

PERTEMUAN 3:

ACTIVITY

A. TUJUAN PEMBELAJARAN

Pada bab ini akan mahasiswa diharapkan mampu:

3.1 Mengetahui Kegunaan Activity

3.2 Membuat project dengan memanfaatkan activity

B. URAIAN MATERI

Tujuan Pembelajaran 3.1:

Mengetahui Kegunaan Activity

Activity adalah salah satu komponen aplikasi android yang digunakan untuk menyediakan sebuah layar bagi user agar dapat berinteraksi dengan aplikasi, misalnya mengirim SMS (Short Message Service), menelepon, mengambil foto, mengirim email, melihat peta, dan lain-lain. Setiap activity memiliki window sebagai tempat meletakkan user interface. Window tersebut biasanya memenuhi layar, bisa juga lebih kecil dari layar dan melayang di atas window lain.

Sebuah aplikasi dapat terdiri dari satu atau lebih activity, biasanya aplikasi dibuat dengan beberapa activity yang satu sama lain terikat secara longgar. Biasanya, satu activity dalam sebuah aplikasi dikhususkan sebagai activity "main" (utama), yang mana akan ditampilkan kepada user di saat aplikasi dijalankan pertama kali. Setiap activity kemudian bisa menjalankan activity lain untuk melakukan pekerjaan yang berbeda. Setiap kali ada activity baru dijalankan, activity sebelumnya dihentikan, tetapi sistem menyimpan activity tersebut di dalam "tumpukan" (stack). Ketika sebuah activity dimulai, activity tersebut ditempatkan di bagian paling atas pada stack dan mendapatkan fokus dari user. Prinsip antrian stack pada dasarnya menggunakan prinsip “yang pertama masuk adalah yang terakhir keluar” (last in first out), jadi ketika user selesai dengan sebuah activity dan menekan tombol KEMBALI, activity tersebut dikeluarkan dari stack (dan dihancurkan) dan activity pada stack di bawahnya dikembalikan.

Ketika sebuah activity dihentikan karena ada activity baru yang dimulai, perubahan keadaan ini dinotifikasikan melalui siklus metode pemanggilan kembali activity. Terdapat beberapa metode pemanggilan kembali yang bisa diterima oleh activity, karena adanya

perubahan keadaan dan setiap metode pemanggilan kembali menyediakan kita kesempatan untuk melakukan pekerjaan yang lebih spesifik yang sesuai dengan perubahan keadaan activity tersebut. Sebagai contoh, ketika activity berhenti, activity tersebut harus melepaskan obyek yang besar, misalnya koneksi jaringan atau database. Ketika activity dimulai kembali, kita bisa memperoleh kembali sumber daya yang dibutuhkan dan memulai kembali kegiatan yang diinterupsi. Transisi keadaan ini adalah bagian dari siklus hidup activity.

Membuat Activity

Untuk membuat sebuah activity, kita harus membuat sebuah subclass dari activity (atau subclass yang sudah ada). Dalam subclass yang dibuat, kita harus mengimplementasikan metode pemanggilan kembali yang akan dipanggil sistem pada saat transisi antar state (keadaan) yang bervariasi dari siklus hidupnya, misalnya ketika activity sedang dibuat, dihentikan, dikembalikan, atau dihancurkan. Metode pemanggilan kembali yang paling penting ada dua, yaitu: `a.onCreate()`

Kita harus mengimplementasikan metode ini. Sistem akan memanggil metode ini ketika membuat activity. Dalam implementasi kita, kita harus menginisialisasikan komponen-komponen penting dari activity kita. Yang paling penting, di sinilah kita harus memanggil metode `setContentView()` untuk mendefinisikan layout dari user interface activity.

b.onPause()

Sistem akan memanggil metode ini sebagai indikasi pertama bahwa user sedang meninggalkan activity kita (meskipun tidak selalu activity tersebut akan dihancurkan). Di sinilah biasanya kita harus melakukan perubahan yang mesti bertahan diluar sesi user yang sekarang. (karena user mungkin tidak akan kembali). terdapat beberapa siklus metode pemanggilan kembali yang harus kita gunakan untuk menyediakan pengalaman user interface yang halus dan lancar antar activity dan menangani interupsi yang tidak diinginkan yang dapat menghentikan activity atau bahkan dapat menghancurkan activity.

Menjalankan Activity

Kita bisa menjalankan activity dengan memanggil method `startActivity()`, memberikan activity tersebut sebuah Intent yang menggambarkan activity yang ingin kita

jalankan. Intent akan memberitahukan nama activity atau bisa juga tipe aksi seperti apa yang ingin dijalankan, dan kemudian sistem akan memilih activity yang sesuai untuk keinginan kita, bisa saja activity tersebut berasal dari aplikasi lain. Intent bisa juga membawa sejumlah kecil data yang akan digunakan oleh activity yang dijalankan.

Ketika kita bekerja dalam aplikasi kita sendiri, kita lebih sering menjalankan activity yang sudah diketahui. Kita juga bisa melakukan hal ini dengan cara membuat intent yang secara eksplisit menggambarkan activity yang ingin dijalankan, menggunakan nama class. Contoh di bawah ini, bagaimana sebuah activity menjalankan activity lain yang bernama NamaClass.

```
Intent intent = new Intent(this, NamaClass.class);  
startActivity(intent);
```

Namun bagaimanapun, aplikasi kita mungkin ingin melakukan beberapa pekerjaan, misalnya mengirimkan email, mengirimkan sms, atau update status, menggunakan data dari activity kita. Dalam hal ini, aplikasi kita mungkin tidak memiliki activity sendiri untuk melakukan pekerjaan tersebut, sehingga kita perlu menggunakan activity milik aplikasi lain dalam perangkat kita, yang bisa melakukan pekerjaan yang kita diinginkan. Di sinilah intent sangat bermanfaat, kita bisa membuat intent yang menggambarkan pekerjaan seperti apa yang ingin kita lakukan, dan sistem kemudian akan menjalankan activity yang sesuai dari aplikasi lain. Jika ada beberapa activity yang bisa menangani intent tersebut, maka user akan diminta memilih activity mana yang akan digunakan. Sebagai contoh, jika kita ingin agar user bisa mengirim sebuah email, kita bisa membuat intent seperti berikut ini.

```
Intent intent = new Intent(Intent.ACTION_SEND);  
intent.putExtra(Intent.EXTRA_EMAIL, recipientArray);  
startActivity(intent);
```

Tambahan ekstra berupa EXTRA_EMAIL yang ditambahkan ke intent adalah sebuah string array dari alamat email tujuan pengiriman email tersebut. Ketika sebuah aplikasi email merespon terhadap intent ini, aplikasi email tersebut akan membaca string array yang diberikan oleh ekstra dan menempatkannya di pengisian "kepada" dari

komposisi form email. Dalam situasi ini, activity dari aplikasi email dijalankan, dan ketika user selesai, akan kembali ke activity dari aplikasi kita.

Mengatur Siklus Hidup Activity

Mengatur siklus hidup activity dengan menerapkan callback method (metode pemanggilan kembali) sangatlah krusial dalam pengembangan aplikasi yang kuat dan fleksibel. Siklus hidup dari sebuah activity terpengaruh secara langsung oleh hubungan activity tersebut dengan activity lain, tugasnya (task), dan tumpukan pada memori (back stack).

