

Modul Perkuliahan

**Pemrograman *Mobile*:
Desain Tematik**

Dr. Habibullah Akbar

Session 6

2019

SESSION OUTCOMES

1. Melakukan *refreshment* mengenai konsep UI beserta komponen-komponennya dan menu navigasi untuk memberikan pengalaman *user* yang baik.
2. Mahasiswa dapat memahami style dan tema pada aplikasi Android
3. Mahasiswa dapat memahami desain material
4. Mahasiswa membuat aplikasi skor pencatat pertandingan dan menerapkan tema dan desain material

OUTLINE MATERI:

1. Pendahuluan
2. Desain Material
3. Mode dan Tema
4. Membuat Aplikasi Pencatat Skor Pertandingan

ISI MATERI

1. Pendahuluan

Saat ini anda telah mampu membuat aplikasi Android menggunakan IDE Android Studio dan menjalankannya baik di *smartphone* maupun pada *emulator*. Lanjutan dari pembahasan *user experience* pada sesi sebelumnya, sesi ini anda membahas cara mempercantik tampilan dari aplikasi anda. Untuk itu, Android telah membuat Bahasa desain secara khusus yang disebut dengan desain material (*material design*). Anda akan belajar untuk membuat aplikasi Pencatat Skor Pertandingan. Kemudian anda akan mencoba untuk meningkatkan tampilan pada komponen UI berdasarkan konsep desain material dan konsep tema yang berbeda pada sebuah layout untuk memberikan efek personalisasi bagi *user*.

2. Desain Material

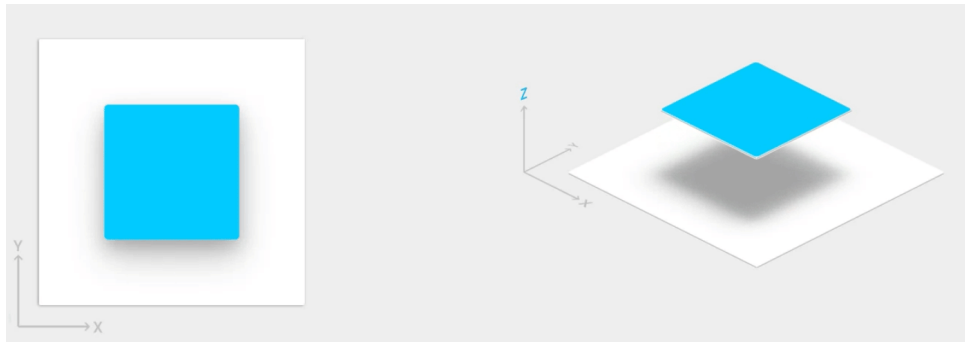
Desain material (*material design*) adalah Bahasa desain yang dikembangkan Google untuk memberikan UX yang baik dengan memberikan rekomendasi pengaturan *view* pada UI. Pengaturan yang dimaksud mencakup aspek pola layout, animasi, efek transisi, jarak antar elemen (*padding*), efek cahaya dan bayangan.

Untuk memberikan *user experience* yang intuitif dan artistik, anda sebaiknya mengikuti panduan spesifikasi desain material (<https://material.io/guidelines/>) dan komponen dan mode yang telah disediakan di referensi pendukung desain material di <https://developer.android.com/topic/libraries/support-library/features.html#material-design>.

