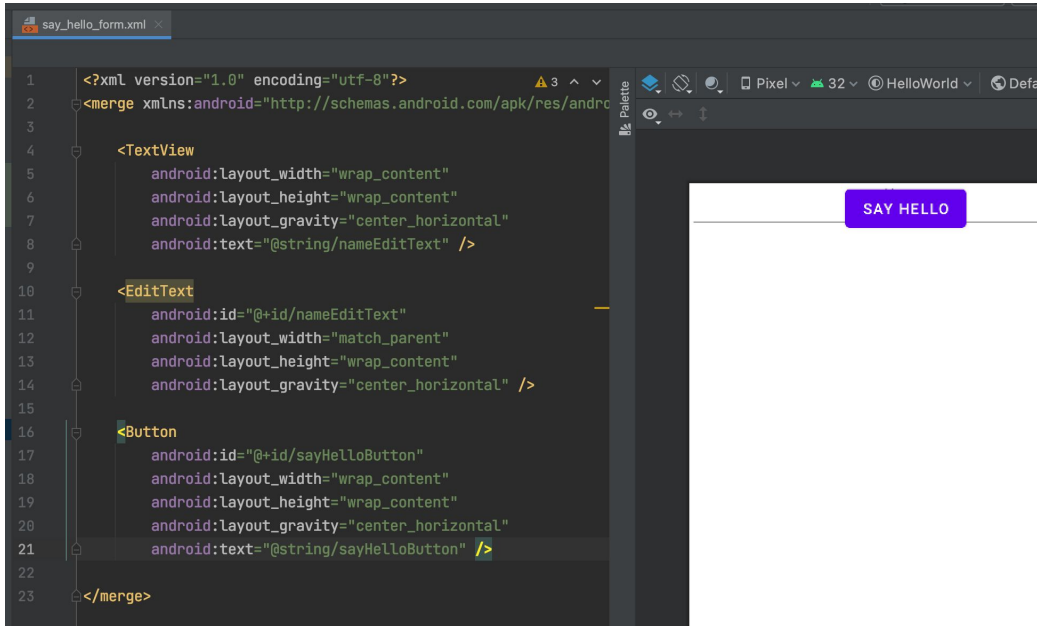
Layout Resource



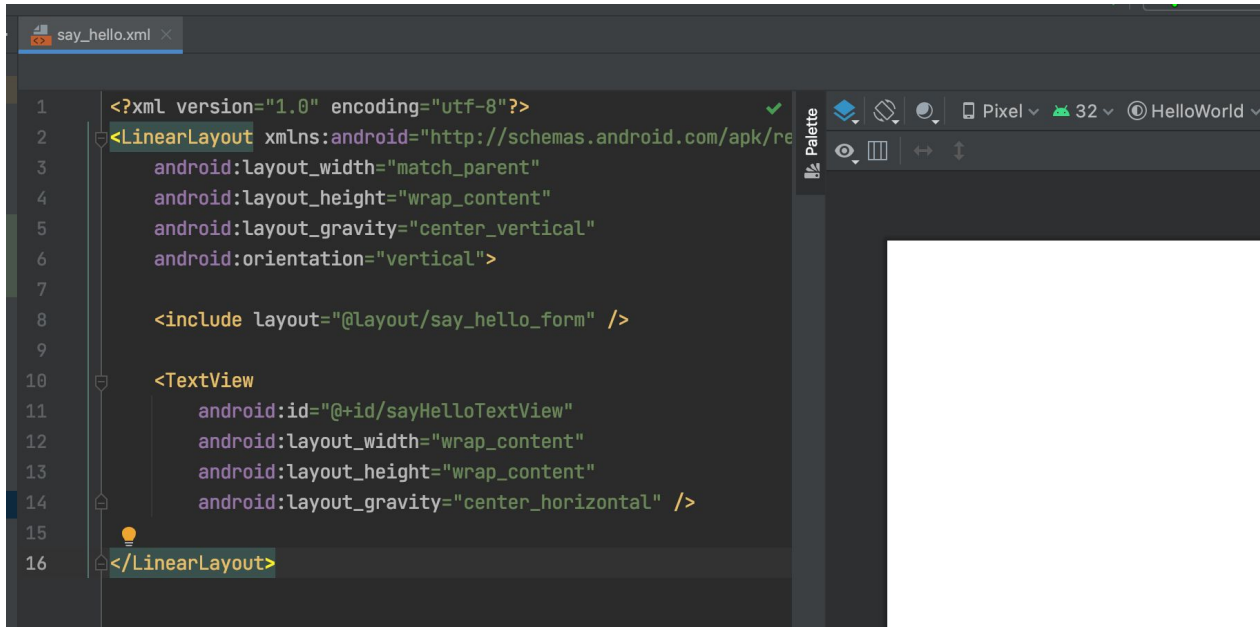
Layout Resource

- Layout resource adalah definisi dari tampilan untuk UI
- Di dalam layout, kita bisa mendefinisikan isi View atau ViewGroup
- View adalah single komponen, sedangkan ViewGroup adalah container atau wadah untuk satu atau lebih komponen View
- Contoh ViewGroup seperti LinearLayout, RelativeLayout, FrameLayout, dan lain-lain
- Layout juga bisa menambahkan layout lain, dengan menggunakan tag `<include>`
- Setiap layout harus memiliki satu root element, jika misal kita tidak ingin memiliki root element, misal untuk digunakan sebagai include di layout lain, kita bisa gunakan root tag `<merge>`

Kode : Layout



Kode : Layout Include



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="wrap_content"
5     android:layout_gravity="center_vertical"
6     android:orientation="vertical">
7
8     <include layout="@layout/say_hello_form" />
9
10    <TextView
11        android:id="@+id/sayHelloTextView"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:layout_gravity="center_horizontal" />
15
16 </LinearLayout>
```



Kode : Menggunakan Layout

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.say_hello)  
  
    initComponents()  
  
    sayHelloButton.setOnClickListener { it: View!  
        Log.i( tag: "PZN", msg: "click say hello button")  
        val name = nameEditText.text.toString()  
        sayHelloTextView.text = "Hello {name}"  
    }
```

Color State List Resource



Color State List Resource

- Color State List Resource merupakan object yang bisa kita buat dalam XML yang merupakan representasi Color (Warna), tapi tidak satu Color, melainkan bisa beberapa Color tergantung state dari View nya
- Color State List Resource di representasikan dalam class `ColorStateList`
<https://developer.android.com/reference/android/content/res/ColorStateList>
- Color State List Resource juga bisa menggunakan Color Resource yang sudah kita buat di Values Resource dengan menggunakan `@color/namaResource`
- <https://developer.android.com/guide/topics/resources/color-list-resource?hl=id>

Kode : Color State List Resource

```
say_hello.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <selector xmlns:android="http://schemas.android.com/apk/res/android">
3      <item android:color="#FFFF00" android:state_pressed="true" />
4      <item android:color="#FFFFFF" android:state_focused="true" />
5      <item android:color="@color/background" />
6  </selector>
7
```

Kode : Menggunakan Color State List

```
<Button
```

```
    android:id="@+id/sayHelloButton"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_gravity="center_horizontal"
```

```
    android:text="@string/sayHelloButton"
```

```
    android:textColor="@color/say_hello"/>
```

