

Lampiran

Login

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/activity_main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"

tools:context="com.caratutorial.aplikasiloginsederhana.MainActivity"
>

<TextView
    android:text="Login Form"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:id="@+id/textView"
    android:textAlignment="center"
    android:textSize="30sp"
    android:textStyle="normal|bold" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPersonName"
    android:ems="10"
    android:id="@+id/username"
    android:hint="Username"
    android:layout_below="@+id/textView"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginTop="37dp" />
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:ems="10"
    android:layout_below="@+id/username"
    android:layout_alignParentLeft="true"
    android:id="@+id/password"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:hint="Password" />
```

```
<Button
    android:text="Log In"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/password"
    android:layout_centerHorizontal="true"
    android:id="@+id/btnLogin" />
```

```
</RelativeLayout>
```

MainActivity.java

```
package com.caratutorial.aplikasiloginsederhana;
```

```
import android.content.Intent;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    EditText username, password;
    Button btnLogin;
```

```

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

    username = (EditText) findViewById(R.id.username);
    password = (EditText) findViewById(R.id.password);
    btnLogin = (Button) findViewById(R.id.btnLogin);

    btnLogin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            String usernameKey = username.getText().toString();
            String passwordKey = password.getText().toString();

            if (usernameKey.equals("admin") &&
passwordKey.equals("123")){
                //jika login berhasil
                Toast.makeText(getApplicationContext(), "LOGIN
SUKSES",
                    Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(MainActivity.this,
Welcome.class);
                MainActivity.this.startActivity(intent);
                finish();
            }else {
                //jika login gagal
                AlertDialog.Builder builder = new
AlertDialog.Builder(MainActivity.this);
                builder.setMessage("Username atau Password Anda
salah!")
                    .setNegativeButton("Retry", null).create().show();
            }
        }
    });
}
}

```

Welcome.java

```
package com.caratutorial.aplikasiloginsederhana;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class Welcome extends AppCompatActivity {

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

        btnExit = (Button) findViewById(R.id.btnExit);

        btnExit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //perintah untuk mengakhiri aplikasi
                finish();
            }
        });
    }
}
```

AndroidManifest.xml

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

<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.root.peta">
```

```
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />

<uses-permission android:name="android.permission.INTERNET"
/>

<uses-permission
android:name="com.google.android.providers.gsf.permission.READ_
GSERVICES" />
```

```
<!--
```

The ACCESS_COARSE/FINE_LOCATION permissions are not required to use

Google Maps Android API v2, but you must specify either coarse or fine

location permissions for the 'MyLocation' functionality.

```
-->
```

```
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"/>

<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
```

```
<application
```

```
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
```

```
<!--
```

The API key for Google Maps-based APIs is defined as a string resource.

(See the file "res/values/google_maps_api.xml").

Note that the API key is linked to the encryption key used to sign the APK.

You need a different API key for each encryption key, including the release key that is used to sign the APK for publishing.

You can define the keys for the debug and release targets in src/debug/ and src/release/.

-->

```
<meta-data
```

```
    android:name="com.google.android.geo.API_KEY"
```

```
    android:value="@string/google_maps_key" />
```

```
<activity
```

```
    android:name=".MapsActivity"
```

```
    android:label="@string/title_activity_maps">
```

```
    <intent-filter>
```

```
        <action android:name="android.intent.action.MAIN" />
```

```
        <category
```

```
            android:name="android.intent.category.LAUNCHER" />
```

```
    </intent-filter>
```

```
</activity>
```

```
</application>
```

```
</manifest>
```

Maps.java

```
package com.example.root.peta;

import android.support.v4.app.FragmentActivity;
import android.os.Bundle;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public class MapsActivity extends FragmentActivity implements
    OnMapReadyCallback {

    private GoogleMap mMap;

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

        // Obtain the SupportMapFragment and get notified when the
        map is ready to be used.

        SupportMapFragment mapFragment = (SupportMapFragment)
            getSupportFragmentManager()
```



```

        .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }

    /**
     * Manipulates the map once available.
     * This callback is triggered when the map is ready to be used.
     * This is where we can add markers or lines, add listeners or move
    the camera. In this case,
     * we just add a marker near Sydney, Australia.
     * If Google Play services is not installed on the device, the user will
    be prompted to install
     * it inside the SupportMapFragment. This method will only be
    triggered once the user has
     * installed Google Play services and returned to the app.
    */
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        // Add a marker in Sydney and move the camera
        LatLng sydney = new LatLng(-34, 151);
        mMap.addMarker(new
    MarkerOptions().position(sydney).title("Marker in Sydney"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
    }

```

```
}
```

activity_maps.xml

```
<fragment android:id="@+id/map"
```

```
android:name="com.google.android.gms.maps.SupportMapFragment"
```

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

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

```
xmlns:tools="http://schemas.android.com/tools"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```

```
tools:context="com.abhiandroid.GoogleMaps.googlemaps.MapsActivity"/>
```

```
@Override
```

```
public void onMapReady(GoogleMap googleMap) {
```

```
    mMap = googleMap;
```

```
    mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
```

```
    mMap.getUiSettings().setZoomControlsEnabled(true);
```

```
    mMap.getUiSettings().setZoomGesturesEnabled(true);
```

```
    mMap.getUiSettings().setCompassEnabled(true);
```

```
    //Initialize Google Play Services
```

```
    if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
```

```
        if (ContextCompat.checkSelfPermission(this,
```

```
            Manifest.permission.ACCESS_FINE_LOCATION)
```

```
            == PackageManager.PERMISSION_GRANTED) {
```

```
        buildGoogleApiClient();
        mMap.setMyLocationEnabled(true);
    }
} else {
    buildGoogleApiClient();
    mMap.setMyLocationEnabled(true);
}
}
```