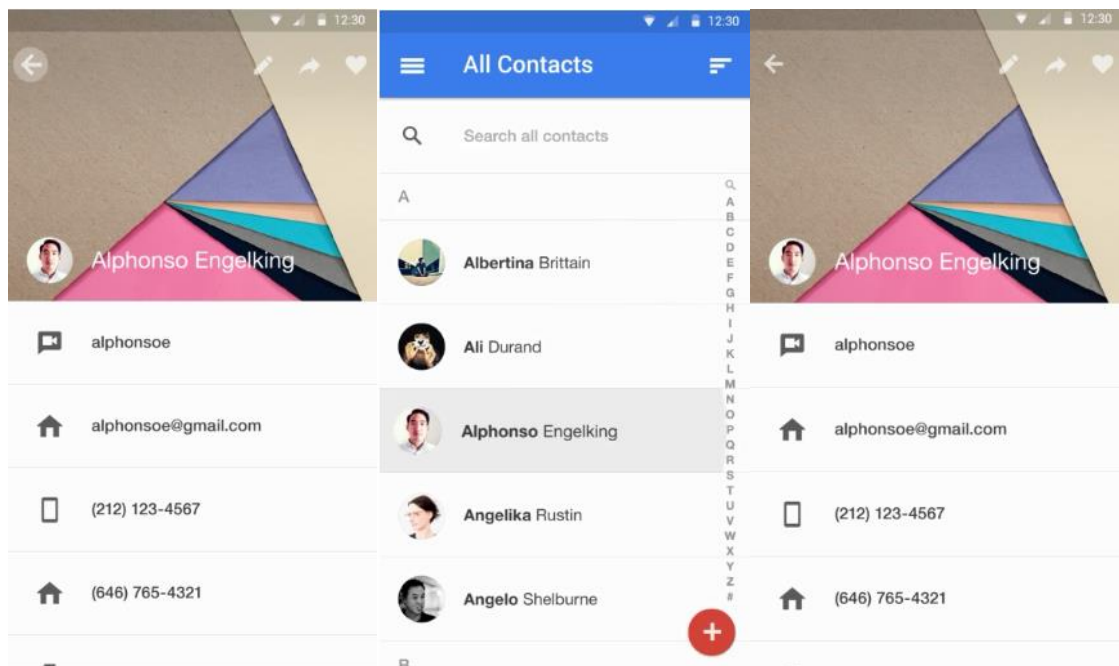
Android menyediakan beberapa fasilitas desain material sebagai berikut

- Tema aplikasi untuk memberikan sentuhan seni pada komponen UI
- Komponen UI yang lebih kompleks yaitu daftar (*lists*) dan kartu (*cards*).
- API untuk efek bayangan dan animasi

Untuk memberikan efek bayangan, *view* memiliki bukan hanya sumbu X dan Y namun juga sumbu Z. Gambar dibawah ini mengilustrasikan bagaimana sumbu-sumbu tersebut dapat memberikan efek bayangan. Besarnya bayangan bergantung pada nilai dari sumbu Z. Nilai yang lebih tinggi akan menjauhkan objek dari permukaan sehingga memperluas ukuran bayangan dari *view*, relatif terhadap induknya.



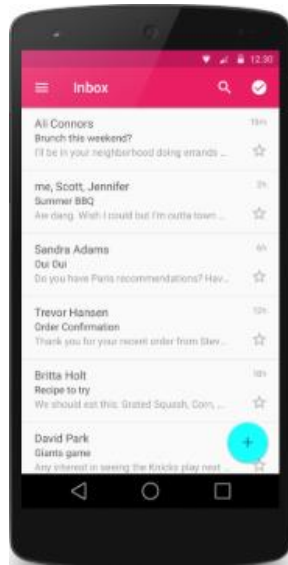
Animasi biasanya digunakan sebagai umpanbalik dari perilaku *user* seperti perubahan keadaan *view* ataupun transisi antar aktifitas. Gambar dibawah ini memberikan ilustrasi: ketika user mengklik tombol balik (gambar-kiri) maka akan terjadi transisi menuju aktifitas *All Contacts* (gambar-tengah). Dan ketika user memilih kontak misal Alphonso maka terjadi transisi ke detik kontaknya (gambar-kanan).



API yang disediakan Android menyediakan beberapa tipe efek berikut:

- Respon saat user menyentuh suatu *view* seperti tombol.
- Menyembunyikan dan menampilkan *view* dengan efek animasi melingkar.
- Kustomisasi transisi antar aktifitas
- Menyediakan animasi yang dapat mengikuti suatu lengkungan
- Menyediakan animasi perubahan antar keadaan baik *view* maupun *drawable*.