Sebuah activity bisa exist dalam tiga keadaan pokok, yaitu:

a. Resumed (Dijalankan)

Activity sedang berada pada bagian terdepan (foreground) layar, dan mendapatkan fokus dari user. Keadaan ini terkadang juga dianggap "running" atau sedang berjalan.

b. Paused (Didiamkan)

Activity lain sedang berada di foreground dan activity tersebut mendapatkan fokus, tapi activity yang ini masih bisa dilihat. Dalam hal ini, activity lain sedang terlihat dan berada di atas activity yang ini dan activity tersebut memiliki layout transparan atau tidak menutupi seluruh layar. Activity yang di-paused masih hidup seutuhnya (object dari activity tetap tinggal di memori, tetap menjaga semua keadaan dan informasi yang dimiliki, dan tetap tersambung dengan window manager), tapi bisa di-kill oleh sistem pada situasi ketika memori yang tersedia sangat kurang.

c. Stopped (Dihentikan)

Activity ini sedang dihalangi sepenuhnya oleh activity lain (activity sekarang berada di background). Activity yang di-stop juga masih hidup, object activity masih tersimpan di memori, masih menjaga keadaan dan semua informasi, tapi tidak tersambung dengan window manager. bagaimanapun juga, activity ini tidak lagi terlihat oleh user dan bisa di-kill oleh sistem sewaktu-waktu jika memori dibutuhkan untuk proses lain.

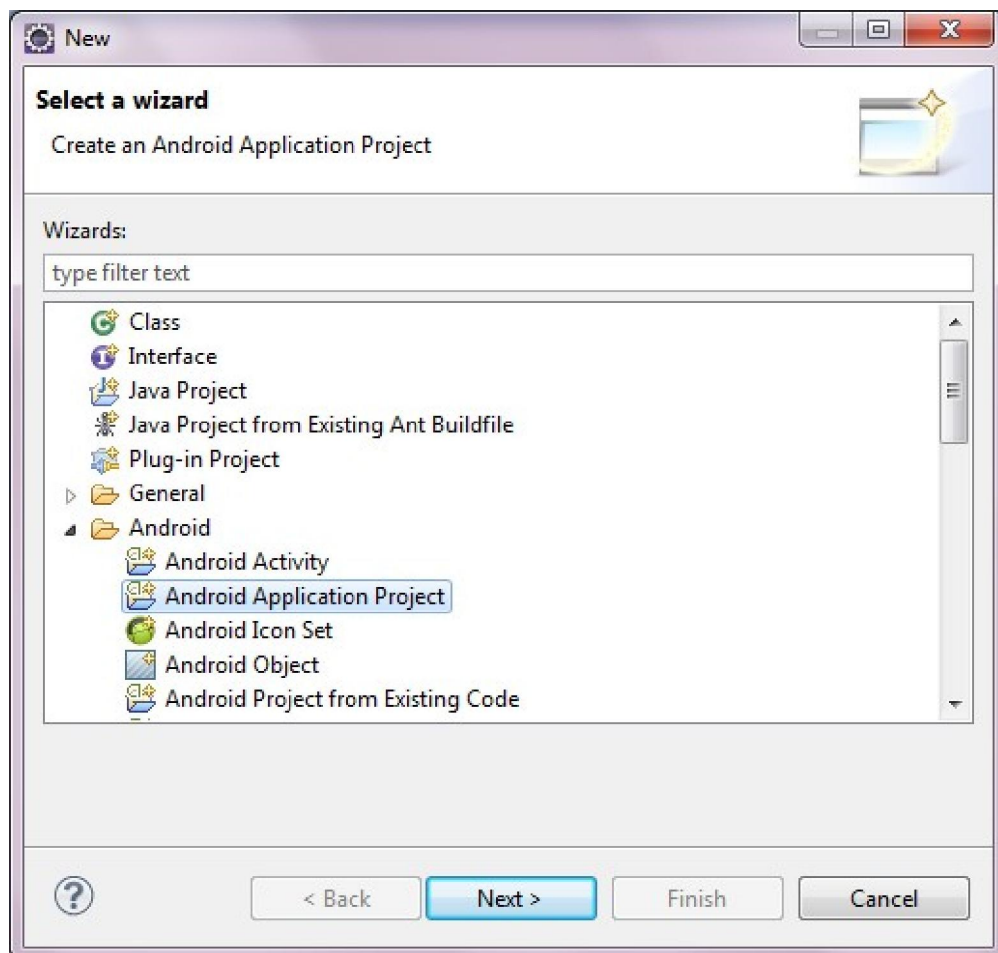
Jika sebuah activity di-stop atau di-pause, sistem menghapusnya dari memori dengan memintanya untuk berhenti (dengan memanggil method finish()), atau cukup

mematikan prosesnya. Ketika activity ingin dimulai lagi (setelah dihentikan atau dimatikan), maka activity tersebut harus di-create dari awal lagi.

Tujuan Pembelajaran 3.2:

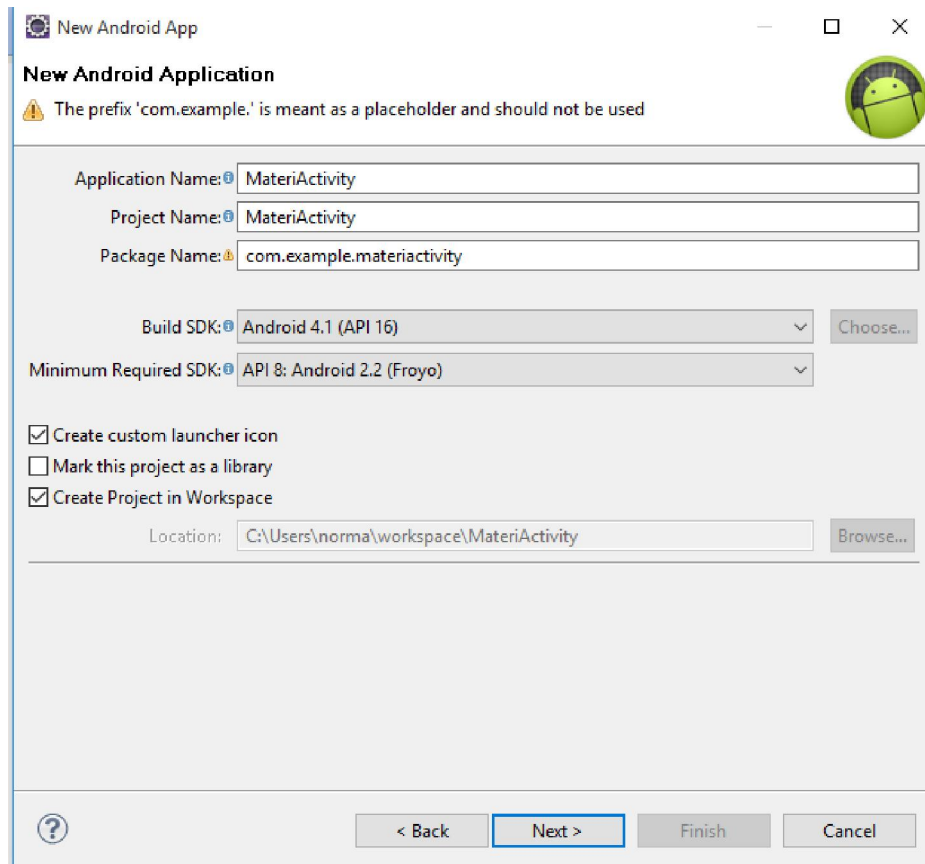
Membuat project dengan memanfaatkan activity

