



UNIVERSITAS
BUDI LUHUR

Drawables, Style, and Themes

Presented By Putri Hayati, S.ST, M.Kom

[Read More](#)



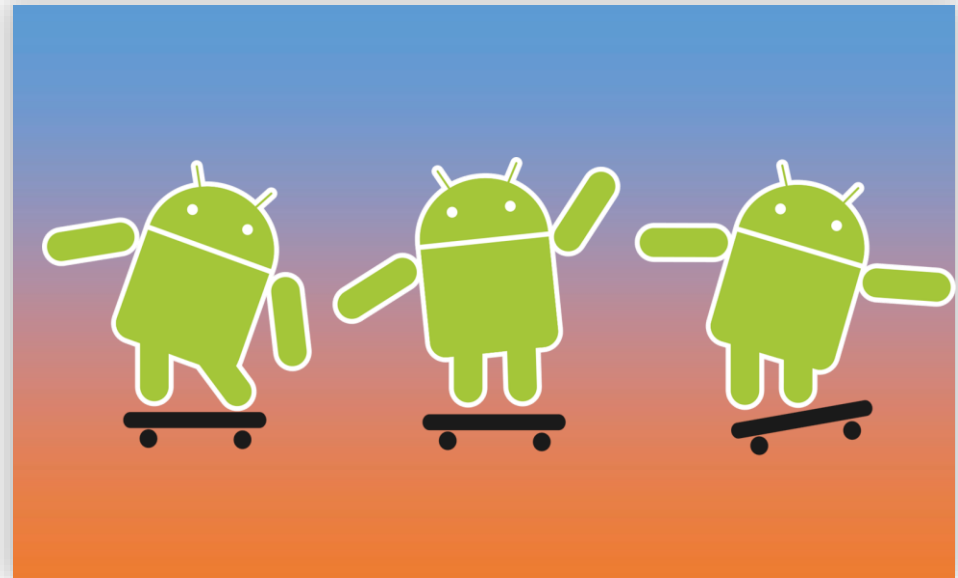
Kategori

1. Drawables

2.Style 5.2

Themes

5.3 Praktikum





Drawables



Drawables

Sumber daya dapat digambar adalah grafik yang bisa digambar ke layar. Anda mengambil sumber daya dapat digambar menggunakan API seperti `getDrawable(int)` , dan menerapkan sumber daya dapat digambar ke sumber daya XML dengan menggunakan atribut seperti `android:drawable` dan `android:icon`.

Drawable classes :

- File gambar
- File nine-patch
- Daftar layer
- Sumber daya dapat digambar untuk bentuk
- Daftar keadaan
- Daftar level
- Sumber daya dapat digambar untuk transisi
- Sumber daya dapat digambar untuk vektor

Drawables - classes

File gambar

File gambar merupakan file bitmap generik. Android mendukung file gambar dalam sejumlah format: WebP (diutamakan), PNG (diutamakan), dan JPG (diterima). Format GIF dan BMP didukung, namun tidak disarankan.

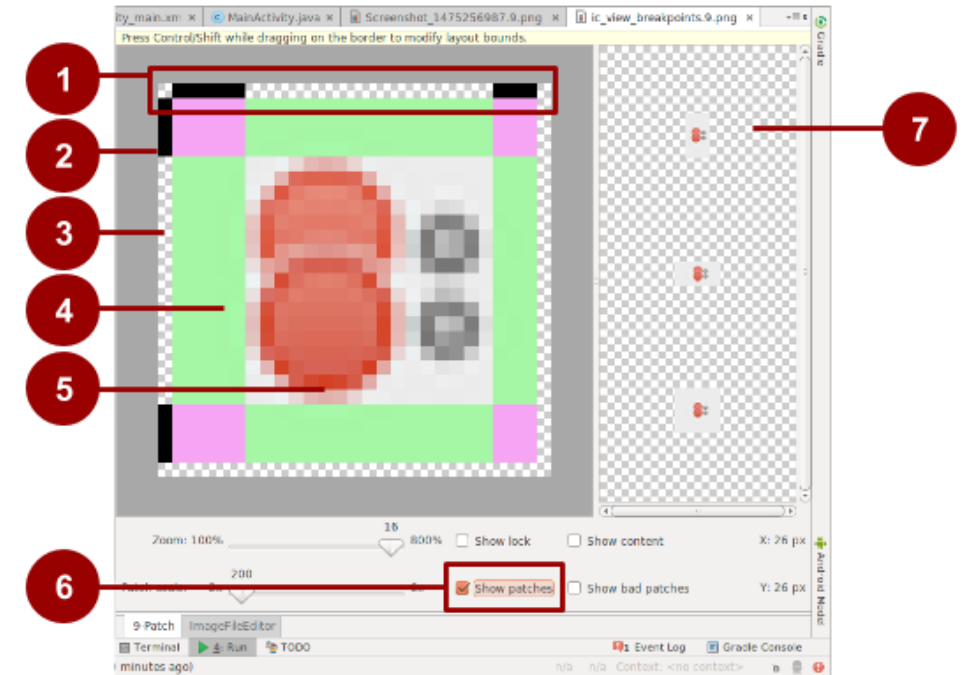
File nine-patch

9-patch adalah gambar PNG yang Anda gunakan untuk mendefinisikan region yang dapat direntang. Gunakan 9-patch sebagai gambar latar belakang bagi Tampilan untuk memastikan Tampilan terlihat pas untuk orientasi dan ukuran layar yang berbeda.

Misalnya, dalam Tampilan yang memiliki `layout_width` disetel ke `"wrap_content"`, Tampilan akan tetap cukup besar untuk menampung isinya (plus pengisi).

