



# INSTITUT TEKNOLOGI BANDUNG

## PROGRAM STUDI TEKNIK ELEKTRO

JALAN GANESHA NO. 10 Gedung Labtek V Lantai 2 ☎ (022)2508135-36, 📠 (022)250 0940  
BANDUNG 40132

### Dokumentasi Produk Tugas Akhir

#### Lembar Sampul Dokumen

Judul Dokumen	<b>TUGAS AKHIR TEKNIK ELEKTRO:</b> <i>Fleet Monitoring and Control System pada Guided Bus</i>
Jenis Dokumen	<b>PENGUJIAN DAN VERIFIKASI</b> <small>Catatan: Dokumen ini dikendalikan penyebarannya oleh Prodi Teknik Elektro ITB</small>
Nomor Dokumen	<b>B500-02-TA1617.01.094</b>
Nomor Revisi	<b>02</b>
Nama File	<b>B500-02-TA1617.01.094</b>
Tanggal Penerbitan	<b>12 May 2017</b>
Unit Penerbit	<b>Prodi Teknik Elektro – ITB</b>
Jumlah Halaman	<b>27</b> (termasuk lembar sampul ini)

Data Pengusul				
Pengusul	Nama	Ali Zaenal Abidin	Jabatan	<b>Mahasiswa</b>
	Tanggal	4 Mei 2017	Tanda Tangan	
	Nama	Shah Dehan Lazuardi	Jabatan	<b>Mahasiswa</b>
	Tanggal	4 Mei 2017	Tanda Tangan	
	Nama	Aulia Hening Darmasti	Jabatan	<b>Mahasiswa</b>
	Tanggal	4 Mei 2017	Tanda Tangan	
Pembimbing	Nama	Arief Syaichu R.	Tanda Tangan	
	Tanggal	4 Mei 2017		
Lembaga Program Studi Teknik Elektro Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung				
Alamat Labtek V, Lantai 2, Jalan Ganesha no. 10, Bandung Telepon : +62 22 250 2260      Faks : +62 22 253 4222      Email: stei@stei.itb.ac.id				

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>2</b>
<b>CATATAN SEJARAH PERBAIKAN DOKUMEN.....</b>	<b>4</b>
<b>FLEET MONITORING AND CONTROL SYSTEM PADA GUIDED BUS.....</b>	<b>5</b>
<b>1 PENGANTAR .....</b>	<b>5</b>
1.1 RINGKASAN ISI DOKUMEN .....	5
1.2 TUJUAN PENULISAN DAN APLIKASI/KEGUNAAN DOKUMEN .....	5
1.3 REFERENSI .....	5
1.4 DAFTAR SINGKATAN.....	5
<b>2 PENGUJIAN .....</b>	<b>6</b>
2.1 PENGUJIAN DATA RATE DAN LATENSI JARINGAN .....	7
2.1.1 <i>Lingkup Pengujian.....</i>	<i>7</i>
2.1.2 <i>Konfigurasi Pengujian.....</i>	<i>7</i>
2.1.3 <i>Syarat dan Asumsi Pengujian .....</i>	<i>8</i>
2.1.4 <i>Prosedur Pengujian .....</i>	<i>8</i>
2.1.5 <i>Harapan Hasil Pengujian.....</i>	<i>8</i>
2.1.6 <i>Hasil Pengujian.....</i>	<i>9</i>
2.2 PENGUJIAN PENYIMPANGAN DATA GPS .....	11
2.2.1 <i>Lingkup Pengujian.....</i>	<i>11</i>
2.2.2 <i>Konfigurasi Pengujian.....</i>	<i>11</i>
2.2.3 <i>Syarat dan Asumsi Pengujian .....</i>	<i>11</i>
2.2.4 <i>Prosedur Pengujian .....</i>	<i>11</i>
2.2.5 <i>Harapan Hasil Pengujian.....</i>	<i>11</i>
2.2.6 <i>Hasil Pengujian.....</i>	<i>12</i>
2.3 PENGUJIAN ALGORITMA PENJADWALAN .....	13
2.3.1 <i>Lingkup Pengujian.....</i>	<i>13</i>
2.3.2 <i>Konfigurasi Pengujian.....</i>	<i>13</i>
2.3.3 <i>Syarat dan Asumsi Pengujian .....</i>	<i>13</i>
2.3.4 <i>Prosedur Pengujian .....</i>	<i>13</i>
2.3.5 <i>Harapan Hasil Pengujian.....</i>	<i>13</i>
2.3.6 <i>Hasil Pengujian.....</i>	<i>14</i>
2.4 PENGUJIAN KEMAMPUAN SERVER.....	17
2.4.1 <i>Lingkup Pengujian.....</i>	<i>17</i>
2.4.2 <i>Konfigurasi Pengujian.....</i>	<i>17</i>
2.4.3 <i>Syarat dan Asumsi Pengujian .....</i>	<i>18</i>
2.4.4 <i>Prosedur Pengujian .....</i>	<i>18</i>
2.4.5 <i>Harapan Hasil Pengujian.....</i>	<i>18</i>
2.4.6 <i>Hasil Pengujian.....</i>	<i>18</i>
2.5 PENGUJIAN PENGAMBILAN DATA CAN.....	19
2.5.1 <i>Lingkup Pengujian.....</i>	<i>19</i>
2.5.2 <i>Konfigurasi Pengujian.....</i>	<i>20</i>
2.5.3 <i>Syarat dan Asumsi Pengujian .....</i>	<i>20</i>
2.5.4 <i>Prosedur Pengujian .....</i>	<i>21</i>
2.5.5 <i>Harapan Hasil Pengujian.....</i>	<i>21</i>
2.5.6 <i>Hasil Pengujian.....</i>	<i>21</i>
2.6 PENGUJIAN ENKRIPSI-DEKRIPSI DATA.....	22

2.6.1	<i>Lingkup Pengujian</i> .....	22
2.6.2	<i>Konfigurasi Pengujian</i> .....	22
2.6.3	<i>Syarat dan Asumsi Pengujian</i> .....	22
2.6.4	<i>Prosedur Pengujian</i> .....	22
2.6.5	<i>Harapan Hasil Pengujian</i> .....	23
2.6.6	<i>Hasil Pengujian</i> .....	23
2.7	PENGUJIAN KEMAMPUAN GUI.....	23
2.7.1	<i>Lingkup Pengujian</i> .....	23
2.7.2	<i>Konfigurasi Pengujian</i> .....	23
2.7.3	<i>Syarat dan Asumsi Pengujian</i> .....	24
2.7.4	<i>Prosedur Pengujian</i> .....	24
2.7.5	<i>Harapan Hasil Pengujian</i> .....	24
2.7.6	<i>Hasil Pengujian</i> .....	24
3	<b>KESIMPULAN</b> .....	26
3.1	KETERCAPAIAN SPESIFIKASI .....	26
3.2	SARAN PENGEMBANGAN.....	27

## Catatan Sejarah Perbaikan Dokumen

VERSI, TGL, OLEH	PERBAIKAN
2, 4 Mei 2017, Ali Zaenal A. Shah Dehan L. Aulia Hening D.	Menambahkan simpulan dan saran pengembangan Menambahkan hasil pengujian CAN Bus

# ***Fleet Monitoring and Control System pada Guided Bus***

## **1 Pengantar**

### **1.1 Ringkasan Isi Dokumen**

Dokumen ini berisi penjelasan tentang metode dan hasil pengujian *fleet monitoring and control system* pada *guided bus*. Dalam dokumen ini terdapat penjelasan mengenai pengujian pemenuhan spesifikasi yang telah ditentukan sebelumnya, oleh implementasi yang sudah dilakukan.

### **1.2 Tujuan Penulisan dan Aplikasi/Kegunaan Dokumen**

Tujuan penulisan dokumen ini adalah menjelaskan tahap dan hasil pengujian pada *fleet monitoring and control system*.

### **1.3 Referensi**

### **1.4 DAFTAR SINGKATAN**

SINGKATAN	ARTI
GPS	<i>Global Positioning System</i>
FMCS	<i>Fleet Monitoring and Control System</i>
ECU	<i>Engine Control Unit</i>
GUI	<i>Graphical User Interface</i>
CAN	<i>Controller Area Network</i>
UART	<i>Universal Asynchronous Receiver Transmitter</i>
MQTT	<i>Message Queue Telemetry Transport</i>

## 2 PENGUJIAN

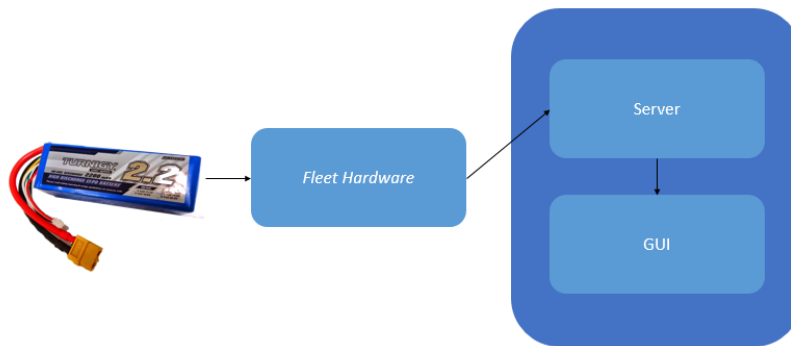
Pada bagian ini akan dijelaskan cara dan hasil pengujian yang harus dilakukan untuk verifikasi FMCS. Pengujian yang dilakukan akan menguji pemenuhan spesifikasi FMCS yang telah diuraikan pada bagian spesifikasi. Berikut tabel spesifikasi yang harus dipenuhi oleh FMCS.

