

live coding

#1

android

Arif Akbarul Huda, S.Si



cara cepat menguasai
pemrograman android





Penulis



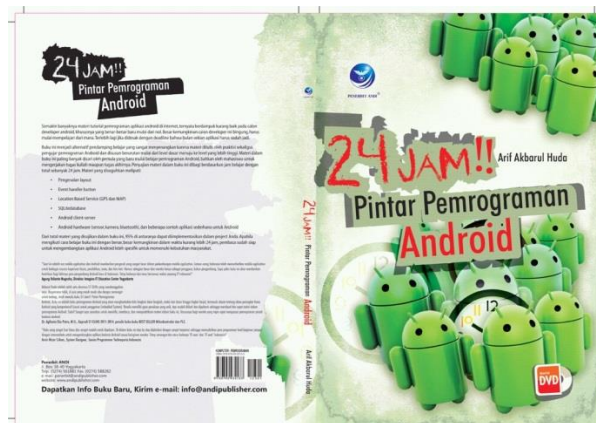
Penulis adalah seorang *programmer freelance* yang aktif dalam bidang mobile application khususnya android. Setelah menerbitkan buku “24 JAM!! Pintar Pemrograman Android” , saat ini penulis yang bernama lengkap Arif Akbarul Huda sedang menyelesaikan studi di Pascasarjana Magister Teknik Informatik UGM.

Dalam kesehariaanya, penulis yang kerap disapa omayib ini berkantor di Imagine IT Education Center (IITEC) Yogyakarta sebagai Manajer. IITEC membuka kelas-kelas pemrograman seperti **Android Academy**, **Web Developer Class**, **Graphic Design Class**, **Multimedia Class** dan lainnya yang dapat Anda lihat di www.imagineitcenter.com . Dalam perjalanan karirnya, omayib kerap kali diundang dalam pelatihan, seminar-seminar atau workshop seputar I.T.

Selain itu, omayib adalah pemilik website www.omyib.com dan www.indocode.net sekaligus sebagai CEO nya. Melalui media ini, omayib sering berbagi ilmu seputar pemrograman Android.

Ebook ini bersifat serial, akan terbit seri **LIVE CODING ANDROID #2, #3** dan seterusnya. saya persembahkan untuk teman-teman yang ingin mendalami pemrograman android. Silakan membagikan ebook ini sebanyak-banyaknya, bisa dijadikan sebagai panduan praktikum atau sekedar belajar dirumah. **Dilarang mengubah isi dan tidak untuk di komersilkan.**

Bagi anda yang mencari referensi buku, berikut ini buku yang sangat saya rekomendasikan sebagai panduan belajar.



Info detil buku www.omyib.com/buku



www.indocode.net



Perkenalkan Kami

Indocode.net merupakan sebuah media penghubung antara programmer dengan klien. **Klien** disini adalah seseorang yang memerlukan jasa programmer untuk mebuat sebuah asoftware aplikasi.

Programmer freelance adalah seorang programer professional yang siap membantu Anda mengerjakan sebuah project pemrograman seperti pembuatan website, aplikasi android, aplikasi dekstop atau lainnya.

Komunikasi antara programmer dan klien dapat dilakukan melalui chating/email/sms/BBM. Seorang programmer berhak menerima upah dari klien sekligus berkewajiban menyelesaikan project yang telah disepakati.



Indocode.net mengklasifikasikan project menjadi 3 yaitu

- Personal
- Instansi
- Product

Saat ini, indocode menyediakan jasa membangun aplikasi sebagai berikut

- Android Apps
- Flash Apps
- Sistem Informasi Web Based (PHP & MySQL)
- Sistem Aplikasi pengelolaan UMKM

Hubungi omayib@gmail.com
atau klik www.indocode.net

Bekerja sama dengan indocode.net caranya sangat mudah. Cukup melewati 5 langkah, maka aplikasi siap di tangan Anda.



1. Konsultasi & Dealing

Pada bagian ini, indocode.net memberi kesempatan kepada Anda untuk berkonsultasi mengenai apa saja yang ingin di kembangkan. Konsultasi dapat melalui [halaman ini](#).

2. Pembayaran 1

Pada tahap ini, Anda perlu melakukan pembayaran 1 atau D.P. agar project bisa segera di develop. Besar pembayaran 1 berkisar antara 30-50% dari seluruh biaya yang disepakati.

3. Developing & Testing

Tahap ini merupakan tahap pembuatan aplikasi (developing). Lama waktu pengerjaan sesuai dengan rentang waktu yang telah disepakati pada tahap 1. Programmer Kami akan memberikan progres atau update secara berkala (misal 1 minggu sekali). Setelah project jadi hingga 90%, project akan dikirimkan kepada Anda. Programmer akan melakukan revisi atau merelease versi terbaru jika ada hal-hal yang belum sempurna.

4. Pembayaran 2

Setelah melalui proses penyempurnaan aplikasi (melibatkan Anda dalam testing), maka aplikasi akan di rilis sebagai fersi FINALxx. Seblum versi final diserahkan sepenuhnya, Anda harus melakukan pelunasan terlebih dahulu.

5. Finish

Pengerjaan project selesai, minimal 1 bulan setelah akhir pengerjaan ini, project akan di publish di www.indocode.net sebagai berkedudukan sebagai klien Kami.

Hubungi omayib@gmail.com
atau klik www.indocode.net



DAFTAR ISI

Bagian 1 | Pengenalan

Instalasi ADT

HelloWordApp

Pengenalan environment

Pengenalan Layout

Bagian 2 | Basic

Menggunakan Button

Memanggil halaman lain

Siklus hidup aplikasi

Bagian 3 | Multimedia

Menampilkan gambar

Memanggil audio

Bagian 4 | Android Academy

Bagian 5 | User Interface

Membuat gradient

Membuat custom button

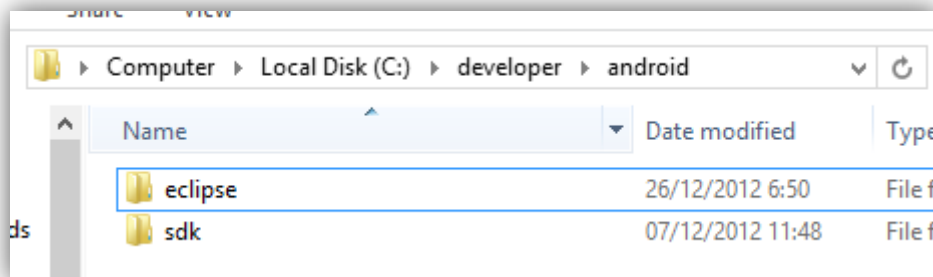
Membuat custom editteks

Bagian 1 | Pengenalan

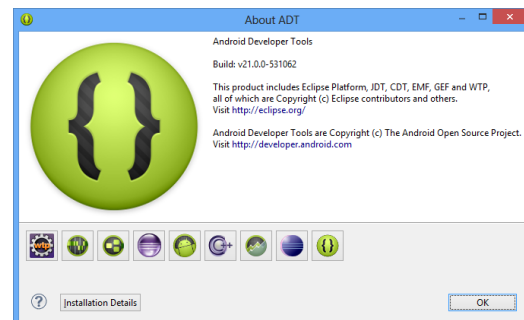
1.1. Instalasi ADT

Sejak 3 Desember 2013, situs www.developer.android.com resmi merilis paket ADT 21.x.x (Android Development Tool) berukuran \pm 300Mb. ADT berisi SDK (Standart Development Kit) Android dan eclipse yang sudah di set khusus untuk mengembangkan aplikasi android. Jika Anda sudah mempunyai ADT, maka bisa langsung dipakai untuk membuat aplikasi tanpa harus melakukan setting seperti versi sebelumnya. Berikut ini langkah menjalankan ADT

1. Lakukan ekstrak file **adt-bundle-windows-x86.zip** ke C:/developer/android (Gambar 1.1)
2. Masuk ke folder **C:/developer/android/eclipse** jalankan eclipse.exe



Gambar 1.1. Letak hasil ekstrak ADT



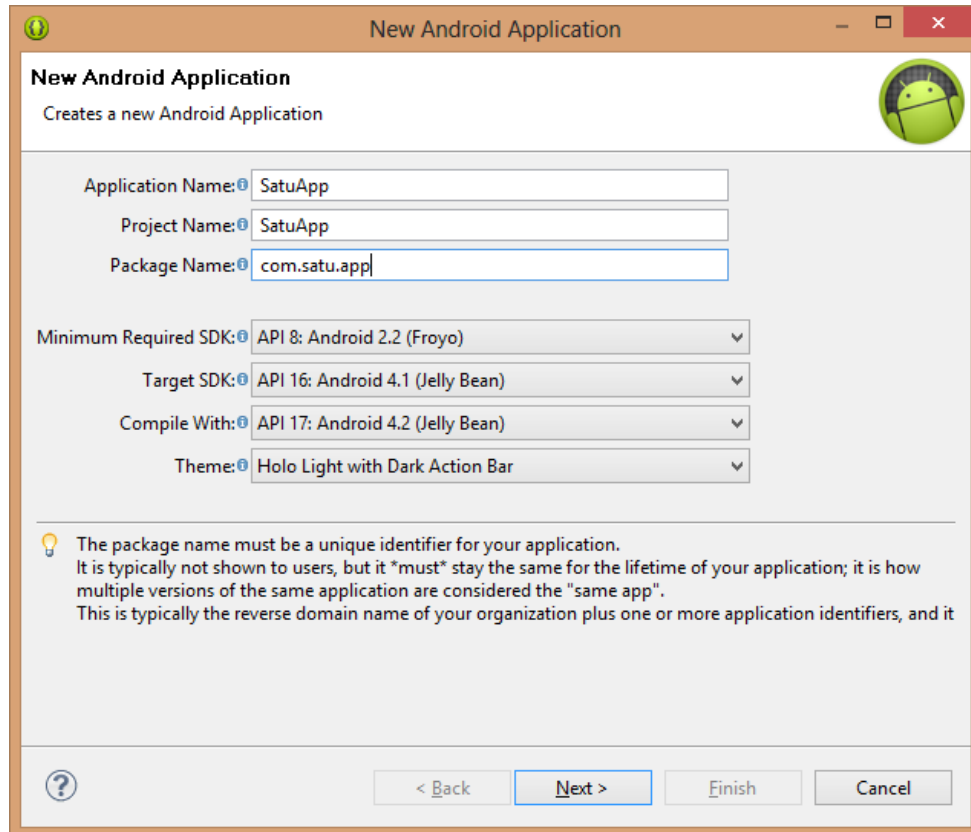
Gambar 1.2. Tampilan saat eclipse ADT dijalankan



