BAB 2
TINJAUAN PUSTAKA

2.1 Profil Perusahaan

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. Melalui pendekatan teknologi informasi, perusahaan ini telah mengembangkan jaringan kerja dan kantor perwakilan ke banyak kota tujuan di Indonesia. PT. Cannonex Indonesia berupaya memberikan kemudahan akses informasi sekaligus memanjakan para pelanggan dan para agen marketing dengan menyediakan fasilitas akses langsung secara online.

2.1.1 Sejarah

Pada awalnya PT. Cannonex Indonesia didirikan pada tahun 1992 dengan nama PT. Maju Perkasa Derap Indah yang bergerak dalam bidang jasa transportasi darat dan bongkar muat. Pada masa itu, PT. Maju Perkasa Derap Indah memiliki kendaraan truk operasional hingga 22 unit dan 2 unit top loader yang seluruhnya dioperasikan di wilayah Tanjung Priok. Sedangkan armada truk digunakan untuk melayani antara lain kegiatan *Over Brengen* (OB) yang berkerja sama dengan PT. Pelindo II.

Pada tahun 2007, perusahaan tersebut mendirikan sebuah perusahaan lagi yang bernama PT. Cakra Citra Nusantara yang memiliki 2 produk utama. Produk pertama adalah *Logistik and City Courier* dengan *brand* Andalan Total Logistik. Produk kedua adalah jasa pengiriman seluruh jenis kendaraan, alat berat dan project heavy cargo dengan *brand* CANNONex.

Melihat perkembangan usaha yang semakin meningkat, maka pada tahun 2008, CANNONex memperluas jaringannya ke berbagai kota di Indonesia (*nationwide*) yang kemudian memberikan kesempatan kepada product brand CANNONex untuk mengembangkan wilayah cakupan hingga mencapai hampir

seluruh kota di Indonesia. Dalam upaya untuk lebih efektif dan fokus dalam berkonsentrasi pada bisnis jasa pengiriman spesialis kendaraan, alat berat dan project heavy cargo, maka didirikan badan usaha baru yaitu PT. Cannonex Indonesia, sehingga diharapkan dengan ditingkatkannya unit usaha menjadi badan usaha, maka perusahaan dapat meningkatkan pencitraan dan image ke para pelanggan. Sedangkan secara internal berdampak positif terhadap kemandirian dan kemampuan organisasi masing-masing unit badan usaha.

2.1.2 Logo

Adapun logo perusahaan ditunjukkan pada Gambar 2.1 dibawah ini.



Sumber Gambar : PT. Cannonex Indonesia

Gambar 2.1 Logo Perusahaan

2.1.3 Visi dan Misi

Visi

"To Become The Number One Nationwide in Vehicle Shipping Specialist"

"To Become a Role Model For Shipping Service Companies"

PT. Cannonex Indonesia memiliki visi menjadi pemimpin di dalam penyediaan layanan pengiriman seluruh jenis kendaraan, alat berat dan project/heavy cargo ke seluruh daerah di Indonesia. Perusahaan ini berupaya menanamkan image sebagai perusahaan terkemuka dibidangnya baik kepada masyarakat perorangan maupun masyarakat industri.

Misi

Service Excellence kepada para pelanggan sehingga memberikan jaminan kepada mereka untuk mendapatkan layanan yang terbaik berupa harga yang kompetitif, kemudahan dalam bertransaksi, kemudahan dalam berinteraksi dengan

PT. Cannonex Indonesia dan sebagai *role model* bagi perusahaan cargo dan ekspedisi di Indonesia.

2.2 Landasan Teori

Landasan Teori merupakan definisi atau konsep yang disusun secara sistematis dengan dasar yang kuat dalam sebuah penelitian. Landasan teori yang digunakan dalam penyusunan Pembangunan Aplikasi Geofencing dalam Monitoring Pengiriman Kendaraan pada PT. Cannonex Indonesia meliputi Pengertian Aplikasi, Monitoring, Android, GPS (Global Positioning System), Geofencing, Metode Analisis Terstruktur, Metode Analisis Berorientasi Objek, Basis Data, JavaScript Object Notation, Internet, World Wide Web, Google APIs, PHP, CSS, Database MySQL, Adobe Dreamweaver, dan Android Studio.

2.2.1 Aplikasi

Aplikasi (*software application*) adalah penggunaan suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output* [6]. Menurut kamus besar bahasa Indonesia, aplikasi adalah penerapan dari rancangan sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Suatu perangkat lunak komputer akan memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan oleh pengguna. Contoh kegunaan perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Aplikasi merupakan rangkaian kegiatan atau perintah yang dieksekusi oleh komputer. Program merupakan kumpulan set instruksi yang akan dijalankan oleh pemroses, yaitu berupa *software*. Bagaimana suatu sistem komputer berpikir diatur oleh program ini. Program sendiri berisi konstruksi logika yang dibuat oleh manusia dan diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada set instruksi. Contoh bentuk aplikasi adalah Microsoft Office sebagai pemproses kata dan Web Browser. Aplikasi yang akan dibuat akan menggunakan sistem operasi

(OS) komputer, Web Browser dan aplikasi lain yang mendukung. Aplikasi sendiri dapat digolongkan menjadi beberapa kelas, antara lain:

1. Perangkat Lunak Perusahaan (Enterprise)
   Aplikasi yang digunakan perusahaan untuk melakukan pengorganisasian kegiatan perusahaan.
2. Perangkat Lunak Infrastruktur Perusahaan
   Aplikasi yang dibuat untuk menyediakan kemampuan-kemampuan umum yang dibutuhkan untuk membantu perangkat lunak perusahaan.
3. Perangkat Lunak Informasi Kerja
   Aplikasi yang biasa dipakai untuk menunjukkan kebutuhan individual untuk membuat dan mengolah informasi. Umumnya untuk tugas-tugas individu dalam sebuah departemen.
4. Perangkat Lunak Media dan Hiburan
   Aplikasi yang biasa digunakan untuk mengakses konten tanpa *editing*, tapi bisa saja termasuk aplikasi yang memungkinkan mengedit konten.
5. Perangkat Lunak Pendidikan
   Aplikasi yang yang memiliki similaritas dengan perangkat lunak media dan hiburan, tetapi mempunyai konten pendidikan.
6. Perangkat Lunak Pengembangan Media
   Aplikasi yang digunakan untuk menunjukkan kebutuhan individu untuk menghasilkan media cetak dan elektronik.
7. Perangkat Lunak Rekayasa Produk
   Aplikasi yang biasa digunakan untuk pengembangan produk hardware dan software.

