

Mengenal jenis Layout pada Android

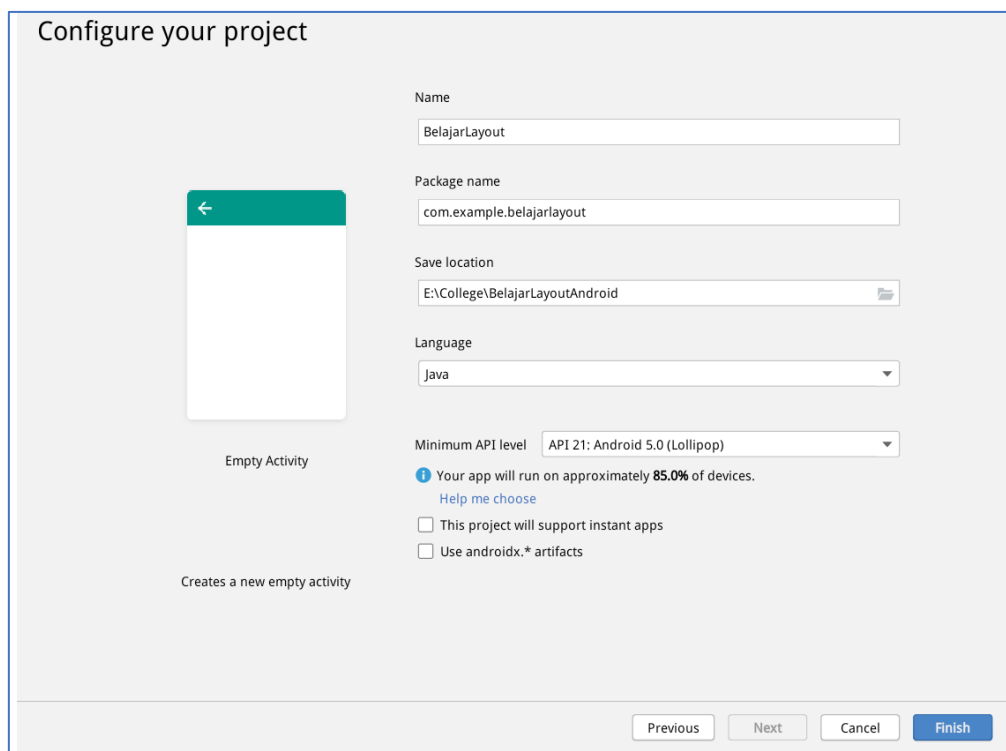
Android menyediakan dukungan penuh untuk pengembangan aplikasi berbasis UI. Android menyediakan berbagai widget yang dapat digunakan oleh pengembang aplikasi untuk membuat tata letak dan tampilan antarmuka yang diinginkan. Elemen tata letak ini dapat dibuat dengan menggunakan bahasa pemrograman secara langsung maupun dengan melalui file layout XML.

Pada android, layout berbasis-XML adalah file yang mendefinisikan berbagai widget yang akan digunakan di UI dan hubungan antara widget serta penampungnya. Android memperlakukan file layout sebagai resource. Oleh karena itu, tata letak disimpan dalam folder reslayout. File layout bertindak sebagai masukan ke Android Asset Packaging Tool (AAPT), yang membentuk file R.java untuk semua sumber.

Android menyediakan model layout standard (viewgroups) yang dapat digunakan dalam aplikasi Android.

