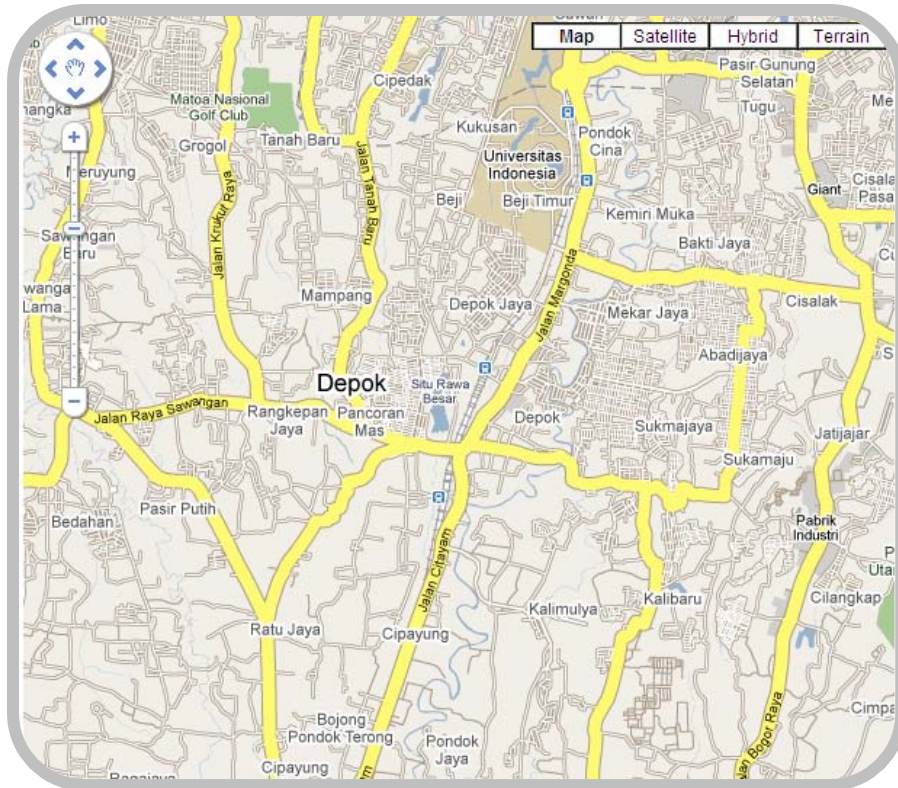

Tutorial Dasar Pemrograman Google Maps API



Amri Shodiq (amri.shodiq@gmail.com)
<http://amrishodiq.blogspot.com>

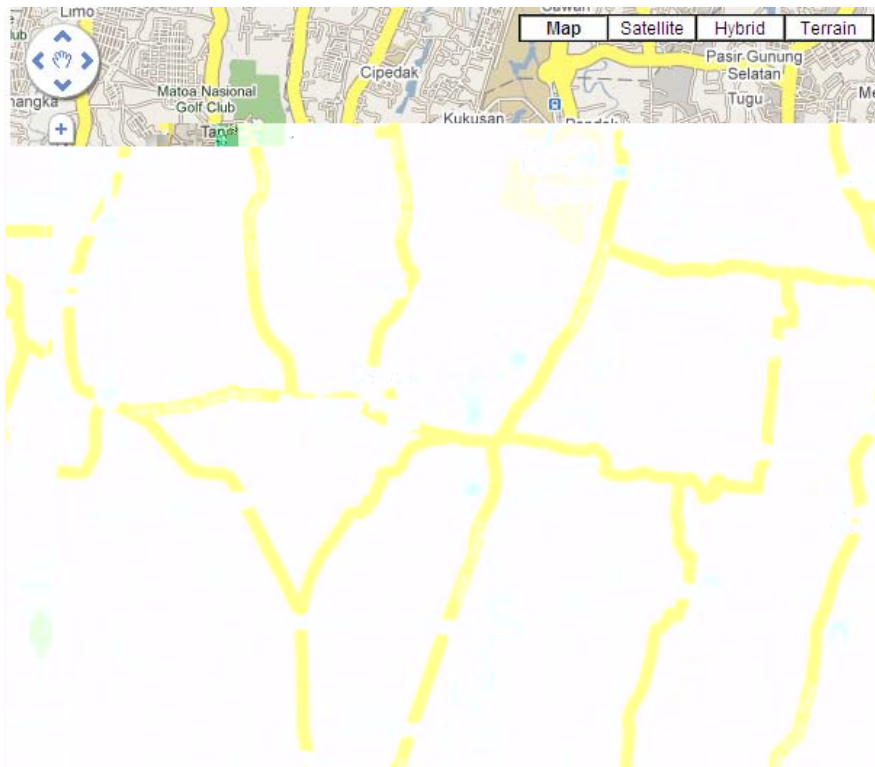
Latar belakang

Google Maps adalah layanan gratis Google yang cukup populer. Anda dapat menambahkan fitur Google Maps dalam web Anda sendiri dengan Google Maps API. Google Maps API adalah library JavaScript. Menggunakan/memprogram Google Maps API sangat mudah. Yang Anda butuhkan adalah pengetahuan tentang HTML dan JavaScript, serta koneksi Internet. Dengan menggunakan Google Maps API Anda dapat menghemat waktu dan biaya Anda untuk membangun aplikasi peta digital yang handal, sehingga Anda dapat focus hanya pada data-data Anda. Biarkan data peta-peta dunia menjadi urusan Google saja.

Saat ini versi terakhir Google Map API adalah versi 3. Versi ini, katanya, akan tampil lebih cepat dari versi sebelumnya khususnya untuk browser ponsel. Ya, ponsel, Anda bisa membangun situs web yang dilengkapi peta untuk iPhone dan ponsel dengan system operasi Android.

Bagaimana memulainya?

Saya ingin menampilkan kota Depok di file HTML saya (di dalam komputer saya saja) pada tag div berukuran 600 pixel x 600 pixel. Bagaimana melakukannya? Saya akan tunjukkan kepada Anda, bagaimana menghasilkan peta seperti ini:



Menggunakan Google Maps API

Kita bisa mulai menulis program Google Map API dengan urutan sebagai berikut:

1. Memasukkan Maps API JavaScript ke dalam HTML kita.
2. Membuat element div dengan nama map_canvas untuk menampilkan peta.
3. Membuat beberapa objek literal untuk menyimpan property-properti pada peta.
4. Menuliskan fungsi JavaScript untuk membuat objek peta.
5. Meng-inisiasi peta dalam tag body HTML dengan event onload.

Beginilah kode program dasarnya:

```
<html>
<head>
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />

<!-- Langkah 1 -->
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=true&key=ABQIAAAA8tt4eKTuB
ZMVnLJfP2BZrBT2yXp_ZAY8_ufC3CFXhHIE1NvwkxS4Rz1LFzG0odNPtk8VLkdrQF5grA"></scri
pt>

<script type="text/javascript">

    // Langkah 4
    function initialize() {
        var latlng = new google.maps.LatLng(-6.4, 106.8186111);
        var myOptions = {
            zoom: 13,
            center: latlng,
            mapTypeId: google.maps.MapTypeId.ROADMAP
        };

        // Langkah 3
        var map = new google.maps.Map(document.getElementById( "map_canvas" ),
myOptions);
    }

</script>
</head>

<!-- Langkah 5 -->
<body onload="initialize()">

    <!-- Langkah 2 -->
    <div id="map_canvas" style="width:600px; height:600px"></div>

</body>
</html>
```

Perhatikan urutan langkah-langkahnya yang saya tuliskan dalam komentar HTML dan JavaScript kode HTML di atas.

Google Maps API key

Perhatikan baris ini:

```
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=true&key=ABQIAAAA8tt4eKTuB
ZMVnLJfP2BZrBT2yXp_ZAY8_ufC3CFXhHIE1NvwkxS4Rz1LFzG0odNPtk8VLkdrQF5grA"></scri
pt>
```

Kode yang saya cetak tebal dan berwarna merah adalah Google Maps API key. [Anda harus mendaftar](#) untuk mendapatkan key untuk website Anda, jika Anda ingin menampilkan peta untuk website Anda. Setelah mendaftar, termasuk memberikan alamat website Anda, Google akan member Anda sebuah API key. Bentuknya kurang lebih seperti ini (tentu saja milik Anda nantinya akan berbeda):

ABQIAAAA8tt4eKTuBZMVnLJfP2BZrBT2yXp_ZAY8_ufC3CFXhHIE1NvwkxS4Rz1LFzG0odNPtk8VLkdrQF5grA

Key yang saya gunakan di atas digunakan untuk computer local (saya daftarkan untuk alamat situ localhost). Jadi, jika Anda ingin mencoba menulis program untuk dijalankan di computer Anda sendiri, sementara Anda bisa pakai key ini. Nanti kalau program Anda sudah siap, baru Anda perlu mendaftar sendiri API key Anda, kemudian dipakai menggantikan API key contoh dari saya.

Parameter sensor berisi true (berwarna hijau) digunakan untuk menggunakan sensor lokasi, misalnya GPS (jika ada, misalnya jika aplikasi web based kita diakses dengan ponsel tertentu yang memiliki GPS).

Elemen peta pada tag div

Pada baris ini:

```
<div id="map_canvas" style="width:600px; height:600px"></div>
```

Kita menentukan lebar tag div adalah 600px, dan lebar juga 600px, dengan style CSS. Ukuran peta yang ditampilkan akan selalu mengikuti ukuran yang Anda tentukan pada tag tersebut.

