MODUL : GOOGLE MAPS API

**A. Tujuan**

Modul ini membahas mengenai Google Maps API, dengan tujuan agar mahasiswa:

* Mengetahui cara untuk mengakses Google Maps pada perangkat android
* Dapat mengimplementasikan Google Maps API pada pembuatan aplikasi android

**B. DasarTeori**

**1. Google Maps API**

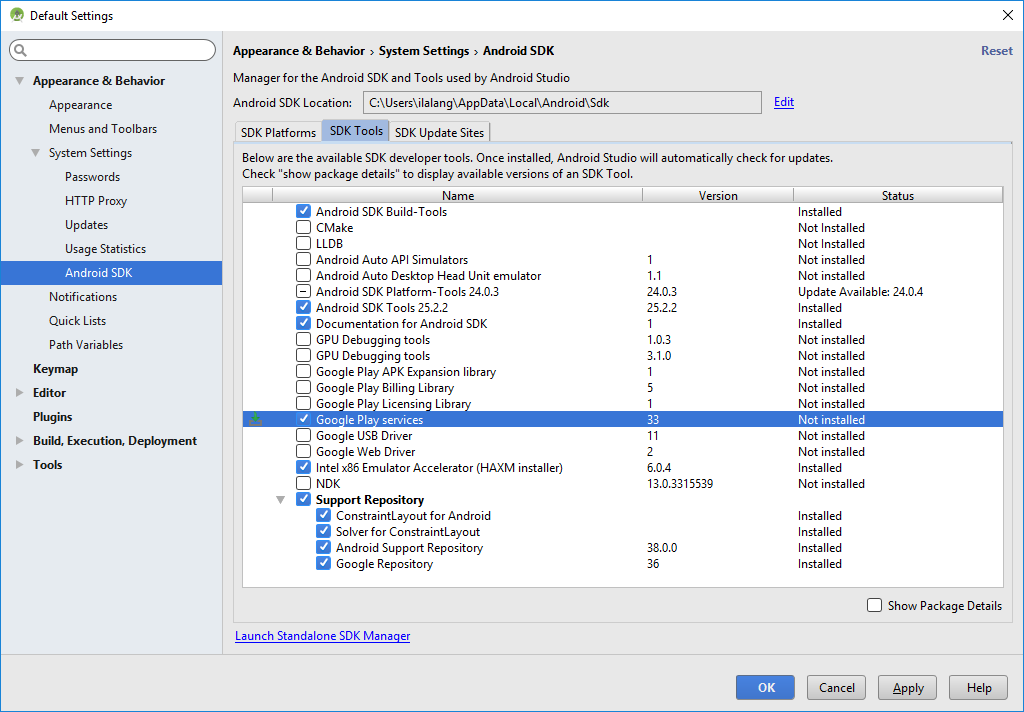
Google Maps merupakan aplikasi pemetaan yang dikembangkan oleh Google, dimana aplikasi ini dapat diimplementasikan pada desktop maupun perangkat mobile dengan berbagai platform, diantaranya Android, iOS, dan BlackBeery OS. Google menyediakan Google Maps API yang memungkinkan kita untuk mengintergrasikan peta yang berasal dari Google Maps ke dalam aplikasi mereka. API secara otomatis akan menangani akses ke server Google Maps, mengunduh data, menampilkan peta, dan merespon *map gesture*. Selain itu, kita dapat menggunakan API untuk menambahkan penanda (*marker*), poligon, dan lapisan (*overlay*) pada peta, serta mengubah user’s view pada wilayah tertentu dalam peta. Objek-objek tersebut digunakan untuk memberikan tambahan informasi lokasi pada peta, dan memungkinkan pengguna untuk berinteraksi langsung dengan peta.

Jika kita menggunakan Google Maps API pada aplikasi yang dikembangkan, kita harus menyertakan atribusi teks Google Play Service sebagai bagian dari “Legal Notices” pada aplikasi kita. Teks atribusi tersebut dapat diperoleh dengan memanggil: GoogleApiAvailability.getOpenSourceSoftwareLicenseInfo

**2. Praktikum Google Maps API**

1. Sebelum membuat project baru, kita perlu menginstall Google Play Service SDK melalui SDK Manager. Kita dapat membuka SDK Manager melalui Android Studio dengan cara:

Tools -> Android -> SDK Manager atau dapat juga dengan memilih icon SDK Manager  pada toolbar. Setelah SDK Manager terbuka, pilihlah tab SDK Tools -> pilih Google Play Services -> klik tombol OK, maka proses pengunduhan dan instalasi akan dimulai.