Pada contoh diatas, layout *All Contacts* tersebut dapat dibuat menggunakan *RecyclerView* dan *CardView*. *RecyclerView* biasanya digunakan untuk menampilkan daftar dari item-item yang dapat di-*scroll*. Contoh penggunaannya diberikan pada gambar dibawah ini



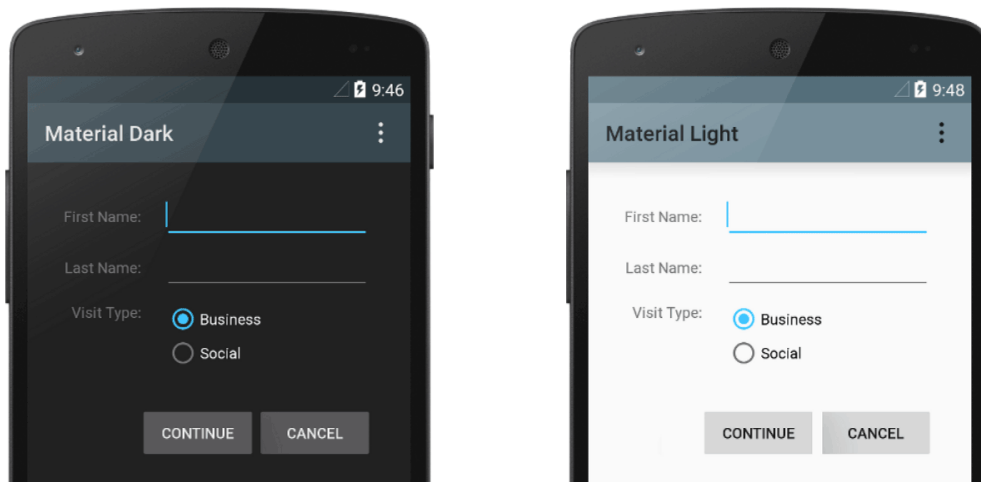
Namun layout dari *RecyclerView* tersebut masih nampak kaku. Oleh karena itu, developer dapat menanamkan *CardView*, sebuah kontainer yang dapat menggabungkan beberapa *views* dalam sebuah kontainer. *RecyclerView* dapat berisi *CardView* seperti pada contoh dibawah ini



3. Mode dan Tema

Istilah mode (*style*) disini maksudnya adalah kumpulan dari atribut yang mengatur penampilan dari sebuah View. Penampilan yang diatur mencakup atribut warna teks, ukuran teks, warna latar belakang layout dan sebagainya.

Adapun istilah tema adalah mode yang dapat diterapkan pada keseluruhan komponen aplikasi dan layout. Contoh penerapan dua tema yang berbeda pada sebuah layout diberikan pada gambar berikut (gambar-kiri: `Theme.AppCompat` dan gambar-kanan: `Theme.AppCompat.Light`)



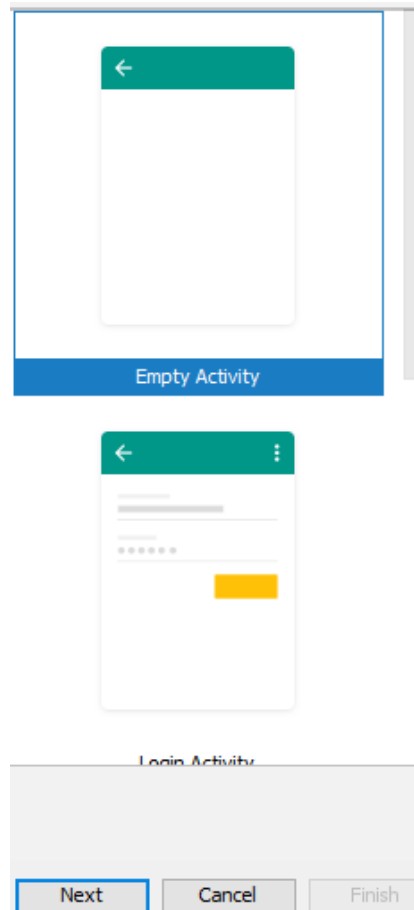
Deklarasi mode dan tema dilakukan pada file `styles.xml` yang berada pada folder `res/values/`.

