PEMBANGUNAN APLIKASI *GEOFENCING* DALAM *MONITORING* PENGIRIMAN KENDARAAN PADA PT. CANNONEX INDONESIA

Joko Priono [1], Eko Budi Setiawan [2]

[1,2] Program Studi Teknik Informatika, Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung
E-mail : joko.priono@yahoo.com [1], eko@email.unikom.ac.id [2]

ABSTRAK

Penelitian ini membahas tentang penerapan metode *geofencing* dalam kasus me-*monitoring* suatu pengiriman kendaraan. *Geofencing* sendiri merupakan suatu teknologi yang memudahkan pengguna atau sistem dalam melakukan pengawasan jarak jauh terhadap suatu area yang telah ditentukan. *Geofence* merupakan area virtual yang dibuat berdasarkan koordinat geografis. Pembuatan geofencing ini akan menggunakan *Google Maps APIs*. Untuk mengaktifasi geofence, dibutuhkan suatu trigger. *Trigger* ini merupakan alat *GPS Tracking Unit* yang memberikan posisi koordinat dari suatu target. Dengan menggunakan dua metode ini, maka sistem akan dapat mengetahui keberadaan dan area apa saja yang telah dilewati oleh suatu pengiriman kendaraan yang sedang berjalan.

Kata kunci : *Geofencing, GPS Tracking, Google Maps, Monitoring*.

1. PENDAHULUAN

PT. Cannonex Indonesia merupakan perusahaan swasta yang mempunyai fokus dalam melayani 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. Dan 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.

Geofences merupakan teknologi inovatif yang yang memanfaatkan koordinat GPS (*Global Positioning System*) dengan menentukan batasan geografis 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 supir keluar dari parameter yang telah ditentukan, berapa lama supir berada dalam parameter, mengukur kecepatan pengendaraan dalam suatu parameter, atau memberi informasi jika supir sudah masuk dalam suatu jalan atau wilayah tertentu. Hal ini akan sangat membantu terutama bagi perusahaan yang bergerak dibidang logistik dalam me-*monitoring* pengiriman kendaraan dan pengendaranya di berbagai lokasi.

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.

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 *monitoring* 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*.

Penelitian ini diharapkan dapat membangun suatu sistem yang dapat membantu memberikan informasi posisi dan rute perjalanan, sehingga dapat meningkatkan efisiensi kerja yang akan berdampak pada penghematan pengeluaran dan memberikan info kendaraan yang *up-to-date*.

2. ISI PENELITIAN

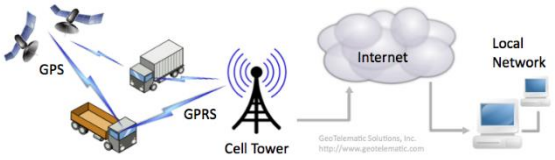
2.1 *Monitoring*

Monitoring dapat didefinisikan sebagai siklus kegiatan yang mencakup pengumpulan, peninjauan ulang, pelaporan dan tindakan atas informasi suatu proses yang sedang diimplementasikan [3]. Kegiatan monitoring lebih terfokus pada pengawasan kegiatan yang sedang dilaksanakan.

Monitoring 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.

2.2 *GPS Tracking*

Perangkat pelacak GPS merupakan alat yang digunakan sebagai penentu lokasi dari target yang biasanya dibawa oleh suatu kendaraan atau seseorang. Data lokasi pada perangkat pelacak GPS merupakan koordinat geografis yang dikirimkan ke server, seperti yang terlihat pada Gambar 1, 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 Tracking*

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. Smartphone android yang digunakan memiliki perangkat GPS dan koneksi internet.

2.3 *Geofencing*

Geofencing/geofence merupakan fitur dalam program 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. Target yang melakukan trigger biasanya merupakan koordinat geografis dari suatu GPS yang dikirim secara reguler ke suatu pengeloka data atau server.

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]. 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.

