

INTEGRASI PHP/MySQL DENGAN GOOGLE MAPS API

Andi Sunyoto

Dosen STMIK AMIKOM Yogyakarta

Abstraksi

*This article is intended who want to learn how to use Google Maps with a MySQL database. After completing this tutorial, you will have a Google Map based off a database of places. An info window with name and address information will display above a marker when clicked. By using an XML file as an intermediary between our database and our Google Map, it makes for a faster initial page load, a more flexible map application, and easier debugging. You can independently verify the XML output from the database and the JavaScript parsing of the XML. And at any point, you could even decide to eliminate your database entirely and just run the map based on static XML files. In the PHP, first initialize a new XML document and create the "markers" parent node. Then connect to the database, execute a `SELECT * (select all)` query on the markers table, and iterate through the results. For each row in the table (each location), create a new XML node with the row attributes as XML attributes, and append it to the parent node. Then dump the XML to the screen.*

Keyword: Google, Map, Ajax, Database, MySQL, PHP

Pendahuluan

Tulisan ini membahas bagaimana mengintegrasikan antara database MySQL dan PHP dan Google Maps API, ditujukan untuk orang yang sudah memahami PHP/MySQL. Setelah selesai membaca tulisan ini, anda akan mempunyai kemampuan Google Map ditambah dengan pengolahan database MySQL dan bagaimana cara mengintegrasikannya. Peta akan menampilkan dua jenis data, yaitu restoran dan bar dengan memberikan penanda yang berbeda icon. Sebuah jendela info dengan nama dan informasi alamat di atas akan menampilkan marker saat diklik.

Tutorial ini dipecah menjadi langkah-langkah berikut:

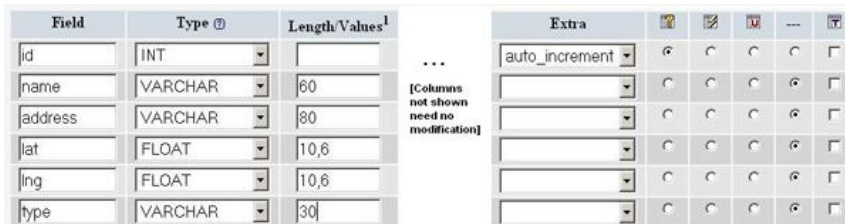
1. Membuat tabel

2. Mengisi tabel
3. Membuat keluaran XML dengan PHP
4. Membuat peta

Pembahasan

Membuat Tabel

Ketika membuat tabel MySQL, anda harus memberi perhatian khusus pada atribut lat dan lng. Atribut lat dan lng digunakan untuk menyimpan data posisi. Atribut lat untuk menyimpan data koordinat latitude dan atribut lng digunakan untuk menyimpan data koordinat longitude. Dengan kemampuan zoom yang ada dari Google Maps, kita hanya memerlukan presisi enam digit setelah decimal. Untuk menjaga agar ruang penyimpanan yang diperlukan pada tabel kita minimum, kita mendefinisikan atribut lat dan lng adalah float (10,6). Itu akan menyediakan 6 digit setelah decimal, ditambah 4 angka sebelum decimal, misal -123,456789 derajat. Tabel Anda juga harus memiliki sebuah atribut id untuk dijadikan primary key, dan atribut untuk membedakan antara restoran dan bar.



Gambar 1 membuat tabel pada phpMyAdmin

Jika tidak mempunyai akses ke phpMyAdmin berikut perintah SQL untuk membuat tabel:

```
CREATE TABLE `markers` (  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
  `name` VARCHAR( 60 ) NOT NULL ,  
  `address` VARCHAR( 80 ) NOT NULL ,
```

```
`lat` FLOAT( 10, 6 ) NOT NULL ,  
`lng` FLOAT( 10, 6 ) NOT NULL ,  
`type` VARCHAR( 30 ) NOT NULL  
) ENGINE = MYISAM ;
```

