ISSN 2085-4552

Implementasi Geofencing dalam Mengawasi Pengiriman Kendaraan di Sebuah Perusahaan Ekspedisi

Joko Priono[1], Eko Budi Setiawan[2]
Program Studi Teknik Informatika, Universitas Komputer Indonesia. Jl. Dipatiukur 112 Bandung
joko.priono@yahoo.com
eko@email.unikom.ac.id

Diterima 21 Oktober 2017
Disetujui 20 Desember 2017

***Abstract* — This paper describe a geofencing approach in monitoring the routes used in the delivery of vehicles in a expedition company. Continuous technological development of information systems provide a variety approach on how to monitor a vehicle on the road. One of those approach is geofencing. Geofencing itself is an innovative technology that enables remote monitoring of a specific geographic areas. It can make a virtual area called geofence on a virtual map with different size and shape. This virtual area would be able to tracked mobile objects that enter or leave these area, and even how long the objects inside the area. The mobile objects that are going to be tracked are object that have GPS Tracking Unit. GPS Tracking Unit is an application or a tool that is used to figure out the position of an object. The position are based on geographical coordinate, latitude and longitude, and the data is received through a Global Positioning System. The implementation of the geofencing are going to use library from Google API. The combination of GPS Tracking and geofencing would be able to track, prevent or even restrict on specific routes that are going to be used in vehicle delivery.**

Index Terms— Pengawasan, GPS Tracker, Geofence area, Route information, Google API

I. PENDAHULUAN

Penelitian ini dilakukan pada sebuah perusahaan swasta yang mempunyai fokus bisnis dalam melayani jasa pengiriman mobil, alat berat dan project *heavy cargo* ke hampir seluruh wilayah di Indonesia. Di masa perkembangan ekonomi yang terus meningkat, perusahaan terus mengembangkan usahanya menggunakan teknologi yang tersedia. Salah satu pengembangan yang dilakukan adalah pada *service* yang ditawarkan oleh perusahaan yaitu *door-to-door* yang merupakan pengiriman kendaraan dengan menjemput dan mengantarkan kendaraan ke hampir seluruh daerah di Indonesia. Pengiriman ini membutuhkan suatu manajemen operasi yang bertugas untuk memonitor pengiriman tersebut. Sistem dalam memanajemen pengiriman kendaraan tentu menjadi hal yang sangat penting untuk menjamin kelancaran dari pengiriman.

Sistem yang ada saat ini juga tidak memiliki informasi atau data tentang jalur pengiriman kendaraan hanya tujuan saja, sehingga sulit untuk melakukan perencanaan rute. Hal ini berpengaruh dalam kelancaran pengiriman kendaraan dan untuk mengetahui rute yang dianggap bahaya untuk dilalui kendaraan. Informasi yang bisa dapat saat ini hanya dengan berbicara dengan supir yang telah melakukan pengiriman.

Berdasarkan wawancara dan observasi yang dilakukan ditempat penelitian, didapatkan beberapa permasalahan sebagai berikut:

a. Sulitnya pihak petugas dalam memonitor posisi dan jalur apa yang digunakan supir dalam melakukan pengiriman kendaraan. Hal ini dikarenakan proses yang berjalan saat ini memiliki proses pengawasan yang minimalis, yaitu melalui telepon, dan hal ini hanya dilakukan jika diperlukan. Selain itu, tidak ada cara untuk mengetahui posisi supir secara berkala di lapangan. Dengan menggunakan suatu tracking GPS dan penggunaan *geofencing*, maka akan membantu mengetahui posisi supir secara berkala serta jalur supir yang telah ditentukan dengan area *geofencing* yang telah ditentukan sebelumnya.

b. Sulitnya mendapatkan informasi tentang pengiriman kendaraan yang dapat membantu perusahaan dalam melakukan perencanaan rute pengiriman yang lebih baik. Saat ini informasi yang didapat hanya dari supir yang sudah melakukan pengiriman dan informasi itu sendiri tidak terlalu komplit. Untuk membantu meningkatkan data atau informasi yang dapat membantu perencanaan rute, maka data dari *gps tracker* dan geofencing akan disimpan dan digunakan untuk membantu petugas.

Geofences merupakan teknologi inovatif yang yang memanfaatkan koordinat geografis dunia nyata dengan menentukan batasan atau parameter secara virtual. *Geofencing* sendiri memungkinkan seseorang untuk

memonitor secara *remote* dari suatu wilayah geografis tertentu.

