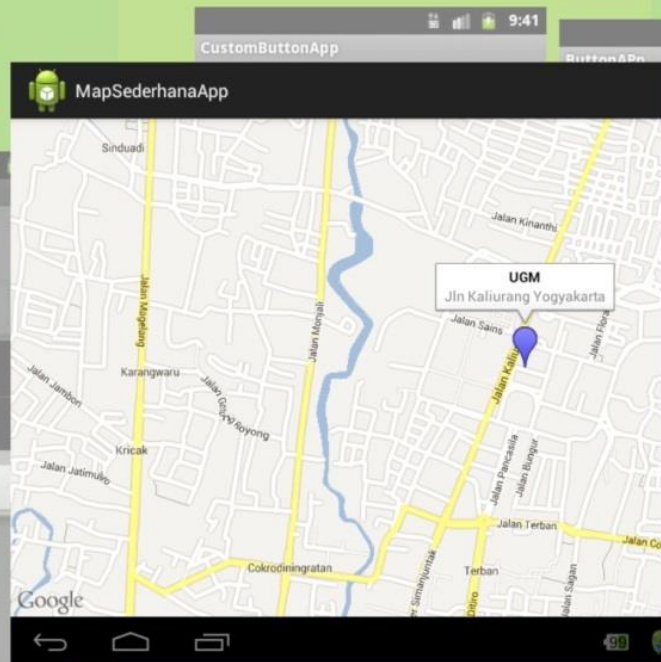
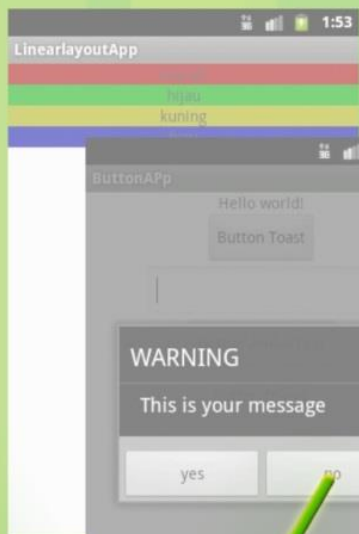


live coding

#2

android

Arif Akbarul Huda, S.Si



**cara cepat menguasai
pemrograman android**

Seri Google
Map API v2

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

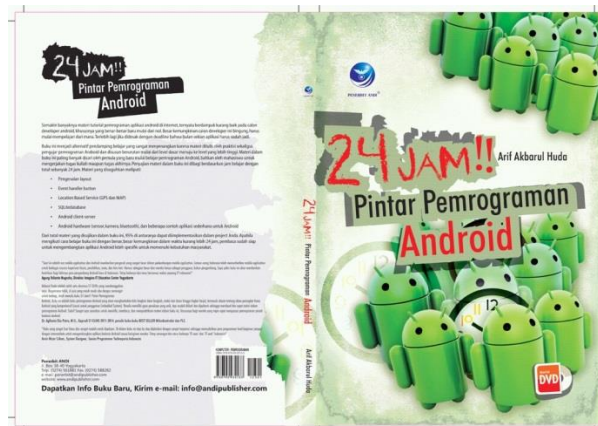
sedang menyelesaikan studi di Pascasarjana Magister Teknik Informatik UGM.

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.odayib.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 membuat 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 sekaligus 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 prorgres 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

1. Menampilkan Map.....	8
A. Instal Google Play service pada ADT Eclipse	8
B. Menambahkan referensi library google-play-service.jar ke dalam project.....	9
C. Generate Debug API KEY SHA1 pada komputer	11
D. Mendaftarkan pada Googel APIs Console.....	12
E. Membuat project dengan target Google API	15
F. Deployment.....	19
2. Menampilkan lokasi dengan marker.....	20
3. Menghitung jarak antara dua lokasi.....	21
4. Menampilkan jalur (direction)	24
5. Menggunakan GPS.....	24





1. Menampilkan Map

Sejak tanggal 3 Desember 2012, Google secara resmi mengumumkan bahwa layanan API Map berubah menjadi versi 2 sedangkan versi sebelumnya masih diberi waktu berjalan hingga satu tahun kedepan. Pada versi dua ini, ada banyak perubahan mulai dari proses pendaftaran API KEY hingga cara menampilkan MAP pada android. Ebook ini akan membahas langkah demi langkah cara baru menampilkan MAP pada android. Berikut ini urutan secara umum yang harus kita lakukan.

- A. Instal Google Play service pada ADT Eclipse
- B. Menambahkan referensi library google-play-service.jar ke dalam project
- C. Membuat project dengan target Google API
- D. Generate Debug API KEY SHA1 pada komputer
- E. Mendaftarkan pada Googel APIs Console (<https://code.google.com/apis/console/>)
- F. Modifikasi AndroidManifest.xml
- G. Deployment

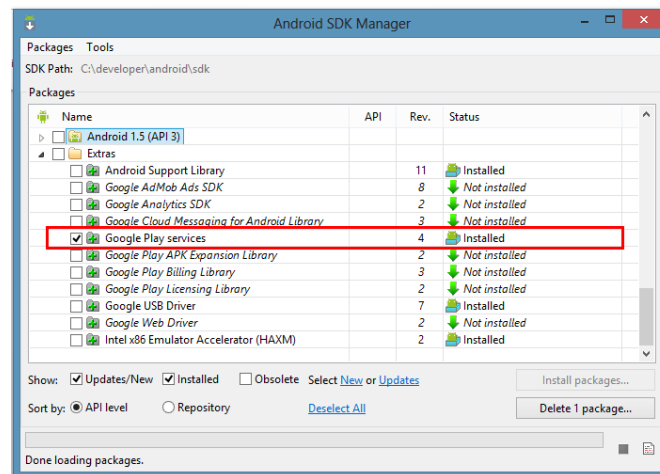
Sampai tulisan ini dibuat, proses dubug Google Map belum bisa dilakukan pada emulator. Kita harus menyiapkan device Android untuk proses debugging. Materi dalam ebook ini berhasil saya debug pada Smartfren Andro Tab melalui ADB. Baiklah, mari kita bahas masing-masing langkah diatas.