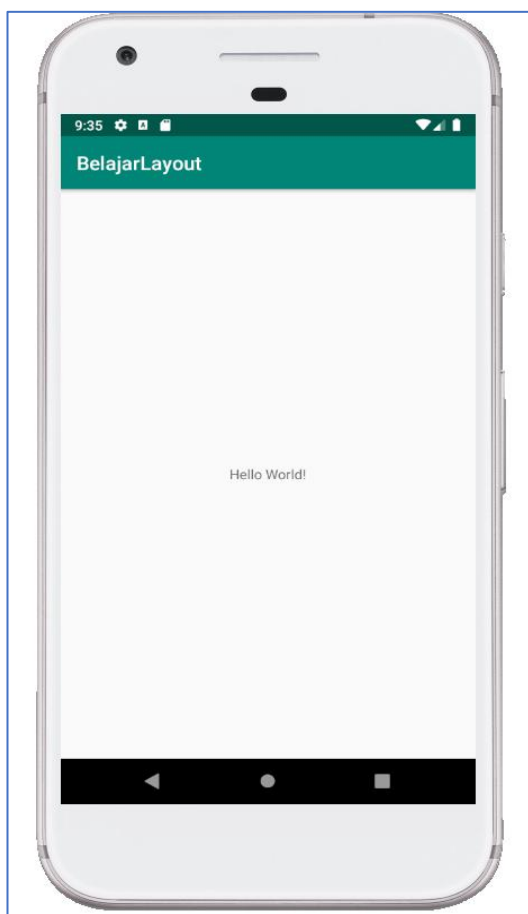
1. AbsoluteLayout
2. FrameLayout
3. LinearLayout
4. RelativeLayout
5. TableLayout

Sebelum memulai mengenal berbagai jenis layout diatas, kita akan membuat sebuah aplikasi “Hello World” sederhana. Buka Android Studio anda, klik File → New → New Project → beri nama projek “**BelajarLayout**”.




Selanjutnya buka file `activity_main.xml` → pilih tab Text. Ketika kita membuat sebuah proyek, maka secara *default* Android telah menyediakan kode xml untuk Hello World seperti pada contoh di bawah ini. Silahkan disesuaikan dengan kode xml yang tertera pada PC anda.

```
activity_main.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.constraint.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".MainActivity">
9
10     <TextView
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:text="Hello World!"
14         app:layout_constraintBottom_toBottomOf="parent"
15         app:layout_constraintLeft_toLeftOf="parent"
16         app:layout_constraintRight_toRightOf="parent"
17         app:layout_constraintTop_toTopOf="parent" />
18
19 </android.support.constraint.ConstraintLayout>
```



Untuk menjalankan aplikasi pada emulator, klik pada AVD manager pada posisi atas dengan

gambar icon  → buat sebuah emulator baru dengan Create Virtual Device → pilih model device → pilih API level (pilih seperlunya karena akan memakan waktu untuk mendownload data) →

setelah itu, klik tombol Run  → pilih emulator yang telah dibuat tadi.

Jika berhasil, maka Aplikasi Hello World pertama anda akan muncul pada emulator seperti pada contoh ini.

Sekarang kita akan mulai mempelajari beberapa jenis layout yang telah dibahas di atas.

Absolute Layout

Layout ini disediakan oleh Google sejak API versi 1 untuk desain tampilan aplikasi yang memungkinkan kita untuk mengatur posisi objek secara horizontal dan vertikal sesuai keinginan. Akan tetapi, semenjak adanya perkembangan API android versi 3, model layout ini tidak lagi digunakan. Mari kita lihat contoh pembuatan tampilan UI lainnya dengan menggunakan kode xml.

Frame Layout

Layout model ini digunakan ketika kita ingin menampilkan sebuah objek saja pada tampilan aplikasi. Tentu saja ini tidak berarti bahwa kita hanya bisa membentuk sebuah objek dengan model ini, tapi ketika kita membuat banyak objek, maka posisi objek tersebut pada layar akan menimpa satu sama lain. FrameLayout sangat bermanfaat ketika kita ingin membuat aplikasi yang memiliki animasi.

Tambahkan sebuah file .xml baru. Caranya adalah buka folder **res** → klik kanan pada folder **layout** → pilih **Layout resource file** → beri nama **activity_frame_layout** → Create → kemudian ketikkan kode xml di bawah ini.