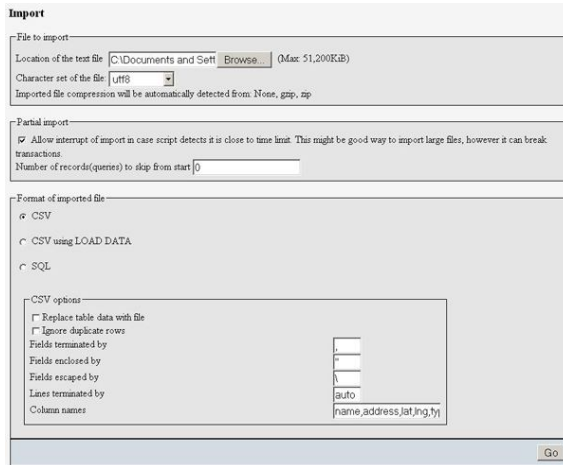
Mengisi Tabel

Setelah tabel telah terbentuk, sekarang waktunya mengisi tabel tersebut. Contoh diambil sepuluh data yang telah disediakan sebelumnya. Pada phpMyAdmin, kita dapat mengimport dari berbagai format, termasuk CSV (*comma-separated value*). Microsoft Excel dan Google Spreadsheet keduanya dapat di export ke format CSV. Format CSV mudah untuk ditransfer ke tabel di MySQL.

Berikut contoh data dalam format CSV.

```
Pan Africa Market,"1521 1st Ave, Seattle, WA",47.608941,-  
122.340145,restaurant  
Buddha Thai & Bar,"2222 2nd Ave, Seattle, WA",47.613591,-  
122.344394,bar  
The Melting Pot,"14 Mercer St, Seattle, WA",47.624562,-  
122.356442,restaurant  
Ipanema Grill,"1225 1st Ave, Seattle, WA",47.606366,-  
122.337656,restaurant  
Sake House,"2230 1st Ave, Seattle, WA",47.612825,-  
122.34567,bar  
Crab Pot,"1301 Alaskan Way, Seattle, WA",47.605961,-  
122.34036,restaurant  
Mama's Mexican Kitchen,"2234 2nd Ave, Seattle,  
WA",47.613975,-122.345467,bar  
Wingdome,"1416 E Olive Way, Seattle, WA",47.617215,-  
122.326584,bar  
Piroshky Piroshky,"1908 Pike pl, Seattle, WA",47.610127,-  
122.342838,restaurant
```

Berikut screenshot untuk import menggunakan CSV ke dalam tabel.



Gambar 2 screenshoot untuk import menggunakan CSV

Jika lebih suka tidak menggunakan antarmuka phpMyAdmin, di sini adalah perintah SQL yang hasilnya sama.

```
INSERT INTO `markers` (`name`, `address`, `lat`, `lng`, `type`)
VALUES ('Pan Africa Market', '1521 1st Ave, Seattle, WA',
'47.608941', '-122.340145', 'restaurant');
INSERT INTO `markers` (`name`, `address`, `lat`, `lng`, `type`)
VALUES ('Buddha Thai & Bar', '2222 2nd Ave, Seattle, WA',
'47.613591', '-122.344394', 'bar');
INSERT INTO `markers` (`name`, `address`, `lat`, `lng`, `type`)
VALUES ('The Melting Pot', '14 Mercer St, Seattle, WA',
'47.624562', '-122.356442', 'restaurant');
INSERT INTO `markers` (`name`, `address`, `lat`, `lng`, `type`)
VALUES ('Ipanema Grill', '1225 1st Ave, Seattle, WA',
'47.606366', '-122.337656', 'restaurant');
INSERT INTO `markers` (`name`, `address`, `lat`, `lng`, `type`)
VALUES ('Sake House', '2230 1st Ave, Seattle, WA', '47.612825',
'-122.34567', 'bar');
```

```
INSERT INTO `markers` (`name`, `address`, `lat`, `lng`, `type`)  
VALUES ('Crab Pot', '1301 Alaskan Way, Seattle, WA',  
'47.605961', '-122.34036', 'restaurant');  
INSERT INTO `markers` (`name`, `address`, `lat`, `lng`, `type`)  
VALUES ('Mama\s Mexican Kitchen', '2234 2nd Ave, Seattle,  
WA', '47.613975', '-122.345467', 'bar');  
INSERT INTO `markers` (`name`, `address`, `lat`, `lng`, `type`)  
VALUES ('Wingdome', '1416 E Olive Way, Seattle, WA',  
'47.617215', '-122.326584', 'bar');  
INSERT INTO `markers` (`name`, `address`, `lat`, `lng`, `type`)  
VALUES ('Piroshky Piroshky', '1908 Pike pl, Seattle, WA',  
'47.610127', '-122.342838', 'restaurant');
```

Menghasilkan XML dengan PHP

Saat ini, anda sudah mempunyai sebuah tabel bernama “markers” diisi dengan sepuluh contoh data. Sekarang Anda perlu menuliskan perintah PHP untuk mengekspor data ke dalam format XML yang dapat diambil oleh peta melalui asynchronous JavaScript. Jika belum pernah membuat skrip PHP untuk mengkoneksikan database MySQL, Anda harus mengunjungi php.net dan membaca di `mysql_connect`, `mysql_select_db`, `mysql_query`, dan `mysql_error`.