Drawables - classes

1. Border untuk menunjukkan region yang siap direntang melebar (secara horizontal).
2. Misalnya, dalam Tampilan yang lebih lebar daripada gambar, setrip hijau di sisi kiri dan kanan 9-patch ini bisa direntang untuk mengisi Tampilan. Tempat yang bisa direntang ditandai dengan warna hitam. Klik untuk menghitamkan piksel.
3. Border untuk menunjukkan region yang siap direntang memanjang (secara vertikal). Misalnya, dalam Tampilan yang lebih tinggi daripada gambar, setrip hijau di bagian atas dan bawah 9-patch ini bisa direntang untuk mengisi Tampilan.
4. Nonaktifkan piksel dengan mengeklik tombol shift (klik ctrl pada Mac).
5. Area yang dapat direntang.
6. Tidak dapat direntang.
7. Centang **Show patches** untuk pratinjau jalur yang dapat direntang dalam area menggambar.



File nine-patch

Drawables - classes

Daftar Layer

Di Android, Anda bisa membangun gambar dengan melapiskan gambar lain, seperti yang bisa dilakukan di **GIMP** dan program manipulasi gambar lainnya. Setiap layer dinyatakan oleh sumber daya dapat digambar individual. Sumber daya dapat digambar yang membentuk gambar tunggal diatur dan dikelola dalam suatu elemen **<layer-list>** dalam XML. Dalam **<layer-list>**, setiap sumber daya dapat digambar dinyatakan melalui elemen **<item>**.

Sumber daya dapat digambar untuk bentuk

Sumber daya dapat digambar untuk bentuk adalah persegi panjang, oval, garis, atau cincin yang Anda definisikan dalam XML. Anda bisa menetapkan ukuran dan gaya bentuk dengan menggunakan atribut XML.

Misalnya, file XML ini membuat persegi panjang dengan sudut tumpul dan gradien warna. Warna pengisi persegi berubah dari **putih (#000000)** di sudut kiri bawah menjadi **biru (#0000dd)** di sudut kanan atas. Atribut **angle** menentukan cara gradien dimiringkan:



Drawables - classes

Daftar Keadaan



StateListDrawable adalah objek sumber daya dapat digambar yang menggunakan gambar berbeda untuk menyatakan objek yang sama, bergantung pada keadaan objek tersebut berada. Misalnya, sebuah widget Button bisa ada dalam salah satu dari sejumlah keadaan (ditekan, difokuskan, diarahkan ke atas, atau tidak satu pun dari keadaan ini).

Daftar Level



Sumber daya dapat digambar untuk daftar level mendefinisikan alternatif sumber daya dapat digambar, setiap mendapat nilai numerik maksimum. Untuk memilih sumber daya dapat digambar yang akan digunakan, panggil metode `setLevel()`, yang meneruskan integer yang cocok dengan integer tingkat maksimum yang didefinisikan dalam XML. Sumber daya dengan tingkat maksimum terendah lebih besar dari atau sama dengan integer yang diteruskan ke dalam `setLevel()` akan dipilih.



Drawables - classes

Transisi

TransitionDrawable adalah sumber daya dapat digambar yang memudar bersilang di antara dua sumber daya dapat digambar. Untuk mendefinisikan sumber daya dapat digambar untuk transisi dalam XML, gunakan elemen `<transition>`. Setiap sumber daya dapat digambar dinyatakan melalui elemen `<item>` di dalam elemen `<transition>`. Tidak lebih dari dua elemen `<item>` yang didukung.

Vektor

Di Android 5.0 (API level 21) dan di atasnya, Anda bisa mendefinisikan sumber daya dapat digambar untuk vektor, yakni gambar yang didefinisikan oleh suatu jalur. Sumber daya dapat digambar untuk vektor menskalakan tanpa kehilangan definisi. Sebagian besar sumber daya dapat digambar untuk vektor menggunakan file SVG, yaitu file teks biasa atau file biner kompresi yang menyertakan koordinat dua dimensi sebagai cara menggambar pada layar.



Style



Style

- Di Android, *gaya/Style* adalah kumpulan atribut yang mendefinisikan tampilan dan format Tampilan. Anda bisa menerapkan gaya yang sama ke sejumlah Tampilan dalam aplikasi; misalnya, sejumlah **TextView** mungkin memiliki ukuran teks dan layout yang sama.
- Penggunaan gaya memungkinkan Anda menyimpan atribut umum ini di satu lokasi dan menerapkannya ke setiap **TextView** dengan menggunakan satu baris kode dalam XML.
- Anda bisa mendefinisikan gaya sendiri atau menggunakan salah satu dari [gaya platform](#) yang disediakan Android.

Mendefinisikan dan menerapkan gaya

- Untuk membuat gaya, tambahkan elemen **<style>** di dalam elemen **<resources>** dalam file XML yang terletak di folder **res/values/**. Bila membuat proyek di Android Studio, file **res/values/styles.xml** akan dibuatkan untuk Anda.

Style

Elemen **<style>** menyertakan yang berikut ini:

- Atribut name . Gunakan nama gaya bila Anda menerapkan gaya ke Tampilan.
- Atribut parent opsional. Anda akan mempelajari cara menggunakan atribut parent di bagian Pewarisan di bawah ini.
- Sejumlah elemen **<item>** sebagai elemen anak **<style>**. Setiap **<item>** menyertakan satu atribut gaya.

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="#00FF00"
    android:typeface="monospace"
    android:text="@string/hello" />
```

```
<TextView
    style="@style/CodeFont"
    android:text="@string/hello"
/>
```

Style

Define a parent style...

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <style name="CodeFont">
    <item name="android:layout_width">match_parent</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:textColor">#00FF00</item>
    <item name="android:typeface">monospace</item>
  </style>
</resources>
```



Themes



Themes

Anda bisa membuat tema dengan cara yang sama dengan membuat gaya, yaitu dengan menambahkan elemen **<style>** di dalam elemen **<resources>** dalam file XML yang terletak di folder **res/values/**.

Apakah perbedaan antara gaya dan tema?

- Gaya diterapkan pada Tampilan. Di XML, Anda menerapkan gaya menggunakan atribut style .
- Tema diterapkan ke semua Aktivitas atau aplikasi, bukan ke Tampilan individual. Di XML, Anda bisa menerapkan sebuah tema menggunakan atribut android:theme .