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <FrameLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent">
6
7     <Button
8         android:id="@+id/username"
9         android:text="Klik Disini\n\nHalaman 1"
10        android:layout_width="match_parent"
11        android:layout_height="match_parent"
12        android:textSize="30dp"
13        android:onClick="onClick"
14        android:gravity="center"/>
15
16    <TextView
17        android:id="@+id/password"
18        android:text="Halaman 2"
19        android:layout_width="match_parent"
20        android:layout_height="match_parent"
21        android:textSize="30dp"
22        android:visibility="gone"
23        android:gravity="center"/>
24 </FrameLayout>
```

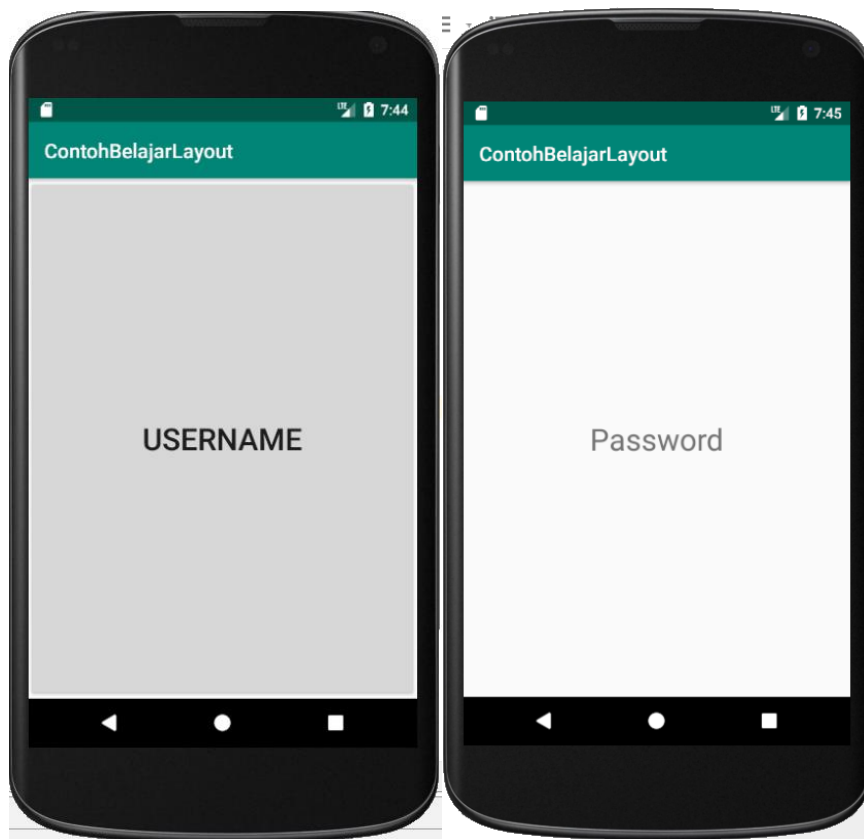
Kita membentuk sebuah tombol dan sebuah TextView yang menggunakan FrameLayout. Selanjutnya kita akan memberikan aksi kepada tombol tersebut, sehingga ketika di klik maka akan menampilkan TextView nya. Seperti yang terlihat pada text, bahwa kita memberikan kode "onClick" pada objek Button di atas.

Langkah yang harus dilakukan yaitu buat sebuah file java class baru. Buka folder java, klik kanan pada folder → New → Java Class → beri nama **FrameLayout** → create → ketik kode di bawah yaitu dengan menambahkan aksi onClick ketika tombol ditekan.

