

**MODUL PRAKTIKUM - 3**  
**PENGEMBANGAN APLIKASI MOBILE**



**PROGRAM STUDI TEKNIK INFORMATIKA**  
**JURUSAN TEKNIK ELEKTRO, INFORMATIKA, DAN SISTEM FISIKA**  
**INSTITUT TEKNOLOGI SUMATERA**  
**TAHUN AKADEMIK 2019/2020**

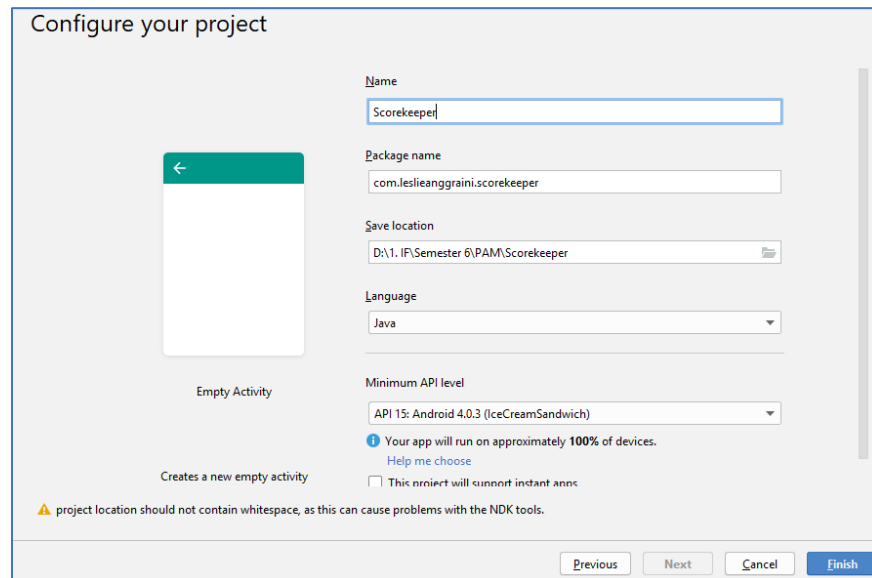
## MODUL PRAKTIKUM 3 (TUGAS 1) – PAM

### Tugas 1. Drawables, Style, and Themes

#### Task 1: Membuat Aplikasi Scorekeeper

##### A. Membuat project

- Membuat new project yang diberi nama **Scorekeeper**



##### B. Buat layout untuk aktivitas utama

- Buka **activity\_main.xml**, hapus TextView yang berisi "Hello World"
- Ubah tampilan root menjadi **LinearLayout**
- Tambahkan atribut **orientation** dan **padding** seperti berikut:



##### C. Membuat kontainer skor

- Di dalam LinearLayout, tambahkan dua **RelativeLayout** dengan atribut berikut:

```

7      android:orientation="vertical"
8      android:padding="16dp"
9      tools:context=".MainActivity">
10
11      <RelativeLayout
12          android:layout_width="match_parent"
13          android:layout_height="0dp">
14      </RelativeLayout>
15
16      <RelativeLayout
17          android:layout_width="match_parent"
18          android:layout_height="0dp">
19      </RelativeLayout>
20
21  </LinearLayout>

```

- Tambahkan dua **ImageButton** dan dua **TextView** untuk masing-masing RelativeLayout: satu untuk mengurangi skor dan satu untuk meningkatkan skor. Gunakan atribut ini (**harus berurutan sesuai gambar**):

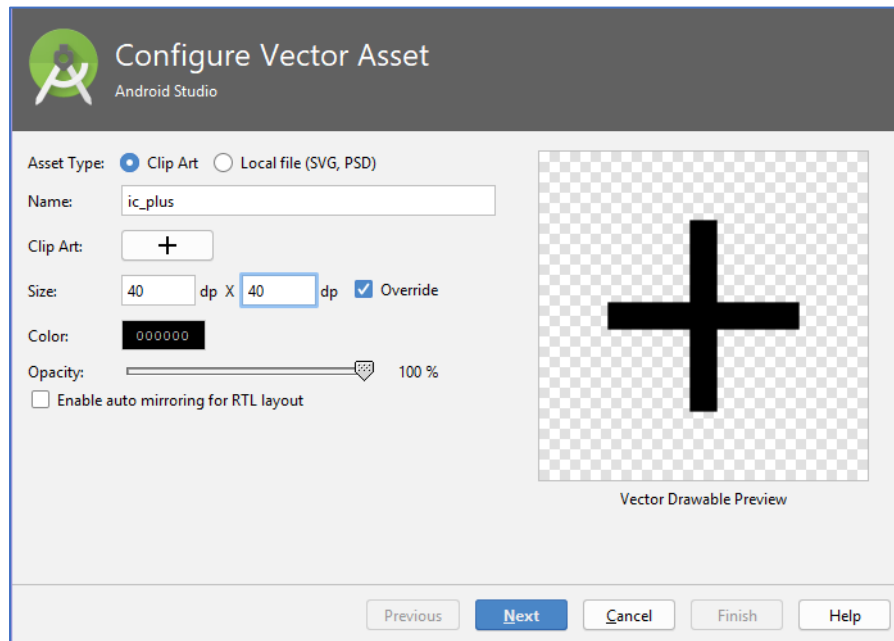
```

9      tools:context=".MainActivity">
10
11      <RelativeLayout
12          android:layout_width="match_parent"
13          android:layout_height="0dp"
14          android:layout_weight="1">
15
16          <ImageButton
17              android:layout_width="wrap_content"
18              android:layout_height="wrap_content"
19              android:id="@+id/decreaseTeam1"
20              android:layout_alignParentLeft="true"
21              android:layout_alignParentStart="true"
22              android:layout_centerVertical="true"/>
23
24          <TextView
25              android:layout_width="wrap_content"
26              android:layout_height="wrap_content"
27              android:id="@+id/score_1"
28              android:layout_centerHorizontal="true"
29              android:layout_centerVertical="true"
30              android:text="0"/>
31
32          <TextView
33              android:layout_width="wrap_content"
34              android:layout_height="wrap_content"
35              android:id="@+id/namaTeam1"
36              android:layout_alignParentTop="true"
37              android:layout_centerHorizontal="true"
38              android:text="Team 1"/>
39
40          <ImageButton
41              android:layout_width="wrap_content"
42              android:layout_height="wrap_content"
43              android:id="@+id/increaseTeam1"
44              android:layout_alignParentRight="true"
45              android:layout_alignParentEnd="true"
46              android:layout_centerVertical="true"/>
47      </RelativeLayout>

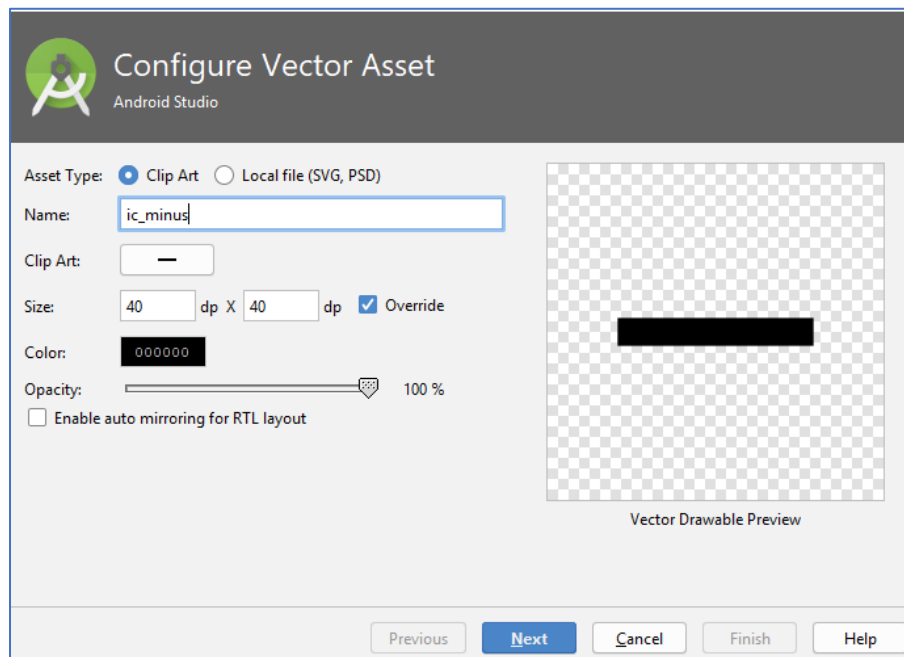
```