Jika parameter dari *geofences* ini aktif, status aktifitas akan secara otomatis tersimpan atau terdata [1]. Seperti saat kendaraan masuk atau keluar dari parameter yang telah ditentukan, berapa lama kendaraan berada dalam suatu parameter *geofence*, mengukur kecepatan pengendaraan dalam suatu parameter, atau memberi informasi jika kendaraan dalam suatu jalan atau wilayah tertentu. Hal ini akan sangat membantu terutama untuk perusahaan dibidang logistik dalam mengawasi pengiriman kendaraan dan pengendaranya di berbagai lokasi.

Reclus dan Drouard [1] telah menyajikan penelitian mengenai konsep dasar *geofencing* dan beberapa aplikasi berdasarkan teknologi ini di dalam manajemen sektor transportasi dan logistik. Beberapa aplikasi berdasarkan *geofencing* yang diajukan penelitian ini adalah pengawasan lokasi kendaraan berdasarkan *Point of Interest* (POI), *Fleet Management*, dan *Defense & Security*. Pada penelitian Nait-Sidi-Moh [2], memperkenalkan konsep sistem atau *platform* yang mengintegrasikan teknik geofencing untuk pelacakan secara *real-time* dengan perangkat *mobile*. Pada penelitian Salim [3], membahas tentang implementasi *web-based vehicle tracking system* yang menggunakan *HTTP protocol* dalam penyimpanan data GPS pada *database server* dan menampilkan data posisi pada *Google Maps*.

Dengan demikian, aplikasi pengawasan yang akan dibangun dalam penelitian ini diharapkan dapat mempermudah pengambilan data lokasi kendaraan dan membantu memberikan informasi rute perjalanan, sehingga dapat meningkatkan efisiensi kerja yang berdampak pada penghematan pengeluaran dan pemberian informasi *up-to-date* kendaraan.

II. TINJAUAN PUSTAKA

A. *Monitoring*

Monitoring atau pengawasan dapat didefinisikan sebagai siklus kegiatan yang mencakup pengumpulan, peninjauan ulang, pelaporan dan tindakan atas informasi suatu proses yang sedang diimplementasikan [4]. Kegiatan pengawasan lebih terfokus pada pengawasan kegiatan yang sedang dilaksanakan.

Pengawasan dilakukan dengan cara memperoleh informasi secara regular berdasarkan indikator tertentu, untuk mengetahui apakah kegiatan yang sedang berlangsung sesuai dengan rencana dan prosedur yang telah ditentukan.

B. *GPS (Global Positioning System)*

Global Positioning System merupakan teknologi penentu lokasi yang banyak digunakan saat ini. GPS adalah suatu sistem radio navigasi penentuan lokasi menggunakan satelit [5]. Dengan bantuan satelit, akan diperoleh posisi yang akurat dan cepat dengan koordinat 3 dimensi (x,y,z) ditambah dengan informasi waktu dan kecepatan bergerak. Posisi unit GPS akan ditentukan berdasarkan titik-titik koordinat *latitude* dan *longitude* yang diperolehnya dari nilai derajat dari suatu titik yang diukur.

Penggunaan GPS akan membantu menemukan letak koordinat *latitude* dan *longitude* dari target. GPS yang digunakan akan memanfaatkan GPS yang terdapat pada smartphone yang akan diaktifkan pada saat target dijalan [6]. Perkembangan penelitian lebih lanjut dapat menggunakan GPS Tracker khusus yang dapat dipasang ke kendaraan. Koordinat dari GPS nantinya akan dikirimkan ke server untuk membantu mendeteksi *geofencing*.

C. *GPS Tracker*

Perangkat pelacak GPS (*GPS Tracker*) merupakan alat yang digunakan sebagai penentu lokasi dari target yang dibawa oleh suatu kendaraan atau seseorang. Data lokasi pada perangkat pelacak GPS merupakan koordinat geografis yang dikirimkan ke *server* yang merupakan bagian dari komponen sistem pelacak. Pengiriman data lokasi akan menggunakan jaringan seluler (GPRS atau SMS), radio atau modem satelit yang tertanam dalam perangkat yang digunakan.

Gambar 1. *GPS Tracker*

Perangkat ini memberikan lokasi aset unit yang dapat ditampilkan dengan latar belakang peta baik secara *real time*. GPS Tracker yang akan digunakan merupakan aplikasi android yang akan dirancang untuk mengirimkan data GPS berupa koordinat *latitude* dan *longitude* ke server secara *real time* dengan konektifitas internet [7].

D. *Geofencing*

Geofencing merupakan perangkat lunak yang digunakan bersamaan dengan *global positioning system* (GPS) dalam menentukan batas-batas geografis atau parameter virtual dari suatu peta. Program yang menggunakan *geofencing* dapat mengatur suatu triggers yang dapat memberikan informasi atau notifikasi apabila suatu target tertentu masuk atau keluar dari suatu batasan yang telah ditetapkan sebelumnya. Beberapa teknik dari geofencing adalah *Geofence Area, Proximity with Point of Interest, Route adherence*, dan *Route and schedule adherence* [1].

