

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 ke depan. Pada versi dua ini, ada banyak perubahan mulai dari proses pendaftaran API Key hingga cara menampilkan Maps pada android.

Tutorial ini akan membahas langkah demi langkah cara baru menampilkan Maps pada android. Berikut ini urutan secara umum yang harus dilakukan.

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

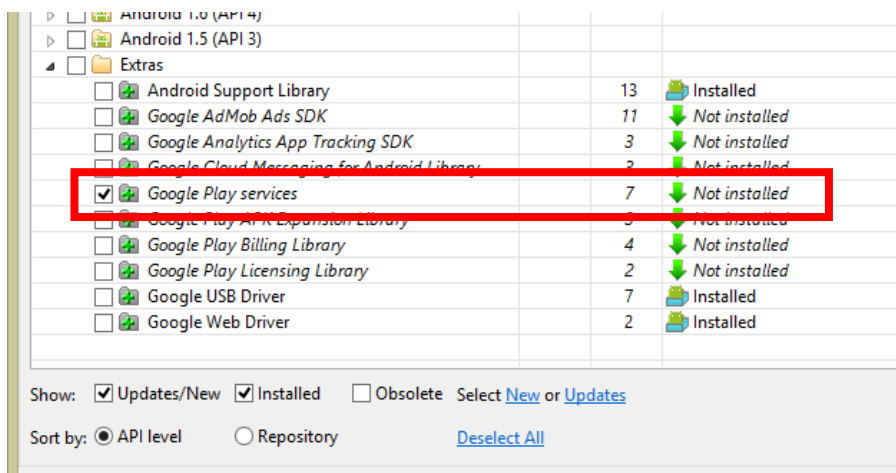
G. Deployment

Sementara, proses debug Google Maps belum bisa dilakukan pada emulator. Harus menyiapkan device Android untuk proses debugging. Materi dalam tutorial ini berhasil di-debug pada Sony Ericsson Xperia Active (ICS) melalui ADB. Berikut pembahasan masing-masing langkah diatas.

A. Instal Google Play service pada ADT Eclipse

Sebelumnya harus dipastikan 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**

