{  
 address: string,   
 location: LatLng,  
 placeId: string,  
 bounds: LatLngBounds,  
 componentRestrictions: GeocoderComponentRestrictions,  
 region: string  
}

**Parameter yang diperlukan:** Anda harus menyediakan salah satu, dan hanya satu, dari bidang-bidang berikut:

* address — Alamat yang ingin Anda geocode.
* location — LatLng (atau LatLngLiteral) yang Anda inginkan dalam memperoleh alamat terdekat yang bisa dibaca orang. Geocoder akan *membalik geocode*. Lihat [Geocoding Terbalik](https://developers.google.com/maps/documentation/javascript/geocoding?hl=id#ReverseGeocoding) untuk informasi selengkapnya.
* placeId — ID tempat dari lokasi yang Anda inginkan dalam memperoleh alamat terdekat yang bisa dibaca orang. ID tempat adalah sebuah identifier unik yang bisa digunakan bersama Google API lainnya. Misalnya, Anda bisa menggunakan placeId yang dikembalikan oleh [Google Maps Roads API](https://developers.google.com/maps/documentation/roads/snap?hl=id) untuk mendapatkan alamat titik yang telah diikat. Untuk informasi selengkapnya tentang ID tempat, lihat [ringkasan ID tempat](https://developers.google.com/places/place-id?hl=id). Jika Anda meneruskan sebuah placeId, geocoder akan *membalik geocode*. Lihat [Geocoding Terbalik](https://developers.google.com/maps/documentation/javascript/geocoding?hl=id#ReverseGeocoding) untuk informasi selengkapnya.

**Parameter opsional:**

* bounds — LatLngBounds yang akan digunakan untuk mencondongkan hasil geocode secara lebih jelas. Parameter bounds hanya akan memengaruhi, tidak sepenuhnya membatasi, hasil dari geocoder. (Untuk informasi selengkapnya lihat [Mencondongkan Viewport](https://developers.google.com/maps/documentation/javascript/geocoding?hl=id" \l "GeocodingViewports) di bawah ini.)
* componentRestrictions — Digunakan untuk membatasi hasil ke daerah tertentu. (Untuk informasi selengkapnya, lihat [Pemfilteran Komponen](https://developers.google.com/maps/documentation/javascript/geocoding?hl=id" \l "ComponentFiltering) di bawah ini.)
* region — Kode region, ditetapkan sebagai [IANA language region subtag](http://www.iana.org/assignments/language-subtag-registry). Umumnya, tag-tag ini memetakan langsung ke nilai yang berisi dua karakter ccTLD ("top-level domain") yang sudah dikenal. Parameter region hanya akan memengaruhi, tidak sepenuhnya membatasi, hasil dari geocoder. (Untuk informasi selengkapnya, lihat [Mencondongkan Kode Region](https://developers.google.com/maps/documentation/javascript/geocoding?hl=id" \l "GeocodingRegionCodes) di bawah ini.)

results[]: {  
 types[]: string,  
 formatted\_address: string,  
 address\_components[]: {  
   short\_name: string,  
   long\_name: string,  
   postcode\_localities[]: string,  
   types[]: string  
 },  
 partial\_match: boolean,  
 place\_id: string,  
 postcode\_localities[]: string,  
 geometry: {  
   location: LatLng,  
   location\_type: GeocoderLocationType  
   viewport: LatLngBounds,  
   bounds: LatLngBounds  
 }  
}

Bidang-bidang ini dijelaskan di bawah:

* types[] adalah larik yang menunjukkan *tipe* hasil yang dikembalikan. Larik ini berisi nol atau beberapa tag yang mengidentifikasi tipe fitur yang dikembalikan dalam hasil. Misalnya, geocode "Chicago" akan mengembalikan "locality" yang menunjukkan "Chicago" adalah kota, juga mengembalikan "political" yang menunjukkan Chicago adalah entitas politik.
* formatted\_address adalah string berisi alamat yang bisa dibaca orang untuk lokasi ini. Sering kali alamat ini sama dengan "alamat pos", yang terkadang berbeda antara satu negara dengan negara lainnya. (Perhatikan, beberapa negara, seperti Britania Raya, tidak memperbolehkan distribusi alamat pos sebenarnya karena adanya pembatasan pemberian lisensi.) Alamat ini biasanya terdiri dari satu atau beberapa *komponen alamat*. Misalnya, alamat "111 8th Avenue, New York, NY" berisi komponen alamat terpisah untuk "111 8th Avenue" (alamat jalan), "New York" (kota) dan "NY" (negara bagian AS). Komponen alamat tersebut tercantum di bawah ini. Untuk informasi selengkapnya tentang tipe, lihat [Tipe](https://developers.google.com/maps/documentation/javascript/geocoding?hl=id" \l "GeocodingAddressTypes) di bawah ini.
* address\_components[] adalah larik berisi komponen alamat terpisah, seperti dijelaskan di atas.
* partial\_match menunjukkan geocoder tidak mengembalikan hasil yang benar-benar cocok dengan permintaan asal, meskipun geocoder bisa mencocokkan sebagian dari alamat yang diminta. Anda mungkin ingin memeriksa permintaan asal untuk mengetahui adanya salah eja dan/atau alamat yang tidak lengkap.

Kecocokan parsial paling sering terjadi untuk alamat jalan yang tidak ditemukan di lokasi yang Anda teruskan dalam permintaan. Kecocokan parsial juga mungkin dikembalikan bila sebuah permintaan memiliki kecocokan terhadap dua atau beberapa lokasi di daerah yang sama. Misalnya, "21 Henr St, Bristol, UK" akan mengembalikan kecocokan parsial untuk Henry Street dan Henrietta Street. Perhatikan, jika permintaan menyertakan komponen alamat yang salah eja, layanan geocoding mungkin akan menyarankan alamat alternatif. Saran yang terpicu melalui cara ini juga akan dinilai sebagai kecocokan parsial.

* place\_id adalah identifier unik suatu tempat, yang bisa digunakan bersama Google API lainnya. Misalnya, Anda bisa menggunakan place\_id bersama pustaka [Google Places API](https://developers.google.com/maps/documentation/javascript/places?hl=id) untuk mendapatkan detail bisnis setempat, seperti nomor telepon, jam buka, ulasan pengguna, dan lainnya. Lihat [ringkasan ID tempat](https://developers.google.com/places/place-id?hl=id).
* postcode\_localities[] adalah larik yang menunjukkan semua lokalitas yang dimuat dalam kode pos. Ini hanya ada bila hasilnya adalah kode pos yang berisi beberapa lokalitas.
* geometry berisi informasi berikut:
* location berisi nilai geocode garis lintang dan garis bujur. Perhatikan, kita mengembalikan lokasi ini sebagai objek LatLng, bukan sebagai string yang diformat.
* location\_type menyimpan data tambahan tentang lokasi yang ditetapkan. Nilai-nilai berikut saat ini didukung:
* ROOFTOP menunjukkan hasil yang dikembalikan mencerminkan sebuah geocode akurat.
* RANGE\_INTERPOLATED menunjukkan hasil yang dikembalikan mencerminkan perkiraan (biasanya di jalan) interpolasi antara dua titik tepat (seperti persimpangan). Hasil interpolasi umumnya dikembalikan bila rooftop-geocode tidak tersedia untuk alamat jalan.
* GEOMETRIC\_CENTER menunjukkan hasil yang dikembalikan adalah pusat geometris dari hasil seperti polyline (misalnya, jalan) atau poligon (region).
* APPROXIMATE menunjukkan hasil yang dikembalikan adalah perkiraan.
* viewport menyimpan viewport yang disarankan untuk hasil yang dikembalikan.
* bounds (secara opsional dikembalikan) menyimpan LatLngBounds yang bisa sepenuhnya berisi hasil yang dikembalikan. Perhatikan, batas-batas ini mungkin tidak cocok dengan viewport yang disarankan. (Misalnya, San Francisco menyertakan [Farallon Islands](https://en.wikipedia.org/wiki/Farallon_Islands), yang secara teknis merupakan bagian dari kota, namun tidak boleh dikembalikan dalam viewport.)

