

# PEMANFAATAN RASPBERRY UNTUK PEMANTAUAN POSISI KENDARAAN (VEHICLE TRACKING)

Mohammad Hamdani<sup>1</sup> dan Amzar Sarlafa<sup>2</sup>

<sup>1</sup>Program Sarjana Program Studi Teknik Elektro, FTI-ISTN

email : [mhamdani@istn.ac.id](mailto:mhamdani@istn.ac.id)

<sup>2</sup>Spv CNO PT. TELKOMSEL, Jl. TB Simatupang no.2 Tanjung Barat, Kec. Jagakarsa Jaksel

email : [amzar.sarlafa@gmail.com](mailto:amzar.sarlafa@gmail.com)

## ABSTRAK :

Pada makalah ini dibahas tentang perancangan Vehicle Tracker yang berguna dalam melacak lokasi dari asset yang bergerak, dengan memanfaatkan penerima (receiver) GPS port USB dengan built in antenna, untuk mendapatkan koordinat. Raspberry sebagai pengendali (controller) dan modem seluler sebagai media penyalur informasi yang tersambung ke internet. Pengendali (controller) menggabungkan informasi titik koordinat tersebut dengan peta digital yang didapatkan dari server Google Map di internet.. Pada waktu client mengakses unit Vehicle Tracker melalui internet, controller (Raspberry) memberikan data koordinat 'real time' bersama dengan peta digital. Pengujian dilakukan dengan membandingkannya dengan GPS Standar dengan hasil yang masih dalam batas toleransi yang dipersyaratkan oleh GPS Internasional. Pengujian latency / delay untuk mengetahui lamanya pengiriman informasi dari alat ke client didapatkan hasil latency / delay dibawah 1 detik.

**Kata kunci :** Vehicle Tracker, GPS, latency, delay, Raspberry

## ABSTRACT :

In this paper, it is explained about the design of the Vehicle Tracker tool which is useful in tracking the location of movable assets, by utilizing a GPS receiver USB receiver with a built in antenna, to get coordinates, a Raspberry as a controller and a cellular modem as a media information channel connected to the internet. The controller combines these coordinate information with digital maps obtained from Google Map servers on the internet. When the client accesses the Vehicle Tracker unit via the internet, the controller (Raspberry) provides 'real time' coordinate data along with a digital map. Testing is done by comparing it with GPS Standards with results that are still within the tolerance limit required by GPS International. Latency / delay testing to find out how long the information is sent from the device to the client results in latency / delay under 1 second.

**Keywords :** Vehicle Tracker, GPS, latency, delay, Raspberry

## 1. PENDAHULUAN

Telekomunikasi pada saat ini telah mengalami beberapa perubahan paradigma, dimana telekomunikasi yang berbasis data lebih banyak digunakan dibandingkan dengan telekomunikasi suara yang mengalami puncak kejayaannya pada tahun 1980-an (analog) dan pada tahun 1990-an yang sudah mulai berbasis digital, yang ditandai dengan munculnya system komunikasi GSM. Subjek dan objek telekomunikasi pun sudah mengalami beberapa perubahan, yang awalnya telekomunikasi manusia kepada manusia, saat ini arah telekomunikasi bisa dari manusia ke mesin, mesin ke manusia ataupun mesin ke mesin. Begitu maraknya perangkat – perangkat baru

yang terintegrasi secara on line dengan system telekomunikasi. Istilah yang sering digunakan untuk perangkat seperti ini adalah Internet over Thing atau sering disingkat dengan IoT. Salah satu perangkat yang termasuk IoT adalah system pelacakan kendaraan yang saat ini sudah marak digunakan pada kendaraan roda empat maupun roda dua. Pelacak kendaraan (*vehicle tracker*) dapat membantu dalam pengamanan asset yang bergerak.

### 1.1 Internet of Things

Dikenal juga dengan singkatan IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus. Salah

satu penerapan IoT dalam pelacakan kendaraan adalah dengan menggunakan penerima (receiver) GPS sebagai sensornya.

### 1.2 Modem Seluler

Modem yang menggunakan jaringan seluler (GPRS, UMTS, HSPA, LTE, EVDO, WiMax, dll), dikenal sebagai modem nirkabel (kadang-kadang juga disebut modem selular). Modem nirkabel dapat tertanam di dalam laptop atau alat eksternal menggunakan port USB.

### 1.3 Raspberry Pi

Merupakan mikro komputer papan tunggal (single-board circuit; SBC) yang seukuran dengan kartu kredit yang dapat digunakan sebagai pengendali mikro (*micro controller*). Menerima input dari sensor, melakukan pengolahan data dan memberikan output, baik berupa data – data maupun eksekusi dari suatu algoritma yang diprogram oleh pengguna.



**Gambar 1.1** Unit Raspberry

Sumber : [www.raspberrypi.org](http://www.raspberrypi.org)

### 1.4 Global Positioning System

Adalah teknologi yang digunakan untuk menemukan posisi akurat di bumi dengan menggunakan sinyal satelit. Data yang didapatkan dengan sistem GPS diantaranya lintang, bujur, ketinggian di atas permukaan laut serta kecepatan pergerakan di permukaan bumi.

### 1.5 Bahasa Python, PHP dan HTML

Python adalah bahasa pemrograman interpretatif multiguna. Bahasa ini muncul pertama kali pada tahun 1991, dirancang oleh seorang bernama Guido van Rossum. Sampai saat ini Python masih dikembangkan oleh Python Software Foundation. PHP: "Hypertext Preprocessor", yaitu bahasa pemrograman yang digunakan pembuatan dan pengembangan sebuah situs web bersamaan dengan HTML yang merupakan bahasa untuk pembuatan halaman Web, dalam format ASCII. PHP

diciptakan oleh Rasmus Lerdorf pertama kali tahun 1994.

### 1.6 Android

Android adalah sistem operasi dengan sumber terbuka berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet, menggunakan antar muka berupa manipulasi langsung, menggunakan gerakan sentuh yang serupa dengan tindakan nyata, misalnya menggeser, mengetuk, dan mencubit untuk memanipulasi objek di layar, serta papan ketik virtual untuk menulis teks.