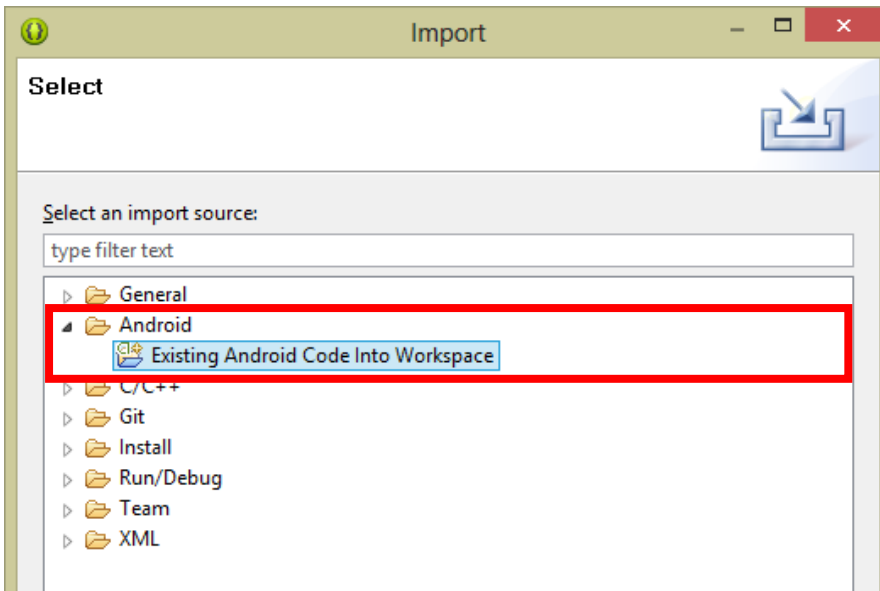


Gambar 1.1. Menginstal Google Play services pada ADT

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

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

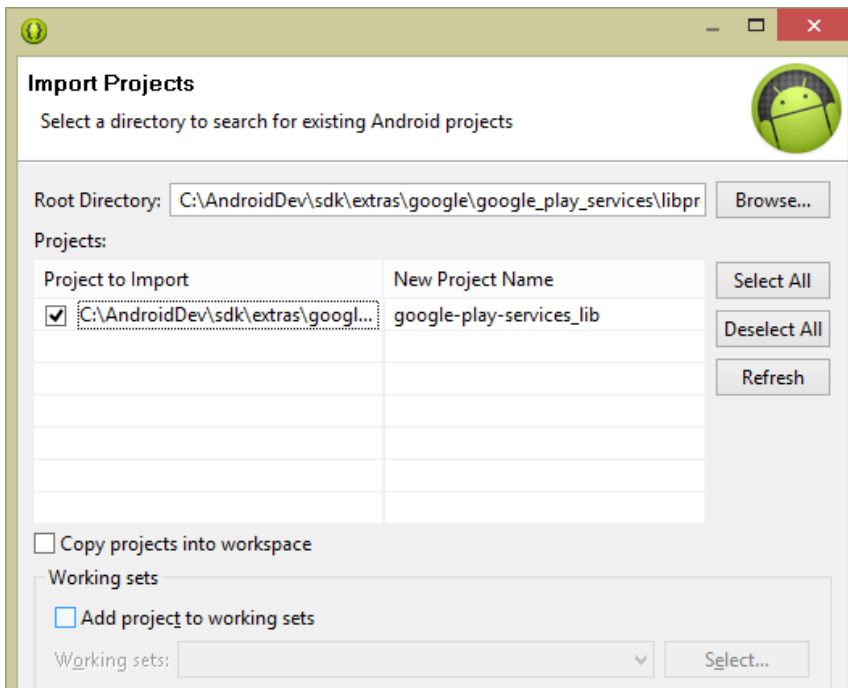
1. Klik **File > import > Existing Android Code Into Workspace**, perhatikan Gambar 1.2.



