**MODUL PRAKTIKUM**

**STYLE DAN THEME**

**Teori**

Pada modul ini, Anda akan mempelajari prinsip desain yang ada di Android. Anda juga akan mempelajari bagaimana menerapkan struktur dan tampilan viewdalam sebuah berkas style.

Prinsip dasar dalam merancang antarmuka aplikasi Android harus mematuhi kaidah yang ditetapkan oleh Design Guideline. Guideline ini dibuat oleh tim Android di Google. Beberapa prinsipnya adalah:

1. Menampilkan informasi yang hanya dibutuhkan.
2. Jika aplikasi meminta izin pengguna untuk melakukan sebuah aksi, maka pengembang harus menyediakan mekanisme untuk membatalkan izin tersebut.
3. Lakukan interupsi jika diperlukan.
4. Menggunakan teks secara singkat. Gunakan gambar untuk menjelaskan informasi secara lebih deskriptif.
5. Jaga data pengguna.
6. Permudah pengguna untuk melakukan sesuatu yang penting secara cepat.
7. Jika terlihat sama, maka perilaku haruslah sama.
8. Bantu pengguna untuk membuat keputusan tapi tetap biarkan pengguna menentukan keputusannya.

**Best Practice**

Terdapat beberapa langkah terbaik (*best practice*) yang harus diperhatikan ketika mengembangkan sebuah aplikasi Android, di antaranya:

1. **Desain yang baik untuk performa aplikasi**  
   Aplikasi yang dirancang dengan baik harus dapat dijalankan dengan cepat dan jika terdapat proses yang memakan waktu, maka jalankan di *background* dan *asynchronous.*
2. **Desain yang baik agar aplikasi dapat bersifat responsif**  
   Berikan *feedback* ke pengguna terhadap aksi yang dilakukannya. Contohnya, jika pengguna menekan sebuah tombol, maka aplikasi harus menampilkan efek tekan.
3. **Desain yang mengakomodasi kebutuhan informasi pengguna**  
   Aplikasi Anda harus menampilkan informasi yang dibutuhkan pengguna. Bila diperlukan, aplikasi perlu menampilkan informasi terakhir yang diperoleh, sehingga pengguna tidak perlu lagi menunggu aplikasi memuat *data* dari *server*.
4. **Desain untuk optimasi pengunaan baterai**  
   Usahakan agar aplikasi menggunakan daya baterai yang kecil. Minimalisir penggunaan *background service* yang tidak perlu. Berhentikan semua *listener* jika aplikasi tidak sedang dijalankan. Manfaatkan alarmmanager dan jobscheduler jika memang terdapat *task* yang harus dilakukan secara berkala.
5. **Desain untuk efisiensi pengunaan koneksi jaringan**  
   Aplikasi yang baik adalah yang efisien dalam memanfaatkan koneksi ke jaringan internet. Ia memilah-milah *task* mana yang perlu dijalankan saat perangkat pengguna terhubung ke wifi *(unmetered network)* atau pun network lain. Penggunaan koneksi jaringan yang baik akan menjadi hal wajib jika aplikasi Anda ingin tetap digunakan oleh pengguna.

Nilai yang bagus pada poin tampilan akan mendukung kualitas fungsi aplikasi yang dibuat. Pengguna akan mempertahankan aplikasi Anda dan tetap menggunakannya selama aplikasi dibutuhkan dan memenuhi poin-poin di atas.

Kembali pada topik style dan theme. Jika Anda pernah mengembangkan sebuah aplikasi berbasis web, maka sudah pasti tidak asing lagi dengan CSS *(Cascading Style Sheet)*. Ia mengatur tampilan dari sebuah halaman website. Pendekatan yang serupa juga berlaku di Android. Inilah yang dinamakan style.

**Style** merupakan sebuah kumpulan properti yang dibutuhkan untuk mendefinisikan bagaimana sebuah komponen view dan layar jendela (bisa activity maupun fragment) ditampilkan. Contoh properti ini adalah height, width, background\_color.

Style terdefinisi dalam file xml sendiri. Anda bisa menemukannya di **res →  values  →  styles.xml**.