```
FrameLayout.java x
1 package com.example.belajarlayout;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5 import android.view.View;
6
7 public class FrameLayout extends AppCompatActivity {
8
9     protected void onCreate (Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_frame_layout);
12     }
13
14     public void onClick (View v){
15         View vUsername = findViewById(R.id.username);
16         vUsername.setVisibility(View.GONE);
17
18         View vPassword = findViewById(R.id.password);
19         vPassword.setVisibility(View.VISIBLE);
20     }
21 }
```

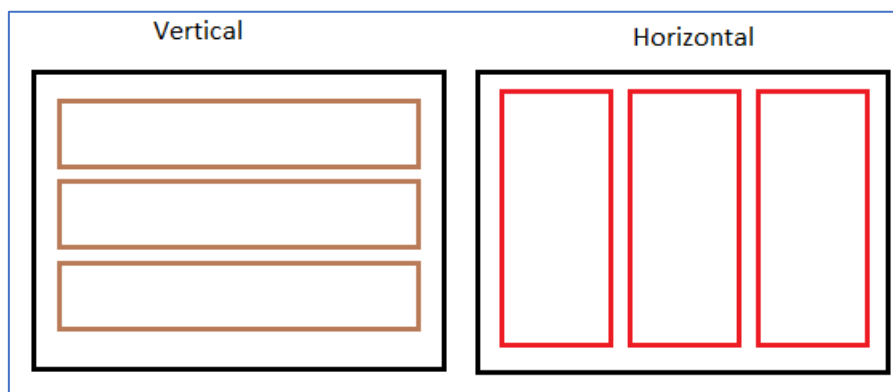
Jangan lupa ubah/tambahkan activity sesuai dengan nama class pada AndroidManifest.xml agar app dapat dijalankan pada emulator.

```
AndroidManifest.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.belajarlayout">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="BelajarLayout"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportRtl="true"
11        android:theme="@style/AppTheme">
12        <activity android:name=".FrameLayout">
13            <intent-filter>
14                <action android:name="android.intent.action.MAIN" />
15
16                <category android:name="android.intent.category.LAUNCHER" />
17            </intent-filter>
18        </activity>
19    </application>
20
21 </manifest>
```



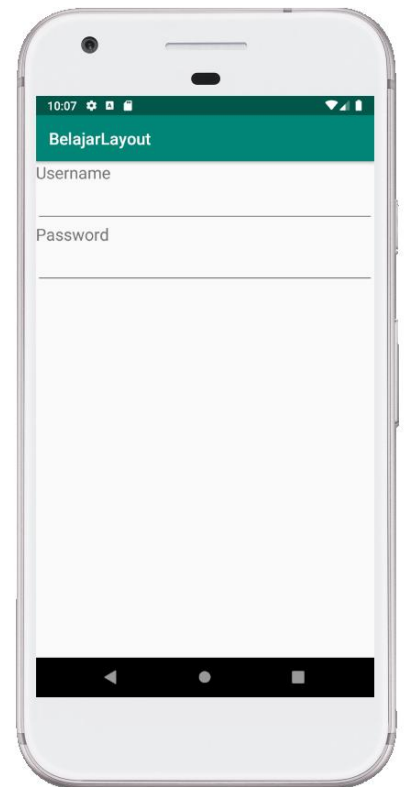
Linear Layout

Model LinearLayout membuat objek terletak pada satu baris atau kolom. Sehingga posisi objek tersebut dapat tersusun dengan rapi dari atas ke bawah. Model ini lah yang paling banyak digunakan untuk membuat tampilan antar muka pada aplikasi android.



Buat sebuah .xml baru → beri nama **activity_linear_layout** → ketik kode seperti di bawah. Anda dapat langsung melihat bentuk tampilan pada tab design tanpa perlu menjalankan aplikasi.

```
activity_linear_layout.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical">
7
8     <TextView
9         android:text="Username"
10        android:layout_width="wrap_content"
11        android:layout_height="wrap_content"
12        android:textSize="20dp"/>
13
14    <EditText
15        android:layout_width="match_parent"
16        android:layout_height="wrap_content"
17        android:textSize="20dp"/>
18
19    <TextView
20        android:text="Password"
21        android:layout_width="wrap_content"
22        android:layout_height="wrap_content"
23        android:textSize="20dp"/>
24
25    <EditText
26        android:layout_width="match_parent"
27        android:layout_height="wrap_content"
28        android:textSize="20dp"/>
29
30 </LinearLayout>
```



Kemudian buka tambahkan file java baru bernama "LinearLayout". Ketikkan kode java berikut ini. Setelah itu jangan lupa ubah **activity android:name** pada AndroidManifest.xml menjadi **LinearLayout**. Kemudian anda dapat mencoba untuk menjalankan aplikasi tersebut. Tampilan akan muncul seperti pada contoh di atas.