#### D. Tambahkan asset vector

- Pilih **File > New > Vector Asset** untuk membuka Vector Asset Studio
- Klik ikon pada **Clip Art > Pilih kategori Content > Pilih ikon plus** dan klik OK
- Ubah nama: **ic\_plus**, ukuran: **40 x 40 dp** dan centang kotak Override



- Ulangi sekali lagi, proses ini untuk menambahkan ikon "minus"
- Ubah nama: **ic\_minus**, ukuran: **40 x 40 dp** dan centang kotak Override



- Tambahkan atribut berikut ke ImageButton elemen di sisi kiri tata letak:  
`android:src="@drawable/ic_minus"`  
`android:contentDescription="Minus Button"`
- Tambahkan atribut berikut ke ImageButton elemen di sisi kanan tata letak:  
`android:src="@drawable/ic_plus"`  
`android:contentDescription="Plus Button"`
- Ekstrak semua sumber daya string Anda. Proses ini menghapus semua string Anda dari kode Java dan menempatkannya di **strings.xml**
- Hasil desain aplikasi



#### E. Menginisialisasi elemen TextView dan variable jumlah skor

- Buka **MainActivity.java** dan tambahkan *code program* seperti pada gambar dibawah ini:

```

1 package com.leslieanggraini.scorekeeper;
2
3 import ...
4
5
6
7
8 public class MainActivity extends AppCompatActivity {
9     private int mScore1, mScore2;
10    TextView mScoreText1, mScoreText2;
11
12    @Override
13    protected void onCreate(Bundle savedInstanceState) {
14        super.onCreate(savedInstanceState);
15        setContentView(R.layout.activity_main);
16
17        mScoreText1 = (TextView) findViewById(R.id.score_1);
18        mScoreText2 = (TextView) findViewById(R.id.score_2);
19    }

```

F. Implementasi fungsionalitas onClick untuk tombol anda

- Buka **activity\_main.xml**
- Tambahkan ke setiap ImageButton yang mengurangi skor dengan atribut **onClick = "decreaseScore"**
- Tambahkan ke setiap ImageButton yang menambah skor dengan atribut **onClick = "increaseScore"**
- Arahkan kursor ke nilai atribut onClick dan klik alt+enter di keyboard anda, kemudian pilih evenhandler.

```

16 <ImageButton
17     android:layout_width="wrap_content"
18     android:layout_height="wrap_content"
19     android:id="@+id/decreaseTeam1"
20     android:layout_alignParentLeft="true"
21     android:layout_alignParentStart="true"
22     android:layout_centerVertical="true"
23     android:src="@drawable/ic_minus"
24     android:contentDescription="@string/minus_button_desc"
25     android:onClick="decreaseScore"/>
26
27 <TextView
28     android:layout_width="wrap_content"
29     android:layout_height="wrap_content"
30     android:id="@+id/score_1"
31     android:layout_centerHorizontal="true"
32     android:layout_centerVertical="true"
33     android:text="@string/count"/>
34

```

- Buka **MainActivity.java**, kemudian tambahkan *code program* di increaseScore dan decreaseScore seperti pada gambar dibawah:

```

18         mScoreText2 = (TextView) findViewById(R.id.score_2);
19     }
20
21     public void decreaseScore(View view) {
22         int viewID = view.getId();
23
24         switch (viewID){
25             case R.id.decreaseTeam1:
26                 mScore1--;
27                 mScoreText1.setText(String.valueOf(mScore1));
28                 break;
29             case R.id.decreaseTeam2:
30                 mScore2--;
31                 mScoreText2.setText(String.valueOf(mScore2));
32             }
33     }
34
35     public void increaseScore(View view) {
36         int viewID = view.getId();
37
38         switch (viewID){
39             case R.id.increaseTeam1:
40                 mScore1++;
41                 mScoreText1.setText(String.valueOf(mScore1));
42                 break;
43             case R.id.increaseTeam2:
44                 mScore2++;
45                 mScoreText2.setText(String.valueOf(mScore2));
46             }
47     }
48 }

```

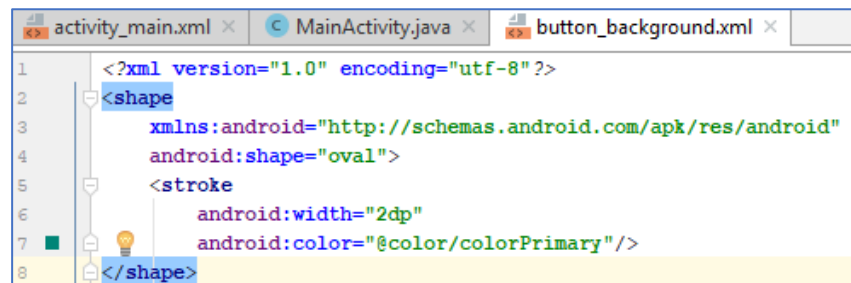
## Task 2: Membuat Drawable

### A. Membuat bentuk drawable

- Arahkan kursor ke folder drawable, kemudian klik kanan pilih **New > Drawable resource file**. Beri nama file menjadi **button\_background**
- Buang semua kode, kecuali:

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

- Tambahkan kode berikut yang membuat bentuk oval dengan garis luar:



```

1 <?xml version="1.0" encoding="utf-8"?>
2 <shape
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:shape="oval">
5     <stroke
6         android:width="2dp"
7         android:color="@color/colorPrimary"/>
8 </shape>

```

### B. Terapkan bentuk drawable sebagai latar belakang

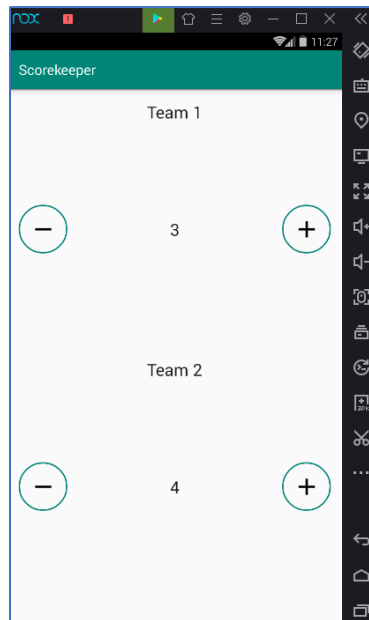
- Buka **activity\_main.xml**, pada setiap elemen ImageButton tambahkan:  
`android:background="@drawable/button_background"`
- Ubah ukuran ImageButton

`android:layout_width="70dp"`  
`android:layout_height="70dp"`

- Kemudian ekstrak sumber daya dimensi baru dari ukuran tersebut, dengan membuat resource file didalam folder values bernama **dimens.xml**, tambahkan code didalam file tersebut:

`<dimen name="button_size">70dp</dimen>`

- Jalankan aplikasi:



### Task 3: Rubah Tata Gaya Tampilan Anda

#### A. Buat gaya tombol

- Buka folder **res > values > strings.xml**, ubah atribut dari **AppTheme** seperti berikut:

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

- Kemudian di antara tag `<resources>`, tambahkan gaya baru dengan atribut seperti berikut:

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

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

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



- Pada setiap elemen ImageButton yang mengurangi nilai, hapuslah atribut: **src**, **contentDescription**, dan **background**. Tambahkan:

`android:style="@style/MinusButton"`

- Pada setiap elemen ImageButton yang menambah nilai, hapuslah atribut **src**, **contentDescription**, dan **background**. Tambahkan:

`android:style="@style/PlusButton"`

## B. Buat gaya untuk TextView

- Buka folder **res > values > strings.xml**, kemudian tambahkan atribut sebagai berikut:

```
<style name="ScoreText">
    <item name="android:textAppearance">@style/TextAppearance.AppCompat.Display3</item>
</style>

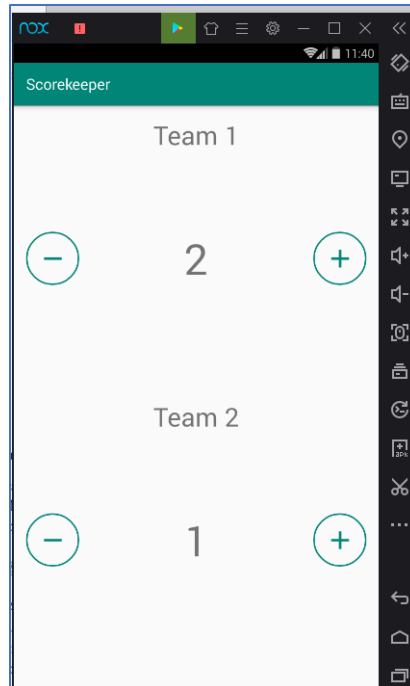
<style name="TeamText">
    <item name="android:textAppearance">@style/TextAppearance.AppCompat.Display1</item>
</style>
```

- Buka **activity\_main.xml**, kemudian tambahkan **style** untuk TextView pada masing-masing Relative Layout

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/score_1"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:text="0"
    style="@style/ScoreText" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/namaTeam1"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:text="Team 1"
    style="@style/TeamText" />
```

- Jalankan aplikasi:



#### Task 4: Tema dan Sentuhan Akhir

##### A. Menjelajahi tema

- Buka **AndroidManifest.xml**, kemudian ubah `android:theme` untuk menghilangkan *action bar* menjadi seperti berikut  
`android:theme="@style/Theme.AppCompat.Light.NoActionBar"`

##### B. Tambahkan tombol tema ke menu

- Buka **strings.xml**, tambahkan string untuk night dan day mode seperti pada gambar dibawah:

```
<resources>
    <string name="app_name">Scorekeeper</string>
    <string name="minus_button_desc">Minus Button</string>
    <string name="count">0</string>
    <string name="team_1">Team 1</string>
    <string name="plus_button_desc">Plus Button</string>
    <string name="team_2">Team 2</string>
    <string name="night_mode">Night Mode</string>
    <string name="day_mode">Day Mode</string>
</resources>
```

- Klik kanan pada folder res dan pilih **New > Directory**, beri nama **menu**
- Kemudian klik kanan pada folder menu dan pilih **New > Android resource file**, beri nama **main\_menu**

- Buka **main\_menu.xml**, tambahkan item menu dengan atribut seperti berikut:

```
<?xml version="1.0" encoding="utf-8" ?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/night_mode"
        android:title="@string/night_mode">
    </item>
</menu>
```

- Pada **MainActivity.java**, tambahkan code program untuk menu inflater dibawah ini:

```
44         case R.id.increaseTeam2:
45             mScore2++;
46             mScoreText2.setText(String.valueOf(mScore2));
47         }
48     }
49
50     @Override
51     public boolean onCreateOptionsMenu(Menu menu) {
52         getMenuInflater().inflate(R.menu.main_menu, menu);
53         return super.onCreateOptionsMenu(menu);
54     }
55
56 }
```

### C. Ubah tema dari menu

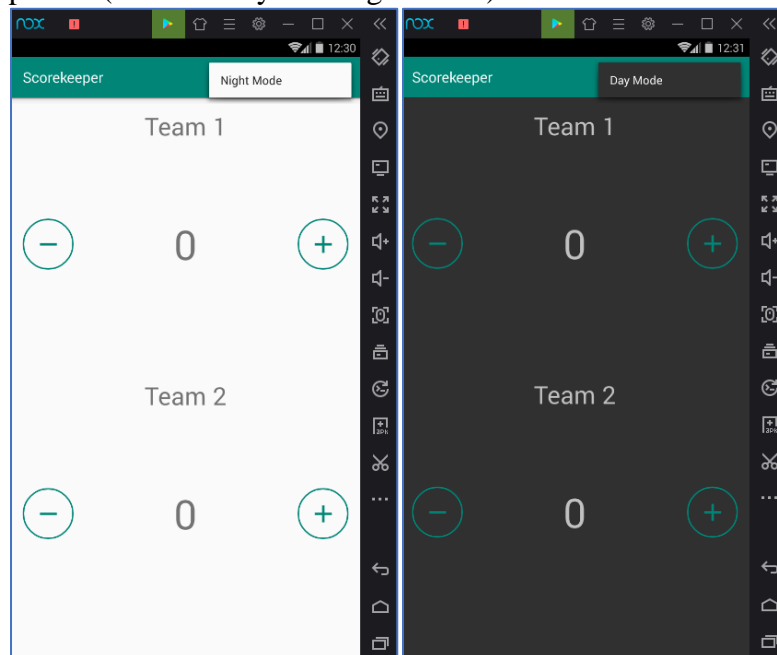
- Pada **style.xml**, ubah AppTheme menjadi berikut:

```
<style name="AppTheme" parent="Theme.AppCompat.DayNight.DarkActionBar">
```

- Pada **ActivityMain.java**, tambahkan code program seperti pada gambar dibawah:

```
52     @Override
53     public boolean onCreateOptionsMenu(Menu menu) {
54         getMenuInflater().inflate(R.menu.main_menu, menu);
55
56         int nightMode = AppCompatActivity.getDefaultNightMode();
57         if (nightMode == AppCompatActivity.MODE_NIGHT_YES) {
58             menu.findItem(R.id.night_mode).setTitle(R.string.day_mode);
59         }
60         else {
61             menu.findItem(R.id.night_mode).setTitle(R.string.night_mode);
62         }
63         return true;
64     }
65
66     @Override
67     public boolean onOptionsItemSelected(MenuItem item) {
68         if (item.getItemId() == R.id.night_mode) {
69             int nightMode = AppCompatActivity.getDefaultNightMode();
70
71             if (nightMode == AppCompatActivity.MODE_NIGHT_YES) {
72                 AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_NO);
73             }
74             else {
75                 AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_YES);
76             }
77             recreate();
78         }
79         return true;
80     }
81 }
```

- Jalankan aplikasi (memilih day atau night mode):



#### D. SaveInstanceState

- Pada **ActivityMain.java**, tambahkan code program seperti pada gambar dibawah:

```

11 public class MainActivity extends AppCompatActivity {
12     private int mScore1, mScore2;
13     TextView mScoreText1, mScoreText2;
14     static final String STATE_SCORE_1 = "Team 1 Score", STATE_SCORE_2 = "Team 2 Score";
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_main);
20
21         mScoreText1 = (TextView) findViewById(R.id.score_1);
22         mScoreText2 = (TextView) findViewById(R.id.score_2);
23
24         if(savedInstanceState != null){
25             mScore1 = savedInstanceState.getInt(STATE_SCORE_1);
26             mScore2 = savedInstanceState.getInt(STATE_SCORE_2);
27             mScoreText1.setText(String.valueOf(mScore1));
28             mScoreText2.setText(String.valueOf(mScore2));
29         }
30     }

