

BAB III ANALISA DAN PEMBAHASAN

A. Pekerjaan/Kegiatan

1. Pekerjaan Secara Umum

Secara umum pekerjaan yang dilakukan dalam Praktik Kerja Lapangan/Kerja Praktik di PT Transdata Sejahtera ada dua yaitu Implementasi IPv4 dan IPv6 dengan menggunakan DualStack dan pembuatan *website* Transdata.

Implementasi IPv4 dan IPv6 dengan menggunakan DualStack yaitu teknik menggunakan dua jaringan yang berbeda yaitu IPv4 dan IPv6 dalam satu *interface* dan dapat berjalan di waktu yang sama. Teknik *dual stack* banyak digunakan karena tidak banyak sistem yang menggunakan *ipv6 only* dan *dual stack* ini akan menjadi cara yang baik untuk sekarang karena seiring berjalan nya waktu IPv4 akan diganti dengan *ipv6 only*.

Pekerjaan kedua yaitu pembuatan *website* Transdata. *Website* ini dibuat karena Transdata sendiri belum memiliki *website* dan membantu Transdata dalam promosi jasa jaringan telekomunikasi.

2. Pekerjaan Secara Spesifik

Pekerjaan secara spesifiknya yaitu pembuatan *website* Transdata. *Website* tersebut memiliki beberapa menu antara lain menu *Home*, *About*, *Services*, *Product*, dan *Contact*. Menu *Home* berisi konten layanan yang disediakan Transdata, Visi, sekilas tentang Transdata, *button* untuk melihat area, dan alasan memilih Transdata. Sedangkan menu *About* berisi penjelasan mengenai Transdata dan tentang staf yang bekerja pada Transdata itu sendiri.

Services berisi konten jenis layanan apa saja yang ada di Transdata, sedangkan *Product* berisi konten mengenai jenis paket yang ditawarkan oleh Transdata. Terakhir yaitu menu *Contact* yang berisi alamat lengkap beserta *telephone* dan *email* serta terdapat *form* untuk mengirim pesan ke Transdata.

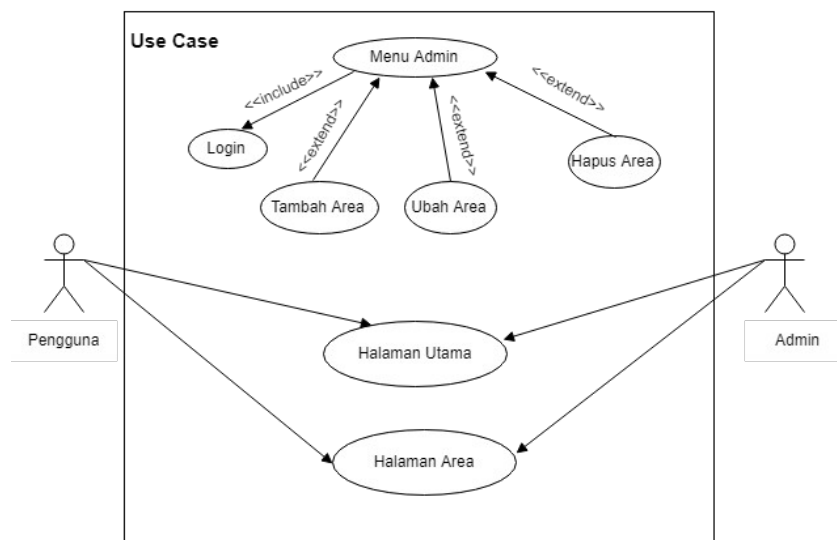
Pengerjaan *website* ini dilakukan oleh dua orang yang saling menuangkan ide mulai dari perancangan, pembuatan halaman masing-masing menu, admin, serta halaman tambah, ubah pada admin, dan tidak lupa dalam mempercantik *website*.

