

## MODUL 8

### AKSES GOOGLE MAPS, GPS

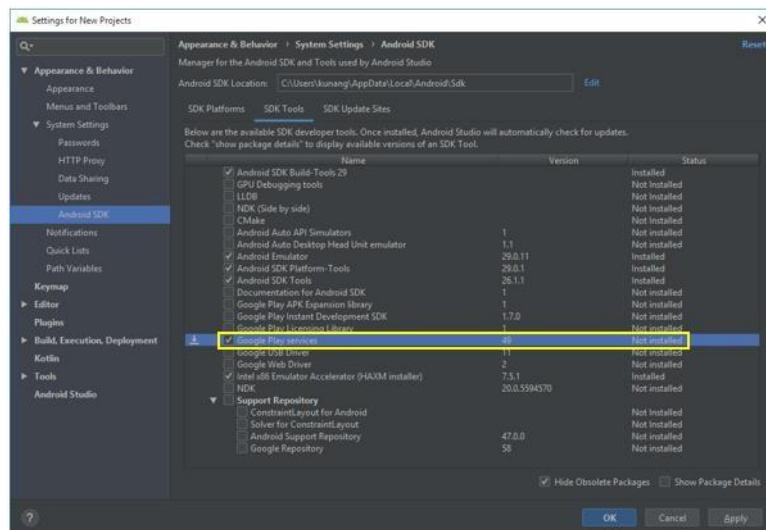
#### A. Persiapan Instalasi

Bab ini akan menampilkan Google Maps dengan menunjuk pada koordinat yang telah kita tentukan. Namun sebelum menampilkan Maps, ada beberapa tahapan yang harus dilakukan, yaitu: Instalasi Google Play Service dan mendaftar pada Google Maps API untuk mendapatkan API Key.

Silahkan download project yang telah disediakan pada link [berikut](#)

Ekstrak dan buka proyek Android tersebut, pastikan bahwa Google Play services telah terinstall pada Android Studio dengan cara buka: Tools > SDK Manager,

maka akan muncul kotak dialog seperti pada gambar di bawah ini. Pilih tab SDK Tools dan ceklis pada Google Play Services, jika statusnya Not Installed, klik Apply pada bagian bawah untuk memulai proses instalasi. Proses dari langkah ini ditampilkan dalam Gambar 8.1.



Gambar 8.1 Instalasi Google Play Service

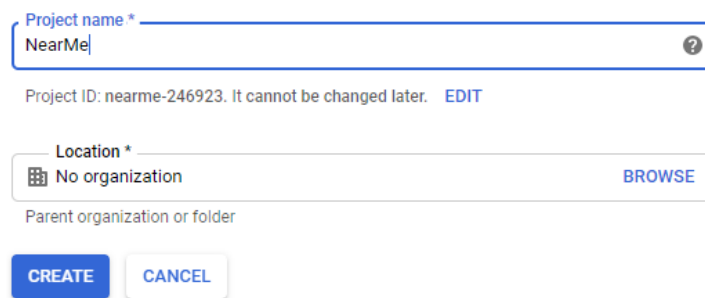
## B. Membuat API Key

API key diperlukan untuk mengakses server Google Maps. Key ini bersifat gratis dan dapat digunakan dengan aplikasi apa pun. Tahapan untuk membuat API Key adalah sebagai berikut:

1. Kunjungi link berikut: [Google developer console] (<https://console.developers.google.com/>), kemudian login menggunakan email gmail yang telah dimiliki. Satu akun gmail hanya bisa mendapatkan satu API key.
2. Membuat project baru dengan cara klik Create pada bagian dashboard, atau melalui dropdown Select a Project. Proses ini ditampilkan dalam Gambar 8.2 dan Gambar 8.3 Pada halaman New Project, isikan nama project dan lokasi, kemudian klik tombol Create.