1.2. Membuat project baru

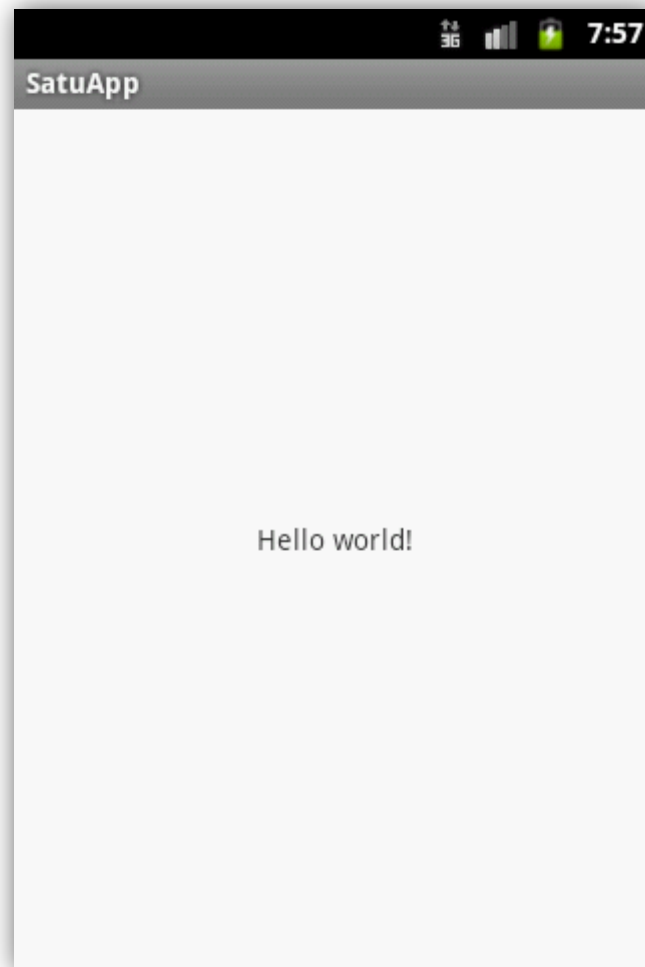
Nah saatnya membuat project baru. Setiap membuat aplikasi baru maka kita akan membuat project baru. Dalam satu project bisa terdiri atas banyak package.

1. Klik **file > new > Android Application Project**. Anda akan dihadapkan dengan dialog box seperti gambar 1.3.



Gambar 1.3. Membuat project baru

2. Isilah field **Application Name**, **Project Name**, **Package Name** dan lainnya persis dengan Gambar 1.3. lalu klik **Next** sebanyak empat kali diakhiri dengan **Finish**.
3. Jalankan aplikasi dengan cara klik kanan **project > Run As > Android Application**

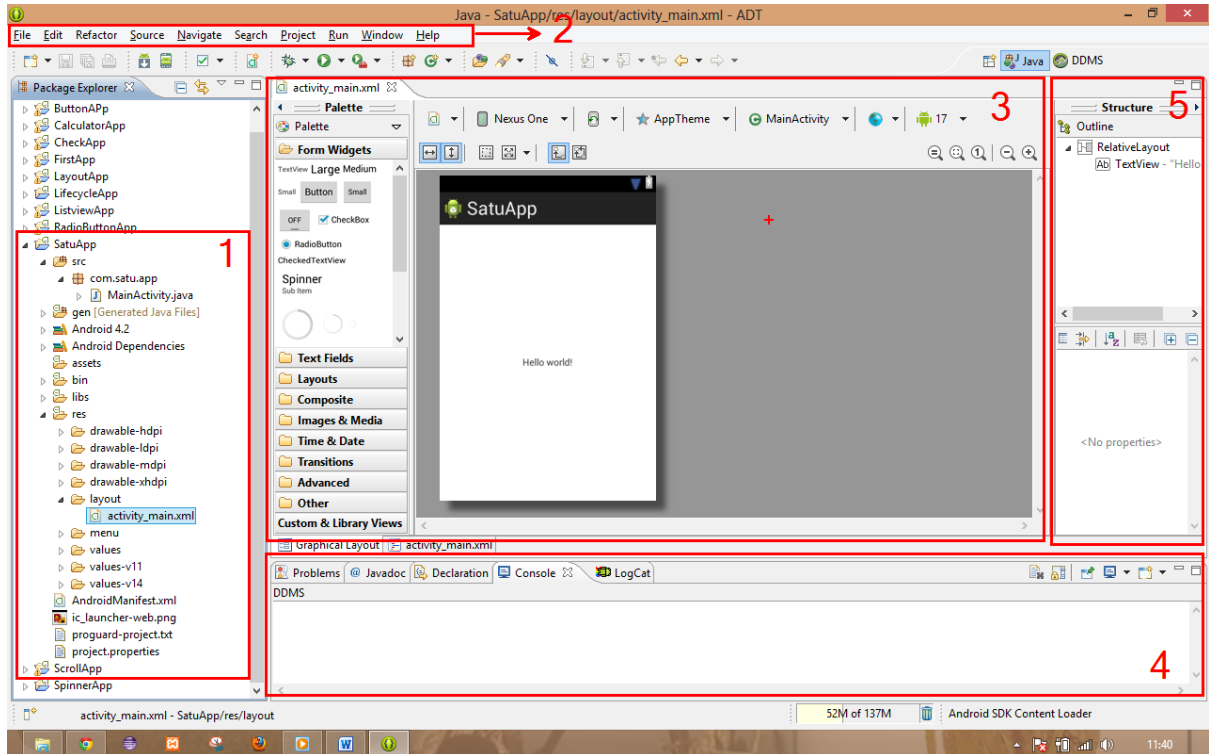


Gambar 1.4. Aplikasi baru helloworld



1.3. Pengenalan environment

Perhatikan kembali editor eclipse yang ada di depan Anda, maka Anda akan menjumpai tampilan default seperti Gambar 1.4.



Gambar 1.4. Tampilan default eclipse ADT

Minimal ada 5 bagian pada Eclipse ADT yang perlu Anda kenali, karena untuk mengembangkan sebuah aplikasi nantinya akan banyak berinteraksi dengan bagian-bagian ini.

1. Package Explorer
2. Menu Bar
3. Stage
4. Other View
5. Properties Panel

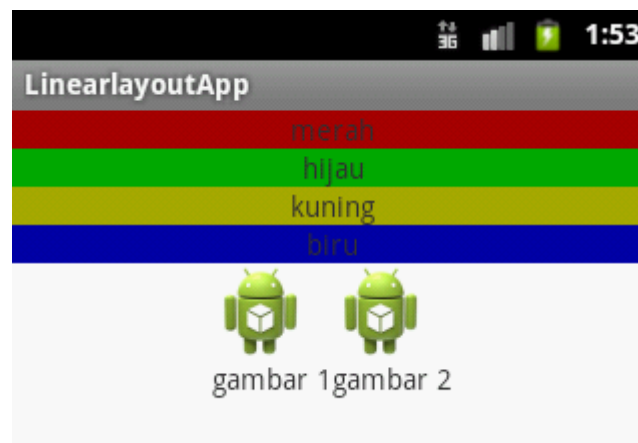
3.1. Pengenalan Layout

3.1.1. Linearlayout

Poin penting!

- | | |
|--|--|
| <ul style="list-style-type: none"> ▪ LinearLayout ▪ setOrientation ▪ setGravity | <ul style="list-style-type: none"> ▪ Layout Width / Layout Height ▪ setId ▪ setText |
|--|--|

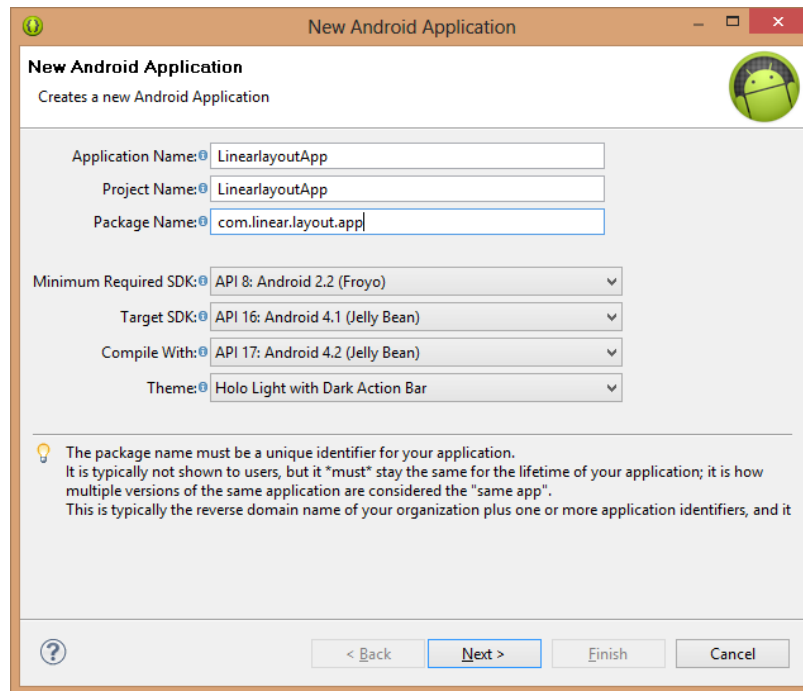
Bagian ini kita akan memahami mengenai LinearLayout. Setiap layout dapat disisipkan komponen atau disebut child. LinearLayout adalah salah satu dari jenis layout yang mempunyai satu kolom saja atau satu baris saja. Hasil akhir bagian ini tampak seperti Gambar 1.5.