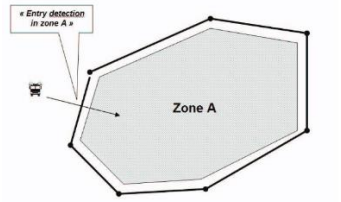


Gambar 2. *Geofence*

Pada Gambar 2 di atas dapat dilihat suatu target atau kendaraan yang memasuki suatu area yang telah di *mapping* dengan *geofence*. Saat target masuk ke area *geofence* tersebut, maka akan muncul suatu notifikasi berupa informasi event atau promosi yang di *trigger* oleh sistem. *Geofencing* dapat diterapkan untuk berbagai bidang terkait dengan pemantauan suatu target atau suatu informasi.

2.4 *Geofence Area*

Ukuran wilayah dari suatu geofencing dapat berkisar dari beberapa puluhan meter sampai dengan beberapa kilometer. Bentuk daerah geofencing sendiri bisa berbeda-beda, dari bentuk lingkaran, persegi atau persegi panjang atau lebih rumit seperti suatu poligon. Contoh salah satu bentuk geofencing dapat dilihat pada Gambar 3.



Gambar 3. Sampel *Geofence Polygon*

Koordinat dari karakteristik point atau titik yang terbentuk merupakan hal yang penting dalam mendefinisikan parameter geofence. Mekanisme *geofencing* dalam menentukan *point* adalah dengan cara menentukan terlebih dahulu latitude dan longitude tiap karakteristik *point*. Contohnya seperti *geofencing* berbentuk lingkaran yang mempunyai karakteristik satu koordinat pusat. Setelah mendapatkan latitude dan longitude tersebut, akan ditentukan radius yang diinginkan yang merupakan satu koordinat untuk mengetahui radius lingkaran. Tiap bentuk *geofencing* memiliki karakteristik *point* yang berbeda.

2.5 Analisis Masalah

Analisis masalah merupakan sebuah asumsi dari masalah yang akan diuraikan dalam prosedur-prosedur pengolahan data. 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

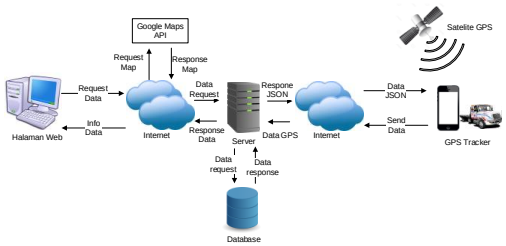
ini dikarenakan proses yang berjalan saat ini memiliki proses *monitoring* yang minimalis, yaitu melalui telepon, dan hal ini hanya dilakukan jika diperlukan. Dan tidak ada cara untuk mengetahui posisi supir secara berkala di lapangan. Dengan menggunakan suatu *tracking* GPS dan penggunaan *geofencing*, 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.

2.5 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. 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*.

2.6 Analisis Teknologi Sensor GPS

GPS Tracking merupakan alat yang digunakan sebagai penentu lokasi dari target. Data lokasi pada perangkat ini menggunakan *Global Positioning System* (GPS) untuk mengetahui koordinat geografis yang nantinya akan dikirimkan ke server atau pengelola data. *GPS Tracking* yang akan digunakan merupakan aplikasi android yang akan 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 *dependencies* terhadap *Google Play service SDK*.

```
1.  apply plugin: 'com.android.application'
2.      ...
3.      dependencies {
4.          compile 'com.google.android.gms:p
    lay-services-location:10.2.4'
5.      }
```

Hal lain yang perlu diperhatikan dalam menentukan lokasi aplikasi adalah parameter *request* yang akan digunakan untuk mendeterminasikan level akurasi dari permintaan lokasi. Untuk menyimpan parameter digunakan *LocationRequest*.

```
1.  protected void createLocationRequest() {

2.      mLocationRequest = new LocationReques
    t();
3.      mLocationRequest.setInterval(10000);

4.      mLocationRequest.setFastestInterval(5
    000);
5.      mLocationRequest.setPriority(Location
    Request.PRIORITY_HIGH_ACCURACY);
6.      mLocationRequest.setSmallestDisplacem
    ent(10);
7.  }
```

