

# Graphical User Interface untuk Fleet Monitoring and Control Sistem pada Guided Bus

Aulia Hening Darmasti<sup>1</sup>, Arief Syaichu R<sup>2</sup>, Agung Darmawan<sup>3</sup>, Arif Sasongko<sup>4</sup>

Program Studi Teknik Elektro – STEI

Institut Teknologi Bandung

Bandung, Indonesia

<sup>1</sup>ahdarmasti@gmail.com, <sup>2</sup>arief@stei.itb.ac.id, <sup>3</sup>agungdar68@yahoo.com<sup>3</sup>, <sup>4</sup>asasongko@gmail.com<sup>4</sup>

**Abstrak**—Pada saat ini, belum tersedia fasilitas transportasi massal yang cukup mumpuni di kota Bandung. Kenyamanan dan waktu menjadi faktor utama bagi masyarakat untuk memilih moda transportasi. Sementara pada saat ini diantara semua pilihan transportasi tidak ada satupun yang menjanjikan kenyamanan dan bebas dari macet. Untuk itu dirancang Fleet Monitoring and Control System (FMCS) untuk *guided bus* sebagai jawaban dari ketiadaan pilihan transportasi yang nyaman, aman dan bebas macet. FMCS mampu untuk melakukan pemantauan posisi armada diseluruh trayek, menampilkan data fisik dari masing-masing armada, penjadwalan, hingga memberi komando dari control station agar pergerakan armada tetap sesuai dengan penjadwalan. Salah satu komponen penting dari FMCS ialah GUI (Graphical User Interface). GUI ini berfungsi sebagai perantara antar sistem dan operator untuk saling berinteraksi. Lewat GUI, operator dapat melakukan pemantauan posisi armada baik dengan peta geografis, peta diagramatis, memantau data fisik armada, mengatur penjadwalan, menangani situasi darurat, hingga memberi instruksi pada armada. GUI yang dibangun dengan menggunakan Windows Forms pada Microsoft Visual Studio ini memiliki kapasitas untuk menampilkan hingga 40 posisi armada dan 10 data lengkap armada secara bersamaan tanpa membebani PC yang digunakan. Selain itu GUI ini juga dirancang dengan memperhitungkan kenyamanan pengguna dalam pemakaiannya. Kedepannya, GUI dapat dikembangkan dengan mengurangi ketelitian array segmentasi. Hal ini disebabkan karena urgensi untuk melakukan segmentasi trayek sesuai dengan panjang bus tidak terlalu dibutuhkan.

**Kata kunci**—Pemantauan, Pengendalian, Antarmuka, Desain Interaksi.

## I. PENDAHULUAN

Beberapa dasawarsa terakhir, dunia telah mengalami perkembangan teknologi yang cukup pesat. Hal ini menyebabkan meningkatnya mobilitas penduduk dan tentu saja, meningkatnya juga kebutuhan akan transportasi. Pemerintah kota Bandung, sesuai dengan itu, turut menyediakan berbagai jenis transportasi massal bagi warganya.

Tetapi ternyata solusi yang disediakan oleh pemerintah belum cukup menjadi pilihan yang baik bagi masyarakat. Kemudahan dan kenyamanan menjadi sorotan utama bagi masyarakat dalam memilih transportasi. Masalah utama yang tidak dapat dihindari dari transportasi massal yang telah ada ialah tidak adanya sebuah sistem yang dapat memantau dan mengendalikan moda transportasi.

Bekerja sama dengan PT. LEN Industri yang sedang merintis prototipe *guided bus* berbahan bakar listrik, Fleet Monitoring

and Control Sistem untuk *guided bus* dirancang sebagai upaya dalam mengatasi permasalahan tersebut. Salah satu subsistem yang dibutuhkan dalam FMCS adalah GUI yang digunakan oleh operator untuk melakukan pemantauan.

Dalam makalah ini akan dijabarkan mulai dari perancangan, implementasi, hingga pengujian dan kesimpulan dalam pembuatan GUI untuk FMCS.