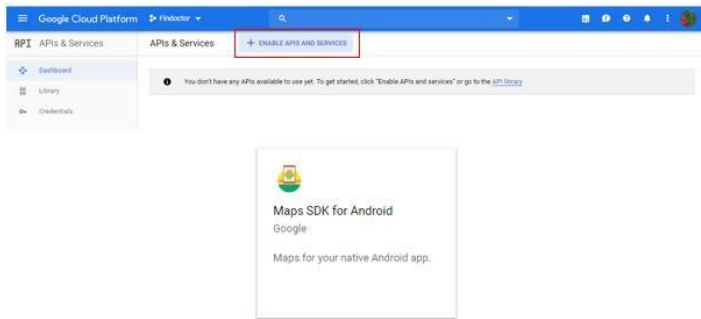


Gambar 8.2 Membuat Project Baru di Google API Part 1

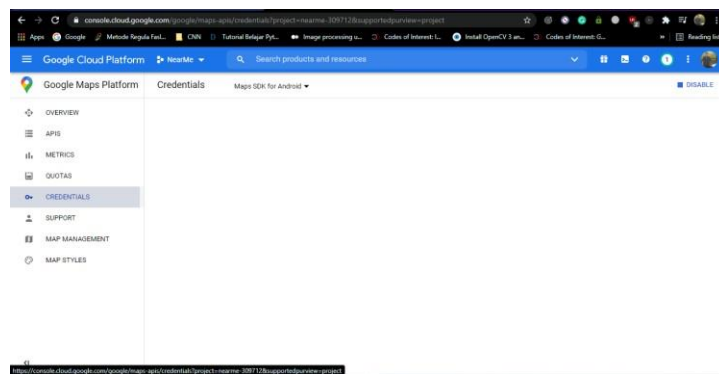
A screenshot of the 'New Project' form in the Google API console. It features a 'Project name' field with the text 'NearMe' and a help icon. Below it, the 'Project ID' is shown as 'nearme-246923' with an 'EDIT' link. The 'Location' field is set to 'No organization' with a 'BROWSE' button. At the bottom are 'CREATE' and 'CANCEL' buttons.

Gambar 8.3 Membuat Project Baru di Google API Part 2

3. Klik link *ENABLE APIS AND SERVICES* pada dashboard, dan pilih Maps SDK for Android. Proses ini ditampilkan dalam Gambar 8.4.
4. Klik tombol *ENABLE* . Setelah itu pilih tab *Credentials*, proses ini ditampilkan dalam Gambar 8.5.