### 1.7 Delay dan Latency

Delay adalah waktu tunda saat paket yang diakibatkan oleh proses transmisi dari satu titik menuju titik lain yang menjadi tujuannya.

$$Delay \text{ rata - rata} = \frac{Total \text{ Delay}}{Total \text{ Paket Diterima}} \dots(1.1)$$

Latency merupakan waktu tunda sejak paket permintaan dikirim oleh pengirim sampai ke penerima dan di respon kembali oleh penerima ke pengirim awal yang berisi jawaban atas paket permintaan.

### 1.8 Wireshark

Wireshark merupakan salah satu tools atau aplikasi free license "Network Analyzer" atau penganalisa Jaringan, mulai dari proses menangkap paket-paket data atau informasi yang berlalu-lalang dalam jaringan, dan juga digunakan untuk *sniffing*.

### 1.9 Map Info

MapInfo merupakan software berbasis Sistem Informasi Geografis (SIG) yang memiliki fungsi untuk pengelolaan peta dalam bentuk data spasial ataupun berupa data MS Excel. Dengan bantuan Map Info, data koordinat, kecepatan dan waktu dapat dianalisa.

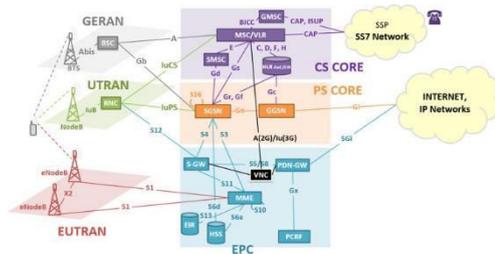
### 1.10 Kamera Digital

Gambar yang ditangkap oleh lensa, dilewatkan pada filter warna yang kemudian akan ditangkap oleh CCD atau sensor gambar. CCD merubah sinyal analog (gambar yang ditangkap oleh lensa) menjadi sinyal listrik. Pada CCD ini terdapat jutaan titik sensor yang dikenal dengan pixel. Semakin kecil sensor dan semakin banyak titik sensornya, maka akan semakin halus dan semakin tinggi resolusi gambar yang dihasilkan. Gambar diteruskan ke bagian pemroses gambar yang tugasnya

memproses semua data dari sensor CCD menjadi data digital berupa file format gambar, serta melakukan proses kompresi sesuai format gambar yang dipilih (RAW, JPEG, dan sebagainya).

**1.11 Komunikasi PS pada Jaringan Seluler**

Untuk tersambung ke jaringan internet, pengguna harus menggunakan layanan khusus yang disebut ISP (Internet Service Provider). Akses pelanggan (handphone) ke jaringan wireless (GSM/WCDMA/LTE), dimulai dengan mendapatkan kanal radio (media gelombang radio), otentikasi, pembangunan interkoneksi, fase komunikasi, pemutusan koneksi serta pencatatan meter (billing). Peralatan (devais) yang terkoneksi dengan menggunakan jaringan wireless dapat berupa modem (PPP – Point to Point Protocol), handset, tablet, aksesoris, mifi, netbook dan juga sensor – sensor IoT.



**Gambar 1.2 Arsitektur Jaringan Seluler**  
 Sumber : [www.cisco.com](http://www.cisco.com)

Pada Packet Switched Networks: Informasi dikirim dalam bentuk potongan informasi yang lebih kecil dan tiap potongan dapat melalui jalur yang berbeda. Komunikasi tiap paket dilakukan dalam bentuk hubungan yang berbeda, sehingga secara keseluruhan komunikasi berbentuk hubungan tidak tetap (connectionless oriented). Pertukaran informasi (message flow) pada jaringan 2G dan 3G berbeda dibandingkan dengan jaringan 4G

**1.12 Kecepatan, Jarak dan Waktu**

Kecepatan didefinisikan sebagai perubahan kedudukan setiap satuan waktu. Gerak Lurus Beraturan (GLB) adalah suatu gerak lurus yang mempunyai kecepatan konstan. Maka nilai percepatannya adalah  $a = 0$ . Gerakan GLB berbentuk linear dan nilai kecepatannya adalah hasil bagi jarak dengan waktu yang ditempuh.

$$v = \frac{s}{t} \dots\dots\dots (1.2)$$

Dengan

- $s$  = Jarak yang ditempuh (m, km)
- $v$  = Kecepatan (km/jam, m/s)
- $t$  = Waktu tempuh (jam, sekon)

Perhitungan jarak antar titik di permukaan bumi bisa didapatkan dengan mengkonversi hasil pengukuran yang didapatkan dari GPS dengan menggunakan rumus Pythagoras dengan dikalikan 111.600 meter (referensi dari software GIS Map Info) yaitu :

$$R = 111.600 \times \sqrt{(Lon_1 - Lon_2)^2 + (Lat_1 - Lat_2)^2} \dots(1.3)$$

**2. METODA**

**2.1 Rancang bangun dan realisasi perangkat**

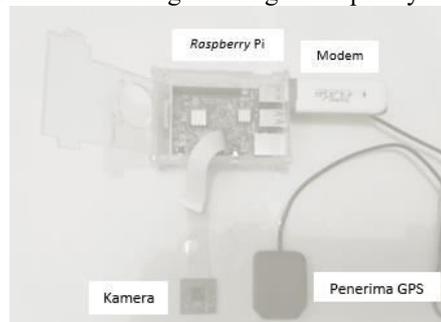
**2.1.1 GPS Tracker**

Data koordinat (longitude dan latitude) yang didapatkan dari penerima GPS (berbentuk antenna) dilewatkan ke pengendali (controller). Pengendali (controller) memproses data dan mengambil peta digital dari internet melalui modem sebagai media pengubungannya. Pengendali (controller) mengkombinasikan data koordinat dengan peta digital secara internal. Pada waktu unit Vehicle/GPS Tracker dihidupkan, secara otomatis pengendali (controller) memberikan identitas dirinya ke Router IoT. Client (pengguna) mengakses Router IoT melalui internet untuk mendapatkan informasi mengenai lokasi kendaraan. Komponen penyusun pelacak lokasi ini adalah :

- a) GPS Receiver dan Antena
- b) Kamera Raspberry Pi
- c) Controller Raspberry Pi
- d) Modem
- e) Web Server Lokal
- f) Client

**2.1.2 Integrasi Hardware**

Hardware yang diintegrasikan adalah modem, penerima GPS, dan kamera. Semua hardware dihubungkan dengan Raspberry.