A. Instal Google Play service pada ADT Eclipse

Sebelumnya kita harus memastikan bahwa ADT eclipse sudah terinstal Google Play service. Berikut langkah demi langkahnya

1. Jalankan ADT, klik Window > Android SDK Manager
2. Akan muncul kotak dialog seperti gambar 1.1. Centang bagian Extras > Google Play Service. Jika statusnya Not Installed, klik Install Package



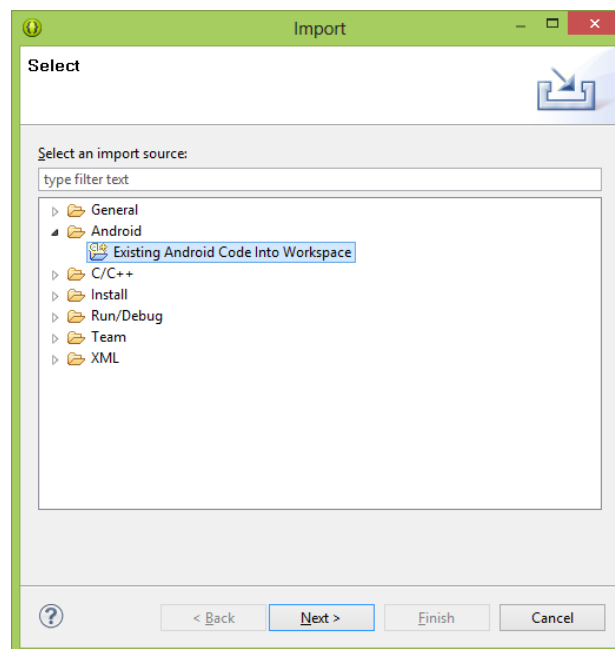


Gambar 1.1. Menginstal Google Play services pada ADT

B. Menambahkan referensi library google-play-service.jar ke dalam project

Setelah melalauai tahap pertama, kita perlu menambahkan library google-play-service ke dalam workspace. Caranya sebagai berikut.

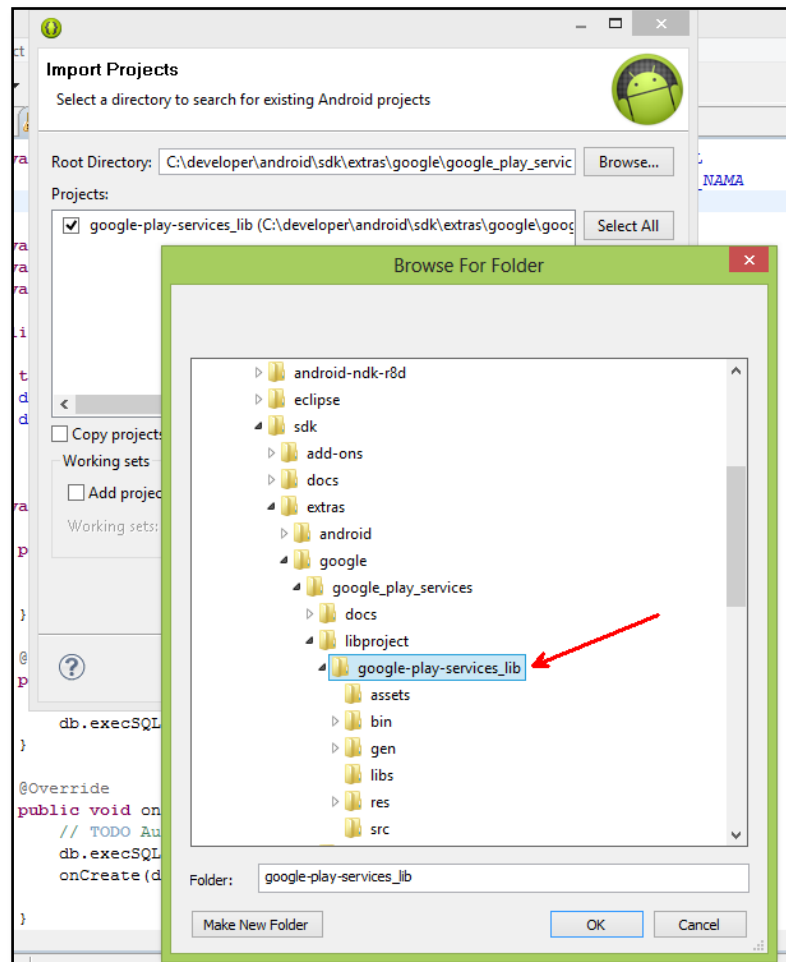
1. Klik File > import > Existing Android Code Into Workspace, perhatikan Gambar 1.2.



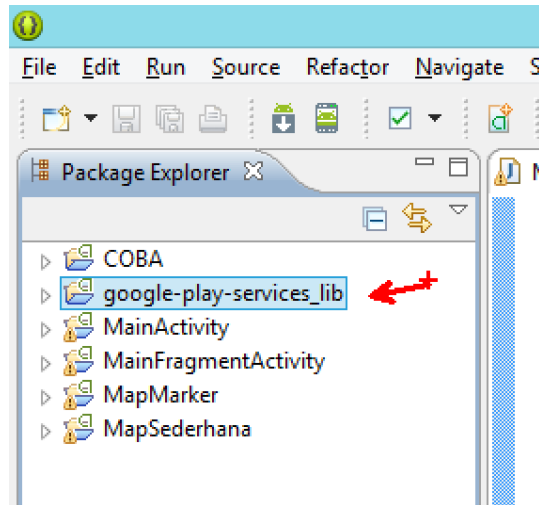
Gambar 1.2. Import library android

2. Pada field Root Directory klik Browse. Cari <android_sdk_folder>/extras/google/

google_play_services/libproject/google-play-service_lib dan klik finish atau perhatikan Gambar 1.3. Jika langkah ini benar, maka pada Package Explorer seperti pada Gambar 1.4 akan muncul library yang baru saja kita masukkan.