Secara garis besar, koordinat geografis digunakan untuk mengetahui posisi target dan juga untuk membuat suatu batasan daerah tertentu (*mapping*) sebagai pagar virtual (*geofence*) suatu daerah [1][8]. Sistem akan menentukan posisi target yang dilacak berada di luar atau di dalam wilayah geofencing.

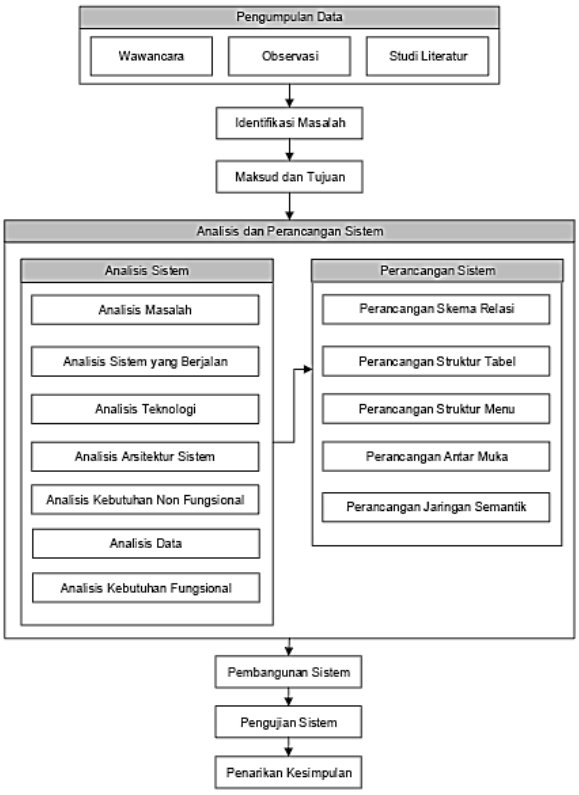
Teknologi ini juga dapat memungkinkan untuk pendeteksian kedekatan antara posisi target dengan area geofencing tertentu.

Gambar 2. *Geofencing*

Penggunaan *geofencing* dalam mengawasi pengiriman kendaraan akan dilakukan dengan cara menggambarkan area-area virtual pada suatu lokasi-lokasi tertentu, seperti gerbang tol, jalan ataupun suatu kota, sehingga semua jalur penting yang dilewati dapat di monitor.

III. METODOLOGI PENELITIAN

Metodologi penelitian yang digunakan dalam penelitian ini yaitu metode penelitian deskriptif. Metode ini bertujuan untuk membuat gambaran secara sistematis dan faktual tentang fakta dan perilaku dari objek yang diteliti. Adapun tahapan penelitian yang dilakukan pada penelitian ini dapat dilihat seperti gambar 3 berikut.



Gambar 3 Tahapan Penelitian

IV. HASIL DAN PEMBAHASAN

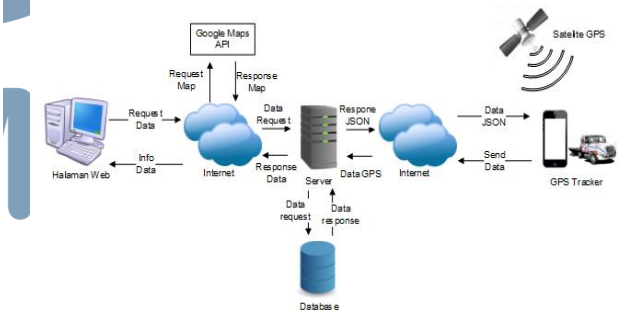
Tahapan ini terdiri dari gambaran umum sistem, analisis dan perancangan sistem yang akan dijelaskan dalam bentuk arsitektur sistem, implementasi teknologi yang terdiri dari analisis teknologi sensor GPS dan teknologi *geofencing*, dan analisis area geofencing. Langkah terakhir yang dilakukan yaitu melakukan pengujian sehingga dapat diperoleh kesimpulan.

A. Gambaran Umum

Secara umum akan terdapat dua sistem yang dibangun. Sub-sistem mobile dan sub-sistem web. Sistem mobile akan menggunakan *smartphone* android yang mempunyai fasilitas GPS dan koneksi internet. Sub-sistem ini digunakan untuk mengirimkan lokasi berupa koordinat *latitude* dan *longitude* ke database server secara berkala. Sub-sistem web akan digunakan untuk menampilkan data koordinat pada peta google maps. Peta tersebut akan berisi area *geofencing* yang telah digambarkan. Saat koordinat dari *smartphone* melewati area *geofencing*, maka akan muncul suatu notifikasi pada sistem.