Pertama, Anda harus memasukkan informasi koneksi database pada file terpisah. Ini umumnya merupakan ide baik, karena sering menggunakan PHP untuk akses database. Berikut isi file yang disimpan untuk akses database (`dbinfo.php`).

```
<?  
$username="username";  
$password="password";  
$database="username-databaseName";  
>
```

Menggunakan Fungsi PHP domXML

Periksa konfigurasi atau berusaha inialisasi sebuah `domxml_new_doc()` untuk menentukan fungsionalitas `dom_xml` pada

server PHP Anda. Jika anda memiliki akses fungsi `dom_xml`, anda dapat menggunakannya membuat XML node, menambahkan node, dan menampilkan node XML ke layar. Fungsi `dom_xml` dapat membantu membuat struktur yang lebih kompleks.

Dalam PHP, pertama menginisialisasi dokumen XML baru dan menciptakan “markers” parent node. Kemudian mengkoneksikan dengan database, mengeksekusi sebuah query `SELECT *` pada tabel markers, dan melakukan perulangan sampai selesai. Tiap baris tabel, membuat node XML baru dengan baris atribut sebagai XML atribut, dan ditambahkan ke simpul orang tua (parent node). Kemudian menampilkan XML ke layar.

Catatan: jika database mengandung karakter internasional atau sebaiknya dipaksa output UTF-8, anda dapat menggunakan `utf8_encode` pada data yang dikeluarkan.

File PHP (`genxml.php`) dapat dilihat pada skrip berikut:

```
<?php
require("dbinfo.php");
// Start XML file, create parent node
$doc = domxml_new_doc("1.0");
$node = $doc->create_element("markers");
$parnode = $doc->append_child($node);
// Opens a connection to a MySQL server
$connection=mysql_connect (localhost, $username, $password);
if (!$connection) {
    die('Not connected : ' . mysql_error());
}
// Set the active MySQL database
$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {
    die ('Can\'t use db : ' . mysql_error());
}
// Select all the rows in the markers table
$query = "SELECT * FROM markers WHERE 1";
$result = mysql_query($query);
```

```
if (!$result) {
    die('Invalid query: ' . mysql_error());
}
header("Content-type: text/xml");
// Iterate through the rows, adding XML nodes for each
while ($row = @mysql_fetch_assoc($result)){
    // ADD TO XML DOCUMENT NODE
    $node = $doc->create_element("marker");
    $newnode = $parnode->append_child($node);
    $newnode->set_attribute("name", $row['name']);
    $newnode->set_attribute("address", $row['address']);
    $newnode->set_attribute("lat", $row['lat']);
    $newnode->set_attribute("lng", $row['lng']);
    $newnode->set_attribute("type", $row['type']);
}
$xmlfile = $doc->dump_mem();
echo $xmlfile;
?>
```

Menggunakan PHP echo

Jika kita tidak memiliki akses ke fungsi `dom_xml` PHP, maka kita dapat membuat output simple menggunakan fungsi `echo`. Ketika menggunakan fungsi `echo`, anda harus menggunakan fungsi penolong (misal `parseToXML`) yang benar meng encode entitas khusus untuk XML (`<.>.'.'').` Dalam PHP, pertama mengkneksikan database, mengeksekusi perintah query `SELECT * pada tabel markers.` Kemudian meng echo node parent marker, dan berulang sampai hasil query habis. Setiap baris masing tabel (masing-masing lokasi), kit amemerlukan echo node XML untuk marker, mengirimkan nama dan alamat field malalui fungsi `parseToXML`. Terakhir mengirimkan skrip menggunakan echo untuk menutup tag marker.