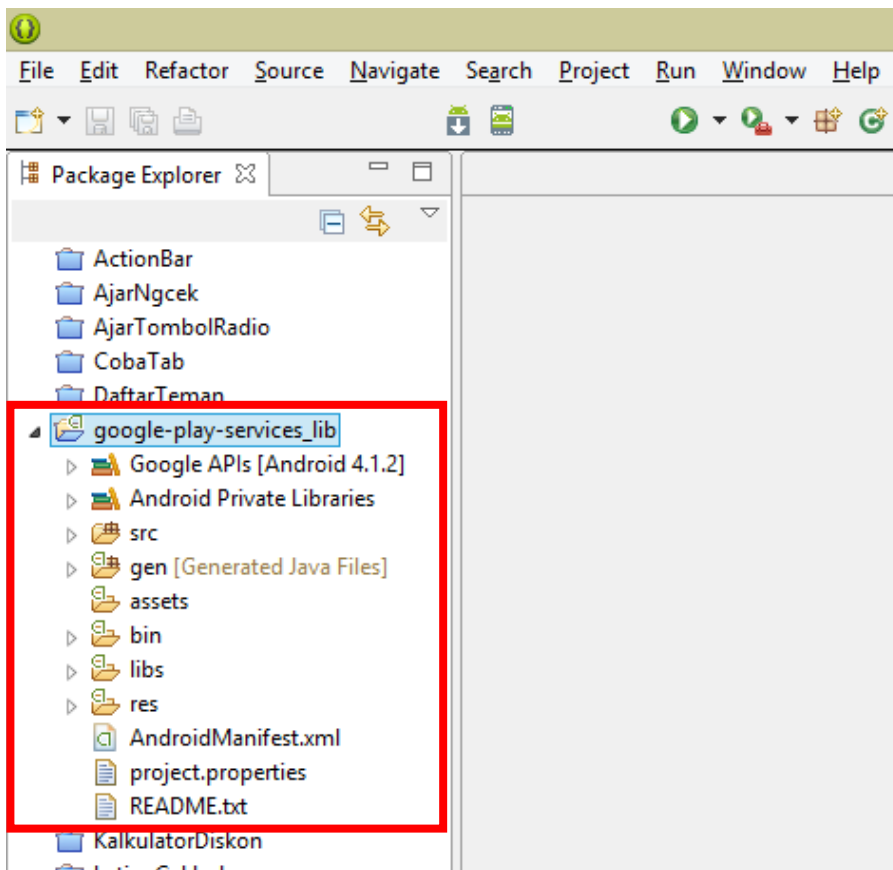
Gambar 1.2. Import library android

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 dimasukkan.



Gambar 1.3. Memasukkan library ke dalam workspace

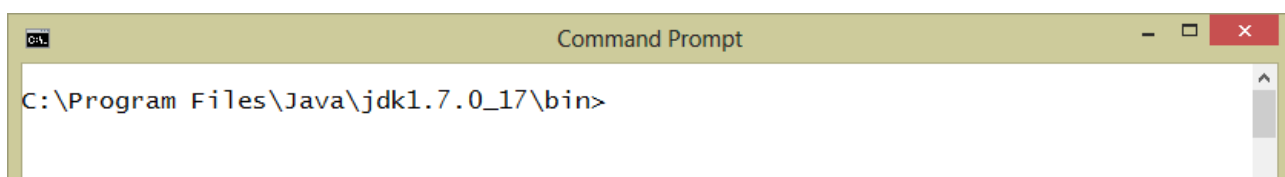


Gambar 1.4. Tampilan Package Explorer setelah ditambah library

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 didalam folder **C:\Program Files\Java\<jdk_version>\bin** dengan seperti Gambar 1.5.



Gambar 1.5. Masuk ke direktori bin pada folder instalasi java

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

```
keytool -list -v -keystore
"c:\users\<nama_users>\.android\debug.keystore" -alias
androiddebugkey -storepass android -keypass android
```

4. 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. Akan menggunakan **key SHA1** untuk mendaftarkan api key ke **Google Apis Console**.

```
Command Prompt
C:\Program Files\Java\jdk1.7.0_17\bin>keytool -list -v -keystore "c:\users\piusangggoro\.android\debug.keystore" -alias androiddebugkey -storepass android -keypass android
```