Setting peta

Kita menentukan setting peta pada baris program ini:

```
var latlng = new google.maps.LatLng(-6.4, 106.8186111);
var myOptions = {
  zoom: 13,
  center: latlng,
  mapTypeId: google.maps.MapTypeId.ROADMAP
};
```

Saya telah memberikan lokasi Latitude dan Longitude pusat kota Depok, yaitu -6,4 dan 106.8186111. Tentu Anda tidak melupakan pelajaran Geografi selama SMP bukan? Ingat Lintang Utara/Selatan dan Bujur Timur? Itulah dia. Untuk kota Depok, posisi dalam peta latitude dan longitude-nya adalah -6,4 dan 106.8186111. Jadi, jika Anda ingin menampilkan kota Anda, misalnya Jakarta Selatan atau Yogyakarta, maka Anda harus cari dulu lokasi kota Anda berdasarkan Latitude dan Longitude. Anda bisa

menggunakan informasi dari situs [ConvertUnits.com](https://www.convertunits.com) untuk mengetahui berapa latitude dan longitude kota Anda.

Parameter *zoom* menentukan zoom level yang Anda inginkan. Semakin kecil nilainya, semakin jauh jarak pandang Anda dari tanah. Nilai 0 akan menunjukkan peta seluruh dunia. Nilai maksimal adalah 19.

Parameter *mapTypeId* menentukan jenis peta yang akan ditampilkan. Pilihannya ada 4:

1. ROADMAP, ini yang saya pilih, untuk menampilkan peta biasa 2 dimensi
2. SATELLITE, untuk menampilkan foto satelit
3. TERRAIN, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai
4. HYBRID, akan menunjukkan foto satelit yang diatasnya tergambar pula apa yang tampil pada ROADMAP (jalan dan nama kota)

Sebenarnya ada 3 kelompok lagi jenis peta yang bisa dipakai, tetapi pembahasannya di luar lingkup ebook ini. Mereka adalah:

- Kelompok Google Map Maker, peta-peta dalam kategori ini merupakan kontribusi dari seluruh dunia
- Kelompok peta Celestial, selain peta bumi, Google juga menyediakan peta bulan, dan Mars
- Kelompok peta 3D, membutuhkan plugins Google Earth

Menampilkan peta pada tag div

Di atas, kita telah membuat fungsi `initialize()`. Fungsi ini menampilkan peta Google Maps pada tag `div` dengan id `map_canvas`. Masalahnya adalah, peta tidak akan muncul jika fungsi ini tidak dipanggil. Maka, fungsi ini akan dipanggil pada event `onload`, ketika semua object sudah siap, termasuk JavaScript Google Maps API sudah terdownload sepenuhnya oleh browser. Pemanggilan tersebut terjadi pada baris ini:

```
<body onload="initialize()">
```

Interaksi dengan peta

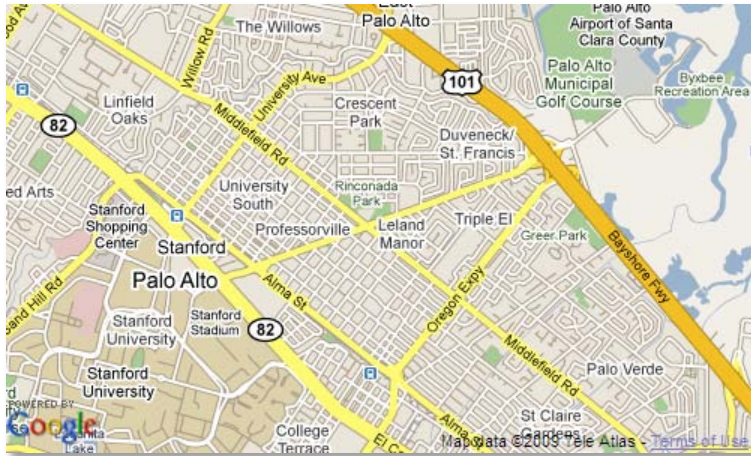
Interaksi dengan peta Google Maps API bisa dilakukan dengan beberapa cara, diantaranya dengan method-method seperti: `setCenter()`, `panTo()` dan `zoomIn()`.

Selain method, Google Maps API juga mendukung event listener. Penjelasan tentang method dan event tidak akan diberikan dalam ebook ini, tetapi Anda langsung dapat melihat bagaimana cara kerja dan fungsinya melalui code snippet-code snippet yang ada di bagian berikutnya.

Code Snippet

Menampilkan peta sederhana

Berikut ini diberikan contoh, bagaimana menampilkan peta sederhana kota Palo Alto di mana Google berada. Peta ini ditampilkan tanpa Control dan Marker. Hanya peta.



```
function initialize() {  
  if (GBrowserIsCompatible()) {  
    var map = new GMap2(document.getElementById("map_canvas"));  
    map.setCenter(new GLatLng(37.4419, -122.1419), 13);  
  }  
}
```

Menambahkan 10 Marker secara random di kota Depok

Berikut saya coba berikan contoh bagaimana menampilkan kota Depok dan menambahkan 10 Marker yang lokasinya acak. Ini ditujukan agar Anda mendapat gambaran bagaimana menambahkan Marker standar di atas peta.



```
function initialize() {
  if (GBrowserIsCompatible()) {
    var map = new GMap2(document.getElementById("map_canvas"));
    map.setCenter(new GLatLng(-6.4, 106.8186111), 13);

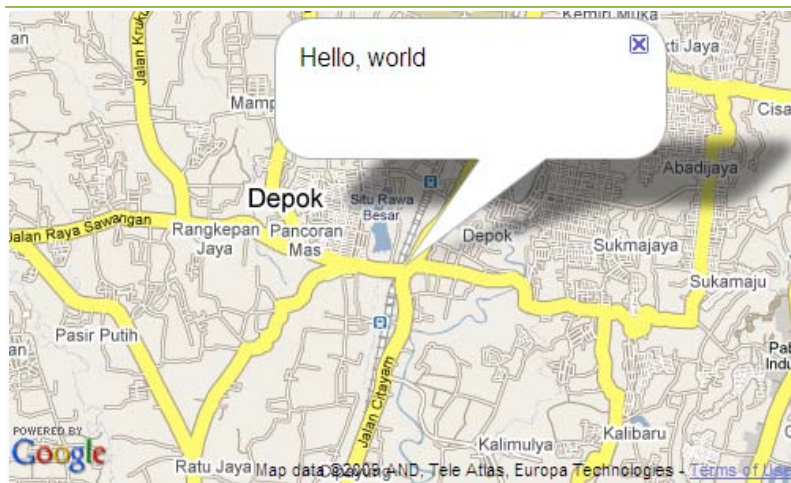
    // Add 10 markers to the map at random locations
    var bounds = map.getBounds();
    var southWest = bounds.getSouthWest();
    var northEast = bounds.getNorthEast();
    var lngSpan = northEast.lng() - southWest.lng();
    var latSpan = northEast.lat() - southWest.lat();
    for (var i = 0; i < 10; i++) {
      var point = new GLatLng(southWest.lat() + latSpan * Math.random(),
                              southWest.lng() + lngSpan * Math.random());
      map.addOverlay(new GMarker(point));
    }
  }
}
```

Membuat peta yang berpindah dari Depok ke Palo Alto setelah 1 detik

Berikut ini contoh membuat peta yang dapat beranimasi, berpindah dari Depok kemudian pindah ke Palo Alto. Ini bukanlah contoh yang baik. Untuk implementasi yang baik, sebaiknya pindah ke lokasi yang dekat saja dan bertahap. Screenshot tidak diberikan karena tidak menggambarkan animasi.

```
function initialize() {
  if (GBrowserIsCompatible()) {
    var map = new GMap2(document.getElementById("map_canvas"));
    map.setCenter(new GLatLng(-6.4, 106.8186111), 13);
    window.setTimeout(function() {
      map.panTo(new GLatLng(37.4569, -122.1569));
    }, 1000);
  }
}
```

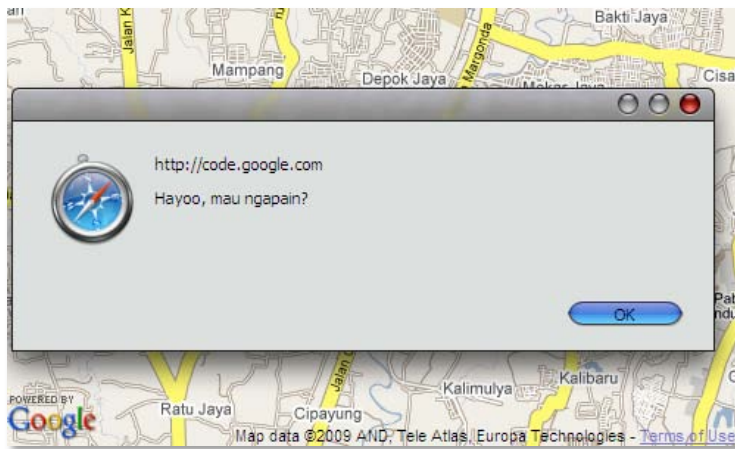
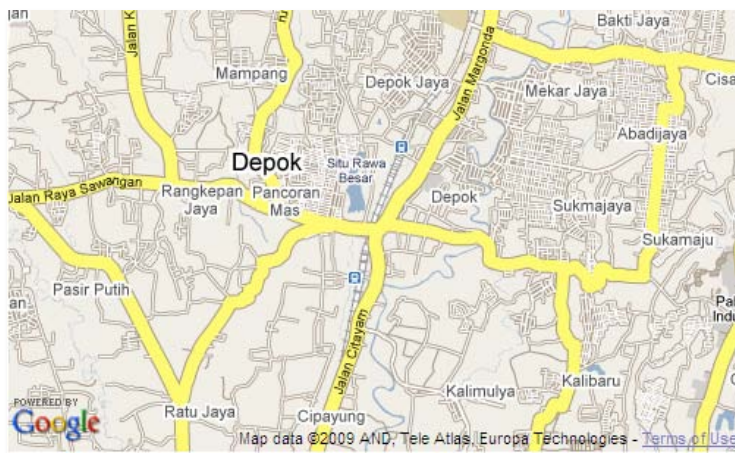
Menampilkan window info di atas peta



```
function initialize() {
  if (GBrowserIsCompatible()) {
    var map = new GMap2(document.getElementById("map_canvas"));
    map.setCenter(new GLatLng(-6.4, 106.8186111), 13);
    map.openInfoWindow(map.getCenter(),
      document.createTextNode("Hello, world"));
  }
}
```