Penulis fokus pada implementasi *map* dengan *marker* di halaman area pada *website* Transdata. Penambahan ini dimaksudkan agar pengunjung *website* mengetahui area mana saja yang telah tercakup oleh Transdata. Seperti yang sudah dipaparkan bahwa untuk melihat area kita harus memilih *button* area yang terdapat pada menu *Home*.

B. Analisa dan Pembahasan Hasil Pekerjaan

Pembahasan pekerjaan secara spesifik yang penulis kerjakan yaitu implementasi *map* dengan *marker* di halaman area menggunakan Google Maps API namun, pada pengerjaan tersebut belum berjalan sesuai dengan yang diharapkan karena masih terdapat *error* pada *map* yang penulis buat. *Error* tersebut akan penulis jelaskan setelah menjelaskan langkah-langkah yang dilakukan dalam pengerjaan *map* ini.

Sebelum memasuki langkah pengerjaan, penulis akan menjelaskan *usecase diagram* pada pembahasan yang penulis ambil seperti gambar di bawah ini.

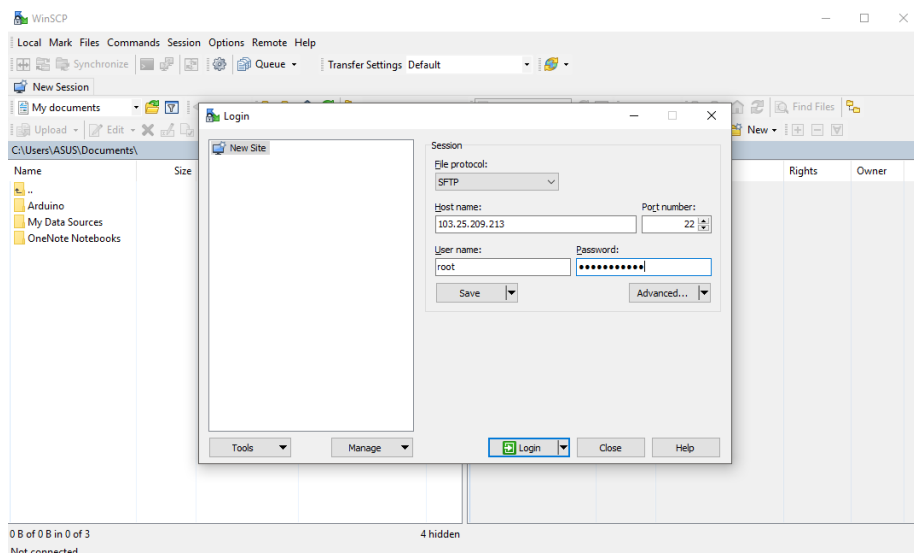


Gambar 3.1 *Usecase diagram*

Usecase diagram mendeskripsikan antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Aktor sendiri merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. *Usecase* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor ^[11].

Aktor pada usecase yang penulis buat yaitu Pengguna dan Admin. Pada *usecase* di atas, penulis menggunakan relasi *include* dan *extend*. *Include* yaitu relasi tambahan ke sebuah *usecase* di mana *usecase* yang ditambahkan memerlukan *usecase* lain untuk menjalankan fungsinya. Seperti pada *usecase* Menu Admin yang terhubung dengan *usecase* Login. Sedangkan relasi *extend* yaitu relasi *usecase* dimana *usecase* yang ditambahkan dapat berdiri sendiri walau tanpa *usecase* lain. Seperti pada *usecase* Tambah Area, Hapus Area, dan Ubah Area.

Salah satu rekan penulis yaitu Faris telah membuat *web server* pkl.transdata.net.id untuk meng-*upload* file *website* yang telah penulis dan rekan penulis (Ade) buat di *localhost*. Agar penulis dapat meng-*upload* file pada *web server* pkl.transdata.net.id, penulis terlebih dahulu me-*install* aplikasi WinSCP. Untuk *login*, penulis harus memasukkan *Host name*, *username*, dan *password* seperti pada gambar di bawah ini.