```
LinearLayout.java
1 package com.example.belajarlayout;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5
6 public class LinearLayout extends AppCompatActivity {
7     protected void onCreate (Bundle savedInstanceState) {
8         super.onCreate(savedInstanceState);
9         setContentView(R.layout.activity_linear_layout);
10    }
11 }
```

Relative Layout

Dengan menggunakan model ini, kita dapat menentukan posisi suatu objek terhadap objek lainnya yang masih berada dalam grup yang sama.

Kali ini kita akan mencoba membuat sebuah halaman login sederhana dengan menggunakan RelativeLayout. Buat sebuah .xml baru → beri nama **activity_relative_layout** → ketikkan kode program seperti pada gambar di bawah ini.



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent">
6
7     <TextView
8         android:id="@+id/txJudul"
9         android:text="Halaman Login"
10        android:layout_width="wrap_content"
11        android:layout_height="wrap_content"
12        android:textSize="30sp"
13        android:textStyle="bold"
14        android:layout_centerHorizontal="true"
15        android:layout_marginTop="5dp"
16        android:layout_marginBottom="20dp"/>
17
18     <TextView
19         android:id="@+id/txUsername"
20         android:layout_width="120dp"
21         android:layout_height="40dp"
22         android:layout_below="@+id/txJudul"
23         android:text="Nama User"
24         android:textSize="20sp" />
25
26     <EditText
27         android:id="@+id/username"
28         android:layout_width="match_parent"
29         android:layout_height="wrap_content"
30         android:layout_below="@+id/txJudul"
31         android:layout_toEndOf="@+id/txUsername"
32         android:textSize="20sp"
33         android:layout_marginBottom="10dp"/>
34
```

```
activity_relative_layout.xml
35 <TextView
36     android:id="@+id/txPass"
37     android:text="Kata Kunci"
38     android:layout_width="120dp"
39     android:layout_height="wrap_content"
40     android:layout_below="@+id/username"
41     android:textSize="20sp"/>
42
43 <EditText
44     android:id="@+id/pass"
45     android:layout_width="match_parent"
46     android:layout_height="wrap_content"
47     android:layout_below="@+id/username"
48     android:layout_toEndOf="@+id/txPass"
49     android:textSize="20sp"
50     android:layout_marginBottom="10dp"/>
51
52 <Button
53     android:id="@+id/btnLogin"
54     android:layout_width="120dp"
55     android:layout_height="wrap_content"
56     android:text="Masuk"
57     android:textSize="20sp"
58     android:layout_below="@+id/pass"
59     android:layout_marginStart="90dp"/>
60
61 <Button
62     android:id="@+id/btnCancel"
63     android:layout_width="120dp"
64     android:layout_height="wrap_content"
65     android:text="Batal"
66     android:textSize="20sp"
67     android:layout_below="@+id/pass"
68     android:layout_toEndOf="@+id/btnLogin"/>
69
70 </RelativeLayout>
```

Kemudian tambahkan file java baru bernama RelativeLayout. Lalu ketikkan kode java berikut ini.

```
RelativeLayout.java
1 package com.example.belajarlayout;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5
6 public class RelativeLayout extends AppCompatActivity {
7
8     protected void onCreate (Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10         setContentView(R.layout.activity_relative_layout);
11     }
12 }
```

Setelah selesai, maka jangan lupa untuk mengubah activity android:name pada AndroidManifest.xml menjadi RelativeLayout. Kemudian anda dapat mencoba untuk menjalankan aplikasi tersebut. Tampilan akan muncul seperti pada contoh di bawah ini.