Pada penelitian ini, pembangunan aplikasi nantinya bertujuan untuk mempermudah proses kerja dari pegawai di perusahaan dengan mengkomputerisasi proses-proses yang sudah ada atau yang terlibat dalam suatu aplikasi monitoring pengiriman kendaraan.

2.2.2 Monitoring

Monitoring dapat didefinisikan sebagai siklus kegiatan yang mencakup pengumpulan, peninjauan ulang, pelaporan dan tindakan atas informasi suatu proses yang sedang diimplementasikan [7]. 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 disepakati. Indikator monitoring mencakup esensi aktivitas dan target yang ditetapkan pada awal perencanaan program. Apabila monitoring ini dilakukan dengan baik akan bermanfaat dalam menjaga proses pelaksanaan kegiatan tetap pada jalurnya (sesuai pedoman dan perencanaan program), dan memberikan informasi kepada pengelola program apabila terjadi hambatan dan penyimpangan, serta sebagai masukan dalam melakukan evaluasi.

Ada beberapa tujuan dari pengembangan sistem *monitoring*. Hal ini dapat dilihat dari beberapa segi, seperti segi subjek dan objek yang dipantau, serta hasil dari proses *monitoring* itu sendiri.

1. Menyediakan informasi data yang akurat bagi petugas yang melakukan *monitoring*.
2. Mengidentifikasi hasil yang tidak diinginkan pada suatu proses dengan cepat (seperti kecepatan mengemudi supir).
3. Mempengaruhi pengembangan motivasi dan kebiasaan positif pekerja (dalam kasus ini supir akan dibatasi kecepatan mengemudi sehingga lebih terjaga dalam mengendarai kendaraan).
4. Pengoleksian data untuk pengecekan progress terhadap suatu target (seperti jalur yang digunakan oleh supir dengan menggunakan *geofencing)*

2.2.3 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi [8]. Android menyediakan platform yang terbuka (*opensource*) sehingga para pengembang atau programmer bisa membuat aplikasi sendiri. Terdapat juga *Android Market* yang menyediakan ribuan aplikasi baik yang gratis maupun berbayar.

Android sendiri bermula dari perusahaan software Android Inc yang didirikan pada bulan Oktober 2003. Pada bulan Agustus 2005, perusahaan ini diakuisisi oleh Google Inc dan ini merupakan langkah awal yang dilakukan Google agar dapat masuk ke pasar mobile phone/smartphone [9]. Saat ini terdapat dua jenis distributor sistem operasi Android, pertama yang mendapatkan dukungan penuh dari Google atau Google Mail Service (GMS) dan kedua adalah Open Handset Distribution (OHD) yang merupakan distribusi bebas tanpa adanya dukungan langsung dari Google.

Saat ini, sistem operasi Android menjadi pilihan yang menguntungkan bagi banyak pengguna smartphone, karena memiliki biaya lisensi yang lebih murah dan sifatnya yang semi-open source. Selain itu, Android tentunya juga memiliki fitur-fitur terkait dalam pengembangan sistem monitoring dan juga akan didukung oleh berbagai layanan dari Google berbentuk API. Dengan adanya dukungan ini, maka pengembangan aplikasi dari segi mobile akan menggunakan perangkat Android.

2.2.3.1 Arsitektur Aplikasi Berbasis Android

Google mengibaratkan Android sebagai sebuah tumpukan software. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Berikut adalah susunan dari lapisan-lapisan jika dilihat dari lapisan dasar hingga lapisan teratas.

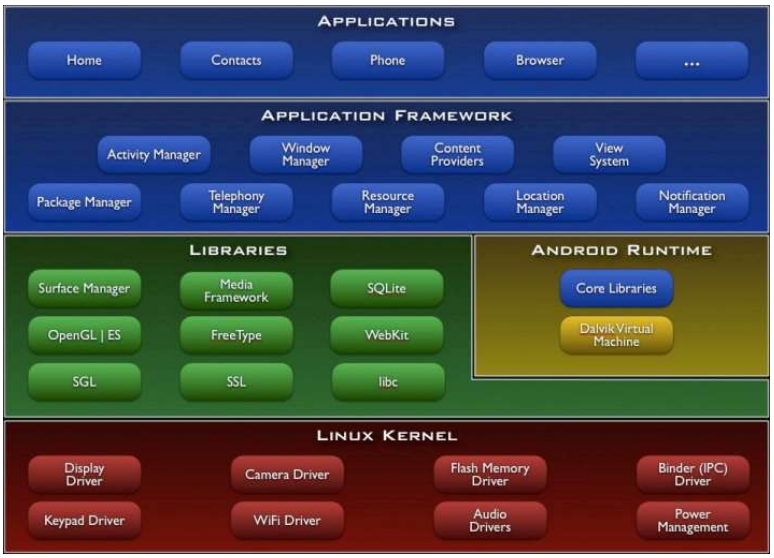
Tumpukan paling bawah pada arsitektur Android ini adalah Linux Kernel. Google menggunakan kernel Linux versi 2.6 untuk membangun sistem Android, yang mencakup *memory management, security setting, power management* dan

beberapa *drive hardware*. Kernel berperan sebagai abstraction layer antara hardware dan keseluruhan software [9].

Lapisan setelah kernel Linux adalah *Android Runtime* yang *berisikan Core Libraries* dan *Dalvik Virtual Machine*. Core Libraries mencakup serangkaian inti library Java, artinya Android menyediakan satu set *library-library* dasar. Sedangkan *Dalvik* adalah *Java Virtual Machine* yang memberi kekuatan pada sistem android. Dilapisan ini juga terdapat beberapa library lain yang bisa digunakan dalam pembangunan aplikasi android.

Lapisan *Application Framework* adalah serangkaian tool dasar seperti alokasi resource smartphone, aplikasi telepon, pergantian antar proses atau program, dan pelacakan lokasi fisik telepon. Pengembang aplikasi memiliki akses kepada *tool-tool* dasar tersebut dan memanfaatkannya untuk menciptakan aplikasi yang lebih kompleks.

Lapisan teratas adalah aplikasi itu sendiri yang merupakan fungsi-fungsi dasar smartphone seperti menelepon, mengirim pesan, browsing, foto, dan sebagainya. Untuk sebagian besar pengguna, lapisan inilah yang paling sering diakses melalui *user interface*.