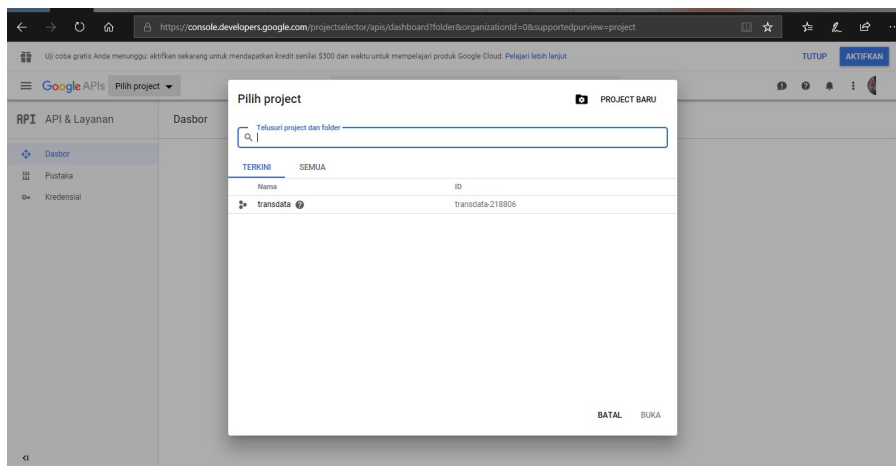


Gambar 3.2 Login WinSCP

Setelah penulis masuk ke dalam *web server*, selanjutnya penulis *upload* file yang dibutuhkan dalam implementasi *map*. Penulis akan menjelaskan langkah-langkah penambahan *google maps* secara runtut yaitu sebagai berikut:

1. Mendapatkan API *maps*

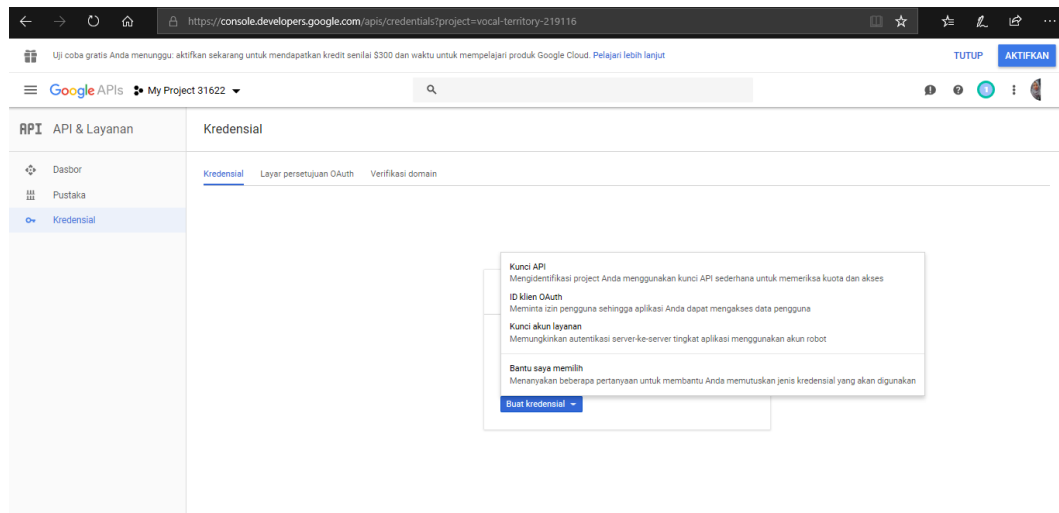
- a. Untuk mendapatkan API Google Maps, kita dapat membuka halaman <https://console.developers.google.com/projectselector/apis/dashboard?folder&organizationId=0&supportedpurview=project>. Tampilan halaman tersebut adalah sebagai berikut.