Gambar 1.3. Memasukkan library ke dalam workspace

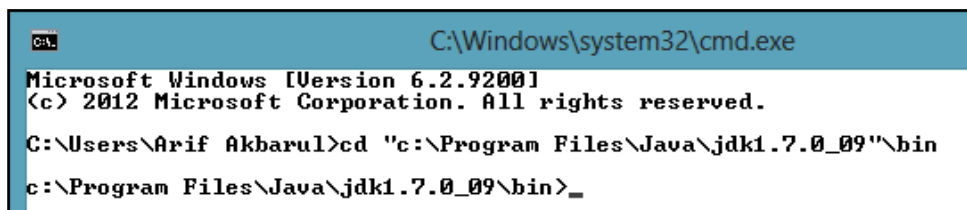


Gambar 1.4. Tampilan Package Explorer setelah ditambah library

C. Generate Debug API KEY SHA1 pada komputer

Setiap komputer/laptop memiliki Debug API KEY berbeda-beda. Hasil akhir tahap ini berupa key dalam format SHA1, yang akan digunakan untuk mendaftarkan pada Google apis console.

1. Jalankan command prompt
2. Masuklah ke direktori **bin** pada folder instalasi Java. Biasanya terdapat di dalam folder **c:\Program Files\Java\<jdk_version>\bin** dengan syntax seperti Gambar 1.5.



Gambar 1.5. Masuk ke direktori bin pada folder instalasi java

3. Kita generate Debug Api Key dengan syntax dibawah ini atau lihat Gambar 1.6.

```
Keytool.exe -list -v -keystore
"c:\Users\<nama_user_komputer_anda>\.android\debug.keystore"
-alias androiddebugkey -storepass android -keypass android
```



```
c:\Program Files\Java\jdk1.7.0_09\bin>keytool.exe -list -v -keystore "c:\Users\Arif Akbarul\.android\debug.keystore" -alias androiddebugkey -storepass android -keypass android
```

Gambar 1.6. proses generate debug api key

- Hasil langkah ke tiga dan ke empat berupa **Certificate fingerprint** berisi key dalam format **MD5, SHA1, SHA256, Signature dan version** dapat dilihat seperti Gambar 1.7. Kita akan menggunakan key **SHA1** untuk mendaftarkan api key ke Google apis console.

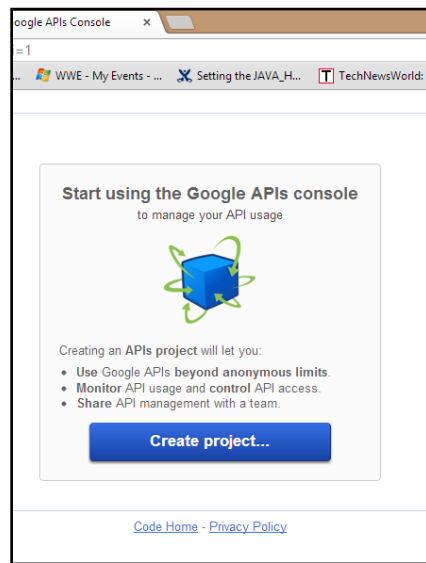
```
C:\Windows\system32\cmd.exe
(c) 2012 Microsoft Corporation. All rights reserved.
C:\Users\Arif Akbarul>cd "c:\Program Files\Java\jdk1.7.0_09\bin"
c:\Program Files\Java\jdk1.7.0_09\bin>keytool.exe -list -v -keystore "c:\Users\Arif Akbarul\.android\debug.keystore" -alias androiddebugkey -storepass android -keypass android
Alias name: androiddebugkey
Creation date: 06 Des 12
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 50c04266
Valid from: Thu Dec 06 13:59:50 ICT 2012 until: Sat Nov 29 13:59:50 ICT 2042
Certificate fingerprints:
    MD5:  CD:CF:14:43:42:5B:E4:49:D2:BB:FA:85:5D:6E:E1:75
    SHA1:  79:B1:ED:9C:D9:FE:1F:7A:35:24:A4:41:4E:BA:09:85:E1:EF:23:16
    SHA256: 33:01:19:9C:E9:79:56:E9:AD:4C:21:43:C3:33:71:26:2F:04:16:70:6C:
64:AB:AC:A8:79:43:8E:B1:BB:98:89
Signature algorithm name: SHA1withRSA
Version: 3
```

Gambar 1.7. Hasil proses generate debug api key

- Copy key SHA1 ke dalam notepad atau tahan dulu command prompt nya. Kita akan lanjutkan ke langkah berikutnya.

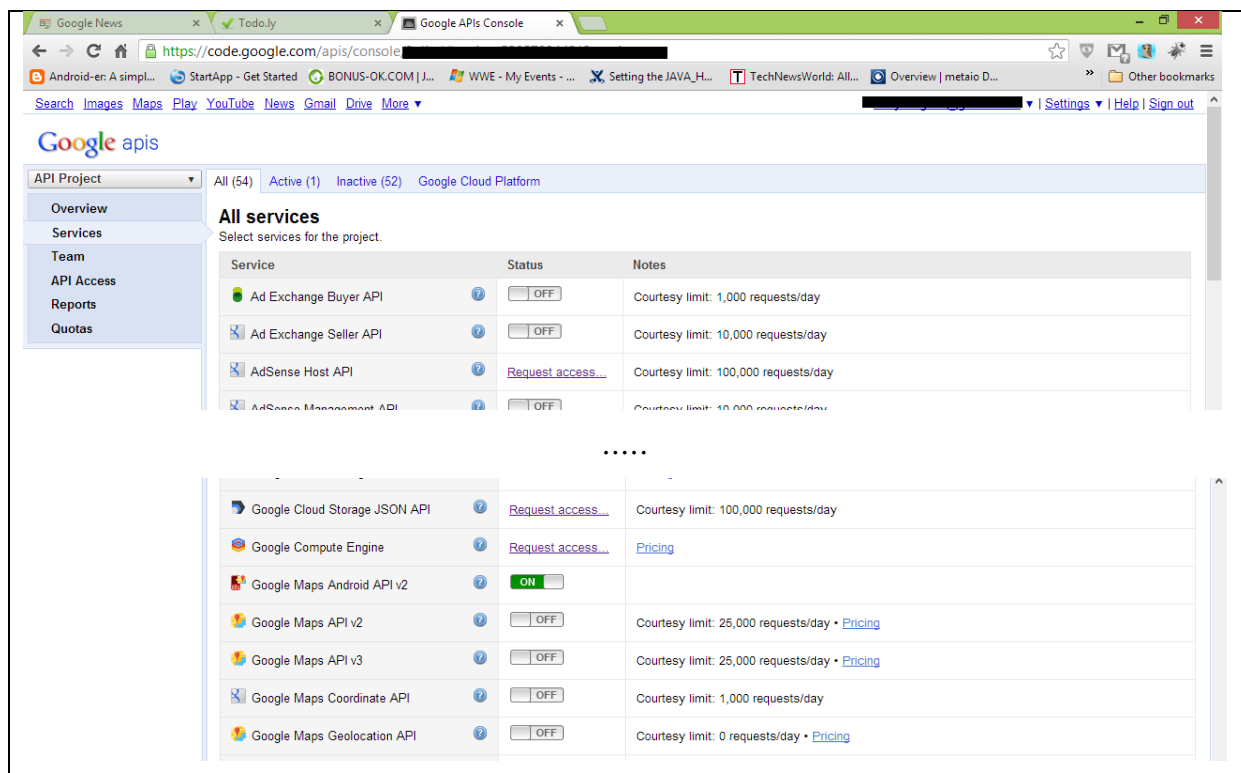
D. Mendaftarkan pada Google APIs Console