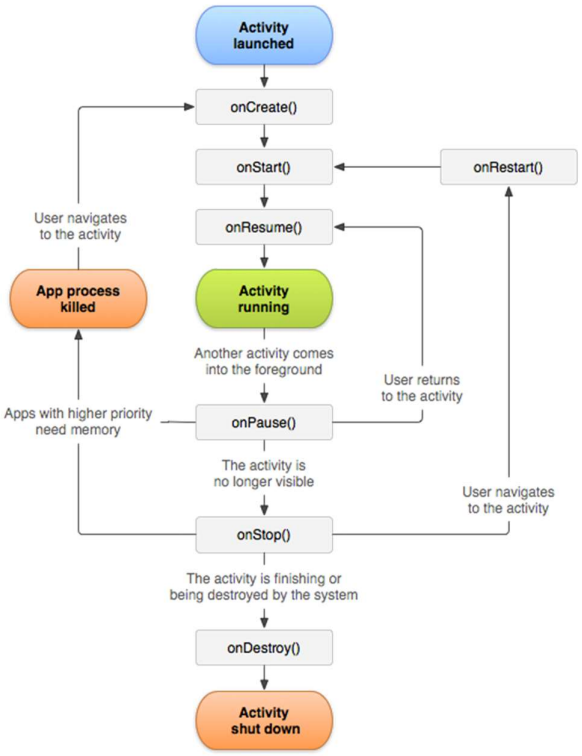
Sumber Gambar : Akhmad Dharma Kasman [9]

Gambar 2.2 Arsitektur Android

Untuk pembangunan aplikasi monitoring untuk subsistem mobile, akan focus pada pengembangan di Lapisan *Application Framework* dan Lapisan teratas *User Interface*. Di lapisan teratas, supir sebagai pengguna akan dapat menggunakan aplikasi dengan tampilan yang mudah dimengerti. Aplikasi yang dibuat juga akan menggunakan tool-tools yang terdapat pada lapisan *Application Framework* seperti pencarian lokasi atau pergantian proses.

2.2.3.2 Activity Lifecycle Android

Saat user menggunakan suatu aplikasi android, aplikasi tersebut akan melakukan transisi ke berbagai *state* dalam *lifecycle*. Sebuah kelas aktifitas akan mempunyai beberapa *callbacks* yang memperbolehkan suatu aktifitas untuk mengetahui bahwa *state* telah berubah. Berbagai bentuk *state* seperti, sistem dibuat, sistem berhenti, sistem mulai lagi, atau sistem dihapus. Dalam menavigasi transisi antar bagian dari lifecyle, suatu kelas aktifitas memberikan 6 inti callbacks, seperti onCreate(), onStart(), onResume(), onPause(), onStop() dan onDestroy(). Aktifitas lifecyle ini dapat dilihat pada Gambar 2.3 berikut.



Sumber Gambar : https://developer.android.com

Gambar 2.3 Activity Lifecycle

Berdasarkan kompleksitas dari aktifitas program yang dibuat, tidak semua metode lifecycle perlu diterapkan. Tetapi penting untuk memahami setiap aktifitas agar aplikasi yang dibuat berjalan sesuai yang diinginkan.

2.2.4 GPS (Global Positioning System)

Global Positioning System atau biasa disingkat dengan GPS merupakan teknologi penentu lokasi yang banyak digunakan saat ini. GPS adalah suatu sistem radio navigasi penentuan lokasi menggunakan satelit [10]. Dengan bantuan satelit, akan diperoleh posisi yang akurat dan cepat dengan koordinat 3 dimensi (x,y,z) ditambah dengan informasi waktu dan kecepatan bergerak. GPS pertama dikembangkan oleh Departemen Pertahanan Amerika Serikat di tahun 1973.

NAVSTAR GPS (*NAVigation Satellite Timing And Ranging*) merupakan nama lengkap dari GPS.

Bagian utama dari sistem GPS adalah 24 satelit yang mengorbit Bumi di ketinggian 20.200 kilometer. Orbit satelit dirancang sehingga setiap titik di Bumi dapat melihat paling sedikit empat satelit pada setiap saat. Tiap satelit mengitari bumi kira-kira sekali dalam 12 jam dengan kecepatan sekitar 11.000 kilometer per jam. Satelit GPS mempunyai panel-panel pengumpul tenaga Matahari untuk membangkitkan energi listrik yang diperlukannya. Selain itu juga ada baterai yang menyimpan tenaga listrik dan mempergunakannya saat satelit tidak memperoleh sinar Matahari.

Awalnya GPS hanya digunakan untuk kepentingan militer, tapi pada tahun 1980-an, penggunaan tersebut berkembang untuk kepentingan sipil. GPS dapat digunakan dimanapun juga dalam 24 jam. Posisi unit GPS akan ditentukan berdasarkan titik-titik koordinat *latitude* dan *longitude* atau disebut juga garis lintang dan bujur (latitude sebgai garis lintang, longitude sebagai garis bujur) yang diperolehnya dari nilai derajat dari suatu titik yang diukur. Koordinat umumnya dituliskan dengan format DD (*Degree Decimal*), tetapi ada juga yang dituliskan dengan format DMS (*Degree-Minute-Second*)

Penggunaan GPS akan membantu menemukan letak koordinat latitude dan longitude dari pengiriman kendaraan. GPS yang digunakan akan memanfaatkan GPS yang terdapat pada smartphone yang akan diaktifkan pada saat supir dijalan. Untuk 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*.

2.2.4.1 Segmen GPS

Secara teknis GPS memiliki 3 segmen utama yaitu *Space Segment, Control Segment* dan *User Segment*.

1. GPS Space Segment

   Space segment adalah sebuah jaringan satelit yang tediri dari beberapa satelit yang berada pada orbit lingkaran yang terdekat dengan tinggi nominal sekitar 20.183 km di atas permukaan bumi. Sinyal yang dipancarkan oleh seluruh satelit tersebut dapat menembus awan, plastik dan kaca, namun tidak bisa menembus benda padat seperti tembok dan rapatnya pepohonan.

2. GPS Control Segment

   Control segment terdiri dari lima stasiun yang berada di pangkalan Falcon Air Force, Colorado Springs, Ascension Island, Hawaii, Diego Garcia dan Kwajalein. Kelima stasiun ini adalah mata dan telinga bagi GPS. Sinyal-sinyal dari satelit diterima oleh bagian kontrol, kemudian dikoreksi, dan dikirimkan kembali ke satelit. Data koreksi lokasi yang tepat dari satelit ini disebut data ephemeris, yang kemudian nantinya dikirimkan ke alat navigasi yang kita miliki.

3. GPS User Segment

   User segment terdiri dari antenna dan *prosesor receiver* yang menyediakan *positioning*, kecepatan dan ketepatan waktu ke pengguna. Bagian ini menerima data dari satelit-satelit melalui sinyal radio yang dikirimkan setelah mengalami koreksi oleh stasiun pengendali (GPS Control Segment). Penggunaan dari GPS smartphone terletak pada bagian ini, sebagai receiver data. Penelitian yang dibangun tidak membahas segmen Space dan Control.