Event: Menampilkan alert ketika user mengklik peta

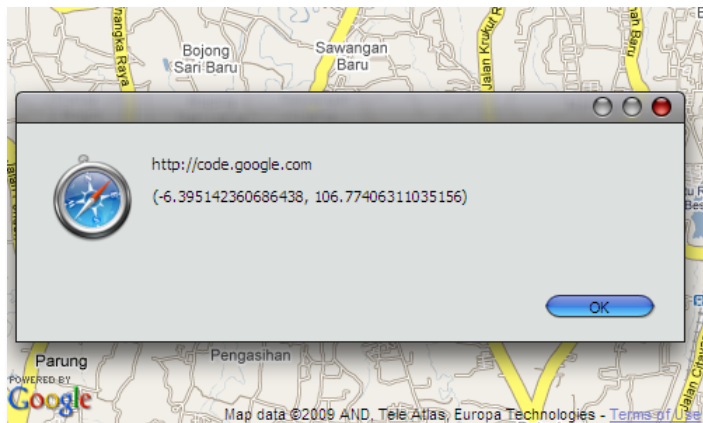
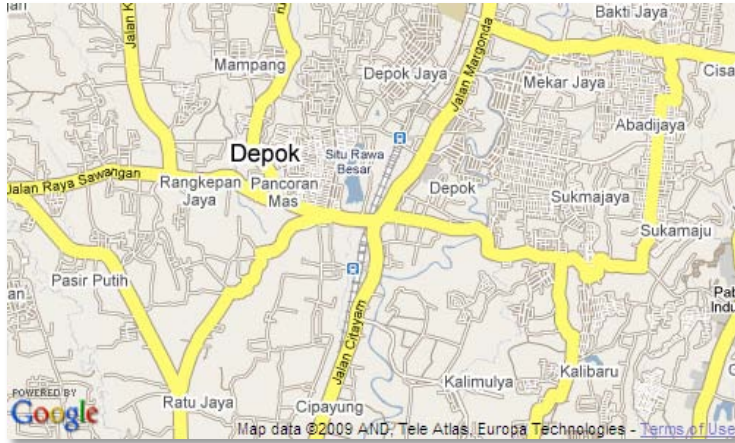
Contoh program berikut ini menunjukkan bagaimana menangkap event click kemudian melakukan proses sederhana, yaitu menampilkan alert.



```
function initialize() {
  if (GBrowserIsCompatible()) {
    var map = new GMap2(document.getElementById("map_canvas"));
    map.setCenter(new GLatLng(-6.4, 106.8186111), 13);
    GEvent.addListener(map, "click", function() {
      alert("You clicked the map.");
    });
  }
}
```


Event: Menampilkan alert posisi tengah peta yang baru setelah drag selesai

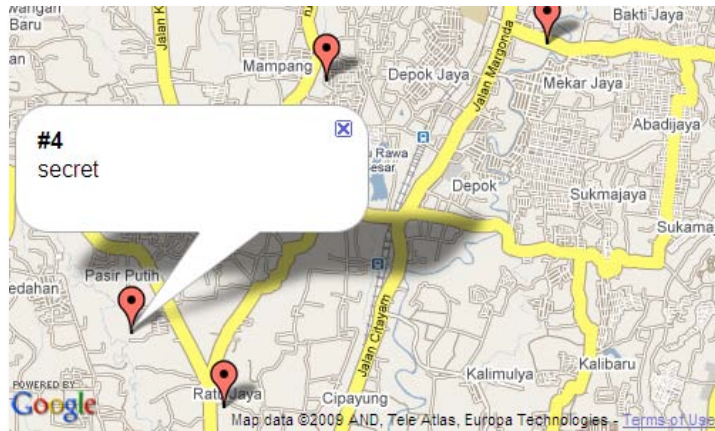
Contoh program berikut menunjukkan penggunaan event *moveend*. Ketika akhir *drag*, baru aksi tertentu dijalankan.



```
function initialize() {  
  if (GBrowserIsCompatible()) {  
    var map = new GMap2(document.getElementById("map_canvas"));  
    map.setCenter(new GLatLng(-6.4, 106.81861119), 13);  
    GEvent.addListener(map, "moveend", function() {  
      var center = map.getCenter();  
      alert(center.toString());  
    });  
  }  
}
```

Event: Menambahkan 5 titik di peta yang masing-masing menampilkan potongan pesan rahasia jika diklik

Kode program berikut berfungsi menyimpan 5 potong pesan yang kemudian dipasang pada window Info 5 buah Marker yang diletakkan secara acak.



```
function initialize() {
    var message = ["This", "is", "the", "secret", "message"];

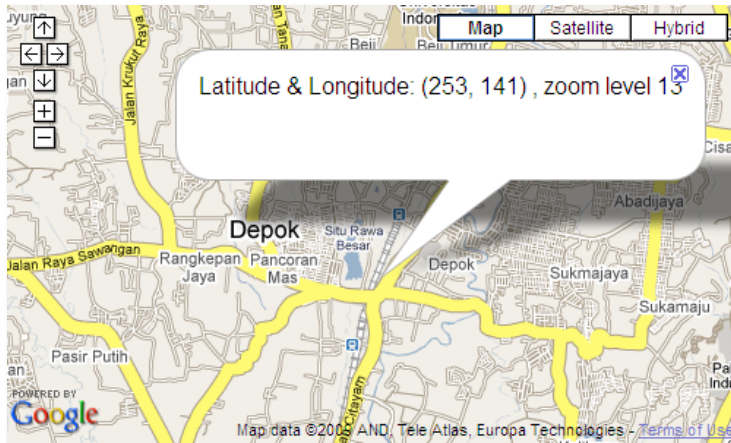
    if (GBrowserIsCompatible()) {
        var map = new GMap2(document.getElementById("map_canvas"));
        map.setCenter(new GLatLng(-6.4, 106.8186111), 13);

        // Creates a marker at the given point
        // Clicking the marker will hide it
        function createMarker(latlng, number) {
            var marker = new GMarker(latlng);
            marker.value = number;
            GEvent.addListener(marker, "click", function() {
                var myHtml = "<b>#</b>" + number + "</b><br/>" + message[number - 1];
                map.openInfoWindowHtml(latlng, myHtml);
            });
            return marker;
        }

        // Add 5 markers to the map at random locations
        // Note that we don't add the secret message to the marker's instance
        data
        var bounds = map.getBounds();
        var southWest = bounds.getSouthWest();
        var northEast = bounds.getNorthEast();
        var lngSpan = northEast.lng() - southWest.lng();
        var latSpan = northEast.lat() - southWest.lat();
        for (var i = 0; i < 5; i++) {
            var latlng = new GLatLng(southWest.lat() + latSpan * Math.random(),
                southWest.lng() + lngSpan * Math.random());
            map.addOverlay(createMarker(latlng, i + 1));
        }
    }
}
```

Event: Menampilkan window info berisi latitude, longitude dan zoom level ketika peta diklik

Dalam program berikut, kita menangkap *event* klik pada peta, kemudian menampilkan window Info berisi latitude, longitude dan zoom level pada lokasi tempat pengguna mengklik.



```
function initialize() {
    if (GBrowserIsCompatible()) {
        var map = new GMap2(document.getElementById("map_canvas"));
        map.setCenter(new GLatLng(-6.4, 106.81861119), 13);

        GEvent.addListener(map, "click", function(overlay, latlng) {
            if (latlng) {
                var myHtml = "Latitude & Longitude: "
                + map.fromLatLngToDivPixel(latlng) + " , zoom level " + map.getZoom();
                map.openInfoWindow(latlng, myHtml);
            }
        });
        map.addControl(new GSmallMapControl());
        map.addControl(new GMapTypeControl());
    }
}
```

Event: Menampilkan alert jumlah klik yang sudah dilakukan kepada peta

Aplikasi Google Maps API juga bisa menyimpan counter, misalnya dalam program ini ditunjukkan untuk menyimpan counter klik.



```
function MyApplication() {
    this.counter = 0;
    this.map = new GMap2(document.getElementById("map_canvas"));
    this.map.setCenter(new GLatLng(-6.4, 106.8186111), 13);
    GEvent.bind(this.map, "click", this, this.onMapClick);
}

MyApplication.prototype.onMapClick = function() {
    this.counter++;
    alert("Anda sudah mengklik " + this.counter +
        "kali.");
}

function initialize() {
    if (GBrowserIsCompatible()) {
        var application = new MyApplication();
    }
}
```

Event: Menghapus event listener

Program berikut menunjukkan bagaimana cara menghilangkan *event listener* setelah klik 1 kali. Sebelumnya, ketika event klik program ini menghapus Marker.

```
function MyApplication() {
    this.counter = 0;
    this.map = new GMap2(document.getElementById("map_canvas"));
    this.map.setCenter(new GLatLng(-6.4, 106.8186111), 13);
    var myEventListener = GEvent.bind(this.map, "click", this, function(overlay,
latlng) {
        if (this.counter == 0) {
            if (latlng) {
                this.map.addOverlay(new GMarker(latlng))
                this.counter++;
            } else if (overlay instanceof GMarker) {
                // This code is never executed as the event listener is
                // removed the second time this event is triggered
                this.removeOverlay(marker)
            }
        }
    })
}
```

```

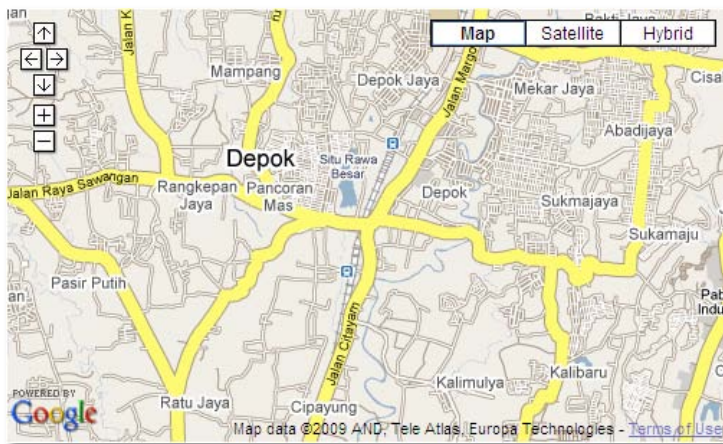
    } else {
        GEvent.removeListener(myEventListener);
    }
}
});

function initialize() {
    var application = new MyApplication();
}