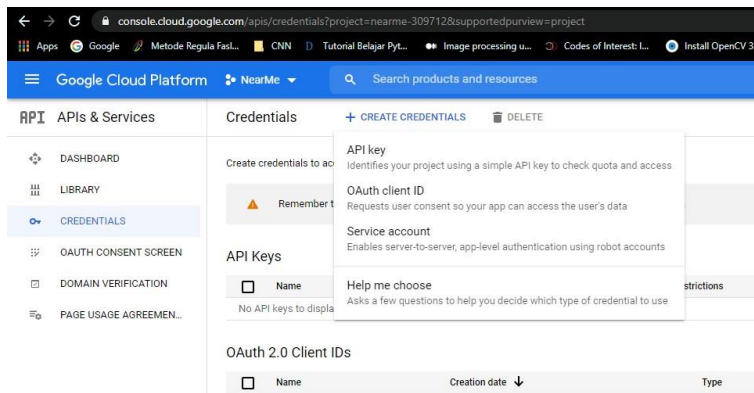


Gambar 8.4 Membuat Project Baru di Google API Part 3

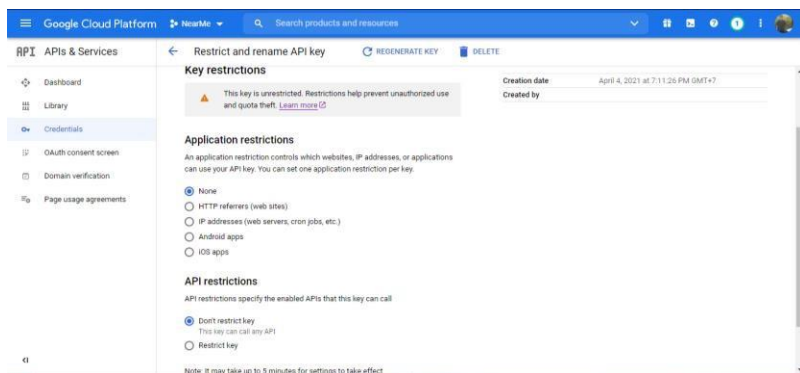


Gambar 8.5 Membuat Project Baru di Google API Part 4

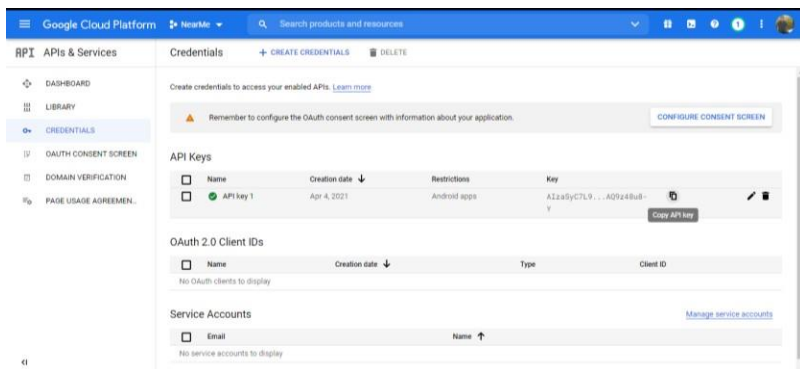
5. Klik Create Credentials pada dropdown, dan pilih api key untuk membuat key. Proses ini ditampilkan dalam Gambar 8.6. Maka akan muncul popup bahwa API key telah berhasil dibuat.
6. Pada bagian Restrict and rename API key, pilih Android apps, dan selanjutnya simpan perubahan. Proses ini ditampilkan dalam Gambar 8.7.
7. Setelah menyimpan pengaturan, copy API Key untuk digunakan di tahap selanjutnya. Proses ini ditampilkan dalam Gambar 8.8.



Gambar 8.6 Membuat Project Baru di Google API Part 5  
Gambar 65:



Gambar 8.7 Membuat Project Baru di Google API Part 6  
Gambar 66:




Gambar 8.8 Membuat Project Baru di Google API Part 7

### C. Menambahkan Kunci API ke Aplikasi

Pengguna tidak dipebolehkan memasukkan kunci API ke dalam sistem kontrol versi, jadi disarankan untuk menyimpan kunci tersebut dalam file `local.properties`, file ini terletak di direktori utama `project`. Informasi selengkapnya tentang file `local.properties`, dapat dipelajari dalam link [File properti Gradle](#). Teknik Gradle berikut menunjukkan cara menyimpan kunci API dalam file `local.properties` agar dapat direferensikan dengan aman oleh aplikasi pengguna.

Langkah-langkah untuk menambahkan kunci API adalah :

1. Buka `local.properties` di direktori [tingkat project](#), lalu tambahkan kode berikut ke file. Ganti `YOUR_API_KEY` dengan kunci API. Ditampilkan dalam Gambar 8.9.



```
MAPS_API_KEY=YOUR_API_KEY ✓
```

Gambar 8.9 Menambahkan Kunci API ke Aplikasi Part 1

2. Pada file `build.gradle` di tingkat aplikasi, tambahkan kode ini dalam elemen `defaultConfig`. Hal ini memungkinkan Android Studio membaca kunci Maps API dari file `local.properties` pada waktu build, lalu memasukkan variabel build `mapsApiKey` ke manifest Android. Proses ini ditampilkan dalam Gambar 8.10.



```
android {
    defaultConfig {
        // ...
        // Set the properties within 'local.properties' into a 'Properties' class so that values
        // within 'local.properties' (e.g. Maps API key) are accessible in this file.
        Properties properties = new Properties()
        if (rootProject.file("local.properties").exists()) {
            properties.load(rootProject.file("local.properties").newDataInputStream())
        }

        // Inject the Maps API key into the manifest
        manifestPlaceholders = [ mapsApiKey : properties.getProperty("MAPS_API_KEY", "") ]
    }
}
```

Gambar 8.10 Menambahkan Kunci API ke Aplikasi Part 2

3. Dalam file AndroidManifest.xml, tambahkan potongan kode berikut sebagai turunan dari elemen application. Proses ini ditampilkan dalam Gambar 8.11.

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="{mapsApiKey}" />
```

Gambar 8.11 Menambahkan Kunci API ke Aplikasi Part 3

4. Catatan: Membatasi Kunci API akan meningkatkan keamanan aplikasi dengan memastikan hanya permintaan sah yang dibuat dengan Kunci API. Sebaiknya ikuti petunjuk untuk menetapkan pembatasan pada Kunci API. Untuk informasi selengkapnya, lihat [disini](#), namun pada Latihan saat ini, belum wajib untuk diterapkan

#### D. Menampilkan Maps pada Android Studio

Proses ini memiliki beberapa Langkah, diantaranya adalah:

1. Silahkan Cek MapsMarkerActivity.Java: lengkapi apabila terdapat coding yang kurang. Penjelasan lebih lanjut dapat dilihat dalam Gambar 8.12 dan Gambar 8.13

```
package com.example.mapwithmarker;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

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;