B. Analisis dan Perancangan Sistem

Sistem yang akan dibangun yaitu aplikasi mobile berbasis android sebagai *front-end* dan aplikasi web sebagai *back-end*. Arsitektur Sistem yang akan dibangun dapat dilihat pada Gambar 4 berikut ini.



Gambar 4. Konsep Arsitektur Sistem

Sistem mobile akan menjadi *receiver* dari satelit GPS dan akan mengirimkan data *latitude* dan *longitude* yang akan disimpan dalam database server. Sistem web akan digunakan untuk menampilkan data dari koordinat yang telah dikirimkan dan melihat apakah koordinat tersebut berada di dalam area *geofencing* atau tidak. Area *geofencing* sendiri akan dibuat melalui sistem web dan akan menggunakan *library* yang disediakan oleh Google APIs.

C. Implementasi Teknologi Sensor GPS

GPS Tracking yang akan digunakan merupakan aplikasi android yang dirancang untuk mengirimkan data koordinat *latitude* dan *longitude* ke server dengan konektifitas internet.

Aplikasi akan menggunakan *Google Play service location APIs* yang memberikan fasilitas seperti *location awareness* pada *app* yang dibangun. Untuk membangun suatu aplikasi menggunakan *Google Play Service APIs*, diperlukannya kode *dependencies* dari *Google Play service SDK* seperti di Gambar 5.

```
apply plugin: 'com.android.application'
    ...
    dependencies {
        compile 'com.google.android.gms:play-services-location:10.2.4'
    }
```

Gambar 5. *Dependencies Code*

Untuk menggunakan *Google Play Services*, aplikasi harus menginisialisasi *GoogleApiClient*. Inisialisasi dilakukan di *onCreate()* pada *Activity* sesuai dengan kode pada Gambar 6 dibawah.

```
mGoogleApiClient = new GoogleApiClient.Builder(this)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .addApi(LocationServices.API).build();
```

Gambar 6. *Google Api Client Code*

Hal lain yang perlu diperhatikan dalam menentukan lokasi aplikasi selain konfigurasi awal diatas adalah parameter *request* yang digunakan untuk mendeterminasikan kondisi dari permintaan lokasi. Gambar 7 dibawah merupakan parameter yang akan digunakan *LocationRequest*.

```
protected void createLocationRequest() {
  mLocationRequest = new LocationRequest();
  mLocationRequest.setInterval(10000);
  mLocationRequest.setFastestInterval(5000);
  mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
  mLocationRequest.setSmallestDisplacement(10);
```

Gambar 7. *Location Request Code*

Berikut adalah beberapa penjelasan parameter yang akan dipakai:

a) *Update Interval*
setInterval() – parameter ini menetapkan interval dalam mili-detik dimana aplikasi memilih untuk menerima update lokasi terbaru.

b) *Fastest Update Interval*
setFastestInterval() – parameter ini melihat tingkat tercepat dalam mili-detik saat aplikasi menangani update lokasi terbaru.

c) *Displacement Update*
setSmallestDisplacement() – parameter ini menetapkan jarak minimum antara lokasi dalam menangani update dalam ukuran meter.

d) *Priority*
setPriority() – parameter ini menetapkan prioritas request.

Setelah terkoneksi dengan *GoogleApiClient* dan parameter dari *request* telah ditetapkan, maka aplikasi akan dapat memulai permintaan menggunakan *requestlocationUpdate*() dengan *FusedLocationApi*.

```
LocationServices.FusedLocationApi.requestLocationUpdates(
      GoogleApiClient, LocationRequest, LocationListener);
```

Gambar 8. *Fused Location Api Request Code*

Untuk memberhentikan permintaan update lokasi, maka digunakan *removeLocationUpdate*().

```
LocationServices.FusedLocationApi.removeLocationUpdates(
      GoogleApiClient, LocationListener);
```

Gambar 9. *Fused Location Api Remove Code*

Untuk mendapatkan isi dari lokasi yang diminta, akan digunakan kode *getLastLocation*().

```
private void displayLocation() {
 Location mLastLocation = LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient);
 String latitude = String.valueOf(mLastLocation.getLatitude());
 String longitude = String.valueOf(mLastLocation.getLongitude());
 float currentspeed = mLastLocation.getSpeed();
}
```

Gambar 10. *Get Last Location Code*

D. Implementasi Teknologi *Geofencing*

Geofencing digunakan untuk menentukan batas-batas geografis virtual dari suatu peta. Bentuk daerah *geofencing* sendiri bisa berbeda-beda. Pada *Google Maps APIs*, penggambaran geofencing menggunakan *Library* yang disediakan oleh *JavaScript API* bernama *Drawing Layer*. *Drawing Layer* merupakan kelas yang menyediakan antarmuka grafis untuk user dalam menggambar *circles*, *polygons*, *rectangles*, *polylines* dan *markers* pada *map*. Contoh *Drawing Layer* dapat dilihat pada Gambar 11.