```

Control: Menampilkan Control sederhana

GSmallMapControl menampilkan 4 tombol panah dan tombol zoom in dan zoom out saja dalam bentuk yang kecil. GMapTypeControl adalah Control untuk menampilkan, defaultnya, 3 tipe peta: peta biasa, foto satelit dan gabungan antara foto satelit yang ditimpa dengan peta.



```

function initialize() {
    if (GBrowserIsCompatible()) {
        var map = new GMap2(document.getElementById("map_canvas"));
        map.setCenter(new GLatLng(-6.4, 106.8186111), 13);
        map.addControl(new GSmallMapControl());
        map.addControl(new GMapTypeControl());
    }
}

```

Control: Menampilkan posisi control di bagian kanan bawah peta setelah peta di-double click

Kode program berikut member contoh bagaimana kita bisa mengatur letak Control. Selain itu juga memperkenalkan event *double click*.



```
function initialize() {
    if (GBrowserIsCompatible()) {
        var map = new GMap2(document.getElementById("map_canvas"));
        map.setCenter(new GLatLng(-6.4, 106.8186111), 13);
        var mapTypeControl = new GMapTypeControl();

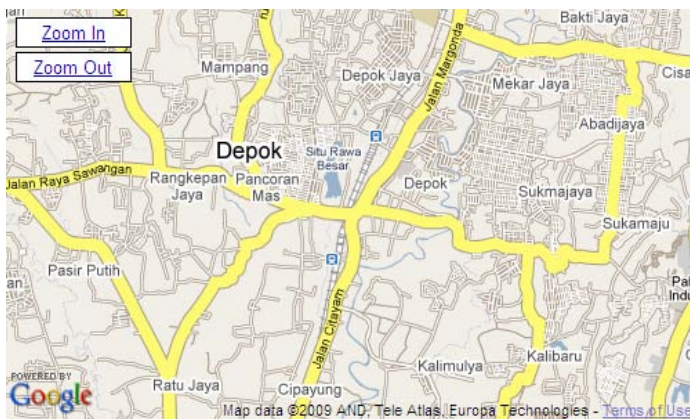
        var topRight = new GControlPosition(G_ANCHOR_TOP_RIGHT, new GSize(10,10));

        var bottomRight = new GControlPosition(G_ANCHOR_BOTTOM_RIGHT, new GSize(10,10));

        map.addControl(mapTypeControl, topRight);
        GEvent.addListener(map, "dblclick", function() {
            map.removeControl(mapTypeControl);
            map.addControl(new GMapTypeControl(), bottomRight);
        });
        map.addControl(new GSmallMapControl());
    }
}
```

Control: Menampilkan bentuk Control yang tidak biasa

Kita bisa menggunakan bentuk Control yang tidak standar, tetapi dengan kemampuan yang sama. Contohnya adalah dengan menampilkan dua tombol, untuk zoom in dan zoom out.



```

// A TextualZoomControl is a GControl that displays textual "Zoom In"
// and "Zoom Out" buttons (as opposed to the iconic buttons used in
// Google Maps).
function TextualZoomControl() {
}
TextualZoomControl.prototype = new GControl();

// Creates a one DIV for each of the buttons and places them in a container
// DIV which is returned as our control element. We add the control to
// the map container and return the element for the map class to
// position properly.
TextualZoomControl.prototype.initialize = function(map) {
    var container = document.createElement("div");

    var zoomInDiv = document.createElement("div");
    this.setButtonStyle_(zoomInDiv);
    container.appendChild(zoomInDiv);
    zoomInDiv.appendChild(document.createTextNode("Zoom In"));
    GEvent.addDomListener(zoomInDiv, "click", function() {
        map.zoomIn();
    });

    var zoomOutDiv = document.createElement("div");
    this.setButtonStyle_(zoomOutDiv);
    container.appendChild(zoomOutDiv);
    zoomOutDiv.appendChild(document.createTextNode("Zoom Out"));
    GEvent.addDomListener(zoomOutDiv, "click", function() {
        map.zoomOut();
    });

    map.getContainer().appendChild(container);
    return container;
}

// By default, the control will appear in the top left corner of the
// map with 7 pixels of padding.
TextualZoomControl.prototype.getDefaultPosition = function() {
    return new GControlPosition(G_ANCHOR_TOP_LEFT, new GSize(7, 7));
}

// Sets the proper CSS for the given button element.
TextualZoomControl.prototype.setButtonStyle_ = function(button) {
    button.style.textDecoration = "underline";
    button.style.color = "#0000cc";
    button.style.backgroundColor = "white";
    button.style.font = "small Arial";
    button.style.border = "1px solid black";
    button.style.padding = "2px";
    button.style.marginBottom = "3px";
    button.style.textAlign = "center";
    button.style.width = "6em";
    button.style.cursor = "pointer";
}

function initialize() {
    if (GBrowserIsCompatible()) {
        var map = new GMap2(document.getElementById("map_canvas"));
    }
}

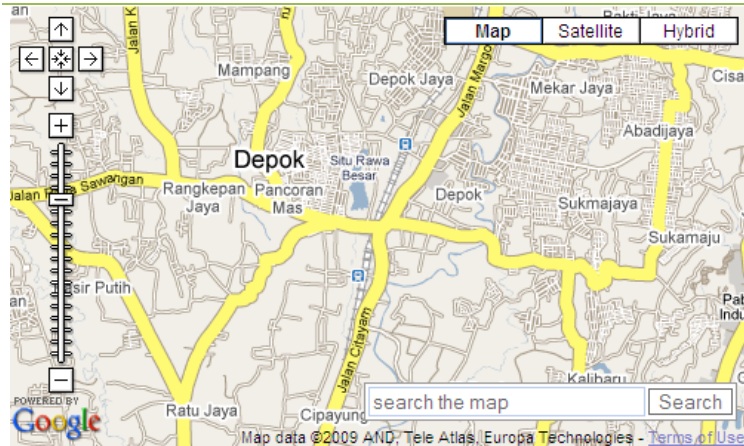
```

```

    map.setCenter(new GLatLng(-6.4, 106.8186111), 13);
    map.addControl(new TextualZoomControl());
}
}

```

Control: Menampilkan Local Search



```

function initialize() {
    if (GBrowserIsCompatible()) {

        // Create and Center a Map
        var map = new GMap2(document.getElementById("map_canvas"));
        map.setCenter(new GLatLng(-6.4, 106.81861119), 13);
        map.addControl(new GLargeMapControl());
        map.addControl(new GMapTypeControl());

        // bind a search control to the map, suppress result list

        map.addControl(new google.maps.LocalSearch(), new GControlPosition(G_ANCHOR_
        BOTTOM_RIGHT, new GSize(10,20)));
    }
}
GSearch.setOnLoadCallback(initialize);

```

Control: Menampilkan Control tipe peta

Kode program berikut menunjukkan kepada Anda bagaimana menampilkan control Map Type tanpa tipe Hybrid.



```
function initialize() {
    if (GBrowserIsCompatible()) {

        var map = new google.maps.Map(document.getElementById( "map_canvas" ),
            { size: new GSize(640,320) });

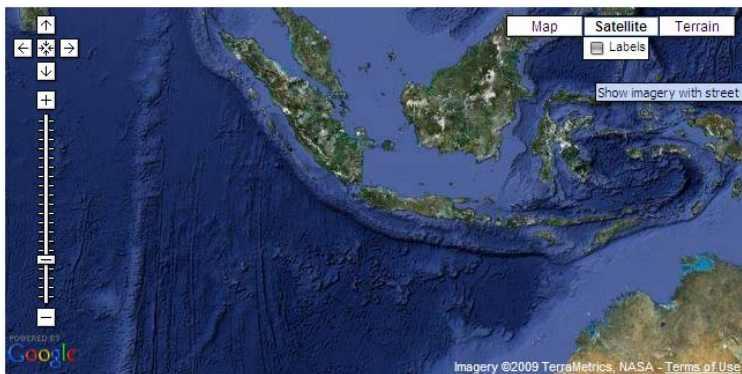
        map.removeMapType(G_HYBRID_MAP);
        map.setCenter(new GLatLng(-6.4, 106.8186111), 11);
        map.addControl(new GLargeMapControl());

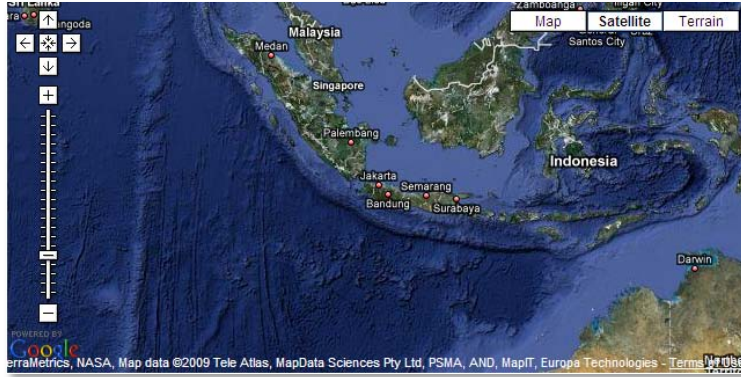
        var mapControl = new GMapTypeControl();
        map.addControl(mapControl);

    }
}
```