Untuk membuat project baru, klik menu File >> New >> Other... sehingga tampil kotak dialog seperti gambar 3.1 berikut ini:



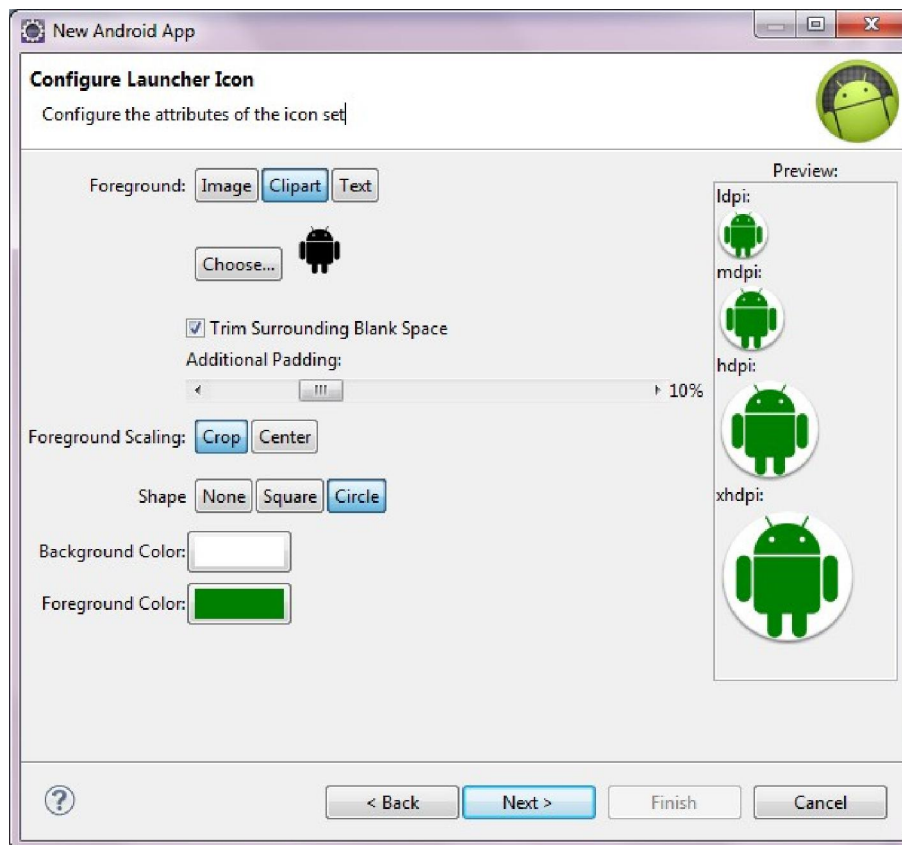
Gambar 3. 1 Kotak dialog membuat project baru

Pilih Android Application Project di dalam folder Android, kemudian klik tombol Next. Setelah tampil kotak dialog pengaturan seperti gambar 3.2, beri nama aplikasi “MateriActivity”, dan pada nama package ubah “example” menjadi “unpam”. Tentukan versi android SDK (Software Development Kit) untuk aplikasi yang akan dibuat pada menu dropdown Build SDK, dan minimum required SDK.

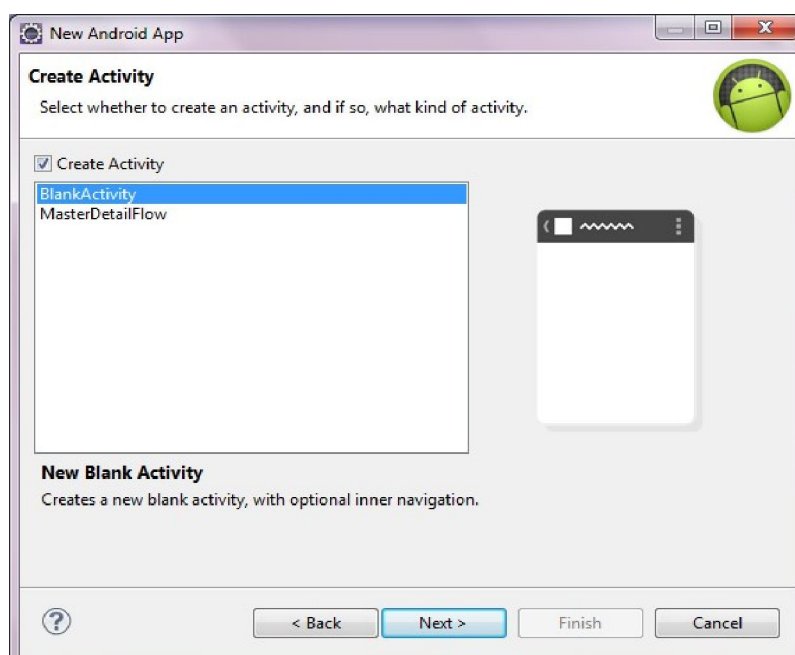


Gambar 3. 2 Kotak dialog pengaturan project

Setelah pengaturan nama aplikasi, nama project, nama package, dan SDK dari aplikasi, klik tombol Next untuk melanjutkan sehingga tampil kotak dialog seperti gambar 3.3. Atur icon dari aplikasi yang akan dibuat, bisa menggunakan image, clipart, maupun text. Atur bentuk dan warna dari icon jika diperlukan, jika sudah selesai lanjutkan dengan mengklik tombol Next.