**Gambar 2.1 Hardware Vehicle Tracker**  
 Catatan : telah diolah Kembali

**2.1.3 Instalasi OS dan Software**

Raspberry sering dinamakan SBC (Single Board Computer), dimana semua hardware-nya terpasang pada papan PCB yang sama. Raspberry juga memerlukan instalasi Operating System dan software supaya bisa bekerja. Operating System dan software disimpan pada kartu micro SD, yang dipasang pada slot yang tersedia. Kemudian kartu micro SD ini dikonfigurasi dan diisi paket instalasi dengan menggunakan software khusus yang dinamakan “NOOBS (New Out Of Box Software)”. Setelah kartu micro SD dipasang dan Raspberry diberi catuan, proses instalasi Operating System dapat dilakukan.

**2.1.4 Instalasi Receiver GPS**

Konektor pada ujung kabel antenna menggunakan port USB, yang disambungkan ke port USB pada Raspberry. Penerima (receiver) GPS dikenali pada Raspberry sebagai “U-Blox AG” dengan identitas pabrik pembuat dan produk “1546:01a7”.

```
lsusb
ID 046d:c001 Logitech, Inc. N48/M-BB48 [FirstMouse Plus]
ID 046d:c002 Logitech, Inc. Classic Keyboard-200
ID 1546:01a7 U-Blox AG
ID 0424:ec00 Standard Microsystems Corp. SMC9512/9514 Fast Ethernet Adapter
ID 0424:9514 Standard Microsystems Corp.
ID 1d6b:0002 Linux Foundation 2.0 root hub
```

**Gambar 2. 2 Identifikasi Hardware Receiver GPS**

Catatan : telah diolah kembali

Identitas hardware tersebut secara otomatis dipetakan dengan identitas software device ttyACM0.

```
pi@raspberrypi:~$ ls /dev/tty*
/dev/tty /dev/tty19 /dev/tty3 /dev/tty40 /dev/tty51 /dev/tty62
/dev/tty0 /dev/tty2 /dev/tty30 /dev/tty41 /dev/tty52 /dev/tty63
/dev/tty10 /dev/tty20 /dev/tty31 /dev/tty42 /dev/tty53 /dev/tty64
/dev/tty11 /dev/tty21 /dev/tty32 /dev/tty43 /dev/tty54 /dev/tty65
/dev/tty12 /dev/tty22 /dev/tty33 /dev/tty44 /dev/tty55 /dev/tty66
/dev/tty13 /dev/tty23 /dev/tty34 /dev/tty45 /dev/tty56 /dev/tty67
/dev/tty14 /dev/tty24 /dev/tty35 /dev/tty46 /dev/tty57 /dev/tty68
/dev/tty15 /dev/tty25 /dev/tty36 /dev/tty47 /dev/tty58 /dev/tty69
/dev/tty16 /dev/tty26 /dev/tty37 /dev/tty48 /dev/tty59 /dev/tty70
/dev/tty17 /dev/tty27 /dev/tty38 /dev/tty49 /dev/tty60
/dev/tty18 /dev/tty28 /dev/tty39 /dev/tty50 /dev/tty61
pi@raspberrypi:~$
```

**Gambar2.3 Mapping Id Hardware ke Software**

Catatan : telah diolah kembali

**2.1.5 Instalasi Modem**

Modem diisi kartu SIM (Subscriber Identity Modul) yang telah terdaftar pada salah satu penyedia layanan internet ISP, dalam hal ini, operator seluler. Modem dihubungkan ke salah satu slot USB yang tersedia. Modem terdeteksi menggunakan hardware “Qualcomm, Inc. Siemens SG75”, dengan identitas produk “05c6:6000”.

```
lsusb
ID 046d:c001 Logitech, Inc. N48/M-BB48 [FirstMouse Plus]
ID 05c6:6000 Qualcomm, Inc. Siemens SG75
ID 046d:c315 Logitech, Inc. Classic Keyboard 200
ID 0424:ec00 Standard Microsystems Corp. SMC9512/9514 Fast Ethernet Adapter
ID 0424:9514 Standard Microsystems Corp.
ID 1d6b:0002 Linux Foundation 2.0 root hub
```

**Gambar2. 4 Identifikasi Hardware Modem**

Catatan : telah diolah kembali

Identitas diberikan secara software untuk modem dengan nama “/dev/ttyUSB0”.

```
pi@raspberrypi:~$ ls /dev/tty*
/dev/tty /dev/tty10 /dev/tty20 /dev/tty30 /dev/tty40 /dev/tty50 /dev/tty60
/dev/tty0 /dev/tty1 /dev/tty11 /dev/tty21 /dev/tty31 /dev/tty41 /dev/tty51 /dev/tty61
/dev/tty10 /dev/tty20 /dev/tty30 /dev/tty40 /dev/tty50 /dev/tty60
/dev/tty11 /dev/tty21 /dev/tty31 /dev/tty41 /dev/tty51 /dev/tty61
/dev/tty12 /dev/tty22 /dev/tty32 /dev/tty42 /dev/tty52 /dev/tty62
/dev/tty13 /dev/tty23 /dev/tty33 /dev/tty43 /dev/tty53 /dev/tty63
/dev/tty14 /dev/tty24 /dev/tty34 /dev/tty44 /dev/tty54 /dev/tty64
/dev/tty15 /dev/tty25 /dev/tty35 /dev/tty45 /dev/tty55 /dev/tty65
/dev/tty16 /dev/tty26 /dev/tty36 /dev/tty46 /dev/tty56 /dev/tty66
/dev/tty17 /dev/tty27 /dev/tty37 /dev/tty47 /dev/tty57 /dev/tty67
/dev/tty18 /dev/tty28 /dev/tty38 /dev/tty48 /dev/tty58 /dev/tty68
pi@raspberrypi:~$
```

**Gambar 2. 5 Identifikasi Software Modem**

Catatan : telah diolah kembali

Dengan aplikasi “Modem Dialer” pada Raspberry, modem berhasil terhubung ke internet, ditandai dengan alokasi IP yang diberikan oleh jaringan operator selular (ISP).

```
TX packets:65984 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1
RX bytes:9558699 (9.1 MiB) TX bytes:9558699 (9.1 MiB)

ppp0 Link encap:Point-to-Point Protocol
inet addr:172.17.184.5 P-t-P:10.64.64.64 Mask:255.255.255.255
UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1
RX packets:17 errors:0 dropped:0 overruns:0 frame:0
TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:3
RX bytes:828 (828.0 B) TX bytes:169 (169.0 B)

wlan0 Link encap:Ethernet HWaddr b8:27:eb:b5:0c:48
inet addr:192.168.43.40 Bcast:192.168.43.255 Mask:255.255.255.0
inet6 addr: fe80::ba27:ebff:fb65:c48764 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:29780 errors:0 dropped:3 overruns:0 frame:0
TX packets:29217 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:2650691 (2.5 MiB) TX bytes:4514050 (4.3 MiB)
```

**Gambar 2 6 Alokasi IP untuk Modem dari ISP Seluler**

Catatan : telah diolah kembali

Modem mendapatkan alokasi IP 172.17.184.5 dengan menggunakan *protocol* PPP (*Point to Point Protocol*).