Gambar 1.6. proses generate debug api key

```
Command Prompt
C:\Program Files\Java\jdk1.7.0_17\bin>keytool -list -v -keystore "c:\users\piusangggoro\.android\debug.keystore" -alias androiddebugkey -storepass android -keypass android
Alias name: androiddebugkey
Creation date: Apr 30, 2013
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: 598f69db
Valid from: Tue Apr 30 13:36:24 ICT 2013 until: Thu Apr 23 13:36:24 ICT 2043
Certificate fingerprints:
MD5: BA:EA:91:A6:8A:64:01:F2:24:48:41:0C:25:5A:CB:DB
SHA1: 8F:6E:6A:92:07:24:69:54:65:C1:49:56:EB:47:30:97:98:83:B4:4E
SHA256: 97:87:22:98:68:20:FE:A8:05:21:CE:7B:7F:F6:90:31:51:CB:2B:83:C7:
5A:43:AB:BE:D5:FE:DC:92:6C:B0:ED
Signature algorithm name: SHA256withRSA
Version: 3

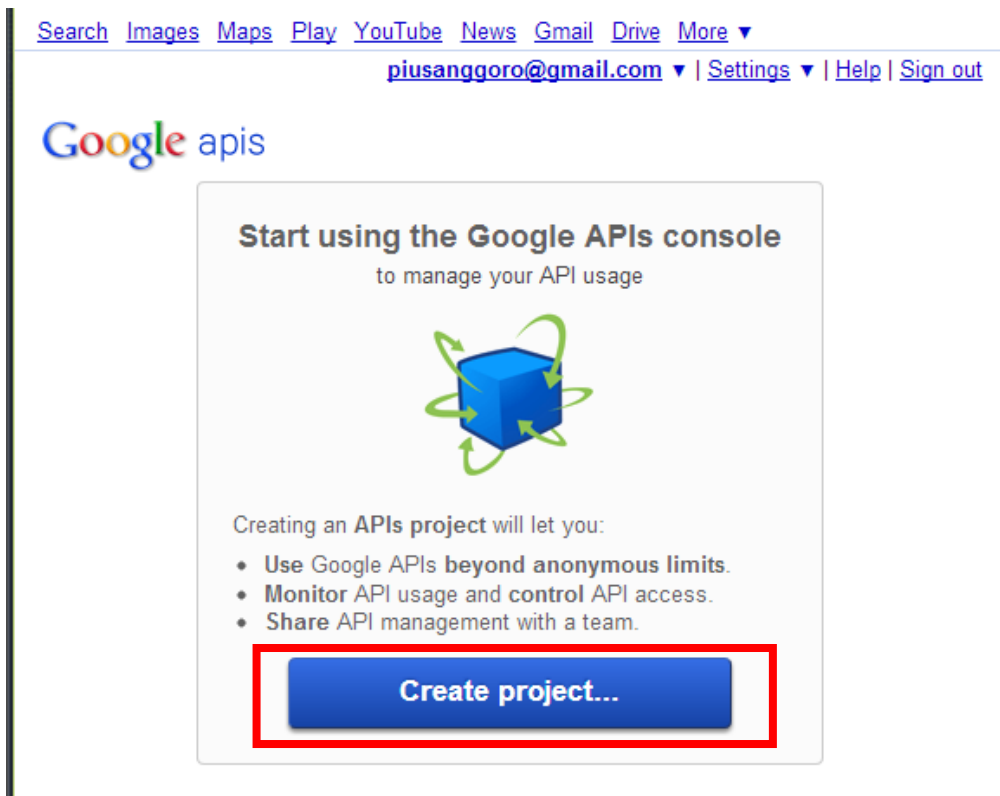
Extensions:
#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 88 88 5D 85 5E 94 7C 46 29 C7 7A 18 34 7E 1F 7A ..].^..F).z.4..z
0010: 11 D3 9E E6 ....
]
]
```

Gambar 1.7. Hasil proses generate debug api key

5. Copy key SHA1 ke dalam notepad atau tahan dulu command prompt nya. Akan dilanjutkan ke langkah berikutnya.

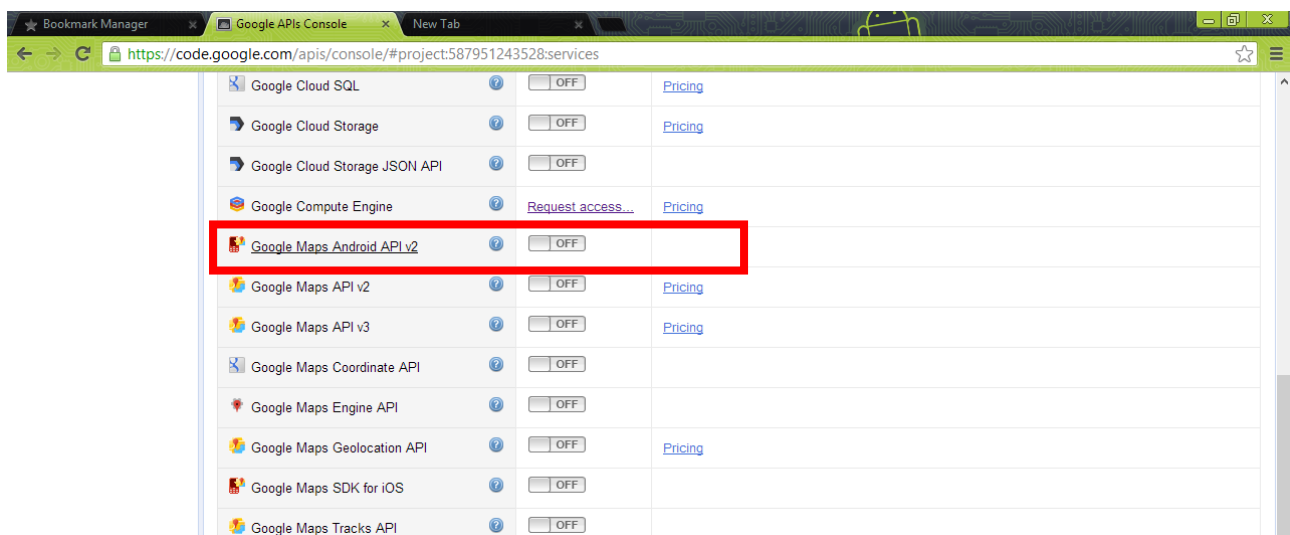
D. Mendaftarkan pada Google APIs Console