Gaya apa pun bisa digunakan sebagai tema. Misalnya, Anda bisa menerapkan gaya **CodeFont** sebagai tema Aktivitas, dan semua teks di dalam Aktivitas akan menggunakan font spasi tunggal abu-abu.

Themes

Untuk menerapkan tema ke aplikasi Anda, deklarasikan tema dalam elemen **<application>** di dalam file AndroidManifest.xml. Contoh ini menerapkan tema AppTheme ke seluruh aplikasi :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
  xmlns:android="http://schemas.android.com/apk/re
s/android"
  package="com.exampledomain.myapp">
  <application
    ...
    android:theme="@style/AppTheme">
  </application>
  ...
```

Themes

Bila membuat proyek baru di Android Studio, tema default akan didefinisikan untuk Anda dalam file style.xml. Misalnya, kode ini mungkin ada dalam file styles.xml :

```
<style name="AppTheme"
parent="Theme.AppCompat.Light.DarkActionBar">
    <!-- Customize your theme here. -->
    <item
name="colorPrimary">@color/colorPrimary</item>
    <item
name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
</style>
```

Themes

Platform Android menyediakan kumpulan gaya dan tema yang bisa Anda gunakan dalam aplikasi. Untuk menemukan daftar semua tema, Anda perlu melihat di dua tempat :

- Kelas R.style mencantumkan sebagian besar gaya dan tema platform yang tersedia.
- Kelas support.v7.appcompat.R.style mencantumkan lebih banyak lagi.
- Gaya dan tema ini memiliki "AppCompat" dalam namanya, dan didukung oleh pustaka v7 appcompat.
- Nama gaya dan tema menyertakan setrip bawah. Untuk menggunakannya dalam kode Anda, ganti setrip bawah dengan titik. Misalnya, inilah cara menerapkan tema Theme_NoTitleBar ke aktivitas:

```
<activity android:theme="@android:style/Theme.NoTitleBar"
```

Dan inilah cara menerapkan gaya **AlertDialog_AppCompat** ke Tampilan :

```
<TextView  
    style="@style/AlertDialog.AppCompat"  
    android:text="@string/code_string" />
```



Praktikum



Membuat Aplikasi Scorekeeper

1. Pertama - tama silahkan kalian buat sebuah project baru di **Android Studio** atau IDE
2. Project name : **Nim_Aplikasi Scorekeeper**
3. Pilih **“Empty Activity”**
4. **Jika sudah berhasil lalu di screenrecord dikirim ke Group Telegram, dengan subject :
Nim_Nama_Aplikasi Scorekeeper**
5. Dikirim ke group telegram paling lambat tanggal **11 Nov 2022 –Pukul 23.59 WIB**

Membuat Aplikasi Scorekeeper

Dalam bagian ini, Anda akan membuat proyek Android Studio, memodifikasi layout, dan menambahkan fungsionalitas onClick pada tombolnya.

1.1. Buat layout untuk aktivitas utama

Definisikan tampilan root:

1. Buka file layout untuk aktivitas utama.
2. Hapus **TextView** yang berisi **"Hello World."**
3. Ubah tampilan root menjadi **LinearLayout** dan tambahkan atribut berikut (tanpa menghapus atribut yang sudah ada):

Atribut	Nilai
android:orientation	"vertical"

Membuat Aplikasi Scorekeeper

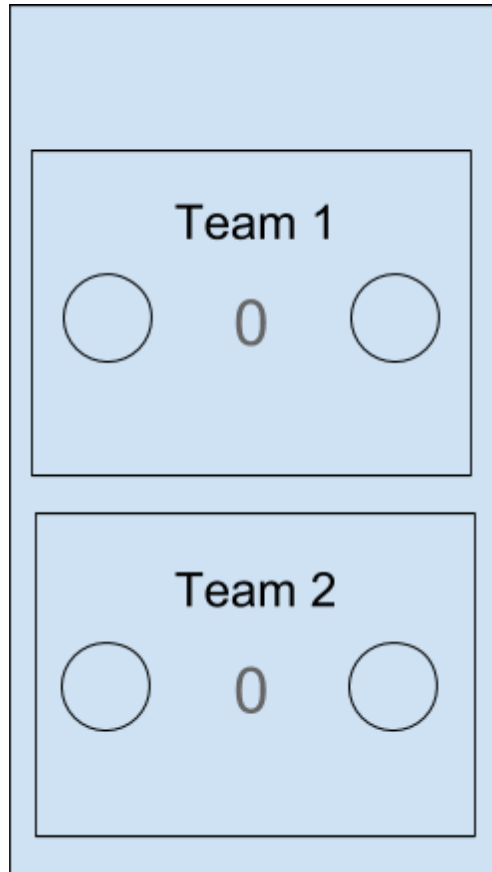
Definisikan kontainer skor

Di dalam **LinearLayout**, tambahkan dua grup tampilan **RelativeLayout** (satu untuk menampung skor untuk setiap tim) dengan atribut berikut :

Atribut	Nilai
android:layout_width	"match_parent"
android:layout_height	"0dp"
android:layout_weight	"1"

Membuat Aplikasi Scorekeeper

Tambahkan tampilan ke UI



- Tambahkan **dua** tampilan **ImageButton** (satu untuk meningkatkan skor dan satu untuk menurunkan skor) dan sebuah **TextView** untuk menampilkan skor di antara tombol ke setiap **RelativeLayout**
- Tambahkan **atribut android:id** ke **TextView** skor dan semua **ImageButton**

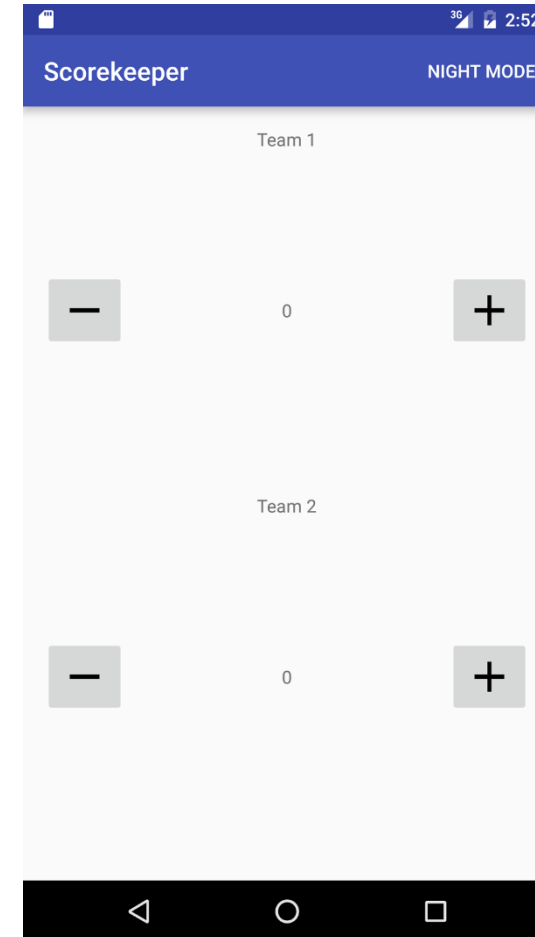
Tambahkan aset vector :

- Pilih **File > New > Vector Asset** untuk membuka **Vector Asset Studio**.
- **Klik ikon** untuk mengubahnya menjadi daftar file ikon material. Pilih **kategori Content**.
- Pilih ikon plus dan klik OK.
- Ubah nama file sumber daya **"ic_plus"** dan centang kotak **Override** di sebelah opsi ukuran.
- Ubah ukuran ikon ke **40dp x 40dp**.
- Klik Next, lalu **Finish**.
- Ulangi proses ini untuk menambahkan ikon "minus" dan beri nama file **"ic_minus"**.

Membuat Aplikasi Scorekeeper

Tambahkan atribut ke tampilan Anda

- Ubah tampilan teks skor agar terbaca "0" dan tampilan teks tim agar terbaca **"Team 1"** dan **"Team 2"**.
- Tambahkan atribut berikut ini ke tombol gambar kiri Anda:
`android:src="@drawable/ic_minus"`
`android:contentDescription="Minus Button"`
- Tambahkan atribut berikut ini ke tombol gambar kanan Anda:
`android:src="@drawable/ic_plus"`
`android:contentDescription="Plus Button"`



Membuat Aplikasi Scorekeeper