## II. GUI UNTUK FMCS PADA GUIDED BUS

Pada bagian ini dijabarkan seluruh ilmu dan referensi yang digunakan selama pengerjaan GUI untuk FMCS pada *Guided Bus*.

### A. Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (suite) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup compiler, SDK, Integrated Development Environment (IDE), dan dokumentasi (umumnya berupa MSDN Library). Compiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, dan masih banyak lagi. [1]

### B. Bahasa Pemrograman C#

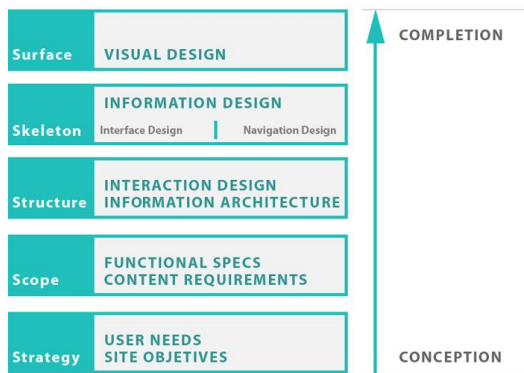
C# atau yang dibaca C sharp adalah bahasa pemrograman sederhana yang digunakan untuk tujuan umum, dalam artian bahasa pemrograman ini dapat digunakan untuk berbagai fungsi misalnya untuk pemrograman server-side pada website, membangun aplikasi desktop ataupun mobile, pemrograman game dsb. Dalam pemrograman C# (mungkin juga berlaku untuk beberapa pemrograman lainnya) memiliki 5 struktur dasar yaitu Resource atau Library, Namespace, Nama Class, Deklarasi Method, Method atau Command. [2]

### C. Object Oriented Programming

OOP (Object Oriented Programming) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, kemudian objek-objek itu saling berkomunikasi, dan saling berkiriman pesan kepada objek yang lain. OOP memiliki konsep

yang terbagi-bagi dalam beberapa bagian berikut: kelas abstrak, enkapsulasi, pewarisan, dan polimorfisme. [3]

#### D. UX Design



Gambar 2. 1 Hirarki Desain Interaksi

User Experience berkonsentrasi pada bagaimana sebuah produk terasa dan apakah itu memecahkan masalah bagi pengguna. UX Design adalah sebuah proses membuat sebuah website atau aplikasi yang dibuat mudah untuk digunakan serta tidak membingungkan ketika digunakan oleh pengguna. Lima elemen utama dari User Experience Design menurut Jasse James Garret yaitu Strategi, Scope, Structure, Skeleton, dan Surface.

Adapun unsur-unsur utama yang harus diperhatikan dalam UX Design adalah kebutuhan pengguna, kebutuhan fungsional dan ketersediaan informasi/konten, arsitektur informasi, hingga layouting dan harmonisasi tampilan.[4]

### III. DESAIN DAN IMPLEMENTASI

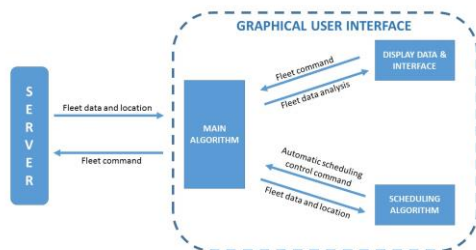
Sistem FMCS untuk guided bus ini terdiri dari tiga subsistem utama, yaitu fleet hardware, server, dan GUI. Ketiga subsistem ini terintegrasi menjadi satu kesatuan.

Spesifikasi dari perancangan GUI untuk FMCS pada guided bus adalah sebagai berikut:

1. Mampu menampilkan data-data berupa posisi, rpm kendaraan, kondisi fault, dan level energi baterai tiap armada,
2. Mampu mengolah data yang diterima dengan algoritma penjadwalan,
3. Mampu mengirimkan perintah operasional ke armada setiap terjadi perubahan perintah.

Serta tambahan fitur mudah untuk digunakan (user friendly).