Tipe Komponen Alamat

Larik types[] dalam hasil yang dikembalikan menunjukkan *tipe alamat*. Tipe ini juga mungkin dikembalikan dalam larik address\_components[] untuk menunjukkan tipe komponen alamat tertentu. Alamat dalam geocoder mungkin memiliki beberapa tipe; tipe tersebut bisa dianggap sebagai "tag". Misalnya, banyak kota diberi tag bertipe political dan locality.

Tipe berikut ini didukung dan dikembalikan oleh HTTP Geocoder:

* street\_address menunjukkan alamat jalan secara tepat.
* route menunjukkan rute yang telah diberi nama (misalnya "US 101").
* intersection menunjukkan persimpangan utama, biasanya persimpangan dua jalan besar.
* political menunjukkan entitas politik. Biasanya, tipe ini menunjukkan poligon dari beberapa pemerintahan sipil.
* country menunjukkan entitas politik nasional, dan biasanya merupakan tipe urutan tertinggi yang dikembalikan oleh Geocoder.
* administrative\_area\_level\_1 menunjukkan entitas sipil urutan pertama di bawah tingkat negara. Di Amerika Serikat, tingkat administratif ini adalah negara bagian. Tidak semua negara memiliki tingkat administratif ini.
* administrative\_area\_level\_2 menunjukkan entitas sipil urutan kedua di bawah tingkat negara. Di Amerika Serikat, tingkat administratif ini adalah county. Tidak semua negara memiliki tingkat administratif ini.
* administrative\_area\_level\_3 menunjukkan entitas sipil urutan ketiga di bawah tingkat negara. Tipe ini menunjukkan divisi sipil kecil. Tidak semua negara memiliki tingkat administratif ini.
* administrative\_area\_level\_4 menunjukkan entitas sipil urutan keempat di bawah tingkat negara. Tipe ini menunjukkan divisi sipil kecil. Tidak semua negara memiliki tingkat administratif ini.
* administrative\_area\_level\_5 menunjukkan entitas sipil urutan kelima di bawah tingkat negara. Tipe ini menunjukkan divisi sipil kecil. Tidak semua negara memiliki tingkat administratif ini.
* colloquial\_area menunjukkan nama alternatif yang umum digunakan untuk entitas.
* locality menunjukkan gabungan entitas politik kota besar atau kota kecil.
* sublocality menunjukkan entitas sipil urutan pertama di bawah daerah. Beberapa lokasi mungkin akan menerima salah satu tipe tambahan: sublocality\_level\_1 hingga sublocality\_level\_5. Setiap tingkat subdaerah adalah entitas sipil. Angka yang lebih besar menunjukkan area geografis yang lebih kecil.
* neighborhood menunjukkan daerah sekitar yang telah diberi nama.
* premise menunjukkan lokasi yang telah diberi nama, biasanya bangunan atau sekumpulan bangunan dengan nama umum
* subpremise menunjukkan entitas urutan pertama di bawah lokasi yang telah diberi nama, biasanya sebuah bangunan dalam sekumpulan bangunan dengan nama umum.
* postal\_code menunjukkan kode pos seperti yang biasa digunakan untuk penulisan alamat pos dalam negara tersebut.
* natural\_feature menunjukkan fitur alami yang menonjol.
* airport menunjukkan sebuah bandara.
* park menunjukkan taman yang telah diberi nama.
* post\_box menunjukkan kotak pos tertentu.
* street\_number menunjukkan nomor rumah secara akurat.
* floor menunjukkan lantai pada alamat bangunan.
* room menunjukkan kamar pada alamat bangunan.