```
<?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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.android.scorekeeper.MainActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1">
```

Membuat Aplikasi Scorekeeper

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:text="@string/team_1"/>

<ImageButton
    android:id="@+id/decreaseTeam1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_centerVertical="true"
    android:contentDescription="@string/minus_button"
    android:src="@drawable/ic_minus" />
```

Membuat Aplikasi Scorekeeper

```
<TextView
    android:id="@+id/score_1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:text="@string/initial_count"/>

<ImageButton
    android:id="@+id/increaseTeam1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_centerVertical="true"
    android:contentDescription="@string/plus_button"
    android:src="@drawable/ic_plus"/>
</RelativeLayout>
```

Membuat Aplikasi Scorekeeper

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:text="@string/team_2"/>

    <ImageButton
        android:id="@+id/decreaseTeam2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        android:contentDescription="@string/minus_button"
        android:src="@drawable/ic_minus"/>
```

Membuat Aplikasi Scorekeeper

```
<TextView
    android:id="@+id/score_2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:text="@string/initial_count"/>

    <ImageButton
        android:id="@+id/increaseTeam2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_centerVertical="true"
        android:contentDescription="@string/plus_button"
        android:src="@drawable/ic_plus"/>
    </RelativeLayout>
</LinearLayout>
```


Membuat Aplikasi Scorekeeper

1.2 Inisialisasi TextView Anda dan variabel hitungan skor

Untuk mencatat skor, Anda memerlukan dua hal:

- **Variabel integer** agar Anda bisa melacak skor.
- Referensi ke **TextView** skor Anda di **MainActivity** sehingga Anda bisa memperbarui skor.
- Dalam metode **onCreate() MainActivity**, temukan **TextView** skor Anda menurut **id** dan tetapkan **TextView** ke **variabel** anggota.
- Buat dua variabel anggota **integer**, yang menunjukkan skor setiap tim, dan inisialisasi ke 0.

1.3 Implementasikan fungsionalitas **onClick** untuk tombol

- Di **MainActivity** Anda, implementasi dua metode **onClick: increaseScore()** dan **decreaseScore()**.
- Tombol kiri akan mengurangi **TextView** skor, sedangkan tombol kanan akan menambahnya.



Membuat Aplikasi Scorekeeper

```
/**
 * Method that handles the onClick of both the decrement buttons
 * @param view The button view that was clicked
 */
public void decreaseScore(View view) {
    //Get the ID of the button that was clicked
    int viewID = view.getId();
    switch (viewID){
        //If it was on Team 1
        case R.id.decreaseTeam1:
            //Decrement the score and update the TextView
            mScore1--;
            mScoreText1.setText(String.valueOf(mScore1));
            break;
        //If it was Team 2
        case R.id.decreaseTeam2:
            //Decrement the score and update the TextView
            mScore2--;
            mScoreText2.setText(String.valueOf(mScore2));
    }
}
```

Membuat Aplikasi Scorekeeper