Sebagai contoh, Anda memiliki sebuah textview yang berisi berbagai atribut seperti contoh kode di bawah ini. Textview ini berguna untuk menampilkan konten dari detail informasi yang terdapat di keseluruhan aplikasi.

|  |
| --- |
| 1. <TextView 2. android:layout\_width="match\_parent" 3. android:layout\_height="wrap\_content" 4. android:textColor="#00FF00" 5. android:typeface="monospace" 6. android:text="@string/hello" /> |

Sangat tidak efektif jika kita melakukan copy paste dari satu layout xml ke layout xml lainnya. Kita dapat menyederhanakan hal tersebut menjadi:

|  |
| --- |
| 1. <TextView 2. style="@style/CodeFont" 3. android:text="@string/hello" /> |

Attribute layout\_width, layout\_height, textcolor, dan typeface bisa kita pindahkan menjadi sebuah style sendiri untuk textview tersebut dan dapat digunakan kembali untuk semua obyek textview sejenis.

|  |
| --- |
| 1. *<?*xml version="1.0" encoding="utf-8"*?>* 2. <resources> 3. <style name="CodeFont" parent="@android:style/TextAppearance.Medium"> 4. <item name="android:layout\_width">match\_parent</item> 5. <item name="android:layout\_height">wrap\_content</item> 6. <item name="android:textcolor">#00FF00</item> 7. <item name="android:typeface">monospace</item> 8. </style> 9. </resources> |

Beberapa aturan yang harus diperhatikan ketika kita menggunakan styles yaitu:

1. Semua style yang dibuat harus berada dalam tag resources.
2. Semua style yang ingin didefinisikan harus berada dalam tag style.

|  |
| --- |
| 1. <style name="CodeFont" parent="@android:style/TextAppearance.Medium"> |

**Name**      :    Nama dari style yang Anda buat.

**Parent**:     Nilai style yang akan mewarisi style (termasuk *attribute* di dalamnya) yang telah ada, umumnya bawaan dari sdk ataupun platform.  
Style yang diwarisi akan dapat diubah dan ditambahkan atributnya dalam style baru yang Anda buat. Android sudah menyediakan beragam style yang bisa Anda gunakan untuk beragam tampilan.

1. Semua atribut yang didefinisikan dalam sebuah style harus berada dalam tag item.

|  |
| --- |
| 1. <item name="android:layout\_width">match\_parent</item> |

**Name**                  :    Nama atribut yang ingin didefinisikan.

**Match\_parent**:    Nilai dari atribut tersebut.

Andaikan dalam satu kasus Anda ingin membuat turunan dari style yang telah Anda buat. Misalnya Anda ingin membuat styleCodeFont berwarna merah, Anda dapat melakukannya dengan cara berikut ini:

|  |
| --- |
| 1. <style name="CodeFont.Red"> 2. <item name="android:textColor">#FF0000</item> 3. </style> |

atau berwarna merah dan juga dengan ukuran yang besar menjadi seperti ini :

|  |
| --- |
| 1. <style name="CodeFont.Red"> 2. <item name="android:textColor">#FF0000</item> 3. <item name="android:textSize">30sp</item> 4. </style> |

Mudah bukan? Anda baru saja belajar tentang bagaimana sebuah style dibuat dan diimplementasikan. Selanjutnya Anda akan mempelajari theme.

**Theme atau tema** itu sendiri merupakan sebuah style yang diterapkan khusus untuk activity dan application pada berkas AndroidManifest.xml. Pada proyek sebelumnya, kita mendefinisikannya dengan cara berikut ini:

|  |
| --- |
| 1. android:theme="@style/AppTheme" |

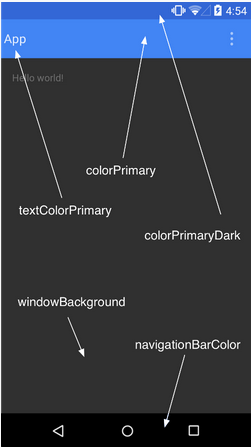
Di mana AppTheme pada styles.xml berisi :

|  |
| --- |
| 1. <resources> 2. <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar"> 3. *<!-- Customize your theme here. -->* 4. <item name="colorPrimary">@color/colorPrimary</item> 5. <item name="colorPrimaryDark">@color/colorPrimaryDark</item> 6. <item name="colorAccent">@color/colorAccent</item> 7. </style> 8. </resources> |