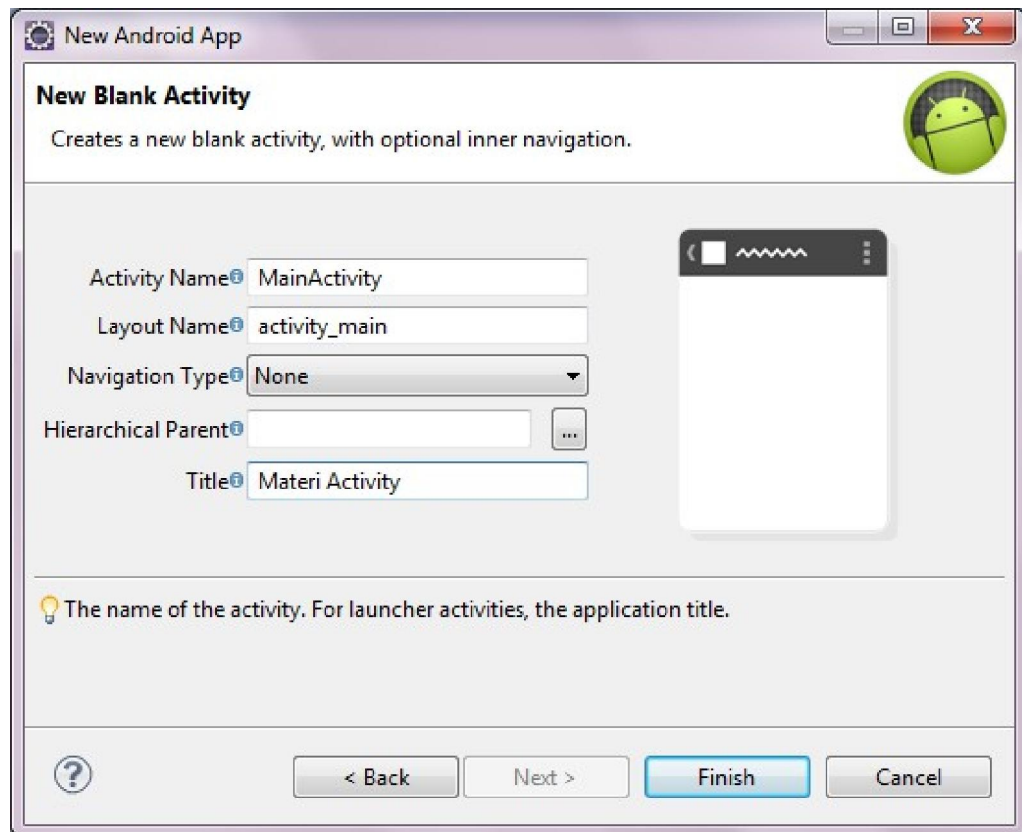


Gambar 3. 3 Pengaturan icon aplikasi



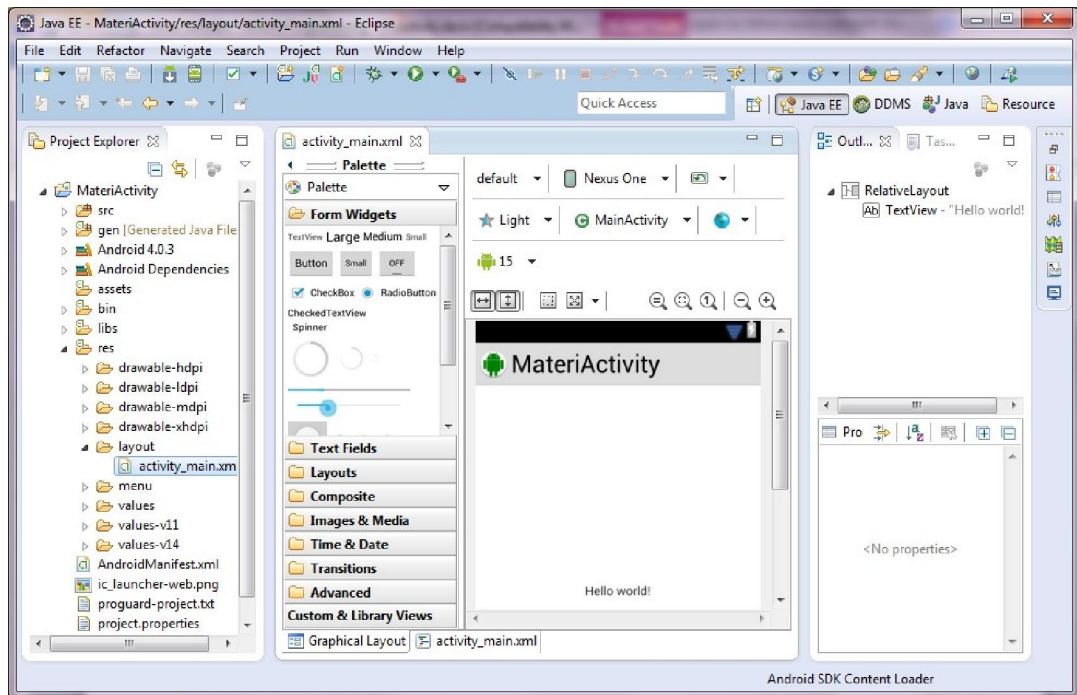
Gambar 3. 4 Memilih jenis Activity

Pilih Blank Activity untuk jenis Activity dari aplikasi yang akan dibuat. Lanjutkan dengan mengklik tombol Next. Ubah judul (Title) aplikasi menjadi “Materi Activity”.



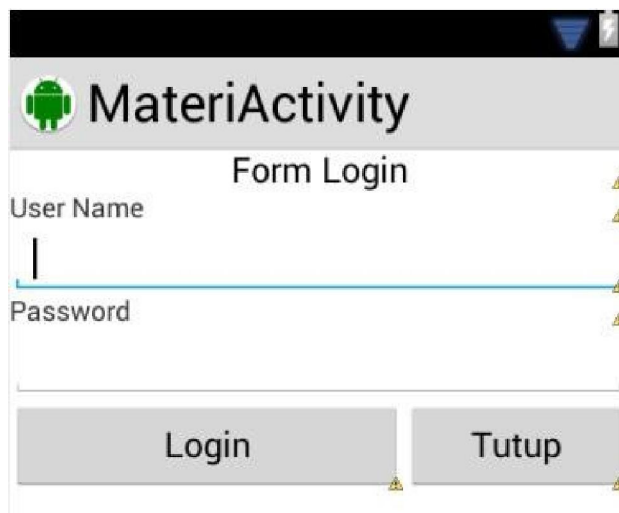
Gambar 3. 5 Mengatur nama dan judul activity

Klik tombol Finish untuk mengakhiri pengaturan pembuatan aplikasi. Tunggu beberapa saat sampai tampil desain aplikasi seperti gambar 3.6. Jika jendela yang tampil tidak sama, dapat diatur dari menu Window >> Show View dan pilih jendela yang ingin ditampilkan. Jika ingin mengubah tampilan ke bentuk standar, dapat dilakukan melalui menu Window >> Reset Perspective.



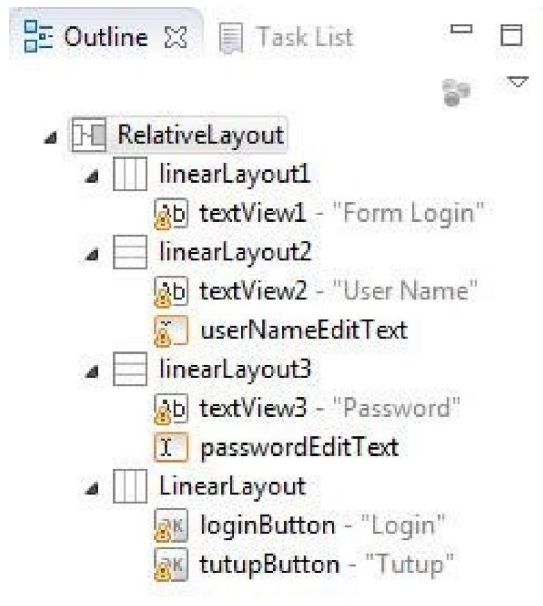
Gambar 3. 6 Tampilan IDE aplikasi android

Hapus TextView (teks “Hello Word!”), tambahkan LinearLayout (Horizontal) di sudut kiri atas, kemudian di dalamnya tambahkan Medium. Tambahkan komponen TextView dan EditText di dalam linear layout (vertical), Tambahkan komponen TextView dan EditText (Password) di dalam linear layout (vertical), dan dua buah Button di dalam linear layout (horizontal). Atur desain tampilan menjadi seperti gambar 3.7 dan desain outline menjadi seperti gambar 3.8.