4. Membuat Aplikasi Pencatat Skor Pertandingan

Pada latihan kali ini anda akan membuat aplikasi pencatat skor pertandingan. Note: pada latihan kali ini, anda dianggap telah lancar dalam menggunakan Android Studio, dapat menerapkan metode klik (`onClick`) pada Button, menggunakan variabel global string pada file `strings.xml` dan menggunakan editor layout.

Buat Proyek

Buatlah proyek baru dan berikan nama `PencatatSkor`, pilih templat *Empty Activity* dan klik *Finish*.



Desain Layout untuk MainActivity

Berikutnya, bukalah file `activity_main.xml` dan pilih tab `Text` kemudian tukarlah layout constraint berikut

```
android.support.constraint.ConstraintLayout
```

menjadi layout linear

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

Lalu hapus kode berikut

```
xmlns:app="http://schemas.android.com/apk/res-auto"
```

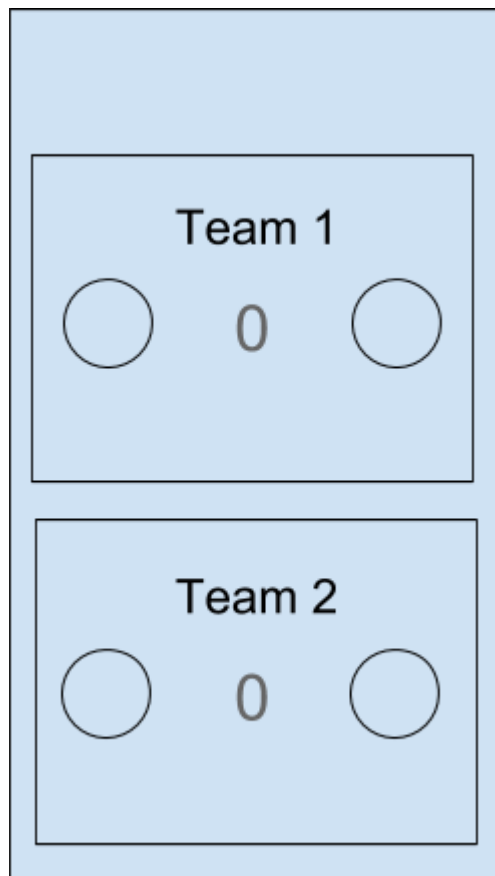
Hasil dari kode xml akan menjadi sebagai berikut

```
<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"
    tools:context="com.example.android.scorekeeper.MainActivity">
```

Lalu tambahkan atribut berikut

Attribute	Value
android:orientation	"vertical"
android:padding	"16dp"

Sekarang marilah kita buat container untuk 2 tim yang akan bertanding. Sketsa sederhana dari aplikasi diberikan pada gambar berikut ini



Didalam layout linear di atas, tambahkan 2 buah layout relatif berserta atribut-atribut berikut ini

RelativeLayout attribute	Value
android:layout_width	"match_parent"
android:layout_height	"0dp"
android:layout_weight	"1"

Atribut *layout_height* diset menjadi 0dp karena anda akan menggunakan atribut *layout_weight* untuk mengatur luas relatifnya terhadap layout induknya. Layout relatif pertama digunakan untuk *Team1* dan layout relatif kedua untuk *Team2*.

Berikutnya tambahkan 2 buah *ImageButton* untuk masing-masing layout relatif tersebut: 1 untuk menambah skor dan satu lagi untuk mengurangi skor. Berikut masing-masing *view* dengan atribut berikut

ImageButton attribute	Value
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"

Untuk *ImageButton* berikutnya, set nilai atribut *android:id* menjadi *increaseTeam1* untuk *ImageButton* kedua, lalu *decreaseTeam2* untuk *ImageButton* ketiga dan *increaseTeam2* untuk *ImageButton* keempat.

Berikutnya tambahkan *TextView* untuk masing-masing layout relatif, letakkan diatas *ImageButton* dan gunakan atribut berikut ini

TextView attribute	Value
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	"0"

Gunakan *score_2* sebagai nilai atribut `android:id` untuk *TextView* kedua.