**2.1.6 Instalasi Kamera**

Kamera pada Raspberry dihubungkan dengan slot khusus yang tersedia pada papan PCB Raspberry dengan menggunakan kabel flexible. Pengaturan kamera dan penyimpanan hasil pengambilan foto atau video selanjutnya dilakukan dengan perintah pada aplikasi Python.

**2.2 Pengambilan dan Penyimpanan Data**

Sumber data dihasilkan dari penerima (*receiver*) GPS. Data yang didapatkan dari penerima (*receiver*) GPS diantaranya :

- Bujur timur (*longitude*), dalam satuan derajat
- Lintang selatan (*latitude*), dalam satuan derajat
- Ketinggian dari permukaan air (*altitude*), dalam satuan meter
- Kecepatan objek bergerak, dalam satuan kilometer per jam (km/Jam)
- Track atau arah perjalanan dilakukan

```
{"Date":"2018-05-02","Time":"16:13:23","lat":-6.30452273333,"lon":106.846127667,"speed":1.335212}
{"Date":"2018-05-02","Time":"16:13:24","lat":-6.30452283333,"lon":106.846118667,"speed":2.335212}
```

**2.3 Instalasi Webserver**

Supaya data dapat dibaca oleh *Client* pada praman yang ada di device-nya, maka

Raspberry dijadikan sebagai Webservice yang dapat diakses oleh *Client*. Agar *Raspberry* memiliki kemampuan sebagai Webservice, maka dilakukan instalasi software *Webservice* menggunakan aplikasi *Apache* atau *Ngix*.

#### 2.4 Setup Remote Akses Raspberry

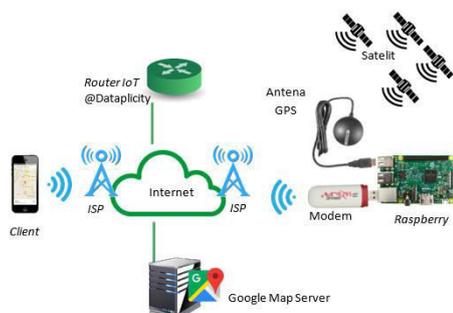
Hubungan komunikasi antara unit GPS / Vehicle Tracker dan client dibangun dengan bantuan router IoT virtual dari “Dataplicity”. Pada server Dataplicity dilakukan pendaftaran untuk Raspberry terkait, dengan memasukan alamat email dan password yang nantinya akan diinputkan pada waktu instalasi software “Dataplicity Agent” di dalam Raspberry. Fungsi dari software ini membuat koneksi otomatis ke router IoT virtual yang berada pada server Dataplicity pada waktu Raspberry dihidupkan.

#### 2.5 API Google Map

Untuk mendapatkan peta digital yang selalu update pada waktu *GPS/ Vehicle Tracker* menampilkan rute perjalanan, diperlukan koneksi ke server Google Map. Untuk dapat mengakses peta digital tersebut diperlukan suatu kode unik dari Google yang dinamakan *API (Application Programming Interface)* dapat berupa beberapa angka, huruf dan karakter secara acak. Kode ini didapatkan dengan cara melakukan pendaftaran API Google Maps. Kode unik dikirim melalui email. API Google Maps memberikan metode kepada developer /programmer untuk menyematkan peta digital Google Maps ke dalam laman web atau mengambil data dari Google Maps, dan memungkinkan penggunaan sederhana atau lebih lanjut. Penyematkan dapat dimasukkan dalam tag HTML.

#### 2.6 Interworking Sistem

Perangkat penerima (receiver) GPS + antenna mendapatkan catuan dari Raspberry, pada saat dihidupkan. Waktu yang ada pada penerima (receiver) GPS akan disesuaikan dengan jam (waktu) yang dikirimkan dari satelit, dimana akurasi mencapai ukuran “nano detik”



Gambar 2.7 Arsitektur GPS / Vehicle Tracker

Catatan : telah diolah kembali

Dengan memperhitungkan waktu tempuh sinyal GPS dari tiga satelit dan posisi pastinya di langit, penerima (receiver) GPS bisa menentukan posisi dalam tiga dimensi yaitu timur, utara dan ketinggian. Untuk menghitung waktu sinyal GPS masuk, penerima (receiver) GPS perlu mengetahui waktunya dengan sangat akurat, hal ini dicapai dengan bantuan satelit keempat.

Data – data dari penerima (receiver) GPS yang berupa kalimat – kalimat GPS standar NMEA kemudian diinput ke kontroler Raspberry untuk proses pengolahan data selanjutnya. Pembacaan dan penerjemahan dilakukan dengan menggunakan script bahasa Python.