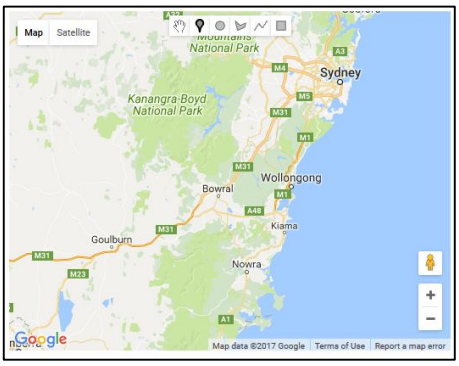
Berikut adalah beberapa deskripsi dari parameter yang akan dipakai:

a. *Update Interval*
   setInterval() – parameter ini menetapkan tingkatan atau interval dalam mili-detik dimana aplikasi memilih untuk menerima update lokasi terbaru.
b. *Fastest Update Interval*
   setFastestInterval() – parameter ini menetapkan tingkat tercepat dalam mili-detik saat aplikasi menangani update lokasi terbaru. Parameter ini digunakan sebagai batasan tertinggi aplikasi dalam menangani update lokasi.
c. *Displacement Update*
   setSmallestDisplacement() – parameter ini menetapkan jarak minimum antara lokasi dalam menangani update dalam ukuran meter. Secara default nilai jarak adalah 0.
d. *Priority*
   setPriority() – parameter ini menetapkan prioritas request

2.7 Analisis Teknologi Geofencing

Geofencing digunakan untuk menentukan batas-batas geografis virtual dari suatu peta. Bentuk daerah geofencing sendiri bisa berbeda-beda, dari bentuk lingkaran, persegi atau persegi panjang atau lebih rumit seperti suatu poligon. Pada *Google Maps*

APIs, penggambaran *geofencing* ini menggunakan *Library* yang disediakan oleh *JavaScript API* yang bernama *Drawing Layer*. *Drawing Layer* yang dapat dilihat pada Gambar 5 dibawah merupakan kelas yang menyediakan antarmuka grafis untuk user dalam menggambar *circles*, *polygons*, *rectangles*, *polylines* dan *markers* pada *map*.



Gambar 5. Drawing Layer

Untuk menggunakan fungsionalitas *Library*, aplikasi harus memuat terlebih dahulu parameter pada *Google Maps API*.

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

</script>
```

Setelah parameter ditambahkan, maka objek *DrawingManager* akan dapat dibuat sebagai berikut:

```
var drawingManager = new google.maps.drawing.DrawingManager();
drawingManager.setMap(map);
```

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

```
var drawingManager = new google.maps.drawing.DrawingManager({
  drawingMode: google.maps.drawing.OverlayType.MARKER,
  drawingControl: true,
  drawingControlOptions: {
    position: google.maps.ControlPosition.TOP_CENTER,
    drawingModes: ['marker', 'circle', 'polygon', 'polyline', 'rectangle']
  },
  circleOptions: {
      fillColor: '#ffff00',
    fillOpacity: 1,
    strokeWeight: 5,
    clickable: false,
  }
});
```

2.8 Analisis Area Geofencing

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

Penggambaran dapat juga dikonfigurasikan agar user dapat mengedit atau menyeret bentuk gambar tersebut. Gambar 6 berikut adalah contoh penggambaran garis pada *google maps*.



Gambar 6. Drawing Garis

```
function initMap() {
  var map = new google.maps.Map(document.getElementById('map'), {
    zoom: 3,
    center: {lat: 0, lng: -180},
    mapTypeId: 'terrain'
  });

  var flightPlanCoordinates = [
    {lat: 37.772, lng: -122.214},
    {lat: 21.291, lng: -157.821},
    {lat: -18.142, lng: 178.431},
    {lat: -27.467, lng: 153.027}
  ];
  var flightPath = new google.maps.Polyline({
    path: flightPlanCoordinates,
    strokeColor: '#FF0000',
    strokeOpacity: 1.0,
    strokeWeight: 2
  });

  flightPath.setMap(map);
}
```

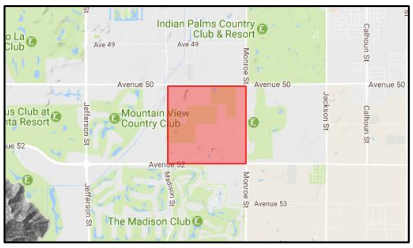
Gambar 7 berikut adalah contoh penggambaran polygon segitiga pada *google maps.*



Gambar 7. Drawing Poligon