**Tabel 1 Fungsi dan spesifikasi**

No.	Fungsi	Spesifikasi
1	Memantau posisi tiap armada.	Posisi yang dibaca oleh operator pada <i>control station</i> memiliki <i>error</i> maksimal sebesar 6 meter.
2	Memantau rpm kendaraan, kondisi fault, dan level energi baterai.	Dapat membaca data rpm kendaraan, kondisi fault, dan level energi baterai sesuai spesifikasi ECU yang sudah terdapat di <i>guided bus</i> .
3	Mengirimkan data ke <i>control station</i> melalui server.	Data yang dikirimkan adalah: <ul style="list-style-type: none"> <li>• posisi,</li> <li>• rpm kendaraan,</li> <li>• kondisi fault, dan</li> <li>• level energi baterai.</li> </ul> Data yang dikirim harus dapat dibaca pada GUI di <i>control station</i> maksimal setiap 0.8 detik.
4	Menampilkan data-data armada pada GUI di <i>control station</i> .	Menampilkan 10 armada dan data-data tiap armada yang akan ditampilkan di GUI pada <i>control station</i> . Data-data tersebut adalah: <ul style="list-style-type: none"> <li>• posisi tiap armada,</li> <li>• rpm tiap armada,</li> <li>• kondisi fault tiap armada, dan</li> <li>• level energi baterai tiap armada.</li> </ul> GUI ini akan dipantau dan dikendalikan oleh operator di <i>control station</i> .
5	<i>Control station</i> dapat melakukan penjadwalan.	Ada algoritma yang dapat mengatur penjadwalan operasi tiap armada.
6	Mengirimkan perintah dari <i>control station</i> untuk pengemudi melalui server.	<i>Control station</i> dapat mengirimkan perintah operasional ke armada setiap terjadi perubahan perintah.
7	Data yang dikirim dari armada ke <i>control station</i> dan dari <i>control station</i> ke armada aman.	Data-data komunikasi antara armada dan <i>control station</i> hanya dapat diakses

		oleh armada yang bersangkutan dan operator pada <i>control station</i> .
8	Menerima data dari <i>fleet hardware</i> untuk dikirim ke GUI pada <i>control station</i> .	Server dapat menerima 40 koneksi klien dengan tiap klien mengirimkan data tiap 0.8 detik.

Pada pengujian ini, secara umum digunakan arsitektur yang sama untuk setiap tahapannya. Berikut ilustrasi konfigurasi pengujian FMCS secara umum.



**Gambar 1 Arsitektur untuk pengujian FMCS**

Pengujian dilakukan secara bertahap mulai dari komunikasi *fleet hardware* dengan GUI sampai dengan pengujian di lapangan (Jalan Soekarno-Hatta) untuk integrasi keseluruhan FMCS.

## **2.1 PENGUJIAN DATA RATE DAN LATENSI JARINGAN**

### **2.1.1 Lingkup Pengujian**

Pada pengujian data rate ini, lingkup spesifikasi yang akan diuji adalah spesifikasi nomor 3 dan 6, yaitu:

- pengiriman data ke GUI pada *control station* dengan periode 0.8 detik, dan
- *control station* dapat mengirimkan perintah operasional ke armada setiap terjadi perubahan perintah.

Selain itu, dari data yang sama dapat diukur data-data statistik seperti tingkat kepercayaan terhadap data sensor yang diterima.

### **2.1.2 Konfigurasi Pengujian**

Pengujian ini dilakukan pada konfigurasi lingkungan, *software* dan *hardware* sebagai berikut.

- GUI pada laptop Intel i7-6500U 2.5 GHz, RAM 4 GB, OS Windows 10, layar 14 inch dengan resolusi 1920x1080 pixel, dan GPU NVIDIA GeForce 940M.
- Server pada laptop Intel Core 2 Duo CPU E8400 3.00 GHz, RAM 4 GB, HDD SATA 7200 RPM.
- Koneksi internet dari *router* ITB, dengan server dan GUI terhubung secara lokal.

- GUI diprogram untuk merekam interval penerimaan data dari *fleet hardware* dan disimpan dalam bentuk file teks. GUI juga diprogram untuk mengirimkan data ke *fleet hardware* untuk dilihat waktu proses penerimaan data pada *fleet hardware*.
- *Fleet hardware* menggunakan *provider* internet Indosat.
- *Fleet hardware* menggunakan baterai Li-Po 5000 mAh 20C.
- *Fleet hardware* diprogram untuk mengirimkan data setiap 500 ms, dan menerima kiriman data dari GUI.
- Lokasi pengujian di Ruang Riset Mandiri, basement Labtek VII Institut Teknologi Bandung.

### 2.1.3 Syarat dan Asumsi Pengujian

Syarat yang harus dipenuhi untuk pengujian ini yaitu koneksi internet pada *fleet hardware* pada kondisi yang baik. Hal ini dapat dicapai dengan meletakkan *fleet hardware* di tempat cukup terbuka. Syarat berikutnya yaitu GUI juga mengirimkan data ke *fleet hardware* untuk memperhitungkan waktu proses penerimaan data oleh *fleet hardware*.

Asumsi yang digunakan pada pengujian ini adalah sumber daya dari baterai tidak mempengaruhi performa *fleet hardware* yang seharusnya mengambil sumber daya dari *guided bus*.

### 2.1.4 Prosedur Pengujian

- Mengaktifkan server dan memulai GUI.
- *Fleet hardware* diprogram untuk mengirimkan data setiap 500 ms, kemudian dinyalakan untuk memulai pengiriman data ke server.
- Sistem dibiarkan menyala selama 135 menit.
- Memindahkan isi file teks yang dibuat oleh GUI ke file excel.
- Data dalam file excel diolah untuk memperoleh grafik dan data-data statistik yang diperlukan.

### 2.1.5 Harapan Hasil Pengujian

Pada pengujian ini akan diperoleh data interval penerimaan data pada GUI. Dari data-data tersebut dapat dicari parameter-parameter:

- persentase interval penerimaan data yang melebihi 700 ms,
- standar deviasi interval penerimaan data,
- grafik interval penerimaan data, dan
- interval penerimaan data yang melebihi 500 ms.

Dari persentase interval data yang melebihi 700 ms dan standar deviasi, diharapkan dapat diperoleh *provider* internet yang memiliki kecepatan pengiriman yang paling tinggi dengan standar deviasi waktu pengiriman yang paling rendah (tidak terlalu beragam) untuk hasil yang stabil. Dari interval penerimaan data yang melebihi 500 ms, dapat dilakukan perhitungan perkiraan rata-rata latensi jaringan dengan cara mengurangkan rata-rata data tersebut dengan 500 ms.



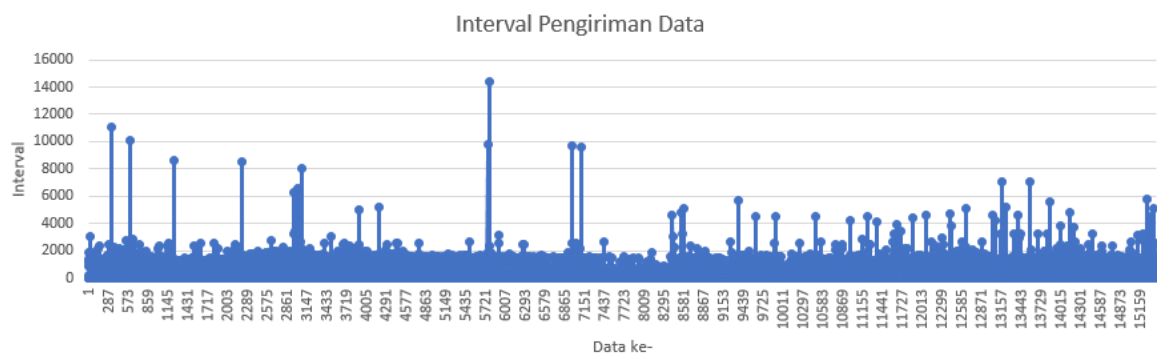
### 2.1.6 Hasil Pengujian

Dari hasil pengumpulan data interval pengiriman data dari *fleet hardware* ke GUI, dapat diperoleh data-data statistik sebagai berikut.

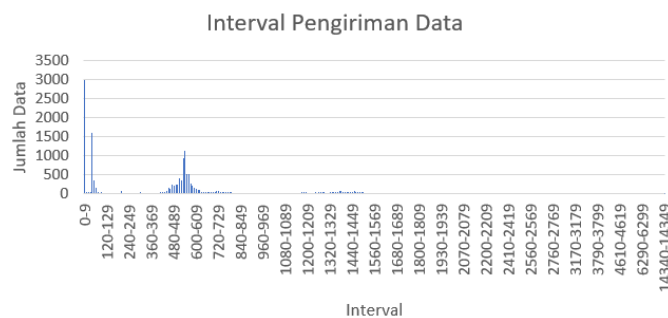
**Tabel 2 Interval pengiriman data**

Jaringan	Lama Pengamatan	Jumlah Data	Jumlah Data Interval Melebihi 700 ms	Rata-Rata	Standar Deviasi
IM3	150 menit	15394	80.2 %	508.618 ms	576.346 ms

Pengamatan dilakukan pada interval pengiriman data selama 150 menit dengan jumlah data 15394 data. Dari tabel di atas dapat dilihat bahwa terdapat 80.2% data interval yang melebihi 700 ms, dengan rata-rata 508.618 ms dan standar deviasi 576.346 ms. Data statistik ini masih belum cukup merepresentasikan keadaan *fleet hardware*, dikarenakan masih ada faktor turbulensi jaringan di tempat pengamatan.

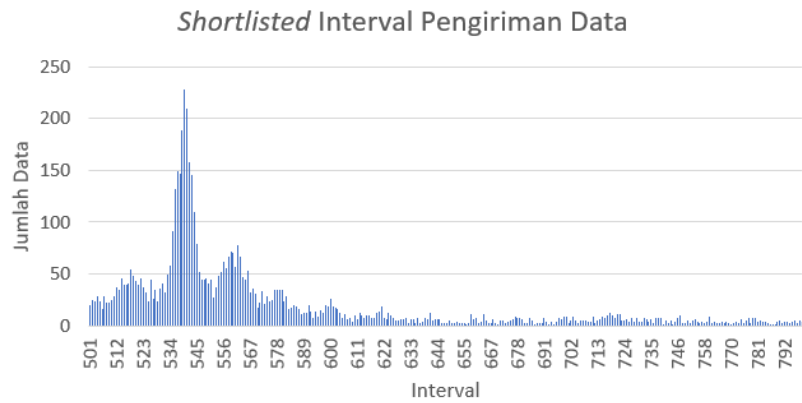


**Gambar 2 Interval pengiriman data berdasarkan waktu penerimaan**



**Gambar 3 Interval pengiriman data berdasarkan pengelompokan interval**

Untuk melihat performa pengiriman data oleh *fleet hardware* tanpa pengaruh turbulensi jaringan, dilakukan pemilahan data. Karena *fleet hardware* diprogram untuk mengirimkan data dengan interval 500 ms dan dengan asumsi latensi jaringan tidak melebihi 300 ms, data interval dipilah untuk hanya menampilkan data dengan interval di atas 500 ms, dan di bawah 800 ms. Berikut grafik data yang dihasilkan.