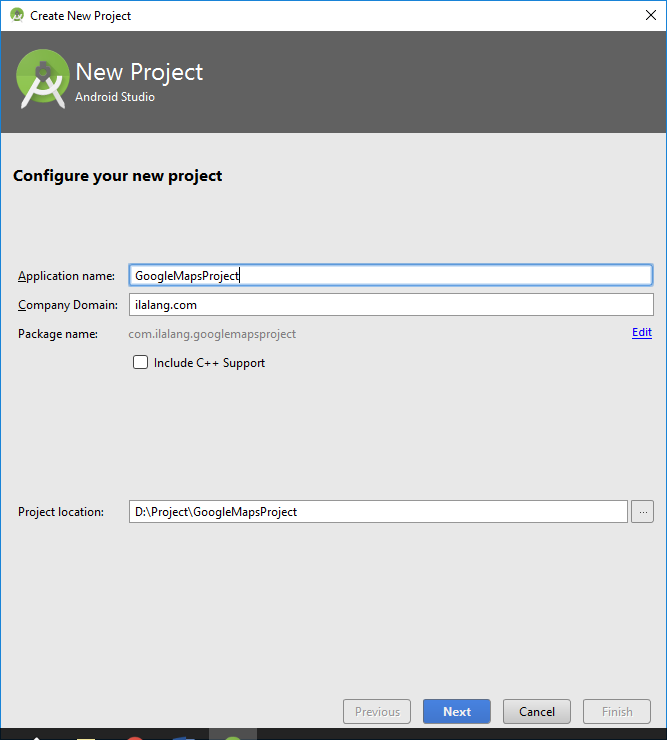
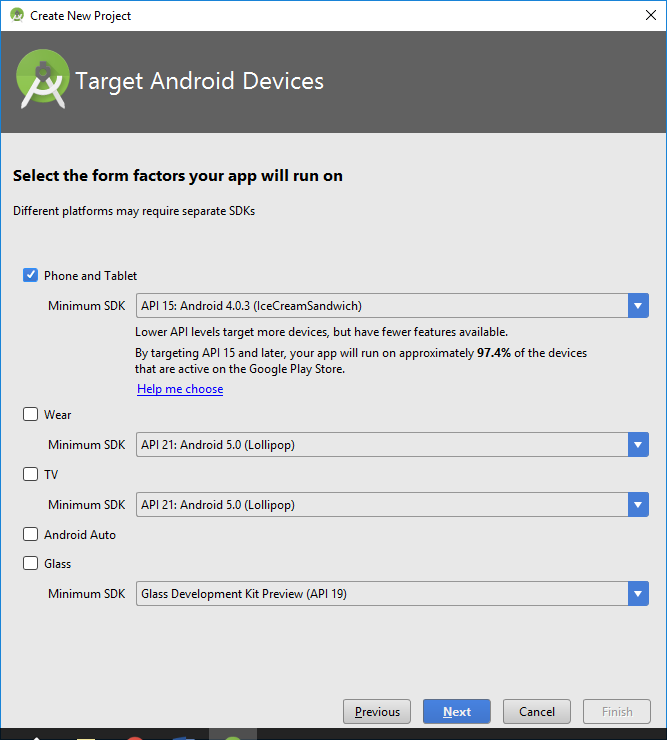
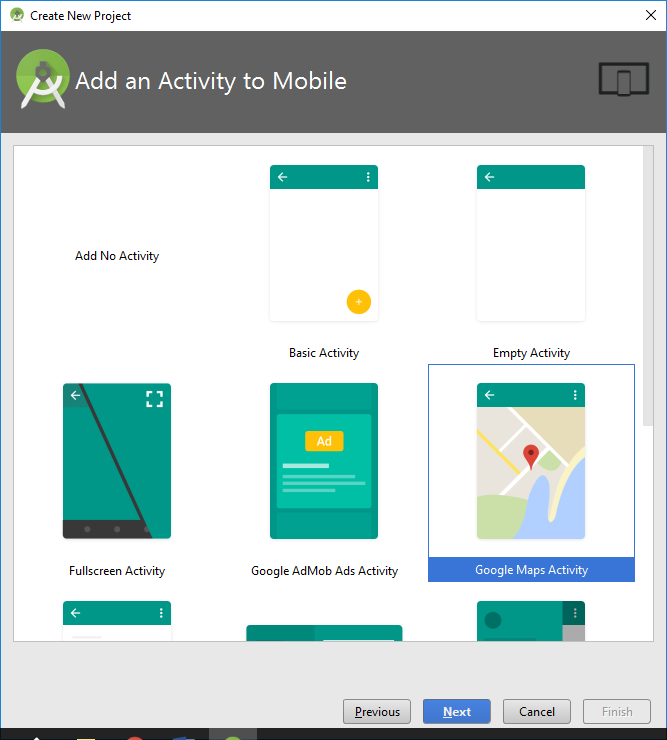


**Gambar 1**. Install Google Play Services

1. Setelah Google Play Services terinstall, buat project baru menggunakan map activity dengan cara membuka Android Studio, kemudian Start a New Android Studio Project atau:

* Pilih File -> New -> New Project.
* Tuliskan nama aplikasi, company domain, dan lokasi project, kemudian klik Next.
* Untuk target android device bias dipilih Phone and Tablet seperti pada modul sebelumnya.
* Pada bagian “Add an Activity to Mobile”, pilihlah Google Maps Activity, kemudia klik Next.
* Tuliskan nama activity, nama layout, judul. Kemudian klik Finish.

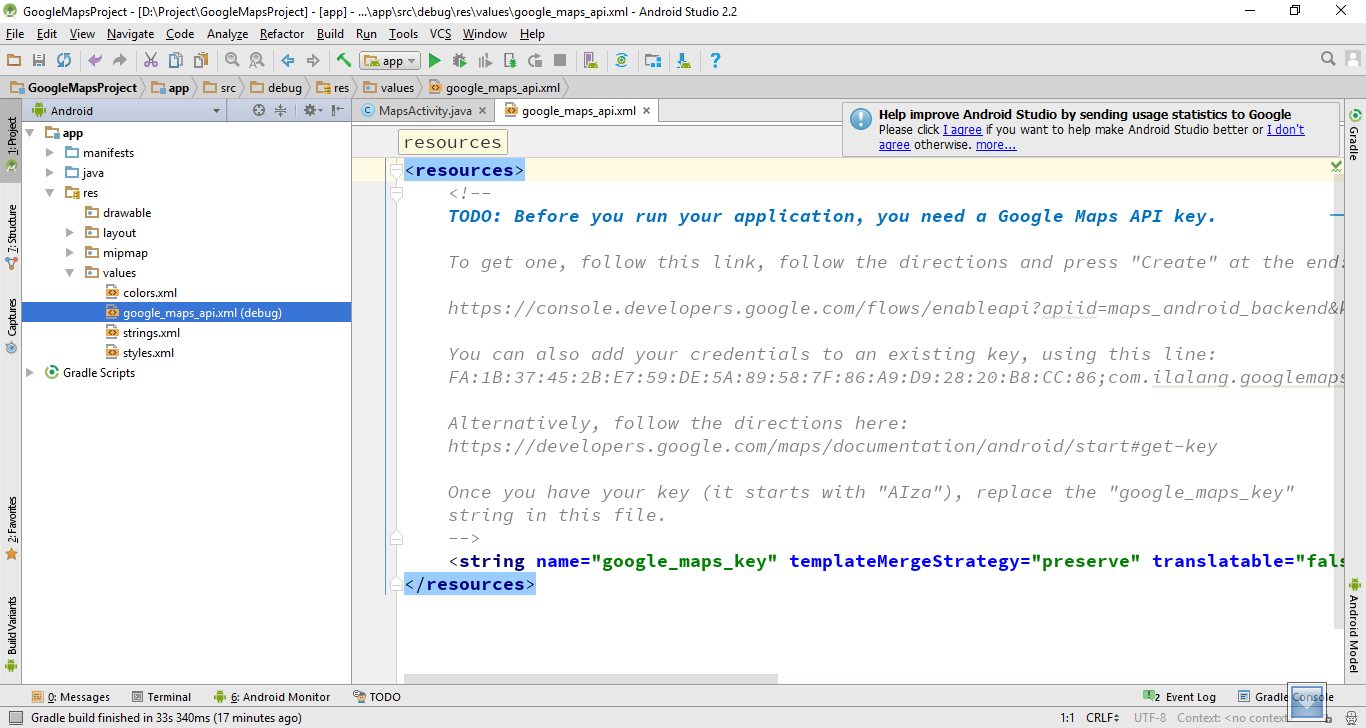
Ketika project telah berhasil di-build, Android Studio akan membuka file google\_maps\_api.xml dan MapsActivity.java pada editor. Pada file google\_maps\_api.xml terdapat petunjuk untuk mendapatkan Google Maps API key sebelum kita mencoba untuk menjalankan aplikasi.