```
/**
 * Method that handles the onClick of both the increment buttons
 * @param view The button view that was clicked
 */
public void increaseScore(View view) {
    //Get the ID of the button that was clicked
    int viewID = view.getId();
    switch (viewID){
        //If it was on Team 1
        case R.id.increaseTeam1:
            //Increment the score and update the TextView
            mScore1++;
            mScoreText1.setText(String.valueOf(mScore1));
            break;
        //If it was Team 2
        case R.id.increaseTeam2:
            //Increment the score and update the TextView
            mScore2++;
            mScoreText2.setText(String.valueOf(mScore2));
    }
}
```

Membuat Aplikasi Scorekeeper

Membuat sumber daya Drawable

- Anda sekarang sudah memiliki **aplikasi Scorekeeper** yang berfungsi! Akan tetapi, *layout-nya tidak menarik dan tidak menyampaikan fungsi tombol*. Untuk memperjelasnya, latar belakang abu-abu standar tombol bisa diubah.
- Di Android, grafik sering kali ditangani oleh sebuah sumber daya yang disebut Drawable. Di latihan berikut, Anda akan mempelajari cara membuat tipe drawable tertentu yang disebut ShapeDrawable dan menerapkannya ke tombol Anda sebagai latar belakang.

Membuat Aplikasi Scorekeeper

1.4 Buat Drawable Bentuk

ShapeDrawable adalah bentuk geometris primitif yang didefinisikan di file xml oleh sejumlah atribut, yang mencakup warna, bentuk, **padding**, dan lainnya. Ini mendefinisikan grafik vektor, yang bisa diskalakan naik dan turun tanpa kehilangan definisi.

- Klik kanan folder drawable di direktori sumber daya.
- Pilih New > Drawable resource file.
- Beri nama file "button_background" dan klik OK.
- Buang semua kode, kecuali `<?xml version="1.0" encoding="utf-8"?>`
- Tambahkan kode berikut yang membuat bentuk oval dengan garis luar:

```
<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="oval">
    <stroke
        android:width="2dp"
        android:color="@color/colorPrimary"/>
</shape>
```

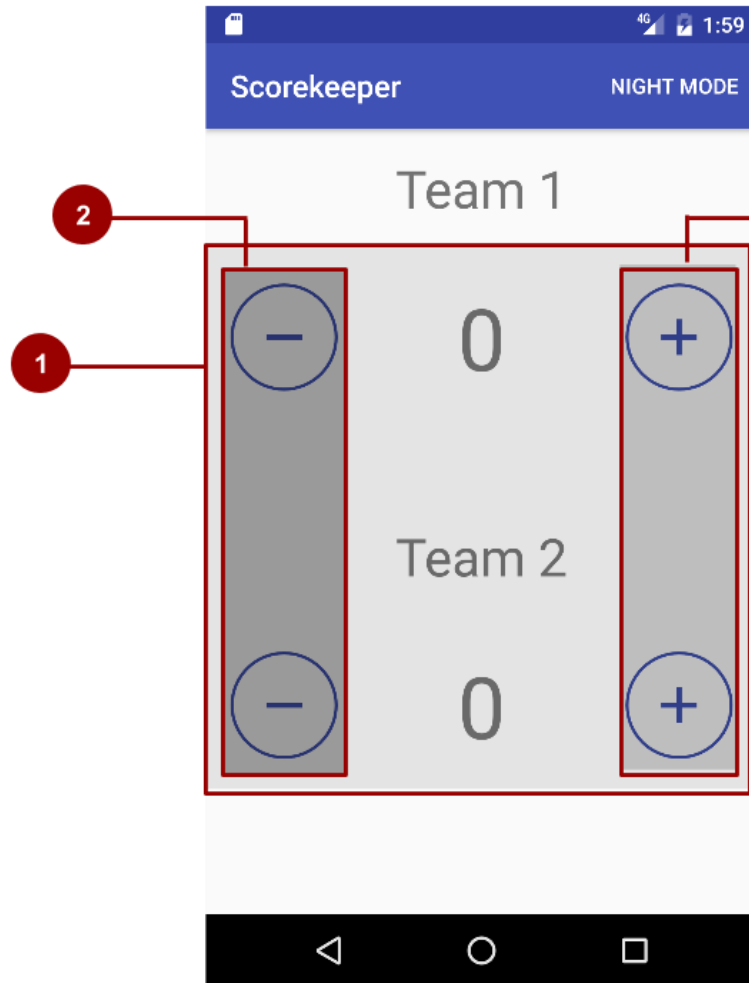
Membuat Aplikasi Scorekeeper

1.5 Terapkan drawable bentuk sebagai latar belakang

- Buka file layout untuk aktivitas utama Anda.
- Untuk semua tombol, tambah drawable sebagai latar belakang: `android:background="@drawable/button_background"`. Perhatikan bahwa latar belakang otomatis diskalakan agar pas dengan ukuran tampilan.
- Ukuran tombol besarnya harus sedemikian rupa sehingga bisa dirender dengan benar di semua perangkat. Ubah "`layout_height`" dan "`layout_width`" untuk setiap tombol menjadi **70dp**, yaitu ukuran yang baik untuk kebanyakan perangkat. Menggunakan dimensi yang di-hardcode bukanlah praktik terbaik, tetapi menggunakan ketebalan dengan layout linear yang dilapiskan untuk mendapatkan ukuran yang diinginkan akan melibatkan terlalu banyak detail untuk praktik ini.
`android:layout_width="70dp"`
`android:layout_height="70dp"`
- Ekstrak sumber daya dimensi sehingga Anda bisa mengaksesnya di satu lokasi. Untuk informasi cara melakukannya,
- Jalankan aplikasi Anda/Gradle

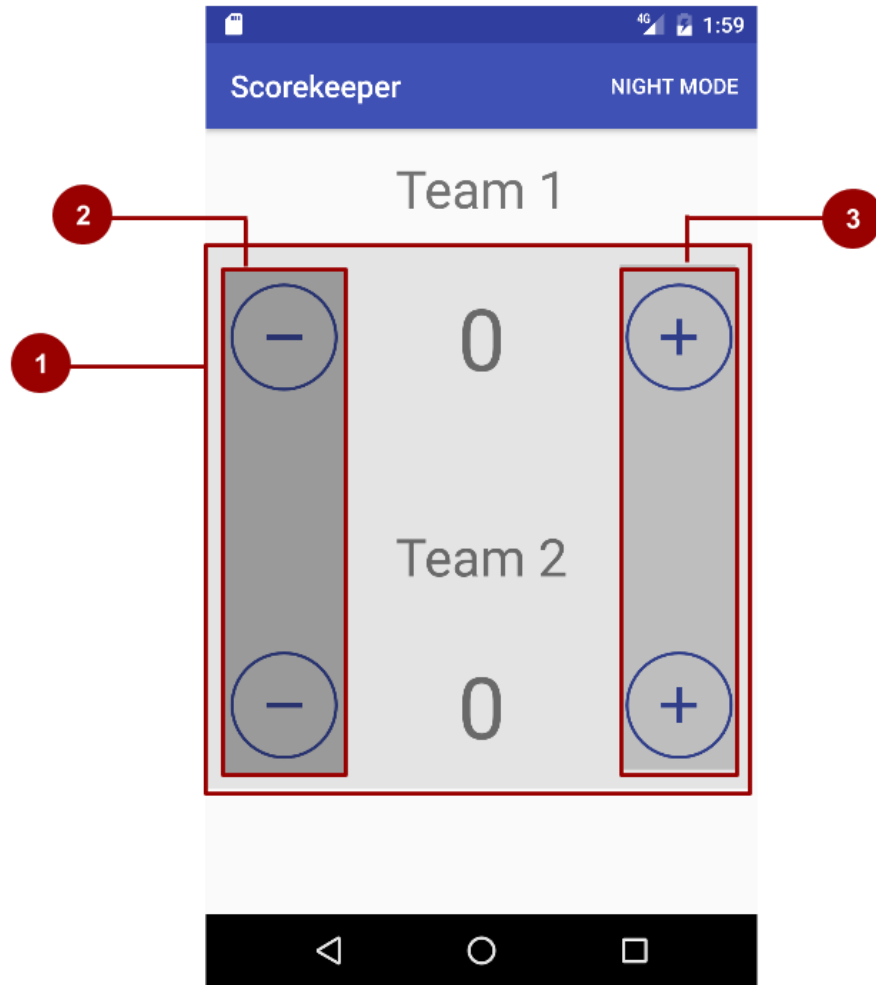
Membuat Aplikasi Scorekeeper

1.6 Buat gaya tombol



- Gaya tombol skor untuk semua tombol, yang menyertakan properti default tampilan ImageButton dan juga latar belakang drawable.
- Gaya tombol minus untuk tombol kurang, yang mewarisi atribut gaya sebelumnya dan juga menyertakan ikon minus.
- Gaya tombol plus untuk tombol tambah, juga mewarisi gaya tombol skor dan juga menyertakan ikon plus.

Membuat Aplikasi Scorekeeper



Lakukan yang berikut :

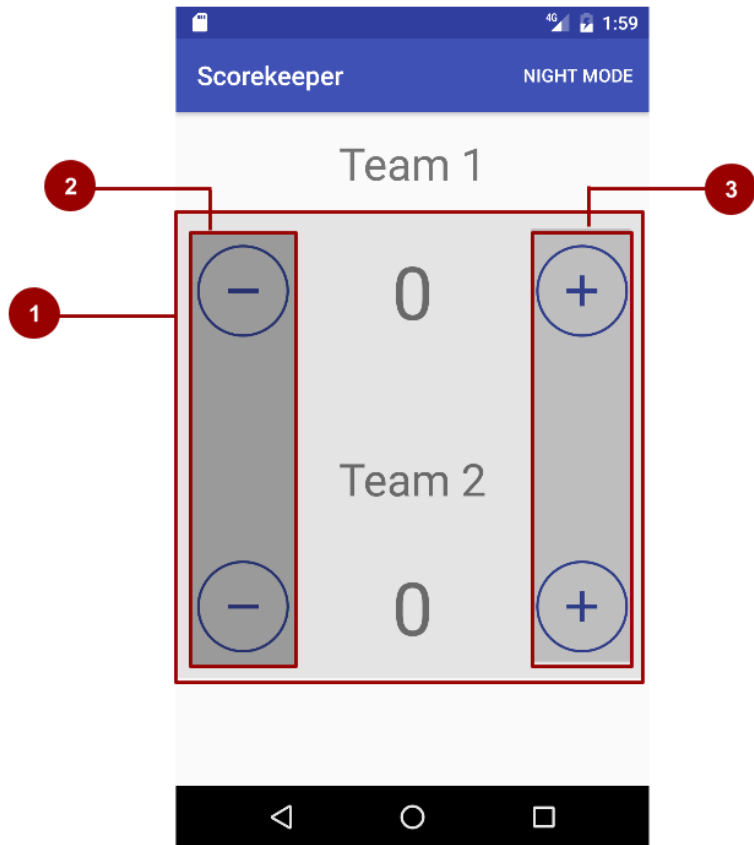
- Dalam direktori sumber daya, cari dan buka file "values/styles.xml". Di file inilah semua kode gaya Anda ditempatkan. Gaya "AppTheme" selalu ditambahkan secara otomatis dan Anda bisa melihat bahwa gaya ini diluaskan dari "Theme.AppCompat.Light.DarkActionBar".

```
<style name="AppTheme"
    parent="Theme.AppCompat.Light.DarkAction
    Bar">
```

- Untuk mempertahankan atribut default tombol. Tambahkan atribut yang mengubah latar belakang drawable menjadi latar belakang seperti :

```
<style name="ScoreButtons"
    parent="Widget.AppCompat.Button">
    <item
        name="android:background">@drawable/button_
        background</item>
</style>
```

Membuat Aplikasi Scorekeeper



Buat gaya untuk tombol plus dengan melluaskan gaya **"ScoreButtons"**:

```
<style name="PlusButtons" parent="ScoreButtons">
<item name="android:src">@drawable/ic_plus</item>
<item
name="android:contentDescription">@string/plus_button</item>
</style>
```

Buat gaya untuk tombol minus :