Control: Menampilkan label dengan check box untuk menampilkan atau menyembunyikan nama kota,dan jalan

Dalam aplikasi, kadang kita ingin menunjukkan atau menyembunyikan nama-nama kota atau jalan. Ini bisa kita lakukan dalam contoh program berikut. Dalam contoh berikut, Anda dapat melihat bagaimana penampilan nama jalan dan kota dapat dikendalikan dengan sebuah check box (dan label) yang menempel pada control Map Type.





```
function initialize() {
  if (GBrowserIsCompatible()) {
    // define the crosshair tile layer and its required functions
    var crossLayer = new GTileLayer(new GCopyrightCollection(""), 0, 15);
    crossLayer.getTileUrl = function(tile, zoom) {
      return "../include/tile_crosshairs.png";
    }
    crossLayer.isPng = function() {return true;}

    // Create a new map type incorporating the tile layer
    var layerTerCross = [ G_PHYSICAL_MAP.getTileLayers()[0],
                        crossLayer ];
    var mtTerCross = new GMapType(layerTerCross,
                                  G_PHYSICAL_MAP.getProjection(), "Ter+");

    var map = new GMap2(document.getElementById("map_canvas"),
                        { size: new GSize(640,320) } );
    map.addMapType(G_PHYSICAL_MAP);
    map.addMapType(mtTerCross);
    map.setCenter(new GLatLng(-6.4, 106.8186111), 4);
    map.addControl(new GLargeMapControl());

    var mapControl = new GHierarchicalMapTypeControl();

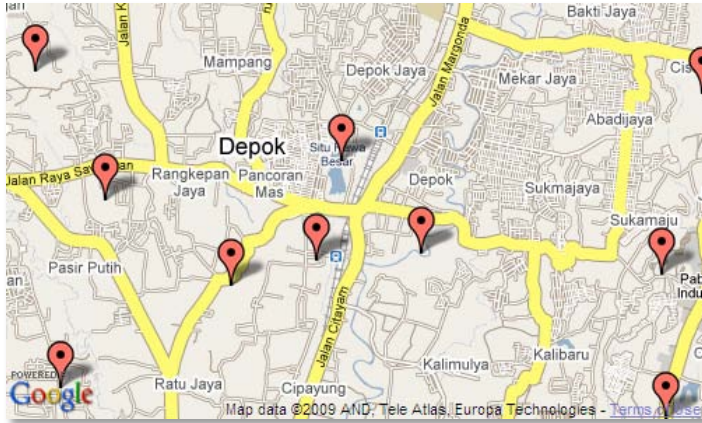
    // Set up map type menu relationships
    mapControl.clearRelationships();

    mapControl.addRelationship(G_SATELLITE_MAP, G_HYBRID_MAP, "Labels", false);
    mapControl.addRelationship(G_PHYSICAL_MAP, mtTerCross, "Crosshairs");

    // Add control after you've specified the relationships
    map.addControl(mapControl);
  }
}
```

Marker: Menampilkan Marker sederhana

Sedikit mengulang yang sudah-sudah. Dalam contoh berikut, Anda akan melihat bagaimana menambahkan beberapa Marker sederhana di atas kota Depo.



```
function initialize() {
  if (GBrowserIsCompatible()) {
    var map = new GMap2(document.getElementById("map_canvas"));
    map.setCenter(new GLatLng(-6.4, 106.8186111), 13);

    // Add 10 markers to the map at random locations
    var bounds = map.getBounds();
    var southWest = bounds.getSouthWest();
    var northEast = bounds.getNorthEast();
    var lngSpan = northEast.lng() - southWest.lng();
    var latSpan = northEast.lat() - southWest.lat();
    for (var i = 0; i < 10; i++) {
      var latlng = new GLatLng(southWest.lat() + latSpan * Math.random(),
                              southWest.lng() + lngSpan * Math.random());
      map.addOverlay(new GMarker(latlng));
    }
  }
}
```

Marker: Menampilkan Marker yang bisa dipindah

Saya tidak bisa menunjukkan screenshot untuk kode berikut karena tidak akan banyak menggambarkan kondisinya. Intinya, dengan kode program berikut Anda bisa membuat Marker yang dapat dipindahkan. Caranya adalah dengan mendefinisikan sebuah Marker dengan option *draggable* bernilai *true*, kemudian menggunakan event listener *dragstart* dan event *dragend*.

```
function initialize() {
  if (GBrowserIsCompatible()) {
    var map = new GMap2(document.getElementById("map_canvas"));
    var center = new GLatLng(-6.4, 106.8186111);
    map.setCenter(center, 13);

    var marker = new GMarker(center, {draggable: true});

    GEvent.addListener(marker, "dragstart", function() {
      map.closeInfoWindow();
    });

    GEvent.addListener(marker, "dragend", function() {
```

```

        marker.openInfoWindowHtml("Just bouncing along...");
    });

    map.addOverlay(marker);
}
}

```

Marker: Menampilkan Marker dalam bentuk lain, misalnya gambar dari URL tertentu

Jika bosan dengan Marker yang begitu-begitu saja, maka dalam contoh berikut, Anda bisa saja menampilkan Marker dalam bentuk lain, yaitu gambar/ikon dari sebuah URL.

Baris ini:

```

var blueIcon = new GIcon(G_DEFAULT_ICON);
blueIcon.image = "http://maps.google.com/mapfiles/ms/micons/cabs.png";

```

membuat sebuah instance class GIcon, kemudian mengubah properties image menjadi gambar yang sudah kita sediakan, misalnya di sini gambar mobil.

Langkah berikutnya adalah mengatur agar markerOptions menggunakan icon yang sudah kita siapkan.

```

// Set up our GMarkerOptions object
markerOptions = { icon:blueIcon };

```

Langkah terakhir adalah menampilkan Marker dengan opsi di atas:

```

map.addOverlay(new GMarker(latlng, markerOptions));

```



```

function initialize() {
    if (GBrowserIsCompatible()) {

```

```

var map = new GMap2(document.getElementById("map_canvas"));
map.setCenter(new GLatLng(-6.4, 106.8186111), 13);
map.addControl(new GSmallMapControl());
map.addControl(new GMapTypeControl());

// Create our "tiny" marker icon
var blueIcon = new GIcon(G_DEFAULT_ICON);
blueIcon.image = "http://maps.google.com/mapfiles/ms/micons/cabs.png";

// Set up our GMarkerOptions object
markerOptions = { icon:blueIcon };

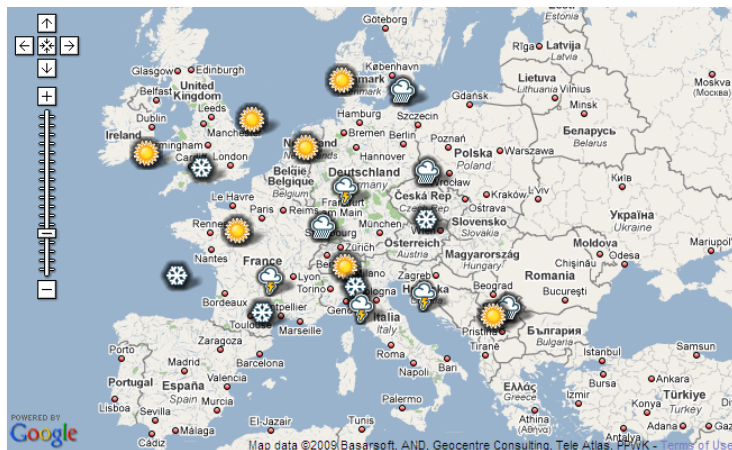
// Add 10 markers to the map at random locations
var bounds = map.getBounds();
var southWest = bounds.getSouthWest();
var northEast = bounds.getNorthEast();
var lngSpan = northEast.lng() - southWest.lng();
var latSpan = northEast.lat() - southWest.lat();
for (var i = 0; i < 10; i++) {
    var latlng = new GLatLng(southWest.lat() + latSpan * Math.random(),
                             southWest.lng() + lngSpan * Math.random());
    map.addOverlay(new GMarker(latlng, markerOptions));
}
}
}

```

Icon marker bisa didapat dari http://abeliantechologies.com/google_map_marker.php atau Anda buat sendiri.