**Gambar 4 Interval pengiriman data setelah dipilah**

Berikut data statistik yang diperoleh dari data tersebut.

**Tabel 3 Interval pengiriman data setelah dipilah**

Jaringan	Lama Pengamatan	Jumlah Data	Jumlah Data Interval Melebihi 700 ms	Rata-Rata	Standar Deviasi
IM3	150 menit	5728	91.72 %	571.42 ms	63.313 ms

Dari tabel di atas dapat dilihat rata-rata pengiriman data adalah 571.42 ms dengan standar deviasi 63.313 ms. Karena *fleet hardware* mengirim data setiap 500 ms, dapat dianggap bahwa perbedaan waktu ini disebabkan oleh faktor latensi jaringan. Oleh karena itu, rata-rata latensi pengiriman data dari *fleet hardware* dapat dihitung:

$$571.42 \text{ ms} - 500 \text{ ms} = 71.42 \text{ ms}$$

Dengan 571.42 ms adalah rata-rata interval pengiriman data dan 500 ms adalah periode pengiriman data dari *fleet hardware*.

Kemudian, dilihat juga data yang dikirimkan dari GUI ke *fleet hardware*. Berikut gambar tampilan *fleet hardware*.



**Gambar 5 Tampilan LCD fleet hardware**

Processing time rata-rata adalah 250 ms. waktu ini terkadang menjadi 350 ms ketika ada data yang dikirim dari GUI.

## 2.2 PENGUJIAN PENYIMPANGAN DATA GPS

### 2.2.1 Lingkup Pengujian

Pada pengujian penyimpangan data GPS, lingkup spesifikasi yang akan diuji adalah spesifikasi nomor 1, yaitu posisi yang dibaca oleh operator pada *control station* memiliki *error* maksimal sebesar 6 meter.

### 2.2.2 Konfigurasi Pengujian

Pengujian ini dilakukan pada konfigurasi *software* dan *hardware* sebagai berikut.

- Aplikasi U-Center pada laptop Intel i5-6200U 2.8 GHz, memori RAM 4 GB dengan sistem operasi Windows 10.
- Modul GPS Neo-M8N yang terhubung dengan Arduino Uno, baterai Li-Po 5000 mAh 20C.
- Arduino Uno terhubung dengan laptop menggunakan kabel USB-B.
- Pengujian ini dilakukan di Labtek VIII lantai 3, Institut Teknologi Bandung dengan antenna diletakkan di luar jendela.

### 2.2.3 Syarat dan Asumsi Pengujian

Syarat yang harus dipenuhi untuk pengujian ini yaitu antenna GPS berada pada tempat yang cukup terbuka sehingga dapat mendeteksi minimal 8 satelit agar data posisinya akurat.

Asumsi yang digunakan pada pengujian ini adalah lama waktu pengujian ini dapat merepresentasikan penggunaan modul GPS pada implementasi.

### 2.2.4 Prosedur Pengujian

- Menghubungkan modul GPS Neo-M8N dengan UART USB (melalui Arduino Uno) pada laptop.
- Membuka aplikasi U-Center.
- Melakukan pengaturan awal GPS, yaitu mengganti *baudrate* menjadi 9600 dan *refresh rate* dengan periode 200 ms.
- Mengubah pengaturan U-Center untuk membaca data dari GPS dengan *baudrate* 9600.
- Melihat kesesuaian posisi awal yang terbaca oleh GPS dengan *Google Maps*.
- Membuka menu *deviation* pada U-Center, kemudian dibiarkan selama 50 menit untuk direkam simpangan pembacaan data koordinatnya pada sebuah peta berskala meter.

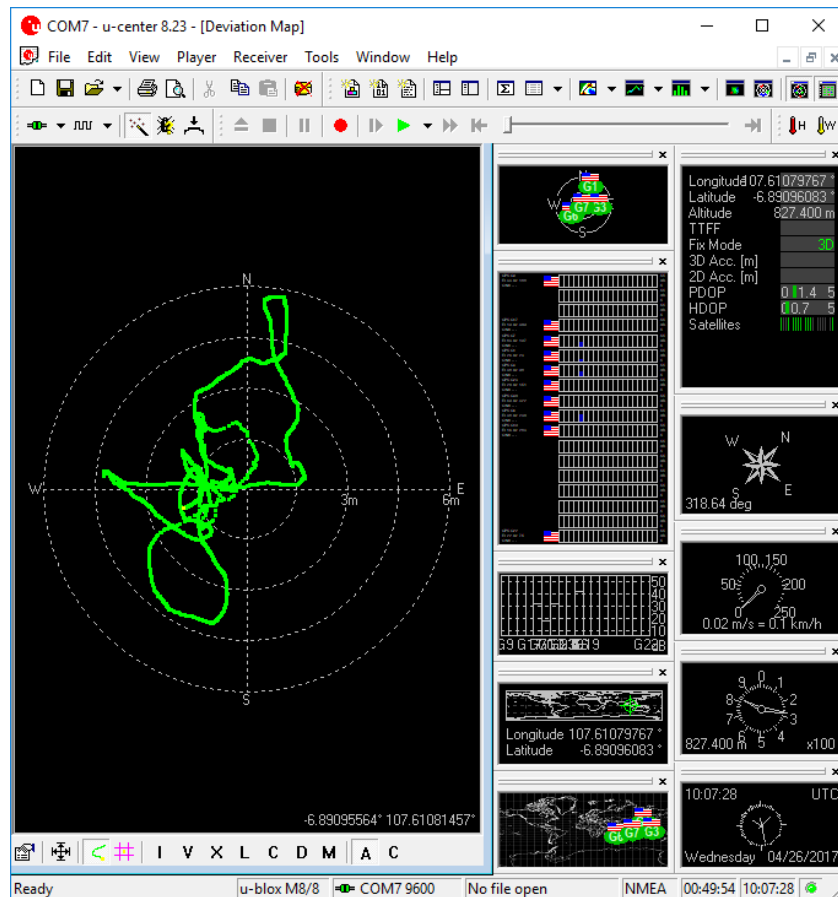
### 2.2.5 Harapan Hasil Pengujian

Pada pengujian ini akan diperoleh data posisi yang dibaca oleh modul GPS. Data posisi ini diambil dengan interval tiap 200 ms, kemudian diplot pada sebuah *deviation map*. Setelah pengujian ini, akan diperoleh peta penyimpangan data GPS dalam satuan meter. Diharapkan *error* data GPS yang diperoleh dari sensor tidak melebihi *error* yang telah ditentukan pada spesifikasi, yaitu 6 meter.

### 2.2.6 Hasil Pengujian

Pada keadaan awal, diperoleh koordinat awal yang dibaca oleh GPS yaitu *latitude* -6,890955 dan *longitude* 107,610754. Nilai koordinat ini sudah sesuai dengan lokasi pengujian penyimpanan data GPS.

Berikutnya diperoleh penyimpangan data GPS dalam satuan meter, seperti pada gambar berikut. Data ini diperoleh pada keadaan GPS terkunci pada 10 satelit dengan mode 3D.



Gambar 6 Penyimpangan data GPS

Dari gambar di atas dapat dilihat bahwa penyimpangan data GPS masih berubah-ubah, tetapi masih dalam batas spesifikasi yaitu penyimpangan maksimal sebesar 6 meter. Dengan pengamatan visual pada bentuk deviasi di atas, dapat disimpulkan penyimpangan data GPS ini berkisar antara 1 sampai dengan 3 meter. Penyimpangan ini cukup sesuai dengan spesifikasi modul GPS Neo-M8N yaitu 2.5 meter.

Namun, terdapat juga data GPS dengan penyimpangan hingga hampir 6 meter. Pada implementasi *guided bus* dengan pengemudi, penyimpangan secara tiba-tiba ini tidak terlalu berpengaruh pada keseluruhan sistem. Lain halnya untuk implementasi ke depannya, yaitu *autodrive guided bus*. Penyimpangan ini mungkin akan menimbulkan masalah pada keseluruhan sistem.

## 2.3 PENGUJIAN ALGORITMA PENJADWALAN

### 2.3.1 Lingkup Pengujian

Pada pengujian algoritma penjadwalan, lingkup spesifikasi yang akan diuji adalah spesifikasi nomor 5, yaitu ada algoritma yang dapat mengatur penjadwalan operasi tiap armada.

### 2.3.2 Konfigurasi Pengujian

Pengujian ini dilakukan pada konfigurasi *software* dan *hardware* sebagai berikut.

- GUI dan *dummy data* pada laptop Intel i7-6500U 2.5 GHz, RAM 4 GB, OS Windows 10, layar 14 inch dengan resolusi 1920x1080 pixel, dan GPU NVIDIA GeForce 940M.
- Lokasi pengujian di Ruang Riset Mandiri, basement Labtek VII ITB Ganesha.
- GUI berfungsi untuk menampilkan posisi penjadwalan dan posisi fleet.
- Dummy data adalah program untuk mensimulasikan fleet.

### 2.3.3 Syarat dan Asumsi Pengujian

Asumsi yang digunakan pada pengujian ini adalah pengemudi *guided bus* selalu mengikuti perintah dari *control station* sehingga program *dummy data* merepresentasikan keadaan armada dengan tepat.