GUI secara umum berfungsi untuk menampilkan data-data yang dikirimkan fleet hardware lewat server. Singkatnya, sistem GUI dapat digambarkan dalam diagram blok berikut:



Gambar 3. 1 Diagram Blok GUI

Pada GUI, terdapat tiga fungsi utama yang digunakan untuk menjalankan FMCS ini, yaitu algoritma utama, display data & interface, serta algoritma penjadwalan.



Gambar 3. 2 Diagram Blok Algoritma Utama

Algoritma utama menerima 3 jenis data. Yang pertama fleet data dan posisi dari server, scheduling command dari algoritma penjadwalan, dan fleet command dari operator. Semua data ini diolah dalam algoritma utama dan dikeluarkan dalam bentuk analisis data fleet, data-data fleet, serta komando fleet yang akan diberikan kepada masing-masing hardware lewat server.

Berikut akan dijelaskan prosedur implementasi untuk menampilkan posisi armada pada peta geografis, pada peta diagramatis, algoritma utama GUI, juga perancangan UX Design dari GUI.

#### A. Peta Geografis

Pada spesifikasi umum FMCS, diinginkan sebuah tampilan yang dapat menangani penyimpanan data GPS. Hal ini menjadi dasar dibuatnya array segmentasi koordinat geografis. Apabila ada data koordinat dari GPS yang keluar dari jalan, marker tetap dapat ditampilkan di rute yang benar.

Langkah-langkah yang dilakukan diantaranya adalah mencari koordinat garis lurus pada Google Map, mengekstraknya hingga mendapatkan koordinat garis lurus, kemudian menginput koordinat-koordinat garis lurus tersebut pada file pencacah hingga didapat coordinate interval untuk setiap 12 meter.

Didalam file coordinate interval, pertama-tama, dilakukan metode untuk mencari jarak terdekat antara dua titik koordinat dengan menggunakan **Formula Haversine**, yang mempunyai rumus sebagai berikut :

$$\begin{aligned}
 1. \quad a &= \left( \sin \frac{\Delta lat}{2} \right)^2 + \cos(lat1rads) \times \\
 &\quad \cos(lat2rads) \times \left( \sin \frac{\Delta long}{2} \right)^2 \\
 2. \quad c &= 2 \times \text{atan2}(\sqrt{a}, \sqrt{1-a}) \\
 3. \quad d &= R \times c
 \end{aligned}$$

$\Delta lat$  merupakan selisih antara latitude 1 dan latitude 2,  $\Delta long$  selisih dari longitude 1 dan longitude 2, dan R jari-jari bumi.

Selanjutnya akan dicari azimuth dengan menggunakan bearing. Azimuth merupakan sudut yang terbentuk antara titik utara peta dengan titik sasaran yang dapat dicari dengan menggunakan rumus berikut:

$$bearing = \left( \text{atan2} \left( \Delta long, \log \left( \frac{\tan \left( \frac{lat2}{2} + \frac{\pi}{4} \right)}{\tan \left( \frac{lat1}{2} + \frac{\pi}{4} \right)} \right) \right) + 360 \right) \bmod 360$$

Setelah didapatkan azimuth dan jarak, dicari latitude dan longitude konversi hasil perhitungan. Ouput dari file ini berupa array koordinat setiap 12 meter disepanjang rute Soekarno-Hatta dengan posisi tepat berada ditengah jalur. Metode ini dapat diimplementasikan pada seluruh rute manapun.

#### B. Peta Diagramatis

Peta diagramatis merupakan peta yang umum digunakan untuk menunjukkan pergerakan kendaraan pada penggua.

Pembuatan peta diagramatis bertujuan untuk memudahkan pengguna memantau posisi kendaraan dengan tampilan yang lebih mudah dimengerti dibandingkan dengan menampilkan pada peta geografis.

Peta diagramatis ini dibuat dengan cara menghilangkan informasi yang tidak dibutuhkan dan cenderung menggunakan simbol-simbol yang dapat memberi informasi tambahan seperti halte kecil, halte besar, interchange, dan nama halte.