**Gambar 2**. Membuat Project Baru dengan Google Maps Activity

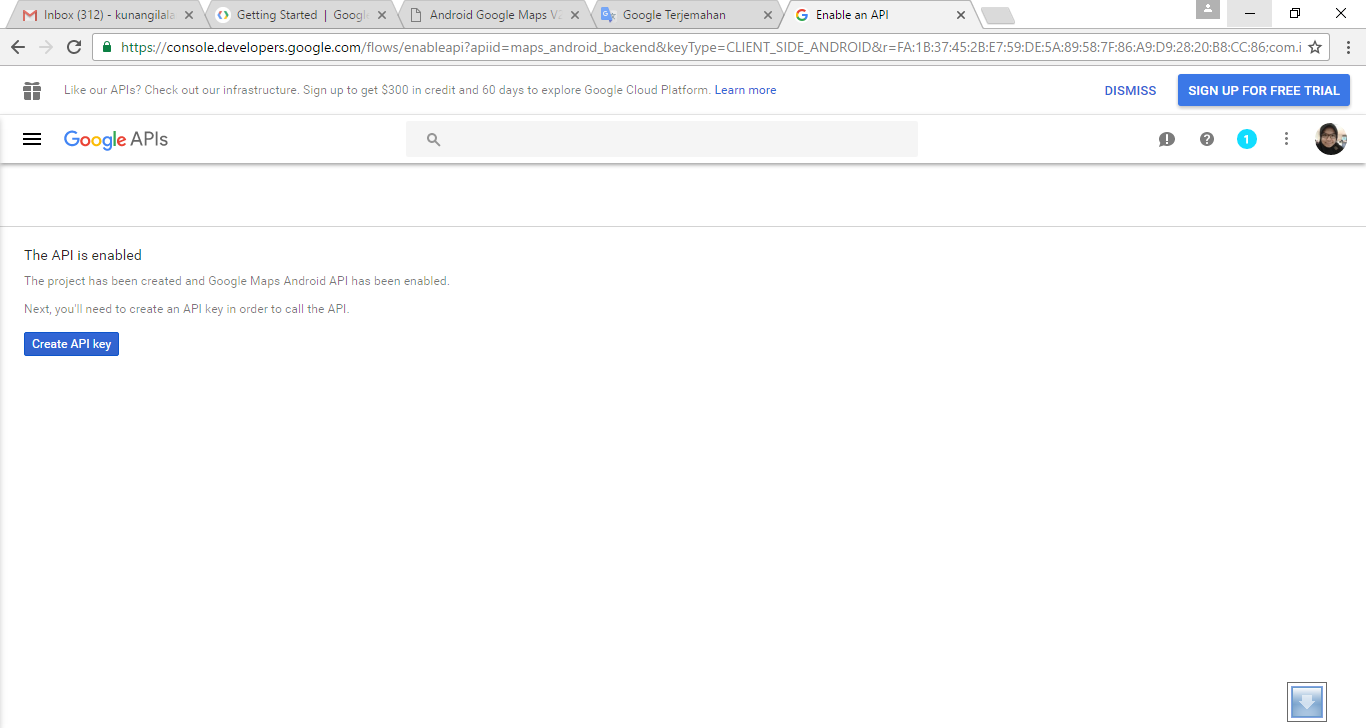
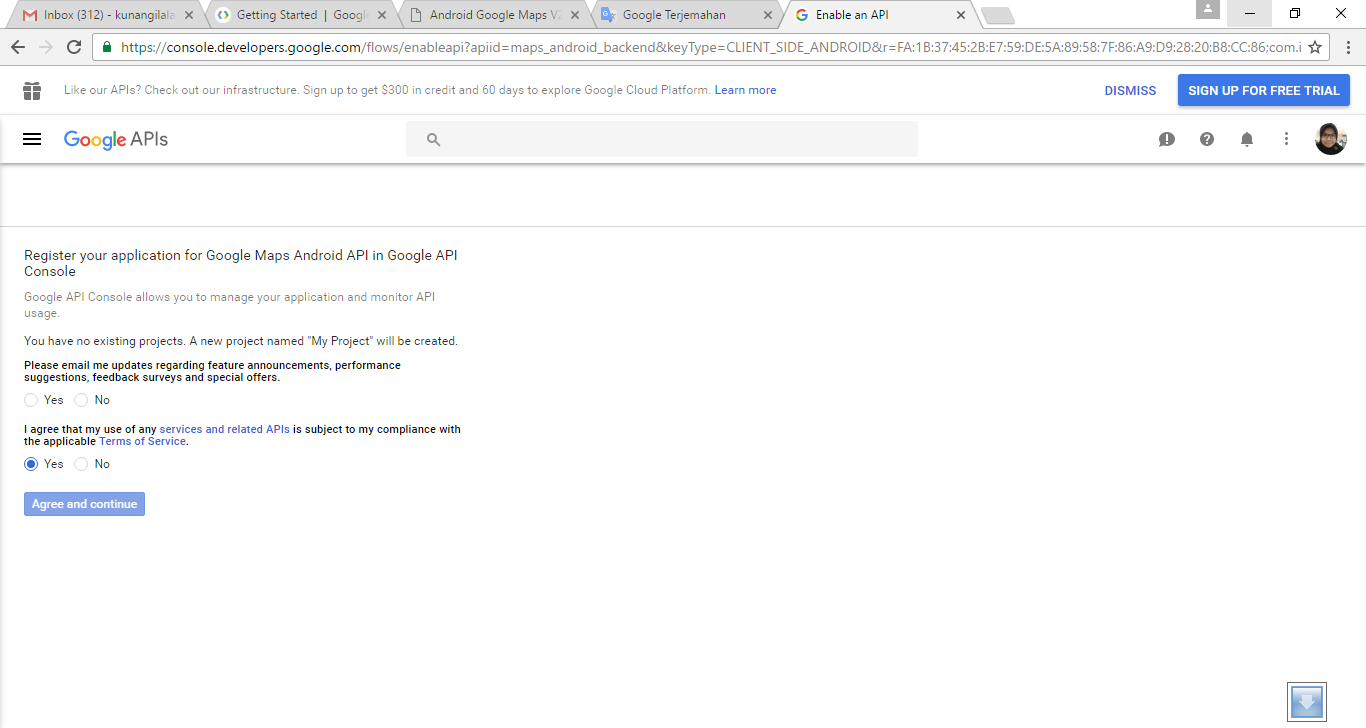
1. Tahapan selanjutnya adalah proses mendapatkan API key. Aplikasi kita membutuhkan API key untuk mengakses server Google Maps. Jenis key yang dibutuhkan adalah API key dengan batasan untuk aplikasi Android, dimana key ini bersifat gratis. Kita dapat menggunakannya untuk berbagai aplikasi yang membutuhkan Google Maps Android API, dan key tersebut juga dapat digunakan oleh pengguna dalam jumlah yang tidak terbatas. Terdapat dua cara untuk mendapatkan API key dari Android Studio, yaitu:

* Cara cepat dan mudah: gunakan link yang terdapat pada file google\_maps\_api.xml, yang telah dibuat oleh Android Studio:



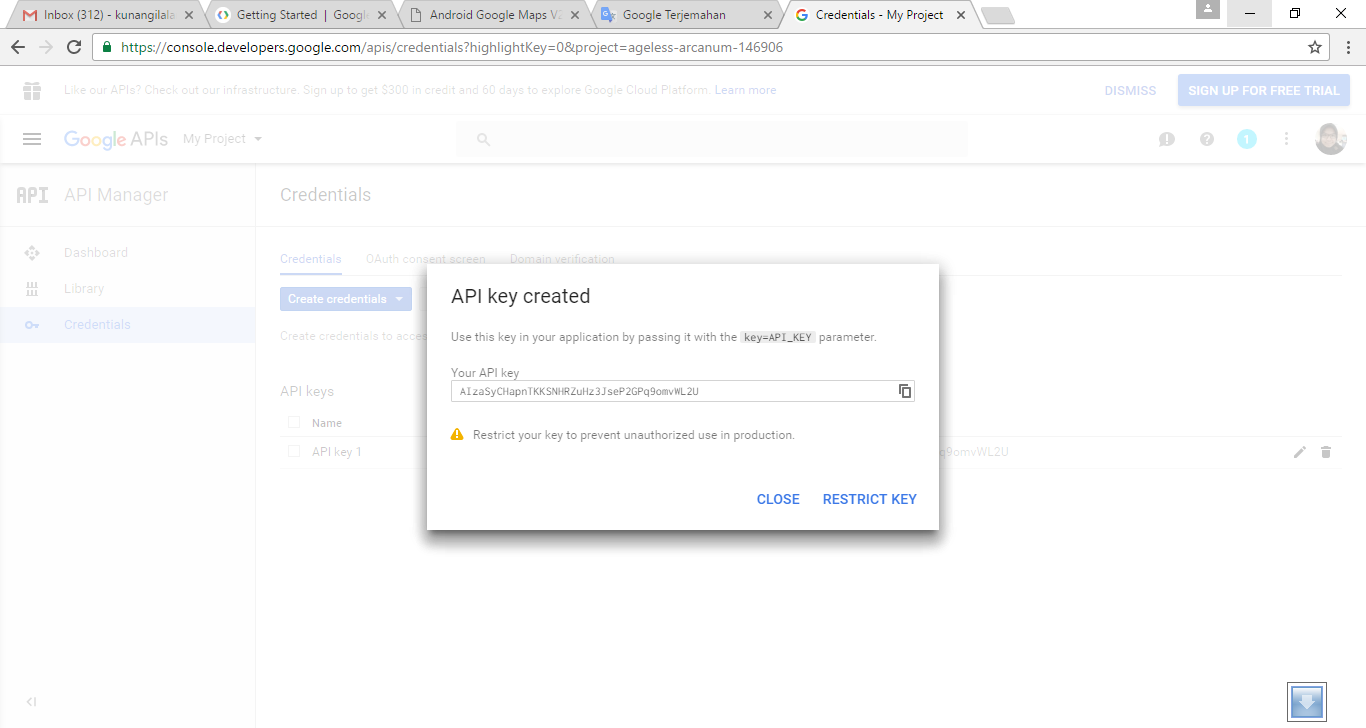
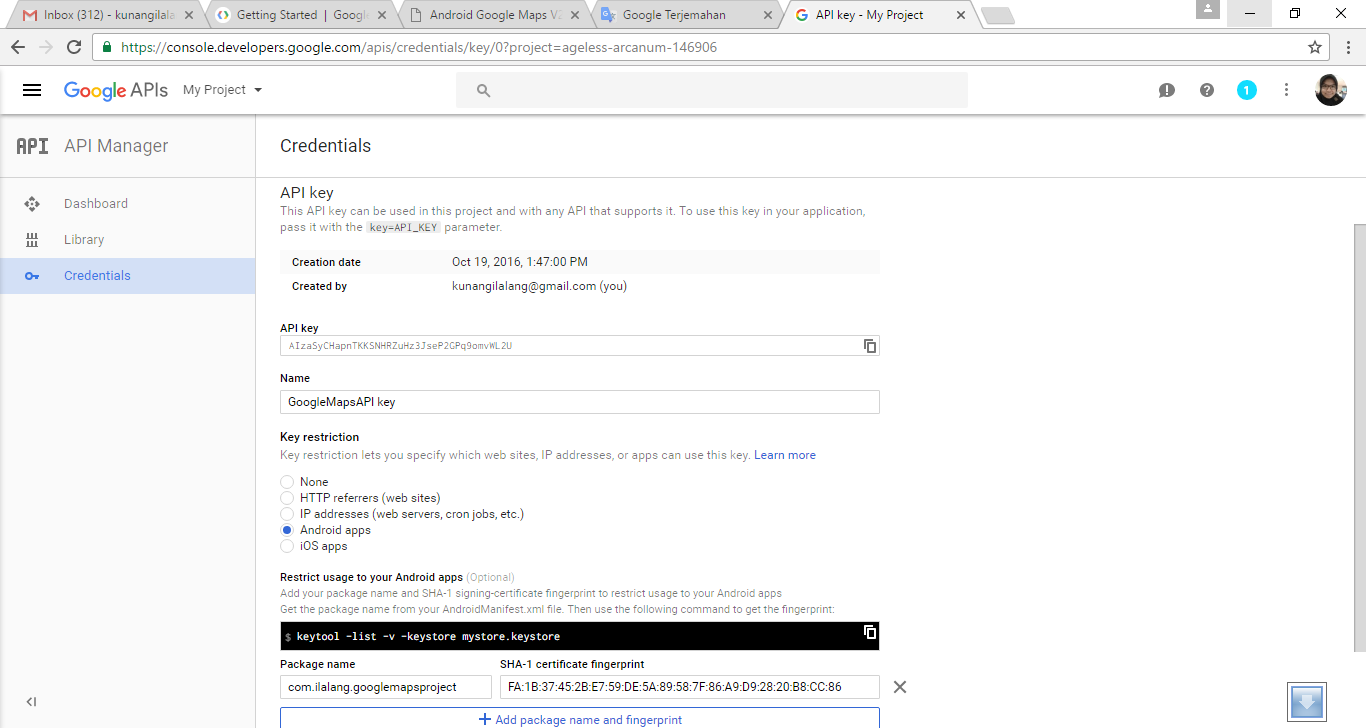
**Gambar 3**. Link yang terdapat pada file google\_maps\_api.xml