Sebuah styles yang inherit ke tema AppCompat untuk varian light dan memiliki DarkActionBar. Semua nilai pada atribut terdapat pada berkas colors.xml yang berisi.

|  |
| --- |
| 1. <resources> 2. <color name="colorPrimary">#3F51B5</color> 3. <color name="colorPrimaryDark">#303F9F</color> 4. <color name="colorAccent">#FF4081</color> 5. </resources> |

Saat ini, pengembangan aplikasi Android mengacu pada implementasi prinsip material design untuk merancang *user interface* dan *us*er *experience*. Anda bisa mempelajari lebih lanjut tentang material design pada tautan berikut: [Material Design](https://developer.android.com/design/index.html)



Semenjak material design ditetapkan sebagai acuan utama untuk pengembangan *user interface* dan *user experience*, terdapat penyederhanaan komponen inti yang digunakan. Pada gambar di atas, terlihat titik-titik mana saja yang hanya kita gunakan untuk menjadi fondasi dasar style aplikasi yang dibuat.

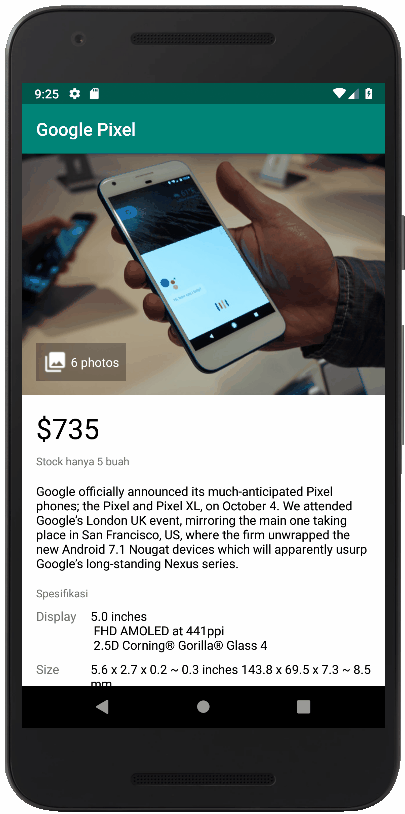
Bila Anda tertarik untuk mengetahui prinsip sebelum material design, Anda dapat melihat video berikut ini: [Video From Holo to Material](https://www.youtube.com/watch?v=x5-ntYM_2UY)

**Praktikum**

Pada praktikum kali kita akan menerapkan style dan theme pada contoh aplikasi yang telah kita buat sebelumnya pada modul Views dan Viewgroup. Beberapa poin yang didapatkan pada materi ini adalah:

1. Bagaimana memanfaatkan style pada aplikasi?
2. Bagaimana memanfaatkan theme pada aplikasi?

Hasil dari aplikasi yang telah dibuat akan jadi seperti ini.



**Praktikum Style dan Theme**

1. Buka kembali proyek sebelumnya (MyViewAndViews) atau unduh melalu repository masing-masing dengan cara clone/pull projeck melalui git.
2. Buka berkas colors.xml di **res → values → colors.xml**. Ubah setiap nilai yang ada saat ini dengan nilai-nilai berikut :

|  |
| --- |
| 1. *<?*xml version="1.0" encoding="utf-8"*?>* 2. <resources> 3. <color name="colorPrimary">**#607D8B**</color> 4. <color name="colorPrimaryDark">**#455A64**</color> 5. <color name="colorAccent">**#FF5722**</color> 6. **<color name="colorSubTitle">#757575</color>** 7. </resources> |

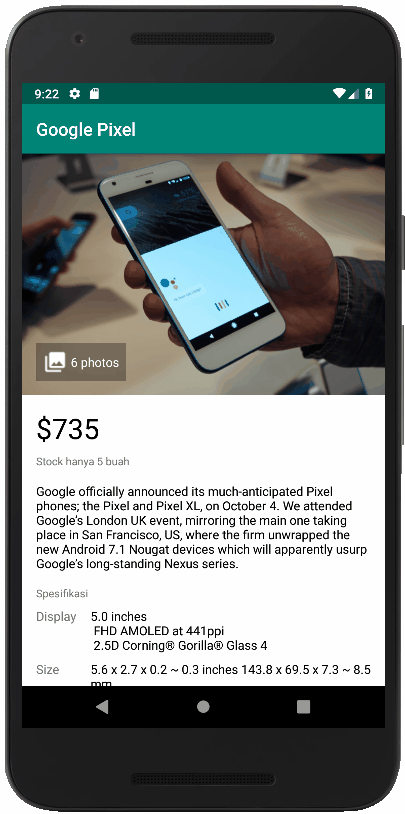
1. Selanjutnya, buka berkas styles.xml di **res → values → styles.xml** dan tambahkan beberapa style seperti di bawah ini :