Karena bertujuan untuk memudahkan pengguna memahami trayek, peta diagramatis digambarkan dengan garis lurus. Pada project ini digunakan tampilan orthogonal karena lebih user-friendly.

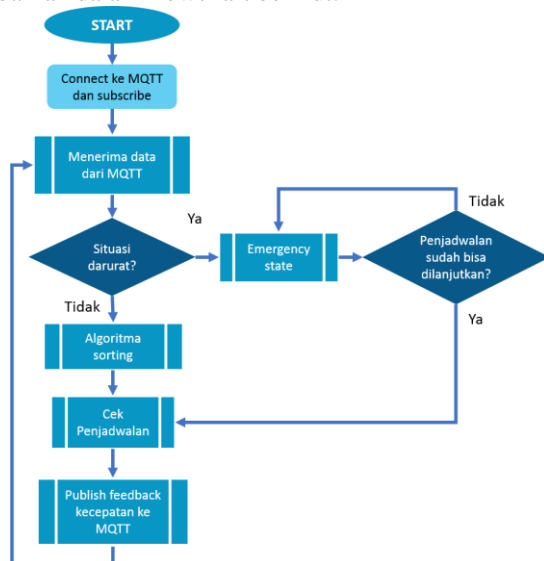
Pemodelan **peta diagramatis** dengan garis lurus ini menyebabkan peta menjadi tidak berskala (karena itu peta diagramatis banyak disebut sebagai **peta tanpa skala**), karena penarikan garis dilakukan hanya berdasarkan pendekatan dan hanya mempertimbangkan tampilan. Peta diagramatis didesain secara manual dengan menggunakan Adobe Illustrator dan kemudian hasilnya di import ke visual .



Gambar 3.3 Peta Diagramatis

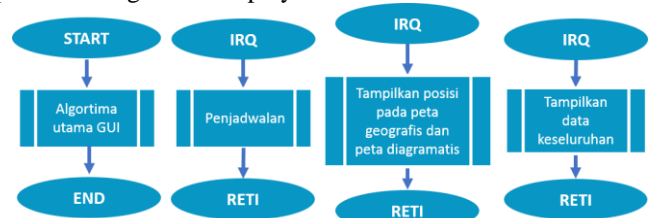
### C. Algoritma Utama GUI

GUI terdiri dari satu algoritma utama yang tersusun dari beberapa fungsi dan metode didalamnya. Algoritma utama merupakan seluruh kesatuan integrasi dari penerimaan data, ekstraksi data, pengolahan data dengan jadwal penjadwalan, menampilkan data yang telah diolah, hingga pengiriman feedback kepada fleet hardware. Satu sistem besar GUI dapat digambarkan dalam flowchart berikut:



Gambar 3.4 Flowchart Algoritma Utama GUI

Disamping satu proses utama tersebut, terdapat beberapa IRQ (Interrupt ReQuest) dalam bentuk timer yang akan menginterrupt proses utama jika dipanggil. Ketiga IRQ itu diantaranya adalah algoritma penjadwalan, algoritma display posisi dan algoritma display data.

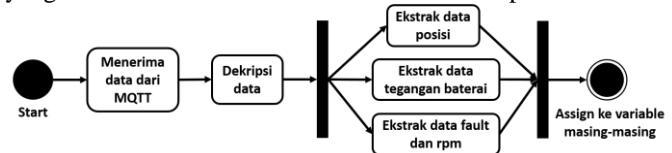


Gambar 3.5 Flowchart Keseluruhan GUI

Berikut ini akan dijabarkan lebih lanjut mengenai subproses dalam sistem GUI.

#### 1. Mengekstrak Data dari MQTT

GUI terhubung dengan MQTT menggunakan metode subscribe. Subscribe merupakan metode untuk berlangganan pada suatu topik tertentu, sehingga setiap kalinya ada client yang melakukan publish pada suatu topik yang sama, client yang melakukan subscribe akan menerima pesan tersebut.