Drawable Resource

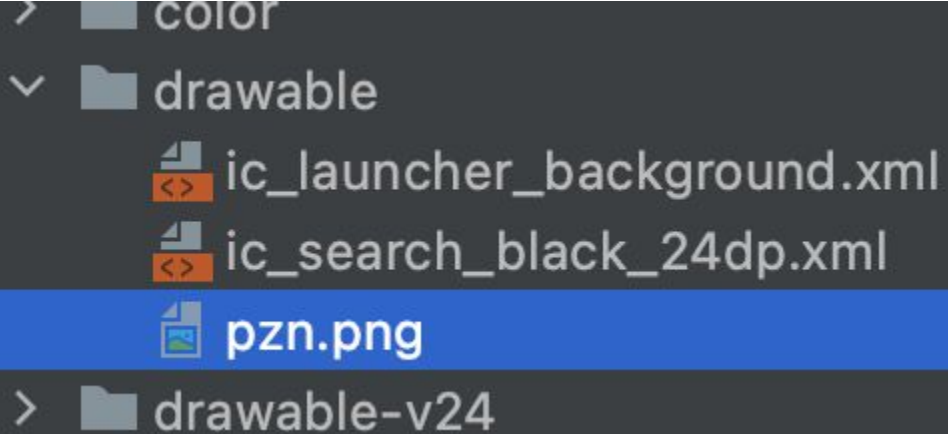


Drawable Resource

- Drawable Resource adalah jenis resource dengan konsep graphics yang bisa digambar di layar
- Ada banyak jenis Drawable, seperti Bitmap File (file gambar), Nine Patch File, Layer List, State List dan lain-lain
- <https://developer.android.com/guide/topics/resources/drawable-resource?hl=id>
- Drawable direpresentasikan dalam class Drawable, dan untuk mendapatkannya, kita bisa gunakan function `getDrawable(resourceId)` pada class Resources
- <https://developer.android.com/reference/android/graphics/drawable/Drawable>

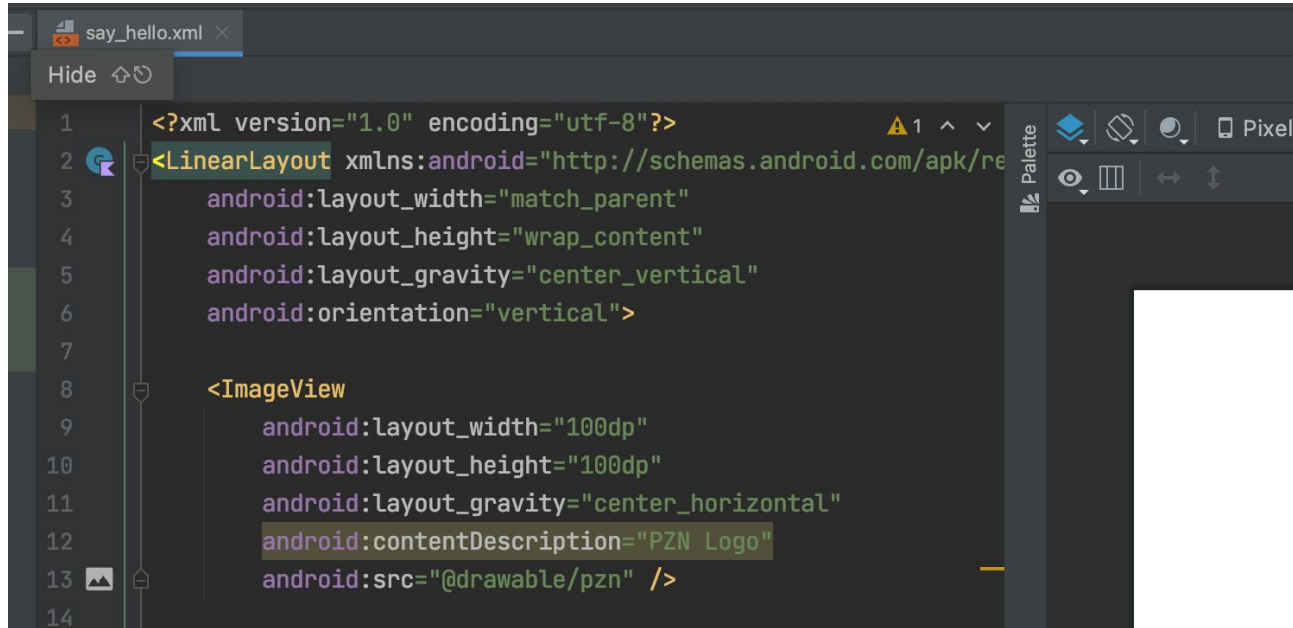


Menambah Drawable



```
> color
v drawable
  ic_launcher_background.xml
  ic_search_black_24dp.xml
  pzn.png
> drawable-v24
```

Kode : Menggunakan Drawable



```
say_hello.xml x
Hide ↻
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="wrap_content"
5     android:layout_gravity="center_vertical"
6     android:orientation="vertical">
7
8     <ImageView
9         android:layout_width="100dp"
10        android:layout_height="100dp"
11        android:layout_gravity="center_horizontal"
12        android:contentDescription="PZN Logo"
13        android:src="@drawable/pzn" />
14
```