1. Jalankan browser, masuk ke alamat <https://code.google.com/apis/console/>. 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 akan diarahkan ke halaman **Service**. Pada halaman ini 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, pindah ke halaman **API Acess**. Pada halaman ini klik tombol **Create new Android Key** seperti pada Gambar 1.10

19.4 **Third Party Beneficiaries.** You acknowledge and agree that each member of the group of companies of which Google is the parent, and each of the Indemnified Parties, will be third party beneficiaries to the Terms and that such other companies will be entitled to directly enforce, and rely upon, any provision of the Terms that confers a benefit on (or rights in favor of) them. Other than this, no other person or company will be a third party beneficiary to the Terms.

19.5 **Assignment.** The Terms may be assigned by Google and will inure to the benefit of Google, its successors, and assigns.

19.6 **Governing Law and Jurisdiction; Injunctive Relief.** The Terms, and your relationship with Google under the Terms, are governed by the laws of the State of California, USA, without regard to its conflict of laws provisions. You and Google agree to submit to the exclusive, personal jurisdiction of the federal or state courts of Santa Clara County, California, USA, to resolve any legal matter arising from or related to the Terms. Notwithstanding this, you agree that Google will be allowed to apply for injunctive remedies (or an equivalent type of urgent legal relief) in any jurisdiction.

19.7 **Complete Agreement.** These Terms constitute the whole legal agreement between you and Google in connection with, and govern your use of, the Service and Content. These Terms completely replace and supersede any prior agreements between you and Google, written or oral, in connection with the Service and Content.

0 of 1 terms of service accepted.
Google Maps/Earth APIs

I agree to these terms.

Accept Decline

API Project

- Overview
- Services
- Team
- API Access**
- Reports
- Quotas

API Access
To prevent abuse, Google places limits on API requests. Using a valid OAuth token or API key allows you to exceed anonymous limits by connecting requests back to your project.

Authorized API Access

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private. A single project may contain up to 20 client IDs. [Learn more](#)

[Create an OAuth 2.0 client ID...](#)

Simple API Access
Use API keys to identify your project when you do not need to access user data. [Learn more](#)

Key for browser apps (with referers)

API key: AIzaSyDuMLscGAzB-Th9XZZveXDw9hQgHfPoBw

Referers: Any referer allowed

Activated on: May 27, 2013 12:21 AM

Activated by: piusanggoro@gmail.com - you

[Generate new key...](#)
[Edit allowed referers...](#)
[Delete key...](#)

[Create new Server key...](#) [Create new Browser key...](#) [Create new Android key...](#) [Create new iOS key...](#)

Notification Endpoints
Use notification endpoints to identify domains that may receive webpush notifications from your API. [Learn more](#)

Gambar 1.10. Halaman API Access

4. Copy **key SHA1** yang diperoleh 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

8F:6E:6A:92:07:24:69:54:65:C1:49:56:EB:47:30:97:98:83:B4:4E;ang.latian.peta
ang.latian.peta adalah package yang akan dibuat sebagai project aplikasi android.