|  |
| --- |
| 1. <resources> 3. <!-- Base application theme. --> 4. <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar"> 5. <!-- Customize your theme here. --> 6. <item name="colorPrimary">@color/colorPrimary</item> 7. <item name="colorPrimaryDark">@color/colorPrimaryDark</item> 8. <item name="colorAccent">@color/colorAccent</item> 9. </style> 10. **<style name="TextContent" parent="@style/TextAppearance.AppCompat">** 11. **<item name="android:layout\_width">wrap\_content</item>** 12. **<item name="android:layout\_height">wrap\_content</item>** 13. **<item name="android:layout\_marginLeft">@dimen/activity\_horizontal\_margin</item>** 14. **<item name="android:layout\_marginRight">@dimen/activity\_horizontal\_margin</item>** 15. **</style>** 16. **<style name="TextContent.Small">** 17. **<item name="android:textAppearance">@style/TextAppearance.AppCompat.Small</item>** 18. **</style>** 19. **<style name="TextContent.Small.White">** 20. **<item name="android:textColor">@android:color/white</item>** 21. **</style>** 22. **<style name="TextContent.Subtitle">** 23. **<item name="android:textColor">@color/colorSubTitle</item>** 24. **<item name="android:textSize">12sp</item>** 25. **</style>** 26. **<style name="TextContent.Black">** 27. **<item name="android:textColor">@android:color/black</item>** 28. **</style>** 29. **<style name="TextSpec">** 30. **<item name="android:textSize">14sp</item>** 31. **</style>** 32. **<style name="TextSpec.Field">** 33. **<item name="android:layout\_width">wrap\_content</item>** 34. **<item name="android:layout\_height">wrap\_content</item>** 35. **<item name="android:layout\_marginRight">16dp</item>** 36. **<item name="android:textColor">@color/colorSubTitle</item>** 37. **</style>** 38. **<style name="TextSpec.Value">** 39. **<item name="android:layout\_width">match\_parent</item>** 40. **<item name="android:layout\_height">wrap\_content</item>** 41. **<item name="android:layout\_weight">1</item>** 42. **<item name="android:textColor">@android:color/black</item>** 43. **</style>** 45. </resources> |

1. Jika selesai, kita akan implementasikan beberapa style yang baru saja dibuat ke dalam berkas activity\_main.xml dan kondisikan seperti ini :