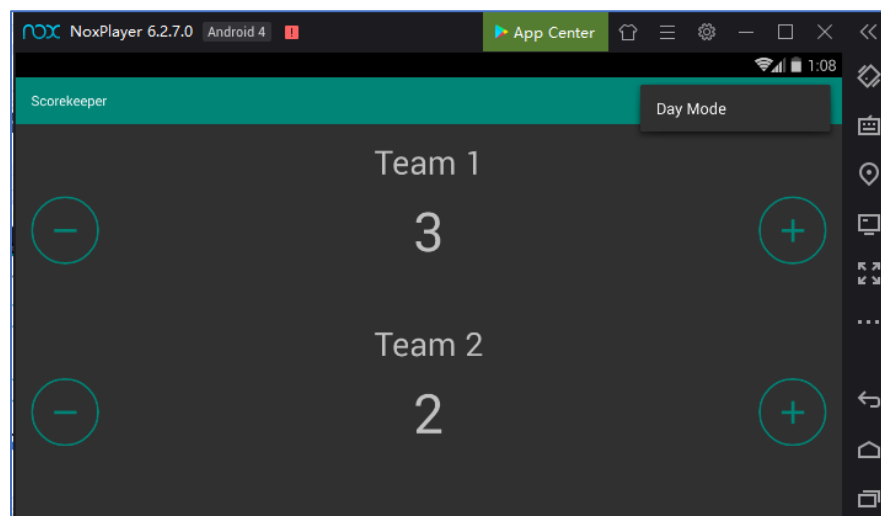
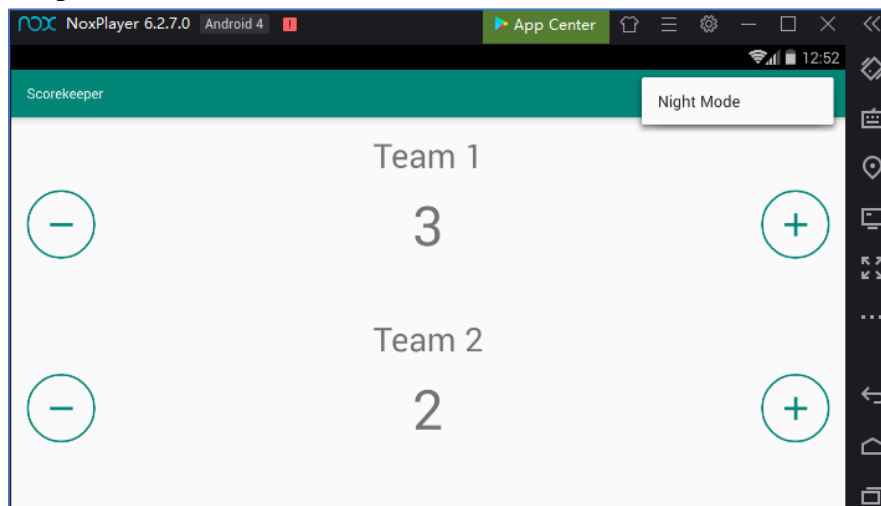
```

```

82         else {
83             AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_YES);
84         }
85         recreate();
86     }
87     return true;
88 }
89
90 @Override
91 protected void onSaveInstanceState(Bundle outState) {
92     outState.putInt(STATE_SCORE_1, mScore1);
93     outState.putInt(STATE_SCORE_2, mScore2);
94     super.onSaveInstanceState(outState);
95 }
96 }

```

- Jalankan aplikasi:



## Tugas 2. Card and Colors

### Task 1 : Mengunduh kode awal

#### A. Buka terlebih dahulu dan jalankan proyek Material Me

- Download the [MaterialMe-Starter](#) code!
- Apply di Android Studio
- Jalankan aplikasi tersebut

#### B. Jelajahi Aplikasi

- Pahami semua struktur yang terdapat pada kode aplikasi tersebut, bila tidak paham tanyakan

### Task 2 : Menambah CardView dan Gambar

#### A. Tambahkan CardView

- Buka file build.gradle (Module: app), dan tambahkan baris berikut ke dependencies bagian:

```
Compile 'com.android.support:cardview-v7:26.1.0'
```

- Buka file list\_item.xml

Attribute	Value
android:layout_width	"match_parent"
android:layout_height	"wrap_content"
android:layout_margin	"8dp"

- Pilih **Kode** > **Format** Ulang Kode untuk memformat ulang kode XML, yang sekarang akan terlihat seperti ini di awal dan akhir file:

```
<android.support.v7.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp">
```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <!-- Rest of LinearLayout -->
    <!-- TextView elements -->
</LinearLayout>
</android.support.v7.widget.CardView>

```

- Jalankan aplikasi

## B. Download file gambar

- Unduh [file zip gambar banner](#).
- Buka direktori **MaterialMe > app > src > main > res** di file explorer sistem operasi Anda, dan buat direktori **drawable**, dan salin masing-masing file grafik ke direktori **drawable**.
- Anda memerlukan array dengan path ke setiap gambar sehingga Anda bisa memasukkannya ke Sportsobjek. Untuk melakukan ini, tentukan array yang berisi semua jalur ke drawable sebagai item dalam **string.xml** file Anda . Pastikan bahwa mereka berada dalam urutan yang sama karena **sports\_titles** array juga didefinisikan dalam file yang sama:

```

<array name="sports_images">
    <item>@drawable/img_baseball</item>
    <item>@drawable/img_badminton</item>
    <item>@drawable/img_basketball</item>
    <item>@drawable/img_bowling</item>
    <item>@drawable/img_cycling</item>
    <item>@drawable/img_golf</item>
    <item>@drawable/img_running</item>
    <item>@drawable/img_soccer</item>
    <item>@drawable/img_swimming</item>
    <item>@drawable/img_tabletennis</item>
    <item>@drawable/img_tennis</item>
</array>

```

### C. Memodifikasi objek Olahraga

- Tambahkan variabel anggota integer ke objek Sport yang akan berisi sumber daya yang dapat digambar:

```
private final int imageResource;
```

- Modifikasi konstruktor sehingga akan mengambil sebuah integer sebagai parameter dan menentukannya ke variabel anggota:

```
public Sport(String title, String info, int imageResource) {  
    this.title = title;  
    this.info = info;  
    this.imageResource = imageResource;  
}
```

- Buat getter untuk integer sumber daya:

```
public int getImageResource() {  
    return imageResource;  
}
```

### D. Memperbaiki metode initializeData ()

- Dalam metode **initializeData()**, dapatkan **TypedArray** id sumber daya dengan memanggil **getResources().obtainTypedArray()**, dan meneruskan nama larik sumber daya yang dapat digambar, yang Anda definisikan di file strings.xml:

```
TypedArray sportsImageResources =  
getResources().obtainTypedArray(R.array.sports_images);
```

- Perbaiki kode di loop yang membuat objek Sport, dengan menambahkan ID sumber daya yang dapat digambar yang sesuai sebagai parameter ketiga dengan memanggil **getResourceId()** di **TypedArray**:

```
for(int i=0;i<sportsList.length;i++){  
    mSportsData.add(new Sport(sportsList[i],sportsInfo[i],  
        sportsImageResources.getResourceId(i,0)));  
}
```

- Bersihkan data di TypedArray setelah Anda membuat ArrayList data Sport:

```
sportsImageResources.recycle();
```



#### E. Tambahkan **ImageView** ke daftar item

- Ubah bagian **LinearLayout** dalam file **list\_item.xml** menjadi **RelativeLayout**, dan hapus **android:orientation** atributnya.
- Tambahkan elemen **ImageView** sebagai pertama dalam **RelativeLayout** dengan atribut berikut:

Atribut	Nilai
android:layout_width	"match_parent"
android:layout_height	"wrap_content"
android:id	"@id/sportsImage"
android:adjustViewBounds	"true"

- Tambahkan atribut berikut ke title **TextView** elemen:

Atribut	Nilai
android:layout_alignBottom	"@id/sportsImage"
android:theme	"@style/ThemeOverlay.AppCompat.Dark"

- Tambahkan atribut berikut ke newsTitle **TextView** elemen:

Atribut	Nilai
android:layout_below	"@id/sportsImage"
android:textColor	"?android:textColorSecondary"

- Tambahkan atribut berikut ke subTitle **TextView** elemen:

Atribut	Nilai
android:layout_below	"@id/newsTitle"

#### F. Memuat gambar menggunakan **Glide**

- Tambahkan dependensi berikut untuk **Glide**, di file **build.gradle** tingkat aplikasi Anda:

```
compile 'com.github.bumptech.glide:glide:3.5.2'
```

- Tambahkan variabel di kelas **SportsAdapter**, kelas **ViewHolder** untuk **ImageView**, dan lakukan inisialisasi di konstruktor **ViewHolder**:

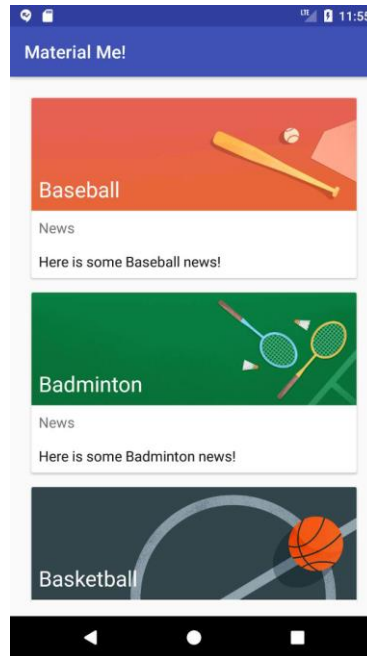
```
mSportsImage = (ImageView) itemView.findViewById(R.id.sportsImage);
```

- Tambahkan baris kode berikut ke **onBindViewHolder()** untuk mendapatkan sumber daya gambar dari objek **Sport** dan muat ke dalam **ImageView** menggunakan **Glide**:

```
Glide.with(mContext).load(currentSport.getImageResource()).into(holder.mSportsImage);
```

- Jalankan aplikasi.