- Jalankan browser, masuk ke alamat <https://code.google.com/apis/console/>. Kita perlu login menggunakan akun Google. Tampilan pertama terlihat seperti Gambar 1.8, klik tombol Create Project untuk memulai



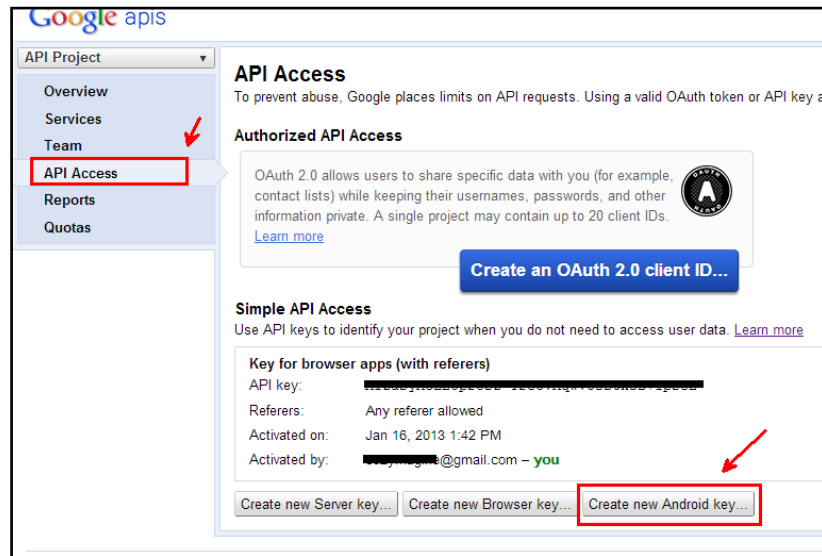
Gambar 1.8. Halaman awal Google Api Console

2. Langkah berikutnya kita akan diawarhkan ke halaman Service. Pada halaman ini kita harus mengaktifkan service **Google Maps Android API v2**. Scroll ke bawah untuk menemukan service ini, seperti terlihat pada Gambar 1.9.



Gambar 1.9. Mengaktifkan service Google Maps Android API v2

3. Langkah berikutnya, kita pindah ke halaman API Access. Pada halaman ini klik tombol Create new Android Key seperti pada Gambar 1.10

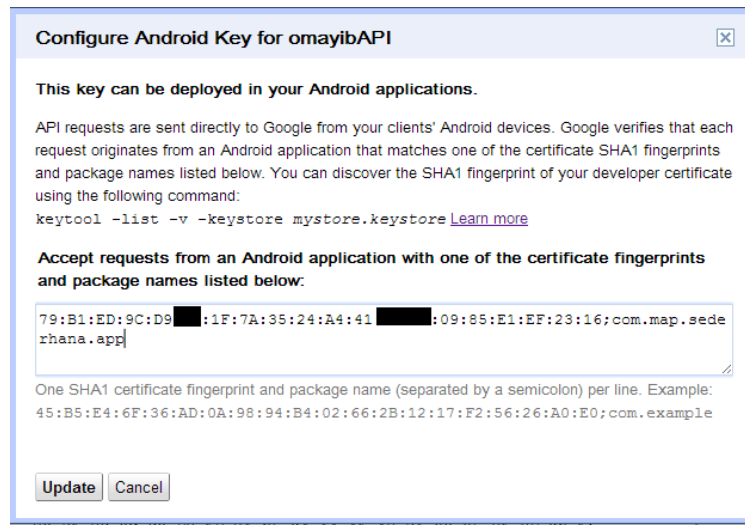


Gambar 1.10. Halaman API Access

4. Copy key **SHA1** yang kita peroleh pada langkah sebelumnya. Perhatikan Gambar 1.11, pada tahap ini akan muncul dialog untuk meletakkan/paste key **SHA1** diikuti **nama package** dipisahkan dengan titik koma (;) misal

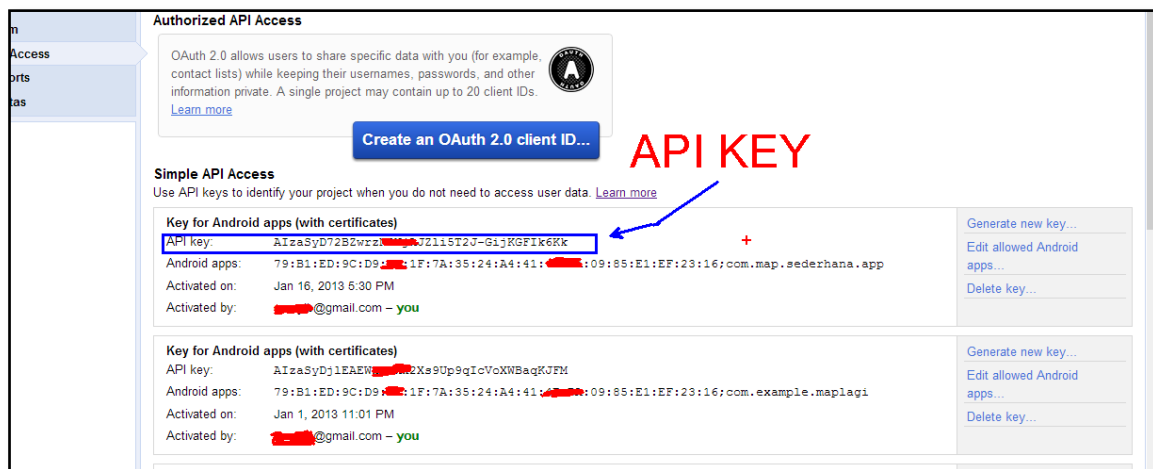
```
79:B1:ED:9C:D9:XX:1F:7A:35:XX:A4:41:XX:BA:09:85:E1:EF:23:16;com.map.sederhana.app
```