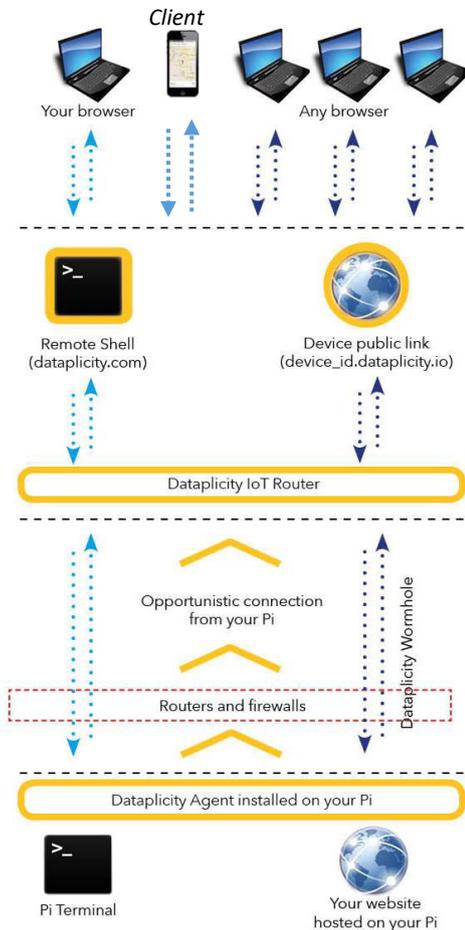
Modem mendapatkan catuan dari kontroler *Raspberry* melalui port USB yang sekaligus sebagai media pertukaran data. Untuk dapat terhubung dengan internet, modem akan melakukan *attach* ke jaringan seluler (ISP). Setelah koneksi terbangun ke jaringan internet, modul program “*Dataplicity Agent*” yang ditanam pada *Raspberry (controller)* akan membangun hubungan komunikasi *HTTPS* dengan *router IoT virtual* yang berada pada *Dataplicity*.

Saat terhubung ke Remote Shell atau ke antarmuka web yang diarahkan melalui URL Wormhole pada perangkat Client, koneksi dari Client akan diarahkan ke Raspberry yang merupakan pengendali dan sekaligus webserver dari unit GPS / Vehicle Tracker melalui router IoT virtual. Traffic disalurkan menggunakan koneksi websocket terenkripsi. IoT virtual. Dalam *Raspberry* (pengendali), data dari penerima (receiver) GPS yang telah didapatkan, diterjemahkan dengan *software* bahasa pemrograman *Python* dan disimpan ke dalam *SD Card*.

Dengan menggunakan bahasa pemrograman *web PHP*, formatnya diubah menjadi bentuk format file \*.php yang diteruskan ke dalam dalam *Webserver Apache* agar dapat diakses oleh *Client* dengan menggunakan *browser* (pramban) internet pada handset / handpone nya.

Dengan bantuan kode *PHP* yang berisi kode *API Google Map*, *Web Server* di dalam *Raspberry* akan menghubungi *Server Google* untuk mendapatkan peta digital (*Google Map*). Setelah berhasil terhubung, peta digital segera diunduh dari *Server Google* ke *Web Server* lokal. Di sisi lain, *Web Server* melalui kode *PHP* mengumpulkan data koordinat yang tersimpan pada *SD Card* dan menggabungkannya dengan peta digital yang telah didapatkan, untuk kemudian siap

ditampilkan dalam *Web Server*. Dengan demikian *Client* yang mengakses *Web Server* dalam *Raspberry* dapat melihat tampilan peta digital dan titik koordinat dari unit *Vehicle (GPS) Tracker* yang telah diolah dan digabungkan.



**Gambar 2.8 Komunikasi Vehicle Tracker dengan Client**

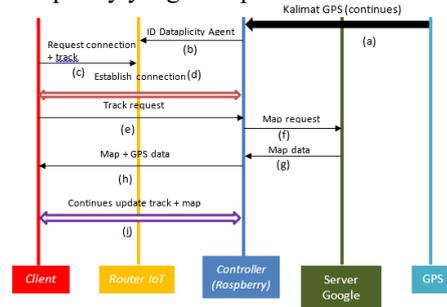
Sumber : www.dataplicity.com

**2.8 Alur Komunikasi**

GPS memberikan kalimat – kalimat GPS ke unit Controller dengan format ASCII yang dipisahkan dengan tanda ‘koma’. Isi dari kalimat – kalimat GPS yaitu tipe data, penanggalan, waktu, lintang utara, bujur timur, kecepatan, ketinggian dari permukaan laut dan juga track. Data dikirim unit GPS setiap detik secara berkelanjutan. Controller menerima kalimat – kalimat GPS dan memilah nya (parse).

Agen *dataplicity* otomatis akan mengontak *router IoT* (server) dengan *protocol HTTP*, untuk dihubungkan dengan *Client* yang mengakses server dari arah lain. Sebelum adanya permintaan dari *Client*, *router IoT* akan

terus mempertahankan komunikasi dengan agen *dataplicity* yang terdapat dalam Controller.



**Gambar 3.7 Communication flow Client ke Unit Tracker**  
Catatan : telah diolah kembali

Pada saat *Client* mengontak *router IoT* (server) *dataplicity* melalui jaringan internet menggunakan *protocol HTTP*, yang berisi informasi nama user dan password. Bila cocok dengan user dan password yang didapatkan dari agen *dataplicity* (Controller) sebelumnya, maka *router IoT* (server) akan menghubungkan *Client* dengan agen *dataplicity* (Controller). Permintaan dari *Client* selain konektivitas, juga berisi permintaan ‘track’ kepada *Vehicle (GPS) Tracker*.

Komunikasi antara *Client* dengan unit *Vehicle (GPS) Tracker* terbangun secara langsung dan transparan. *Router IoT* berfungsi sebagai jembatan penghubung. Permintaan ‘track’ langsung dari *Client* ke *Vehicle (GPS) Tracker*, melalui *protocol HTTP*. *Vehicle (GPS) Tracker* bertindak sebagai ‘Web Server’.

*Vehicle (GPS) Tracker* akan melakukan permintaan peta digital (‘Map’) ke server Google Map, dengan titik koordinat tertentu sesuai dengan posisi koordinat *Vehicle (GPS) Tracker* berada. Peta digital dikirim oleh server Google Map sesuai permintaan *Vehicle (GPS) Tracker*, untuk dikombinasikan dengan titik koordinat yang di dapatkan dari GPS.

*Vehicle (GPS) Tracker* mengirimkan koordinat dan Peta digital secara bersama ke *Client*, dengan *protocol HTTP* dilakukan terus – menerus dengan periode 1 detik. Pengiriman data ini berhenti ketika *Client* menonaktifkan atau menutup jendela *browser* internet.