Localization



Localization

- Saat kita membuat aplikasi Android, kadang aplikasi yang kita buat akan diakses dari berbagai negara menggunakan berbagai bahasa
- Agar aplikasi kita bisa digunakan oleh banyak orang, ada baiknya kita membuat text, files, number, gambar dan lain-lain sesuai dengan lokasi pengguna
- Android mendukung fitur Localization, dimana kita membuat resource yang kita buat bisa menyesuaikan dengan bahasa yang digunakan oleh pengguna aplikasi kita



Default Resource

- Saat kita membuat resource, maka Android akan mengakses default resource, contoh default values resource akan diambil dari `res/values/namafile.xml`
- Jika tidak ada default resource, maka akan terjadi error yang menyebabkan aplikasi kita tidak bisa berjalan

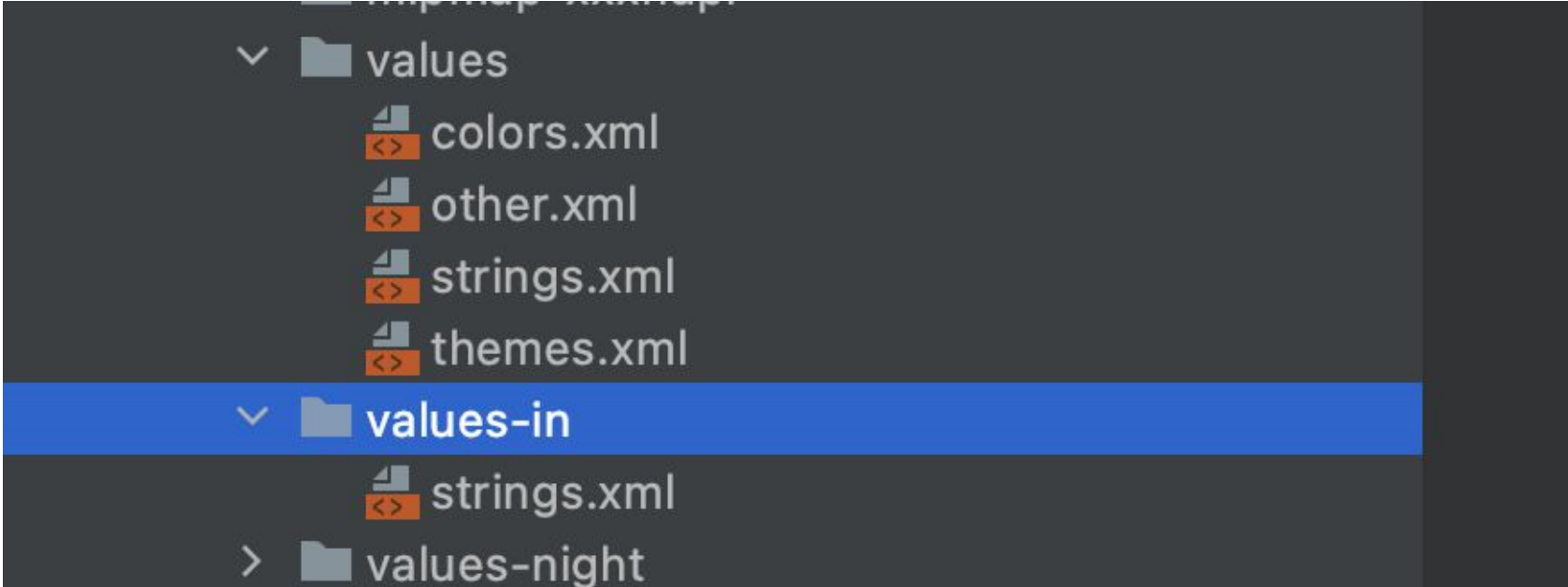


Localization Resource

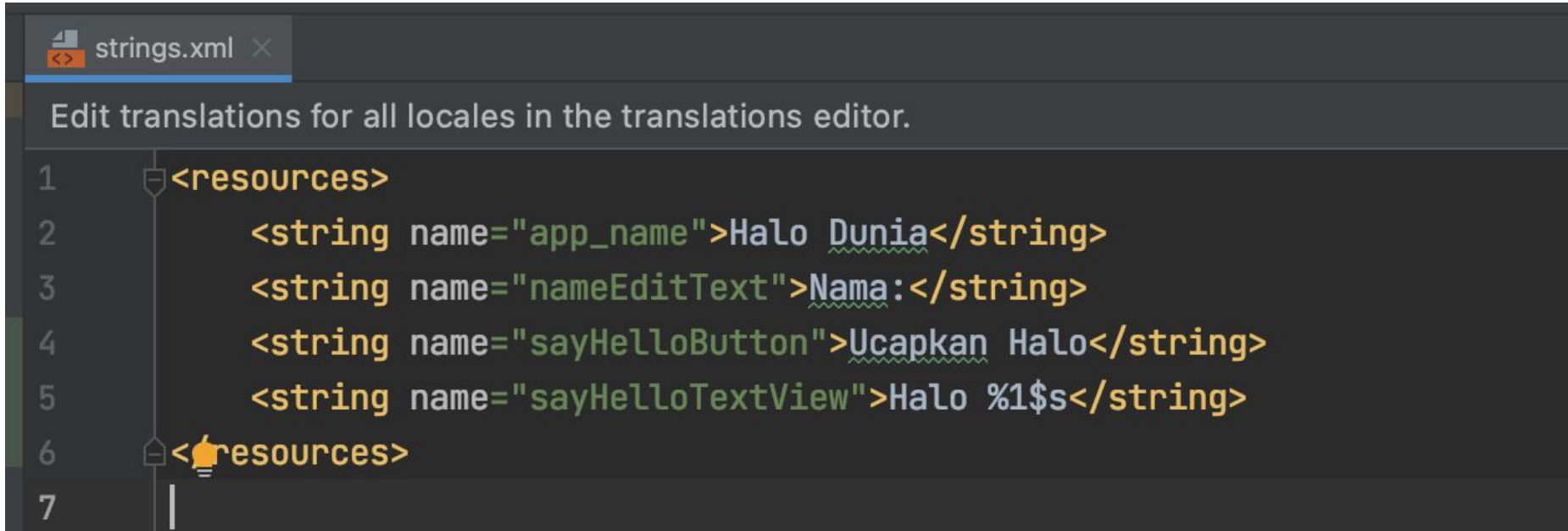
- Setelah kita membuat default resource, selanjutnya kita bisa membuat resource alternatif untuk bahasa lain
- Caranya kita bisa membuat resource dengan nama `res/{type}-{locale}/namafile.xml`
- Misal `res/values-in/strings.xml` untuk values resource bahasa Indonesia
- Misal `res/drawable-in/pzn.png` untuk drawable resource bahasa Indonesia
- Saat kita membuat resource alternatif, jika ternyata resource dengan id yang dimaksud tidak tersedia, maka secara otomatis akan menggunakan default resource



Resource Localization

- 
- values
 - colors.xml
 - other.xml
 - strings.xml
 - themes.xml
 - values-in**
 - strings.xml