Marker: Menampilkan contoh perkiraan cuaca

Fitur ini berjalan lancar ketika peta saya arahkan ke Eropa. Tetapi saya belum berhasil menggunakan fitur ini untuk Indonesia. Saya terus terang belum tahu sebabnya.



```

var IMAGES = [ "sun", "rain", "snow", "storm" ];
var ICONS = [ ];
var map = null;
var mgr = null;

```

```

function setupMap() {
    if (GBrowserIsCompatible()) {
        map = new GMap2(document.getElementById("map"));
        map.addControl(new GLargeMapControl());
        map.setCenter(new GLatLng(48.25, 11.00), 4);
        map.enableDoubleClickZoom();
        window.setTimeout(setupWeatherMarkers, 0);
    }
}

function getWeatherIcon() {
    var i = Math.floor(IMAGES.length*Math.random());
    if (!ICONS[i]) {
        var icon = new GIcon();
        icon.image = "http://gmaps-utility-
library.googlecode.com/svn/trunk/markermanager/release/examples/images/"
+ IMAGES[i] + ".png";
        icon.iconAnchor = new GPoint(16, 16);
        icon.infoWindowAnchor = new GPoint(16, 0);
        icon.iconSize = new GSize(32, 32);
        icon.shadow = "http://gmaps-utility-
library.googlecode.com/svn/trunk/markermanager/release/examples/images/"
+ IMAGES[i] + "-shadow.png";
        icon.shadowSize = new GSize(59, 32);
        ICONS[i] = icon;
    }
    return ICONS[i];
}

function getRandomPoint() {
    var lat = 48.25 + (Math.random() - 0.5)*14.5;
    var lng = 11.00 + (Math.random() - 0.5)*36.0;
    return new GLatLng(Math.round(lat*10)/10, Math.round(lng*10)/10);
}

function getWeatherMarkers(n) {
    var batch = [];
    for (var i = 0; i < n; ++i) {
        batch.push(new GMarker(getRandomPoint(), { icon: getWeatherIcon() }));
    }
    return batch;
}

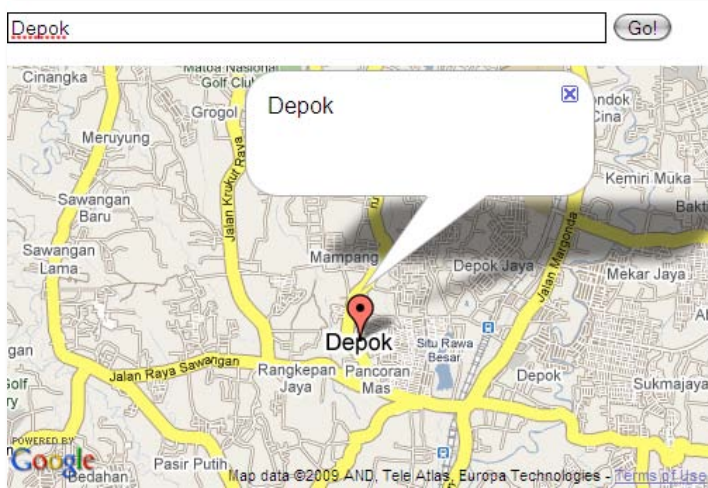
function setupWeatherMarkers() {
    mgr = new MarkerManager(map);
    mgr.addMarkers(getWeatherMarkers(20), 3);
    mgr.addMarkers(getWeatherMarkers(200), 6);
    mgr.addMarkers(getWeatherMarkers(1000), 8);
    mgr.refresh();
}

```

Service: Contoh Service sederhana mencari lokasi berdasarkan input

Google Maps sudah memiliki fitur yang cukup lengkap, termasuk mencari lokasi kota pada peta. Dalam contoh berikut, kita menggunakan class `GClientGeocoder` untuk mendapatkan lokasi alamat yang kita berikan. Class ini akan menanyakan langsung kepada server Google tentang lokasi tersebut.

`GClientGeocoder` termasuk dalam [Google Maps API versi 2](#). Class ini memiliki method-method yang canggih seperti `getLatLng()` untuk mendapatkan Latitude dan Longitude dari input berupa String alamat. Method inilah yang kita gunakan dalam contoh program di bawah ini.



```
var map = null;
var geocoder = null;

function initialize() {
    if (GBrowserIsCompatible()) {
        map = new GMap2(document.getElementById("map_canvas"));
        map.setCenter(new GLatLng(37.4419, -122.1419), 13);
        geocoder = new GClientGeocoder();
    }
}

function showAddress(address) {
    if (geocoder) {
        geocoder.getLatLng(
            address,
            function(point) {
                if (!point) {
                    alert(address + " not found");
                } else {
                    map.setCenter(point, 13);
                    var marker = new GMarker(point);
                    map.addOverlay(marker);
                    marker.openInfoWindowHtml(address);
                }
            }
        );
    }
};
```



```
}  
}
```

Service: Mencari alamat tertentu dengan Geolocation



```
var map;  
var geocoder;  
  
function initialize() {  
    map = new GMap2(document.getElementById("map_canvas"));  
    map.setCenter(new GLatLng(34, 0), 1);  
    geocoder = new GClientGeocoder();  
}  
  
// addAddressToMap() is called when the geocoder returns an  
// answer. It adds a marker to the map with an open info window  
// showing the nicely formatted version of the address and the country code.  
function addAddressToMap(response) {  
    map.clearOverlays();  
    if (!response || response.Status.code != 200) {  
        alert("Sorry, we were unable to geocode that address");  
    } else {  
        place = response.Placemark[0];  
        point = new GLatLng(place.Point.coordinates[1],  
                             place.Point.coordinates[0]);  
        marker = new GMarker(point);  
        map.addOverlay(marker);  
        marker.openInfoWindowHtml(place.address + '<br>' +  
                                  '<b>Country code:</b>' + place.AddressDetails.Country.CountryNameCode);  
    }  
}  
  
// showLocation() is called when you click on the Search button  
// in the form. It geocodes the address entered into the form  
// and adds a marker to the map at that location.  
function showLocation() {  
    var address = document.forms[0].q.value;
```

```

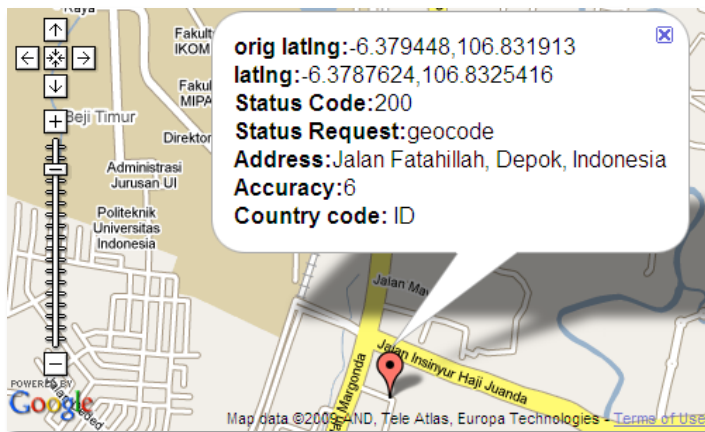
geocoder.getLocations(address, addAddressToMap);
}

// findLocation() is used to enter the sample addresses into the form.
function findLocation(address) {
    document.forms[0].q.value = address;
    showLocation();
}

```

Service: Mendapatkan alamat dari lokasi yang di-klik di peta

Anda penasaran dengan alamat suatu tempat yang Anda temukan? Coba cara ini. Kita bisa mendapatkan alamat dari suatu titik di peta. Meskipun tidak begitu lengkap. Coba test kode program ini.



```

var map;
var geocoder;
var address;

function initialize() {
    map = new GMap2(document.getElementById("map_canvas"));
    map.setCenter(new GLatLng(-6.4, 106.8186111), 15);
    map.addControl(new GLargeMapControl);
    GEvent.addListener(map, "click", getAddress);
    geocoder = new GClientGeocoder();
}

function getAddress(overlay, latlng) {
    if (latlng != null) {
        address = latlng;
        geocoder.getLocations(latlng, showAddress);
    }
}

function showAddress(response) {
    map.clearOverlays();
    if (!response || response.Status.code != 200) {
        alert("Status Code:" + response.Status.code);
    } else {
        place = response.Placemark[0];
    }
}

```

```

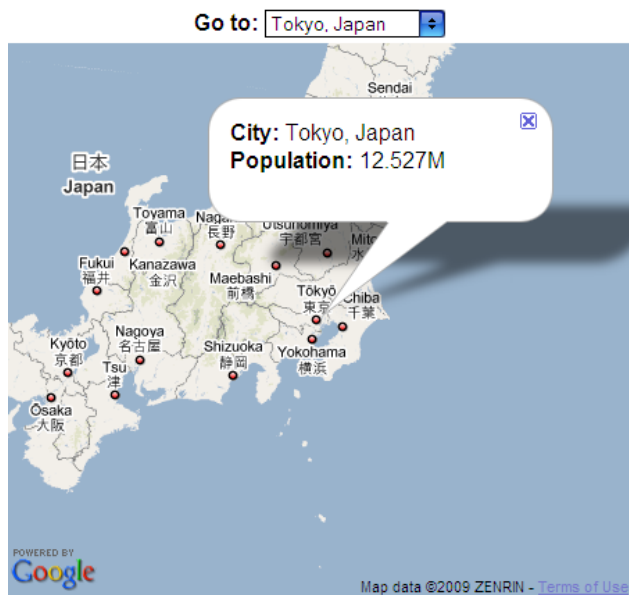
        point = new GLatLng(place.Point.coordinates[1],
                             place.Point.coordinates[0]);
        marker = new GMarker(point);
        map.addOverlay(marker);
        marker.openInfoWindowHtml(
            '<b>orig latlng:</b>' + response.name + '<br/>' +

            '<b>latlng:</b>' + place.Point.coordinates[1] + "," + place.Point.coordinate
s[0] + '<br>' +
            '<b>Status Code:</b>' + response.Status.code + '<br>' +
            '<b>Status Request:</b>' + response.Status.request + '<br>' +
            '<b>Address:</b>' + place.address + '<br>' +
            '<b>Accuracy:</b>' + place.AddressDetails.Accuracy + '<br>' +
            '<b>Country code:</b>' + place.AddressDetails.Country.CountryNameCode);
    }
}

```

Service: Menampilkan lokasi-lokasi yang telah disimpan

Dalam contoh berikut, kita akan coba menyimpan Latitude, Longitude dan data-data lainnya tentang kota-kota besar di dunia ke dalam array. Kemudian memerintahkan peta untuk menuju lokasi kota yang dipilih oleh pengguna menggunakan tag Select.



```
// Builds an array of geocode responses for the 5 cities.
```

```

var city = [
    {
        name: "Washington, DC",
        Status: {
            code: 200,
            request: "geocode"
        },
        Placemark: [
            {
                address: "Washington, DC, USA",