Output:



### Task 3 : Membuat CardView yang bisa diswipe, dipindahkan, dan diklik

#### A. Menerapkan Swipeab

- Buka **MainActivity** dan buat **ItemTouchHelper** objek baru di **onCreate()** metode di akhir, di bawah **initializeData()** metode.

```
ItemTouchHelper helper = new ItemTouchHelper(new
    ItemTouchHelper.SimpleCallback() {

        @Override
        public boolean onMove( RecyclerView recyclerView,
                               RecyclerView.ViewHolder viewHolder,
                               RecyclerView.ViewHolder target) {
            return false;
        }

        @Override
        public void onSwiped(RecyclerView.ViewHolder viewHolder,
                              int direction) {

        }
    });
```

- Karena kita hanya mengimplementasikan gesek untuk menutup sekarang, Anda harus meneruskan 0 untuk arah pemindahan yang didukung dan **ItemTouchHelper.LEFT** | **ItemTouchHelper.RIGHT** untuk arah penggesekan yang didukung:

```
ItemTouchHelper helper = new ItemTouchHelper(
    new ItemTouchHelper.SimpleCallback(0, ItemTouchHelper.LEFT |
ItemTouchHelper.RIGHT) {
```

- Sekarang Anda harus mengimplementasikan perilaku yang diinginkan di **onSwiped()**. Dalam hal ini, menggesek kartu ke kiri atau ke kanan akan menghapusnya dari daftar. Panggil **remove()** di rangkaian data, dengan meneruskan indeks yang sesuai dengan mendapatkan posisi dari **ViewHolder**:

```
mSportsData.remove(viewHolder.getAdapterPosition());
```

- Agar **RecyclerView** dapat menggerakkan penghapusan dengan benar, Anda juga harus memanggil **notifyItemRemoved()**, lagi dengan meneruskan indeks yang sesuai dengan mendapatkan posisi dari **ViewHolder**:

```
mAdapter.notifyItemRemoved(viewHolder.getAdapterPosition());
```

- Setelah membuat objek **ItemTouchHelper** baru di metode **onCreate()** **MainActivity**, panggil **attachToRecyclerView()** di instance **ItemTouchHelper** untuk menambahkannya ke **RecyclerView** Anda:

```
helper.attachToRecyclerView(mRecyclerView);
```

- Jalankan aplikasi Anda

## B. Implementasi Drag and Drop

- Ubah argumen pertama SimpleCallback dari 0 untuk menyertakan setiap arah, karena kita ingin bisa melakukan seret dan lepas di mana pun:

```
ItemTouchHelper helper = new ItemTouchHelper(new ItemTouchHelper
.SimpleCallback(ItemTouchHelper.LEFT | ItemTouchHelper.RIGHT |
ItemTouchHelper.DOWN | ItemTouchHelper.UP, ItemTouchHelper.LEFT |
ItemTouchHelper.RIGHT) {}
```

- Dalam metode **onMove()**, dapatkan indeks asli dan target dari argumen kedua dan ketiga yang diteruskan (sesuai dengan **ViewHolder** asli dan target).

```
int from = viewHolder.getAdapterPosition();  
int to = target.getAdapterPosition();
```

- Pindahkan item di kumpulan data dengan memanggil **Collections.swap()** dan meneruskan kumpulan data dan indeks awal dan akhir:

```
Collections.swap(mSportsData, from, to);
```

- Beri tahu adaptor bahwa item sudah dipindahkan, dengan meneruskan indeks lama dan baru:

```
mAdapter.notifyItemMoved(from, to);
```

- Jalankan aplikasi Anda.

### C. Menerapkan tata letak **DetailActivity**

- Buat aktivitas baru dengan masuk ke **File > New > Activity > Empty Activity**.
- Beri nama **DetailActivity**, dan terima semua default.
- Di file layout yang baru dibuat, buang padding dari **RelativeLayout** rootview.
- Salin semua dari tampilan **TextView** dan **ImageView** dari file **list\_item.xml** ke file **activity\_detail.xml**.
- Tambahkan kata "Detail" ke setiap referensi ke sebuah id, untuk membedakannya dengan id **list\_item**. Misalnya, ubah id **ImageView** dari **sportsImage** ke **sportsImageDetail**, serta semua referensi ke id ini untuk penempatan relatif, seperti **layout\_below**.
- Untuk **textview subTitleDetail**, buang semua string teks placeholder dan tempelkan paragraf teks generik untuk menggantikan semua teks detail (Misalnya, beberapa paragraf Lorem Ipsum).
- Ubah padding **TextView** menjadi **16dp**.
- Bungkus keseluruhan **activity\_detail.xml** di **ScrollView** dan ubah atribut **layout\_height RelativeLayout** ke **"wrap\_content"**.

```
<?xml version="1.0" encoding="utf-8"?>  
<ScrollView  
  xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">
```

```
<RelativeLayout xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    tools:context="com.example.android.materialme.DetailActivity">
```

#### D. Terapkan tampilan detail dan klik pendengar

- Buka **SportsAdapter** dan ubah ViewHolder kelas dalam, yang sudah diperluas **RecyclerView.ViewHolder**, untuk juga mengimplementasikan **View.OnClickListener**, dan mengimplementasikan metode yang diperlukan (**onClick()**).

```
class ViewHolder extends RecyclerView.ViewHolder
    implements View.OnClickListener{

    // Rest of ViewHolder code.
    //
    @Override
    public void onClick(View view) {
    }
}
```

- Setel **OnClickListener** ke dalam itemView di **ViewHolder** konstruktor. Seluruh konstruktor sekarang akan terlihat seperti ini:

```
ViewHolder(View itemView) {
    super(itemView);

    //Initialize the views
    mTitleText = itemView.findViewById(R.id.title);
    mInfoText = itemView.findViewById(R.id.subTitle);
    mSportsImage = itemView.findViewById(R.id.sportsImage);

    // Set the OnClickListener to the entire view.
    itemView.setOnClickListener(this);
}
```

- Dalam **onClick()** metode ini, dapatkan Sport objek untuk item yang diklik menggunakan **getAdapterPosition()**:

```
Sport currentSport = mSportsData.get(getAdapterPosition());
```

- Dalam metode yang sama, menambahkan **Intent** yang meluncurkan **DetailActivity**, menempatkan **title** dan **image\_resource** sebagai tambahan dalam **Intent**, dan panggilan **startActivity()** pada **mContextvariabel**, lewat di baru **Intent**.

```
Intent detailIntent = new Intent(mContext, DetailActivity.class);
detailIntent.putExtra("title", currentSport.getTitle());
detailIntent.putExtra("image_resource", currentSport.getImageResource(
));
mContext.startActivity(detailIntent);
```

- Buka **DetailActivity** dan inisialisasi **ImageView** dan **title TextView** di **onCreate()**:

```
TextView sportsTitle = findViewById(R.id.titleDetail);
ImageView sportsImage = findViewById(R.id.sportsImageDetail);
```

- Dapatkan **title** dari yang masuk **Intent** dan atur ke **TextView**:

```
sportsTitle.setText(getIntent().getStringExtra("title"));
```

- Gunakan **Glide** untuk memuat gambar ke **ImageView**.

```
Glide.with(this).load(getIntent().getIntExtra("image_resource", 0))
.into(sportsImage);
```

- Jalankan aplikasi.