com.map.sederhana.app adalah package yang akan kita buat sebagai project aplikasi android.



Gambar 1.11. Memasukkan key SHA1 ke dalam API Project

5. Hasil akhir tahap ini tampak seperti Gambar 1.12. Kita akan memperoleh android API key (didalam kotak ditunjuk anak panah) yang akan digunakan pada saat membangun aplikasi berbasis Google Service.



Gambar 1.12. Hasil generate API Key

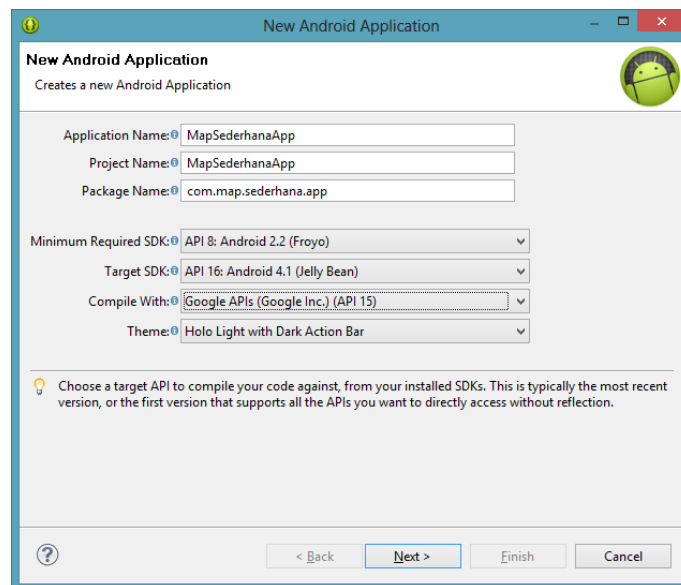
E. Membuat project dengan target Google API

Pada tahap ini, kita mulai membuat project paling sederhana yaitu menampilkan Google Map menggunakan SupportFragment dan FragmentActivity. Langkah-langkah detailnya sebagai berikut

1. Klik **File > new > Android Application Project**, kemudian isilah field-fieldnya seperti Tabel 1.1. atau Gambar 1.5.

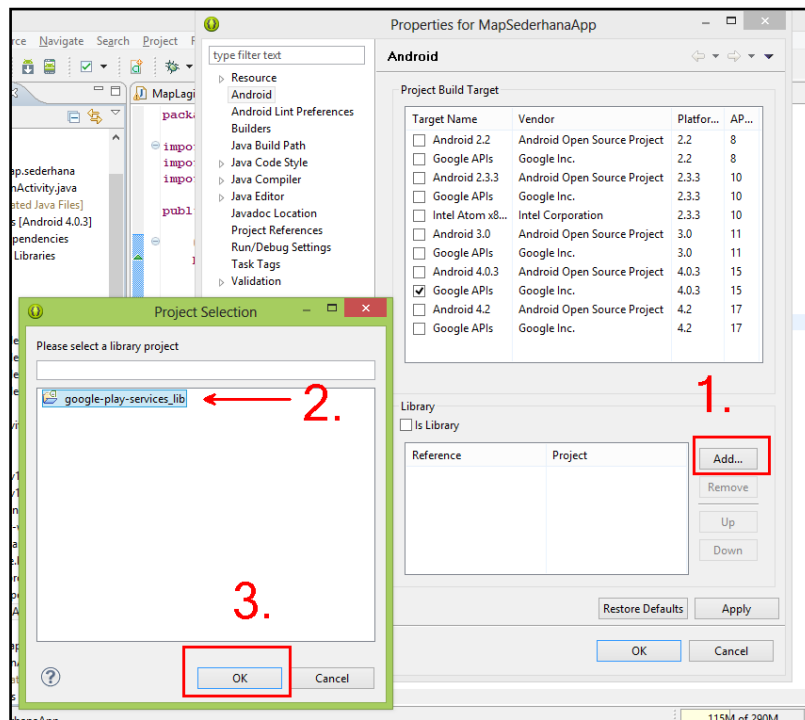
Tabel 1.1. Parameter pembuatan project baru

Application Name	MapSederhanaApp
Project Name	MapSederhanaApp
Package Name	com.map.sederhana.app
Target SDK	API 16
Compile With	Google APIs (API 15)



Gambar 1.13. Membuat project baru

2. Kita akan menambahkan library Google-play-service (langkah B) ke dalam project. Caranya **klik kanan project > properties > android**. Pada bagian Library (lihat Gambar 1.14) klik tombol **Add > google-play-services_lib > ok > ok**.



Gambar 1.14. Menambahkan library ke dalam project

3. Tambahkan **fragment** pada activity_main.xml seperti code dibawah ini

```

1: <RelativeLayout
2:     xmlns:android="http://schemas.android.com/apk/res/android"
3:     xmlns:tools="http://schemas.android.com/tools"
4:     android:layout_width="match_parent"
5:     android:layout_height="match_parent"
6:     tools:context=".MainActivity" >
7:
8:     <TextView
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="@string/hello_world" />
14:
15:     <fragment
16:         android:id="@+id/map"
17:         android:layout_width="match_parent"
18:         android:layout_height="match_parent"
19:         class="com.google.android.gms.maps.SupportMapFragment" />
20: </RelativeLayout>

```

4. Tambahkan FragmentActivity ke dalam MainActivity.java. Berikut ini code untuk activitynya

```

1. import com.google.android.gms.maps.GoogleMap;
2. import com.google.android.gms.maps.SupportMapFragment;
3.