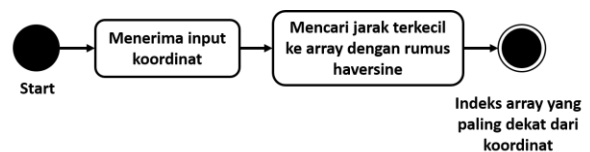


Gambar 3.6 Diagram Aktivitas Penerimaan Data

Data yang telah didekrip kemudian diklasifikasikan sesuai jenis informasinya masing-masing, diantaranya data posisi, tegangan baterai, fault, juga rpm. Data-data tersebut kemudian di assign ke variable dalam array yang telah disiapkan.

#### 2. Algoritma Sorting

Sistem segmentasi pada peta geografis diimplementasikan dengan cara mencari posisi terdekat dalam array dari koordinat yang dikirimkan. Pengolahan algoritma sorting dibantu dengan fungsi haversine yang akan menghasilkan perhitungan jarak dari masing-masing array koordinat ke koordinat MQTT yang diterima. Detail diagram aktivitas algoritma sorting adalah sebagai berikut:



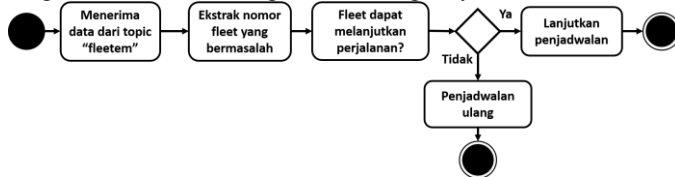
Gambar 3.7 Diagram Aktivitas Algoritma Sorting

Pada algoritma sorting ini, digunakan fungsi haversine yang berbeda dengan yang digunakan untuk melakukan segmentasi array. Fungsi haversine yang digunakan disini tidak memperhitungkan azimuth dan bearing. Hal ini disebabkan karena satuan pada algoritma sorting hanya berkisar di satuan meter sehingga pengaruh kelengkungan bumi dapat diabaikan.

#### 3. Penanganan Situasi Darurat

Pada GUI disediakan algoritma untuk mengatasi situasi darurat. Pada kasus ini, fleet hardware yang mengalami masalah akan mengirim nomor fleet mereka ke topic “fleetem” ke MQTT. Ketika GUI menerima topik ini, otomatis setiap

armada akan diinstruksikan untuk berhenti ke halte terdekat didepannya dan sistem FMCS dihentikan sementara. Berikut diagram aktivitas dari algoritma emergency:

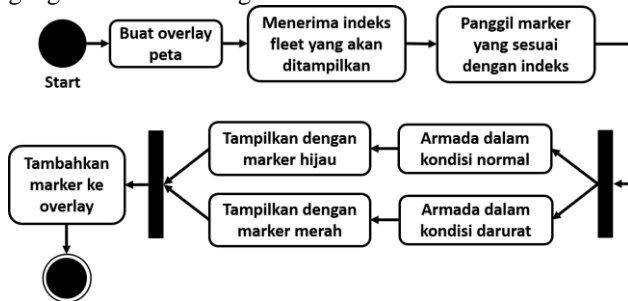


Gambar 3. 8 Diagram Aktivitas Situasi Darurat

Tugas operator kemudian memastikan apakah fleet yang bermasalah dapat melanjutkan perjalanannya atau tidak. Jika iya, penjadwalan tetap dilakukan untuk jumlah fleet yang sama, sementara jika tidak, penjadwalan akan diatur ulang untuk jumlah fleet n-1.