Gambar 11. *Drawing Layer*

Untuk menggunakan fungsionalitas *Library*, aplikasi harus memuat terlebih dahulu kode izin penggunaan *Google Maps API* seperti pada Gambar 12.

```
<script type="text/javascript"
  src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&libraries=drawing">
</script>
```

Gambar 12. *Google Maps Permission Code*

Setelah izin ditambahkan, maka *DrawingManager* akan dapat dipanggil dengan menggunakan kode seperti pada Gambar 13.

Penggunaan *DrawingManager* memiliki opsi yang akan mendefinisikan beberapa kontrol yang akan ditampilkan, seperti posisi kontrol dan keadaan gambar pada saat inisialisasi kode.

```
1.  var drawingManager = new google.maps.drawi
    ng.DrawingManager();
2.  drawingManager.setMap(map);
3.  var drawingManager = new google.maps.drawi
        ng.DrawingManager({
4.    drawingMode: google.maps.drawing.Overlay
    Type.MARKER,
5.    drawingControl: true,
6.    drawingControlOptions: {
7.      position: google.maps.ControlPosition.
    TOP_CENTER,
8.      drawingModes: ['marker', 'circle', 'po
    lygon', 'polyline', 'rectangle']
9.    },
10.   circleOptions: {
11.     fillColor: '#ffff00',
12.     fillOpacity: 1,
13.     strokeWeight: 5,
14.     clickable: false,
15.   }
16. });
```

Gambar 13. *Drawing Manager Code*

Berikut adalah beberapa penjelasan dari *drawing manager* yang dipakai:

a) *drawingMode* – bagian ini mendefinisikan keadaan kursor awal pada saat inisialisasi.

b) *drawingControl* – bagian ini mendefinisikan visibilitas alat yang digunakan untuk penggambaran geofence pada pilihan antarmuka.

c) *drawingControlOption* – mendefinisikan posisi dari kontrol pada google map dengan position dan tipe overlay yang akan tersedia dengan drawingModes.

d) Setiap jenis *overlay* dapat ditetapkan suatu properti yang akan mendefinisikan hasil penggambaran dari *overlay*. Isi dari properti salah satunya adalah warna gambar, transparansi, ukuran ketebalan garis, dan lainnya.

E. Analisis Area Geofencing

Penggambaran *geofencing* pada peta aplikasi mempunyai beberapa bentuk, seperti lingkaran, persegi dan poligon. Bentuk penggambaran ini merupakan objek pada peta yang terikat dengan koordinat *latitude* dan *longitude*.

Bentuk penggambaran dapat juga dikonfigurasikan agar user dapat mengedit atau menyeret bentuk gambar tersebut. Gambar 14 dan Gambar 15 adalah contoh dan kode penggambaran polygon segitiga pada *google maps.*

Gambar 14. *Simple Polygon*

```
1.  function initMap() {
2.    var map = new google.maps.Map(document.ge
    tElementById('map'), {
3.      zoom: 5,
4.      center: {lat: 24.886, lng: -70.268},
5.      mapTypeId: 'terrain'
6.    });
7.    // Define polygon's path.
8.    var triangleCoords = [
9.      {lat: 25.774, lng: -80.190},
10.     {lat: 18.466, lng: -66.118},
11.     {lat: 32.321, lng: -64.757},
12.     {lat: 25.774, lng: -80.190}
13.   ];
14.   // Construct the polygon.
15.   var bermudaTriangle = new google.maps.Pol
    ygon({
16.     paths: triangleCoords,
17.     strokeColor: '#FF0000',
18.     strokeOpacity: 0.8,
19.     strokeWeight: 2,
20.     fillColor: '#FF0000',
21.     fillOpacity: 0.35
22.   });
23.   bermudaTriangle.setMap(map);
24. }
```

Gambar 15. *Drawing Polygon Code*

Gambar 16 dan Gambar 17 berikut adalah contoh dan kode penggambaran persegi pada *google maps*.

Gambar 16. *Rectangles*

```
1.  function initMap() {
2.    var map = new google.maps.Map(document.ge
    tElementById('map'), {
3.      zoom: 11,
4.      center: {lat: 33.678, lng: -116.243},
5.      mapTypeId: 'terrain'
6.    });
7.    var rectangle = new google.maps.Rectangle
    ({
8.      strokeColor: '#FF0000',
9.      strokeOpacity: 0.8,
10.     strokeWeight: 2,
11.     fillColor: '#FF0000',
12.     fillOpacity: 0.35,
13.     map: map,
14.     bounds: {
15.       north: 33.685,
16.       south: 33.671,
17.       east: -116.234,
18.       west: -116.251
19.     }
20.   });
21. }
```

Gambar 17. *Drawing Rectangle Code*

Gambar 18 dan Gambar 19 berikut adalah contoh dan kode penggambaran lingkaran pada *google maps*.