Gambar 3. 7 Graphical layout activity_main.xml

Ubah ID dari EditText dan Button dengan cara klik kanan >> EditID, sehingga outline menjadi seperti berikut ini:



Gambar 3. 8 Outline activity_main.xml

Desain activity dapat juga diatur melalui source code XML sebagai berikut:

```

<RelativeLayoutxmlns: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">

    <LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_alignParentRight="true"
        android:layout_alignParentTop="true">

        <TextView
            android:id="@+id/textView1"android:layout_width="match_
            parent"android:layout_height="wrap_content"android:gravit
            y="center"
            android:text="Form Login"
            android:textAppearance="?android:attr/textAppearanceMedium"/>
        </LinearLayout>

    <LinearLayout
        android:id="@+id/linearLayout2"
        android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_alignParentRight="true"android:layout_below="@+id/linearLayout1
    "

```

```
android:orientation="vertical">

<TextView
    android:id="@+id/textView2"android:layout_width="match_parent"android:layout_height="wrap
android:text="User Name"/>

<EditText
    android:id="@+id/userNameEditText" android:layout_width="match_parent"
android:layout_height="wrap_content">

<requestFocus/>
</EditText>
</LinearLayout>

<LinearLayout
android:id="@+id/linearLayout3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"android:layout_alignParentRight="true"android:layout_belo
android:orientation="vertical">

<TextView
android:id="@+id/textView3"
android:layout_width="match_parent"android:layout_height="wrap_content"
android:text="Password"/>

<EditText
    android:id="@+id/passwordEditText"android:layout_width="match_parent"
android:layout_height="wrap_content"android:inputType="textPassword"/>
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_alignParentLeft="true"
```

```
    android:layout_alignParentRight="true" android:layout_below="@+id/linearLayout3">
```

```
<Button
```

```
    android:id="@+id/loginButton" android:layout_width="wrap_content" android:layout_height="wrap_content"
```

```
    android:text="Login"/>
```

```
<Button
```

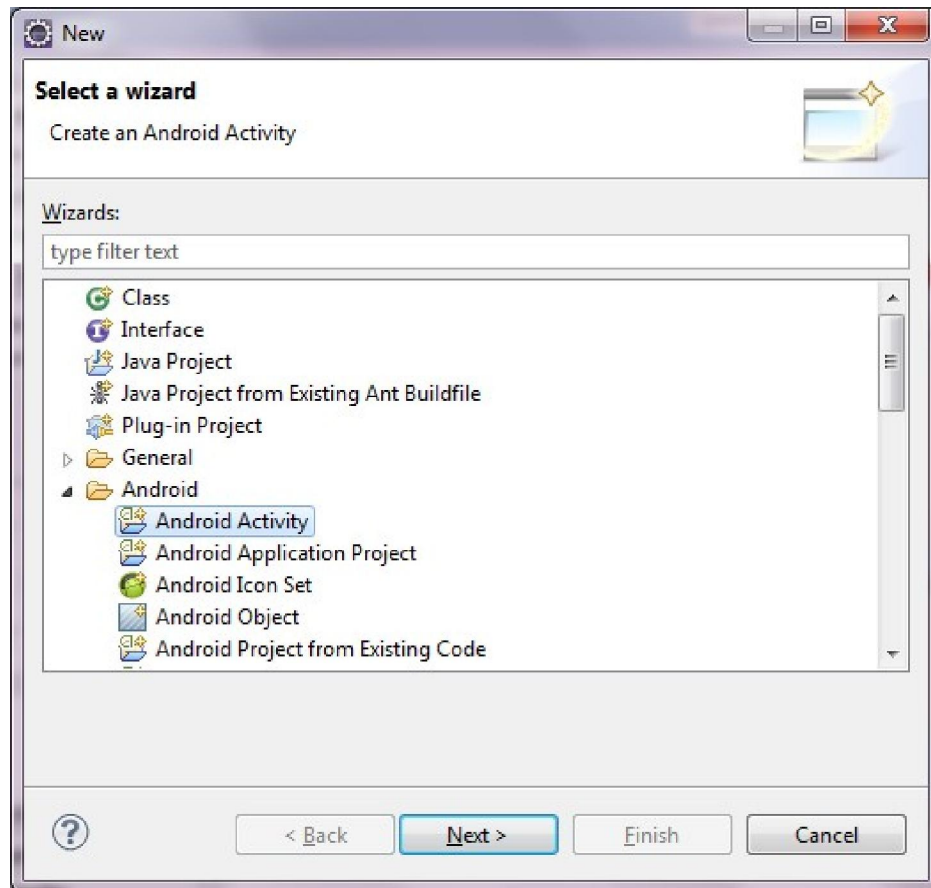
```
    android:id="@+id/tutupButton" android:layout_width="wrap_content" android:layout_height="wrap_content"
```

```
    android:text="Tutup"/>
```

```
</LinearLayout>
```

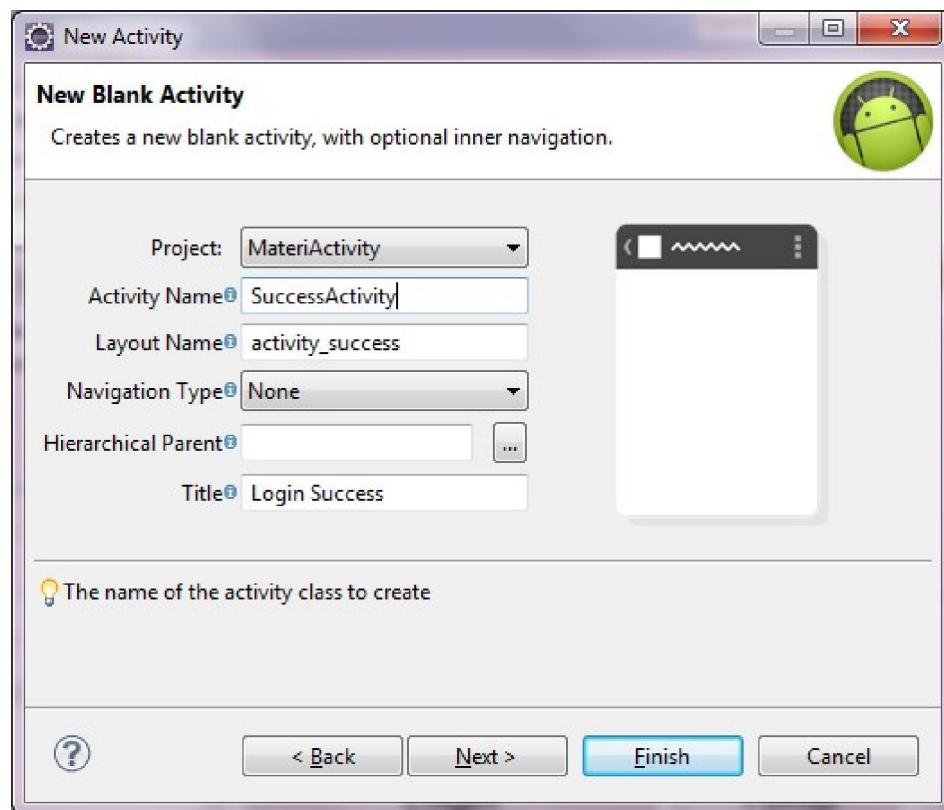
```
</RelativeLayout>
```

Selanjutnya tambahkan activity baru dengan cara klik kanan pada project, pilih New dan Other (atau Ctrl+N) sehingga tampil kotak dialog seperti gambar 3.9 berikut ini:



Gambar 3. 9 Menambahkan activity baru

Klik tombol Next, pada kotak dialog seperti gambar 3.10 ubah nama activity menjadi “SuccessActivity” dan ubah judul menjadi “Login Success”.