Gambar 3.3 Tampilan Halaman Google API

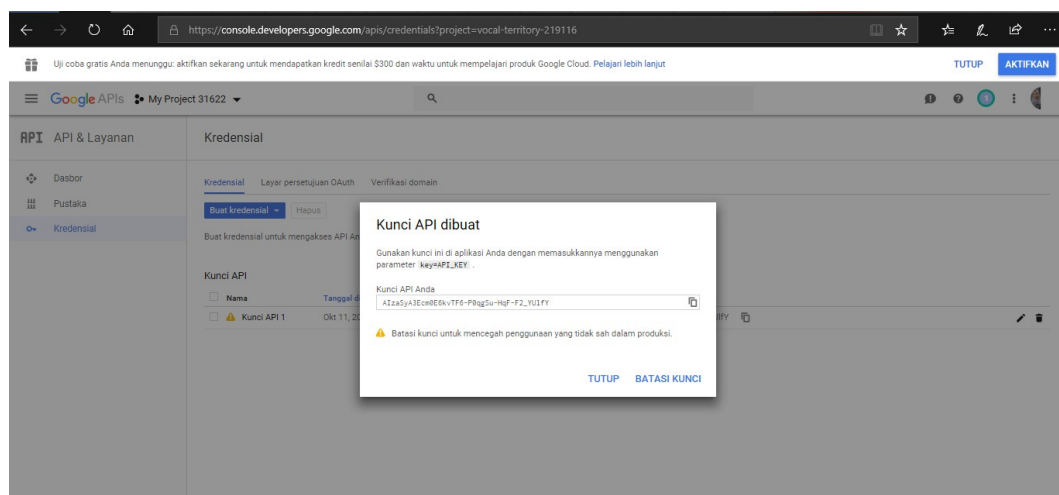
Untuk memulainya kita harus membuat *project* dulu dengan cara klik **Pilih *project*** lalu pada jendela yang muncul klik **PROJECT BARU** lalu isikan nama *project* nya.

- b. Setelah masuk ke *project* nya, pilih menu **Kredensial**, klik **Buat kredensial**, lalu pilih **Kunci API** seperti pada gambar dibawah ini.



Gambar 3.4 Menu Kredensial untuk API

c. Tunggu sampai muncul jendela bahwa kunci API telah dibuat. Seperti gambar di bawah ini.



Gambar 3.5 Kunci API

d. Selanjutnya yaitu membuat kredensial pada kunci API yang telah kita dapatkan. Kredensial ini bertujuan untuk membatasi kunci API pada aplikasi tertentu agar API tersebut dapat digunakan. Langkahnya yaitu pilih menu **Kredensial** lalu klik saja nama API nya, lalu buat seperti gambar di bawah ini. Penulis memilih pembatasan aplikasi pada perujuk HTTP (situs *web*) karena *API maps* akan digunakan pada *website* pkl.transdata.net.id.

☰ Google APIs trans ▼ 🔍

← Kunci API 🔄 BUAT ULANG KUNCI 🗑️ HAPUS

Kunci API 1

Pembatasan kunci

Pembatasan mencegah penggunaan yang tidak sah dan pencurian kuota. [Learn more](#)

Pembatasan aplikasi: Perujuk HTTP ⚠️ Pembatasan API: Tidak ada

Pembatasan aplikasi Pembatasan API

Pembatasan aplikasi menentukan situs, alamat IP, atau aplikasi yang dapat menggunakan kunci ini. Anda dapat menyetel satu jenis pembatasan per kunci.

Pembatasan aplikasi

Tidak ada

Perujuk HTTP (situs web)

Alamat IP (server web, cron job, dll.)