|  |
| --- |
| 1. <?xml version="1.0" encoding="utf-8"?> 2. <ScrollView xmlns:android="http://schemas.android.com/apk/res/android" 3. android:id="@+id/activity\_main" 4. android:layout\_width="match\_parent" 5. android:layout\_height="match\_parent" 6. android:background="@android:color/white"> 8. <LinearLayout 9. android:layout\_width="match\_parent" 10. android:layout\_height="wrap\_content" 11. android:orientation="vertical"> 13. <FrameLayout 14. android:layout\_width="match\_parent" 15. android:layout\_height="wrap\_content"> 17. <ImageView 18. android:layout\_width="match\_parent" 19. android:layout\_height="wrap\_content" 20. android:adjustViewBounds="true" 21. android:scaleType="fitXY" 22. android:src="@drawable/pixel\_google" /> 24. <TextView 25. **style="@style/TextContent.Small.White"** 26. android:layout\_gravity="bottom" 27. android:layout\_marginStart="16dp" 28. android:layout\_marginBottom="@dimen/activity\_vertical\_margin" 29. android:background="#4D000000" 30. android:drawableStart="@drawable/ic\_collections\_white\_18dp" 31. android:drawablePadding="4dp" 32. android:gravity="center\_vertical" 33. android:padding="8dp" 34. android:text="@string/dummy\_photos" 35. android:textColor="@android:color/white" /> 36. </FrameLayout> 38. <TextView 39. **style="@style/TextContent.Black"** 40. android:layout\_marginTop="@dimen/activity\_vertical\_margin" 41. android:layout\_marginBottom="8dp" 42. android:text="@string/dummy\_value" 43. android:textSize="32sp" /> 45. <TextView 46. **style="@style/TextContent.Subtitle"** 47. android:layout\_marginBottom="@dimen/activity\_vertical\_margin" 48. android:text="@string/stock" /> 50. <TextView 51. **style="@style/TextContent"** 52. android:layout\_marginBottom="@dimen/activity\_vertical\_margin" 53. android:lineSpacingMultiplier="1" 54. android:text="@string/content\_text" 55. android:textColor="@android:color/black" /> 57. <TextView 58. **style="@style/TextContent.Subtitle"** 59. android:layout\_marginBottom="8dp" 60. android:text="@string/specification" /> 62. <TableLayout 63. android:layout\_width="match\_parent" 64. android:layout\_height="wrap\_content" 65. android:layout\_marginLeft="@dimen/activity\_horizontal\_margin" 66. android:layout\_marginRight="@dimen/activity\_horizontal\_margin" 67. android:layout\_marginBottom="@dimen/activity\_vertical\_margin"> 69. <TableRow 70. android:layout\_width="match\_parent" 71. android:layout\_height="wrap\_content" 72. android:layout\_marginBottom="8dp"> 74. <TextView 75. **style="@style/TextSpec.Field"** 76. android:text="@string/display" /> 78. <TextView 79. **style="@style/TextSpec.Value"** 80. android:text="@string/content\_specs\_display" /> 81. </TableRow> 83. <TableRow 84. android:layout\_width="match\_parent" 85. android:layout\_height="wrap\_content" 86. android:layout\_marginBottom="8dp"> 88. <TextView 89. **style="@style/TextSpec.Field"** 90. android:text="@string/size" /> 92. <TextView 93. **style="@style/TextSpec.Value"** 94. android:text="@string/content\_specs\_size" /> 95. </TableRow> 97. <TableRow 98. android:layout\_width="match\_parent" 99. android:layout\_height="wrap\_content" 100. android:layout\_marginBottom="8dp"> 102. <TextView 103. **style="@style/TextSpec.Field"** 104. android:text="@string/battery" /> 106. <TextView 107. **style="@style/TextSpec.Value"** 108. android:text="@string/content\_specs\_battery" /> 109. </TableRow> 110. </TableLayout> 112. <TextView 113. **style="@style/TextContent.Subtitle"** 114. android:layout\_marginBottom="8dp" 115. android:text="@string/seller" /> 117. <RelativeLayout 118. android:layout\_width="match\_parent" 119. android:layout\_height="wrap\_content" 120. android:layout\_marginLeft="@dimen/activity\_horizontal\_margin" 121. android:layout\_marginRight="@dimen/activity\_horizontal\_margin" 122. android:layout\_marginBottom="@dimen/activity\_vertical\_margin"> 124. <de.hdodenhof.circleimageview.CircleImageView 125. android:id="@+id/profile\_image" 126. android:layout\_width="56dp" 127. android:layout\_height="56dp" 128. android:layout\_centerVertical="true" 129. android:layout\_marginEnd="16dp" 130. android:layout\_marginRight="@dimen/activity\_horizontal\_margin" 131. android:src="@drawable/photo\_2" /> 133. <TextView 134. **style="@style/TextContent"** 135. android:layout\_toEndOf="@+id/profile\_image" 136. android:layout\_toRightOf="@id/profile\_image" 137. android:text="@string/my\_name" 138. android:textColor="@android:color/black" /> 139. </RelativeLayout> 141. <Button 142. android:layout\_width="match\_parent" 143. android:layout\_height="wrap\_content" 144. android:layout\_marginLeft="@dimen/activity\_horizontal\_margin" 145. android:layout\_marginRight="@dimen/activity\_horizontal\_margin" 146. android:layout\_marginBottom="@dimen/activity\_vertical\_margin" 147. android:text="@string/buy" /> 148. </LinearLayout> 149. </ScrollView> |