```

```

4. import android.os.Bundle;
5. import android.support.v4.app.FragmentActivity;
6. import android.support.v4.app.FragmentManager;
7. import android.view.Menu;
8.
9. public class MainActivity extends FragmentActivity {
10.
11.     final int RQS_GooglePlayServices = 1;
12.     private GoogleMap myMap;
13.
14.     @Override
15.     protected void onCreate(Bundle savedInstanceState) {
16.         super.onCreate(savedInstanceState);
17.         setContentView(R.layout.activity_main);
18.         FragmentManager myFragmentManager =
19.             getSupportFragmentManager();
20.         SupportMapFragment mySupportMapFragment =
21.             (SupportMapFragment) myFragmentManager
22.                 .findFragmentById(R.id.map);
23.         myMap = mySupportMapFragment.getMap();
24.     }
25.
26.     @Override
27.     public boolean onCreateOptionsMenu(Menu menu) {
28.         getMenuInflater().inflate(R.menu.activity_main, menu);
29.         return true;
30.     }
31.
32. }

```

5. Tambahkan beberapa code berikut pada AndroidManifest.xml. Pada baris 31, gantilah dengan api key yang Anda peroleh dari Google console pada langkah D.

```

1: <?xml version="1.0" encoding="utf-8"?>
2: <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3:     package="com.map.sederhana.app"
4:     android:versionCode="1"
5:     android:versionName="1.0" >
6:
7:     <uses-sdk
8:         android:minSdkVersion="8"
9:         android:targetSdkVersion="16" />
10:
11:     <permission
12:         android:name="com.map.sederhana.app.permission.MAPS_RECEIVE"
13:         android:protectionLevel="signature" />
14:
15:     <uses-permission android:name="com.map.sederhana.app.permission.MAPS_RECEIVE" />
16:     <uses-permission android:name="android.permission.INTERNET" />
17:     <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
18:     <uses-permission
19:         android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
20:     <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
21:     <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
22:
23:     <uses-feature
24:         android:glEsVersion="0x00020000"
25:         android:required="true" />
26:
27:     <application
28:         android:allowBackup="true"
29:         android:icon="@drawable/ic_launcher"

```



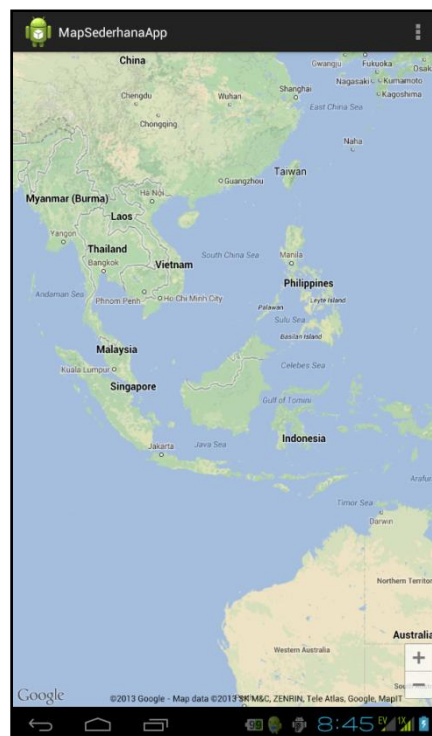
```

30:     android:label="@string/app_name"
31:     android:theme="@style/AppTheme" >
32:
33:     <meta-data
34:         android:name="com.google.android.maps.v2.API_KEY"
35:         android:value="your_api_key_from_google_console"/>
36:
37:     <activity
38:         android:name="com.map.sederhana.app.MainActivity"
39:         android:label="@string/app_name" >
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: </manifest>
    
```

6. Sampai tahap ini, kita sudah membuat aplikasi map sederhana yang siap untuk dijalankan.

F. Deployment

Bagian ini adalah melakukan debug aplikasi pada device Android. Saya menggunakan Smartfren Andro Tab dengan android versi 4.0.4. Gambar 1.15. adalah hasil akhir pembuatan project sederhana MapSederhanaApp.