## Task 4 : Tambahkan FAB dan pilih palet warna Desain Material

### A. Tambahkan FAB

- Buka file **build.gradle (Module: app)** dan tambahkan baris kode berikut untuk perpustakaan dukungan desain di dependencies bagian:

```
compile 'com.android.support:design:24.2.1'
```

- Tambahkan ikon untuk FAB dengan mengklik kanan (atau Kontrol-mengklik) folder res di **Project > Android pane**, dan memilih **New > Vector Asset**. FAB akan mengatur ulang isi **RecyclerView**, Ubah nama refresh ikon menjadi **ic\_reset**, klik Berikutnya, dan klik Selesai.
- Buka **activity\_main.xml** dan tambahkan **FloatingActionButton** dengan atribut berikut:

Atribut	Nilai
android:layout_width	"wrap_content"
android:layout_height	"wrap_content"

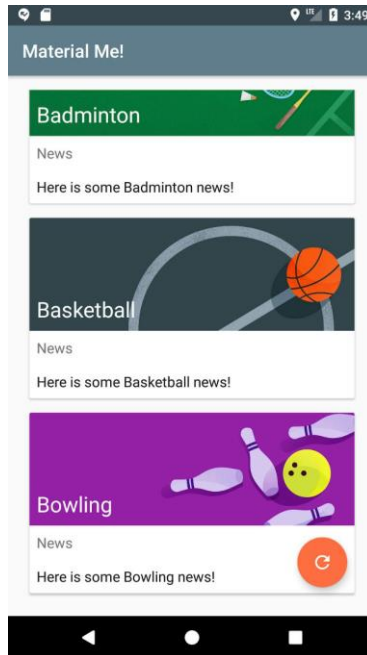
<code>android:layout_alignParentBottom</code>	<code>"true"</code>
<code>android:layout_alignParentRight</code>	<code>"true"</code>
<code>android:layout_alignParentEnd</code>	<code>"true"</code>
<code>android:layout_margin</code>	<code>"16dp"</code>
<code>android:src</code>	<code>"@drawable/ic_reset"</code>
<code>android:tint</code>	<code>"@android:color/white"</code>
<code>android:onClick</code>	<code>"resetSports"</code>

- Buka **MainActivity** dan tambahkan **resetSports()** metode dengan pernyataan untuk memanggil **initializeData()** untuk mereset data.
- Jalankan aplikasi.

## B. Pilih palet Desain Bahan

- Di panel **Project > Android** , navigasikan ke file **styles.xml** Anda (terletak di direktori nilai). Tema gaya mendefinisikan tiga warna secara default: **colorPrimary**, **colorPrimaryDark**, dan **colorAccent**. Gaya-gaya ini ditentukan oleh nilai-nilai dari **colors.xml** file.
- Pilih warna untuk digunakan sebagai warna utama Anda, seperti **#607D8B** (dalam contoh warna Biru Abu-abu). Itu harus berada dalam kisaran 300-700 dari swatch warna sehingga Anda masih bisa memilih aksen yang tepat dan warna gelap.
- Buka file **colors.xml** , dan modifikasi nilai **colorPrimary** hex agar sesuai dengan warna yang Anda pilih.
- Pilih warna yang lebih gelap dengan warna yang sama untuk digunakan sebagai warna gelap utama Anda, seperti **#37474F**. Sekali lagi, modifikasi nilai hex **colors.xml** **colorPrimaryDark** agar sesuai.
- Pilih warna aksen untuk FAB Anda dari warna yang nilainya dimulai dengan A, dan yang warnanya kontras dengan warna primer (seperti Deep Orange A200). Ubah **colorAccent** nilai dalam **colors.xml** agar sesuai.
- Jalankan aplikasi.

Output :





### Tugas 3. Adaptive Layout

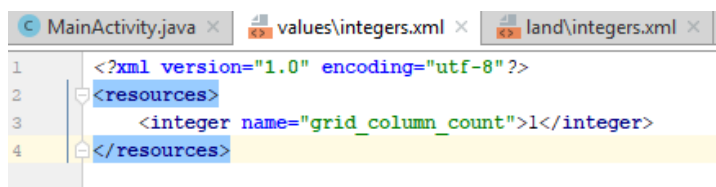
Perangkat Android hadir dalam berbagai bentuk dan ukuran sehingga tata letak aplikasi yang dibangun harus bersifat fleksibel. Artinya, tata letak tidak boleh ditentukan dengan dimensi mutlak yang mengasumsikan rasio tinggi lebar dan ukuran layar tertentu; tetapi harus mampu merespon berbagai ukuran dan orientasi layar tanpa masalah.

Terdukungnya berbagai ukuran layar, aplikasi nantinya dapat tersedia bagi sebanyak mungkin pengguna di berbagai perangkat cukup dengan satu APK. Selain itu, memungkinkan aplikasi untuk berfungsi pada berbagai ukuran layar juga akan memastikan bahwa aplikasi bisa menangani perubahan konfigurasi jendela di perangkat.

*Praktik ini melanjutkan aplikasi "Material Me!" yang telah dibuat sebelumnya. Selesaikan praktikum sebelumnya untuk membuat aplikasi berikut.*

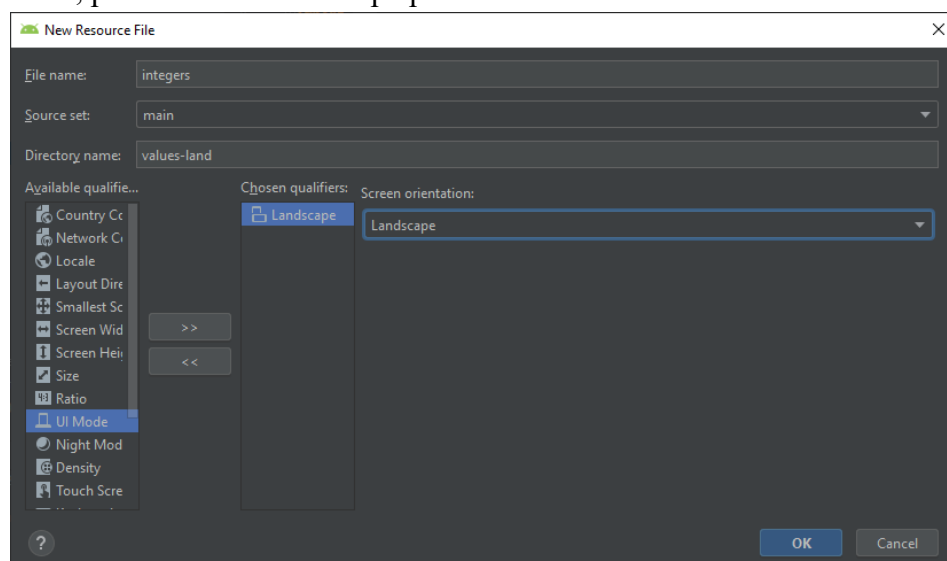
#### A. Membuat file resource

- Buatlah resource file baru dan beri nama integers.xml. Tambahkan integer bernilai 1, dimana ketika aplikasi ini dibuka di hp dengan orientasi portrait maka tampilan CardViewnya berjumlah satu.

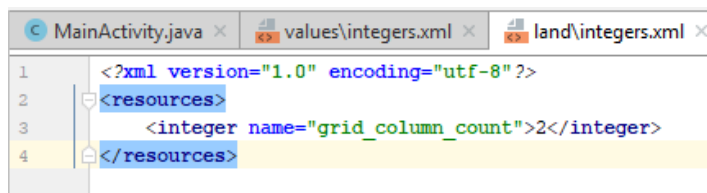


```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <integer name="grid_column_count">1</integer>
4 </resources>
```

- Buatlah resource file baru, kemudian beri nama integers, pilih Orientation lalu klik tombol ">>", pilih orientasi landscape pada Screen orientation.



- Tambahkan integer bernilai 2, dimana ketika aplikasi ini dibuka di hp dengan orientasi landscape maka tampilan CardViewnya berjumlah dua.