Gambar 1.5. Tampilan aplikasi LinearLayoutApp

1. Siapkan project baru dengan parameter seperti Gambar 1.6. berikut

Application Name	LinearLayoutApp
Project Name	LinearLayoutApp
Package Name	com.linear.layout.app



Gambar 1.6. membuat aplikasi LinearLayoutApp

2. Tambahkan code berikut pada activity_main.xml

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     android:layout_width="match_parent"
4.     android:layout_height="match_parent"
5.     android:orientation="vertical" >
6.
7.     <LinearLayout
8.         android:layout_width="match_parent"
9.         android:layout_height="wrap_content"
10.        android:orientation="vertical" >
11.
12.        <TextView
13.            android:layout_width="match_parent"
14.            android:layout_height="wrap_content"
15.            android:background="#aa0000"
16.            android:gravity="center_horizontal"
17.            android:text="merah" />
18.
19.        <TextView
20.            android:layout_width="match_parent"
21.            android:layout_height="wrap_content"
22.            android:background="#00aa00"
23.            android:gravity="center_horizontal"
24.            android:text="hijau" />
25.
26.        <TextView
27.            android:layout_width="match_parent"
28.            android:layout_height="wrap_content"
29.            android:background="#aaaa00"
30.            android:gravity="center_horizontal"

```

```

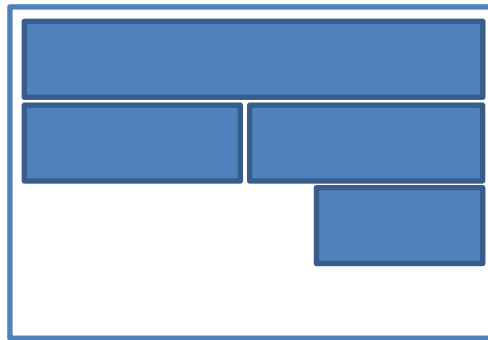
31.         android:text="kuning" />
32.
33.         <TextView
34.             android:layout_width="match_parent"
35.             android:layout_height="wrap_content"
36.             android:background="#0000aa"
37.             android:gravity="center_horizontal"
38.             android:text="biru" />
39.     </LinearLayout>
40.
41.     <LinearLayout
42.         android:layout_width="match_parent"
43.         android:layout_height="wrap_content"
44.         android:gravity="center_horizontal" >
45.
46.         <LinearLayout
47.             android:layout_width="wrap_content"
48.             android:layout_height="wrap_content"
49.             android:orientation="vertical" >
50.
51.             <ImageView
52.                 android:id="@+id/ImageView01"
53.                 android:layout_width="wrap_content"
54.                 android:layout_height="wrap_content"
55.                 android:src="@drawable/ic_launcher" />
56.
57.             <TextView
58.                 android:id="@+id/TextView01"
59.                 android:layout_width="wrap_content"
60.                 android:layout_height="wrap_content"
61.                 android:text="gambar 1" />
62.         </LinearLayout>
63.
64.         <LinearLayout
65.             android:layout_width="wrap_content"
66.             android:layout_height="wrap_content"
67.             android:orientation="vertical" >
68.
69.             <ImageView
70.                 android:id="@+id/imageView1"
71.                 android:layout_width="wrap_content"
72.                 android:layout_height="wrap_content"
73.                 android:src="@drawable/ic_launcher" />
74.
75.             <TextView
76.                 android:id="@+id/textView1"
77.                 android:layout_width="wrap_content"
78.                 android:layout_height="wrap_content"
79.                 android:text="gambar 2" />
80.
81.         </LinearLayout>
82.     </LinearLayout>
83.
84.
85. </LinearLayout>

```

3. Selesai, jalankan aplikasinya

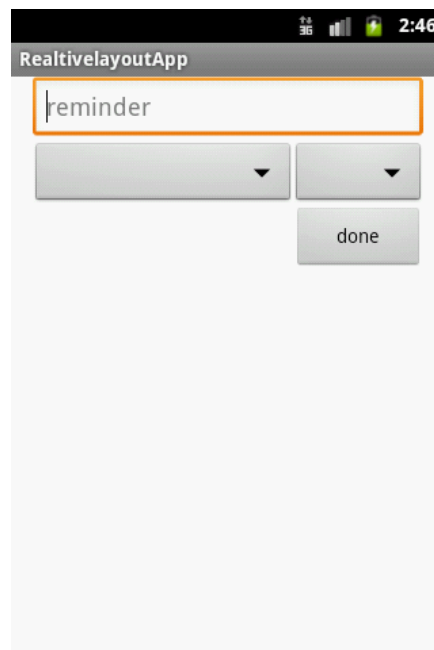
3.1.2. RelativeLayout

Dengan `RelativeLayout`, posisi setiap child bisa diset lebih spesifik terhadap child lainnya. Misal, posisi button berada di sebelah kanan bawah spinner, sedangkan spinner posisinya berada dibawah `EditText`. Sederhananya, setiap child posisinya relatif terhadap child-child lainnya. Ilustrasinya seperti gambar 1.7.



Gambar 1.7. Ilustrasi penggunaan RelativeLayout

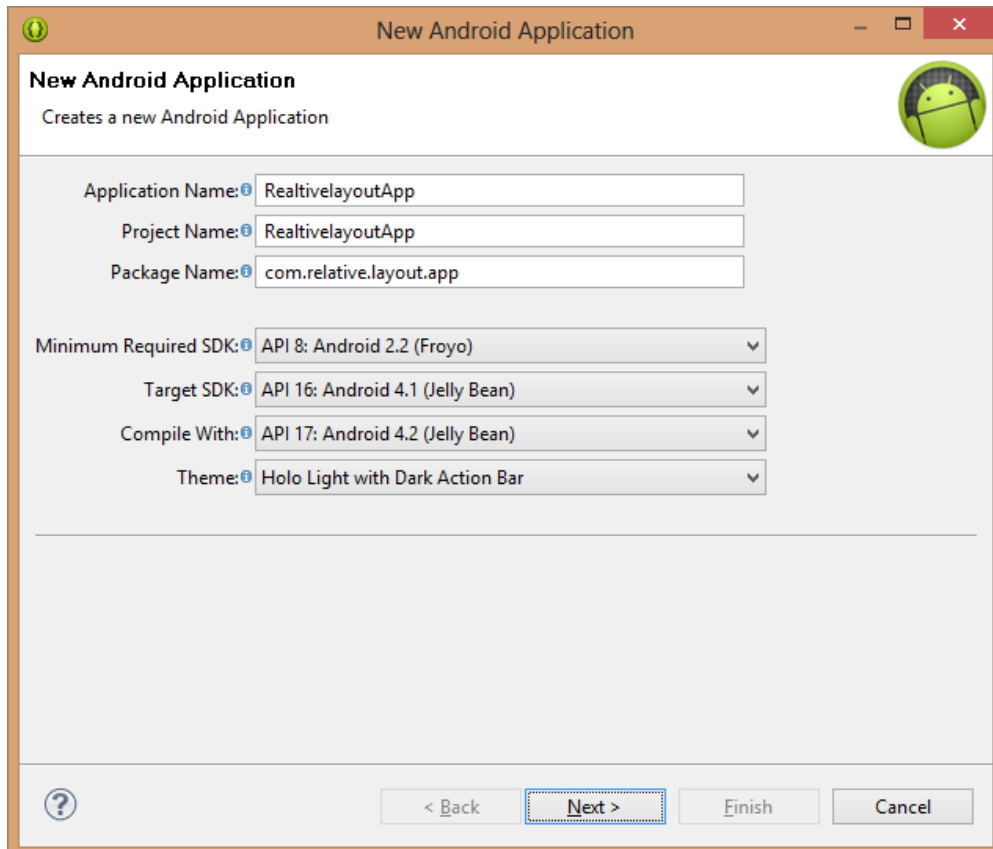
Sebagai pandangan, hasil akhir sesi ini tampak seperti pada Gambar 1.8. Mari kita praktikkan.



Gambar 1.8. Hasil akhir aplikasi RelativeLayoutApp

1. Siapkan project baru dengan parameter sebagai berikut seperti Gambar 1.9

Application Name	RelativeLayoutApp
Project Name	RelativeLayoutApp
Package Name	com.relative.layout.app



Gambar 1.9. Membuat aplikasi RelativeLayoutApp

2. Ubahlah activity_main.xml seperti berikut

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <RelativeLayout
3.     xmlns:android="http://schemas.android.com/apk/res/android"
4.     android:layout_width="fill_parent"
5.     android:layout_height="fill_parent"
6.     android:paddingLeft="16dp"
7.     android:paddingRight="16dp" >
8.     <EditText
9.         android:id="@+id/name"
10.        android:layout_width="fill_parent"
11.        android:layout_height="wrap_content"
12.        android:hint="reminder" />
13.     <Spinner
14.         android:id="@+id/dates"
15.         android:layout_width="0dp"
16.         android:layout_height="wrap_content"

```

```

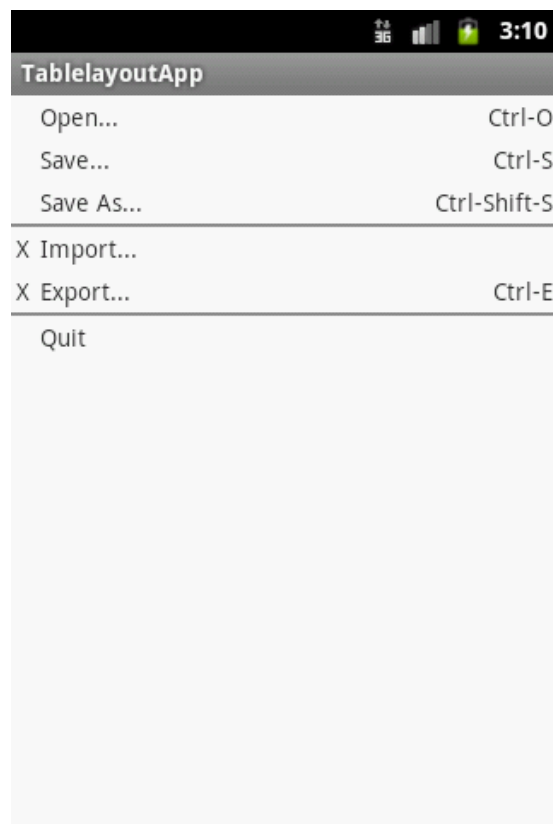
16.         android:layout_below="@id/name"
17.         android:layout_alignParentLeft="true"
18.         android:layout_toLeftOf="@+id/times" />
19.     <Spinner
20.         android:id="@id/times"
21.         android:layout_width="96dp"
22.         android:layout_height="wrap_content"
23.         android:layout_below="@id/name"
24.
25.         android:layout_alignParentRight="true" />
26.     <Button
27.         android:layout_width="96dp"
28.         android:layout_height="wrap_content"
29.         android:layout_below="@id/times"
30.         android:layout_alignParentRight="true"
31.         android:text="done" />
32. </RelativeLayout>

```

3. Selesai, jalankan aplikasinya.

3.1.3. Tablelayout

Analoginya mirip table yang sering kita jumpai di HTML atau pada EXCEL. Tablelayout ini bisa terdiri atas banyak Tablerow. Setiap Tablerow bisa memiliki banyak cell yang berisi widget seperti button, textview, spinner dan lainnya.