Gambar 1.15. Tampilan aplikasi MapSederhanaApp



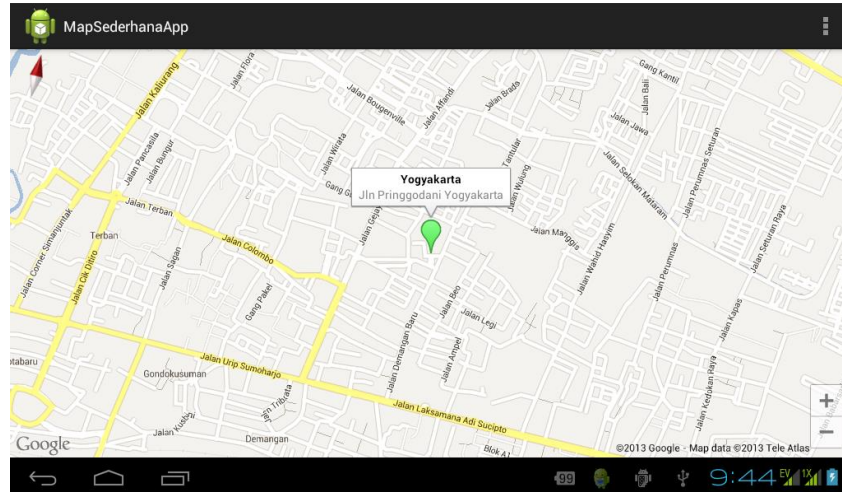
2. Menampilkan lokasi dengan marker

Nah pada sesi ini kita akan menambahkan sebuah marker diatas kota Yogyakarta. Masih menggunakan project MapSederhanaApp, tambahkan sourcecode baris 29-42 berikut pada MainActivity.java. Hasil akhir aplikasi tampak seperti Gambar 2.1.

```
1: import com.google.android.gms.maps.CameraUpdateFactory;
2: import com.google.android.gms.maps.GoogleMap;
3: import com.google.android.gms.maps.SupportMapFragment;
4: import com.google.android.gms.maps.model.BitmapDescriptorFactory;
5: import com.google.android.gms.maps.model.LatLng;
6: import com.google.android.gms.maps.model.MarkerOptions;
7:
8: import android.os.Bundle;
9: import android.support.v4.app.FragmentActivity;
10:     import android.support.v4.app.FragmentManager;
11:     import android.view.Menu;
12:
13:     public class MainActivity extends FragmentActivity {
14:
15:         final int RQS_GooglePlayServices = 1;
16:         GoogleMap myMap;
17:
18:         @Override
19:         protected void onCreate(Bundle savedInstanceState) {
20:             super.onCreate(savedInstanceState);
21:             setContentView(R.layout.activity_main);
22:             FragmentManager myFragmentManager =
23:                 getSupportFragmentManager();
24:             SupportMapFragment mySupportMapFragment =
25:                 (SupportMapFragment) myFragmentManager
26:                     .findFragmentById(R.id.map);
27:             myMap = mySupportMapFragment.getMap();
28:
29:             LatLng jogja = new LatLng(-7.775184, 110.392733);
30:
31:             MarkerOptions markerJogja = new MarkerOptions();
32:             markerJogja.position(jogja);
33:             markerJogja.title("Yogyakarta");
34:             markerJogja.snippet("Jln Pringgodani Yogyakarta");
35:             markerJogja.icon(BitmapDescriptorFactory
36:                 .defaultMarker(BitmapDescriptorFactory.HUE_GREEN));
37:
38:             myMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
39:             myMap.addMarker(markerJogja);
40:             myMap.getUiSettings().setCompassEnabled(true);
41:             myMap.getUiSettings().setZoomControlsEnabled(true);
42:             myMap.animateCamera(CameraUpdateFactory.newLatLngZoom(jogja, 15));
43:         }
44:
45:         @Override
46:         public boolean onCreateOptionsMenu(Menu menu) {
47:             getMenuInflater().inflate(R.menu.activity_main, menu);
48:             return true;
49:         }
50:     }
```

```

49:         }
50:     }
    
```



Gambar 2.1. Menampilkan marker diatas wilayah Yogyakarta

3. Menghitung jarak antara dua lokasi

Jarak antara dua lokasi dapat diketahui menggunakan method `distanceto()`. Masih menggunakan project `MapSederhanaApp`, kita perlu menambahkan satu lokasi baru pada `MainActivity.java` agar bisa menghitung jarak antara dua lokasi. Beberapa sourcecode tambahan saya bedakan dengan font berwarna merah.

```

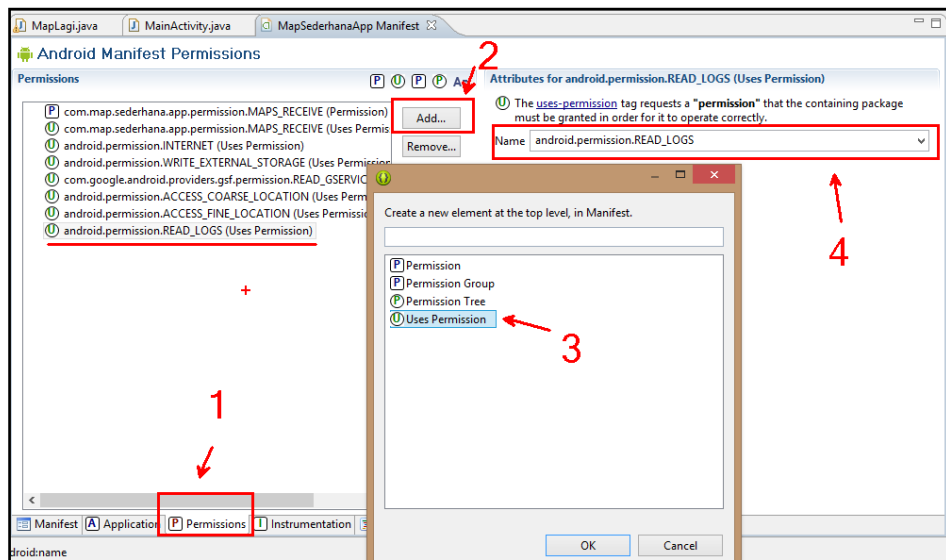
1: import com.google.android.gms.maps.CameraUpdateFactory;
2: import com.google.android.gms.maps.GoogleMap;
3: import com.google.android.gms.maps.SupportMapFragment;
4: import com.google.android.gms.maps.model.BitmapDescriptorFactory;
5: import com.google.android.gms.maps.model.LatLng;
6: import com.google.android.gms.maps.model.MarkerOptions;
7:
8: import android.location.Location;
9: import android.os.Bundle;
10: import android.support.v4.app.FragmentActivity;
11: import android.support.v4.app.FragmentManager;
12: import android.util.Log;
13: import android.view.Menu;
14: import android.widget.Toast;
15:
16: public class MainActivity extends FragmentActivity {
17:
18:     final int RQS_GooglePlayServices = 1;
19:     GoogleMap myMap;
20:
21:     @Override
22:     protected void onCreate(Bundle savedInstanceState) {
23:         super.onCreate(savedInstanceState);
    
```

```

24:         setContentView(R.layout.activity_main);
25:         FragmentManager myFragmentManager =
26:             getSupportFragmentManager();
27:         SupportMapFragment mySupportMapFragment =
28:             (SupportMapFragment) myFragmentManager
29:                 .findFragmentById(R.id.map);
30:         myMap = mySupportMapFragment.getMap();
31:
32:         LatLng jogja = new LatLng(-7.775184, 110.392733);
33:         LatLng ugm = new LatLng(-7.769581, 110.377789);
34:
35:         MarkerOptions markerJogja = new MarkerOptions();
36:         markerJogja.position(jogja);
37:         markerJogja.title("Yogyakarta");
38:         markerJogja.snippet("Jln Pringgodani Yogyakarta");
39:         markerJogja.icon(BitmapDescriptorFactory
40:             .defaultMarker(BitmapDescriptorFactory.HUE_GREEN));
41:
42:         MarkerOptions markerUgm = new MarkerOptions();
43:         markerUgm.position(ugm);
44:         markerUgm.title("UGM");
45:         markerUgm.snippet("Jln Kaliurang Yogyakarta");
46:         markerUgm.icon(BitmapDescriptorFactory
47:             .defaultMarker(BitmapDescriptorFactory.HUE_BLUE));
48:
49:         myMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
50:         myMap.addMarker(markerJogja);
51:         myMap.addMarker(markerUgm);
52:         myMap.getUiSettings().setCompassEnabled(true);
53:         myMap.getUiSettings().setZoomControlsEnabled(true);
54:
55:         myMap.animateCamera(CameraUpdateFactory.newLatLngZoom(jogja, 15));
56:
57:         Location lokasiA = new Location("lokasi_a");
58:         lokasiA.setLatitude(jogja.latitude);
59:         lokasiA.setLongitude(jogja.longitude);
60:
61:         Location lokasiB = new Location("lokasi_b");
62:         lokasiB.setLatitude(ugm.latitude);
63:         lokasiB.setLongitude(ugm.longitude);
64:
65:         Double distance = (double) lokasiA.distanceTo(lokasiB);
66:         String jarak = String.valueOf(distance);
67:         Log.i("jarak", jarak);
68:         Toast.makeText(this, "jarak : " + jarak + " meter",
69:             Toast.LENGTH_LONG).show();
70:     }
71:
72:     @Override
73:     public boolean onCreateOptionsMenu(Menu menu) {
74:         getMenuInflater().inflate(R.menu.activity_main, menu);
75:         return true;
76:     }
77:
78: }

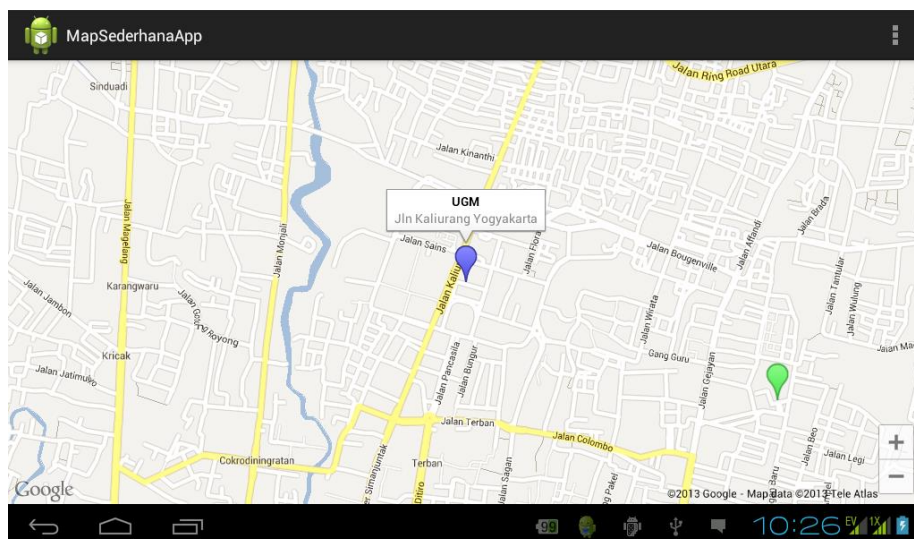
```