Gambar 18. *Circle Google Maps*

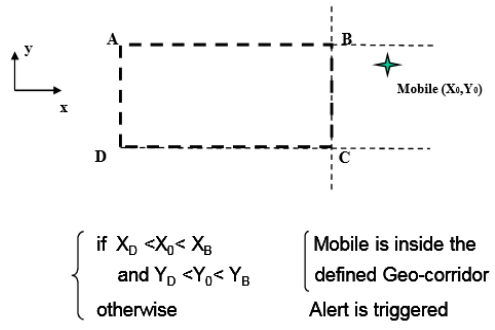
Tiap penggambaran area *geofence* memiliki atribut yang sama dengan sedikit karakteristik berbeda. Misalnya untuk penggambaran *Circle* menggunakan *google.maps.Circle* dengan beberapa *atribut* yang dapat diubah yaitu *color*, *opacity*, *weight* dan lainnya.

```
1.  var citymap = {
2.    chicago: {
3.      center: {lat: 41.878, lng: -
    87.629},population: 2714856}
4.  };
5.  function initMap() {  // Create the map.
6.    var map = new
    google.maps.Map(document.getElementById('ma
    p'), {
7.      zoom: 4, center: {lat: 37.090, lng: -
    95.712},
8.      mapTypeId: 'terrain'
9.    });
10.   for (var city in citymap) {
11.     // Add the circle for the map.
12.     var cityCircle = new
    google.maps.Circle({
13.       strokeColor: '#FF0000',
14.       map: map,
15.       center: citymap[city].center,
16.       radius:
    Math.sqrt(citymap[city].population) * 100
17.     });
18.   }}
```

Gambar 19. *Drawing Circle Code*

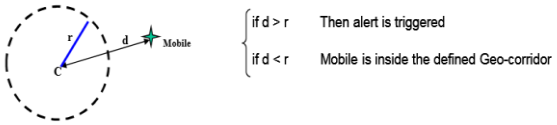
F. *Analisis Pendeteksian Geofencing*

Analisis pendeteksian *Geofencing* merupakan analisis cara sistem mengetahui koordinat lokasi yang dikirim oleh *GPS Tracker* sudah masuk ke dalam area *geofencing*. Tiap pendeteksian *geofencing* berbeda-beda, dari area lingkaran, area persegi dan area polygon. Untuk pendeteksian persegi akan menggunakan metode seperti pada Gambar 20 dibawah ini.



Gambar 20. Perhitungan Persegi

Perhitungan *geofencing* persegi dilakukan dengan melihat apakah koordinat target berada di dalam 4 poin koordinat yang menggambarkan persegi pada peta geografis. Jika benar, maka sistem akan melakukan *trigger* yang memberitahukan bahwa target berada di dalam area *geofence*. Untuk pendeteksian lingkaran digunakan metode seperti pada Gambar 21.



Gambar 21. Perhitungan Lingkaran

Perhitungan *geofencing* pada lingkaran melihat koordinat posisi target dan posisi koordinat tujuan. Jika jarak antara koordinat target dan tujuan lebih besar dari radius lingkaran, berarti target berada diluar lingkaran. Jika nilai lebih kecil, maka target berada di dalam lingkaran dan sistem akan melakukan pemberitahuan tentang informasi tersebut.

Karena geografis bumi tidak berbentuk lingkaran, melainkan berbentuk bola. Maka perhitungan jarak tidak berada pada permukaan datar. Untuk itu diperlukan perhitungan yang mendeterminasikan jarak antar dua point pada suatu permukaan bola berdasarkan *latitude* dan *longitude*. Perhitungan yang akan digunakan adalah *haversine formula*.

```
1.  function haversineGreatCircleDistance(
2.    $latitudeFrom, $longitudeFrom, $latitudeT
    o, $longitudeTo, $earthRadius = 6371000)
3.  {
4.    // convert from degrees to radians
5.    $latFrom = deg2rad($latitudeFrom);
6.    $lonFrom = deg2rad($longitudeFrom);
7.    $latTo = deg2rad($latitudeTo);
8.    $lonTo = deg2rad($longitudeTo);
9.
10.   $latDelta = $latTo - $latFrom;
11.   $lonDelta = $lonTo - $lonFrom;
12.
13.   $angle = 2 * asin(sqrt(pow(sin($latDelta
    / 2), 2) +
14.     cos($latFrom) * cos($latTo) * pow(sin($
    lonDelta / 2), 2)));
15.   return $angle * $earthRadius;
16. }
```

Gambar 22. *Haversine Formula Code*

G. *Implementasi Sistem*

Adapun beberapa pembahasan implementasi yang dilakukan antara lain adalah menerapkan hasil analisis dan perancangan dalam implementasi perangkat keras, implementasi perangkat lunak dan implementasi antarmuka.