1. Copy link yang terdapat pada file google\_maps\_api.xml dan paste pada browser. Link tersebut akan mengarahkan kita pada Google API Console dan memberikan informasi yang diperlukan ke Google API Console melalui parameter URL, sehingga dapat mengurangi input manual.
2. Ikuti petunjuknya untuk membuat project baru pada Google API Console atau pilih project yang sudah ada



**Gambar 4**. Proses pembuatan API Key pada Google API Console

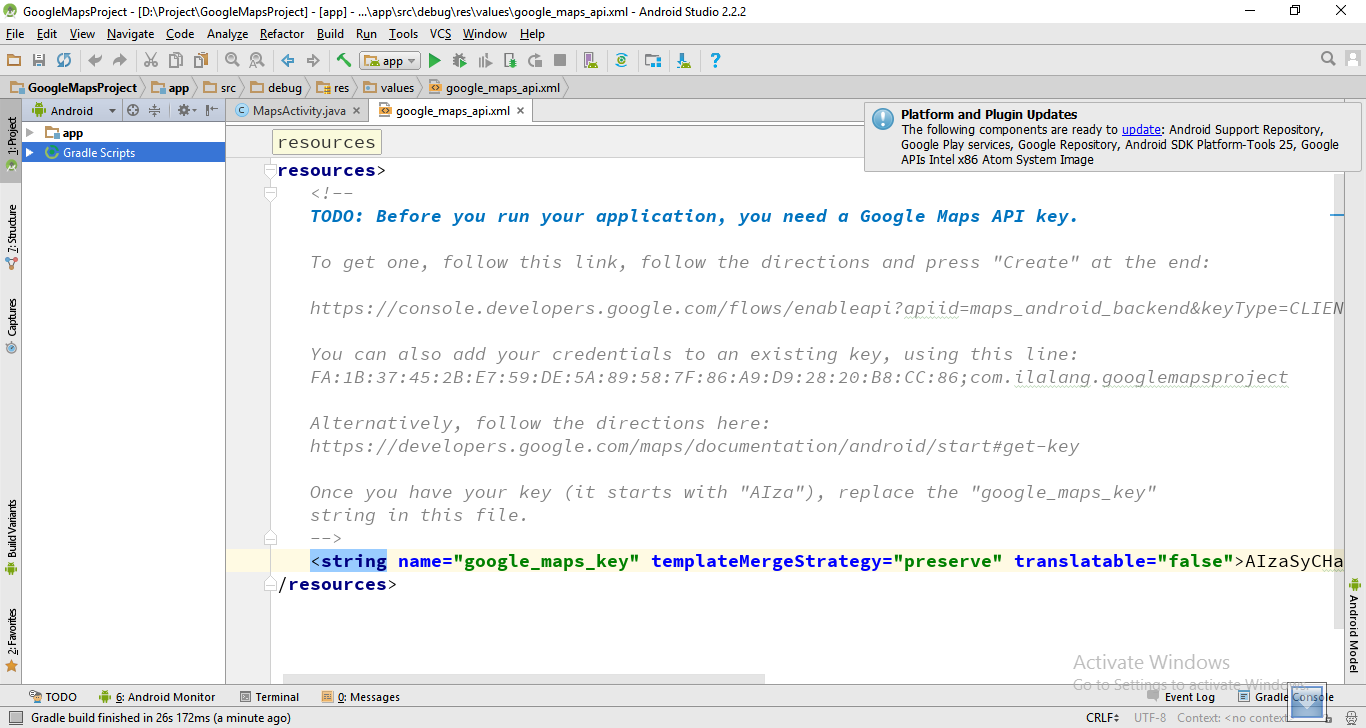
1. Buat Android API Key yang terbatas untuk project kita