```
<style name="MinusButtons" parent="ScoreButtons">
<item name="android:src">@drawable/ic_minus</item>
<item name="android:contentDescription">@string/minus_button</item>
</style>
```

Di file layout untuk aktivitas utama, buang semua atribut yang Anda definisikan di gaya untuk setiap tombol dan ganti dengan gaya yang sesuai :

```
style="@style/MinusButtons"
style="@style/PlusButtons"
```

Membuat Aplikasi Scorekeeper

1.7 Buat gaya **TextView**

Tampilan teks untuk menampilkan nama tim dan skor juga bisa ditata gayanya karena warna dan font-nya sama. Lakukan yang berikut :

- Tambahkan atribut berikut ke semua **TextView**:
`android:textAppearance="@style/TextAppearance.AppCompat.Headline"`
- Klik kanan di mana saja di atribut **TextView** skor pertama dan pilih **Refactor > Extract > Style...**
- Beri nama gaya "**ScoreText**" dan centang kotak `textAppearance` (atribut yang baru Anda tambahkan) serta Launch '**Use Styles Where Possible**' **refactoring after the style is extracted** (menggunakan kotak centang). Ini akan memindai file layout untuk tampilan dengan atribut yang sama dan menerapkan gaya untuk Anda. Jangan ekstrak atribut yang terkait dengan layout.
- Pilih OK.
- Pastikan scope disetel ke file `layout activity_main.xml` dan klik OK.
- Sebuah panel di bagian bawah Android Studio akan terbuka jika gaya yang sama ditemukan di tampilan lain.
- Pilih **Do Refactor** untuk menerapkan gaya baru ke tampilan dengan atribut yang sama.
- Jalankan aplikasi Anda.

Membuat Aplikasi Scorekeeper

styles.xml

```
<resources>
  <!-- Base application theme. -->
  <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
  </style>

  <style name="ScoreButtons" parent="AppTheme">
    <item name="android:background">@drawable/button_background</item>
  </style>
```

Membuat Aplikasi Scorekeeper

styles.xml

```
<style name="PlusButtons" parent="ScoreButtons">
    <item name="android:src">@drawable/ic_plus</item>
    <item name="android:contentDescription">@string/plus_button</item>
</style>

<style name="MinusButtons" parent="ScoreButtons">
    <item name="android:src">@drawable/ic_minus</item>
    <item name="android:contentDescription">@string/minus_button</item>
</style>

<style name="ScoreText">
    <item name="android:textAppearance">@style/TextAppearance.AppCompat.Headline</item>
</style>
</resources>
```

Membuat Aplikasi Scorekeeper

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="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:weightSum="2"
    tools:context="com.example.android.scorekeeper.MainActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1">
```

Membuat Aplikasi Scorekeeper

activity_main.xml

```
<TextView
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:text="@string/team_1"
    style="@style/ScoreText" />

<ImageButton
    android:id="@+id/decreaseTeam1"
    android:layout_height="@dimen/button_size"
    android:layout_width="@dimen/button_size"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_centerVertical="true"
    android:onClick="decreaseScore"
    style="@style/MinusButtons"/>
```

Membuat Aplikasi Scorekeeper

activity_main.xml

```
<TextView
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true"
    android:id="@+id/score_1"
    android:text="@string/initial_count"
    style="@style/ScoreText" />

<ImageButton
    android:id="@+id/increaseTeam1"
    android:layout_height="@dimen/button_size"
    android:layout_width="@dimen/button_size"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_centerVertical="true"
    android:onClick="increaseScore"
    style="@style/PlusButtons"/>

</RelativeLayout>
```


Membuat Aplikasi Scorekeeper

activity_main.xml

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1">

    <TextView
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:text="@string/team_2"
        style="@style/ScoreText" />
```

Membuat Aplikasi Scorekeeper

activity_main.xml

```
<ImageButton
    android:id="@+id/decreaseTeam2"
    android:layout_height="@dimen/button_size"
    android:layout_width="@dimen/button_size"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_centerVertical="true"
    android:onClick="decreaseScore"
    style="@style/MinusButtons"/>

<TextView
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true"
    android:id="@+id/score_2"
    android:text="@string/initial_count"
    style="@style/ScoreText" />
```

Membuat Aplikasi Scorekeeper

activity_main.xml

```
<ImageButton
    android:id="@+id/increaseTeam2"
    android:layout_height="@dimen/button_size"
    android:layout_width="@dimen/button_size"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_centerVertical="true"
    android:onClick="increaseScore"
    style="@style/PlusButtons"/>
</RelativeLayout>
</LinearLayout>
```

Membuat Aplikasi Scorekeeper

1.8 Tema dan Sentuhan Akhir

Jelajahi tema

- Di file manifest Android, temukan atribut `<application>` tag dan ubah `android:theme` ke:
`android:theme="@style/Theme.AppCompat.Light.NoActionBar"`
- Jalankan aplikasi Anda.
- Ubah tema aplikasi kembali ke AppTheme yang merupakan anak tema `Theme.Appcompat.Light.DarkActionBar` seperti yang bisa dilihat di `styles.xml`.

Tambahkan tombol tema ke menu

- Klik kanan direktori `"res"` dan pilih `New > Android resource file`.
- Beri nama file `"main_menu"`, ubah `Resource Type` ke `Menu`, dan klik OK.

Tambahkan item menu dengan atribut berikut:

```
<item  
    android:id="@+id/night_mode"  
    android:title="@string/night_mode"/>
```

Membuat Aplikasi Scorekeeper

1.8 Tema dan Sentuhan Akhir

Buka "strings.xml" dan buat dua sumber daya string:

```
<string name="night_mode">Night Mode</string>
```

```
<string name="day_mode">Day Mode</string>
```

- Tekan **Ctrl - O** untuk membuka menu Override Method di file Java aktivitas utama Anda dan pilih metode **onCreateOptionsMenu** yang terletak di bawah kategori "**android.app.Activity**". Klik OK.
- Masukkan menu yang baru Anda buat di dalam metode **onCreateOptionsMenu()** :