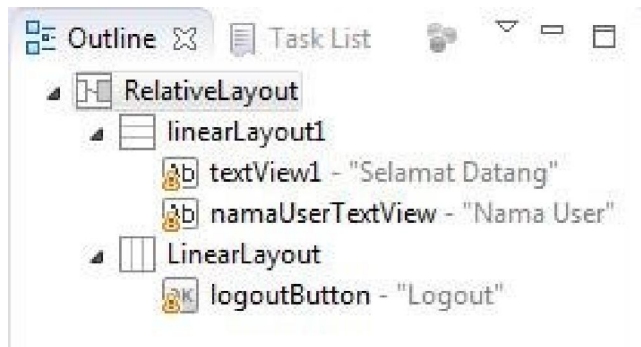


Gambar 3. 10 Pengaturan activity

Klik tombol Finish untuk mengakhiri pengaturan nama dan judul Activity. Selanjutnya atur layout `activity_success` menjadi seperti gambar 3.11 dan 3.12 berikut ini:



Gambar 3. 11 Graphical layout `activity_success.xml`



Gambar 3. 12 Outline activity_success.xml

Source code activity_success.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">
```

```
<LinearLayout
    android:id="@+id/linearLayout1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:orientation="vertical">
```

```
<TextView
    android:id="@+id/textView1"
    android:layout_width="match_parent" android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Selamat Datang"/>
```

```
<TextView
    android:id="@+id/namaUserTextView" android:layout_width="match_parent" android:layout_height="wrap_content"
    android:text="Nama User"
    android:textAppearance="?android:attr/textAppearanceMedium"/>
</LinearLayout>
```

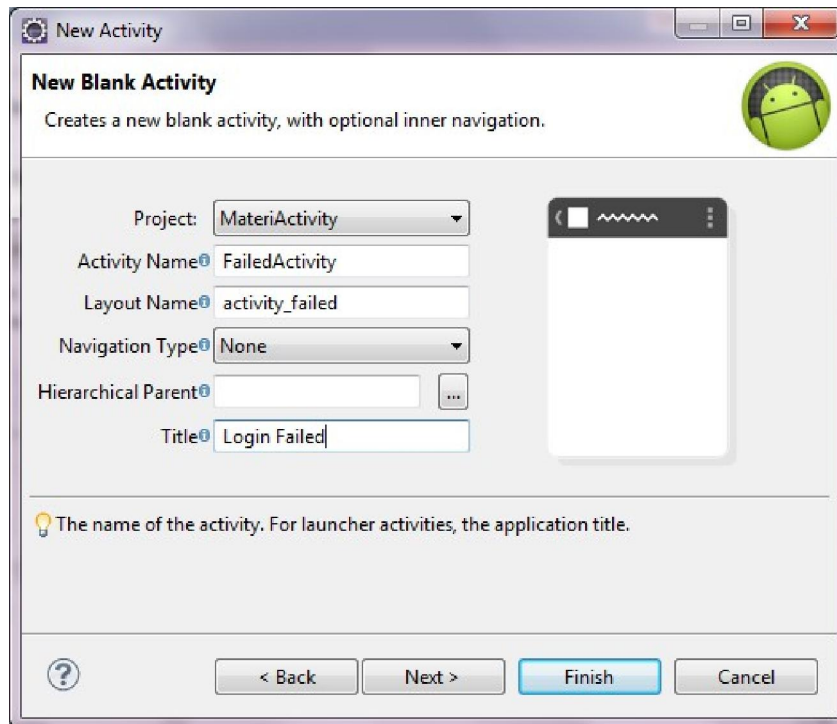
```
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
```

```
        android:layout_alignParentRight="true"android:layout_below="@+id/linearLayout1">

<Button
    android:id="@+id/logoutButton"
        android:layout_width="wrap_content"android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Logout"/>
</LinearLayout>

</RelativeLayout>
```

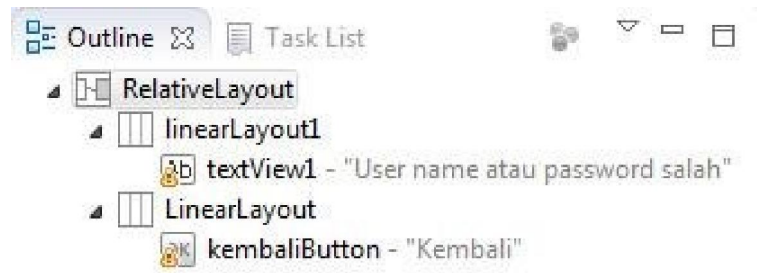
Dengan cara yang sama tambahkan activity_failed.xml.



Gambar 3. 13 Pengaturan activity



Gambar 3. 14 Graphical layout activity_failed.xml



Gambar 3. 15 Outline activity_failed.xml

Source code activity_failed.xml:

```
<RelativeLayoutxmlns: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">

<LinearLayout
android:id="@+id/linearLayout1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"android:layout_alignParentRight="true"
android:layout_alignParentTop="true">

<TextView
android:id="@+id/textView1"android:layout_width="match_parent"android:layout_height="wrap_co
android:text="User name atau password salah"
android:textAppearance="?android:attr/textAppearanceMedium"/>

</LinearLayout>

<LinearLayout
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"android:layout_below="@+id/linearLayout1">

<Button
    android:id="@+id/kembaliButton"android:layout_width="wrap_content"android:layout_heigh
android:layout_weight="1"
android:text="Kembali"/>
```

```
</LinearLayout>
```

```
</RelativeLayout>
```

Setelah selesai mengatur desain activity, selanjutnya menambahkan source code java. Pertama double klik file FailedActivity.java yang berada di project dalam folder src >> com.unpam.materiactivity, sehingga menjadi seperti berikut ini:

```
package com.unpam.materiactivity;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;

publicclass FailedActivity extends Activity implements OnClickListener{

    @Override
    publicvoid onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_failed);

        findViewById(R.id.kembaliButton).setOnClickListener(this);
    }

    @Override
    publicboolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_failed, menu);
        returntrue;
    }

    publicvoid onClick(View v) {
        // TODO Auto-generated method stub
        switch (v.getId()){
        case R.id.kembaliButton:
            finish();
        }
    }
}
```

```
}  
}
```

Kedua, dengan cara yang sama tambahkan source code untuk file `SuccessActivity.java` menjadi:


```
package com.unpam.materiaactivity;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.TextView;

publicclass SuccessActivity extends Activity implements
OnClickListener{

    @Override
    publicvoid onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_success);

        TextView namaUserTV = (TextView) findViewById(R.id.namaUserTextView);
        namaUserTV.setText (getIntent().getExtras().getString("namaUser"));

        findViewById(R.id.logoutButton).setOnClickListener(this);
    }

    @Override
    publicboolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_success, menu);
        returntrue;
    }

    publicvoid onClick(View v) {
        // TODO Auto-generated method stub
    }
}
```

```
        switch (v.getId()){  
            case R.id.logoutButton:  
                finish();  
            }  
        }  
    }  
}
```

Ketiga, tambahkan source code di dalam file MainActivity.java sehingga menjadi seperti berikut ini:

```
package com.unpam.materiactivity;

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.EditText;

publicclass MainActivity extends Activity implements OnClickListener{
private String[][] dataUser = {{ "ADMIN","ADMIN","Administrator"},
                                {"user1","password1","Nama User 1"},
                                {"user2","password2","Nama User 2"} };
private EditText userNameET, passwordET;
privatefinalintSUCCESS=1;

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

        userNameET = (EditText) findViewById(R.id.userNameEditText);
        passwordET = (EditText) findViewById(R.id.passwordEditText);

        findViewById(R.id.loginButton).setOnClickListener(this);
        findViewById(R.id.tutupButton).setOnClickListener(this);
    }

@Override
```

```
public boolean onCreateOptionsMenu(Menu menu) {    getMenuInflater().inflate(R.menu.activity_m  
menu);  
return true;  
}
```

```
public void onClick(View v) {  
    // TODO Auto-generated method stub  
    switch (v.getId()) {  
        case R.id.loginButton:  
            int i=0;  
            boolean success=false;  
  
            while ((i<dataUser.length) && (!success)) {  
                if  
                    ((userNameET.getText().toString().equals(dataUser[i][0])) &&  
                     (passwordET.getText().toString().equals(dataUser[i][1]))) {  
                        success=true;  
                    } else {  
                        i++;  
                    }  
            }  
  
            Intent intent;  
  
            if (success) {  
                intent = new Intent(this, SuccessActivity.class);  
                intent.putExtra("nama",  
dataUser[i][2]);  
startActivityForResult(intent,SUCCESS);  
            } else {  
                intent = new Intent(this, FailedActivity.class);  
                startActivity(intent);  
            }  
            startActivity(intent);  
        }  
    }  
}
```

```
                break;                case
R.id.tutupButton:
    System.exit(0);
    }
}

@Override
public void onActivityResult(int requestCode, int resultCode, Intent data){
    switch (requestCode) {                case SUCCESS:
        userNameET.setText("");                passwordET.setText("");
        break;                }
    }
}
```

Tujuan Pembelajaran 3.3:

Menjalankan Aplikasi

Untuk mengeksekusi, klik kanan pada nama project atau pilih menu Run, kemudian pilih Run As >> Android Application.

Jika kita lihat di daftar aplikasi seperti pada gambar 3.16, maka akan terlihat 3 icon aplikasi baru, yaitu Materi Activity, Login Success, dan Login Failed. Agar aplikasi yang kita buat hanya menampilkan satu icon aplikasi, double klik file AndroidManifest.xml di dalam project seperti pada gambar 3.17, kemudian klik tab AndroidManifest.xml yang berada pada bagian bawah seperti pada gambar 3.18, hapus tag intent-filter untuk SuccessActivity dan FailedActivity sehingga source code-nya menjadi seperti berikut ini:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.unpam.materiactivity" android:versionCode="1"
    android:versionName="1.0">

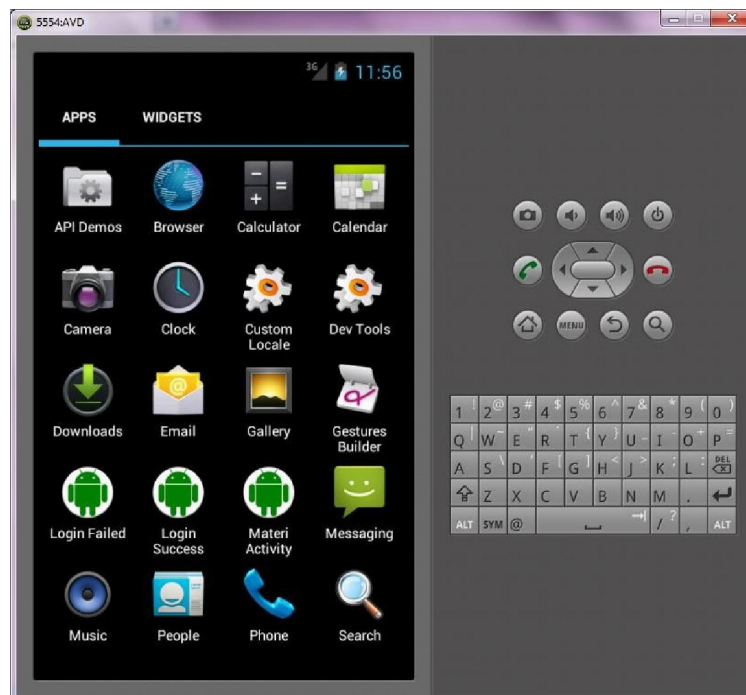
    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="15"/>

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">

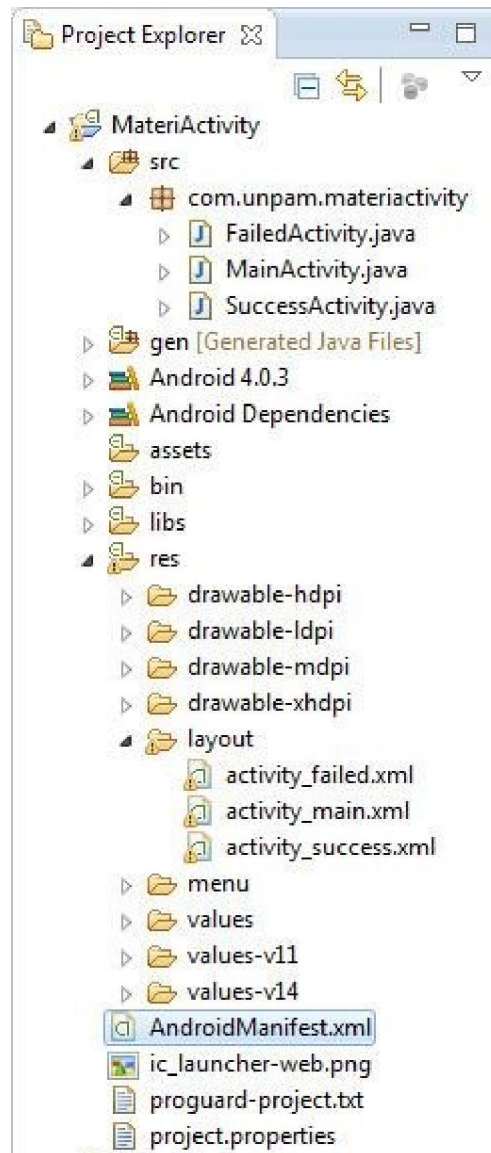
        <activity
            android:name=".MainActivity"
            android:label="@string/title_activity_main">

            <intent-filter>
```