### 2.3.4 Prosedur Pengujian

- Menjalankan server dan hubungkan pada jaringan internet.
- Menjalankan GUI dan hubungkan pada server melalui jaringan local.
- Atur posisi *fleet* agar lebih cepat dari penjadwalan pada aplikasi *Dummy Data*.
- Menjalankan aplikasi *Dummy Data* dan hubungkan pada server melalui internet.
- Merekam hasil *screenshot* dan perbedaan waktu yang dibutuhkan fleet untuk sesuai dengan penjadwalan.
- Atur posisi *fleet* agar lebih lambat dari penjadwalan pada aplikasi *Dummy Data*.
- Menjalankan aplikasi *Dummy Data* dan hubungkan pada server melalui internet.
- Merekam hasil *screenshot* dan perbedaan waktu yang dibutuhkan fleet untuk sesuai dengan penjadwalan.
- Mengatur satu *fleet* menjadi non-aktif pada aplikasi *Dummy Data*.
- Menjalankan aplikasi *Dummy Data* dan hubungkan pada server melalui internet.
- Merekam hasil *screenshot* dan perbedaan waktu yang dibutuhkan fleet untuk sesuai dengan penjadwalan.

### 2.3.5 Harapan Hasil Pengujian

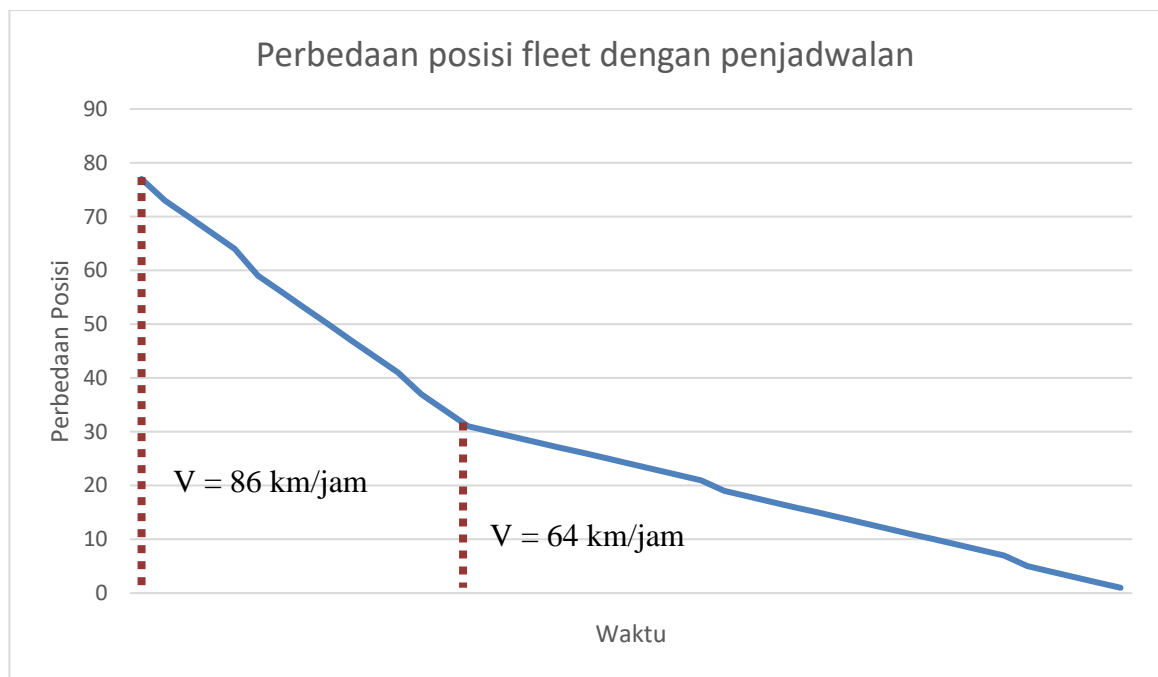
Pada pengujian ini akan diperoleh perkiraan waktu yang dibutuhkan fleet untuk mencapai penjadwalan sesuai dengan algoritma yang telah diimplementasikan. Diharapkan fleet dapat mencapai penjadwalan setelah berada dalam keadaan yang tidak sesuai dengan seharusnya, yaitu saat fleet lebih cepat dan lebih lambat dari penjadwalan yang seharusnya.

### 2.3.6 Hasil Pengujian

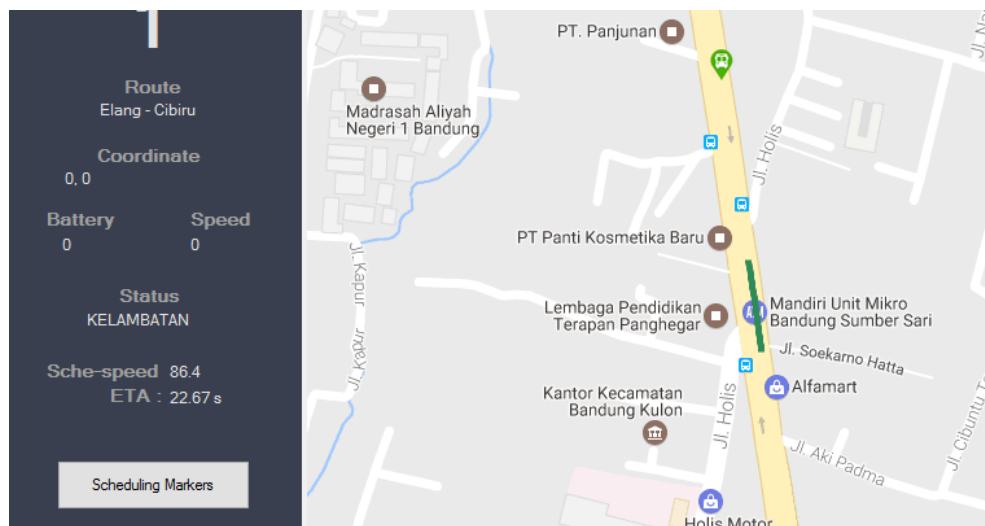
Hasil pengujian ini dibagi menjadi tiga, yaitu hasil pengujian untuk *fleet* lebih cepat dari penjadwalan, *fleet* lebih lambat dari penjadwalan dan ada satu *fleet* yang non-aktif di tengah perjalanan.

#### 2.3.6.1 Fleet Lebih Lambat dari Penjadwalan

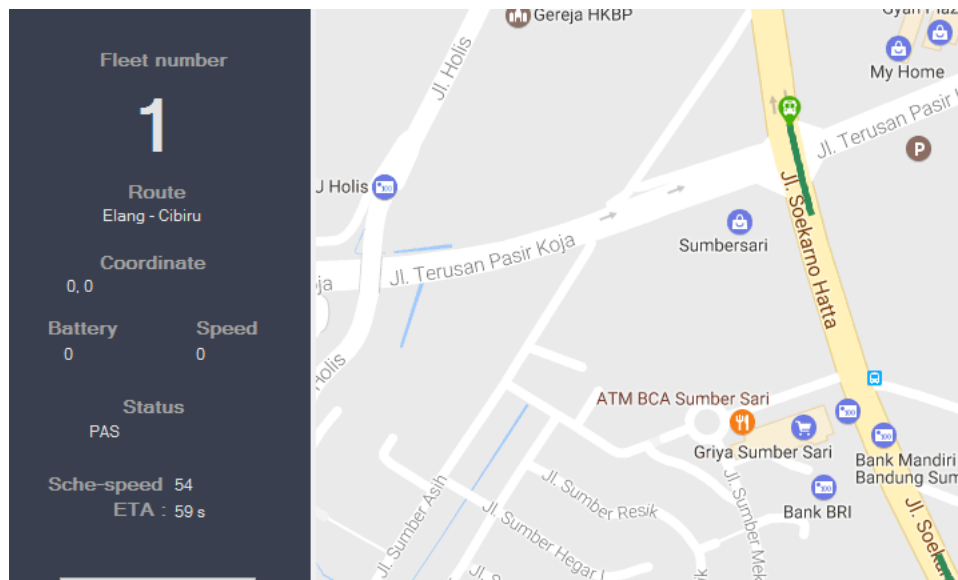
Untuk pengujian ini, *fleet* diatur mulai dari segmen ke-135 pada Jalan Soekarno-Hatta dengan posisi penjadwalan di segmen ke-230 pada Jalan Soekarno-Hatta. *Fleet* telah berhasil mencapai penjadwalannya dengan mengatur kecepatan untuk menyusul ketertinggalan. Berikut grafik yang menunjukkan perbedaan posisi *fleet* dengan posisi penjadwalannya terhadap waktu.



Gambar 7 Perbedaan posisi fleet dengan penjadwalan



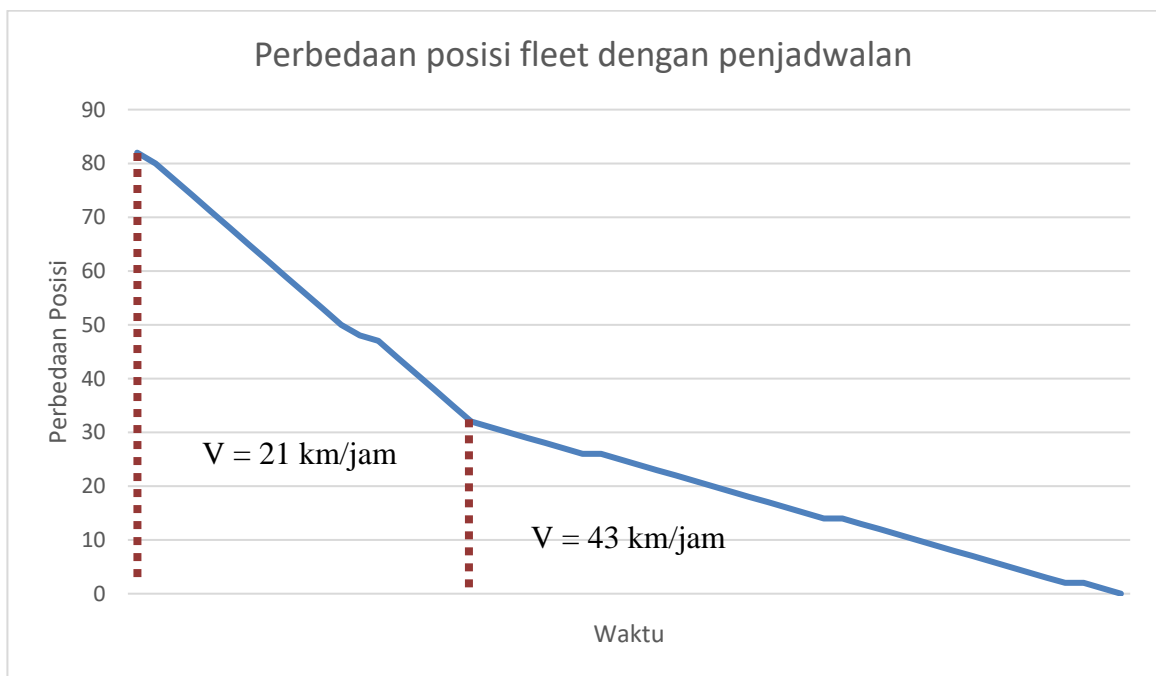
Gambar 8 Tampilan GUI saat fleet (pin hijau muda) tertinggal dari penjadwalan (garis hijau)