```
getMenuInflater().inflate(R.menu.main_menu, menu);
```

Membuat Aplikasi Scorekeeper

1.9 Ubah Tema dari Menu

- Tema DayNight menggunakan kelas `AppCompatActivity` untuk menyetel opsi mode malam di aktivitas Anda. Untuk mengetahui selengkapnya tentang tema ini,
- Di file `styles.xml`, ubah induk `AppTheme` menjadi `Theme.AppCompat.DayNight.DarkActionBar`.
- Ganti metode `onOptionsItemSelected()` di `MainActivity` dan periksa item menu mana yang diklik :
`@Override`

```
public boolean onOptionsItemSelected(MenuItem item) {  
    //Check if the correct item was clicked  
    if(item.getItemId()==R.id.night_mode){  
    }  
}
```
- Sebagai respons terhadap klik pada tombol menu, Anda bisa memverifikasi setelan mode malam saat ini dengan memanggil `AppCompatActivity.getDefaultNightMode()`.

Membuat Aplikasi Scorekeeper

1.9 Ubah Tema dari Menu

- Jika mode malam diaktifkan, ubah menjadi status nonaktif:

```
//Get the night mode state of the app  
int nightMode = AppCompatActivity.getDefaultNightMode();  
//Set the theme mode for the restarted activity  
if(nightMode == AppCompatActivity.MODE_NIGHT_YES) {  
    AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_NO);  
}
```
- Jika tidak, aktifkan mode ini:

```
else {  
    AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_YES);  
}
```

Membuat Aplikasi Scorekeeper

1.9 Ubah Tema dari Menu

- Tema hanya bisa berubah saat aktivitas sedang dibuat, sehingga panggil `recreate()` agar perubahan tema bisa diterapkan.
- Metode `onOptionsItemSelected()` Anda harus mengembalikan `true`, karena klik item ditangani.
- Jalankan aplikasi Anda. Menu "Mode Malam" sekarang akan mengalihkan tema aktivitas Anda. Anda mungkin memperhatikan bahwa label untuk item menu Anda selalu tampak sebagai "Night Mode" yang mungkin membingungkan pengguna jika aplikasi Anda sudah dalam tema gelap.

Membuat Aplikasi Scorekeeper

1.9 Ubah Tema dari Menu

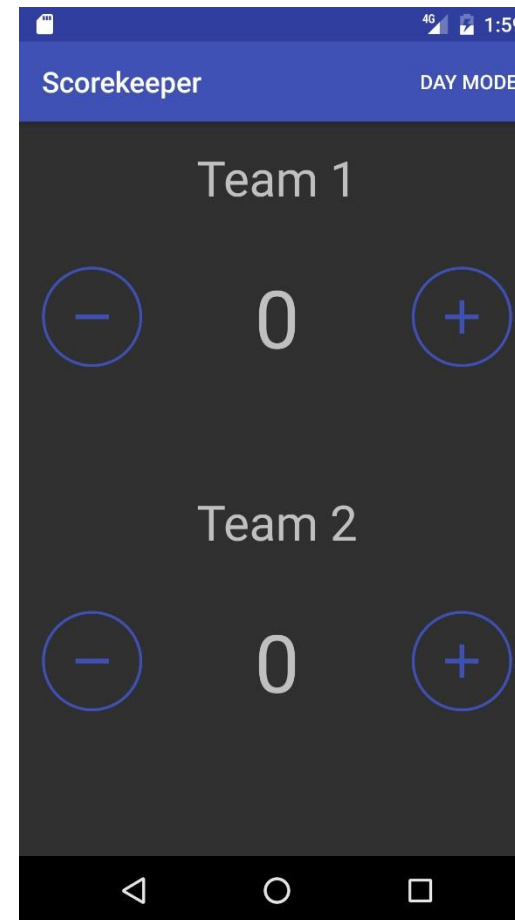
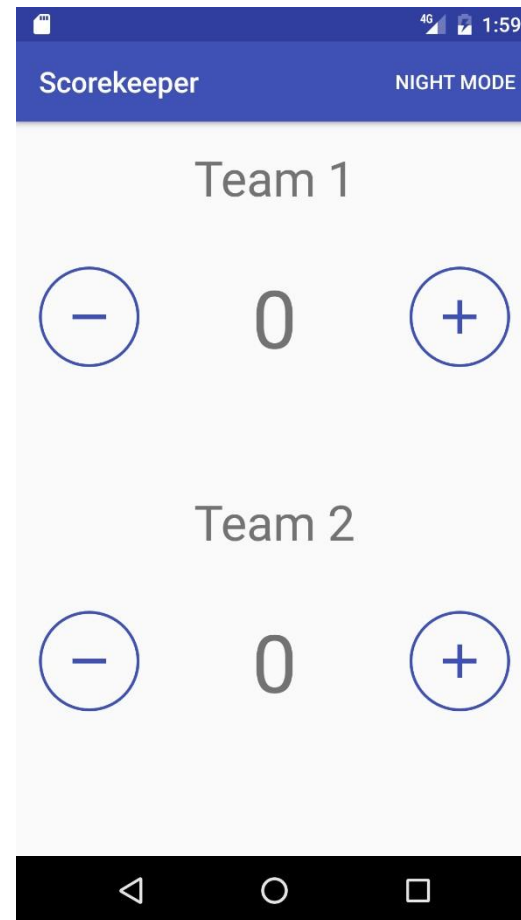
- Tambahkan kode berikut di metode `onCreateOptionsMenu`:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    //Inflate the menu from XML
    getMenuInflater().inflate(R.menu.main_menu, menu);

    //Change the label of the menu based on the state of the app
    int nightMode = AppCompatActivity.getDefaultNightMode();
    if(nightMode == AppCompatActivity.MODE_NIGHT_YES){
        menu.findItem(R.id.night_mode).setTitle(R.string.day_mode);
    }
    else{
        menu.findItem(R.id.night_mode).setTitle(R.string.night_mode);
    }
    return true;
}
```

Membuat Aplikasi Scorekeeper

Gradle



Any Question ?