2.2.4.2 Akurasi GPS

Akurasi atau ketepatan perlu mendapat perhatian bagi penentuan koordinat sebuah titik atau lokasi. Koordinat posisi ini akan selalu mempunyai faktor kesalahan, yang lebih dikenal dengan tingkat akurasi. Misalnya, alat tersebut menunjukkan sebuah titik koordinat dengan akurasi 3 meter, artinya posisi sebenarnya bisa berada dimana saja dalam radius 3 meter dari titik koordinat (lokasi) tersebut. Makin kecil angka akurasi (artinya akurasi makin tinggi), maka

posisi alat akan menjadi semakin tepat. Harga alat juga akan meningkat seiring dengan kenaikan tingkat akurasi yang bisa dicapainya.

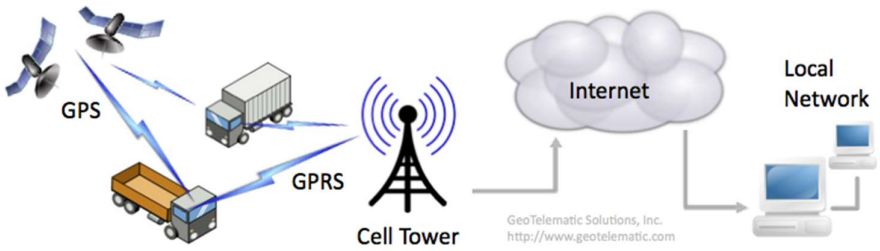
Pada pemakaian sehari-hari, tingkat akurasi ini lebih sering dipengaruhi oleh faktor sekeliling yang mengurangi kekuatan sinyal satelit. Karena sinyal satelit tidak dapat menembus benda padat dengan baik, maka ketika menggunakan alat, penting sekali untuk memperhatikan luas langit yang dapat dilihat.

Ketika alat berada disebuah lembah yang dalam (misal, akurasi 15 meter), maka tingkat akurasinya akan jauh lebih rendah daripada di padang rumput (misal, akurasi 3 meter). Di padang rumput atau puncak gunung, jumlah satelit yang dapat dijangkau oleh alat akan jauh lebih banyak daripada dari sebuah lembah gunung.

GPS smartphone akan bergantung pada satelit dan BTS (Base Transceiver Station), maka sinyal satelit atau posisi BTS menjadi sangat penting. Apabila alat navigasi berbasis satelit ini tidak dapat bekerja maksimal karena adanya gangguan, maka smartphone nantinya akan mengambil sinyal posisi dari BTS sebagai sinyal backup.

2.2.4.3 GPS Tracker

Perangkat pelacak GPS (*GPS Tracker*) 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 atau pengelola data 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. Hal ini memberikan lokasi aset unit yang dapat ditampilkan dengan latar belakang peta baik secara *real time* atau ketika menganalisis jalur rute perjalanan. Gambar 2.4 berikut merupakan contoh implementasi perangkat penerima GPS pada sistem pelacak.



Sumber Gambar : http://www.geotelematic.com

Gambar 2.4 GPS Tracker

GPS Tracker yang akan digunakan merupakan aplikasi android yang akan dirancang untuk mengirimkan data GPS berupa koordinat *latitude* dan *longitude* ke server secara *realtime* dengan konektifitas internet. Smartphone android yang digunakan harus memiliki perangkat GPS dan harus mempunyai koneksi internet. Petugas Supir akan membawa smartphone dalam pengiriman kendaraan dan akan mengaktifkan GPS Tracker selama diperjalanan.

2.2.5 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*. 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]. 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 2.5 Geofencing

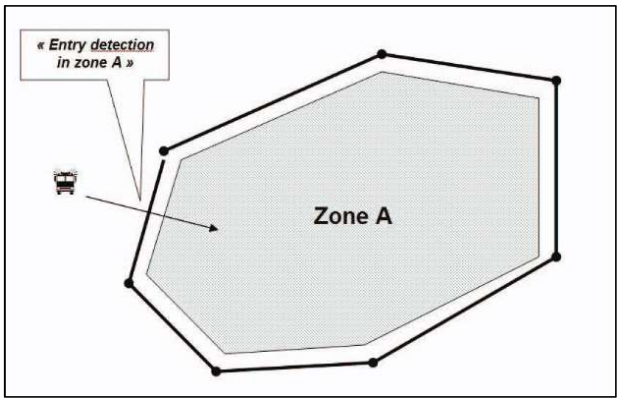
Pada Gambar 2.5 di atas dapat dilihat suatu target atau kendaraan yang memasuki suatu area yang telah di *mapping* dengan *geofence*. Saat target masuk ke area *geofence* tersebut, maka akan muncul suatu notifikasi berupa informasi *event* atau promosi yang di *trigger* oleh sistem. *Geofencing* dapat diterapkan untuk berbagai bidang terkait dengan pemantauan suatu target atau suatu informasi.

Pada penelitian ini, *geofencing* akan digunakan dalam me-*monitoring* pengiriman kendaraan. Hal ini akan dilakukan dengan menggambarkan area-area virtual pada suatu daerah, seperti gerbang tol, jalan tertentu ataupun suatu kota. Karena supir dapat mengubah jalur sesuai kebutuhan, area *geofencing* akan diletakkan pada posisi tertentu, jadi semua jalur penting yang dilewati dapat di monitor. Pada saat petugas supir melewati area yang telah ditetapkan, maka sistem akan mengetahui area yang dilalui oleh supir. Area tersebut akan ditampilkan pada aplikasi *monitoring* sebagai penanda bahwa supir telah melewati area tersebut. Area ini juga akan disimpan ke *database*, sehingga dapat dijadikan sebagai informasi rute apa yang dilewati oleh supir.

2.2.5.1 Area Geofencing

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 2.6.



Sumber Gambar : Geofencing for Fleet and Freight Management [1]

Gambar 2.6 Contoh Geofencing Poligon

Koordinat dari karakteristik *point* atau titik yang terbentuk merupakan hal yang penting dalam mendefinisikan area 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 kordinat untuk mengetahui radius lingkaran. Tiap bentuk *geofencing* memiliki karakteristik point yang berbeda.

Koordinat area *geofencing* yang telah di *mapping* akan disimpan ke *database* berdasarkan karakteristiknya. Koordinat ini nantinya akan digunakan pada saat *monitoring* pengiriman kendaraan dengan cara menggabungkan data koordinat sistem GPS Tracker dengan data area *geofencing*. Saat data koordinat dari GPS memasuki daerah yang telah di *mapping*, sistem akan meng-*update* data lokasi sehingga memberikan informasi posisi terbaru pengendara. Penggambaran area *geofencing* dan pendeteksian target masuk ke dalam area akan menggunakan *Library* yang disediakan oleh *Google API*.