Gambar 1.10. Tampilan akhir aplikasi TablelayoutApp

1. Siapkan project baru dengan parameter sebagai berikut

Application Name	TablelayoutApp
Project Name	TablelayoutApp
Package Name	com.table.layout.app

2. Tambahkan coding berikut pada activity_main.xml

```

3. <?xml version="1.0" encoding="utf-8"?>
4. <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
5.     android:layout_width="fill_parent"
6.     android:layout_height="fill_parent"
7.     android:stretchColumns="1" >
8.
9.     <TableRow>
10.
11.         <TextView
12.             android:layout_column="1"
13.             android:padding="3dip"
14.             android:text="Open..." />
15.
16.         <TextView
17.             android:gravity="right"
18.             android:padding="3dip"
19.             android:text="Ctrl-O" />
20.     </TableRow>
21.
22.     <TableRow>
23.
24.         <TextView
25.             android:layout_column="1"
26.             android:padding="3dip"
27.             android:text="Save..." />
28.
29.         <TextView
30.             android:gravity="right"
31.             android:padding="3dip"
32.             android:text="Ctrl-S" />
33.     </TableRow>
34.
35.     <TableRow>
36.
37.         <TextView
38.             android:layout_column="1"
39.             android:padding="3dip"
40.             android:text="Save As..." />
41.
42.         <TextView
43.             android:gravity="right"
44.             android:padding="3dip"
45.             android:text="Ctrl-Shift-S" />
46.     </TableRow>
47.
48.     <View
49.         android:layout_height="2dip"
50.         android:background="#FF909090" />
51.
52.     <TableRow>

```



```
53.
54.         <TextView
55.             android:padding="3dip"
56.             android:text="X" />
57.
58.         <TextView
59.             android:padding="3dip"
60.             android:text="Import..." />
61.     </TableRow>
62.
63.     <TableRow>
64.
65.         <TextView
66.             android:padding="3dip"
67.             android:text="X" />
68.
69.         <TextView
70.             android:padding="3dip"
71.             android:text="Export..." />
72.
73.         <TextView
74.             android:gravity="right"
75.             android:padding="3dip"
76.             android:text="Ctrl-E" />
77.     </TableRow>
78.
79.     <View
80.         android:layout_height="2dip"
81.         android:background="#FF909090" />
82.
83.     <TableRow>
84.
85.         <TextView
86.             android:layout_column="1"
87.             android:padding="3dip"
88.             android:text="Quit" />
89.     </TableRow>
90.
91. </TableLayout>
```


Bagian 2 | Basic

2.1. Menggunakan Button

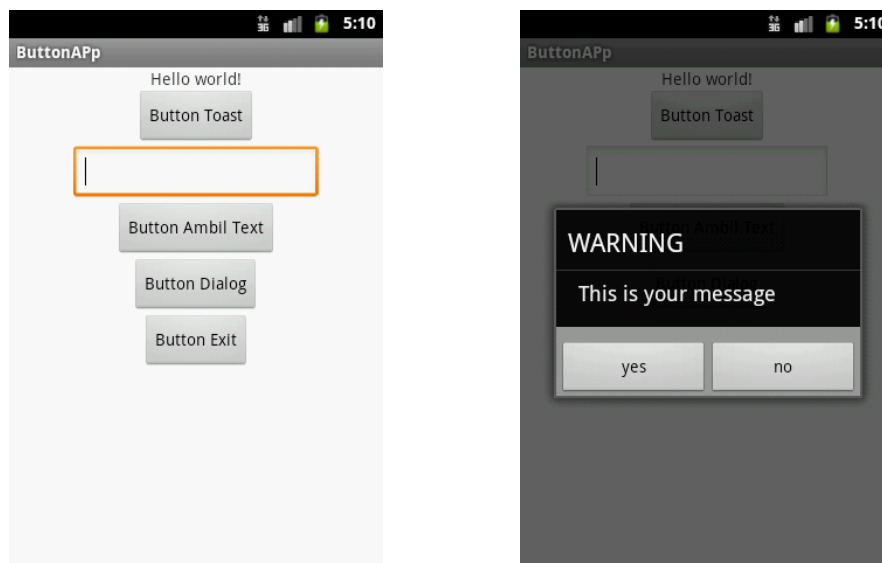
Poin Penting!

- | | |
|--|--|
| <ul style="list-style-type: none"> - findViewById(); - setOnClickListener(); - getId(); - AlertDialog(); | <ul style="list-style-type: none"> - Toast(); - getText(); - setText(); |
|--|--|

Setelah memahami dasar-dasar layout, sekarang saatnya kita belajar ke tingkat berikutnya. Nah bagian dua ini, kita akan banyak berinteraksi dengan code java dan belajar mengenai beberapa hal yaitu

1. mengontrol button
2. menampilkan toast
3. menampilkan dialog
4. mengambil teks pada field Edittext

Supaya ada gambaran mengenai aplikasi yang akan dibuat, berikut ini adalah tampilan aplikasi



Gambar 2.1. Tampilan aplikasi ButtonApp saat dijalankan

Ok, mari kita mulai

1. siapkan project baru dengan parameter sebagai berikut

Application Name	ButtonApp
Project Name	ButtonApp
Package Name	com.button.app



2. Siapkan activity_main.xml seperti Gambar 2.1. atau lihat coding berikut ini

```

1. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2.     xmlns:tools="http://schemas.android.com/tools"
3.     android:id="@+id/LinearLayout1"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:gravity="center_horizontal"
7.     android:orientation="vertical" >
8.
9.     <TextView
10.        android:id="@+id/textView1"
11.        android:layout_width="wrap_content"
12.        android:layout_height="wrap_content"
13.        android:text="@string/hello_world" />
14.
15.     <Button
16.        android:id="@+id/bSatu"
17.        android:layout_width="wrap_content"
18.        android:layout_height="wrap_content"
19.        android:text="Button Toast" />
20.
21.     <EditText
22.        android:id="@+id/inName"
23.        android:layout_width="wrap_content"
24.        android:layout_height="wrap_content"
25.        android:ems="10" />
26.
27.     <Button
28.        android:id="@+id/bDua"
29.        android:layout_width="wrap_content"
30.        android:layout_height="wrap_content"
31.        android:text="Button Ambil Text" />
32.
33.     <Button
34.        android:id="@+id/bTiga"
35.        android:layout_width="wrap_content"
36.        android:layout_height="wrap_content"
37.        android:text="Button Dialog" />
38.
39.     <Button
40.        android:id="@+id/bEmpat"
41.        android:layout_width="wrap_content"
42.        android:layout_height="wrap_content"
43.        android:text="Button Exit" />
44.
45. </LinearLayout>

```

3. Salinlah code MainActivity.java seperti dibawah

```

1. package com.button.app;
2.
3. import android.os.Bundle;
4. import android.app.Activity;
5. import android.app.AlertDialog;
6. import android.content.DialogInterface;
7. import android.view.View;
8. import android.view.View.OnClickListener;

```

```

9. import android.widget.Button;
10. import android.widget.EditText;
11. import android.widget.Toast;
12.
13. public class MainActivity extends Activity implements OnClickListener {
14.
15.     Button btnSatu, btnDua, btnTiga, btnEmpat;
16.     EditText nama;
17.     String tempNama;
18.
19.     @Override
20.     protected void onCreate(Bundle savedInstanceState) {
21.         super.onCreate(savedInstanceState);
22.         setContentView(R.layout.activity_main);
23.
24.         btnSatu = (Button) findViewById(R.id.bSatu);
25.         btnSatu.setOnClickListener(this);
26.
27.         btnDua = (Button) findViewById(R.id.bDua);
28.         btnDua.setOnClickListener(this);
29.
30.         btnTiga = (Button) findViewById(R.id.bTiga);
31.         btnTiga.setOnClickListener(this);
32.
33.         btnEmpat = (Button) findViewById(R.id.bEmpat);
34.         btnEmpat.setOnClickListener(this);
35.
36.         nama = (EditText) findViewById(R.id.inName);
37.
38.     }
39.
40.     @Override
41.     public void onClick(View v) {
42.         // TODO Auto-generated method stub
43.         switch (v.getId()) {
44.             case R.id.bSatu:
45.
46.                 Toast.makeText(this, "hahahaha saya di pencet",
47.                     Toast.LENGTH_SHORT).show();
48.
49.                 break;
50.
51.             case R.id.bDua:
52.
53.                 tempNama = nama.getText().toString();
54.                 Toast.makeText(this, "nama saya : " + tempNama,
55.                     Toast.LENGTH_SHORT).show();
56.                 break;
57.
58.             case R.id.bTiga:
59.                 AlertDialog.Builder ad=new AlertDialog.Builder(this);
60.                 ad.setTitle("WARNING");
61.                 ad.setMessage("This is your message");
62.                 ad.show();
63.                 break;
64.             case R.id.bEmpat:
65.                 AlertDialog.Builder d=new AlertDialog.Builder(this);
66.                 d.setTitle("WARNING");
67.                 d.setMessage("This is your message");
68.                 d.setPositiveButton("yes", new
69.                     DialogInterface.OnClickListener() {
70.
71.                         @Override
72.                         public void onClick(DialogInterface dialog, int

```

```

73.     which) {
74.                                     // TODO Auto-generated method stub
75.                                     finish();
76.     }
77.     });
78.     d.setNegativeButton("no", new
79. DialogInterface.OnClickListener() {
80.
81.         @Override
82.         public void onClick(DialogInterface dialog, int
83. which) {
84.                                     // TODO Auto-generated method stub
85.                                     dialog.dismiss();
86.     }
87.     });
88.     d.show();
89.     break;
90. }
91. }
92.
93. }
94.

```

4. Selesai. Selamat menjalankan aplikasi. 😊

2.2. Memanggil halaman lain

Key point!

- | | |
|------------------|-------------------------------|
| 5. Activity | 8. SharedPreferences |
| 6. Intent | 9. AndroidManifest |
| 7. startActivity | 10. Override/implement method |

Sejauh ini kita belajar hanya menggunakan satu activity saja, nah sekarang saatnya kita melibatkan dua halaman saling memanggil. Siapkan dulu dua buah layout halaman satu dan dua masing-masing berisi satu Button, satu EditText dan satu TextView. Skenarionya, kita bisa menuliskan sebuah kata pada halaman satu. Kemudian jika button pada halaman satu diklik maka aplikasi akan berpindah ke halaman dua dengan menampilkan kata yang telah dimasukkan sebelumnya. Ilustrasinya seperti pada Gambar 2.2.