Koding dalam HTML

<!DOCTYPE html>  
<html>  
  <head>  
    <title>Geocoding service</title>  
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no">  
    <meta charset="utf-8">  
    <style>  
      /\* Always set the map height explicitly to define the size of the div  
       \* element that contains the map. \*/  
      #map {  
        height: 100%;  
      }  
      /\* Optional: Makes the sample page fill the window. \*/  
      html, body {  
        height: 100%;  
        margin: 0;  
        padding: 0;  
      }  
      #floating-panel {  
        position: absolute;  
        top: 10px;  
        left: 25%;  
        z-index: 5;  
        background-color: #fff;  
        padding: 5px;  
        border: 1px solid #999;  
        text-align: center;  
        font-family: 'Roboto','sans-serif';  
        line-height: 30px;  
        padding-left: 10px;  
      }  
    </style>  
  </head>  
  <body>  
    <div id="floating-panel">  
      <input id="address" type="textbox" value="Sydney, NSW">  
      <input id="submit" type="button" value="Geocode">  
    </div>  
    <div id="map"></div>  
    <script>  
      function initMap() {  
        var map = new google.maps.Map(document.getElementById('map'), {  
          zoom: 8,  
          center: {lat: -34.397, lng: 150.644}  
        });  
        var geocoder = new google.maps.Geocoder();  
  
        document.getElementById('submit').addEventListener('click', function() {  
          geocodeAddress(geocoder, map);  
        });  
      }  
  
      function geocodeAddress(geocoder, resultsMap) {  
        var address = document.getElementById('address').value;  
        geocoder.geocode({'address': address}, function(results, status) {  
          if (status === 'OK') {  
            resultsMap.setCenter(results[0].geometry.location);  
            var marker = new google.maps.Marker({  
              map: resultsMap,  
              position: results[0].geometry.location  
            });  
          } else {  
            alert('Geocode was not successful for the following reason: ' + status);  
          }  
        });  
      }  
    </script>  
    <script async defer  
    src="https://maps.googleapis.com/maps/api/js?key=***YOUR\_API\_KEY***&callback=initMap">  
    </script>  
  </body>  
</html>

**HTML**

<div id="floating-panel">

<input id="address" type="textbox" value="Sydney, NSW">

<input id="submit" type="button" value="Geocode">

</div>

<div id="map"></div>

<!-- Replace the value of the key parameter with your own API key. -->

<script async defer

src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCkUOdZ5y7hMm0yrcCQoCvLwzdM6M8s5qk&callback=initMap">

</script>

**JAVASCRIPT**

function initMap() {

var map = new google.maps.Map(document.getElementById('map'), {

zoom: 8,

center: {lat: -34.397, lng: 150.644}

});

var geocoder = new google.maps.Geocoder();

document.getElementById('submit').addEventListener('click', function() {

geocodeAddress(geocoder, map);

});

}

function geocodeAddress(geocoder, resultsMap) {

var address = document.getElementById('address').value;

geocoder.geocode({'address': address}, function(results, status) {

if (status === 'OK') {

resultsMap.setCenter(results[0].geometry.location);

var marker = new google.maps.Marker({

map: resultsMap,

position: results[0].geometry.location

});

} else {

alert('Geocode was not successful for the following reason: ' + status);

}

});

}

**CSS**

/\* Always set the map height explicitly to define the size of the div

\* element that contains the map. \*/

#map {

height: 100%;

}

/\* Optional: Makes the sample page fill the window. \*/

html, body {

height: 100%;

margin: 0;

padding: 0;

}

#floating-panel {

position: absolute;

top: 10px;

left: 25%;

z-index: 5;

background-color: #fff;

padding: 5px;

border: 1px solid #999;

text-align: center;

font-family: 'Roboto','sans-serif';

line-height: 30px;

padding-left: 10px;

}