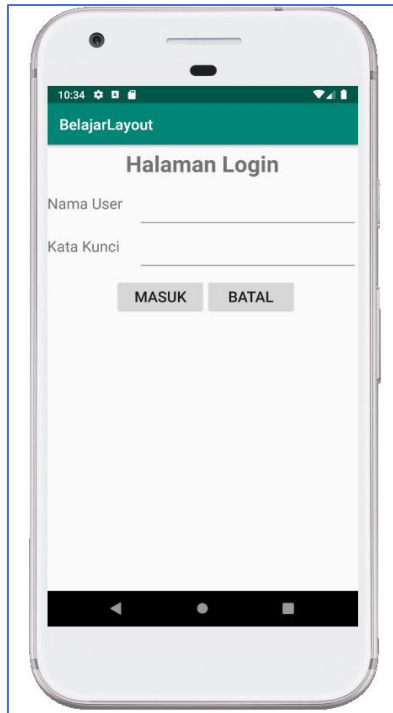


Table Layout

Table layout memungkinkan kita untuk menyusun tampilan halaman *user* kedalam bentuk baris dan kolom. Setiap baris dapat memiliki lebih dari sebuah kolom dan sebuah kolom dapat menyimpan sebuah objek pada Android. Untuk membuat tampilan table, android menyediakan tag berupa `<TableLayout>` yang berisi susunan `<TableRow>`.

Silahkan buat sebuah file xml baru bernama `activity_table_layout`. Lalu tambahkan kode xml seperti pada contoh di bawah ini.

Baris 1	
Baris 2 Kolom 1	Baris 2 Kolom 2
Baris 3 Kolom 1	Baris 3 Kolom 2
Baris 4 Kolom 1	Baris 4 Kolom 2

```
activity_table_layout.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <TableLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent">
6
7      <TableRow
8          android:layout_width="match_parent"
9          android:layout_height="match_parent"
10         android:gravity="center">
11
12         <TextView
13             android:layout_width="match_parent"
14             android:layout_height="wrap_content"
15             android:layout_column="1"
16             android:text="Halaman Rating"
17             android:textSize="20sp"
18             android:textStyle="bold" />
19
20         </TableRow>
21
22         <TableRow>
23
24             <TextView
25                 android:text="Judul Film"
26                 android:layout_width="100dp"
27                 android:layout_height="wrap_content"
28                 android:layout_column="1" />
29
30             <EditText
31                 android:width="200dp"
32                 android:layout_width="wrap_content"
33                 android:layout_height="wrap_content"
34                 android:layout_column="2" />
35         </TableRow>
```

```
activity_table_layout.xml
37 <TableRow>
38
39     <TextView
40         android:text="Review"
41         android:layout_width="100dp"
42         android:layout_height="wrap_content"
43         android:layout_column="1" />
44
45     <EditText
46         android:width="200dp"
47         android:layout_width="wrap_content"
48         android:layout_height="wrap_content"
49         android:layout_column="2" />
50 </TableRow>
51
52 <TableRow
53     android:layout_width="match_parent"
54     android:layout_height="match_parent">
55
56     <RatingBar
57         android:layout_width="match_parent"
58         android:layout_height="wrap_content"
59         android:id="@+id/ratingBar"
60         android:layout_column="2" />
61 </TableRow>
62
63 <TableRow
64     android:layout_width="match_parent"
65     android:layout_height="match_parent">
66
67     <Button
68         android:id="@+id/button"
69         android:layout_width="wrap_content"
70         android:layout_height="wrap_content"
71         android:layout_column="2"
72         android:text="Submit" />
73 </TableRow>
74
75 </TableLayout>
```

Setelah selesai menyusun kode xml, silahkan tambahkan kode Java berikut. Buat sebuah file java baru bernama **"TableLayout"**.

```
TableLayout.java
1 package com.example.belajarlayout;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5
6 public class TableLayout extends AppCompatActivity {
7
8     protected void onCreate (Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10         setContentView(R.layout.activity_table_layout);
11     }
12 }
```

Kemudian jangan lupa untuk mengubah `activity android:name` pada `AndroidManifest.xml` menjadi **TableLayout**. Lalu jalankan aplikasi pada emulator android anda. Jika berhasil, maka emulator akan menampilkan aplikasi seperti pada contoh di bawah ini.