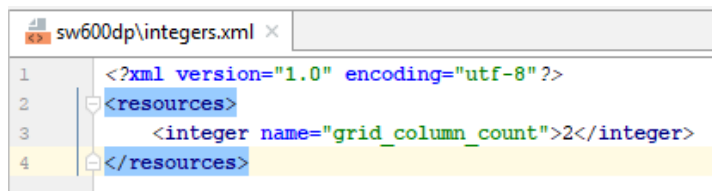


```

1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <integer name="grid_column_count">2</integer>
4  </resources>

```

- Buatlah resource file baru, beri nama integers, pilih Smallest-screen width, klik tombol ">>>", lalu masukkan angka 600. Artinya, resource akan dipanggil ketika ukuran lebar layar minimal 600 px atau ukuran standar Tablet.
- Tambahkan integer bernilai 2, dimana ketika aplikasi ini dibuka di tablet dengan orientasi potrait maka tampilan CardViewnya berjumlah dua.

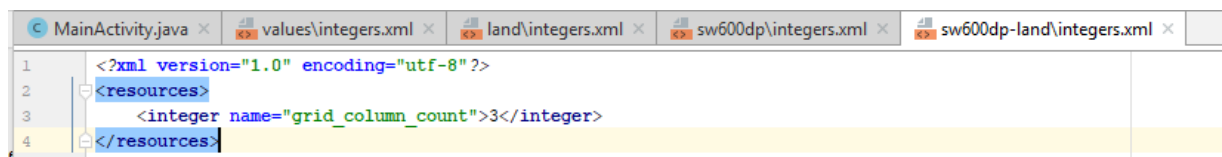


```

1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <integer name="grid_column_count">2</integer>
4  </resources>

```

- Buatlah resource file baru, beri nama integers, pilih Smallest-screen width, klik tombol ">>>", lalu masukkan angka 600. Lalu pilih Orientation, klik tombol ">>>", dan masukkan Landscape.
- Tambahkan integer bernilai 2, dimana ketika aplikasi ini dibuka di tablet dengan orientasi landscape maka tampilan CardViewnya berjumlah dua.



```

1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <integer name="grid_column_count">3</integer>
4  </resources>

```

- Pada MainActivity.java, ubah LinearLayoutManager dari praktikum sebelumnya dengan GridLayoutManager. Setelah ditambahkan, akan tampak pada gambar di bawah ini.

```

42      @Override
43      protected void onCreate(Bundle savedInstanceState) {
44          super.onCreate(savedInstanceState);
45          setContentView(R.layout.activity_main);
46
47          //Initialize the RecyclerView
48          mRecyclerView = (RecyclerView) findViewById(R.id.recyclerView);
49
50          //Set the Layout Manager
51          //mRecyclerView.setLayoutManager(new LinearLayoutManager(this));
52
53          int gridColumnCount = getResources().getInteger(R.integer.grid_column_count);
54          //Set to GridLayout
55          mRecyclerView.setLayoutManager(new GridLayoutManager(context, this, gridColumnCount));
56

```

## B. Memperbarui tema

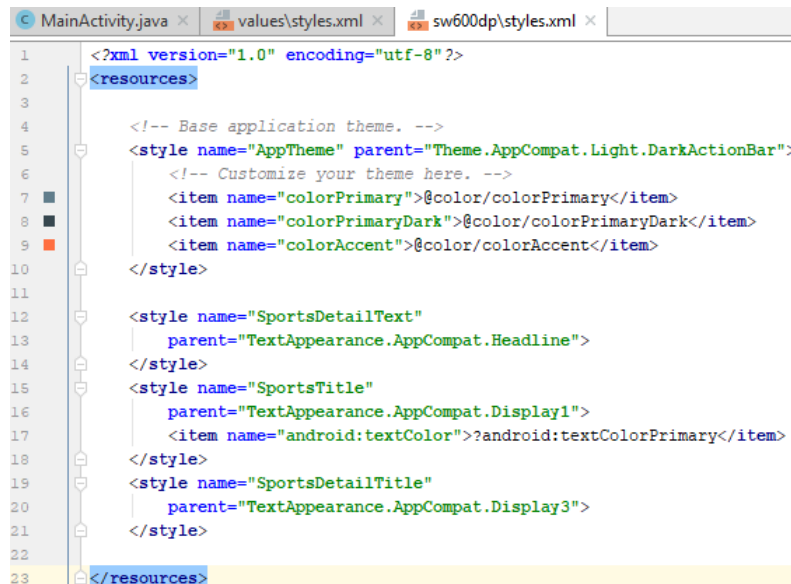
- Tambahkan beberapa style baru di bawah AppTheme pada file styles.xml.

```

16      <resources>
17
18          <!-- Base application theme. -->
19          <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
20              <!-- Customize your theme here. -->
21              <item name="colorPrimary">@color/colorPrimary</item>
22              <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
23              <item name="colorAccent">@color/colorAccent</item>
24          </style>
25
26          <style name="SportsDetailText"
27              parent="TextAppearance.AppCompat.Subhead">
28          </style>
29          <style name="SportsTitle"
30              parent="TextAppearance.AppCompat.Headline">
31          </style>
32          <style name="SportDetailTitle"
33              parent="TextAppearance.AppCompat.Headline">
34          </style>
35
36      </resources>

```

- Tambahkan file resource baru dengan nama styles.xml, pilih Smallest-screen width, lalu masukkan angka 600.
- Salin source code dari styles.xml ke file yang baru dibuat tadi.

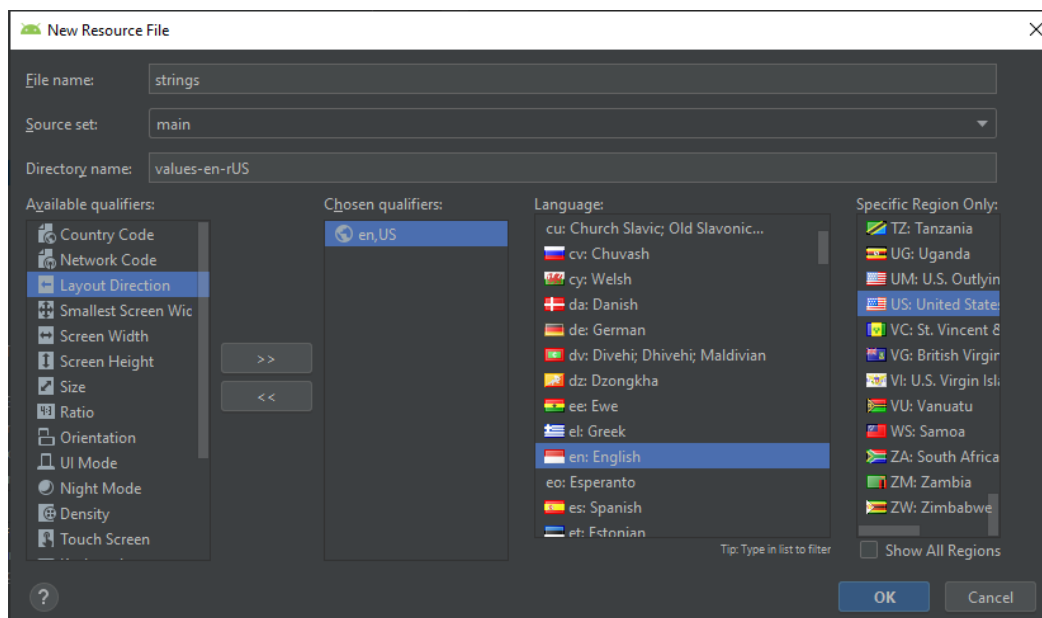


```

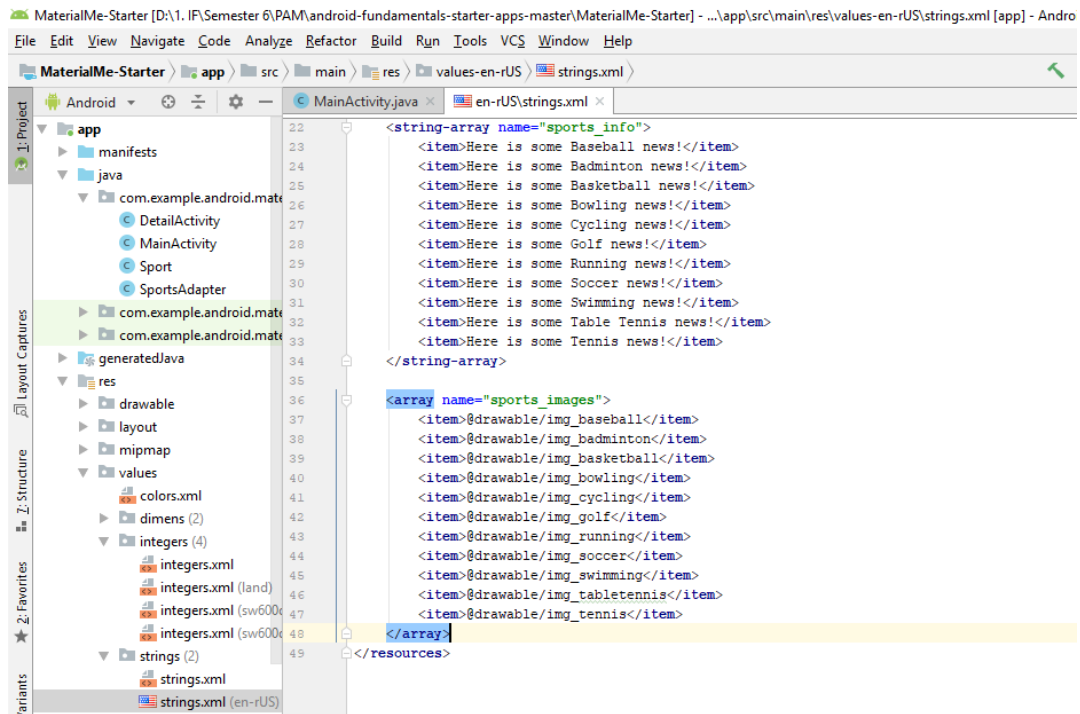
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3
4      <!-- Base application theme. -->
5      <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
6          <!-- Customize your theme here. -->
7          <item name="colorPrimary">@color/colorPrimary</item>
8          <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
9          <item name="colorAccent">@color/colorAccent</item>
10     </style>
11
12     <style name="SportsDetailText"
13         parent="TextAppearance.AppCompat.Headline">
14     </style>
15
16     <style name="SportsTitle"
17         parent="TextAppearance.AppCompat.Display1">
18         <item name="android:textColor">?android:textColorPrimary</item>
19     </style>
20
21     <style name="SportsDetailTitle"
22         parent="TextAppearance.AppCompat.Display3">
23     </style>
24 </resources>
  