PHP file yang melakukan semua ini ditampilkan di bawah ini (`genxml2.php`):

```
<?php
require("dbinfo.php");
function parseToXML($htmlStr)
{
```

```
$xmlStr=str_replace('<','&lt;',$htmlStr);
$xmlStr=str_replace('>','&gt;',$xmlStr);
$xmlStr=str_replace('"','&quot;',$xmlStr);
$xmlStr=str_replace("'",'&#39;',$xmlStr);
$xmlStr=str_replace("&","&amp;",$xmlStr);
return $xmlStr;
}
// Opens a connection to a MySQL server
$connection=mysql_connect (localhost, $username, $password);
if (!$connection) {
    die('Not connected : ' . mysql_error());
}
// Set the active MySQL database
$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {
    die ('Can\'t use db : ' . mysql_error());
}
// Select all the rows in the markers table
$query = "SELECT * FROM markers WHERE 1";
$result = mysql_query($query);
if (!$result) {
    die('Invalid query: ' . mysql_error());
}
header("Content-type: text/xml");
// Start XML file, echo parent node
echo '<markers>';
// Iterate through the rows, printing XML nodes for each
while ($row = @mysql_fetch_assoc($result)){
    // ADD TO XML DOCUMENT NODE
    echo '<marker ' ;
    echo 'name="' . parseToXML($row['name']) . '" ' ;
    echo 'address="' . parseToXML($row['address']) . '" ' ;
    echo 'lat="' . $row['lat'] . '" ' ;
    echo 'lng="' . $row['lng'] . '" ' ;
```



```
    echo 'type="' . $row['type'] . '"';  
    echo '>';  
}  
// End XML file  
echo '</markers>';  
>
```

Menggunakan Fungsi PHP DOM

Pertama, periksa konfigurasi dan pastikan menggunakan PHP5. Jika tidak, maka gunakan teknik sebelumnya. Pada PHP, pertama menginisialisasi document XML dan membuat parent node “markers”. Kemudian koneksi ke database, mengeksekusi perintah SELECT * pada tabel marker, mengulang sampai seluruh hasil. Pada tiap baris dalam tabel (masing-masing likasi), buat node XML baru dengan atribut sebagai atribut XML, tambahkan parent node. Kemudian tampilkan hasil XML ke layar.

PHP file yang melakukan semua ini ditampilkan di bawah ini (genxml3.php):

```
<?php  
require("dbinfo.php");  
// Start XML file, create parent node  
$dom = new DOMDocument("1.0");  
$node = $dom->createElement("markers");  
$parnode = $dom->appendChild($node);  
// Opens a connection to a MySQL server  
$connection=mysql_connect (localhost, $username, $password);  
if (!$connection) { die('Not connected : ' . mysql_error());}  
// Set the active MySQL database  
$db_selected = mysql_select_db($database, $connection);  
if (!$db_selected) {  
    die ('Can\'t use db : ' . mysql_error());  
}  
// Select all the rows in the markers table  
$query = "SELECT * FROM markers WHERE 1";  
$result = mysql_query($query);  
if (!$result) {  
    die('Invalid query: ' . mysql_error());
```

```
}  
header("Content-type: text/xml");  
// Iterate through the rows, adding XML nodes for each  
while ($row = @mysql_fetch_assoc($result)){  
    // ADD TO XML DOCUMENT NODE  
    $node = $dom->createElement("marker");  
    $newnode = $parnode->appendChild($node);  
    $newnode->setAttribute("name", $row['name']);  
    $newnode->setAttribute("address", $row['address']);  
    $newnode->setAttribute("lat", $row['lat']);  
    $newnode->setAttribute("lng", $row['lng']);  
    $newnode->setAttribute("type", $row['type']);  
}  
echo $dom->saveXML();  
?>
```

Loading File XML

File XML harus diload ke halaman web, anda dapat memanfaatkan fungsi API `GDownloadURL`. `GDownloadURL` adalah pembungkus untuk `XMLHttpRequest` yang digunakan untuk request file XML dari server dimana halaman HTML berada. Parameter pertama `GDownloadURL` adalah path file kita, biasanya paling mudah adalah file XML jadi satu direktori dengan file HTML sehingga hanya merefensi nama filenya saja. Parameter kedua `GDownloadURL` fungsi yang dipanggil saat XML dikembalikan lagi ke JavaScript.

Dalam fungsi callback, anda perlu mencari semua elemen “marker” dalam XML, dan beriterasi melalui mereka. Untuk setiap penanda elemen yang anda temukan, atribut mengambil nama, alamat, jenis, dan lat / lng dan meneruskannya ke `createMarker`, yang akan mengembalikan sebuah marker yang akan ditambahkan ke peta.

```
GDownloadUrl("genxml.php", function(data) {  
    var xml = GXml.parse(data);  
    var markers =  
xml.documentElement.getElementsByTagName("marker");  
    for (var i = 0; i < markers.length; i++) {
```

```
var name = markers[i].getAttribute("name");
var address = markers[i].getAttribute("address");
var type = markers[i].getAttribute("type");
var point = new
GLatLng(parseFloat(markers[i].getAttribute("lat")),
parseFloat(markers[i].getAttribute("lng")));
var marker = createMarker(point, name, address, type);
map.addOverlay(marker);
}
});
```