**2.9 Implementasi Alat (Vehicle Tracker)**

- a. Tracking mobile gensek pada waktu mobilisasi gensek ke site – site pada perusahaan penyedia layanan seluler
- b. Tracking paket perusahaan ekspedisi Untuk keamanan, mengetahui rute yang dilalui dan memprediksi kedatangan paket
- c. Tracking mobil rental sebagai antisipasi pencurian

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Pengujian Controller

Controller harus bisa memberikan catuan untuk komponen yang tersambung dengannya, yaitu penerima GPS, modem dan kamera. Untuk pengujian dilakukan dengan memasang layar monitor komputer dan melihat dashboard yang ditampilkan. Tampilan dashboard raspberry terlihat pada gambar 3.1

GPS LOCATOR DASHBOARD	
DATE :	2018-05-02
TIME :	17:15:53
LON :	106.84619
LAT :	-6.30447
SPD :	0.6
amz(c)2018	

Gambar 3.1 Tampilan Dashboard

Catatan : telah diolah kembali

#### 3.2 Pengujian Penerima (Receiver) GPS

Pengujian penerima (receiver) GPS dilakukan untuk memastikan bahwa GPS telah bekerja dengan baik, karena hal ini akan mempengaruhi keakuratan koordinat yang dibaca oleh alat. Jika pembacaan data dari penerima GPS error maka posisi tidak akan terbaca oleh client. Titik – titik koordinat yang terbaca pada GPS, terlihat pada tabel 3.1

Tabel 3.1 Output Penerima GPS

Lokasi	Latitude	Longitude	Cuaca
Gedung TTC TBS	106,8457527	-6,30453433	Berawan
Jl. TBS no.7	106,8428205	-6,30386866	Berawan
Jl. Tanjung Barat no.99	106,8402928	-6,3047145	Berawan
St Tanjung Barat	106,83912	-6,30850183	Berawan
Indomart Tj Barat	106,8385618	-6,3117255	Berawan
Perempatan Condet	106,8555597	-6,30257883	Mendung
Rancho	106,8509175	-6,30459	Mendung
ACC Rancho	106,8487858	-6,30447033	Mendung
Tanjung Barat Indah	106,8464735	-6,305035	Mendung
Tanjung Barat Indah	106,8482072	-6,30517867	Mendung

Catatan : telah diolah kembali

#### 3.3 Pengujian Modem

Modem memiliki peranan penting dalam membangun komunikasi antara alat dengan server di internet, dan juga komunikasi dengan client sebagai pemantau / pengguna. Pengujian modem dengan melakukan test ping ke internet. Hasil test ping terlihat pada tabel 3.2.

Tabel 3.2 Pengujian Modem Seluler

Situs	IP Public	Ltney (ms)	St
google.com	74.125.200.100	58,9	OK
facebook.com	157.240.13.35	62	OK
kompas.co.id	202.146.4.100	39,3	OK
detik.com	103.49.221.211	36,4	OK
youtube.com	74.125.68.190	61,57	OK
istn.ac.id	45.112.125.198	38,4	OK
dataplicity.com	104.25.102.106	375,1	OK
telkomsel.co.id	202.3.208.158	32	OK
Instagram.com	34.197.43.75	429,4	OK

Catatan : telah diolah kembali

#### 3.4 Pengujian Akses Alat

Dengan menggunakan aplikasi browser internet klien, alat dapat diakses melalui alamat situs "<https://unsummed-beetle-9947.dataplicity.io/>". Untuk memastikan alat dapat diakses dari berbagai device maka pengujian yang dengan diakses dari beberapa device, untuk memastikan bahwa akses dapat dilakukan dari berbagai jenis device. Hal ini terlihat pada tabel 3.3

Tabel 3.3 Pengujian Akses dari Client

No	Device Client	OS	IP Client	St
1	Smart Phone Xiaomi R4	Android	114.124.198.145	OK
2	Smart Phone Xiaomi R4	Android	202.3.212.201	OK
3	PC 14" Screen	Windows	202.3.212.134	OK
4	PC 14" Screen	Linux	202.3.212.129	OK
5	Iphone 5	ios	202.3.212.201	OK
6	Smart Phone Redmi3	Android	114.124.203.54	OK

Catatan : telah diolah kembali

#### 3.5 Pengujian Akurasi Vehicle Tracker

Pengujian secara stasionary (tidak bergerak), dilakukan di beberapa titik dengan jarak setiap 50 meter, dengan cara membandingkannya dengan GPS standar yaitu "Garmin etrek 10", hasil terlihat pada tabel 3.4

Tabel 4.4 Akurasi Longitude

Lokasi	Lon GPS Tracker	Lon GPS Etrek10	Margin Err
Gedung TTC TBS	106,8457527	106,84577	0,000017
Jl. Simatupang no.7	106,8428205	106,84282	0,000000
Jl. Tanjung Barat no.99	106,8402928	106,84028	0,000013
Stasiun Tanjung Barat	106,83912	106,83912	0,000000
Rancho	106,8509175	106,85089	0,000027

Catatan : telah diolah kembali

Berdasarkan pengukuran longitude yang dibandingkan dengan GPS Standar, di dapatkan penyimpangan (error) rata - rata sebesar 0,000021 derajat bujur timur atau bila dikonversi ke dalam satuan meter sebesar 2,33 meter, sampel pengujian pada tabel 4.4

(dihitung dengan bantuan software “Map Info”). Adapun pada pengukuran *latitude*, didapatkan penyimpangan (error) rata – rata sebesar 0,000015 derajat atau bila dikonversi ke dalam satuan meter sebesar 1,65 meter. Sampel pengujian tercantum pada tabel 3.5.

Tabel 3.5 Akurasi Latitude

Lokasi	Lat GPS Tracker	Lat GPS Etrek10	Margin Err
Gedung TTC TBS	6,30453433	-6,30455	0,000016
Jl. Simatupang no.7	6,30386867	-6,30384	0,000029
Jl. Tanjung Barat no.99	-6,3047145	-6,30471	0,000005
Stasiun Tanjung Barat	6,30850183	-6,30847	0,000032
Rancho	-6,304599	-6,30457	0,000029

Catatan : telah diolah Kembali

### 3.6 Pengujian Kesesuaian Data

Pengujian kesesuaian data yang dikirim dari unit *Vehicle / GPS Tracker* sampai diterima oleh *Client* diperlukan untuk memastikan data koordinat dikirim dengan benar dan tepat, hal ini terlihat pada tabel 3.6

### 3.7 Pengujian Delay

Pada waktu kendaraan bergerak, terjadi perubahan koordinat yang di-update setiap satu detik (periode sampling). Proses pengolahan dan pengiriman data membutuhkan waktu tunda (*delay*). Pengujian *delay* pengiriman data diperlukan untuk mengetahui apakah data koordinat yang dikirim dari *Vehicle Tracker* masih dalam batas toleransi daripada perubahan titik koordinat yang dikukur dibandingkan dengan waktu sampling. Penghitungan dimulai pada saat permintaan data kemudian dibalas oleh *Server (Vehicle Tracker)*.