2.2.6 Metode Analisis Terstruktur

Metode Analisis Terstruktur (*Structured Analysis*) merupakan salah satu teknik analisis yang mengunakan pendekatan berorientasi fungsi. Teknik ini mempunyai sekumpulan petunjuk dan perangkat komunikasi grafis yang memungkinkan analis sistem mendefinisikan spesifikasi fungsional perangkat lunak secara terstruktur. Pada metode ini, semua fungsi sistem direpresentasikan sebagai sebuah proses transformasi informasi dan disusun secara hirarkis sesuai tingkat abstraksinya (sistem maupun perangkat lunak) yang hasilnya ditujukan untuk entitas-entitas eksternal [11].

Analisis Terstruktur pertama kali diperkenalkan oleh Tom DeMarco sekitartahun 1978 [12]. Prinsip dari teknik ini adalah dekomposisi fungsi dari sistem berdasarkan aliran data dan proses-prosesnya untuk mendapatkan produk analisis yang dapat diubah dan diperbaiki secara mudah. Dalam bukunya, DeMarco mendefinisikan *Analisis Terstruktur* sebagai teknik untuk mendeskripsikan spesifikasi sistem baru melalui *Data Flow Diagrams* (DFD), *Data Dictionary* (Kamus Data), *Structured English*, dan *Data Structure Diagrams*. Spesifikasi sistem tersebut dinyatakan dalam suatu dokumen yang disebut Spesifikasi Terstuktur (Structured Specification). Dalam perkembangannya, teknik analisis terstruktur mengalami perubahan, penambahan, dan penyempurnaan, baik untuk perangkat pemodelannya maupun mekanisme atau cara pelaksanaannya.

2.2.6.1 Diagram Konteks

Diagram Konteks adalah data flow diagram tingkat atas (DFD Top Level), yaitu diagram yang paling tidak detail, dari sebuah sistem informasi yang menggambarkan aliran-aliran data ke dalam dan ke luar sistem dan ke dalam dan ke luar entitas-entitas eksternal.

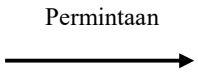
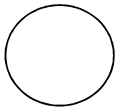
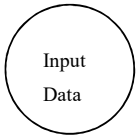
2.2.6.2 Data Flow Diagram

Data Flow Diagram (DFD) adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas.

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluaran dari sistem, dimana data di simpan, proses apa yang menghasilkan data tersebut, dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. DFD terdiri dari empat komponen utama, diantaranya adalah:

1. Entitas (Entity)
2. Proses
3. Aliran Data (Data Flow)
4. Data Storage

Tabel 2.1 Simbol DFD

Simbol	Makna	Contoh
	Entity	
	Aliran Data	
	Proses	
	Data Storage	

Simbol-sombol pada Tabel 2.1 yang digunakan dalam DFD mewakili maksud tertentu, seperti:

1. External Entity (kesatuan Luar) atau Boundary (batas sistem)

   Setiap sistem pasti memiliki batas sistem (boundary) yang memisahkan suatu sistem

dengan lingkungan luarnya. Kesatuan luar (external entity) merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainya yang berada di lingkungan luarnya yang memberikan input atau menerima output dari sistem.

2. Data flow (arus data)

   Arus data di DFD diberi simbol panah. Arus data ini mengalir diantara proses, simpanan, dan kesatuan luar.

3. Process (proses)

   Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.

4. Data store (simpanan data)

   Simpanan data (data store) merupakan simpanan dari data yang dapat berupa suatu file atau database di komputer, suatu arsip atau catatan manual dan lain sebagainya.

2.2.6.3 Spesifikasi Proses

Spesifikasi Proses adalah tabel yang berisi keterangan atau deskripsi dari semua proses yang terdapat pada DFD. Logika proses yang ditulis pada spesifikasi proses harus dituliskan secara jelas, baik menggunakan bahasa deskriptif atau pseudecode, tetapi tidak boleh mengkkombinasikan kedua bahasa tersebut. Tabel 2.2 berikut adalah contoh tabel spesifikasi proses.

Tabel 2.2 Contoh Spesifikasi Proses

No	Proses	Keterangan
1.	No Proses	Menyatakan no proses
	Nama Proses	Menyatakan nama proses
	Source atau sumber	Menyatakan sumber data input menuju proses
	Input	Menyatakan isi data yang masuk ke proses
	Output	Menyatakan informasi yang keluar dari proses
	Destination	Menyatakan tujuan informasi output dari proses
	Logika Proses	Menyatakan algoritma dari proses

2.2.6.4 Kamus Data

Kamus data adalah tabel yang berisi deskripsi dari data yang mengalir pada DFD, Penjelasan struktur data dapat berupa field tiap data harus sama dengan yang sudah dimodelkan di ERD. Tabel 2.3 berikut adalah tabel contoh kamus data.

Tabel 2.3 Contoh Kamus Data

Nama aliran data	Data nama dari data yang digunakan
Where use/How used	Daftar pada poin proses apa data digunakan
Keterangan	Uraian singkat dari data yang digunakan
Struktur Data	Daftar komponen data yang ada pada data yang digunakan
Deskripsi	Jenis data dalam representasi komputer untuk masing-masing data

2.2.7 Metode Analisis Berorientasi Objek

UML singkatan dari *Unifed Modeling Language* yang berarti bahasa pemodelan standar. UML merupakan bahasa standar untuk merancang dan mendokumentasikan perangkat lunak dengan cara berorientasi objek. Ada beberapa diagram yang digunakan proses pembuatan perangkat lunak berorientasi objek diantaranya, *use case diagram, activity diagram, class diagram* dan *sequence diagram* [11].

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*). UML sendiri terdiri atas pengelompokkan diagram-diagram system menurut aspek atau sudut pandang tertentu.

2.2.7.1 Use Case Diagram

Use Case Diagram menggambarkan sekelompok *use cases* dan aktor yang disertai dengan hubungan diantaranya. Diagram *use cases* ini menjelaskan dan menerangkan kebutuhan atau *requirement* yang diinginkan atau dikehendaki user atau pengguna, serta sangat berguna dalam menentukan struktur organisasi dan model dari pada sebuah sistem.

Diagram menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Yang menjadi persoalan itu apa yang dilakukan bukan bagaimana melakukannya.