Kemudian tambahkan lagi *TextVew* untuk masing-masing layout relatif diatas setiap skor untuk melabelkan nama masing-masing tim. Gunakan atribut berikut ini

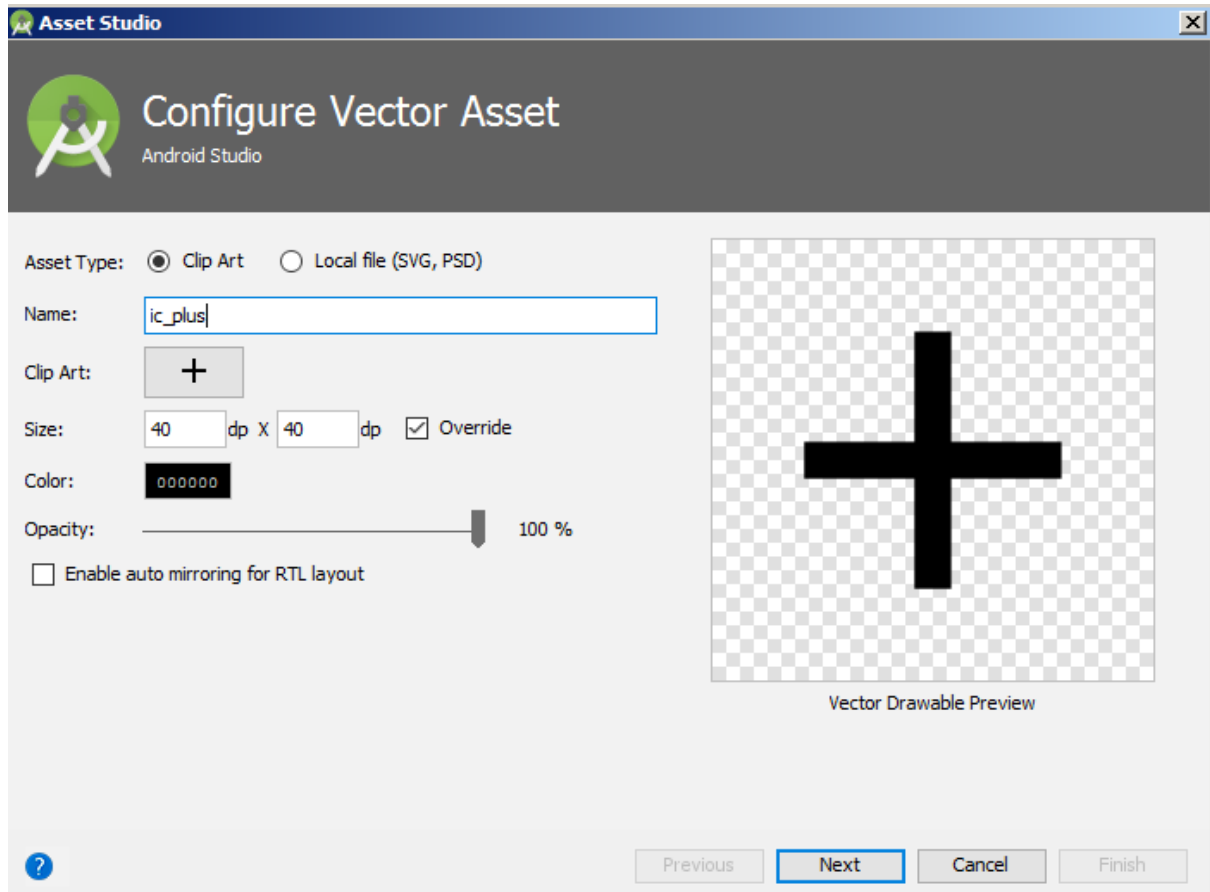
TextView attribute	Value
android:layout_width	"wrap_content"
android:layout_height	"wrap_content"
android:layout_alignParentTop	"true"
android:layout_centerHorizontal	"true"
android:text	"Team 1"

Kemudian gunakan *Team2* untuk atribut `android:text` pada *TextView* berikutnya.

Berikutnya tambahkan aset vektor.

Anda akan menggunakan ikon dari *vector asset studio* untuk komponen skor *ImageButton* yang telah dideklarasikan.