Tabel 3.6 Pengujian Kesesuaian Data

Vehicle / GPS Tracker (Pengirim)		Client (Penerima)		Ket
Long	Lat	Long	Lat	
106,8458	-6,30453	106,8458	-6,30453	Sesuai
106,8428	-6,30387	106,8428	-6,30387	Sesuai
106,8403	-6,30471	106,8403	-6,30471	Sesuai
106,8391	-6,3085	106,8391	-6,3085	Sesuai
106,8386	-6,31173	106,8386	-6,31173	Sesuai
106,8378	-6,31634	106,8378	-6,31634	Sesuai

Catatan : telah diolah kembali

Tabel 3.7 Pengujian waktu tunda (delay)

No.	Source	Destination	Length (byte)	Delay (sec)
1	52.72.155.122	172.21.70.6	82	0,284
2	52.72.155.122	172.21.70.6	82	0,285
3	52.72.155.122	172.21.70.6	82	0,269
4	52.72.155.122	172.21.70.6	82	0,275
5	52.72.155.122	172.21.70.6	82	0,265
6	52.72.155.122	172.21.70.6	82	0,263
7	52.72.155.122	172.21.70.6	82	0,272
8	52.72.155.122	172.21.70.6	82	0,278
9	52.72.155.122	172.21.70.6	82	0,266

Catatan : telah diolah kembali

### 3.8 Analisa Akurasi Vehicle Tracker

Digunakan rumus (1.3) untuk mengetahui penyimpangan pengukuran *Vehicle Tracker* dengan GPS Standar. Pada kondisi berawan dengan kecendrungan cerah, didapatkan penyimpangan rata – rata dari *Vehicle Tracker* sebesar 1,76 meter. Nilai maksimum penyimpangan sebesar 3,55 meter. Nilai minimum penyimpangan (paling akurat) sebesar 0,41 meter.

Tabel 4.8 Analisa Akurasi Vehicle Tracker

Vehicle / GPS Tracker		GPS Standar Etrek		Akurasi (meter)
Lon	Lat	Lon	Lat	
106,8458	-6,30453	106,8458	-6,30455	2,61
106,8428	-6,30387	106,8428	-6,30384	3,20
106,8403	-6,30471	106,8403	-6,30471	1,52
106,8391	-6,3085	106,8391	-6,30847	3,55
106,8386	-6,31173	106,8386	-6,31172	1,46
106,8378	-6,31634	106,8378	-6,31634	1,16

Catatan : telah diolah kembali

Pada kondisi mendung didapatkan akurasi GPS Tracker sekitar 4,21 meter, jika dibandingkan pada kondisi cerah / berawan mengalami penyimpangan nilai lebih jauh sekitar 2,45 meter. Sedangkan rata – rata akurasi pada dua kondisi tersebut sekitar 2,99 meter. Nilai ini masih masuk dalam batas toleransi standar GPS Amerika Serikat yaitu maksimum sebesar 7,8 meter (*gps.gov*).

### 3.9 Analisa Waktu Tunda (Delay)

Agar waktu pembacaan pada *Client* sesuai dengan *sampling* yang dikirimkan, maka *delay* pengiriman data koordinat dari unit *Vehicle Tracker* harus lebih kecil atau sama dengan 1 detik. Dengan menggunakan software “*Wireshark*”, akan didapatkan pengukuran *delay* yang terjadi pada pengiriman data koordinat antara unit *Vehicle Tracker* (sebagai pengirim) dengan *client* sebagai penerima.

Tabel 3.9 Analisa Delay

No.	Source	Destination	Delay (sec)
1	52.72.155.122	172.21.70.6	0,284
2	52.72.155.122	172.21.70.6	0,285
3	52.72.155.122	172.21.70.6	0,269
4	52.72.155.122	172.21.70.6	0,275
5	52.72.155.122	172.21.70.6	0,265
6	52.72.155.122	172.21.70.6	0,263
	52.72.155.122	172.21.70.6	0,272
8	52.72.155.122	172.21.70.6	0,278
9	52.72.155.122	172.21.70.6	0,266
	Average		0,276
	Max		0,313
	Min		0,263

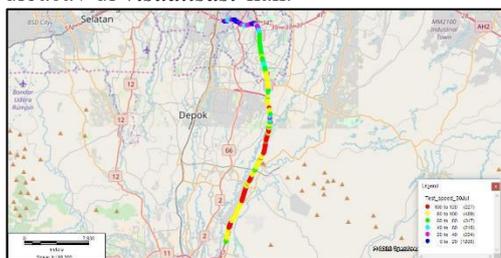
Catatan : telah diolah kembali

Rata – rata pengiriman data dari unit GPS Tracker ke *Client* sebesar 0,276 detik, sedangkan *delay* terbesar (maksimum) sebesar 0,313 detik. Dengan demikian *delay* pengiriman data koordinat dari unit *Vehicle Tracker* ke *Client*, kurang dari 1 detik. Hal ini dapat

dikatakan bahwa *delay* dari perangkat ini masih memadai untuk system *Vehicle Tracker*, dimana *client* akan mendapatkan data yang sama dengan yang didapatkan untuk *Vehicle Tracker*, dengan waktu yang hampir sama dengan waktu pengambilan data.

**3.10 Analisa *Vehicle Tracker* Bergerak**

Data perjalanan *Vehicle Tracker* dapat di-plot ke dalam bentuk peta yang lebih mudah dibaca / di-visualisasi-kan.



**Gambar 4. 1 Plot Track *Vehicle Tracker* pada Map Info**

Catatan : telah diolah kembali

Untuk mendapatkan jarak antar titik secara teoritis digunakan rumus (1.2), GLB (Gerak Lurus Beraturan). Adapun berdasarkan pengukuran koordinat, akan didapatkan jarak antar dua titik koordinat yang saling berurutan, sesuai dengan rumus (1.1). Kedua penghitungan ini akan dibandingkan dan dihitung marginnya. Nilai kecepataannya disesuaikan dengan kecepatan dari hasil pengukuran supaya didapatkan nilai yang sepadan (*peer to peer*). Dengan pemanfaatan *software "Microsoft Excel"*, hasil perhitungan berdasarkan rumus terlihat pada tabel 3.10.