```

```

    population: "0.563M",
    Point: {
      coordinates: [-77.036667, 38.895000, 0]
    },
    AddressDetails: {
      Country: {
        CountryNameCode: "US",
        AdministrativeArea: {
          AdministrativeAreaName: "DC",
          Locality: {LocalityName: "Washington"}
        }
      }
    }
  }
},
{
  name: "Tokyo, Japan",
  Status: {
    code: 200,
    request: "geocode"
  },
  Placemark: [
    {
      address: "Tokyo, Japan",
      population: "12.527M",
      Point: {
        coordinates: [139.770004, 35.669998, 0]
      },
      AddressDetails: {
        Country: {
          CountryNameCode: "US",
          AdministrativeArea: {
            AdministrativeAreaName: "CA",
            Locality: {LocalityName: "Los Angeles"}
          }
        }
      }
    }
  ]
},
{
  name: "Paris, France",
  Status: {
    code: 200,
    request: "geocode"
  },
  Placemark: [
    {
      address: "Paris, France",
      population: "2.144M",
      Point: {
        coordinates: [2.351019, 48.856662, 0]
      },
      AddressDetails: {
        Country: {
          CountryNameCode: "FR",

```

```

        Locality: {LocalityName: "Paris"}
    }
}
]
},
{
    name: "Rome, Italy",
    Status: {
        code: 200,
        request: "geocode"
    },
    Placemark: [
        {
            address: "Rome, Italy",
            population: "2.553M",
            Point: {
                coordinates: [12.482181, 41.895431, 0]
            },
            AddressDetails: {
                Country: {
                    CountryNameCode: "IT",
                    Locality: {LocalityName: "Roma"}
                }
            }
        }
    ]
},
{
    name: "Berlin, Germany",
    Status: {
        code: 200,
        request: "geocode"
    },
    Placemark: [
        {
            address: "Berlin, Germany",
            population: "3.396M",
            Point: {
                coordinates: [13.411895, 52.523781, 0]
            },
            AddressDetails: {
                Country: {
                    CountryNameCode: "DE",
                    Locality: {LocalityName: "Berlin"}
                }
            }
        }
    ]
},
{
    name: "Madrid, Spain",
    Status: {
        code: 200,
        request: "geocode"
    },
    Placemark: [

```



```

        {
            address: "Madrid, Spain",
            population: "3.228M",
            Point: {
                coordinates: [-3.703270, 40.416712, 0]
            },
            AddressDetails: {
                Country: {
                    CountryNameCode: "ES",
                    Locality: {LocalityName: "Madrid"}
                }
            }
        }
    ]
}
];

var map;
var geocoder;

// CapitalCitiesCache is a custom cache that extends the standard
GeocodeCache.
// We call apply(this) to invoke the parent's class constructor.
function CapitalCitiesCache() {
    GGeocodeCache.apply(this);
}

// Assigns an instance of the parent class as a prototype of the
// child class, to make sure that all methods defined on the parent
// class can be directly invoked on the child class.
CapitalCitiesCache.prototype = new GGeocodeCache();

// Override the reset method to populate the empty cache with
// information from our array of geocode responses for capitals.
CapitalCitiesCache.prototype.reset = function() {
    GGeocodeCache.prototype.reset.call(this);
    for (var i in city) {
        this.put(city[i].name, city[i]);
    }
}

function initialize() {
    map = new GMap2(document.getElementById("map_canvas"));
    map.setCenter(new GLatLng(37.441944, -122.141944), 6);
    geocoder = new GClientGeocoder();
    geocoder.setCache(new CapitalCitiesCache());
}

function addAddressToMap(response) {
    map.clearOverlays();

    if (response && response.Status.code != 200) {
        alert("Unable to locate " + decodeURIComponent(response.name));
    } else {
        var place = response.Placemark[0];
        var point = new GLatLng(place.Point.coordinates[1],

```

```

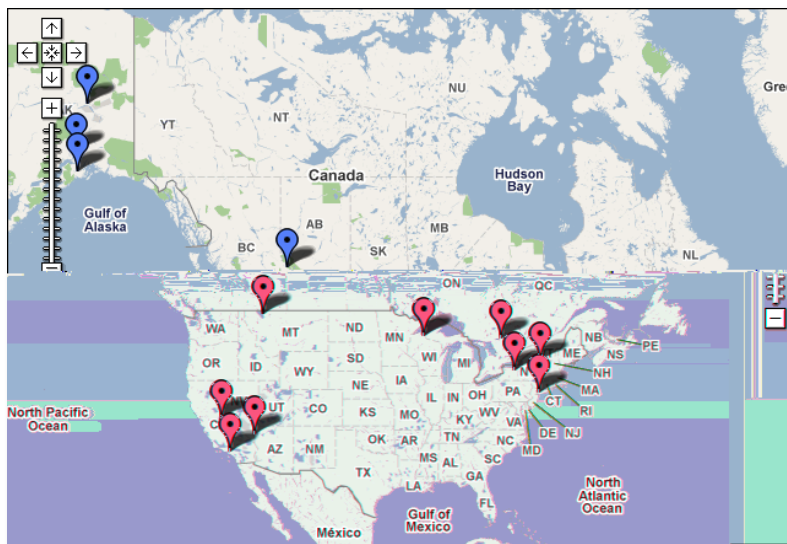
        place.Point.coordinates[0]);
    map.setCenter(point, 6);
    map.openInfoWindowHtml(point, "<b>City:</b> " + place.address
        + "<br><b>Population:</b> " + place.population);
}
}

function findCity(which) {
    if (which != 0) {
        geocoder.getLocations(city[which - 1].name, addAddressToMap);
    }
}
}

```

Service: Menampilkan titik-titik berdasarkan RSS

Kita juga dapat menyimpan titik-titik atau data lain yang dibutuhkan dalam bentuk layanan RSS, kemudian meminta Google Maps API untuk menampilkannya.



```

var map;
var geoXml;
var toggleState = 1;

function initialize() {
    if (GBrowserIsCompatible()) {

        geoXml = new GGeoXml("http://api.flickr.com/services/feeds/geo/?g=322338@N2
0&lang=en-us&format=feed-georss");
        map = new GMap2(document.getElementById("map_canvas"));
        map.setCenter(new GLatLng(49.496675, -102.65625), 3);
        map.addControl(new GLargeMapControl());
        map.addControl(new GLargeMapControl());
        map.addOverlay(geoXml);
    }
}

function toggleMyKml() {

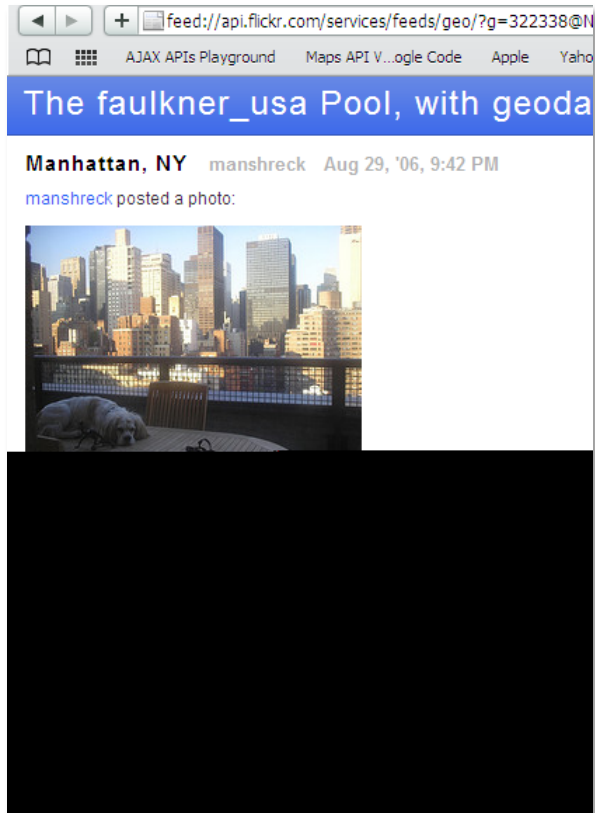
```

```

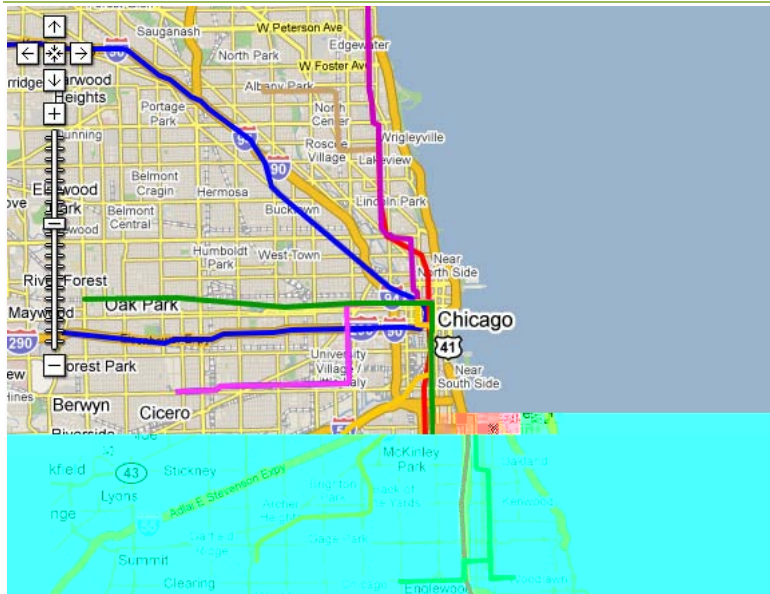
if (toggleState == 1) {
    map.removeOverlay(geoXml);
    toggleState = 0;
} else {
    map.addOverlay(geoXml);
    toggleState = 1;
}
}

```

Jika RSS-nya ditampilkan dalam Browser Safari, hasilnya sebagai berikut:



Service: Menampilkan lokasi-lokasi berdasarkan KML



```
var map;
var geoXml;
var toggleState = 1;

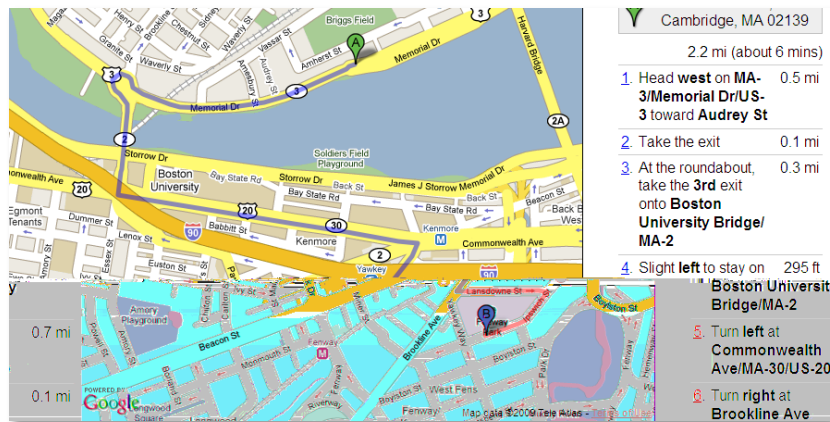
function initialize() {
    if (GBrowserIsCompatible()) {
        geoXml = new GGeoXml("http://mapgadgets.googlepages.com/cta.kml");
        map = new GMap2(document.getElementById("map_canvas"));
        map.setCenter(new GLatLng(41.875696, -87.624207), 11);
        map.addControl(new GLargeMapControl());
        map.addControl(new GLargeMapControl());
        map.addOverlay(geoXml);
    }
}

function toggleMyKml() {
    if (toggleState == 1) {
        map.removeOverlay(geoXml);
        toggleState = 0;
    } else {
        map.addOverlay(geoXml);
        toggleState = 1;
    }
}
```

[Tutorial tentang KML bisa dilihat di sini.](#)

Map Direction: Menampilkan bagaimana menuju suatu tempat dari suatu tempat lain

Kita bisa pula menampilkan bagaimana menuju suatu tempat dari tempat lain dalam peta. Contohnya berikut ini kita coba menampilkannya untuk lokasi di Boston.



```
// Create a directions object and register a map and DIV to hold the
// resulting computed directions
```

```
var map;
var directionsPanel;
var directions;

function initialize() {
    map = new GMap2(document.getElementById("map_canvas"));
    map.setCenter(new GLatLng(42.351505, -71.094455), 15);
    directionsPanel = document.getElementById("route");
    directions = new GDirections(map, directionsPanel);
    directions.load("from: 500 Memorial Drive, Cambridge, MA to: 4 Yawkey Way,
Boston, MA 02215 (Fenway Park)");
}
```

Map Direction: Menampilkan jalan dari ... ke ... secara lebih detail

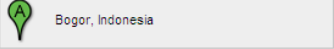
Jika sebelumnya kita melihat data-data untuk kota Boston, mari kita coba lihat untuk kota yang lebih dekat dengan kita. Arah dari Bogor menuju Depok. Contoh programnya bisa Anda lihat berikut ini. Sayang petunjuk-petunjuknya (sebelah kiri) tertulis dalam bahasa Inggris saat ini. Tetapi, ada solusinya agar petunjuk tersebut menggunakan Bahasa Indonesia.

Maps API Directions Illustrated

From: To:

Language:

Formatted Directions

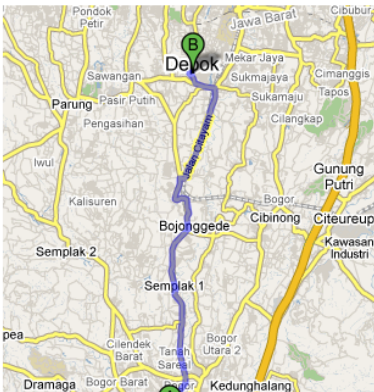


Bogor, Indonesia

26.6 km (about 31 mins)

1. Head north on Jalan Sawo Jajar toward Jalan Cibalok 0.5 km
2. Turn left at Jalan Jenderal Sudirman - Bogor 0.1 km
3. Continue on Jalan Jenderal Ahmad Yani - Bogor 0.6 km
4. Continue on Jalan Pemuda 1.4 km
5. Slight left at Jalan Kebon Pedes 1.0 km
6. Turn left at Jalan Kyai Haji Soleh Iskandar 65 m
7. Make a U-turn 94 m
8. Turn left at Jalan Cilebut Raya 3.5 km
9. Continue on Jalan Bojongsari 2.0 km

Map



```

var map;
var gdir;
var geocoder = null;
var addressMarker;

function initialize() {
    if (GBrowserIsCompatible()) {
        map = new GMap2(document.getElementById("map_canvas"));
        gdir = new GDirections(map, document.getElementById("directions"));
        GEvent.addListener(gdir, "load", onGDirectionsLoad);
        GEvent.addListener(gdir, "error", handleErrors);

        setDirections("San Francisco", "Mountain View", "en_US");
    }
}

function setDirections(fromAddress, toAddress, locale) {
    gdir.load("from: " + fromAddress + " to: " + toAddress,
        { "locale": locale });
}

function handleErrors(){
    if (gdir.getStatus().code == G_GEO_UNKNOWN_ADDRESS)
        alert("No corresponding geographic location could be found for one of the
specified addresses. This may be due to the fact that the address is
relatively new, or it may be incorrect.\nError code:
" + gdir.getStatus().code);
    else if (gdir.getStatus().code == G_GEO_SERVER_ERROR)
        alert("A geocoding or directions request could not be successfully
processed, yet the exact reason for the failure is not known.\n Error code:
" + gdir.getStatus().code);

    else if (gdir.getStatus().code == G_GEO_MISSING_QUERY)
        alert("The HTTP q parameter was either missing or had no value. For
geocoder requests, this means that an empty address was specified as input.
For directions requests, this means that no query was specified in the
input.\n Error code: " + gdir.getStatus().code);
}

```

```

// else if (gdir.getStatus().code == G_UNAVAILABLE_ADDRESS) <--- Doc
bug... this is either not defined, or Doc is wrong
// alert("The geocode for the given address or the route for the given
directions query cannot be returned due to legal or contractual reasons.\n
Error code: " + gdir.getStatus().code);

else if (gdir.getStatus().code == G_GEO_BAD_KEY)
    alert("The given key is either invalid or does not match the domain for
which it was given. \n Error code: " + gdir.getStatus().code);

else if (gdir.getStatus().code == G_GEO_BAD_REQUEST)
    alert("A directions request could not be successfully parsed.\n Error
code: " + gdir.getStatus().code);

else alert("An unknown error occurred.");
}

function onGDirectionsLoad(){
    // Use this function to access information about the latest load()
    // results.

    // e.g.
    // document.getElementById("getStatus").innerHTML =
gdir.getStatus().code;
    // and yada yada yada...
}

```

Karena arah ini ditunjukkan dalam bahasa English, ada baiknya kita menggunakan Google Translate API untuk menerjemahkannya. Hasilnya pun lumayan. Contoh bagaimana menggunakan Google Translate API. Kebetulan contohnya adalah menerjemahkan bahasa Spanyol ke Inggris. Tapi, saya juga sudah coba menerjemahkannya ke dalam bahasa Indonesia. Hasilnya cukup memuaskan.

```

/*
 * How to translate text.
 */

google.load("language", "1");

function initialize() {
    var content = document.getElementById('content');
    // Setting the text in the div.
    content.innerHTML = '<div id="text">Hola, me alegro mucho de
verte.</div><div id="translation">';

    // Grabbing the text to translate
    var text = document.getElementById("text").innerHTML;

    // Translate from Spanish to English, and have the callback of the request
    // put the resulting translation in the "translation" div.
    // Note: by putting in an empty string for the source language ('es') then
the translation
    // will auto-detect the source language.
    google.language.translate(text, 'es', 'id', function(result) {

```

```
var translated = document.getElementById("translation");
if (result.translation) {
    translated.innerHTML = result.translation;
}
});
}
google.setOnLoadCallback(initialize);
```

Penutup

Google Maps API muncul dengan berbagai fitur yang menakjubkan. Kita bisa memanfaatkan API ini untuk keperluan development aplikasi-aplikasi kita yang membutuhkan kemampuan standar sebuah peta. Dengan Google Maps API, kita tak perlu lagi repot-repot membangun system peta apalagi (dan ini yang paling sulit) mendapatkan data peta-peta yang luar biasa besar.

Tantangan sebenarnya saat ini adalah, bagaimana memanfaatkan apa yang ada di depan mata kita (Google Maps API) untuk menghasilkan *sesuatu* yang lebih berarti lagi. Saya telah menunjukkan dasar-dasar pemrograman Google Maps API. Sekarang, terserah Anda bagaimana memanfaatkannya.

Referensi

1. [Website resmi Google Maps versi 3](#)
2. [Referensi API](#)
3. [Static Maps](#)
4. [Contoh implementasi](#)
5. [Beberapa artikel tentang Google Maps API](#)

Tentang penulis



Idealisme saya sebagai mantan dosen memancing saya untuk terus menulis. Ebook pertama yang saya release sekitar seminggu yang lalu, [Ebook Tutorial J2ME Client Server Database](#), terbilang cukup sukses. Inilah yang membuat saya semakin semangat untuk menulis dan menulis. Saya yakin, sesederhana apapun isi tulisan saya, pasti ada juga orang lain yang bisa mendapatkan manfaatnya.

Riwayat pekerjaan:

- Programmer di Bidang Perangkat Lunak, Deputy Pengkajian Persandian, Lembaga Sandi Negara
- Dosen mata kuliah Rekayasa Perangkat Lunak di Sekolah Tinggi Sandi Negara
- Java Developer dan konsultan keamanan di PT. Artajasa Pembayaran Elektronik
- ERP Developer di PT. Saptaindra Sejati