```

### C. Melokalkan aplikasi

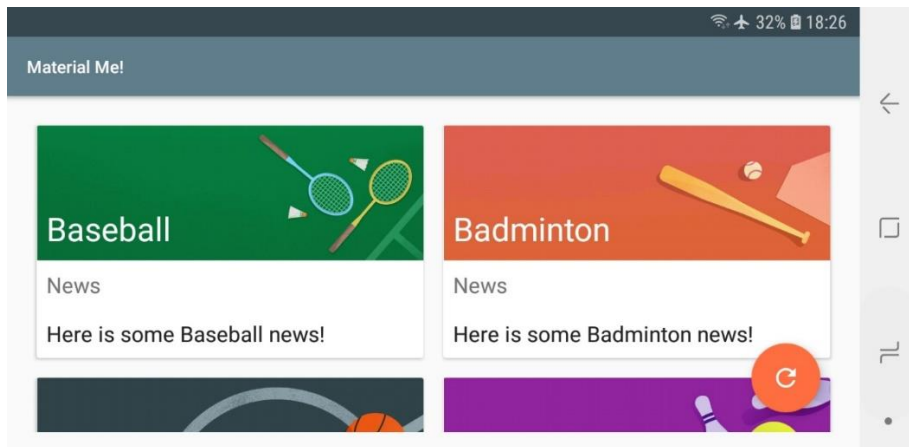
- Tambahkan file resource baru bernama strings.xml, pilih Locale, klik English dan pilih US.



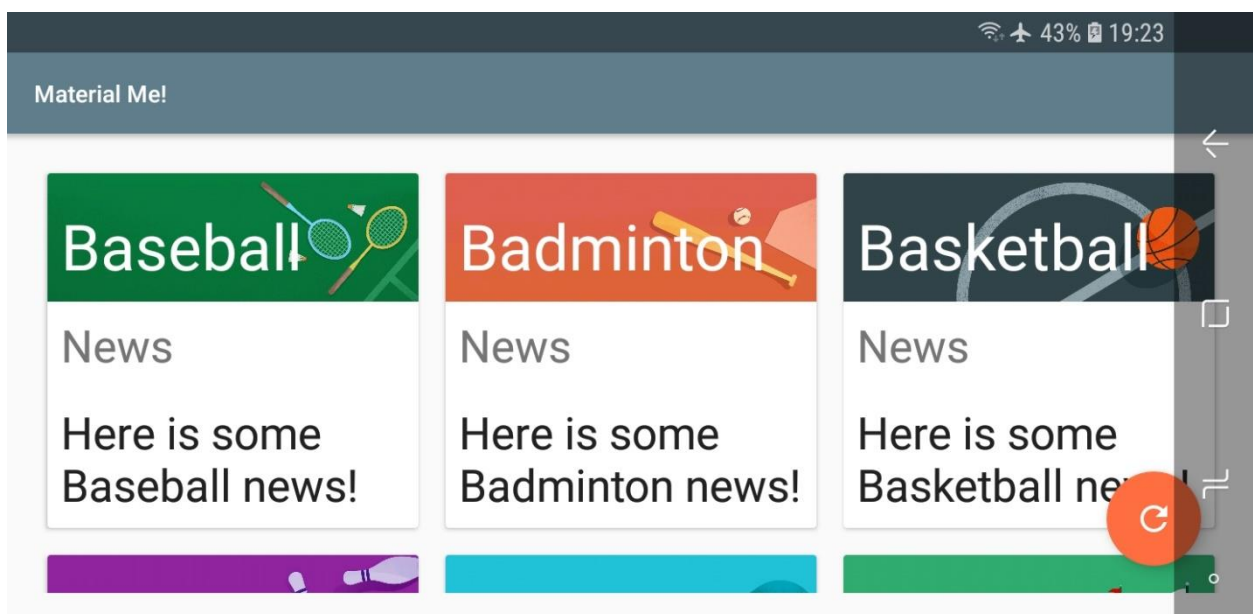
- Salin source code dari strings.xml yang sudah dibuat pada praktikum sebelumnya, kemudian terjemahkan teks sesuai dengan bahasa lokal yang dipilih yaitu bahasa US.



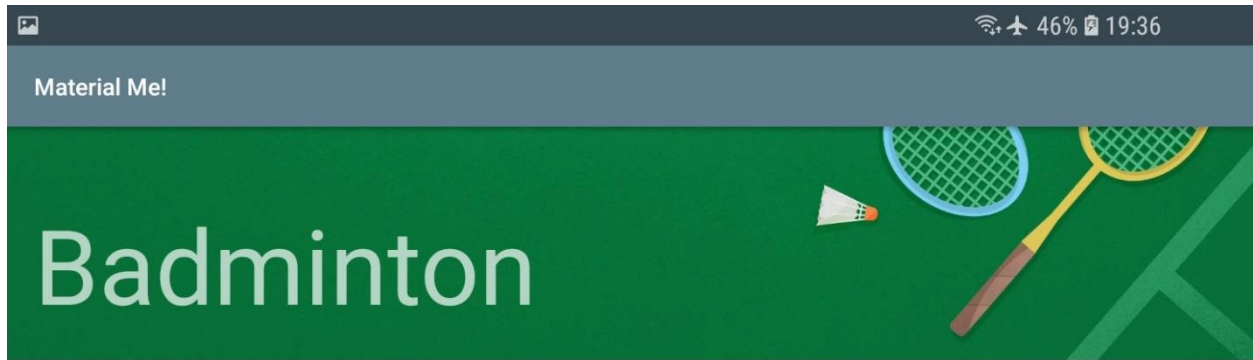
**Tampilan akhir :**



Tampilan emulator hp landscape.



Tampilan emulator tablet landscape.



## News

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna

Tampilan emulator tablet ketika konten yang dipilih berhasil ditampilkan.

.