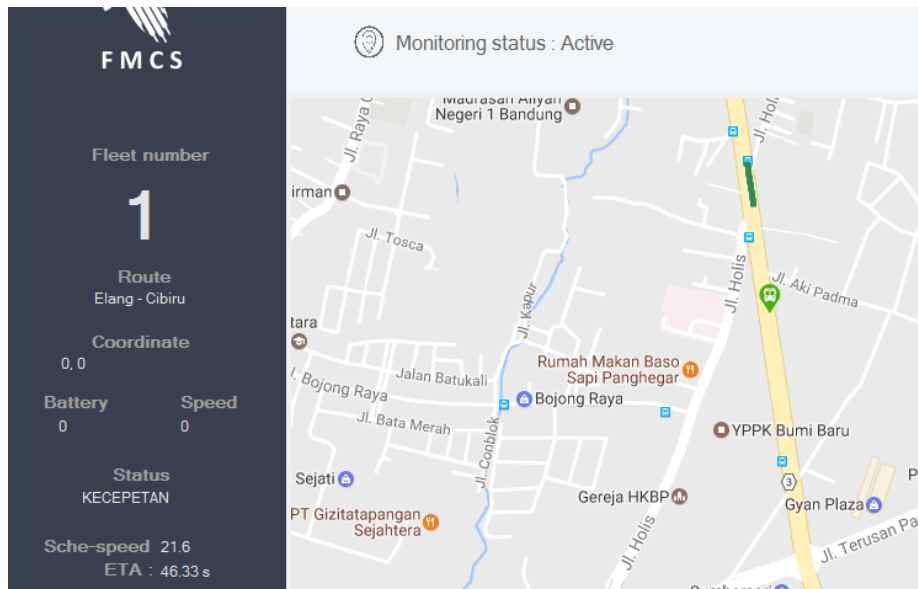
Gambar 9 Tampilan GUI saat fleet (pin hijau muda) tepat dengan penjadwalan (garis hijau)

### 2.3.6.2 Fleet Lebih Cepat dari Penjadwalan

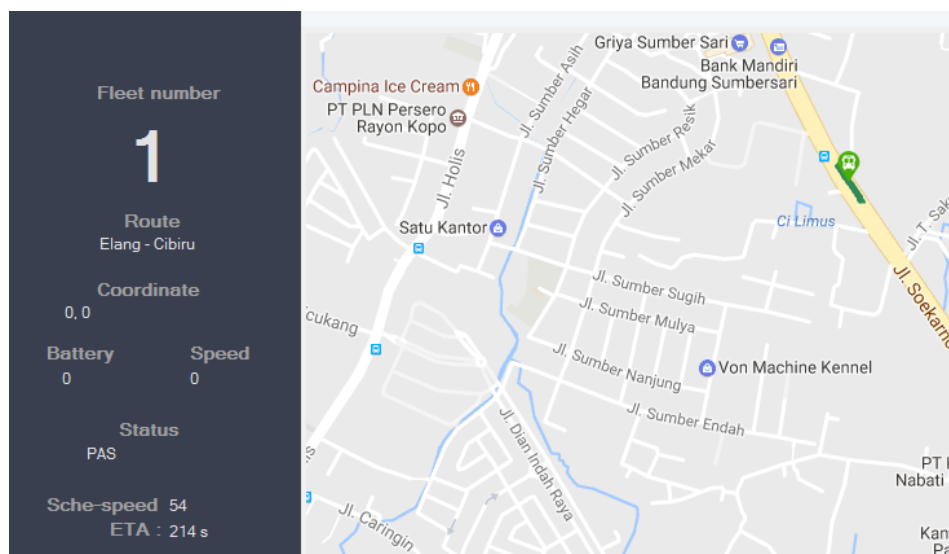
Untuk pengujian ini, *fleet* diatur mulai dari segmen ke-325 pada Jalan Soekarno-Hatta dengan posisi penjadwalan di segmen ke-230 pada Jalan Soekarno-Hatta. *Fleet* telah berhasil mencapai penjadwalannya dengan mengatur kecepatan untuk kembali ke penjadwalannya. Berikut grafik yang menunjukkan perbedaan posisi *fleet* dengan posisi penjadwalannya terhadap waktu.



Gambar 10 Perbedaan posisi fleet dengan penjadwalan



**Gambar 11 Tampilan GUI saat fleet (pin hijau muda) mendahului dari penjadwalan (garis hijau)**

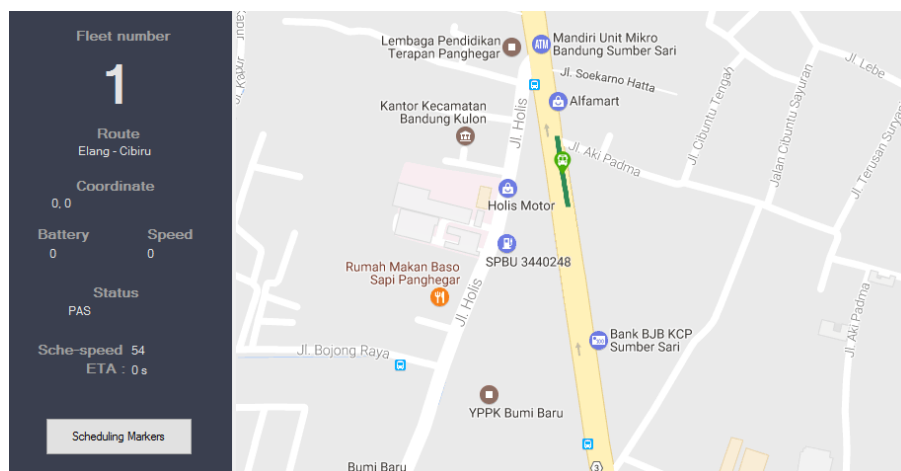
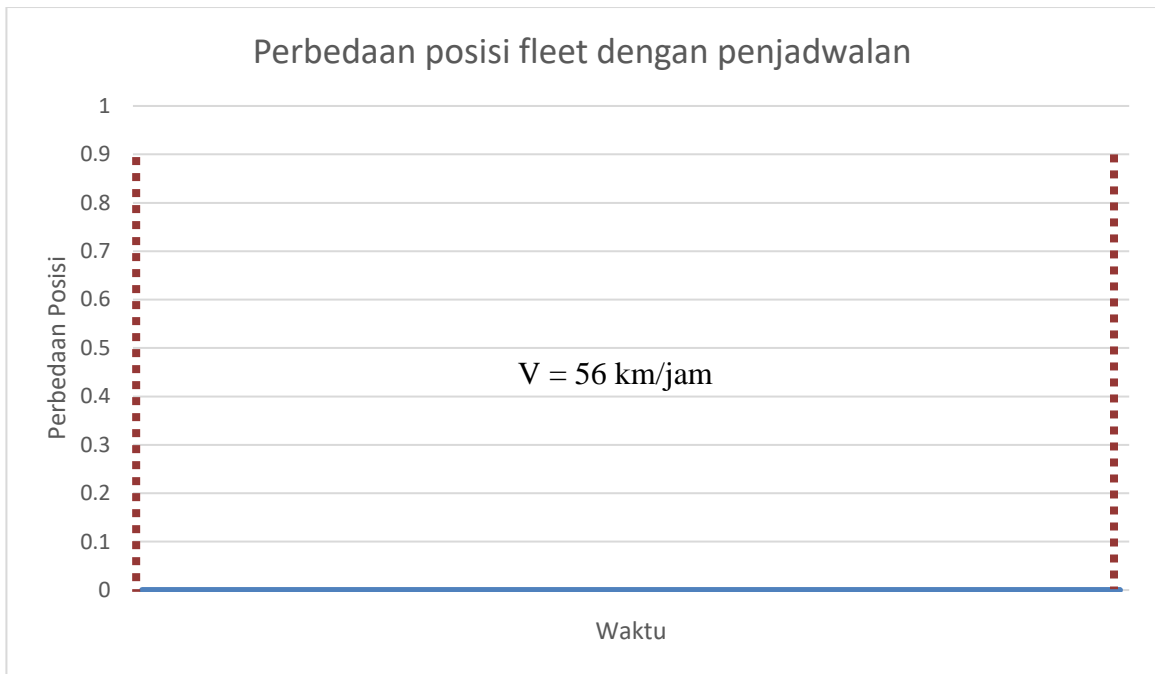


**Gambar 12 Tampilan GUI saat fleet (pin hijau muda) tepat dengan penjadwalan (garis hijau)**

### 2.3.6.3 Fleet Sesuai Dengan Penjadwalan

Untuk pengujian ini, *fleet* diatur mulai dari segmen ke-230 pada Jalan Soekarno-Hatta dengan posisi penjadwalan di segmen ke-230 pada Jalan Soekarno-Hatta. *Fleet* telah berhasil mencapai penjadwalannya dengan mengatur kecepatan untuk kembali ke penjadwalannya. Berikut grafik yang menunjukkan perbedaan posisi *fleet* dengan posisi penjadwalannya terhadap waktu.