Gambar 2.2. Ilustrasi skenario aplikasi IntentStartApp

Lets make it real! 😊

1. Siapkan project baru dengan parameter sebagai berikut

Application Name	ActivityApp
Project Name	ActivityApp
Package Name	com.activity.app

2. Pada activity_main.xml, tambahkan satu EditText dan satu Button dengan tampilan seperti Gambar 2.2. Berikut ini code nya


```

1. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2.     xmlns:tools="http://schemas.android.com/tools"
3.     android:id="@+id/LinearLayout1"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:orientation="vertical"
7.     tools:context=".MainActivity" >
8.
9.     <TextView
10.         android:id="@+id/textView1"
11.         android:layout_width="wrap_content"
12.         android:layout_height="wrap_content"
13.         android:text="Page-1" />
14.
15.     <EditText
16.         android:id="@+id/inText"
17.         android:layout_width="match_parent"
18.         android:layout_height="wrap_content"
19.         android:ems="10" />
20.
21.     <Button
22.         android:id="@+id/bNext"
23.         android:layout_width="wrap_content"
24.         android:layout_height="wrap_content"
25.         android:layout_gravity="right"
26.         android:text="Next" />
27.
28. </LinearLayout>

```

3. Buat layout baru **activity_dua.xml**. Klik kanan folder res/layout > new > android XML file.
4. Pada activity_dua.xml, tambahkan komponen yang sama dengan activity_main.xml. Ubah Textview menjadi tulisan "Page-2". Berikut code activity_dua.xml.
5. Buat class baru DuaActivity.java. Caranya klik kanan package > new > class.
6. Salin code MainActivity.java berikut.

```

1. import android.os.Bundle;
2. import android.app.Activity;
3. import android.content.Intent;
4. import android.content.SharedPreferences;
5. import android.view.Menu;
6. import android.view.View;
7. import android.view.View.OnClickListener;
8. import android.widget.Button;
9. import android.widget.EditText;
10.
11. public class MainActivity extends Activity implements OnClickListener {
12.
13.     Button bNext;
14.     EditText inText;
15.     SharedPreferences sp;
16.     SharedPreferences.Editor spe;
17.
18.     @Override
19.     protected void onCreate(Bundle savedInstanceState) {
20.         super.onCreate(savedInstanceState);
21.         setContentView(R.layout.activity_main);
22.

```



```

23.         bNext = (Button) findViewById(R.id.bNext);
24.         bNext.setOnClickListener(this);
25.
26.         inText = (EditText) findViewById(R.id.inText);
27.
28.         sp = this.getSharedPreferences("data", 0);
29.         spe = sp.edit();
30.
31.     }
32.
33.     @Override
34.     public void onClick(View v) {
35.         // TODO Auto-generated method stub
36.
37.         String input = inText.getText().toString();
38.
39.         spe.putString("isidata", input);
40.         spe.commit();
41.
42.         Intent i = new Intent(MainActivity.this, DuaActivity.class);
43.         startActivity(i);
44.
45.     }
46.
47. }

```

7. Salin code untuk DuaActivity.java berikut ini

```

1. import android.app.Activity;
2. import android.content.Intent;
3. import android.content.SharedPreferences;
4. import android.os.Bundle;
5. import android.view.View;
6. import android.view.View.OnClickListener;
7. import android.widget.Button;
8. import android.widget.EditText;
9.
10. public class DuaActivity extends Activity implements OnClickListener {
11.     Button bBack;
12.     EditText outText;
13.     SharedPreferences sp;
14.     SharedPreferences.Editor spe;
15.
16.     @Override
17.     protected void onCreate(Bundle savedInstanceState) {
18.         // TODO Auto-generated method stub
19.         super.onCreate(savedInstanceState);
20.         setContentView(R.layout.activity_dua);
21.
22.         bBack = (Button) findViewById(R.id.bBack);
23.         bBack.setOnClickListener(this);
24.         outText = (EditText) findViewById(R.id.outText);
25.
26.         sp = this.getSharedPreferences("data", 0);
27.         spe = sp.edit();
28.
29.         String out = sp.getString("isidata", "");
30.         outText.setText(out);
31.
32.     }
33.
34.     @Override

```



```

35.     public void onClick(View arg0) {
36.         // TODO Auto-generated method stub
37.         Intent i = new Intent(DuaActivity.this, MainActivity.class);
38.         startActivity(i);
39.     }
40.
41. }

```

42. Daftarkan DuaActivity.java pada AndroidManifest > application (baris 24). Berikut ini AndroidManifest keseluruhan.

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3.     package="com.activity.app"
4.     android:versionCode="1"
5.     android:versionName="1.0" >
6.
7.     <uses-sdk
8.         android:minSdkVersion="8"
9.         android:targetSdkVersion="9" />
10.
11.     <application
12.         android:allowBackup="true"
13.         android:icon="@drawable/ic_launcher"
14.         android:label="@string/app_name"
15.         android:theme="@style/AppTheme" >
16.         <activity
17.             android:name="com.activity.app.MainActivity"
18.             android:label="@string/app_name" >
19.             <intent-filter>
20.                 <action android:name="android.intent.action.MAIN" />
21.                 <category android:name="android.intent.category.LAUNCHER" />
22.             </intent-filter>
23.         </activity>
24.         <activity android:name="DuaActivity"></activity>
25.     </application>
26.
27. </manifest>

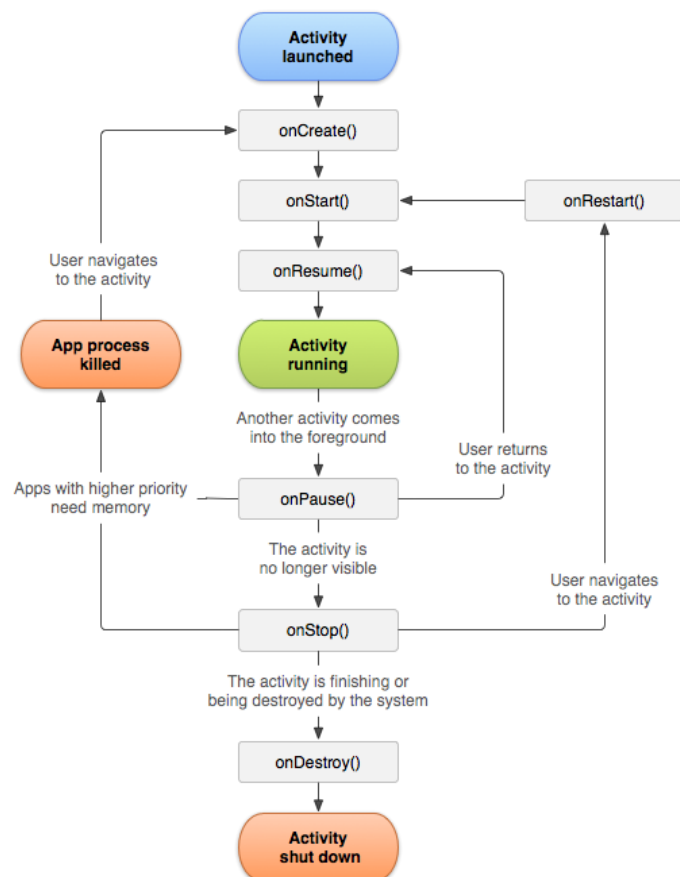
```

2.3. Siklus hidup aplikasi

Key poin!!	
11. Override/implement method	16. onResume()
12. onCreate()	17. onPause()
13. onResume()	18. onStop()
14. onStart()	19. onDestroy()
15. onRestart()	

Salah satu kelebihan smartphone adalah kehebatannya mengelola sendiri memori atau aktifitas prosesor. Dalam rangka mengatur penggunaan memori, aplikasi android mempunyai cara unik untuk mengendalikan status.

Sejauh ini kita membuat aplikasi, pasti menjumpai method onCreate() pada MainActivity.java. Method onCreate() adalah salah satu method untuk mengontrol siklus hidup aplikasi, di eksekusi pertama kali saat aplikasi dijalankan. Masih ada method-method lain yang di eksekusi pada kondisi tertentu. Gambar 2.3. menjelaskan urutan terjadinya siklus hidup sebuah aplikasi, menunjukkan saat method-methodnya dieksekusi.





Gambar 2.3. Ilustrasi siklus hidup aplikasi

1. buat project baru dengan parameter

Application Name	LifecycleApp
Project Name	LifecycleApp
Package Name	com.life.cycle.app

2. Tambahkan method-method seperti code dibawah ini pada MainActivity.java

```

1. import android.os.Bundle;
2. import android.app.Activity;
3. import android.view.Menu;
4. import android.widget.Toast;
5.
6. public class MainActivity extends Activity {
7.
8.     @Override
9.     protected void onCreate(Bundle savedInstanceState) {
10.         super.onCreate(savedInstanceState);
11.         setContentView(R.layout.activity_main);
12.
13.         Toast.makeText(this, "onCreate executed", Toast.LENGTH_SHORT).show();
14.     }
15.
16.     @Override
17.     protected void onPause() {
18.         // TODO Auto-generated method stub
19.         super.onPause();
20.         Toast.makeText(this, "onStop executed", Toast.LENGTH_SHORT).show();
21.     }
22.
23.     @Override
24.     protected void onRestart() {
25.         // TODO Auto-generated method stub
26.         super.onRestart();
27.         Toast.makeText(this, "onRestart executed", Toast.LENGTH_SHORT).show();
28.     }
29.
30.     @Override
31.     protected void onResume() {
32.         // TODO Auto-generated method stub
33.         super.onResume();
34.         Toast.makeText(this, "onResume executed", Toast.LENGTH_SHORT).show();
35.     }
36.
37.     @Override
38.     protected void onStart() {
39.         // TODO Auto-generated method stub
40.         super.onStart();
41.         Toast.makeText(this, "onRestart executed", Toast.LENGTH_SHORT).show();
42.     }
43.
44.     @Override
45.     protected void onStop() {
46.         // TODO Auto-generated method stub
47.         super.onStop();
48.         Toast.makeText(this, "onStop executed", Toast.LENGTH_SHORT).show();
49.     }
50.
51.     @Override
52.     public boolean onCreateOptionsMenu(Menu menu) {
53.
54.         getMenuInflater().inflate(R.menu.activity_main, menu);