/**
 * An activity that displays a Google map with a marker (pin) to indicate a particular location.
 */
public class MapsMarkerActivity extends AppCompatActivity
    implements OnMapReadyCallback {
```

Gambar 8.12 Menampilkan Maps pada Android Studio Part 1

2. Menambahkan Kunci API ke dalam aplikasi: Edit file gradle.properties project. Tempelkan kunci API ke nilai properti GOOGLE\_MAPS\_API\_KEY. Saat membuat aplikasi, Gradle menyalin kunci API ke dalam manifest Android untuk aplikasi tersebut. Proses ini ditampilkan dalam Gambar 8.14.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // Retrieve the content view that renders the map.
    setContentView(R.layout.activity_maps);
    // Get the SupportMapFragment and request notification
    // when the map is ready to be used.
    SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
        .findFragmentById(R.id.map);
    mapFragment.getMapAsync(this);
}

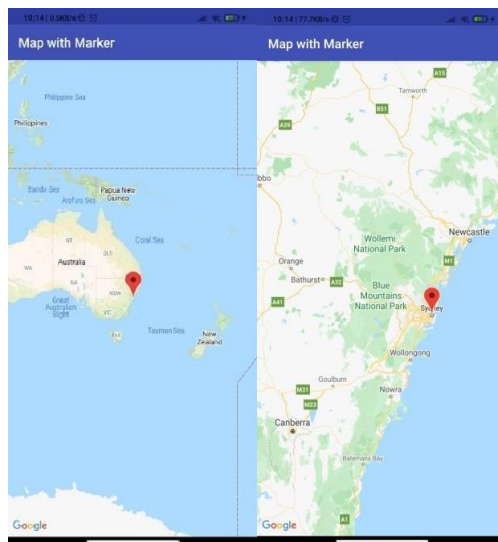
```

Gambar 8.13 Menampilkan Maps pada Android Studio Part 2

GOOGLE\_MAPS\_API\_KEY=PASTE-YOUR-API-KEY-HERE

Gambar 8.14 Menampilkan Maps pada Android Studio Part 3

3. Jika default program di atas dijalankan, maka akan muncul Google Maps dengan maker yang mengarah secara default ke Sydney, seperti pada Gambar 8.15 berikut:



Gambar 8.15 Menampilkan Maps pada Android Studio Part 4

4. Silahkan melakukan pengeditan kode tersebut sehingga dapat menampilkan lokasi tertentu seperti gambar berikut: Contoh dalam Gambar 8.16: alun2 Batu. Cara mendapatkan latitude dan longitude dapat membaca artikel [berikut](#).



Gambar 8.16 Menampilkan Maps pada Android Studio Part 5

## E. Memahami Kode

Bagian ini akan menjelaskan perihal yang paling signifikan dari sebuah aplikasi **MapWithMarker** untuk membantu memahami cara membuat aplikasi yang serupa.

1. Memeriksa manifest Android: Silahkan membuka file `AndroidManifest.xml` di Aplikasi, tambahkan elemen meta-data untuk meletakkan versi layanan Google Play yang digunakan dalam mengkompilasi Aplikasi. Bagian ini dijelaskan dalam Gambar 8.17

```
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
```

Gambar 8.17 Menyematkan Kode di Meta-data Part 1

2. Menambahkan elemen meta-data yang merupakan kunci API yang telah dibuat sebelumnya. Petakan nilai untuk kunci API ke dalam string `google_maps_key` saat membuat aplikasi, Gradle akan menyalin kunci API dari file `gradle.properties` project yang telah dibuat ke nilai string. Proses ini ditampilkan dalam Gambar 8.18



```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="@string/google_maps_key" />
```

Gambar 8.18 Menambahkan Kode di Meta-data Part 2

3. Untuk melihat bagaimana kunci API dipetakan ke nilai string, silahkan perhatikan build.gradle aplikasi yang telah dibuat. File build.gradle ini berisi baris yang ditampilkan dalam Gambar 8.19 yang memetakan string google\_maps\_key ke dalam property gradle GOOGLE\_MAPS\_API\_KEY.

```
resValue "string", "google_maps_key",
    (project.findProperty("GOOGLE_MAPS_API_KEY") ?: "")
```

Gambar 8.19 Pemetaan Kunci API ke Nilai String

4. Contoh Manifest Lengkap akan ditampilkan dalam Gambar 8.20 dan Gambar 8.21 Berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.mapwithmarker">

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

Gambar 8.20 Kode Manifes Part 1

```

<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />

<!--
    The API key for Google Maps-based APIs.
-->
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="@string/maps_api_key" />

<activity
    android:name=".MapsMarkerActivity"
    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>

```

Gambar 8.21 Kode Manifes Part 2