**Gambar 5**. Pembatasan terhadap API Key yang baru saja dibuat

1. Copy hasil API Key, dan kembali ke Android Studio. Kemudian paste API Key tersebut ke elemen <string> yang terdapat pada file google\_maps\_api.xml (ganti bagian “YOUR\_KEY\_HERE” dengan API Key kita).

* Cara yang sedikit kurang cepat: Gunakan credentials yang telah disediakan oleh Android Studio pada file google\_maps\_api.xml:
  + Copy credentials yang terdapat pada file google\_maps\_api.xml



**Gambar 6**. Credentials pada file google\_maps\_api.xml

* + Buka Google API Console pada browser
  + Gunakan credentials yang telah di-copy untuk menambahkan aplikasi pada API Key yang sudah ada atau untuk membuat API Key yang baru.

1. Setelah mendapatkan API key, kita perlu menambahkan beberapa pengaturan pada manifest di aplikasi, antara lain:

* Menentukan versi dari Google Play services, dengan cara menambahkan deklarasi pada elemen <application> pada file AndroidManifest.xml untuk mendeklarasikan versi Google Play services yang akan digunakan untuk meng-*compile* aplikasi. Deklarasinya adalah sebagai berikut:

|  |
| --- |
| <meta-data     android:name="com.google.android.gms.version"     android:value="@integer/google\_play\_services\_version" /> |

* Pastikan bahwa API Key telah dicantumkan pada file AndroidManifest.xml
* Menentukan *Android permission* yang diperlukan oleh aplikasi, dengan cara menambahkan elemen <uses-permission> pada AndroidManifest.xml. Google maps membutuhkan beberapa permissions, antara lain:
  + **ACCESS\_NETWORK\_STATE**, untuk pengecekan terhadap kondisi jaringan, apakah data dapat diunduh atau tidak
  + **INTERNET**, untuk pengecekan terhadap status koneksi internet
  + **WRITE\_EXTERNAL\_STORAGE**, untuk menulis pada penyimpanan eksternal (*external storage*) karena google maps menyimpan data peta pada external storage
  + **ACCESS\_COARSE\_LOCATION**, untuk menentukan lokasi pengguna dengan mengunakan WiFi dan data seluler ponsel
  + **ACCESS\_FINE\_LOCATION**, untuk menentukan lokasi pengguna dengan menggunakan GPS
  + **OpenGL ES V2**, dibutuhkan Google Maps V2 untuk melakukan *render* peta

File AndroidManifest.xml setelah dilakukan konfigurasi adalah seperti berikut:

|  |
| --- |
| **AndroidManifest.xml** |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="com.modul.googlemapsproject"**>    <**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"** />  <**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"** />   *<!-- Required to show current location -->* <**uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION"** />  <**uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"** />   *<!-- Required OpenGL ES 2.0. for Maps V2 -->* <**uses-feature  android:glEsVersion="0x00020000"  android:required="true"** />   <**application  android:allowBackup="true"  android:icon="@mipmap/ic\_launcher"  android:label="@string/app\_name"  android:supportsRtl="true"  android:theme="@style/AppTheme"**>  <**meta-data  android:name="com.google.android.geo.API\_KEY"  android:value="@string/google\_maps\_key"** />   <**meta-data  android:name="com.google.android.gms.version"  android:value="@integer/google\_play\_services\_version"** />   <**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**> |

1. Langkah selanjutnya adalah menambahkan obyek Fragment pada Activity yang akan menangani peta dengan cara menambahkan elemen <fragment> pada file *layout*. Google maps yang baru diimplementasikan menggunakan MapFragment, *sub class* dari *class* Fragment. Buka *layout activity* pada file activity\_maps.xml dan tambahkan elemen <fragment> seperti berikut:

|  |
| --- |
| **Activity\_maps.xml** |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent"** >   <**fragment 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:id="@+id/map"  android:name="com.google.android.gms.maps.SupportMapFragment"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context="com.modul.googlemapsproject.MapsActivity"** /> </**RelativeLayout**> |

1. Ubah nilai latitude dan longitude sesuai dengan lokasi yang diinginkan (pada modul ini menggunakan lokasi kota Malang), pada file MapsActivity.java menggunakan perintah:

|  |
| --- |
| LatLng malang = **new** LatLng(-7.9797, 112.6304);  **googleMap**.addMarker(**new** MarkerOptions().position(malang).title(**"Marker in Malang"**)); **googleMap**.moveCamera(CameraUpdateFactory.*newLatLngZoom*(malang,14.0f)); |

Source code file MapsActivity.java secara keseluruhan adalah sebagai berikut:

|  |
| --- |
| **MapsActivity.java** |
| **package** com.modul.googlemapsproject;  **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***);  *// Mendapatkan SupportMapFragment mendapatkan notifikasi saat peta siap digunakan* SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()  .findFragmentById(R.id.***map***);  mapFragment.getMapAsync(**this**);  }    @Override  **public void** onMapReady(GoogleMap googleMap) {  **mMap** = googleMap;  *// Menambahkan marker untuk kota Malang, memindahkan camera, dan melakukan zoom in* LatLng malang = **new** LatLng(-7.9797, 112.6304);  **mMap**.addMarker(**new** MarkerOptions().position(malang).title(**"Marker in Malang"**));  **mMap**.moveCamera(CameraUpdateFactory.*newLatLngZoom*(malang,14.0f));  } } |