Supaya hasil perhitungan jarak bisa dibaca melalui logcat, kita harus membuka kembali AndroidManifest.xml untuk menambahkan uses-permission READ_LOGS. Cara menambahkan uses-permission ini perhatikan urutan angka pada Gambar 2.2. Hasil aplikasi ini ditunjukkan seperti Gambar 2.3.

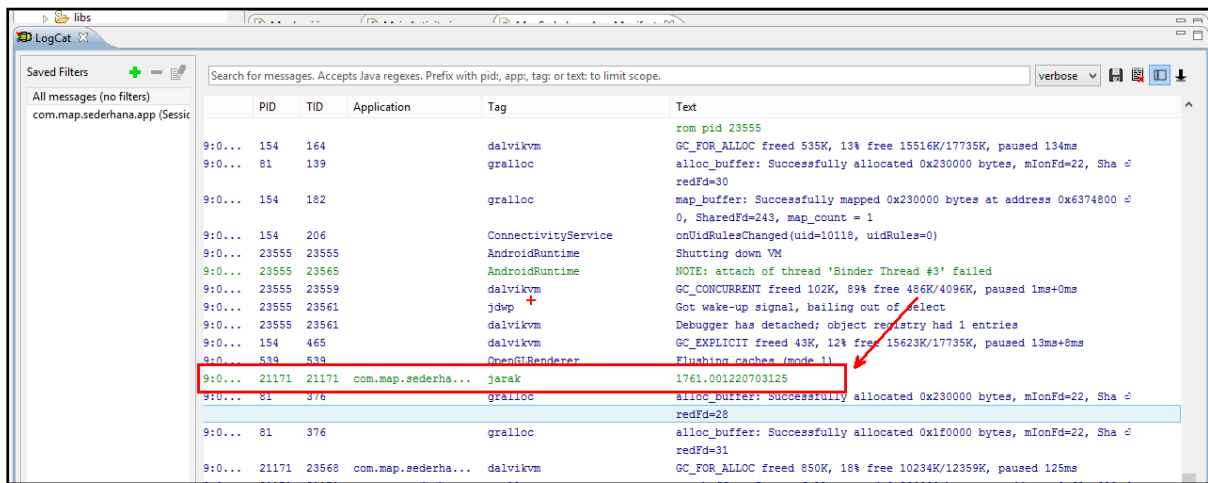


Gambar 2.2. Menambahkan uses-permission pada AndroidManifest.xml

Sampai disini, Jalankan Aplikasi dan lihat hasilnya pada logcat (**window > show view > logcat**). Tampilan logcat seperti Gambar 2.4.



Gambar 2.3. tampilan menghitung jarak antara 2 lokasi



Gambar 2.4. Tampilan logcat untuk melihat hasil perhitungan jarak

4. Menampilkan jalur (direction)

<comming soon>

5. Menggunakan GPS

<comming soon>