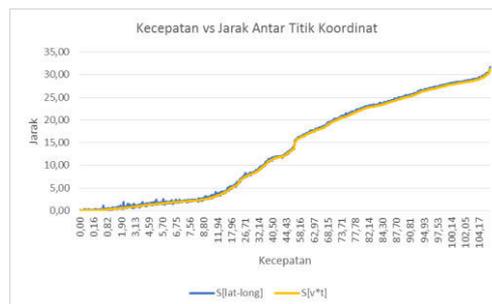
**Tabel 3.10 Jarak Titik Koordinat vs Teoritis**

Lat	Long	S <sub>[lat-long]</sub> (m)	Kec (v)	S <sub>[v*t]</sub> (m)	Margin (meter)
-6.54578	106.84968	0,00	73,38	-	-
-6.54561	106.84974	20,66	73,66	20,46	0,20
-6.54543	106.84980	20,75	73,63	20,45	0,29
-6.54525	106.84986	20,69	73,71	20,48	0,21
-6.54508	106.84992	20,71	74,06	20,57	0,14
-6.54490	106.84998	20,82	74,19	20,61	0,21
-6.54473	106.85004	20,84	74,36	20,66	0,18
-6.54455	106.85010	20,91	74,62	20,73	0,18
-6.54437	106.85016	20,97	74,84	20,79	0,18
-6.54419	106.85022	21,07	74,98	20,83	0,25
-6.54401	106.85028	21,03	74,93	20,81	0,22
-6.54383	106.85034	21,22	75,62	21,00	0,21
-6.54365	106.85039	21,34	76,19	21,16	0,17

Catatan : telah diolah Kembali

Nilai jarak secara teoritis (kolom "S<sub>[s\*v]</sub>"), didapatkan dari hasil perkalian antara kecepatan dengan waktu sampling (1 detik). Nilai kecepatan disamakan dengan kecepatan dari hasil pengukuran. Untuk mendapatkan hasil dalam satuan meter, maka satuan kecepatan dalam km/jam dirubah menjadi m/ det, dimana 1 km/jam = 0,2778 m/det.

Dari total 2683 sampling yang diperoleh, dihasilkan margin rata – rata sebesar 0,31 meter. Nilai ini masih dalam batas toleransi.



**Gambar 3.2 Kecepatan vs Jarak (Teoritis vs Jarak Antar Titik Koordinat)**

Catatan : telah diolah Kembali

**4. SIMPULAN**

Dari pengujian dan analisis sistem yang telah dilakukan, dapat diambil kesimpulan sebagai berikut :

1. Sistem *Vehicle / GPS Tracker* atau pelacakan kendaraan dapat bekerja dengan baik dimana *client / pemonitor* mendapatkan informasi lokasi yang '*real time*' dari unit pelacak dengan *device* yang berbasis perangkat android kapan pun dan dimanapun, baik dalam kondisi bergerak ataupun diam, dengan syarat terhubung ke jaringan *internet*.
2. Perangkat ini dapat bekerja pada kondisi cerah, berawan maupun mendung, dengan menunjukkan hasil pengukuran yang masih dalam toleransi dengan penyimpangan rata – rata sebesar 4,21 meter, sedangkan batas penyimpangan yang diijinkan sebesar 7,8 meter (*gps.gov*).
3. Waktu tunda (*delay*) pengiriman data dari unit *Vehicle / GPS Tracker* ke *client* lebih singkat dibandingkan periode *sampling* (pencuplikan data), dengan nilai rata – rata sebesar 0,276 detik, nilai maksimum sebesar 0,313 detik. Adapun periode *sampling* sebesar 1 detik.
4. Pada sisi *client*, dapat digunakan berbagai *device* baik berupa komputer (*PC*), *handphone*, *tablet* dengan berbagai jenis *operating system* (*android, windows, Mac, Linux*).

**DAFTAR PUSTAKA**

- [1]. Abidin, HZ. 2002. *Survey dengan GPS*. Jakarta : Pradnya Paramita.
- [2]. Arief, MR. 2011. *Pemrograman Web Dinamis Menggunakan PHP dan MySQL*. Yogyakarta : Andi Offset.
- [3]. Behmann, F.,Wu, K. 2015. *Collaborative Internet of Things (C-IoT): for Future Smart Connected Life and Business*. Texas USA : Wiley.
- [4]. Bowes, Pitney. 2013. *MapInfo Professional® 12.0 Userguide*. New York (USA) : Pitney Bowes Software Inc.

- [5]. Cayongkun. 2013. *Pengertian Modem GSM dan Modem GPRS* di <http://cayongkun.blogspot.co.id> (akses 17 Juli 2018)
- [6]. Chappell, Laura. 2012. *Wireshark Network Analysis The Official Wireshark Certified Network Analyst Study Guide*. Nevada USA : Chappell University”.
- [7]. Grimes, JG.2008. *Global Positioning System Standard Positioning Service Performance Standar*. Washington DC : US Defense.
- [8]. Halfacree, G., Upton, E. 2016. *Raspberry Pi User Guide, 4th Edition*. Hoboken USA : Wiley
- [9]. Hill, Kelly. 2016. *Understanding LTE call flow in-depth* di <https://www.rerwireless.com> (diakses pada 29 Januari 2019)
- [10]. Juhara ZP. 2016. *Panduan Lengkap Pemrograman Android*. Yogyakarta : Andi Offset.
- [11]. Kaplan, ED., CJ. Hegarty. 2006. *Understanding GPS Principles and Applications Second Edition*. Boston London : Artech House.
- [12]. Kasera, Sumit., Narang, Nishit. 2004. *3G Network Architecture, Protocols and Procedures*. New Delhi : McGraw-Hill.
- [13]. Rao, Srivansa. 2018. *Protocol Signaling Procedures in LTE* di <http://www.cppu.com> (diakses pada 3 Februari 2019)
- [14]. Rogier, Boris. 2016. *Measuring Network Performance: Links Between Latency, Throughput and Packet Loss* di <https://accedian.com> (diakses pada 23 Desember 2018)
- [15]. Rossum GV. 2019. *Python Tutorial Release 3.7.2*. California USA : Python Software Foundation.
- [16]. Sreetharan, Muthuthamby. 1996. *Cellular Digital Packet Data*. Massachusetts : Artech House Publishers.