1) *Implementasi Perangkat Keras*

Perangkat keras yang digunakan dalam mengimplementasikan aplikasi pengawasan

pengiriman kendaraan terbagi menjadi 3, perangkat keras server, pengguna dan *smartphone*. Berikut adalah perangkat keras untuk kebutuhan sistem.

Tabel 1. Implementasi Perangkat Keras

No.	Perangkat Keras	Spesifikasi
1.	Server Processor	Intel(R) Xeon(R) CPU 5140 @ 2.33GHz
2.	Server RAM	2 GB
3.	Server Hardisk	20 GB
4.	Server LAN Card	10/100Mb
5.	User Processor	Intel Core i5
6.	User RAM	4 GB
7.	User Monitor	Standar
8.	User Hardisk	500 GB
9.	Android Processor	1.6GHz dual-core Intel Atom Z2560
10.	Android RAM	2 GB
11.	GPS Module	GLONASS
12.	GSM Card	Telkomsel 3G

2) Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk mengimplementasikan aplikasi ini terbagi menjadi tiga tipe, yaitu implementasi perangkat lunak pada server, pengguna dan perangkat lunak pada *smartphone*. Untuk selengkapnya dapat dilihat pada Tabel 2 berikut.

Tabel 2. Implementasi Perangkat Lunak

No.	Perangkat Lunak	Spesifikasi
1.	OS Server	Linux CentOS 6.8
2.	Webserver	Apache Webserver
3.	Database Server	MySQL Database
4.	Control Panel	Webmin 1.821
5.	OS User	Windows
6.	Web Browser	Mozilla Firefox
7.	Kode Editor	Sublime Text
8.	IDE	Android Studio
9.	OS Smartphone	Android 4.2 Jelly Bean

3) Implementasi Antarmuka

Antarmuka merupakan tampilan dari hasil penerapan aplikasi pengawasan pengiriman kendaraan. Implementasi antarmuka yang dibangun dapat dilihat pada gambar dibawah ini.

a) Antarmuka Aplikasi Mobile

Tampilan antarmuka aplikasi *GPS Tracker* pada smartphone yang digunakan dalam mengirimkan data lokasi koordinat GPS. Menu berisi *login* berdasarkan *username* dan *password*, serta tombol aktivasi GPS *Tracker*.

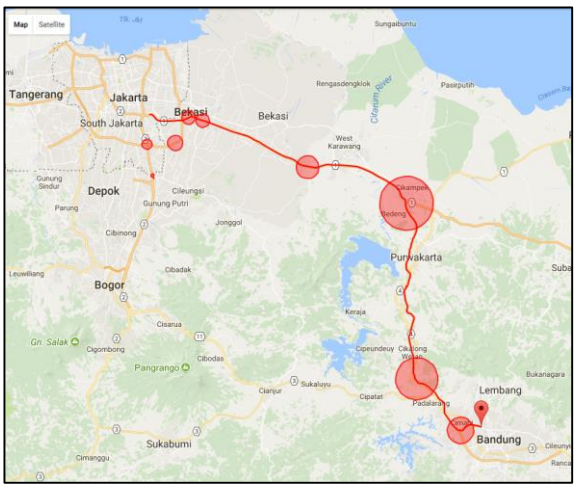
Gambar 23. Tampilan Antarmuka Aplikasi *Mobile*

b) Antarmuka website

Tampilan antarmuka website yang merupakan program *back-end* dari aplikasi pengawasan pengiriman kendaraan.

Gambar 24. Tampilan Antarmuka Pengawasan

Tampilan antarmuka website yang menampilkan jalur atau rute yang melalui area *geofencing* dari aplikasi pengawasan pengiriman kendaraan.



Gambar 25. Tampilan Antarmuka Detail Pengawasan

Tampilan antarmuka website menampilkan fitur-fitur yang dibutuhkan dalam melihat jalur perjalanan kendaraan.

H. *Pengujian Sistem*

Pengujian bertujuan untuk menjamin bahwa perangkat lunak yang dibangun memiliki kriteria yang teruji dan dapat berjalan dengan baik tanpa mengalami gangguan atau *error*. Pengujian ini dilakukan khususnya pada pengujian fungsionalitas sistem serta kinerja sistem yang telah dibangun. Pengujian dilakukan dengan teknik pengujian *black box* yang lebih berfokus pada menemukan kesalahan program secara fungsional.