Aplikasi Android

Aplikasi iOS

Terima permintaan dari perujuk HTTP ini (situs web) (Opsional)

Gunakan * untuk karakter pengganti. Jika Anda biarkan kosong, permintaan akan diterima dari perujuk mana pun. Pastikan untuk menambahkan perujuk sebelum menggunakan kunci ini dalam produksi.

pkl.transdata.net.id/*	×
.pkl.transdata.net.id/	×
<input type="text" value="https://*.example.com/*"/>	

Catatan: Perlu waktu hingga 5 menit agar setelan berlaku

Simpan Batalan

Gambar 3.6 Pembatasan Aplikasi

- e. Selanjutnya yaitu mengaktifkan *Maps Javascript* API agar *map* dapat *loading* dengan baik. Langkahnya yaitu pilih menu **Dasbor**, lalu pilih **Aktifkan API dan Layanan**, pilih **Maps Javascript API** lalu klik **Aktifkan**. API sudah bisa digunakan.
2. Membuat tabel markers pada *database* dengan nama *db_transdata* di MySQL yang berisi *attribute* seperti *id*, *name*, *address*, *lat*, dan *lng*. *Lat* yaitu kependekan dari *latitude* atau yang biasa disebut garis lintang sedangkan *lng* yaitu kependekan dari *longitude* atau garis bujur. *Lat* dan

Ing disini mewakili koordinat pada sistem koordinat geografis dan pada tabel menggunakan *type float*. Berikut adalah tampilan tabel markers.

id	name	address	lat	lng	type
1	Perumahan Puri Hijau	Jalan Kyai Haji Wachid Hasyim Karangklesem	-7.442805	109.245773	Perumahan
2	Perumahan Taman Anggrek	Jalan Gerilya Purwokerto Selatan	-7.437994	109.246605	Perumahan
3	Perumahan Puri Indah	Windusara Karangklesem	-7.445628	109.242989	Perumahan aja
0	apa	ewq	12.000000	23.000000	wr

Gambar 3.7 Tabel Markers

- Mengeluarkan data XML dengan PHP. Bagian ini dilakukan untuk mengekspor data tabel dari *database* MySQL dalam format XML menggunakan pernyataan PHP. Menggunakan file XML sebagai perantara antara *database* dan Google Map agar dapat memuat halaman lebih cepat^[9]. Penulis membuat file php yang digunakan sebagai koneksi antara *database* dengan file php yang nanti akan dibuat untuk *output* XML. File tersebut penulis beri nama `phpsqlajax_dbinfo.php`, berikut isi dari filenya.

```

<?php |
$username = "";
$password = "";
$database = "db_transdata";
?>

```

Gambar 3.8 Isi File `phpsqlajax_dbinfo.php`

Selanjutnya untuk file *output* XML yang menggunakan *echo* PHP penulis beri nama `about-area.php` dimana isi file tersebut adalah sebagai berikut.

```
<?php require("phpsqlajax_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
$connection=mysqli_connect ('localhost', root, Moncer12354);
if (!$connection) {
    die('Not connected : ' . mysqli_connect_error());
}
// Set the active MySQL database
$db_selected = mysqli_select_db( $connection , "transdata");
if (!$db_selected) {
    die ('Can\'t use db : ' . mysqli_error());
}
// Select all the rows in the markers table
$result = mysqli_query($connection, "SELECT * FROM markers" );
if (!$result) {
    die('Invalid query: ' . mysqli_error());
}
header("Content-type: text/xml");
// Start XML file, echo parent node
echo "<?xml version='1.0'?>";
echo "<markers>";
$ind=0;
// Iterate through the rows, printing XML nodes for each
while ($row = @mysqli_fetch_assoc($result)){
    // Add to XML document node
    echo '<markers ' ;
    echo 'id="' . $row['id'] . "' ' ;
    echo 'name="' . parseToXML($row['name']) . "' ' ;
```



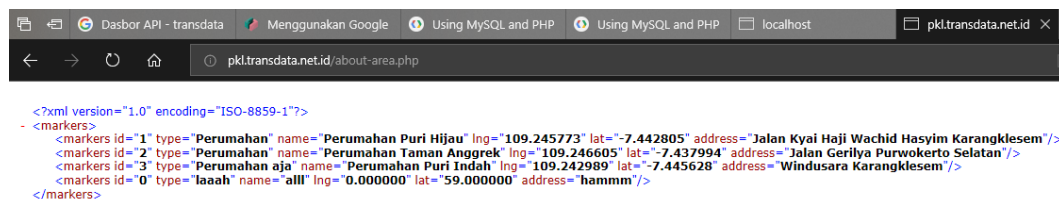
```

echo 'address="' . parseToXML($row['address']) . "'
';
    echo 'lat="' . $row['lat'] . "' ";
    echo 'lng="' . $row['lng'] . "' ";
    echo 'type="' . $row['type'] . "' ";
    echo '>';
    $ind = $ind + 1;
}
// End XML file
echo "</markers>";
?>