#### 4. Menampilkan posisi dan data armada

Layar utama dari GUI FMCS adalah halaman dengan peta geografis. Halaman ini akan menampilkan seluruh posisi armada secara bersamaan dalam satu screen. Pada halaman ini disediakan fitur berupa marker penanda, apabila ada fleet yang mengalami situasi darurat, maka warna markernya akan berubah menjadi merah, sementara dalam kondisi normal akan berwarna hijau. Alur untuk menampilkan marker pada peta geografis adalah sebagai berikut:



Gambar 3. 9 Diagram Aktivitas Marker Geografis

Halaman selanjutnya menunjukkan tampilan dalam peta diagramatis. Proses display pada peta diagramatis ini berkesinambungan dengan peta geografis. Pertama-tama akan diterima input berupa indeks array geografis, kemudian proses penampilan marker armada pada peta diagramatis akan dilakukan secara manual dengan menggunakan picturebox dan segmentasi point x-y. Proses segmentasi hingga didapatkan koefisien pengali point x dan y dapat dilihat pada tabel berikut:

dx	dy	Halte dan Belokan	x	y	dxxy	fdxy	xi	fxi	i-ke	lx	ly
51	52	Elang	102	174							
1	62	Pasirkoja	153	226	72,83543094	73	511	511	511	0,099804305	0,10176125
42	44	Belokan1	154	288	62,00806399	62	411,821	412	923	0,002427184	0,15048544
23	25	Kopo	196	332	60,82697428	61	405,179	405	1328	0,103753086	0,10859259
12	0	Belokan2	219	357	33,97176475	34	163,348	163	1491	0,140981595	0,15349693
82	0	Cibaduyut	231	357	12	12	57,6522	58	1549	0,206896552	0
12	0	Moh Toha	313	357	82	82	545	545	2094	0,150458716	0
114	0	Belokan3	427	357	114	114	558,684	559	2653	0,203935599	0
11	-8	Batununggal	438	349	13,60147051	14	54,3165	54	2707	0,203703704	-0,1481481
62	-64	Bubat	500	285	89,10667764	89	289	289	2996	0,214532872	-0,2214533
6	-7	Belokan4	506	278	9,219544457	9	31,3163	31	3027	0,193548387	-0,2258065
89	0	Kircon	595	278	89	89	309,684	310	3337	0,287096774	0
116	0	Margahayu	711	278	116	116	988	988	4325	0,117408907	0
113	0	Gedebage	824	278	113	113	954	954	5279	0,118448637	0
95	0	Cibiru	919	278	95	95	920	920	6199	0,10326087	0

Gambar 3. 10 Tabel Data Pencarian Koefisien

Dengan

dx : jarak antar dua point x  
 dy : jarak antar dua point y  
 x : point patokan  
 y : point patokan

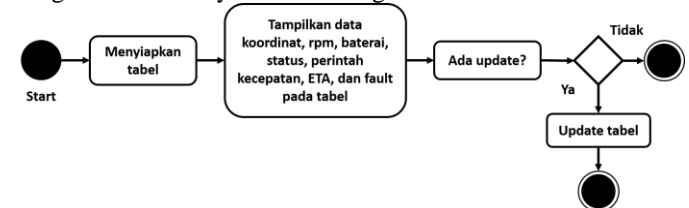
kx : koefisien pengali x  
 ky : koefisien pengali y  
 xi : range array yang ditampung antar dua titik  
 fxi : range array antar dua titik yang sudah dibulatkan  
 dxxy : jarak antara dua titik  
 fdxy : jarak antar dua titik yang dibulatkan

Diagram aktivitas untuk menampilkan marker diagramatis adalah sebagai berikut:



Gambar 3. 11 Diagram Aktivitas Marker Diagramatis

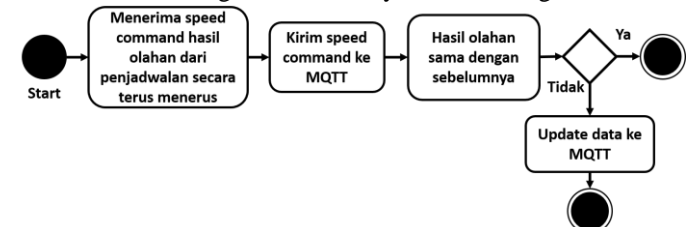
Selain dapat menampilkan seluruh posisi secara bersamaan, GUI juga dapat menampilkan seluruh data yang diterima. Diagram aktivitasnya adalah sebagai berikut:



Gambar 3. 12 Diagram Aktivitas Update Tabel Data

#### 5. Memberi komando kecepatan

Seperti yang telah dijelaskan, salah satu spesifikasi GUI ialah dapat mengirimkan feedback berupa instruksi kecepatan pada fleet hardware. Diagram aktivitasnya adalah sebagai berikut:



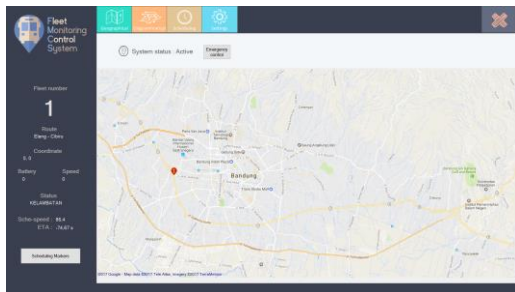
Gambar 3. 13 Diagram Aktivitas Komando Kecepatan

#### D. Tampilan GUI (UX Design)

Tampilan GUI dibuat dengan skema warna triadic. **Skema warna triadic** membawa kontras skema warna yang tinggi dengan mempertahankan nada yang sama[5]. Pembuatan skema triadic dibuat dengan memilih tiga warna dengan garis dengan jari-jari yang sama dalam roda warna. Pada skema warna GUI ini dipilih dengan jari-jari 65% dari roda warna agar warna yang ditampilkan dapat membawa **kesan segar, bersemangat dan tidak menjemukan**.

Default screen pada GUI diletakkan pada peta geografis karena menampilkan posisi armada pada peta merupakan inti dari pembuatan GUI ini. Pada bar disebelah kiri, diberikan informasi mengenai data personal GUI. Mousehover pada masing-masing marker akan memunculkan tooltip mengenai nomor marker, dan mouseclick pada masing-masing marker akan mengubah informasi pada sidebar tergantung dari marker yang di klik. Misalkan operator melakukan klik pada marker fleet 4, maka sidebar akan menampilkan informasi mengenai fleet 4.





**Gambar 3. 14 Tampilan Halaman Peta Geografis**

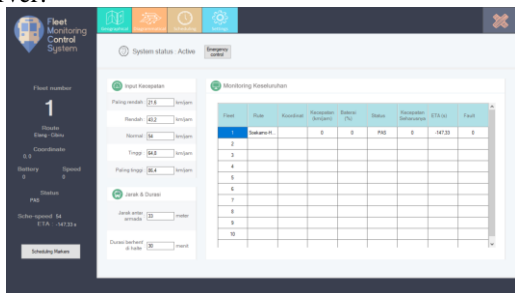
Tombol scheduling marker pada bagian bawah sidebar dapat menampilkan seluruh marker dari range penjadwalan yang harus dituruti oleh armada. Karena itu scheduling marker ini ditempatkan sebagai toggle button agar marker penjadwalan tidak memenuhi tampilan peta geografis.

Pemilihan warna kontras pada penyusunan peta diagramatis bertujuan agar operator dapat melihat dengan jelas perbedaan satu rute dengan rute lainnya. Pemilihan skema warna untuk metro map ini telah banyak digunakan pada metro map umumnya.



**Gambar 3. 15 Tampilan Halaman Peta Diagramatis**

Sementara pada halaman scheduling, di kolom sebelah kiri operator dapat mengubah input kecepatan penjadwalan, jarak antar bus, dan durasi berhenti disetiap halte. Pada dasarnya angka-angka ini sudah ditentukan oleh sistem, namun operator dapat mengubahnya jika dibutuhkan. Sementara pada kolom sebelah kanan, terdapat tabel monitoring keseluruhan. Pada tabel ini operator dapat mengamati seluruh data yang diterima dari server.