Diagram *Use Case* dekat kaitannya dengan kejadian-kejadian (*scenario*) yang merupakan contoh apa yang akan terjadi ketika seseorang berinteraksi dengan sistem. Terdapat dua hal utama yang diperlukan dalam pembentukan suatu *use case diagram*, yaitu aktor dan *use case*. Aktor merupakan orang, benda ataupun sistem lain yang berinteraksi dengan sistem yang akan dibangun. Sedangkan *use case* merupakan fungsionalitas atau layanan yang disediakan oleh sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor.

2.2.7.2 Class Diagram

Diagram *Class* memberikan pandangan secara luas dari suatu sistem dengan menunjukan kelas-kelasnya dan hubungan mereka. Diagram *Class* bersifat statis menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Atribut dan metode dapat memiliki salah satu sifat sebagai berikut:

1. *Private*, tidak dapat dipanggil dari luar kelas yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh kelas yang bersangkutan dan anak-anak yang mewarisinya.
3. *Public*, dapat dipanggil oleh siapa saja.

Diagram *Class* mempunyai 3 macam *relationships* (hubungan), sebagai berikut:

1. *Association*

   Suatu hubungan antara bagian dari dua kelas. Terjadi association antara dua kelas jika salah satu bagian dari kelas mengetahui yang lainnya dalam melakukan suatu kegiatan. Di dalam diagram, sebuah *association* adalah penghubung yang menghubungkan dua kelas.
2. *Aggregation*

   Suatu *association* dimana salah satu kelasnya merupakan bagian dari suatu kumpulan. *Aggregation* memiliki titik pusat yang mencakup keseluruhan bagian.
3. Generalization

   Suatu hubungan turunan dengan mengasumsikan satu kelas merupakan suatu *superClass* (kelas super) dari kelas yang lain. *Generalization* memiliki tingkatan yang berpusat pada *superClass*.

2.2.7.3 Sequence Diagram

Diagram *Class* dan diagram *Object* merupakan suatu gambaran model statis. Namun ada juga yang bersifat dinamis, seperti Diagram *Interaction*. Diagram

sequence merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan, *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

Banyaknya *sequence* diagram yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada *sequence* diagram sehingga semakin banyak *use case* yang didefinisikan maka *sequence* diagram yang harus dibuat juga semakin banyak. Penomoran pesan berdasarkan urutan iteraksi pesan. Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu.

2.2.7.4 Activity Diagram

Pada dasarnya diagram *Activity* sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram Statechart. Diagram Statechart berfokus pada obyek yang dalam suatu proses (atau proses menjadi suatu obyek), diagram *Activity* berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain. Setiap *use case* yang telah dibentuk digambarkan aktivitasnya dalam *activity* diagram, mulai dari peran aktor, peran sistem, dan *decision*.

2.2.8 Basis Data

Basis data (*database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system*, DBMS).

Menurut Stephens dan Plew [13], basis data adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang

kita gunakan sehari-hari untuk berbagai alasan. Dengan basis data, pengguna dapat menyimpan data secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data disimpan dalam basis data menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan kedalam basis data, dimodifikasi dan dihapus.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya, penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu basis data dan hubungan di antara obyek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data dan ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom. Dalam model ini hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel.

2.2.8.1 Database Management System (DBMS)

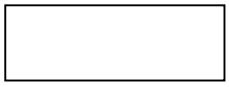
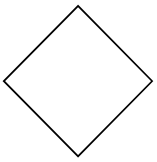
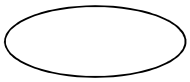
Pengelolaan suatu basis data ditangani oleh suatu perangkat lunak atau sistem khusus yang akan menentukan bagaimana data diorganisasikan, disimpan, diubah, dan diambil kembali. Sistem ini juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, dan keakuratan atau konsistensi data. Perangkat lunak yang termasuk DBMS misalnya MySQL Server, Oracle, dan sebagainya. Aplikasi yang dibangun akan menggunakan MySQL Server.

2.2.8.2 Entity Relationship Diagram (ERD)

ERD atau *Entity Relationship Diagram* merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

Menurut Jogiyanto [6], ERD adalah suatu komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut yang mempresentasikan seluruh fakta dari dunia nyata yang ditinjau. Entity Relationship Diagram menggambarkan data dan hubungan antar data secara global. Tabel 2.4 berikut menjelaskan keterangan Simbol ERD.

Tabel 2.4 Simbol ERD

Notasi	Keterangan
	Entitas, yaitu kumpulan dari objek yang dapat diidentifikasi secara unik.
	Relasi, yaitu hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain: satu ke satu, satu ke banyak, dan banyak ke banyak.
	Atribut, yaitu karakteristik dari entity atau relasi yang merupakan penjelasan detail tentang entitas.
	Garis, yaitu hubungan antara entity dengan atributnya dan himpunan entitas dengan himpunan relasi.
	Input/output data, yaitu proses input atau output data, parameter, atau informasi.

Dalam memodelkan struktur data dan hubungan antar data, ERD menggunakan beberapa notasi dan simbol. Pada dasarnya ada tiga komponen yang digunakan, yaitu :

1. Entitas

   Entitas adalah segala sesuatu yang dapat digambarkan oleh data. Entitas juga dapat diartikan sebagai individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain (Fathansyah, 1999). Ada dua macam entitas yaitu entitas kuat dan entitas lemah. Entitas kuat merupakan entitas yang tidak memiliki ketergantungan dengan entitas lainnya. Contohnya entitas anggota. Sedangkan entitas lemah merupakan entitas yang kemunculannya tergantung pada keberadaaan entitas lain dalam suatu relasi. Simbol dari entitas ini biasanya digambarkan dengan persegi panjang.

2. Atribut

   Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasikan isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol elips

3. Hubungan Relasi

   Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Suatu entitas yang berelasi dengan entitas lain mempunyai kardinalitas. Kardinalitas relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain. Terdapat 3 macam kardinalitas relasi yaitu :

   a. Satu ke satu (*One to one*)

      Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

   b. Satu ke banyak (*One to many*)

      Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada

entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

c. Banyak ke banyak (*Many to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

2.2.9 JavaScript Object Notation (JSON)

JSON *(JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data [9]. JSON sendiri terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*. 2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).
2. Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data ini [9].

JSON nantinya akan digunakan sebagai pertukaran data antara android dan *MySQL server* dimana hasil dari *query* data yang dilakukan pada *server side* akan

menghasilkan data yang akan di format ke dalam *array* dan di *publish* ke dalam bentuk JSON.