Membuat Kustomisasi Icon

Anda dapat menggunakan kelas `GIcon` untuk mendefinisikan icon yang kemudian akan di pasang pada marker. Dimulai dari mendefinisikan dua obyek `GIcon` `iconBlue` dan `iconRed` dan menetapkan properti mereka. Anda kemudian membuat sebuah array asosiatif yang masing-masing `GIcon` dengan tipe string” ‘restaurant’ atau ‘bar’. Hal ini akan membuat icon mudah untuk referensi kemudian ketika membuat marker dari XML.

```
var iconBlue = new GIcon();
iconBlue.image =
'http://labs.google.com/ridefinder/images/mm_20_blue.png';
iconBlue.shadow =
'http://labs.google.com/ridefinder/images/mm_20_shadow.png';
iconBlue.iconSize = new GSize(12, 20);
iconBlue.shadowSize = new GSize(22, 20);
iconBlue.iconAnchor = new GPoint(6, 20);
iconBlue.infoWindowAnchor = new GPoint(5, 1);
var iconRed = new GIcon();
iconRed.image =
'http://labs.google.com/ridefinder/images/mm_20_red.png';
iconRed.shadow =
'http://labs.google.com/ridefinder/images/mm_20_shadow.png';
iconRed.iconSize = new GSize(12, 20);
iconRed.shadowSize = new GSize(22, 20);
iconRed.iconAnchor = new GPoint(6, 20);
```

```
iconRed.infoWindowAnchor = new GPoint(5, 1);  
var customIcons = [];  
customIcons["restaurant"] = iconBlue;  
customIcons["bar"] = iconRed;
```

Membuat Marker dan Info Window

Anda harus mempunyai semua marker yang telah diciptakan pada fungsi `createMarker`. Anda dapat mengambil `GIcon` yang sesuai dengan menggunakan tipe sebagai kunci untuk array asosiatif yang didefinisikan secara global, dan melewatkan ke konstruktor `GMarker`. Kemudian, membangun HTML yang ingin anda munculkan di info window yang berisi nama, alamat, beberapa tag dibekalkan.

Tip: beberapa artikel menyarankan menyimpan dalam format HTML di dalam database. Tetapi dengan cara demikian, susah untuk dimodifikasi tampilannya. Dengan menunggu sampai data diterima JavaScript terpisah, kita lebih bebas bermain dengan HTML pada sisi client dapat dengan cepat dilihat untuk format baru. Setelah membangun string HTML, tambahkan satu event listener ke marker ketika diklik, sebuah info window ditampilkan.

```
function createMarker(point, name, address, type) {  
    var marker = new GMarker(point, customIcons[type]);  
    var html = "<b>" + name + "</b> <br/>" + address;  
    GEvent.addListener(marker, 'click', function() {  
        marker.openInfoWindowHtml(html);  
    });  
    return marker;  
}
```

Mengabungkan Semua Menjadi Satu

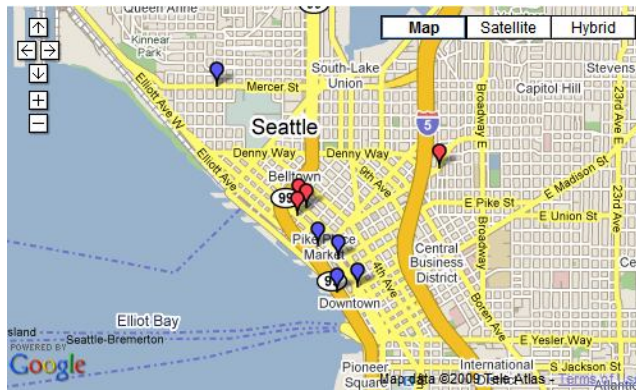
Berikut halaman web yang mengikat marker, icon, dan XML bersama. Ketika halaman diload, fungsi `load` dipanggil. Fungsi ini menset peta kemudian memanggil `GDownloadURL`. Pastikan melalui `GDownloadURL` melewatkan file untuk menampilkan XML dan sudah di preview di browser.