```

Kode di atas terhubung ke *database* dan mengeksekusi *query SELECT * (select all)* pada table *markers*. Kode menggunakan *parent node* dan mengulangi melalui hasil *query* serta menggunakan *node XML* untuk penanda di setiap baris table pada setiap lokasi. Informasi yang muncul nanti yaitu pada kolom *name* dan *address* melalui fungsi **parseToXML** terlebih dahulu.

Berikut adalah hasil setelah halaman *about-area.php* dijalankan di *browser*.



```

<?xml version="1.0" encoding="ISO-8859-1"?>
<markers>
<markers id="1" type="Perumahan" name="Perumahan Puri Hijau" lng="109.245773" lat="-7.442805" address="Jalan Kyai Haji Wachid Hasyim Karangklesem"/>
<markers id="2" type="Perumahan" name="Perumahan Taman Angrek" lng="109.246605" lat="-7.437994" address="Jalan Gerilya Purwokerto Selatan"/>
<markers id="3" type="Perumahan aja" name="Perumahan Puri Indah" lng="109.242989" lat="-7.445628" address="Windusara Karangklesem"/>
<markers id="0" type="laah" name="all" lng="0.000000" lat="59.000000" address="hamm"/>
</markers>

```

Gambar 3.9 Hasil *Output XML*

4. Selanjutnya yaitu membuat *map* pada halaman *area.php* dengan cara memanggil file yang telah dibuat. Berikut adalah perintah yang penulis gunakan pada halaman *area.php*.

a) Membuat kotak untuk *map* dan memanggil API *map* yang telah dibuat.

```

<div id="dvMap" style="width: 1000px; height: 550px"></div>
<script src="https://maps.googleapis.com/maps/api/js?
key=AlzaSyCpT7DH9B4XAsy_kK7C0VQyeE4qFAslv80" async
defer"></script>

```

- b) Menghubungkan tabel markers pada *database* dengan *map* yang kita buat.

```
<script type="text/javascript">
var markers = [
<?php
    $sql = mysqli_query($db, "SELECT * FROM markers");
    while(($data = mysqli_fetch_assoc($sql))) { ?>
    {
        "lat": '<?php echo $data['lat']; ?>',
        "lng": '<?php echo $data['lng']; ?>',
        "address": '<?php echo $data['address']; ?>' },
    <?php
    }
    ?>
    ];
</script>
```

- c) Menampilkan hasil program JavaScript Dimana pengaturan *center map* pada `LatLng(-7.44, 109.24)` yang merupakan letak geografis Purwokerto dengan *zoom* sebesar 12.