2.2.10 Internet

Interconnection network (internet) adalah sistem global dari seluruh jaringan komputer yang saling terhubung. Internet berasal dari bahasa latin "*inter*" yang berarti "antara". Internet merupakan jaringan yang terdiri dari milyaran komputer yang ada di seluruh dunia. Internet melibatkan berbagai jenis komputer serta topology jaringan yang berbeda. Dalam mengatur integrasi dan komunikasi jaringan, digunakan standar protokol internet yaitu TCP/IP. TCP bertugas untuk memastikan bahwa semua hubungan bekerja dengan baik, sedangkan IP bertugas untuk mentransmisikan paket data dari satu komputer ke komputer lainya.

Sejarah internet awalnya berasal dari proyek ARPA yang dibentuk tahun 1969 oleh Departemen Pertahanan Amerika Serikat. Proyek ini kemudian dikenal dengan ARPANET (*Advanced Research Project Agency Network*) yang melakukan riset tentang cara menghubungkan komputer satu dengan komputer lainnya agar bisa saling berkomunikasi. Pada tahun 1970, proyek ini berhasil menghubungkan lebih dari 10 komputer dalam bentuk jaringan, dan beberapa tahun kemudian, hasil riset proyek ini dikembangkan di luar Amerika. Karena jumlah komputer yang terhubung semakin banyak, maka pada tahun 1980 dibuatlah protokol resmi yang dikenal dengan TCP/IP (*Transmission Control Protocol/Internet Protocol*).

2.2.11 World Wide Web

World Wide Web atau WWW atau juga dikenal dengan WEB adalah salah satu layanan yang didapat oleh pemakai computer yang terhubung ke internet. Web ini menyediakan informasi bagi pemakai computer yang terhubung ke internet dari sekedar informasi “sampah” atau informasi yang tidak berguna sama sekali sampai informasi yang serius; dari informasi yang gratisan sampai informasi yang komersial.

Website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi,

suara, dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).

Halaman-halaman dari website dapat diakses melalui URL (*Uniform Resource Locator*) yang biasa disebut dengan *homepage*. URL ini mengatur halaman-halaman dari situs untuk menjadi sebuah hirarki, sehingga berbagai hyperlink yang berada pada halaman tersebut dapat mengatur para pembaca tentang susunan atau arus informasi yang berjalan dalam sebuah website.

Aplikasi *geofencing* dalam *monitoring* pengiriman kendaraan yang dibangun nantinya akan terhubung dengan web dan dan dapat diakses oleh petugas melalui suatu URL.

2.2.12 Google APIs

Google APIs bisa di katakan bagian dari Framework Google. Google menyediakan berbagai API (*Application Programming Interface*) yang sangat berguna bagi pengembang web maupun aplikasi mobile untuk memanfaatkan berbagai fitur yang disediakan oleh Google seperti misalnya: AdSense, Search Engine, Translation maupun YouTube.

API secara sederhana bisa diartikan sebagai kode program yang merupakan antarmuka atau penghubung antara aplikasi atau web yang kita buat dengan fungsi-fungsi yang dikerjakan. Misalnya dalam hal ini *Google API* berarti kode program (yang disederhanakan) yang dapat kita tambahkan pada aplikasi atau web kita untuk menjalankan fungsi atau fitur yang disediakan Google. Misalnya saja kita bisa menambahkan fitur Google Map pada website yang dibangun.

Google API dapat dipelajari langsung melalui lewat situs *https://developers/google/com*. Ada banyak API yang disediakan oleh Google, beberapa diantaranya adalah:

1. *Gmail API*: mengakses Gmail *mailbox* termasuk mengirimkan email.
2. *Google Earth Engine API*: memanfatkan fitur *geospatial processing* yang ada pada Google.

3. *Google Maps Geolocation API*: memanfaatkan fitur pencarian lokasi dari Google.
4. *Google Maps Javascript API*: memanfaatkan fitur yang ada pada Google Maps.

2.2.12.1 Google Maps

Google Maps merupakan suatu layanan yang disediakan oleh perusahaan Google dalam menggunakan peta virtual berbasis *Web* yang dapat diakses melalui suatu browser di komputer atau *app* di *mobile device*. Layanan ini gratis untuk digunakan dan bisa diakses di *http://maps.google.com*. *Google Maps* menawarkan peta dunia virtual yang dapat digeser, diperbesar, diperkecil dan mempunyai beberapa fitur lain, seperti pencarian rute, penunjuk arah, pencarian tempat dan sebagainya.

Google Map APIs merupakan aplikasi *interface* yang dapat diakses lewat javascript yang memungkinkan untuk embedding Google Maps ke halaman web yang dikembangkan diluar perusahaan Google. Hal ini dirancang agar dapat bekerja pada *mobile device* dan juga aplikasi *browser desktop*. API sudah termasuk lokalisasi bahasa dan juga memiliki mekanisme untuk perusahaan pengembang yang ingin memanfaatkan API Google Maps dalam sebuah internet atau intranet. Layanan API HTTP dapat diakses melalui HTTPS dengan menggunakan Google Map API. Salah satu API yang dapat digunakan adalah Google *Map Geolocation API* yang memberikan informasi lokasi, Google *Map JavaScript API* yang digunakan untuk mengkostumasi peta Google dengan konten tambahan, Google *Map Embed API* untuk mengimplementasikan Google Map di website sendiri tanpa kuota limit dan lain-lain. Dokumentasi yang tersedia di situs *https://developers.google.com/maps* menjelaskan protokol apa yang digunakan untuk mengirim data tersebut ke server dan mengembalikan *respond* ke *client*.

2.2.13 Pretext Hyper-Processor (PHP)

PHP adalah kependekan dari *Pretext Hyper-Processor* yang dibangun oleh Rasmus Lerdorf pada tahun 1994. Pada awal pengembangannya PHP disebut sebagai kependekan dari *Personal Home Page*.

PHP adalah bahasa *scripting* yang menyatu dengan HTML dan dijalankan pada *server side*. Artinya semua sintaks yang kita berikan akan sepenuhnya dijalankan pada server sedangkan yang dikirimkan ke browser hanya hasilnya saja. PHP menyatu dengan bahasa HTML untuk membuat halaman web yang menarik. PHP mampu berjalan di atas beberapa platfrom seperti Windows, Unix serta varian Linux. Beberapa kelebihan PHP antara lain dapat membuat situs yang interaktif dengan forum diskusi, *guestbook* dan sebagainya, koneksitas yang baik dengan bermacam-macam database seperti Oracle, MySQL, PostgreSQL dan lain-lain.

Kode PHP diawali dengan tag <? atau <?php dan ditutup dengan tag ?>. File yang berisi tag HTML dan kode PHP diberi ekstensi. Berdasarkan ekstensi ini, pada saat file diakses, server akan tahu bahwa file tersebut mengandung kode PHP. Server akan menerjemahkan kode dan menghasilkan output dalam bentuk tag HTML yang akan dikirim ke browser klien yang mengakses file tersebut.