Hasil dari Map yang telah dibuat adalah sebagai berikut:



**Gambar 7**. Hasil Maps yang menunjukkan lokasi Malang

1. Pengaturan lokasi juga dapat diatur agar bisa menyesuaikan dengan posisi pengguna saat membuka aplikasi. Untuk menyesuaikan dengan current location, ada beberapa hal yang perlu ditambahkan pada file MapsActivity.java, antara lain:
2. Menambahkan interface yang akan diimplementasi pada class MapsActivity, yaitu

|  |
| --- |
| **public class** MapsActivity **extends** FragmentActivity **implements** OnMapReadyCallback,  GoogleApiClient.ConnectionCallbacks, GoogleApiClient.OnConnectionFailedListener,  LocationListener |

1. Melakukan pengecekan versi Android dan location permission, dimana Android permission-nya telah dideklarasikan pada file AndroidManifest.xml. Pengecekan ini dilakukan pada metode onCreate, seperti berikut:

|  |
| --- |
| **if** (android.os.Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***M***) {  checkLocationPermission(); } |

Dimana metode checkLocationPermission digunakan untuk melakukan request terhadap location permission, dengan code sebagai berikut:

|  |
| --- |
| **public boolean** checkLocationPermission(){  **if** (ContextCompat.*checkSelfPermission*(**this**,  Manifest.permission.***ACCESS\_FINE\_LOCATION***)  != PackageManager.***PERMISSION\_GRANTED***) {  **if** (ActivityCompat.*shouldShowRequestPermissionRationale*(**this**,  Manifest.permission.***ACCESS\_FINE\_LOCATION***)) {  ActivityCompat.*requestPermissions*(**this**,  **new** String[]{Manifest.permission.***ACCESS\_FINE\_LOCATION***},  ***MY\_PERMISSIONS\_REQUEST\_LOCATION***);   } **else** { ActivityCompat.*requestPermissions*(**this**,  **new** String[]{Manifest.permission.***ACCESS\_FINE\_LOCATION***},  ***MY\_PERMISSIONS\_REQUEST\_LOCATION***);  }  **return false**;  } **else** {  **return true**;  } } |

1. Menambahkan proses inisialisasi Google Play Services pada metode onMapReady, seperti berikut:

|  |
| --- |
| **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**); } |

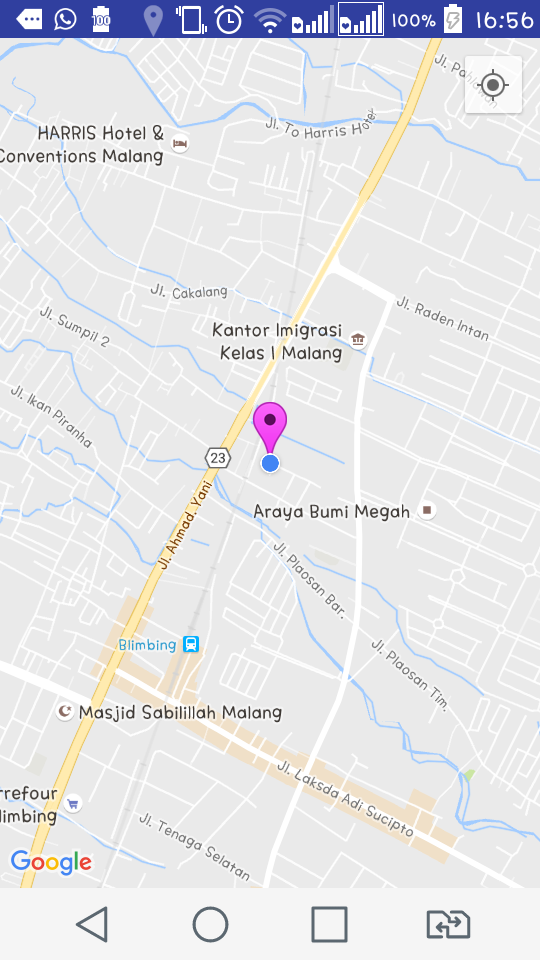
1. Memberikan marker pada lokasi pengguna saat menjalankan aplikasi (current location), mengarahkan camera pada lokasi tersebut, dan melakukan perbesaran peta agar lebih focus pada lokasi. Proses ini dilakukan pada metode onLocationChanged.

|  |
| --- |
| LatLng latLng = **new** LatLng(location.getLatitude(), location.getLongitude()); MarkerOptions markerOptions = **new** MarkerOptions(); markerOptions.position(latLng); markerOptions.title(**"Current Position"**); markerOptions.icon(BitmapDescriptorFactory.*defaultMarker*(BitmapDescriptorFactory.***HUE\_MAGENTA***)); **mCurrLocationMarker** = **mMap**.addMarker(markerOptions); **mMap**.moveCamera(CameraUpdateFactory.*newLatLng*(latLng)); **mMap**.animateCamera(CameraUpdateFactory.*zoomTo*(15)); |