## 2.4 PENGUJIAN KEMAMPUAN SERVER

### 2.4.1 Lingkup Pengujian

Pada pengujian kemampuan server, lingkup spesifikasi yang akan diuji adalah spesifikasi nomor 8, yaitu server dapat menerima 40 koneksi klien dengan tiap klien mengirimkan data sebanyak 32 byte tiap 0.8 detik dan server dapat mempublish pesan kepada subscriber dengan *realtime*.

### 2.4.2 Konfigurasi Pengujian

Pengujian ini dilakukan pada konfigurasi *software* dan *hardware* sebagai berikut.

- Komputer server dengan spesifikasi Intel i5 4000M 2.4 GHz, RAM 8 GB, Seagate 2.5" SSHD 1TB, SSD 8GB.
- Server MQTT mosquitto-1.14.11 pada port 22 komputer server.

- Komputer *client dummy data* dengan spesifikasi Intel i5-6200U 2.8 GHz dan memori RAM 4 GB dengan sistem operasi Windows 10. Laptop terhubung dengan jaringan local dengan server.
- Aplikasi *dummy data* terhubung dengan server melalui port 22
- Aplikasi *dummy data* membuat 40 koneksi pada server sebagai client yang berbeda-beda.
- Aplikasi *dummy data* mem-*publish* pesan sebesar 32 byte setiap 0.8 detik ke 40 topik yang berbeda.
- Aplikasi *dummy data* men-*subscribe* pada 40 topik tersebut.
- Aplikasi *dummy data* menghitung semua interval data pada setiap topik.
- Router yang digunakan adalah TP-Link TL-WR841ND v7 dengan spesifikasi sebagai berikut CPU freq = 400 Mhz, 300Mbps wireless connection speed.

### 2.4.3 Syarat dan Asumsi Pengujian

Asumsi yang digunakan dalam pengujian server adalah *latency* yang ditimbulkan murni dari server.

### 2.4.4 Prosedur Pengujian

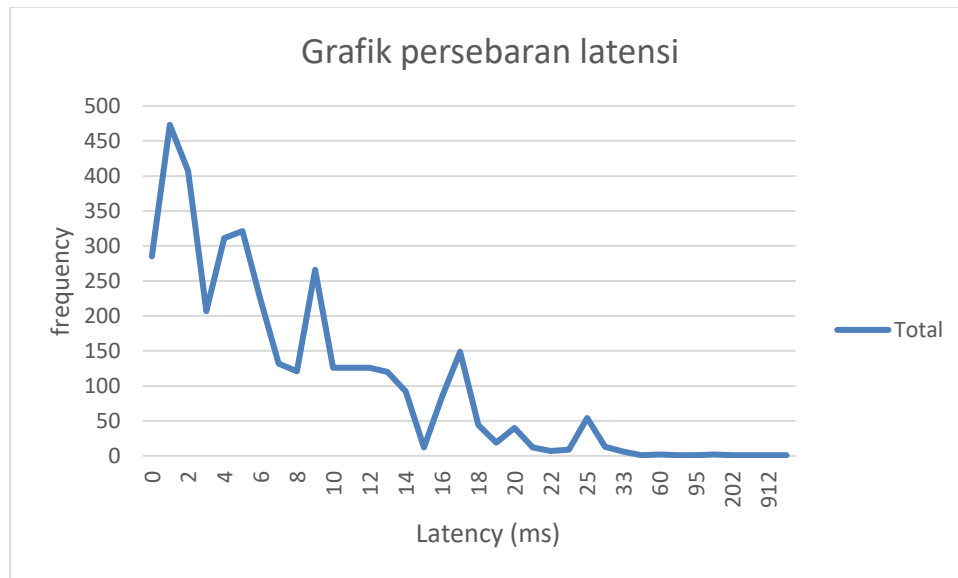
- Menjalankan server dan hubungkan dengan jaringan lokal.
- Menjalankan aplikasi dummy data dan hubungkan aplikasi dengan server pada jaringan lokal
- Sistem dibiarkan menyala selama 95 detik dan diambil datanya.

### 2.4.5 Harapan Hasil Pengujian

Dari pengujian ini akan diperoleh data latensi yang diterima oleh server. Diharapkan server mampu menerima 40 koneksi dari client serta menerima pesan dari tiap client dan mempublish pesan yang diterima sesuai dengan topiknya secara realtime.

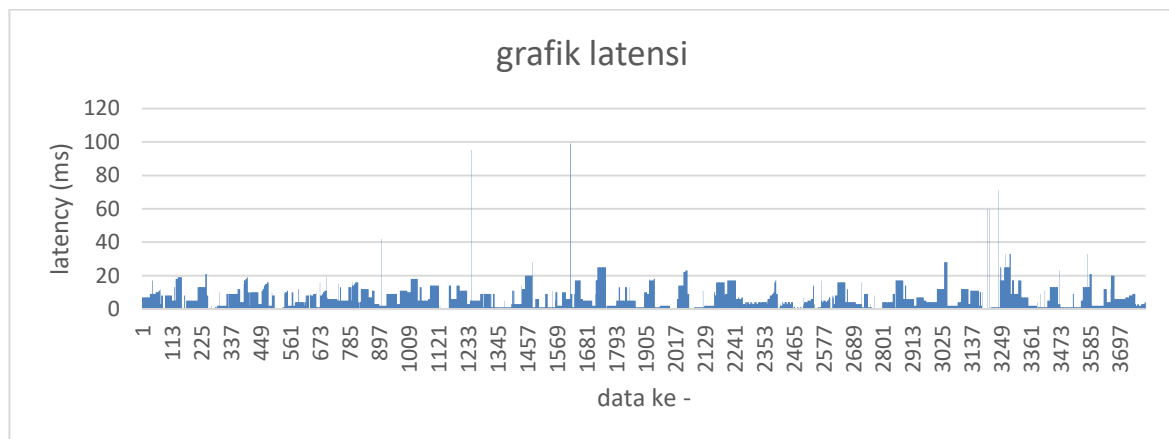
### 2.4.6 Hasil Pengujian

Pada pengujian ini dilakukan selama 95 detik didapatkan rata-rata latensi adalah 7.58 ms dengan persebaran data seperti grafik dibawah.



**Gambar 13 Grafik persebaran latensi**

Terdapat 3793 data yang didapat dari hasil pengujian server seperti pada grafik dibawah.



**tabel 4 grafik persebaran latensi**

Hasil pengujian menunjukan server dapat melayani client secara realtime.

## 2.5 PENGUJIAN PENGAMBILAN DATA CAN

### 2.5.1 Lingkup Pengujian

Pada pengujian data rate ini, lingkup spesifikasi yang akan diuji adalah spesifikasi nomor 2 dan 4, yaitu:

- dapat membaca data rpm kendaraan, kondisi fault, dan level energi baterai sesuai spesifikasi ECU yang sudah terdapat di guided bus, dan
- menampilkan 10 armada dan data-data tiap armada yang akan ditampilkan di GUI pada *control station*. Data-data tersebut adalah posisi tiap armada, rpm tiap armada, kondisi fault tiap armada, dan level energi baterai tiap armada. GUI ini akan dipantau dan dikendalikan oleh operator di *control station*.

### 2.5.2 Konfigurasi Pengujian

Pengujian ini dilakukan pada konfigurasi lingkungan, *software* dan *hardware* sebagai berikut.

- GUI pada laptop Intel i7-6500U 2.5 GHz, RAM 4 GB, OS Windows 10, layar 14 inch dengan resolusi 1920x1080 pixel, dan GPU NVIDIA GeForce 940M.
- Server pada laptop Intel Core 2 Duo CPU E8400 3.00 GHz, RAM 4 GB, HDD SATA 7200 RPM.
- Koneksi internet dari *tethering*, dengan server dan GUI terhubung secara lokal.
- *Fleet hardware* menggunakan baterai Li-Po 5000 mAh 20C.
- *Fleet hardware* menggunakan *provider* internet Indosat.
- Lokasi pengujian di PT. LEN Industri.

### 2.5.3 Syarat dan Asumsi Pengujian

Syarat yang harus dipenuhi untuk pengujian ini yaitu koneksi internet pada *fleet hardware* pada kondisi yang baik. Hal ini dapat dicapai dengan meletakkan *fleet hardware* di tempat cukup terbuka. Syarat lain yaitu CAN Bus yang ada sudah menyediakan data-data yang dibutuhkan.

Asumsi yang digunakan pada pengujian ini adalah sumber daya dari baterai tidak mempengaruhi performa *fleet hardware* yang seharusnya mengambil sumber daya dari *guided bus*. Asumsi berikutnya yaitu alamat dan isi data CAN Bus pada Mmini AGT akan sama dengan alamat dan isi dan CAN Bus yang akan digunakan pada *guided bus*.

ID	Equipment	Remarks							
		Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
1	INVERTER	ID 1	102 (FAILUREIW) 103 (FAILUREVDC) 104 (FAILUREZERODEG)	101 (FAILUREIU) 102 (FAILUREIW) 103 (FAILUREVDC) 104 (FAILUREZERODEG)	0	0	0	0	0
	BMS	ID 2	FAULT_REG	0	0	0	0	0	0
	DC-DC	ID 3	1: Input UV LIRE 2: Input UV FON 3: Input OV LIRE 4: Input OV NAON 5: Output UV LIRE 6: Output UV NAON 7: Output OV LIRE 8: Output OV NAON 9: Output OC LIRE 10: Output OC NAON 11: Powermodule NOK LIRE 12: Powermodule NOK NAON 13: Commu Lost LINDRE	0	0	0	0	0	0
	BCU	ID 4							

2	ECU to Inv	Mode 0: SensorInit 2: ZeroDeg 1: Neutral 4: Forward 8: Reverse	Throttle 0 - 100	Brake 0 - 100	0	0	0	0	0	
3	Inv to ECU	Mode Status 1: ZeroDeg 2: Neutral 3: Forward 4: Reverse	Applied Throttle 0 - 100	Applied Brake 0 - 100	RPM Byte0 (signed int) (Data)	RPM Byte1 (signed int) (Data)	Motor Temp (Data)	IGBT Temp (Data)	0	
4	ECU to BMS	Mode 1: Close Contactor 0: Open Contactor	Mode 1: Charge 0: Idle	0	0	0	0	0	0	
5	BMS to ECU	1	V_BAT Byte0	V_BAT Byte 1	LBAT Byte 0	LBAT Byte 1	V_HVDC Byte 0	V_HVDC Byte 1	0	
16	ECU to DC-DC 1	2 Status	SOC 0	g 0	BMS_STATE 0	0	0	0	0	
26	ECU to DC-DC 2	1: Comm OK Status	0	0	0	0	0	0	0	
7	DC-DC 1 to ECU	1: OK 2: Auto Restart Status	1	0	0	0	0	0	0	
7	DC-DC 2 to ECU	1: OK 2: Auto Restart Status	2	0	0	0	0	0	0	
8	ECU to BCU									
9	BCU to ECU									

Gambar 14 Isi data CAN Mini AGT LEN

## 2.5.4 Prosedur Pengujian

- *Fleet hardware* dihubungkan dengan CAN Bus pada Mini AGT milik PT. LEN Industri.
- Mengaktifkan server dan memulai GUI.
- Menyalakan *fleet hardware* untuk mulai mengirimkan data ke server.