 - values-night

Kode : Values Resource Localization



The screenshot shows an IDE window titled "strings.xml" with a close button. Below the title bar, a message reads "Edit translations for all locales in the translations editor." The main area displays XML code for resource localization. The code is as follows:

```
1 <resources>
2     <string name="app_name">Halo Dunia</string>
3     <string name="nameEditText">Nama:</string>
4     <string name="sayHelloButton">Ucapkan Halo</string>
5     <string name="sayHelloTextView">Halo %1$s</string>
6 </resources>
7
```

The code is color-coded: XML tags are orange, attribute names are green, and string values are blue. Underlines are present under "Dunia", "Nama:", "Ucapkan", and "%1\$s". A vertical line with a small icon is on the left side of the code editor.

Asset Manager



Asset Manager

- Kadang kita ingin menambahkan resource di Aplikasi Android kita, tapi bukan resource yang di manage oleh Android, misal kita ingin menambahkan file JSON atau TXT misalnya
- Untuk kasus seperti ini, kita bisa menggunakan AssetManager
- AssetManager adalah class yang bisa kita gunakan untuk mengakses resource secara manual
- <https://developer.android.com/reference/android/content/res/AssetManager>
- AssetManager akan mengambil resource pada directory assets, sehingga kita perlu membuatnya terlebih dahulu
- Dan untuk mendapatkan AssetManager, kita bisa gunakan function getAssets() di Context / Activity



File Asset