Simple API Access
Use API keys to identify your project when you do not need to access user data. [Learn more](#)

Key for browser apps (with referers)

API key: AIzaSyDuMLscGAzB-Th9XZZveXDw9hQgHfPoBw

Referers: Any referer allowed

Activated on: May 27, 2013 12:21 AM

Activated by: piusanggoro@gmail.com - you

[Generate new key...](#)
[Edit allowed referers...](#)
[Delete key...](#)

[Create new Server key...](#) [Create new Browser key...](#) [Create new Android key...](#) [Create new iOS key...](#)

Notification Endpoints
Use notification endpoints to identify domains that may receive webpush notifications from your API. [Learn more](#)

Accept requests from an Android application with one of the certificate fingerprints and package names listed below:

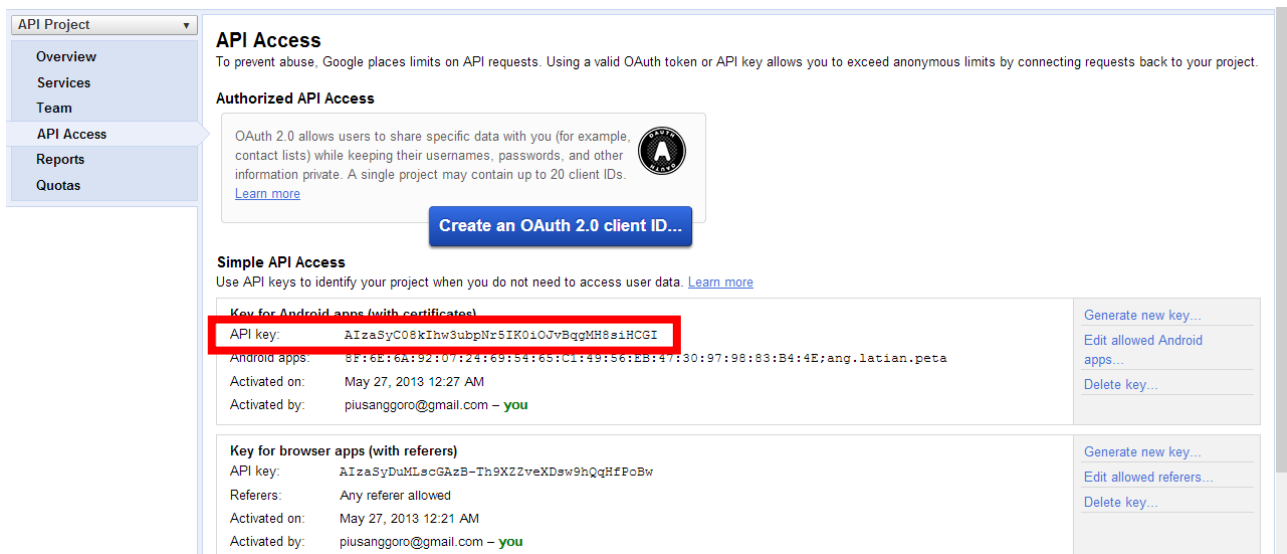
8F:6E:6A:92:07:24:69:54:65:C1:49:56:EB:47:30:97:98:83:B4:4E;ang.latian.peta

One SHA1 certificate fingerprint and package name (separated by a semicolon) per line. Example:
45:B5:E4:6F:36:AD:0A:98:94:B4:02:66:2B:12:17:F2:56:26:A0:E0;com.example

[Create](#) [Cancel](#)