- Pertama-tama pilih **File > New > Vector Asset** untuk membuka studio tersebut.
- Kemudian klik ikon untuk membuka daftar file ikon yang tersedia. Lalu klik *Clip Art* dan pilih kategori *content*.
- Pilih ikon *add* dan klik OK.
- Kemudian ganti nama file menjadi *ic_plus* dan cek *Override checkbox* yang berada disebelah ukuran opsi.
- Ubah ukuran ikon menjadi 40dp x 40dp.
- Klik *Next* lalu *Finish* (lihat gambar dibawah).
- Ulangi langkah-langkah diatas untuk membuat ikon *remove* dan beri nama *ic_minus*.



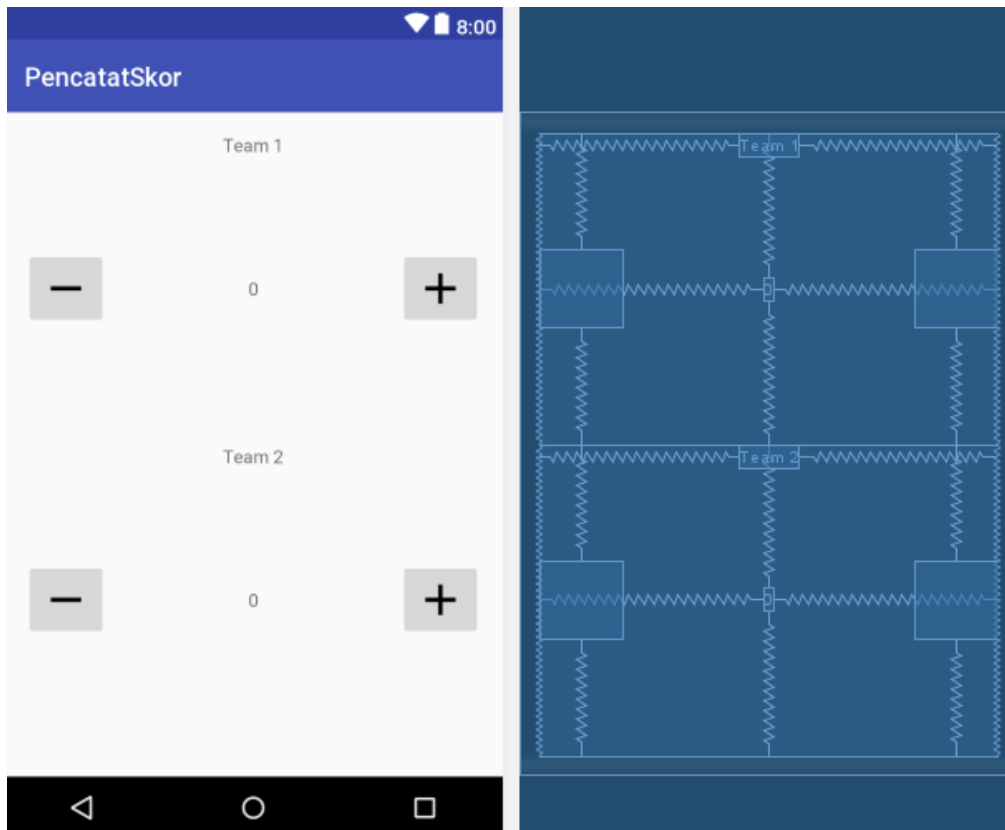
Berikutnya anda sudah dapat menggunakan ikon tersebut pada *ImageButton* dengan menambahkan atribut berikut pada bagian kiri layout.

```
android:src="@drawable/ic_minus"  
android:contentDescription="Minus Button"
```

Berikutnya lakukan hal yang sama untuk *ImageButton* pada bagian kanan layout dengan atribut berikut

```
android:src="@drawable/ic_plus"  
android:contentDescription="Plus Button"
```

Kemudian aturlah agar seluruh *ImageButton* dan *TextView* sesuai dengan layout berikut ini