1. Jika selesai coba jalankan aplikasi Anda. Seharusnya hasilnya seperti ini :



Perhatikan bagaimana penulisan di setiap obyek TextView menjadi lebih sederhana :

|  |
| --- |
| 1. <TextView 2. **style="@style/TextContent.Subtitle"** 3. android:layout\_marginBottom="8dp" 4. android:text="@string/seller" /> |

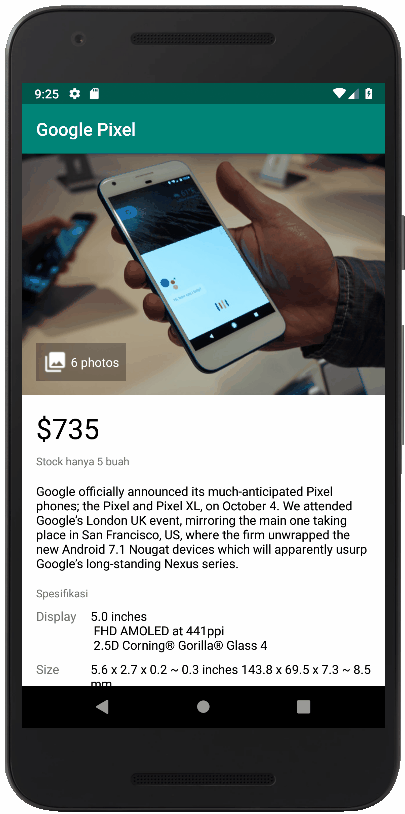
1. Masih ada yang perlu kita perbaiki. Tombol masih berada dalam kondisi default.  
   Yuk coba ubah agar lebih menyatu dengan tema aplikasi. Buka kembali styles.xml dan tambahkan kode berikut di atas tag </resources>.

|  |
| --- |
| 1. **<style name="ButtonGeneral" parent="@style/Widget.AppCompat.Button.Colored">** 2. **<item name="android:layout\_width">match\_parent</item>** 3. **<item name="android:layout\_height">wrap\_content</item>** 4. **<item name="android:layout\_marginRight">@dimen/activity\_horizontal\_margin</item>** 5. **<item name="android:layout\_marginLeft">@dimen/activity\_horizontal\_margin</item>** 6. **<item name="android:layout\_marginBottom">@dimen/activity\_vertical\_margin</item>** 7. **</style>** |

Pada activity\_main.xml kondisikan bagian obyek Button menjadi seperti ini:

|  |
| --- |
| 1. <Button 2. android:text="@string/buy" 3. **style="@style/ButtonGeneral"**/> |

Jalankan aplikasi Anda. Seharusnya tampilannya akan menjadi seperti ini :



Sekarang coba Anda klik dan tahan tombolnya. Bila perangkat Android Anda berada di bawah Lollipop (API  level 21), seharusnya tombol akan berwarna lebih gelap. Sedangkan jika di atas Lollipop, akan ada efek air (*ripple*) ketika tombol Anda tekan.

Sistem akan secara otomatis mengambil nilai accentColor sebagai nilai warna indikator aktif untuk Button dan nilai pada atribut colorButtonNormal sebagai nilai gelap ketika obyek Button ditekan.

### Bedah Kode

#### **Style**

Selesai! Saat ini Anda sudah memahami bagaimana Theme dan Style diterapkan dalam membangun sebuah aplikasi di Android. Pengelompokan atribut untuk view yang sama akan lebih memudahkan Anda dalam menentukan dan menyeragamkan format view yang akan ditampilkan ke dalam layar.

Contohnya lihat kode berikut :

|  |
| --- |
| 1. <TextView 2. **style="@style/TextContent.Subtitle"** 3. android:layout\_marginBottom="8dp" 4. android:text="@string/seller" /> |

Anda bisa menyusun View menjadi lebih rapi dan cepat, ketika Anda memiliki komponen yang sama. Anda bisa memanggil style tersebut setiap halaman yang Anda butuhkan.

Anda dapat mendalami lebih lanjut materi modul ini dengan membaca tautan berikut: [Ui Theme](https://developer.android.com/guide/topics/ui/themes.html) dan [Material Design](https://material.google.com)

Selamat!

Satu langkah besar sudah anda lalui, sekarang ayo kita lanjut ke materi yang lebih seru!