Gambar 1.11. Memasukkan key SHA1 ke dalam API Project

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

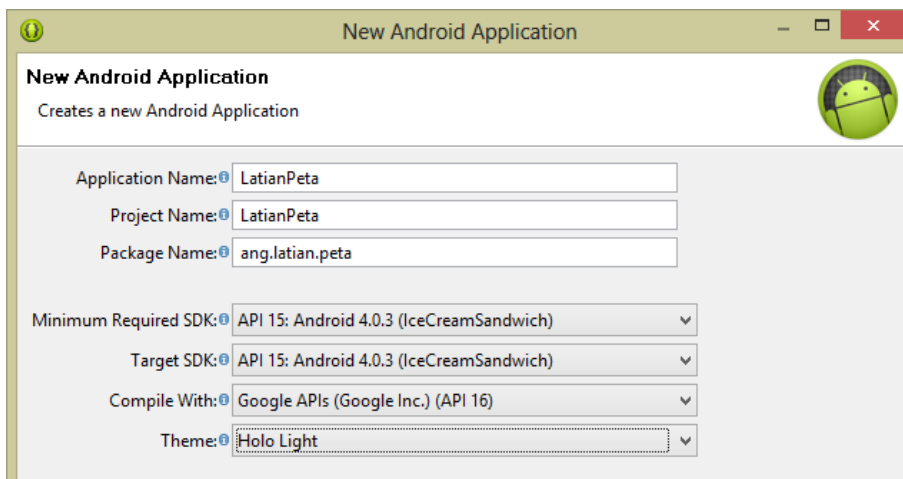


Gambar 1.12. Hasil generate API Key

E. Membuat project dengan target Google API

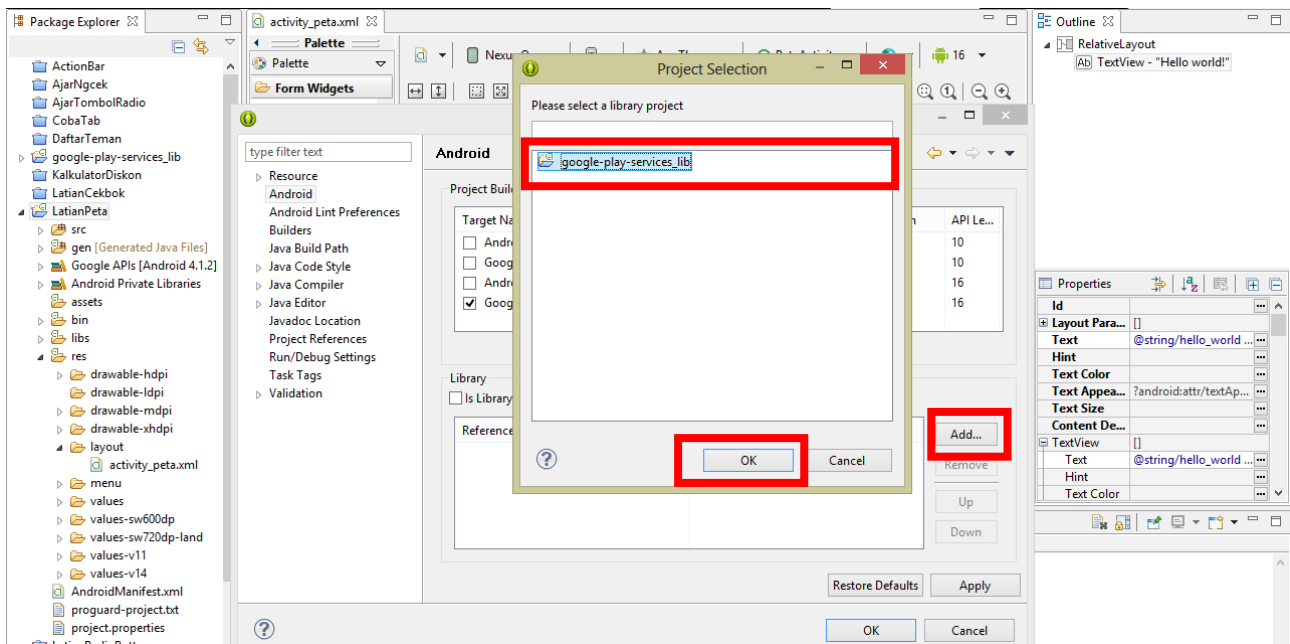
Pada tahap ini, akan dimulai membuat project paling sederhana yaitu menampilkan Google Map menggunakan **SupportFragment** dan **FragmentActivity**. Langkah-langkah detailnya sebagai berikut