HTML yang lengkap yang menyelesaikan ini ditunjukkan di bawah ini (`map.htm`):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="content-type" content="text/html;
charset=utf-8"/>
    <title>Google Maps AJAX + MySQL/PHP Example</title>
    <script
src="http://maps.google.com/maps?file=api&v=2&key=ABQIAA
AAjU0EJWnWPMv7oQ-
jjS7dYxTPZYEIJSBeBUeMSX5xXgq6lLjHthSAk20WnZ_iuuzh
Mt60X_ukms-AUg"
type="text/javascript"></script>
    <script type="text/javascript">
  <<![CDATA[
    var iconBlue = new GIcon();
    iconBlue.image =
'http://labs.google.com/ridefinder/images/mm_20_blue.png';
    iconBlue.shadow =
'http://labs.google.com/ridefinder/images/mm_20_shadow.png';
    iconBlue.iconSize = new GSize(12, 20);
    iconBlue.shadowSize = new GSize(22, 20);
    iconBlue.iconAnchor = new GPoint(6, 20);
    iconBlue.infoWindowAnchor = new GPoint(5, 1);
    var iconRed = new GIcon();
    iconRed.image =
'http://labs.google.com/ridefinder/images/mm_20_red.png';
    iconRed.shadow =
'http://labs.google.com/ridefinder/images/mm_20_shadow.png';
    iconRed.iconSize = new GSize(12, 20);
    iconRed.shadowSize = new GSize(22, 20);
    iconRed.iconAnchor = new GPoint(6, 20);
    iconRed.infoWindowAnchor = new GPoint(5, 1);
```

```
var customIcons = [];  
customIcons["restaurant"] = iconBlue;  
customIcons["bar"] = iconRed;  
function load() {  
    if (GBrowserIsCompatible()) {  
        var map = new GMap2(document.getElementById("map"));  
        map.addControl(new GSmallMapControl());  
        map.addControl(new GMapTypeControl());  
        map.setCenter(new GLatLng(47.614495, -122.341861), 13);  
        GDownloadUrl("phpsqlajax_genxml.php", function(data) {  
            var xml = GXml.parse(data);  
            var markers =  
xml.documentElement.getElementsByTagName("marker");  
            for (var i = 0; i < markers.length; i++) {  
                var name = markers[i].getAttribute("name");  
                var address = markers[i].getAttribute("address");  
                var type = markers[i].getAttribute("type");  
                var point =  
new  
GLatLng(parseFloat(markers[i].getAttribute("lat")),  
            parseFloat(markers[i].getAttribute("lng")));  
                var marker = createMarker(point, name, address, type);  
                map.addOverlay(marker);  
            }  
        });  
    }  
}  
function createMarker(point, name, address, type) {  
    var marker = new GMarker(point, customIcons[type]);  
    var html = "<b>" + name + "</b> <br/>" + address;  
    GEvent.addListener(marker, 'click', function() {  
        marker.openInfoWindowHtml(html);  
    });  
    return marker;  
}
```

```
//]]>  
</script>  
</head>  
<body onload="load()" onunload="GUnload()">  
  <div id="map" style="width: 500px; height: 300px"></div>  
</body>  
</html>
```



Gambar 3 marker di atas peta

Penutup

Sebuah website dapat ditambahi aplikasi peta menggunakan Google Maps API. Google Maps API dapat diintegrasikan dengan database MySQL dan PHP. Menggunakan file XML sebagai perantara antara database dan Google Map, ini membuat lebih cepat dalam membukan halaman awal, aplikasi peta akan lebih fleksibel, dan mudah dalam proses debugging. Proses di PHP, pertama menginisialisasi dokumen XML baru dan menciptakan “markers” parent node. Kemudian mengkoneksikan dengan database, mengeksekusi sebuah query SELECT * pada tabel markers, dan melakukan perulangan sampai selesai. Tiap baris tabel, membuat node XML baru dengan baris atribut sebagai XML atribut, dan

ditambahkan ke simpul orang tua (parent node). Kemudian menampilkan XML ke layar.

Daftar Pustaka

- Brown, C, Martin., Hacking Google® Maps and Google® Earth., Wiley Publishing, 2006
- Google.Code.,MapV3,.Tutorial.,<http://code.google.com/apis/maps/documentation/v3/introduction.html#Intro>, 2009
- Fox , Pamela., Google Geo Team, Stucker, Lary., Using PHP/MySQL with Google Maps., <http://code.google.com/apis/maps/articles/phpsqlajax.html>, 2007
- Purvis, Michael., Sambells, Jeffrey., Turner, Cameron., Beginning Google Maps Applications with PHP and Ajax From Novice to Professional., Apress, 2006