## 2.5.5 Harapan Hasil Pengujian

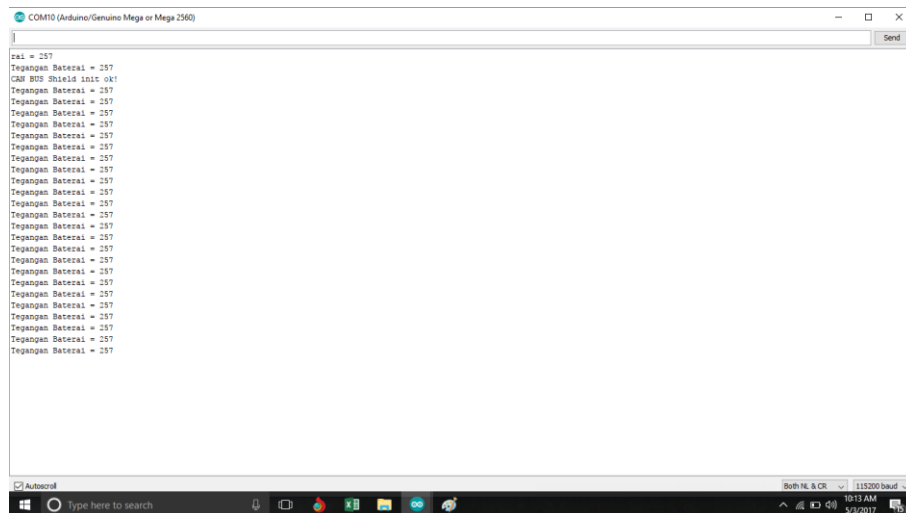
Pada pengujian ini akan diperoleh data dari CAN Bus pada Mini AGT milik PT. LEN Industri. Data yang diperoleh hanya data yang diperlukan, yaitu tegangan baterai, rpm dan kondisi *fault*.

Pada pengujian ini diharapkan data CAN Bus pada Mini AGT milik PT. LEN Industri dapat dibaca dan ditampilkan pada GUI.

## 2.5.6 Hasil Pengujian

*Fleet hardware* telah berhasil membaca data-data dari CAN Bus pada Mini AGT. Data-data yang dikirim oleh CAN Bus ini memiliki ID berbeda-beda tergantung mesin yang sedang mengirimkan data yang bersangkutan.

Pada Mini AGT, data yang secara periodik dikirimkan adalah data dengan ID 4 dan 5. Untuk implementasi *fleet hardware* ini, data yang diperlukan berada pada ID 4, 5 dan 1. Data dengan ID 1 hanya dikirimkan saat terjadi *fault* pada Mini AGT.



Gambar 15 Hasil pembacaan tegangan baterai dari ECU baterai

## 2.6 PENGUJIAN ENKRIPSI-DEKRIPSI DATA

### 2.6.1 Lingkup Pengujian

Pada pengujian algoritma penjadwalan, lingkup spesifikasi yang akan diuji adalah spesifikasi nomor 7, yaitu data-data komunikasi antara armada dan *control station* hanya dapat diakses oleh armada yang bersangkutan dan operator pada *control station*.

### 2.6.2 Konfigurasi Pengujian

Pengujian ini dilakukan pada konfigurasi *software* dan *hardware* sebagai berikut.

- GUI pada laptop Intel i7-6500U 2.5 GHz, RAM 4 GB, OS Windows 10, layar 14 inch dengan resolusi 1920x1080 pixel, dan GPU NVIDIA GeForce 940M.
- Server pada laptop Intel Core 2 Duo CPU E8400 3.00 GHz, RAM 4 GB, HDD SATA 7200 RPM.
- Koneksi internet dari *router* ITB, dengan server dan GUI terhubung secara lokal.
- *Fleet hardware* menggunakan *provider* internet Indosat, menggunakan *power supply* 9 volt eksternal dan diprogram untuk mengirimkan data setiap 500 ms, dan menerima kiriman data dari GUI.
- Laptop yang digunakan untuk menjalankan GUI juga menjalankan aplikasi MQTT *client* untuk membandingkan hasil yang diterima oleh GUI (terdekripsi) dan yang diterima oleh *client* tersebut (belum terdekripsi).
- Lokasi pengujian di Ruang Riset Mandiri, basement Labtek VII Institut Teknologi Bandung.

### 2.6.3 Syarat dan Asumsi Pengujian

Syarat yang harus terpenuhi dalam pengujian ini adalah koneksi MQTT *client* yang digunakan stabil dan dapat menerima data berupa karakter-karakter yang tidak umum.

### 2.6.4 Prosedur Pengujian

- Menjalankan server dan hubungkan pada jaringan internet.
- Menjalankan GUI dan hubungkan pada server melalui jaringan lokal.

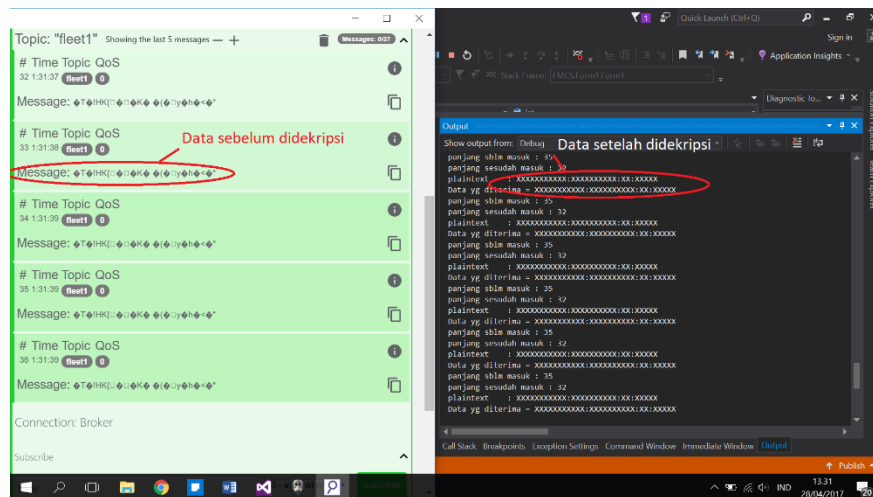
- Menyalakan *fleet hardware*.
- Menjalankan aplikasi MQTT *client* lain.
- Mengamati hasil data yang diterima oleh GUI dan *client*.

### 2.6.5 Harapan Hasil Pengujian

Pada pengujian ini akan diperoleh pengamatan data yang diterima oleh GUI dan yang diterima oleh MQTT *client*. Kedua data ini kemudian dibandingkan untuk melihat perbedaan data setelah dan sebelum didekripsi.

### 2.6.6 Hasil Pengujian

Pada pengujian ini diperoleh *screenshot* hasil pengamatan data sebagai berikut.



**Gambar 16 Perbedaan data terdekripsi dan tidak terdekripsi**

Dari gambar di atas dapat dilihat bahwa data terdekripsi dan belum terdekripsi berbeda. Sebelum didekripsi, data yang terbaca berupa karakter-karakter yang tidak bisa langsung dimengerti oleh manusia. Hal ini membuktikan bahwa data yang dikirimkan sudah terenkripsi dengan baik.

## 2.7 PENGUJIAN KEMAMPUAN GUI

### 2.7.1 Lingkup Pengujian

Pada pengujian ini, lingkup spesifikasi yang akan diuji adalah spesifikasi nomor 4, yaitu menampilkan 10 armada dan data-data tiap armada yang akan ditampilkan di GUI pada *control station*.

Akan ditampilkan posisi dari 40 armada pada peta geografis dan 10 data lengkap armada pada tabel monitoring. Data-data tersebut adalah koordinat tiap armada, rpm tiap armada, kondisi fault tiap armada, dan level energi baterai tiap armada. GUI ini akan dipantau dan dikendalikan oleh operator di *control station*.

### 2.7.2 Konfigurasi Pengujian

Pengujian ini dilakukan pada konfigurasi *software* dan *hardware* sebagai berikut.

- GUI pada laptop Intel i7-6500U 2.5 GHz, RAM 4 GB, OS Windows 10, layar 14 inch dengan resolusi 1920x1080 pixel, dan GPU NVIDIA GeForce 940M.

- Server pada laptop Intel Core 2 Duo CPU E8400 3.00 GHz, RAM 4 GB, HDD SATA 7200 RPM.
- Koneksi internet dari *router* ITB, dengan server dan GUI terhubung secara lokal.
- Aplikasi *dummy data* pada laptop Intel i5-6200U 2.8 GHz dan memori RAM 4 GB dengan sistem operasi Windows 10. Laptop terhubung ke internet.
- *Fleet hardware* menggunakan baterai Li-Po 5000 mAh 20C.
- *Fleet hardware* menggunakan *provider* internet XL.

### 2.7.3 Syarat dan Asumsi Pengujian

Syarat yang harus dipenuhi untuk pengujian ini yaitu koneksi internet yang cukup baik pada *fleet hardware*, server, dan PC yang digunakan untuk menampilkan GUI. Selain itu pada PC yang digunakan untuk menampilkan GUI tidak dijalankan aplikasi lain selain FMCS ini.