1. File MapsActivity.java secara keseluruhan dapat dilihat pada tabel berikut ini:

|  |
| --- |
| **MapsActivity.java** |
| **package** com.modul.googlemapsproject;  **import** android.content.pm.PackageManager; **import** android.location.Location; **import** android.os.Build; **import** android.support.annotation.NonNull; **import** android.support.annotation.Nullable; **import** android.support.v4.app.ActivityCompat; **import** android.support.v4.app.FragmentActivity; **import** android.os.Bundle; **import** android.support.v4.content.ContextCompat; **import** android.widget.Toast; **import** android.Manifest;  **import** com.google.android.gms.common.ConnectionResult; **import** com.google.android.gms.common.api.GoogleApiClient; **import** com.google.android.gms.location.LocationListener; **import** com.google.android.gms.location.LocationRequest; **import** com.google.android.gms.location.LocationServices; **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.BitmapDescriptorFactory; **import** com.google.android.gms.maps.model.LatLng; **import** com.google.android.gms.maps.model.Marker; **import** com.google.android.gms.maps.model.MarkerOptions;  **public class** MapsActivity **extends** FragmentActivity **implements** OnMapReadyCallback,  GoogleApiClient.ConnectionCallbacks,  GoogleApiClient.OnConnectionFailedListener,  LocationListener {   **private** GoogleMap **mMap**;  SupportMapFragment **mapFrag**;  LocationRequest **mLocationRequest**;  GoogleApiClient **mGoogleApiClient**;  Location **mLastLocation**;  Marker **mCurrLocationMarker**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_maps***);   **if** (android.os.Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***M***) {  checkLocationPermission();  }   *// Mendapatkan SupportMapFragment mendapatkan notifikasi saat peta siap digunakan* SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()  .findFragmentById(R.id.***map***);  mapFragment.getMapAsync(**this**);  } @Override  **public void** onPause() {  **super**.onPause();   *//menghentikan update lokasi ketika Activity tidak lagi aktif* **if** (**mGoogleApiClient** != **null**) {  LocationServices.***FusedLocationApi***.removeLocationUpdates(**mGoogleApiClient**, **this**);  }  }   @Override  **public void** onMapReady(GoogleMap googleMap) {  **mMap** = googleMap;   *//Inisialisasi 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**);  }  }   **protected synchronized void** buildGoogleApiClient() {  **mGoogleApiClient** = **new** GoogleApiClient.Builder(**this**)  .addConnectionCallbacks(**this**)  .addOnConnectionFailedListener(**this**)  .addApi(LocationServices.***API***)  .build();  **mGoogleApiClient**.connect();  }   @Override  **public void** onConnected(@Nullable Bundle bundle) {  **mLocationRequest** = **new** LocationRequest();  **mLocationRequest**.setInterval(1000);  **mLocationRequest**.setFastestInterval(1000);  **mLocationRequest**.setPriority(LocationRequest.***PRIORITY\_BALANCED\_POWER\_ACCURACY***);  **if** (ContextCompat.*checkSelfPermission*(**this**,  Manifest.permission.***ACCESS\_FINE\_LOCATION***)  == PackageManager.***PERMISSION\_GRANTED***) {  LocationServices.***FusedLocationApi***.requestLocationUpdates(**mGoogleApiClient**, **mLocationRequest**, **this**);  }  }   @Override  **public void** onConnectionSuspended(**int** i) {   }   @Override  **public void** onConnectionFailed(@NonNull ConnectionResult connectionResult) {   }   @Override  **public void** onLocationChanged(Location location) {  **mLastLocation** = location;  **if** (**mCurrLocationMarker** != **null**) {  **mCurrLocationMarker**.remove();  }   *// Memberikan marker pada current location* LatLng latLng = **new** LatLng(location.getLatitude(), location.getLongitude());  MarkerOptions markerOptions = **new** MarkerOptions();  markerOptions.position(latLng);  markerOptions.title(**"Current Position"**);  markerOptions.icon(BitmapDescriptorFactory.*defaultMarker*(BitmapDescriptorFactory.***HUE\_MAGENTA***));  **mCurrLocationMarker** = **mMap**.addMarker(markerOptions);  *//memindahkan camera map* **mMap**.moveCamera(CameraUpdateFactory.*newLatLng*(latLng));  **mMap**.animateCamera(CameraUpdateFactory.*zoomTo*(15));   *//stop update lokasi* **if** (**mGoogleApiClient** != **null**) {  LocationServices.***FusedLocationApi***.removeLocationUpdates(**mGoogleApiClient**, **this**);  }   }   **public static final int *MY\_PERMISSIONS\_REQUEST\_LOCATION*** = 99;  **public boolean** checkLocationPermission(){  **if** (ContextCompat.*checkSelfPermission*(**this**,  Manifest.permission.***ACCESS\_FINE\_LOCATION***)  != PackageManager.***PERMISSION\_GRANTED***) {  **if** (ActivityCompat.*shouldShowRequestPermissionRationale*(**this**,  Manifest.permission.***ACCESS\_FINE\_LOCATION***)) {  ActivityCompat.*requestPermissions*(**this**,  **new** String[]{Manifest.permission.***ACCESS\_FINE\_LOCATION***},  ***MY\_PERMISSIONS\_REQUEST\_LOCATION***);  } **else** { ActivityCompat.*requestPermissions*(**this**,  **new** String[]{Manifest.permission.***ACCESS\_FINE\_LOCATION***},  ***MY\_PERMISSIONS\_REQUEST\_LOCATION***);  }  **return false**;  } **else** {  **return true**;  }  }   @Override  **public void** onRequestPermissionsResult(**int** requestCode, String permissions[], **int**[] grantResults) {  **switch** (requestCode) {  **case *MY\_PERMISSIONS\_REQUEST\_LOCATION***: {  *//Jika request dibatalkan, hasil array akan kosong* **if** (grantResults.**length** > 0  && grantResults[0] == PackageManager.***PERMISSION\_GRANTED***) {   *// permission telah diberikan* **if** (ContextCompat.*checkSelfPermission*(**this**,  Manifest.permission.***ACCESS\_FINE\_LOCATION***)  == PackageManager.***PERMISSION\_GRANTED***) {   **if** (**mGoogleApiClient** == **null**) {  buildGoogleApiClient();  }  **mMap**.setMyLocationEnabled(**true**);  }   } **else** {   *// permission tidak diberikan* Toast.*makeText*(**this**, **"permission denied"**, Toast.***LENGTH\_LONG***).show();  }  **return**;  }}  } } |

1. Hasil peta yang menunjukkan lokasi pengguna pada saat membuka aplikasi adalah sebagai berikut:



**Gambar 8**. Hasil Maps yang menunjukkan lokasi pengguna