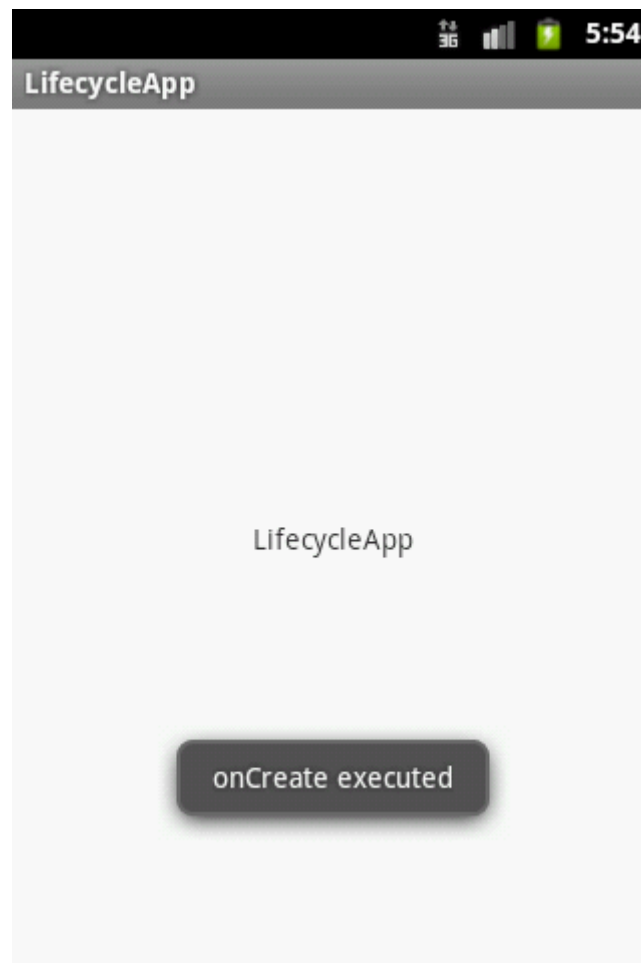
```



```

55.         return true;
56.     }
57.
58.     @Override
59.     protected void onDestroy() {
60.         // TODO Auto-generated method stub
61.         super.onDestroy();
62.         Toast.makeText(this, "onDestroy executed", Toast.LENGTH_SHORT).show();
63.     }
64.
65. }
    
```

3. Selesai dan jalankan, hasilnya tampak seperti Gambar 2.4



Gambar 2.4. Tampilan LifecycleApp saat dijalankan

Lakukan ujicoba dengan menjalankan aplikasi, memencet tombol Home atau memencet tombol Back secara acak atau bergantian. Toast akan muncul pada setiap method dieksekusi. Sambil mencoba cocokkan dengan siklus pada Gambar 2.3.

Bagian 3 | Multimedia

3.1. Menampilkan gambar

Ada dua cara untuk menampilkan suatu gambar, menggunakan widget ImageView atau menggunakan Webview. Pada bab ini kita menampilkan gambar menggunakan widget ImageView. Seperti apa sih hasilnya nanti? Perhatikan gambar 3.1 berikut ini



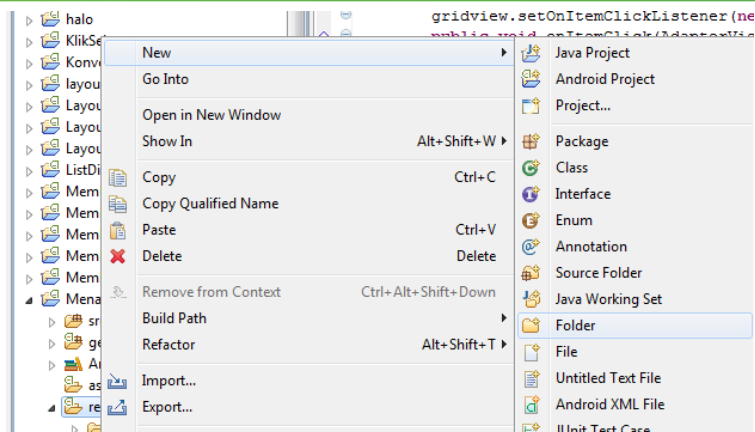
Gambar 3. 1

Kita mulai dengan cara yang sederhana.

1. Buat project baru dengan parameter

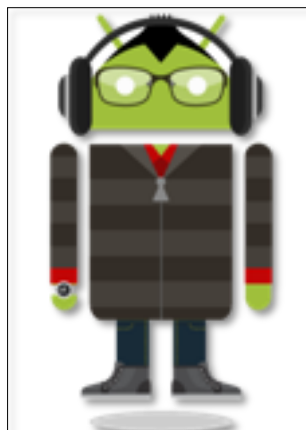
Application Name	GambarApp
Project Name	GambarApp
Package Name	com.gambar.app

2. Bikin folder baru dengan nama **drawable** di dalam **GambarApp/res**. Caranya klik kanan folder **res** > **new** > **folder** (Gambar 3.2).

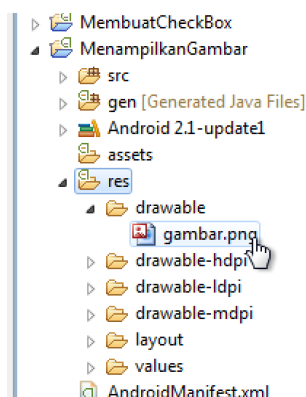


Gambar 3.2

3. Masukkan sebuah gambar PNG atau JPG ke dalam folder drawable (Gambar 3.5). **Klik kanan folder drawable > paste.** Latihan ini memakai gambar format PNG berukuran 250px x 275px. (Gambar 3.3).



Gambar 3.3. Gambar yang akan ditampilkan



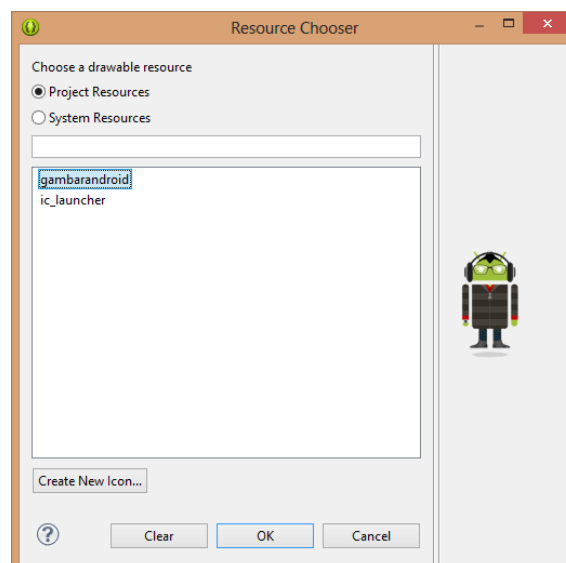
Gambar 3.5. Meletakkan gambar pada folder drawable

4. Tambahkan widget ImageView pada activity_main.xml. Pada saat menaruh ImageView pada layout, akan muncul kotak dialog seperti Gambar 3.6. Pilih gambar yang telah disiapkan. Berikut code untuk activity_main.xml.

```
<RelativeLayout 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=".MainActivity" >

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="82dp"
        android:src="@drawable/gambarandroid" />

</RelativeLayout>
```



Gambar 3.6. Memasukkan gambar pada ImageView

5. Pastikan kode pada **tampilanGambar.java** seperti berikut

```
1. import android.os.Bundle;
2. import android.app.Activity;
3. import android.view.Menu;
4.
5. public class MainActivity extends Activity {
6.
7.     @Override
8.     protected void onCreate(Bundle savedInstanceState) {
9.         super.onCreate(savedInstanceState);
10.        setContentView(R.layout.activity_main);
```

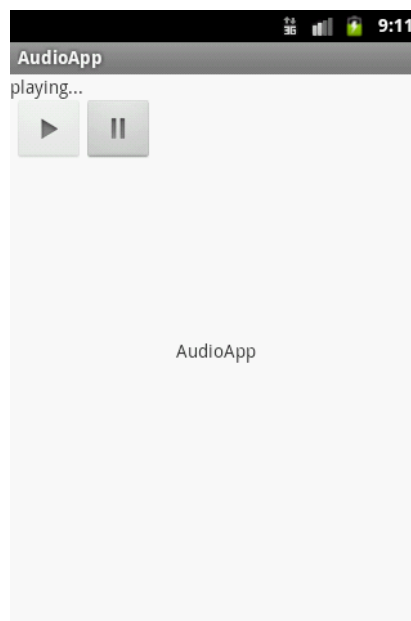
```

11.     }
12.
13.     @Override
14.     public boolean onCreateOptionsMenu(Menu menu) {
15.
16.         getMenuInflater().inflate(R.menu.activity_main, menu);
17.         return true;
18.     }
19.
20. }
    
```

21. Lakukan RUN dan lihat hasilnya. Mudah bukan??

3.2. Memanggil audio

Bagian ini kita akan membuat aplikasi sederhana untuk memutar sebuah lagu mp3. Hasil akhir aplikasi ini nantinya seperti Gambar 3.7. Gunakan file mp3 yang ukurannya tidak terlalu besar.



Gambar 3.7. Tampilan aplikasi audioApp

1. Buat project baru dengan parameter
2. Pada activity_main.xml, tambahkan ImageButton dua buah dengan resource gambar ic_media_play dan ic_media_pause. Berikut code untuk activity_main.xml.