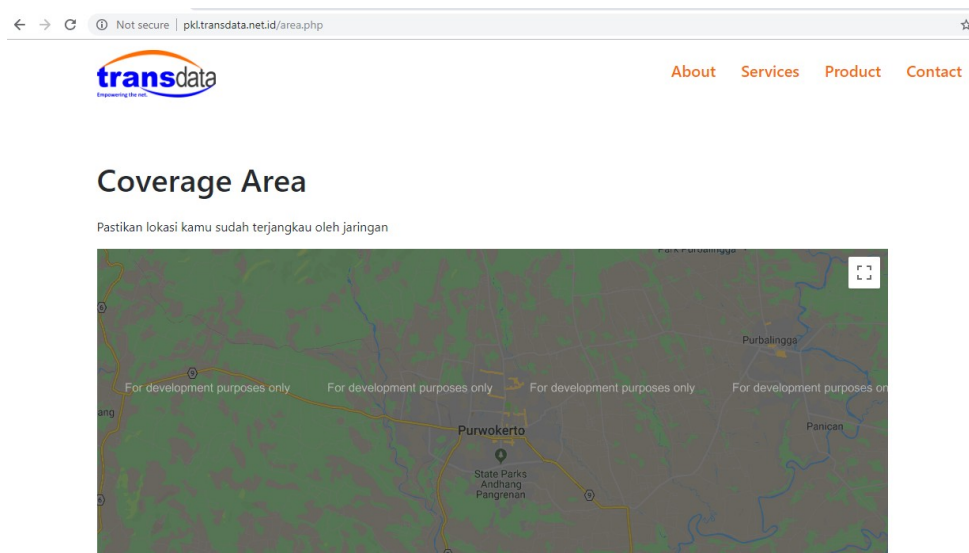
```
window.onload = function () {
    var mapOptions = {center: new google.maps.LatLng(-7.44,
    109.24), zoom: 12, mapTypeId:google.maps.MapTypeId.ROADMAP
    };
};
```

- d) Menampilkan *marker* dan info dari lokasi yang telah di *input* kan pada tabel markers. Fungsi yang digunakan yaitu **infoWindow** dimana biasanya digunakan untuk menampilkan konten(teks atau gambar) di jendela *popup* di atas *map*.

```
var infoWindow = new google.maps.InfoWindow();
var map = new google.maps.Map(document.getElementById("dvMap"),
mapOptions);
for (i = 0; i < markers.length; i++) {
    var data = markers[i];
        var latnya = data.lat;
        var lngnya = data.lng;
        var myLatLng = new google.maps.LatLng(latnya, lngnya);
        var marker = new google.maps.Marker({position: myLatLng, map:
        map, title: data.address});
```

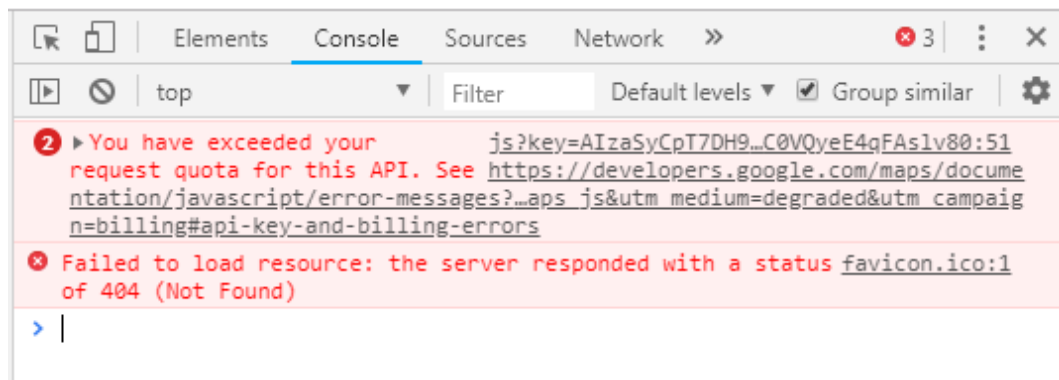
```
(function (marker, data) {  
    google.maps.event.addListener(marker, "click", function (e) {  
    infoWindow.setContent('<b>Lokasi</b> :' + data.address + '<br>');  
    infoWindow.open(map, marker);  
    });  
})(marker, data);
```

Tampilan halaman `are.php` setelah di buka pada *web server* Transdata adalah sebagai berikut.



Gambar 3.10 Tampilan `area.php`

Penulis mengecek *error* dengan menekan tombol **Ctrl+Shift+I** (*Inspect*) lalu muncul *error* seperti gambar di bawah ini.



Gambar 3.11 *Error area.php*

Setelah penulis me-klik URL yang ada, *error* tersebut disebabkan karena kita harus mengaktifkan penagihan pada akun penulis serta harus menyertakan kunci API yang valid agar *map* dapat digunakan.