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 : LatianPeta - Project Name : LatianPeta
Package Name : ang.latian.peta
Target SDK : API 15 - Compile With : Google APIs (API 16)



Gambar 1.13. Membuat project baru

2. Akan ditambahkan 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**.



Gambar 1.14. Menambahkan library ke dalam project

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

```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:name="com.google.android.gms.maps.MapFragment"
    map:cameraZoom="50"
/>
```

4. Tambahkan FragmentActivity ke dalam MainActivity.java. Berikut ini code untuk Activity-nya

```
package ang.latian.peta;
```

```
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapFragment;
```

```
import android.annotation.SuppressLint;
import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
```

```
public class PetaActivity extends Activity {
    private GoogleMap peta;

    @SuppressWarnings("NewApi")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_peta);

        peta = ((MapFragment)
            getFragmentManager().findFragmentById(R.id.map)).getMap();
        peta.setMapType(GoogleMap.MAP_TYPE_NORMAL);
    }
    ...
}
```


5. Tambahkan beberapa code berikut pada AndroidManifest.xml. gantilah dengan **api key** yang diperoleh dari Google console pada langkah D.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ang.latian.peta"
    android:versionCode="1"
    android:versionName="1.0" >
    ...

    <permission
        android:name="ang.latian.peta.permission.MAPS_RECEIVE"
        android:protectionLevel="signature" />
    <uses-permission android:name="ang.latian.peta.permission.MAPS_RECEIVE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission
        android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

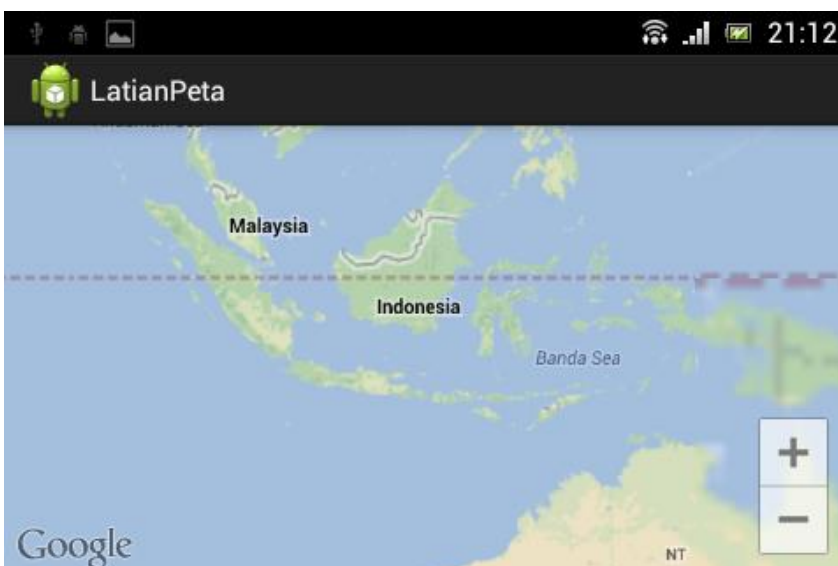
    <uses-feature
        android:glEsVersion="0x00020000"
        android:required="true" />
    <application
        ...

        <meta-data
            android:name="com.google.android.maps.v2.API_KEY"
            android:value="AIzaSyC08kIhw3ubpNr5IK0iOJvBqgMH8siHCGI" />
        ...
    </application>
</manifest>
```

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

F. Deployment

Bagian ini adalah melakukan debug aplikasi pada device Android. Gambar 1.15. adalah hasil akhir pembuatan project sederhana LatianPeta.



Gambar 1.15. Tampilan aplikasi LatianPeta