```

1. <RelativeLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
2.     xmlns:tools="http://schemas.android.com/tools"
3.     android:layout_width="match_parent"
4.     android:layout_height="match_parent"
5.     tools:context=".MainActivity" >
6.
7.     <TextView
    
```

```

8.         android:id="@+id/textView1"
9.         android:layout_width="wrap_content"
10.        android:layout_height="wrap_content"
11.        android:layout_centerHorizontal="true"
12.        android:layout_centerVertical="true"
13.        android:text="AudioApp" />
14.
15.        <ImageButton
16.            android:id="@+id/bPause"
17.            android:layout_width="wrap_content"
18.            android:layout_height="wrap_content"
19.            android:layout_alignTop="@+id/bPlay"
20.            android:layout_toRightOf="@+id/bPlay"
21.            android:src="@android:drawable/ic_media_pause" />
22.
23.        <TextView
24.            android:id="@+id/tKet"
25.            android:layout_width="wrap_content"
26.            android:layout_height="wrap_content"
27.            android:layout_alignParentLeft="true"
28.            android:layout_alignParentTop="true"
29.            android:text="TextView" />
30.
31.        <ImageButton
32.            android:id="@+id/bPlay"
33.            android:layout_width="wrap_content"
34.            android:layout_height="wrap_content"
35.            android:layout_alignRight="@+id/tKet"
36.            android:layout_below="@+id/tKet"
37.            android:src="@android:drawable/ic_media_play" />
38.
39.    </RelativeLayout>

```

3. Salin code MainActivity.java berikut ini

```

1. import java.io.IOException;
2. import android.media.MediaPlayer;
3. import android.media.MediaPlayer.OnCompletionListener;
4. import android.os.Bundle;
5. import android.app.Activity;
6. import android.view.Menu;
7. import android.view.View;
8. import android.view.View.OnClickListener;
9. import android.widget.ImageButton;
10. import android.widget.TextView;
11.
12. public class MainActivity extends Activity implements OnClickListener {
13.     ImageButton bplay, bpause;
14.     TextView tket;
15.     MediaPlayer mp;
16.
17.     @Override
18.     protected void onCreate(Bundle savedInstanceState) {
19.         super.onCreate(savedInstanceState);
20.         setContentView(R.layout.activity_main);
21.
22.         bplay = (ImageButton) findViewById(R.id.bPlay);
23.         bpause = (ImageButton) findViewById(R.id.bPause);
24.         tket = (TextView) findViewById(R.id.tKet);

```

```

25.
26.         bplay.setOnClickListener(this);
27.         bpause.setOnClickListener(this);
28.
29.         mp = MediaPlayer.create(MainActivity.this, R.raw.forsaken);
30.     }
31.
32.     @Override
33.     public boolean onCreateOptionsMenu(Menu menu) {
34.
35.         getMenuInflater().inflate(R.menu.activity_main, menu);
36.         return true;
37.     }
38.
39.     @Override
40.     public void onClick(View v) {
41.         // TODO Auto-generated method stub
42.         switch (v.getId()) {
43.             case R.id.bPlay:
44.                 bplay.setEnabled(false);
45.                 tket.setText("playing...");
46.                 if (!mp.isPlaying()) {
47.                     try {
48.                         mp.prepare();
49.                     } catch (IllegalStateException e) {
50.                         // TODO Auto-generated catch block
51.                         e.printStackTrace();
52.                     } catch (IOException e) {
53.                         // TODO Auto-generated catch block
54.                         e.printStackTrace();
55.                     }
56.                 }
57.
58.                 mp.start();
59.                 mp.setOnCompletionListener(new OnCompletionListener() {
60.                     public void onCompletion(MediaPlayer arg0) {
61.                         bplay.setEnabled(true);
62.                         tket.setText("play again..");
63.                     }
64.                 });
65.
66.                 break;
67.
68.             case R.id.bPause:
69.                 tket.setText("stop...");
70.                 mp.pause();
71.                 bplay.setEnabled(true);
72.                 break;
73.         }
74.     }
75.
76. }

```

4. Selesai dan jalankan! ☺



Bagian 4 | Android Academy

android academy
CARA CEPAT MENJADI ANDROID PROFESSIONAL DEVELOPER

Level 1 : Android Basic
Level 2 : Android Intermediate
Level 3: Android Advanced

Pilihan hari Senin-kamis atau selasa-jumat.
Mulai kelas 21 JANUARI 2013

Biaya :
 Level 1 : Rp 750.000,- (10 x pertemuan)
 Level 2 : Rp 750.000,- (10 x pertemuan)
 Level 3 : Rp 600.000,- (8 x pertemuan)
 All : Rp 2.100.000,-
 Harga Promo : Rp. 1.785.000,-

Jl. Wahid Hasyim No. 99B
 Nologaten - Yogyakarta
 Telp. 0274-4333224
 SMS Center : 085 747 969 669

imagineitcenter @imagineitcenter
www.imagineitcenter.com

IMAGINE
IT EDUCATION CENTER

▶ one place to learn all about IT

INFO :

Klik www.imagineitcenter.com atau
 Gedug Imagine IT Education Center
 Jalan Wahid Hasyim no 99 Nologaten, Condongcatur, Depok,
 Sleman
 Pendaftaran langsung ke IMAGINE IT Education Center
 Booking sms ke 085747969669 atau hub call centre 0274-4333224

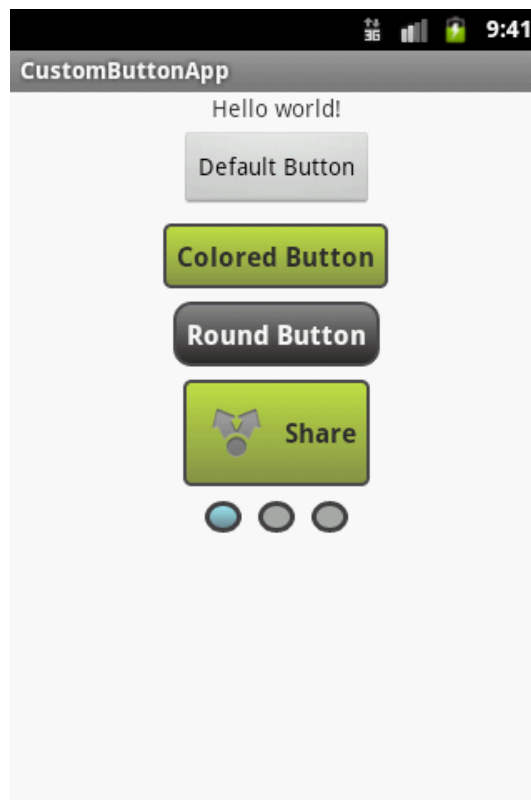
Bagian 5 | User Interface

5.1. Membuat gradient

5.1.1. *Not availabel in this version*

5.2. Membuat custom button

Pada bagian ini kita akan mengetahui trik cara membuat custom button sehingga tmapilan button terlihat lebih menarik. Hasil aplikasi seperti Gambar 5.1.



Gambar 5.1. hasil aplikasi customButtonApp

1. Buat project baru dengna parameter

Application Name	CustomButtonApp
Project Name	CustomButtonApp
Package Name	com.custom.button.app

2. Buat folder **drawable** didalam res/layout., lalu buat empat buah file xml masing masing code berikut ini

- circlebutton.xml

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <selector xmlns:android="http://schemas.android.com/apk/res/android">
3.
4.     <item android:state_pressed="true"><shape>

```

```

5.         <solid android:color="#677e86" />
6.
7.         <stroke android:width="3dp" android:color="#4b4c4c" />
8.
9.         <corners android:bottomLeftRadius="16dp"
10.                android:bottomRightRadius="16dp"
11.                android:topLeftRadius="16dp"
12.                android:topRightRadius="16dp" />
13.     </shape></item>
14.     <item><shape>
15.         <gradient android:angle="270"
16.                android:endColor="#677e86"
17.                android:startColor="#a3f2ff" />
18.
19.         <stroke android:width="3dp"
20.                android:color="#444" />
21.
22.         <corners android:bottomLeftRadius="16dp"
23.                android:bottomRightRadius="16dp"
24.                android:topLeftRadius="16dp"
25.                android:topRightRadius="16dp" />
26.     </shape></item>
27.
28. </selector>

```

- circlebuttoninactive.xml

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <selector xmlns:android="http://schemas.android.com/apk/res/android">
3.
4.     <item android:state_pressed="true"><shape>
5.         <solid android:color="#677e86" />
6.
7.         <stroke android:width="3dp"
8.                android:color="#4b4c4c" />
9.
10.        <corners android:bottomLeftRadius="16dp"
11.                android:bottomRightRadius="16dp"
12.                android:topLeftRadius="16dp"
13.                android:topRightRadius="16dp" />
14.    </shape></item>
15.    <item><shape>
16.        <gradient android:angle="270"
17.                android:endColor="@android:color/darker_gray"
18.                android:startColor="@android:color/darker_gray" />
19.
20.        <stroke android:width="3dp"
21.                android:color="#444" />
22.
23.        <corners android:bottomLeftRadius="16dp"
24.                android:bottomRightRadius="16dp"
25.                android:topLeftRadius="16dp"
26.                android:topRightRadius="16dp" />
27.    </shape></item>
28.
29. </selector>

```

- colorbutton.xml

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <selector xmlns:android="http://schemas.android.com/apk/res/android">
3.
4.     <item android:state_pressed="true"><shape>
5.         <solid android:color="#879548" />
6.
7.         <stroke android:width="2dp"
8.             android:color="#4b4c4c" />
9.
10.        <corners android:bottomLeftRadius="4dp"
11.            android:bottomRightRadius="4dp"
12.            android:topLeftRadius="4dp"
13.            android:topRightRadius="4dp" />
14.    </shape></item>
15.    <item><shape>
16.        <gradient android:angle="270"
17.            android:endColor="#879548"
18.            android:startColor="#c4e347" />
19.
20.        <stroke android:width="2dp"
21.            android:color="#4b4c4c" />
22.
23.        <corners android:bottomLeftRadius="4dp"
24.            android:bottomRightRadius="4dp"
25.            android:topLeftRadius="4dp"
26.            android:topRightRadius="4dp" />
27.    </shape></item>
28.
29.</selector>

```

- roundbutton.xml

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <selector xmlns:android="http://schemas.android.com/apk/res/android">
3.
4.     <item android:state_pressed="true"><shape>
5.         <solid android:color="#262523" />
6.
7.         <stroke android:width="2dp"
8.             android:color="#4b4c4c" />
9.
10.        <corners android:bottomLeftRadius="10dp"
11.            android:bottomRightRadius="10dp"
12.            android:topLeftRadius="10dp"
13.            android:topRightRadius="10dp" />
14.    </shape></item>
15.    <item><shape>
16.        <gradient android:angle="270"
17.            android:endColor="#262523"
18.            android:startColor="#8c8c8c" />
19.
20.        <stroke android:width="2dp"
21.            android:color="#4b4c4c" />

```

```

22.
23.         <corners android:bottomLeftRadius="10dp"
24.             android:bottomRightRadius="10dp"
25.             android:topLeftRadius="10dp"
26.             android:topRightRadius="10dp" />
27.     </shape></item>
28.
29. </selector>

```

3. Pada activity_main.xml, tambahkan tujuh buah button dengan code sebagai berikut