```
1.  function initMap() {
2.    var map = new google.maps.Map(document
    .getElementById('map'), {
3.      zoom: 5,
4.      center: {lat: 24.886, lng: -
    70.268},
5.      mapTypeId: 'terrain'
6.    });
7.
8.    // Define the LatLng coordinates for t
    he polygon's path.
9.    var triangleCoords = [
10.     {lat: 25.774, lng: -80.190},
11.     {lat: 18.466, lng: -66.118},
12.     {lat: 32.321, lng: -64.757},
13.     {lat: 25.774, lng: -80.190}
14.   ];
15.
16.   // Construct the polygon.
17.   var bermudaTriangle = new google.maps.
    Polygon({
18.     paths: triangleCoords,
19.     strokeColor: '#FF0000',
20.     strokeOpacity: 0.8,
21.     strokeWeight: 2,
22.     fillColor: '#FF0000',
23.     fillOpacity: 0.35
24.   });
25.   bermudaTriangle.setMap(map);
26. }
```

Gambar 8 berikut adalah contoh penggambaran persegi pada *google maps*.



Gambar 8. Drawing Persegi

```
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.
8.    var rectangle = new google.maps.Rectangle
    ({
9.      strokeColor: '#FF0000',
10.     strokeOpacity: 0.8,
11.     strokeWeight: 2,
12.     fillColor: '#FF0000',
13.     fillOpacity: 0.35,
14.     map: map,
15.     bounds: {
16.       north: 33.685,
17.       south: 33.671,
18.       east: -116.234,
19.       west: -116.251
20.     }
21.   });
```

Gambar 9 berikut adalah contoh penggambaran persegi pada *google maps*.



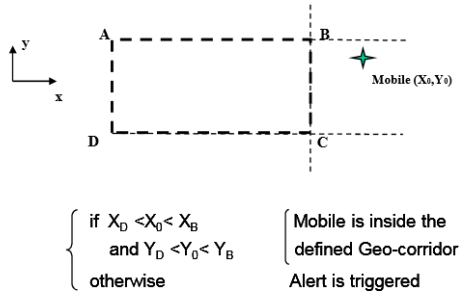
Gambar 9. Drawing Lingkaran

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.
6.  function initMap() {  // Create the map.
7.    var map = new
    google.maps.Map(document.getElementById('
    map'), {
8.      zoom: 4, center: {lat: 37.090, lng: -
    95.712},
9.      mapTypeId: 'terrain'
10.   });
11.
12.   // Construct the circle for each value
    in citymap.
13.   // Note: We scale the area of the
    circle based on the population.
14.   for (var city in citymap) {
15.     // Add the circle for this city to
    the map.
16.     var cityCircle = new
    google.maps.Circle({
17.       strokeColor: '#FF0000',
18.       map: map,
19.       center: citymap[city].center,
20.       radius:
    Math.sqrt(citymap[city].population) * 100
21.     });
```

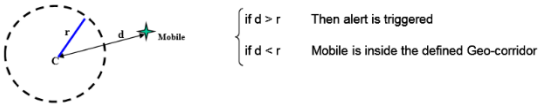
2.9 Analisis Pendeteksian Geofencing

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 dibawah ini:



Gambar 10. Perhitungan Persegi

Perhitungan *geofencing* persegi dilakukan dengan melihat apakah koordinat target berada di dalam 4 *point* 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 23 dibawah ini.



