

Menampilkan Peta Dengan Google Maps API Pada Android Studio

Persiapan

Beberapa hal yang perlu disiapkan antara lain :

1. Android Studio.
2. Perangkat android untuk uji coba aplikasi yang akan dibuat, pastikan perangkat terhubung dengan internet dan aktifkan akses lokasi.

Membuat Project Baru

1. Buka Android Studio dan buat Project baru dengan nama "Peta", lalu klik next.
2. Pilih versi SDK android minimal, disini dipilih Android 4.1, lalu klik next.
3. Pada opsi "Add an Activity to mobile" pilih "Google maps activity", lalu klik next.
4. Tinggalkan nama activity, lalu klik Finish.

Mendaftar Maps API

1. Buka google_maps_api.xml pada package res/value, di google_maps_api.xml terdapat alamat untuk mengaktifkan Maps API, disini alamatnya adalah:
Copy alamat tersebut lalu buka dalam browser, bila perlu daftarkan akun gmail untuk konfigurasi API.
2. Setelah halama google API terbuka, pilih "Tidak" untuk opsi dapatkan email, dan pilih "Ya" untuk opsi setuju dengan Syarat dan ketentuan, lalu klik setuju dan lanjutkan.
3. Setelah halaman "API diaktifkan" terbuka, klik buat kunci API.
4. Setelah terbuka halaman "Kredensial" lalu muncul kotak dialog "Kunci API Dibuat" copy kode API key yang ada didalam box "Kunci API Anda".
5. Buka kembali google_maps_api.xml pada package res/value di Android Studio, lalu masukan kode API key diantara <string name="google_maps_key" templateMergeStrategy="preserve" translatable="false"> dan </string> menggantikan text YOUR_KEY_HERE,

Manifests dan build.gradle(Module:app)

Manifest

1. Buka AndroidManifest.xml pada package manifests.
2. Masukkan Permission yang diperlukan antara lain ACCESS_NETWORK_STATE Untuk memeriksa apakah perangkat terhubung ke suatu jaringan atau tidak, INTERNET Untuk memeriksa apakah perangkat terhubung ke Internet atau tidak, ACCESS_COARSE_LOCATION Untuk memperkirakan lokasi pengguna menggunakan WiFi atau sinyal mobile, ACCESS_FINE_LOCATION Untuk memperkirakan lokasi terkini pengguna, OpenGL ES V2 Dibutuhkan untuk Google Maps V2

sehingga AndroidManifest.xml menjadi :

```
3. <?xml version="1.0" encoding="utf-8"?>
4. <manifest xmlns:android="http://schemas.android.com/apk/res/android"
5.     package="com.example.root.peta">
6.
7.     <uses-permission
8.         android:name="android.permission.ACCESS_NETWORK_STATE" />
9.     <uses-permission android:name="android.permission.INTERNET" />
10.    <uses-permission
11.        android:name="com.google.android.providers.gsf.permission.READ_GSERVIC
12.        ES" />
13.    <!--
14.        The ACCESS_COARSE/FINE_LOCATION permissions are not required
15.        to use
16.        Google Maps Android API v2, but you must specify either coarse or fine
17.        location permissions for the 'MyLocation' functionality.
18.    -->
19.    <uses-permission
20.        android:name="android.permission.ACCESS_COARSE_LOCATION"/>
21.    <uses-permission
22.        android:name="android.permission.ACCESS_FINE_LOCATION" />
23.
24.    <application
25.        android:allowBackup="true"
26.        android:icon="@mipmap/ic_launcher"
27.        android:label="@string/app_name"
28.        android:supportsRtl="true"
29.        android:theme="@style/AppTheme">
```

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

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

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

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

30. sign the APK for publishing.

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

```

32. -->
33. <meta-data
34.     android:name="com.google.android.geo.API_KEY"
35.     android:value="@string/google_maps_key" />
36.
37. <activity
38.     android:name=".MapsActivity"
39.     android:label="@string/title_activity_maps">
40.     <intent-filter>
41.         <action android:name="android.intent.action.MAIN" />
42.
43.         <category android:name="android.intent.category.LAUNCHER" />
44.     </intent-filter>
45. </activity>
46. </application>
47.
48. </manifest>

```

maps.java

1. Buka MapsActivity.java pada package java, tidak perlu mengubah script tersebut namun pastikan MapsActivity.java terlihat seperti berikut

```

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));
    }
}

```

```
}
```

Pada script diatas telah ditambahkan koordinat Sydney Australia secara default di `onMapReady()`. Jadi saat aplikasi dijalankan pertama kali marker akan ditempatkan di Sydney Australia.