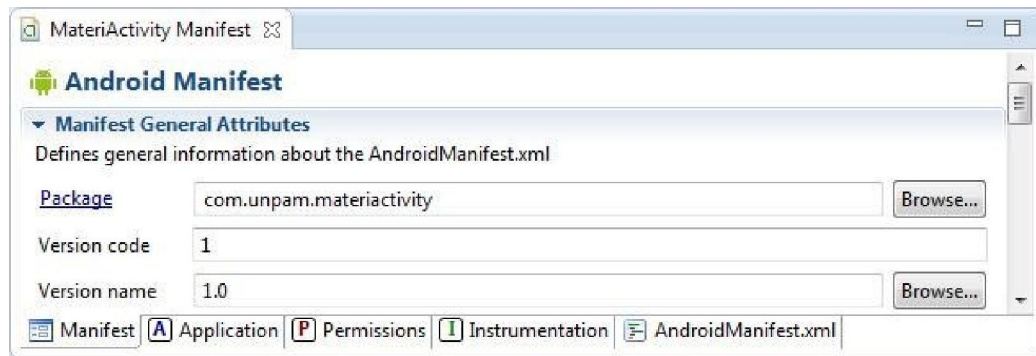
```
<actionandroid:name="android.intent.action.MAIN"/>
<categoryandroid:name="android.intent.category.LAUNCHER"/>
</intent-filter>
</activity>
<activity
android:name=".SuccessActivity"
android:label="@string/title_activity_success">
</activity>
<activity
android:name=".FailedActivity"
android:label="@string/title_activity_failed">
</activity>
</application>
</manifest>
```



Gambar 3. 16 Tampilan daftar aplikasi



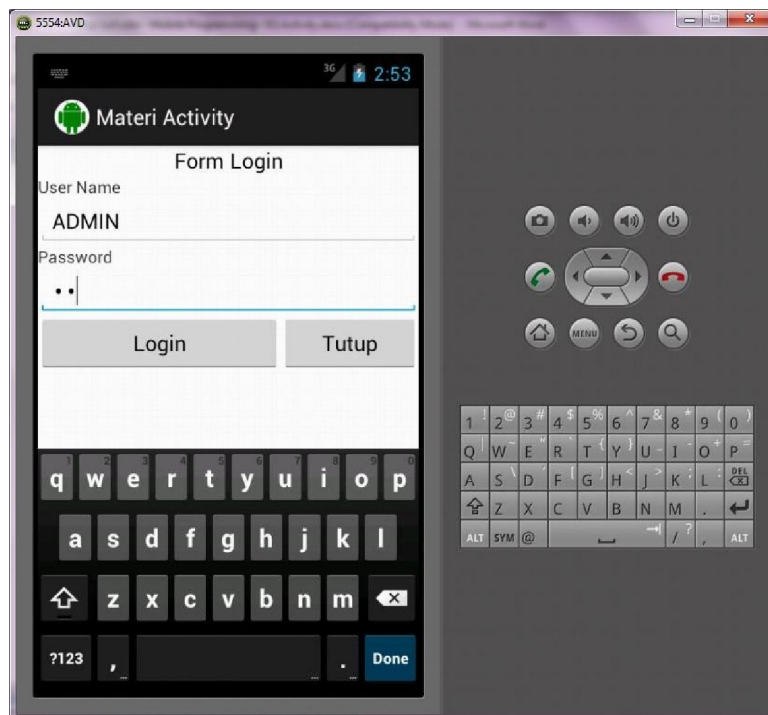
Gambar 3. 17 Project explorer



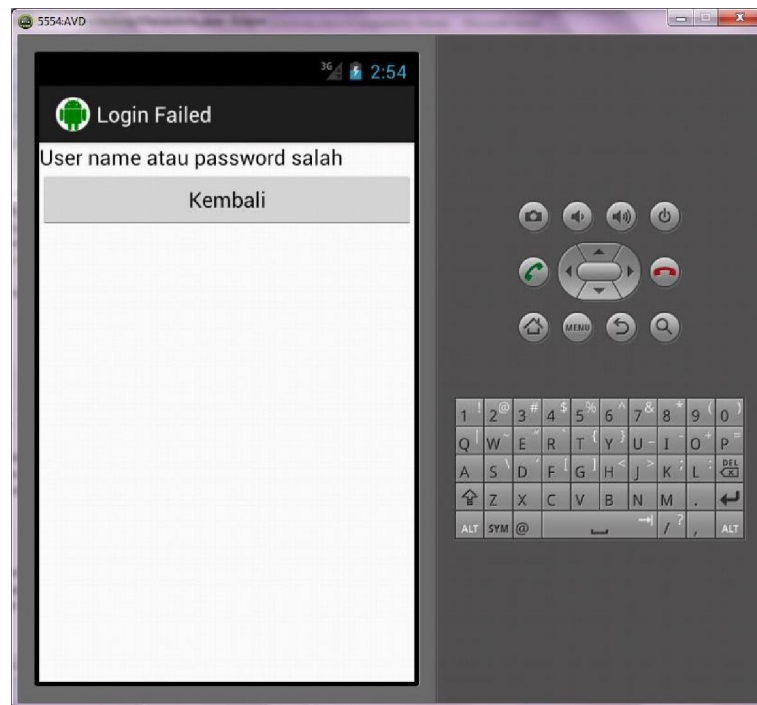
Gambar 3. 18 Android manifest

Tampilan Aplikasi

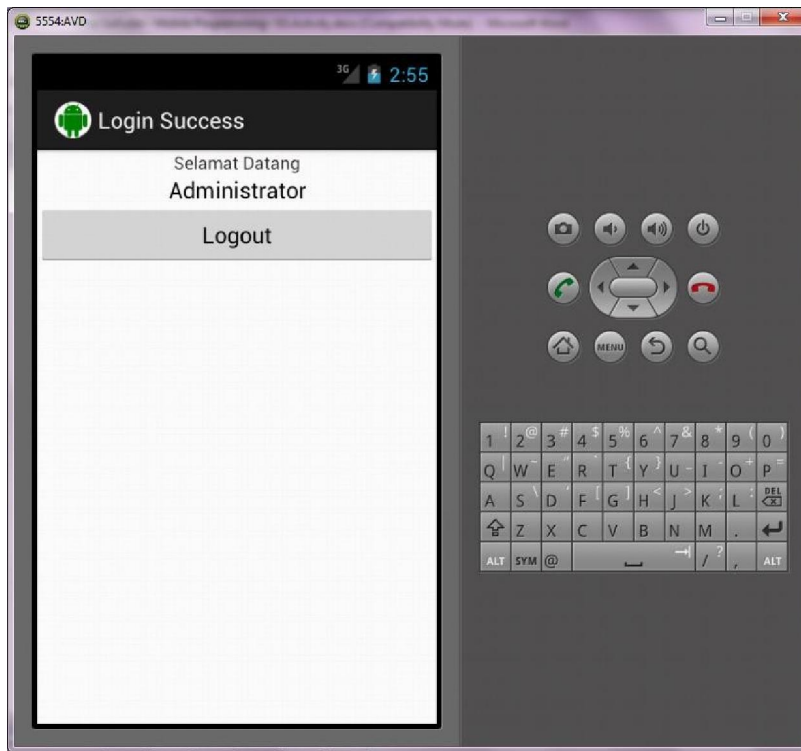
Tampilan aplikasi yang telah dibuat pada AVD (Android Virtual Device) adalah sebagai berikut:



Gambar 3. 19 Tampilan MainActivity di AVD



Gambar 3. 20 Tampilan FailedActivity di AVD



Gambar 3. 21 Tampilan SuccessActivity di AVD

C. LATIHAN SOAL/TUGAS

Buatlah halaman login yang menampilkan halaman lainnya seperti halaman utama, contoh login email yang menampilkan halaman inbox!

D. DAFTAR PUSTAKA

Allen, Grant. 2012. Beginning Android 4. New York : Apress.

Safaat, H. Nazruddin. 2015. ANDROID Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Bandung: Informatika