Gambar 12. Perhitungan Lingkaran

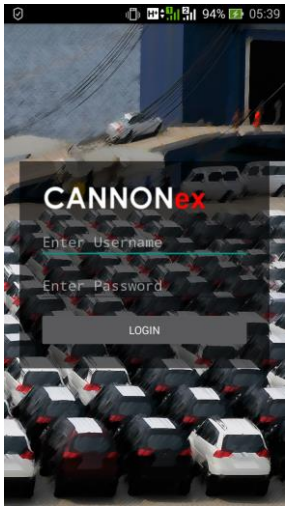
Perhitungan *geofencing* pada lingkaran melihat koordinat posisi target dan posisi koordinat tujuan. Jika jarak antara koordinat target dan tujuan lebih besar dari radius lingkaran, berarti target berada diluar lingkaran. Jika nilai lebih kecil, maka target berada di dalam lingkaran dan sistem akan melakukan 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, $latitud
    eTo, $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.   $angle = 2 * asin(sqrt(pow(sin($latDelt
    a / 2), 2) +
14.     cos($latFrom) * cos($latTo) * pow(sin
    ($lonDelta / 2), 2)));
15.   return $angle * $earthRadius;
16. }
```

2.10 Antarmuka 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 13. Tampilan Mobile (1)



Gambar 14. Tampilan Mobile (2)

Data lokasi yang dikirim berupa koordinat latitude dan longitude dengan waktu ditiap pengirimannya. Sistem mobile ini juga dapat menampilkan data pesanan yang diterima oleh supir.

2.11 Antarmuka Web

Tampilan antarmuka website yang merupakan program back-end dari aplikasi monitoring pengiriman kendaraan. Gambar 15 merupakan semua status posisi terakhir dari pengiriman.



Gambar 15. Tampilan Web (1)

Gambar 16 menampilkan antarmuka website dapat menampilkan jalur atau rute yang dilalui oleh supir.



Gambar 16. Tampilan Web (2)

.

2.12 Pengujian Fungsional

Pengujian merupakan bagian dalam pembangunan perangkat lunak yang 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.

Berdasarkan hasil pengujian *blackbox* yang dilakukan, maka dapat ditarik kesimpulan bahwa sistem berjalan dengan benar dan mengeluarkan informasi sesuai harapan. Jika ada data yang tidak sesuai, maka sistem akan menunjukkan pesan kesalahan.

2.13 Pengujian Beta

Pengujian Beta dilakukan dengan menyebarkan kuesioner kepada petugas operasional. Tabel 1 merupakan hasil dari kuesioner tersebut.

Tabel 1. Hasil Pengujian Beta

Pertanyaan 1 : 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				
Pertanyaan 2 : 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				
Pertanyaan 3 : 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				

Dari hasil diatas dapat diambil kesimpulan bahwa sekitar 82.2% responden setuju pembangunan geofencing dalam monitoring pengiriman Kendaraan membantu dan memudahkan dalam memanajemen logistic pengiriman.

3. PENUTUP

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

3.1 Kesimpulan

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

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

3.2 Saran

Implementasi *geofencing* dalam *monitoring* rute pengiriman kendaraan masih jauh dari sempurna. Oleh karena itu perlu dilakukan pengembangan dan penyempurnaan lebih lanjut. Adapun beberapa saran yang dapat diterapkan untuk pengembangan selanjutnya adalah:

a. Pemanfaatan *geofencing* selain menentukan rute pengiriman kendaraan, seperti dalam hal *security* atau POI.
b. Penggabungan *Google APIs* lain seperti *Road API* atau *Direction API* untuk meningkatkan kualitas dari sistem.
c. Penggunaan *geofecing* 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] M. Corps, *Design, monitoring, and evaluation guidebook*, 2005.

[4] H. Jogiyanto, Analisis & Desain Sistem: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis, Yogyakarta: Andi, 2005.

[5] N. Safaat, Pemrograman Aplikasi Mobile Smarthphone dan Tablet PC Berbasis Android, Bandung: Informatika Bandung, 2015.

[6] A. D. Kasman, Trik Kolaborasi Android dengan PHP dan MySQL, Yogyakarta: CV. Lokomedia, 2015.

[7] R. T. Setiadi and W. Pujiyono, "Spektrum Industri," *Sistem Pelacak Kendaraan Berbasis OpenGTS*, vol. 11, no. 2, pp. 117-242, 2013.
[8] A. Küpper, U. Bareth and B. Freese, "Geofencing and Background Tracking – The Next Features," 2011.
[9] R. C. Martin, UML for JAVA Programers, New Jersey: Alan Apt, 2002.
[10] D. Namiot and M. Sneps-Sneppe, "Geofence and Network Proximity," *Lecture Notes in Computer Science*, vol. 8121, 2013.