Script dasar PHP :

```
<?php echo "Hello World";
?>
```

Ketika skrip tersebut dijalankan di server, server akan mengembalikan ke browser klien dalam bentuk tag HTML. Sehingga ketika dilihat hasilnya di browser menggunakan menu *view source,* hasilnya adalah sebagi berikut :

Hasil script PHP di browser :

```
<html>
<body>Hello World</body>
</html>
```

2.2.14 Cascading Style Sheet (CSS)

Cascading Style Sheet (CSS) adalah salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga mempunyai bentuk lebih terstruktur dan seragam. Sama hal-nya *styles* dalam aplikasi pengolahan kata seperti Microsoft Word yang dapat mengatur beberapa style, misalnya *heading*, subbab, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa file. Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML, PHP dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna *body* teks, warna tabel, ukuran border, warna border, warna hyperlink, warna mouse over, spasi antar paragraf, spasi antar teks, margin kiri/kanan/atas/bawah, dan parameter lainnya. CSS adalah bahasa style sheet yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

CSS yang akan digunakan berasal dari Material Design Lite pada situs *https://getmdl.io*, dimana file CSS dapat dikostumasi dan dibangun sesuai kebutuhan aplikasi web. Penggunaan CSS dapat dilakukan melalui link *Hosted* atau dapat di-download langsung di situs mereka. Berikut adalah contoh penggunaan melalui link *Hosted* dengan menambahkan <*link*> pada halaman HTML atau PHP yang dibangun.

```
<link rel="stylesheet"
href="https://code.getmdl.io/1.3.0/material.indigo-pink.min.css">
```

2.2.15 Database MySQL

Perkembangannya disebut SQL yang merupakan kepanjangan dari *Structured Query Language*. SQL merupakan bahasa terstruktur yang khusus digunakan untuk mengolah database. SQL pertama kali didefinisikan oleh

American National Standards Institute (ANSI) pada tahun 1986. MySQL adalah sebuah sistem manajemen database yang bersifat open source. MySQL adalah pasangan serasi dari PHP. MySQL dibuat dan dikembangkan oleh MySQL AB yang berada di Swedia.



Sumber Gambar : www.mysql.com

Gambar 2.7 Logo MySQL

MySQL dapat digunakan untuk membuat dan mengola database beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah dan menghapus data yang berada dalam database. MySQL merupakan sisitem manajemen database yang bersifat relational. Artinya data-data yang dikelola dalam database akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan menjadi jauh lebih cepat. MySQL dapat digunakan untuk mengelola database mulai dari yang kecil sampai dengan yang sangat besar. MySQL juga dapat menjalankan perintah-perintah *Structured Query Language* (SQL) untuk mengelola database-database yang ada di dalamnya. Hingga kini, MySQL sudah berkembang hingga versi 5. MySQL 5 sudah mendukung *trigger* untuk memudahkan pengelolaan tabel dalam database.

Script SQL pada MySQL :

1. *Create*

   Digunakan untuk membuat tabel baru.

```
CREATE TABLE nama_tabel (
Kolom1 tipedata(ukuran) Option_Kolom,
Kolom2 tipedata(ukuran) Option_Kolom,
...
KolomN tipedata(ukuran) Option_Kolom,
) ENGINE=Nama_Engine;
```

Digunakan untuk membuat database baru.

```
CREATE DATABASE nama_database;
```

2. *Select*

Digunakan untuk menampilkan data dari tabel database.

```
SELECT * | {ekspresi}
FROM nama_tabel
[WHERE kondisi];
```

3. *Update*

Digunakan untuk merubah informasi/data pada tabel database.

```
UPDATE nama_tabel
SET kolom1=value [, kolom2=value...]
[WHERE kondisi];
```

4. *Delete*

Digunakan untuk menghapus record/data dari tabel database.

```
DELETE FROM nama_tabel
[WHERE kondisi];
```

5. *Insert*

   Digunakan untuk menambah informasi/data baru ke dalam tabel database.

```
INSERT INTO nama_tabel (kolom1 [,kolom2,..])
VALUES (value1 [, value2, value3...]);
[WHERE kondisi];
```

6. *Drop*

   Digunakan untuk menghapus tabel database.

```
DROP TABLE nama_tabel;
```

Database MySQL nantinya akan digunakan sebagai tempat penyimpanan data pada server dimana aplikasi *geofencing* dalam *monitoring* pengiriman kendaraan akan dibangun.

2.2.16 Adobe Dreamweaver

Adobe Dreamweaver adalah program untuk membuat, mendesain dan mengedit dokumen program secara visual dan mengelola halaman sebuah situs. *Dreamweaver* menyediakan banyak perangkat yang berkaitan dengan pengkodean dan fitur seperti HTML, CSS, JavaScript, PHP, ASP, ColdFusion, dan XML.



Sumber Gambar : Sceenshot

Gambar 2.8 Adobe Dreamweaver CS3

Sebagai editor web yang handal, Adobe Dreamweaver tentunya dilengkapi dengan kemampuan manajemen situs, yang memudahkan kita mengelola keseluruhan elemen yang ada dalam sebuah situs. Kita juga dapat melakukan evaluasi situs dengan melakukan pengecekan broken link, kompatibilitas browser, termasuk validasi *tag-tag* HTML dan CSS yang tidak sesuai dengan pedoman secara otomatis pada sebuah halaman web. Salah satu fitur penting yang akan digunakan adalah kemampuan untuk memanajemen situs di server secara *remote*, sehingga hasil setiap perubahan kode dapat dilihat secara langsung.

2.2.17 Android Studio

Android Studio adalah sebuah IDE untuk pengembangan aplikasi di platform Android. Sama seperti kombinasi antara Eclipse dan *Android Developer Tools* (ADT), Android Studio juga dapat di-download di situs resmi Android https://developer.android.com/studio/index.html. Saat ini Android Studio sudah berkembang menjadi kode editor kelas dunia dengan versi terbaru yaitu Android Studio 2.3.1 untuk OS Windows.



Gambar 2.9 Android Studio Logo

Sebelum munculnya Android Studio, SDK android selalu bundle bersama dengan Eclipse. Sekarang Android Studio menggunakan *IntelliJ IDEA Community Edition*. Selain itu, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas saat membuat aplikasi android, seperti sistem versi berbasis *Gradle* yang fleksible. Gradle adalah *build automation tool* yang dapat dikonfigurasi melalui DSL berbasis Groovy. Ini yang membedakan Gradle dari Ant atau Maven yang memakai XML. Penggunaan DSL berbasis *Groovy* menyebabkan *Gradle* lebih fleksibel dan dapat diprogram dengan mudah.