Pengujian *black box* dilakukan untuk dua tahap, pengujian sub-sistem mobile dan sub-sistem web. Sub-sistem web, pengujian yang dilakukan adalah untuk komponen fungsional *login*, pembuatan *geofencing*, penampilan posisi pada *google map*, penginputan data, pengubahan data, dan fungsional pendeteksian area *geofencing*. Untuk pengujian *mobile*, dilihat dari aktivasi aplikasi apakah sudah dapat mengirimkan data posisi koordinat secara berkala ke server. Dari hasil pengujain *black box* dapat disimpulkan bahwa aplikasi ini sudah berhasil dapat berjalan sesuai dengan yang diharapkan.

Tabel 3. Pengujian Beta

Q1 : Aplikasi sudah memberikan informasi posisi kendaraan?				
SS	ST	RG	TS	STS
1	2	0	0	0
Rata-rata = (5+8+0+0+0) / 3 = 4.33				
Q2 : Aplikasi sudah membantu me-*monitor* rute yang dilewati menggunakan *geofencing*?				
SS	ST	RG	TS	STS
0	3	0	0	0
Rata-rata = (0+12+0+0+0) / 3 = 4				
Q3 : Aplikasi mudah untuk digunakan?				
SS	ST	RG	TS	STS
0	3	0	0	0
Rata-rata = (0+12+0+0+0) / 3 = 4				
Rata-rata Akhir = (4.33+4+4) / 3 = 4.11				

Selain itu, pengujian beta juga dilakukan pada bulan Juli 2017 dengan menyebarkan kuesioner terhadap responden yang merupakan petugas operasional pada perusahaan ekspedisi kendaraan di Jakarta. Responden tersebut memberikan jawaban yang berbeda, namun berdasarkan hasil perhitungan dari nilai rata-rata akhir yang ada dapat ditarik kesimpulan bahwa sekitar 82.2% responden setuju implementasi *geofencing* dalam pengawasan pengiriman kendaraan membantu dalam memudahkan pengoperasian untuk mengetahui posisi dan rute pengiriman kendaraan.

V. KESIMPULAN DAN SARAN

Adapun kesimpulan dan saran yang didapatkan dari hasil penelitian ini yaitu:

A. *Kesimpulan*

Berdasarkan hasil dari penelitian, analisis, perancangan, implementasi, serta pengujian, maka didapat kesimpulan sebagai berikut:

1. Implementasi *geofencing* dapat membantu memberikan informasi posisi dari suatu kendaraan.
2. Implementasi *geofencing* dapat mendeteksi rute apa saja yang dilewati saat pengiriman kendaraan.

B. *Saran*

Implementasi *geofencing* dalam pengawasan rute pengiriman kendaraan masih jauh dari sempurna dan masih memiliki kekurangan. Adapun beberapa saran yang mungkin dapat diterapkan untuk pengembangan selanjutnya adalah:

1. Pemanfaatan *geofencing* selain menentukan rute pengiriman kendaraan, seperti dalam hal *security* atau *POI*.
2. Penggabungan *Google APIs* lain *seperti Road API* atau *Direction API* untuk meningkatkan kualitas dari sistem.
3. Penggunaan *geofencing* dalam menentukan kecepatan kendaraan pada area-area tertentu.

DAFTAR PUSTAKA

[1] F. Reclus and K. Drouard, "Geofencing for Fleet & Freight Management," *Geofencing for Fleet & Freight Management*, pp. 353-356, 2009.

[2] A. Nait-Sidi-Moh, "On the Use of Location-Based Services and Geofencing Concepts for Safety and Road Transport Efficiency," *Trends in Mobile Web Information Systems*, pp. 135-144, 2013.

[3] K. A. Salim, "Design and Implementation of Web-Based GPS-GPRS Vehicle Tracking System," *IJCSET* vol. 3, no. 12, pp. 443–448, 2013.

[4] M. Corps, *Design, monitoring, and evaluation guidebook*, 2005.

[5] R. T. Setiadi and W. Pujiyono, "Spektrum Industri," *Sistem Pelacak Kendaraan Berbasis OpenGTS*, vol. 11, no. 2, pp. 117-242, 2013..

[6] A. Küpper, U. Bareth and B. Freese, "Geofencing and Background Tracking – The Next Features," 2011.

[7] H. Pranjoto and L. Agustine, "GPS Based Vehicle Tracking Over GPRS for Fleet Management and Passenger/ Payload/ Vehicle Security," *Journal of Engineering and Applied Sciences*, vol. 9 no. 11, 2014.

[8] D. Namiot and M. Sneps-Sneppe, "Geofence and Network Proximity," *Lecture Notes in Computer Science*, vol. 8 121, 2013.