Kemudian jadikan seluruh string pada kode xml tersebut menjadi variabel global dengan mendefinisikannya di file strings.xml seperti berikut ini

```

<resources>
    <string name="app_name">PencatatSkor</string>
    <string name="team_1">Team 1</string>
    <string name="team_2">Team 2</string>
    <string name="minus_button_description">minus_button_description</string>
    <string name="plus_button_description">plus_button_description</string>
    <string name="initial_count">0</string>
</resources>

```

Hasil kode xml anda akan menjadi seperti berikut

```

<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:padding="16dp"
    tools:context="com.example.android.scorekeeper.MainActivity">

    <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_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:src="@drawable/ic_minus"  
  android:contentDescription=  
    "@string/minus_button_description" />
```

```
<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:src="@drawable/ic_plus"  
  android:contentDescription=  
    "@string/plus_button_description" />
```

```
</RelativeLayout>
```

```
<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:src="@drawable/ic_minus"  
  android:contentDescription=  
    "@string/minus_button_description" />
```

```
<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:src="@drawable/ic_plus"
    android:contentDescription=
        "@string/plus_button_description" />
</RelativeLayout>
</LinearLayout>

```

Untuk *tracking* skor secara berkala, aplikasi memerlukan variabel untuk menyimpan skor sementara dan referensi terhadap `TextView` terkait pada file `MainActivity`.

Untuk itu lakukan langkah-langkah berikut ini:

Bukalah file `MainActivity.java`. Buatlah dua variabel integer sebagai representasi skor untuk setiap tim. Kemudian buat dua variabel `TextView` sebagai representasi `TextView` di layout. Kode java anda akan menjadi seperti ini

```

public class MainActivity extends AppCompatActivity {

    // Variables for holding the score
    private int mScore1;
    private int mScore2;

    //Variables for holding references of TextView
    private TextView mScoreText1;
    private TextView mScoreText2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Associate the TextViews with score
        mScoreText1 = (TextView) findViewById(R.id.score_1);
        mScoreText2 = (TextView) findViewById(R.id.score_2);
    }
}

```

Berikutnya anda harus menambahkan atribut `android:onClick` pada setiap `ImageButton` dan membuat 2 buah metode untuk mengelola klik `ImageButton` tersebut. Tujuannya adalah jika `ImageButton` sebelah kiri layout diklik maka skor akan berkurang dan akan bertambah saat user mengklik `ImageButton` yang disebelah kanan.

Buka kembali file xml dan tambahkan atribut android:onClick untuk masing-masing ImageButton dan set nilai berikut untuk ImageButton yang berada pada sisi kiri layout

```
android:onClick="decreaseScore"
```

Untuk ImageButton yang berada pada sisi kanan layout, set nilainya menjadi

```
android:onClick="increaseScore"
```

Pada file xml tersebut, anda akan melihat garisbawah berwarna merah pada nilai *decreaseScore* karena metode tersebut belum dibuat pada file java. Untuk membuat pada file java, arahkan kursor mouse anda ke nilai tersebut kemudian klik Alt + Enter, kemudian pilih **Create 'decreaseScore(view)' in 'MainActivity'**. Lakukan hal yang sama untuk membuat metode *increaseScore* pada file MainActivity.java.

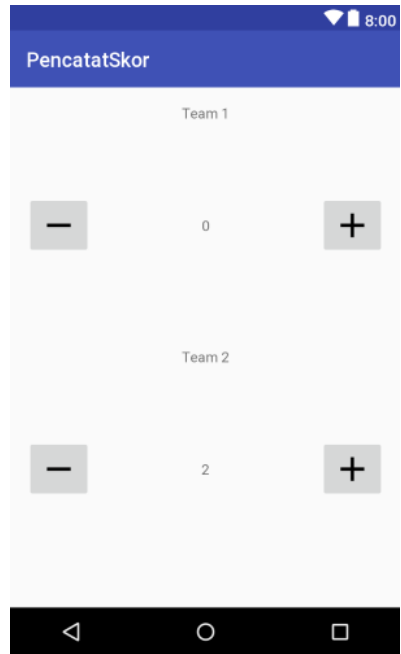
Isi metode tersebut menggunakan kode seperti berikut ini

```
/**
 * 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));
    }
}

/**
 * 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));
}
```