```

1. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2.     xmlns:tools="http://schemas.android.com/tools"
3.     android:id="@+id/LinearLayout1"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:layout_gravity="center_horizontal"
7.     android:gravity="center_horizontal"
8.     android:orientation="vertical"
9.     tools:context=".MainActivity" >
10.
11.     <TextView
12.         android:layout_width="wrap_content"
13.         android:layout_height="wrap_content"
14.         android:text="@string/hello_world" />
15.
16.     <Button
17.         android:id="@+id/button1"
18.         android:layout_width="wrap_content"
19.         android:layout_height="wrap_content"
20.         android:layout_margin="4dp"
21.         android:text="Default Button" />
22.
23.     <Button
24.         android:id="@+id/button2"
25.         style="@style/ButtonText"
26.         android:layout_width="wrap_content"
27.         android:layout_height="wrap_content"
28.         android:layout_margin="4dp"
29.         android:background="@drawable/colorbutton"
30.         android:padding="8dp"
31.         android:text="Colored Button" />
32.
33.     <Button
34.         android:id="@+id/button3"
35.         style="@style/ButtonWhiteText"
36.         android:layout_width="wrap_content"
37.         android:layout_height="wrap_content"
38.         android:layout_margin="4dp"
39.         android:background="@drawable/roundbutton"
40.         android:padding="8dp"
41.         android:text="Round Button" />
42.
43.     <Button
44.         android:id="@+id/Button01"
45.         style="@style/ButtonText"
46.         android:layout_width="wrap_content"
47.         android:layout_height="wrap_content"
48.         android:layout_margin="4dp"

```



```

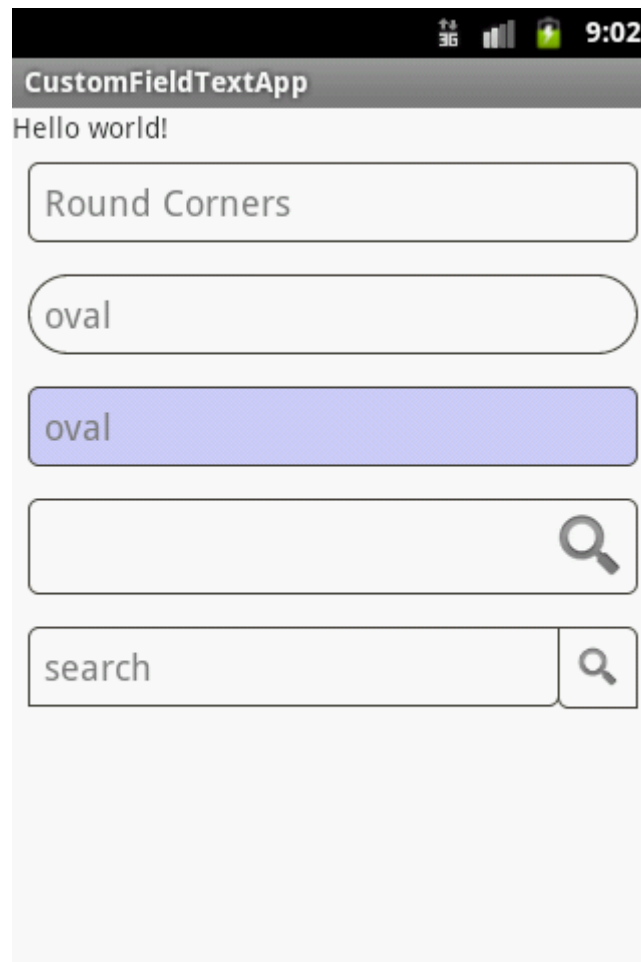
49.         android:background="@drawable/colorbutton"
50.         android:drawableLeft="@android:drawable/ic_menu_share"
51.         android:drawablePadding="5dp"
52.         android:padding="8dp"
53.         android:paddingRight="10dp"
54.         android:text="Share" />
55.
56.     <LinearLayout
57.         android:layout_width="wrap_content"
58.         android:layout_height="wrap_content" >
59.
60.         <Button
61.             android:id="@+id/button4"
62.             style="?android:attr/buttonStyleSmall"
63.             android:layout_width="wrap_content"
64.             android:layout_height="wrap_content"
65.             android:layout_margin="5dp"
66.             android:background="@drawable/circlebutton"
67.             android:paddingLeft="11dp"
68.             android:paddingRight="11dp" />
69.
70.         <Button
71.             android:id="@+id/button5"
72.             style="?android:attr/buttonStyleSmall"
73.             android:layout_width="wrap_content"
74.             android:layout_height="wrap_content"
75.             android:layout_margin="5dp"
76.             android:background="@drawable/circlebuttoninactive"
77.             android:paddingLeft="11dp"
78.             android:paddingRight="11dp" />
79.
80.         <Button
81.             android:id="@+id/button6"
82.             style="?android:attr/buttonStyleSmall"
83.             android:layout_width="wrap_content"
84.             android:layout_height="wrap_content"
85.             android:layout_margin="5dp"
86.             android:background="@drawable/circlebuttoninactive"
87.             android:paddingLeft="11dp"
88.             android:paddingRight="11dp" />
89.     </LinearLayout>
90.
91. </LinearLayout>

```

4. Finish! ☺

5.3. Membuat custom editteks

Selain mengcustomize button, kita juga bisa membuat tampilan EditText sendiri. Dengan mengcustom seperti ini, aplikasi yang akan kita bangun nantinya akan terlihat lebih menarik. Hasil akhir aplikasi ini seperti Gambar 5.2.



Gambar 5.2. Tampilan aplikasi CustomFieldTextApp

1. Buat project baru

Application Name	CustomFieldTextApp
Project Name	CustomFieldTextApp
Package Name	com.custom.edittext

2. Buat folder drawable, tambahkan masing-masing file xml berikut
 - a. fillcolor.xml

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <shape xmlns:android="http://schemas.android.com/apk/res/android" >
3.     <corners android:radius="5dp"/>
    
```




```

4.     <stroke android:width="1dp"
5.           android:color="#44433A"/>
6.     <solid android:color="#2D0000FF"/>
7. </shape>

```

b. leftroundborder.xml

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <shape xmlns:android="http://schemas.android.com/apk/res/android" >
3.
4.     <corners
5.         android:bottomLeftRadius="5dp"
6.         android:bottomRightRadius="0dp"
7.         android:topLeftRadius="5dp"
8.         android:topRightRadius="0dp" />
9.
10.    <stroke
11.        android:width="1dp"
12.        android:color="#44433A" />
13.
14.    <solid android:color="#FFF" />
15.
16. </shape>

```

c. rightroundborder.xml

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <shape xmlns:android="http://schemas.android.com/apk/res/android" >
3.
4.     <corners
5.         android:bottomLeftRadius="0dp"
6.         android:bottomRightRadius="5dp"
7.         android:topLeftRadius="0dp"
8.         android:topRightRadius="5dp" />
9.
10.    <stroke
11.        android:width="1dp"
12.        android:color="#44433A" />
13.
14.    <solid android:color="#FFF" />
15.
16. </shape>

```

d. noborder.xml

```

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

```

e. oval.xml

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <shape xmlns:android="http://schemas.android.com/apk/res/android" >
3.     <corners android:radius="50dp"/>
4.     <stroke android:width="1dp"
5.         android:color="#44433A"/>
6.     <solid android:color="#FFF"/>
7.
8. </shape>

```

f. round.xml

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <shape xmlns:android="http://schemas.android.com/apk/res/android" >
3.     <corners android:radius="5dp"/>
4.     <stroke android:width="1dp"
5.         android:color="#44433A"/>
6.     <solid android:color="#FFF"/>
7.
8. </shape>

```

3. Tambahkan lima EditText pada activity_main.xml dengna code seperti berikut

```

1. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2.     xmlns:tools="http://schemas.android.com/tools"
3.     android:id="@+id/LinearLayout1"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:orientation="vertical"
7.     tools:context=".MainActivity" >
8.
9.     <TextView
10.         android:layout_width="wrap_content"
11.         android:layout_height="wrap_content"
12.         android:text="@string/hello_world" />
13.
14.     <EditText
15.         android:id="@+id/editText1"
16.         android:layout_width="match_parent"
17.         android:layout_height="wrap_content"
18.         android:layout_margin="8dp"
19.         android:background="@drawable/round"
20.         android:ems="10"
21.         android:hint="Round Corners"
22.         android:padding="8dp" >
23.
24.         <requestFocus />
25.     </EditText>
26.
27.     <EditText
28.         android:id="@+id/EditText01"

```

```

29.         android:layout_width="match_parent"
30.         android:layout_height="wrap_content"
31.         android:layout_margin="8dp"
32.         android:background="@drawable/oval"
33.         android:ems="10"
34.         android:hint="oval"
35.         android:padding="8dp" />
36.
37.     <EditText
38.         android:id="@+id/EditText02"
39.         android:layout_width="match_parent"
40.         android:layout_height="wrap_content"
41.         android:layout_margin="8dp"
42.         android:background="@drawable/fillcolor"
43.         android:ems="10"
44.         android:hint="oval"
45.         android:padding="8dp" />
46.
47.     <LinearLayout
48.         android:layout_width="match_parent"
49.         android:layout_height="wrap_content"
50.         android:layout_margin="8dp"
51.         android:background="@drawable/round" >
52.
53.         <EditText
54.             android:id="@+id/editText2"
55.             android:layout_width="wrap_content"
56.             android:layout_height="match_parent"
57.             android:layout_weight="1"
58.             android:background="@drawable/noborder"
59.             android:ems="10" />
60.
61.         <ImageView
62.             android:id="@+id/imageView1"
63.             android:layout_width="wrap_content"
64.             android:layout_height="wrap_content"
65.             android:src="@android:drawable/ic_menu_search" />
66.     </LinearLayout>
67.
68.     <LinearLayout
69.         android:layout_width="fill_parent"
70.         android:layout_height="wrap_content"
71.         android:layout_margin="8dp"
72.         android:gravity="center"
73.         android:orientation="horizontal" >
74.
75.         <EditText
76.             android:layout_width="0dp"
77.             android:layout_height="wrap_content"
78.             android:layout_weight="1"
79.             android:background="@drawable/leftroundborder"
80.             android:hint="search"
81.             android:padding="8dp"
82.             android:singleLine="true" />
83.
84.         <ImageView
85.             android:layout_width="40dp"

```



```
86.         android:layout_height="41dp"
87.         android:layout_marginLeft="-1dp"
88.         android:background="@drawable/rightroundborder"
89.         android:padding="5dp"
90.         android:src="@android:drawable/ic_menu_search" />
91.     </LinearLayout>
92.
93. </LinearLayout>
```

4. Selesai.