### 2.7.4 Prosedur Pengujian

- Mengaktifkan server dan memulai GUI.
- Mengaktifkan aplikasi dummy data
- Dikirimkan 10 data koordinat, fault, dan rpm dari 10 armada secara bersamaan
- Mengamati kemampuan GUI menampilkan data lengkap dari 10 armada secara bersamaan pada tabel monitoring.
- Dikirimkan 40 data posisi dari aplikasi *dummy data* secara bersamaan yang terus bergerak setiap waktunya di sepanjang trayek Soekarno-Hatta.
- Mengamati kemampuan GUI menampilkan posisi dari 40 armada secara bersamaan pada peta geografis.

### 2.7.5 Harapan Hasil Pengujian

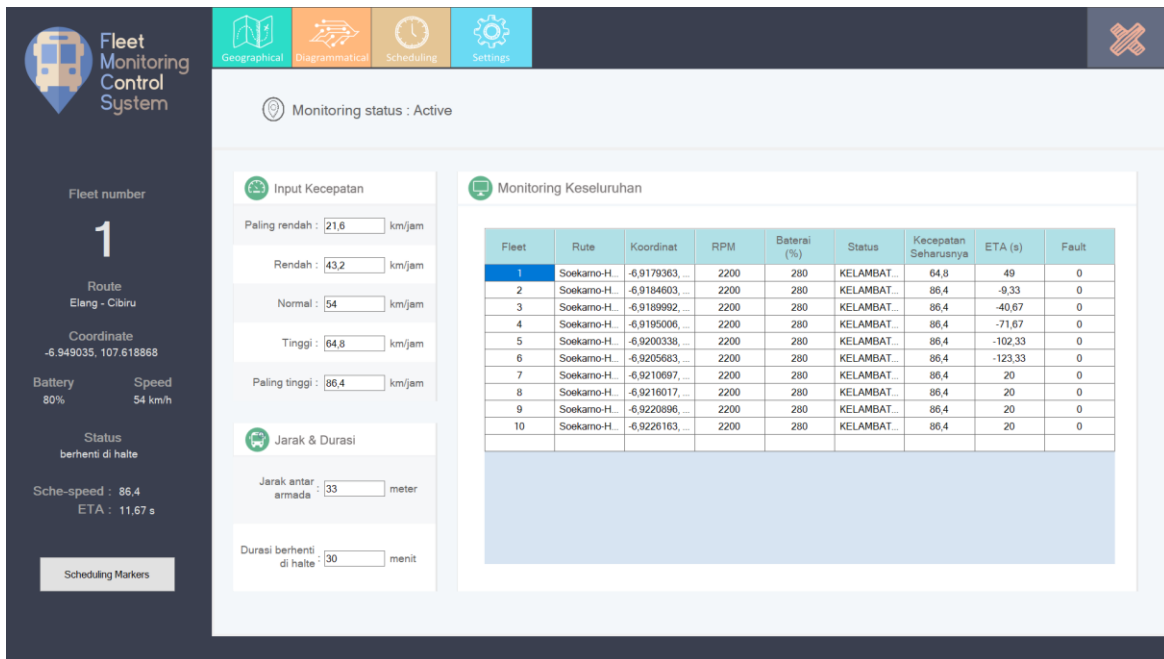
Pada pengujian ini akan diperoleh data analisis dari hasil pengamatan terhadap GUI ketika menerima 10 data fleet secara sekaligus, yaitu longitude latitude, tegangan baterai, rpm, dan kondisi fault, serta tampilan posisi 40 fleet pada peta geografis.

Pada pengujian ini diharapkan seluruh data dapat ditampilkan dengan tepat, baik, dan tanpa menyebabkan aplikasi *lag* atau *error*.

### 2.7.6 Hasil Pengujian

Pengujian tahap pertama dilakukan dengan mengirimkan 10 data lengkap armada berupa koordinat, rpm, fault, dan persentase baterai dari dummy data. Status armada, kecepatan seharusnya, dan ETA merupakan hasil olahan dari algoritma penjadwalan yang telah diintegrasikan dengan GUI. Hasil pengujian adalah sebagai berikut :

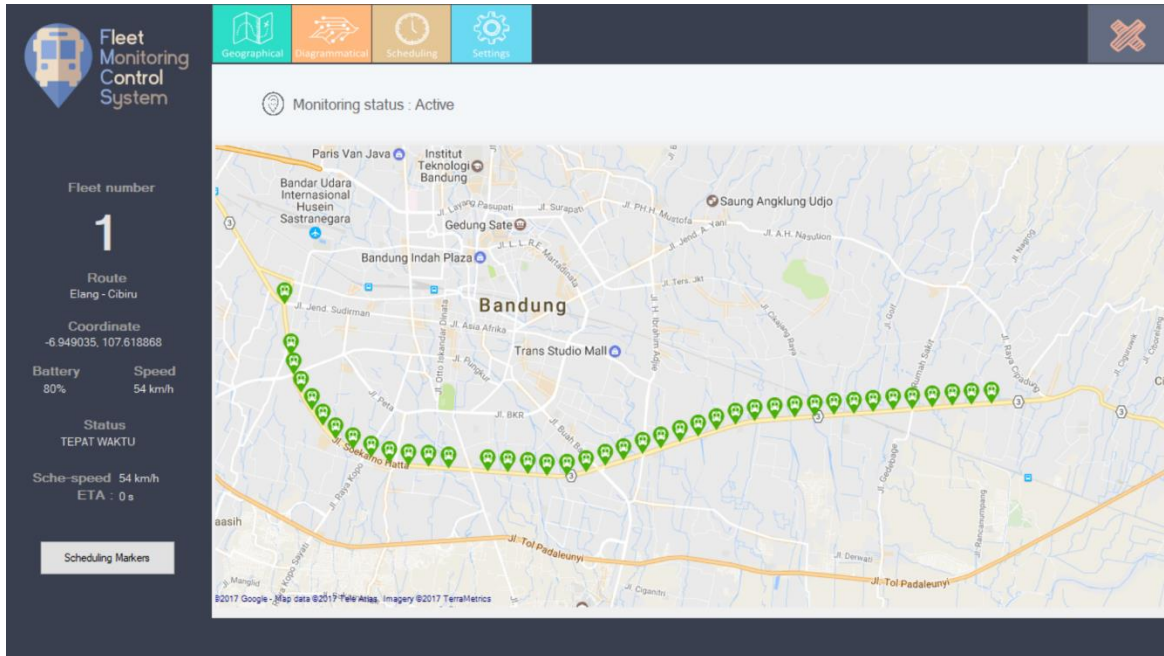




**Gambar 17 Tampilan GUI untuk 10 data armada**

Dari pengamatan ketika dilakukan pengiriman data, GUI dapat menampilkan seluruh data dengan baik tanpa menyebabkan aplikasi *lag/error*.

Pengujian tahap kedua dilakukan dengan mengirimkan data posisi 40 armada dari dummy data, dan kemudian ditampilkan pada peta geografis. Hasil pengujian adalah sebagai berikut:



**Gambar 18 Tampilan GUI untuk posisi 40 armada**

Dari pengamatan yang dilakukan ketika pengujian, dapat diamati bahwa GUI sudah mampu menampilkan posisi 40 kendaraan yang sedang bergerak secara bersamaan tanpa menyebabkan aplikasi *lag/error*.

### 3 KESIMPULAN

#### 3.1 Ketercapaian Spesifikasi

Berdasarkan pengujian-pengujian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut untuk tiap spesifikasi sistem.

**Tabel 5 Ketercapaian spesifikasi**

Spesifikasi	Ketercapaian	Keterangan
Posisi yang dibaca oleh operator pada <i>control station</i> memiliki <i>error</i> maksimal sebesar 6 meter.	100 %	
Dapat membaca data rpm kendaraan, kondisi fault, dan level energi baterai sesuai spesifikasi ECU yang sudah terdapat di <i>guided bus</i> .	100 %	
Fleet hardware mengirimkan data posisi, rpm, fault, dan level energi baterai tiap armada. Data yang dikirim harus dapat dibaca pada GUI di <i>control station</i> maksimal setiap 0.8 detik.	80.2 %	Ketercapaian tidak 100 % dikarenakan adanya faktor turbulensi latensi jaringan. Pada keadaan normal, waktu pengiriman dari <i>fleet hardware</i> ke GUI sekitar 500 ms. Namun, pada kondisi jaringan yang buruk, waktu pengiriman ini dapat meningkat sampai 1300 ms.
Menampilkan 10 armada dan data-data tiap armada yang akan ditampilkan di GUI pada <i>control station</i> . Data-data tersebut adalah posisi, rpm, fault, dan level energi baterai tiap armada.  GUI ini akan dipantau dan dikendalikan oleh operator di <i>control station</i> .	100 %	
Ada algoritma yang dapat mengatur penjadwalan operasi tiap armada.	100 %	
<i>Control station</i> dapat mengirimkan perintah operasional ke armada setiap terjadi perubahan perintah.	100 %	
Data-data komunikasi antara armada dan <i>control station</i> hanya dapat diakses oleh armada yang	100 %	

bersangkutan dan operator pada <i>control station</i> .		
Server dapat menerima 40 koneksi klien dengan tiap klien mengirimkan data tiap 0.8 detik.	100 %	

### 3.2 Saran Pengembangan

Berikut beberapa saran pengembangan untuk FMCS pada *guided bus*.

- Menggunakan *provider* internet yang sudah bekerja sama dengan pihak penyedia FMCS untuk mengurangi biaya internet dan memperbaiki kualitas jaringan, atau dengan menggunakan frekuensi lain di luar jaringan GSM. Namun, untuk penggunaan frekuensi lain ini dibutuhkan infrastruktur tambahan.
- ECU *Monitoring* dibuat menjadi satu kesatuan *board* yang terhubung langsung dengan ECU lain pada *guided bus*.
- Tampilan LCD dipindahkan ke GUI dalam *guided bus*.
- Integrasi server, GUI dengan sebuah *database*. *Database* ini akan menampung semua data yang diterima kecuali data posisi.