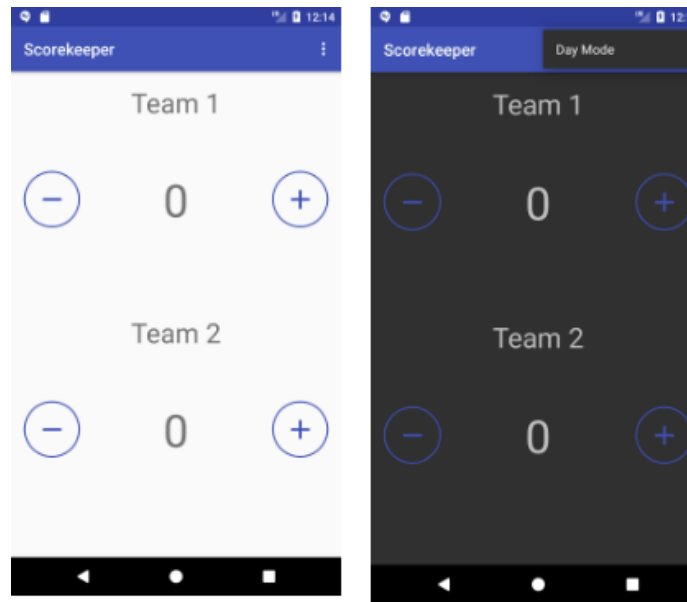
Coba jalankan aplikasi Pencatat Skor pada *smartphone* anda kemudian klik tanda + sebanyak 2 kali untuk team 2, maka anda akan mendapatkan hasil sebagai berikut



Jika anda perhatikan maka aplikasi tersebut belum nampak menarik. Oleh karena coba anda berlatih untuk membuat aplikasi tersebut menjadi lebih menarik menggunakan desain tematik.

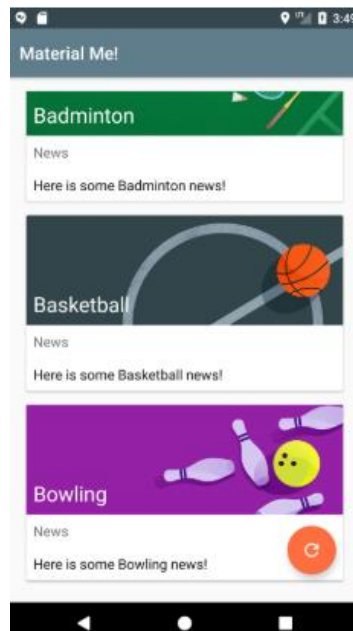
Tugas 1: Coba modifikasi aplikasi agar menjadi seperti layout berikut. (Anda dapat mengikuti tutorial di

<https://codelabs.developers.google.com/codelabs/android-training-drawables-styles-and-themes/index.html?index=..%2F..%2Fandroid-training#1.>)



Tugas 2: Coba buat halaman utama sebelum masuk pada layout Pencatat Skor. Halaman utama akan berbentuk seperti berikut. Gunakan intent implisit agar user dapat berpindah dari halaman ini kepada halaman Pencatat Skor. (Untuk membuat halaman tersebut anda dapat mengikuti tutorial di

<https://codelabs.developers.google.com/codelabs/android-training-drawables-styles-and-themes/index.html?index=..%2F..%2Fandroid-training#1.>)



Tugas 3: Coba ubah layout sesuai selera anda sendiri. Contoh-contoh layout skor pertandingan adalah sebagai berikut.



Simpulan

Sesi ini telah menjelaskan konsep dan latihan untuk membuat aplikasi pencatat skor pertandingan menjadi lebih intuitif dan menarik. Konsep yang dibahas mencakup desain material, mode dan tema. Tentunya, ilustrasi aplikasi tersebut hanyalah awal untuk membuat aplikasi dengan *user experience* yang baik. Mulai sekarang anda dapat mencoba membuat aplikasi-aplikasi sederhana lainnya. Pada sesi berikutnya anda akan mempelajari komponen-komponen aplikasi lainnya *services*, *broadcast receiver* dan *content provider*.

Referensi

[1] <https://developer.android.com/guide/topics/ui/look-and-feel>