**Gambar 3. 16 Tampilan Halaman Penjadwalan**

Pada panel utama (disebelah status sistem), terdapat tombol emergency control. Jika terjadi situasi darurat, maka tombol ini akan berwarna merah. Operator harus melakukan konfirmasi keberlanjutan penjadwalan fleet dengan tombol ini.

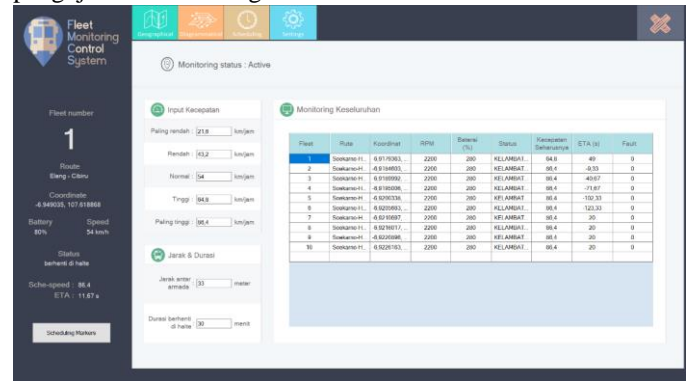
#### IV. HASIL PENGUJIAN

##### A. Pengujian Kemampuan Fungsional GUI

Pada pengujian ini, akan diuji kemampuan fungsional GUI.

Syarat yang harus dipenuhi untuk pengujian ini yaitu koneksi internet yang cukup baik dan PC yang digunakan tidak menjalankan aplikasi lain selain aplikasi FMCS.

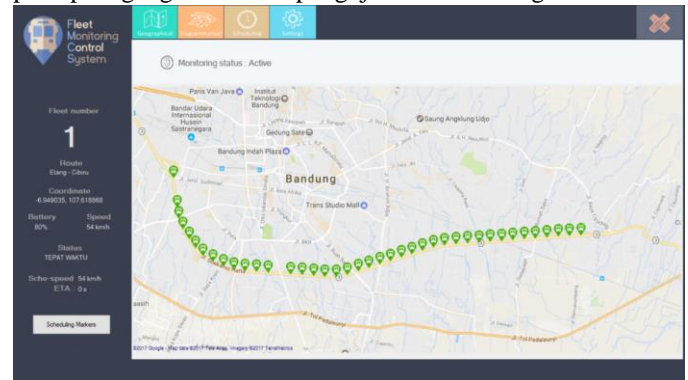
Pengujian tahap pertama dilakukan dengan mengirimkan 10 data lengkap armada berupa koordinat, rpm, fault, dan persentase baterai dari dummy data. Status armada, kecepatan seharusnya, dan ETA merupakan hasil olahan dari algoritma penjadwalan yang telah diintegrasikan dengan GUI. Hasil pengujian adalah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



**Gambar 4. 1 Pengujian 10 Data Lengkap**

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 4. 2 Pengujian 40 Posisi Fleet**

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

##### B. Pengujian UX Design dan UI Design

Pada pengujian ini, lingkup spesifikasi yang akan diuji adalah fitur FMCS, yaitu aplikasi memiliki UX Design yang user-friendly bagi pengguna. Subjek penelitian akan diberikan penjelasan singkat mengenai GUI kemudian diminta untuk mengoperasikan dan mengisi kuesioner.

Syarat yang digunakan pada pengujian ini adalah subjek penelitian sebelumnya tidak pernah mengetahui cara kerja dan sistem GUI FMCS, serta diasumsikan subjek mengerti bahasa inggris sederhana, mengenal bahasa operasi sederhana dan mampu mengoperasikan PC.

Pengujian pertama merupakan pengujian UX Design yang dilakukan secara langsung kepada 12 orang subjek. Sementara pengujian kedua merupakan pengujian UI Design yang dilakukan secara virtual kepada 11 orang subjek lewat screenshot. Data hasil pengujian ini akan diperoleh dari kuesioner yang diisi subjek setelah mengoperasikan GUI. Data yang akan diamati adalah:

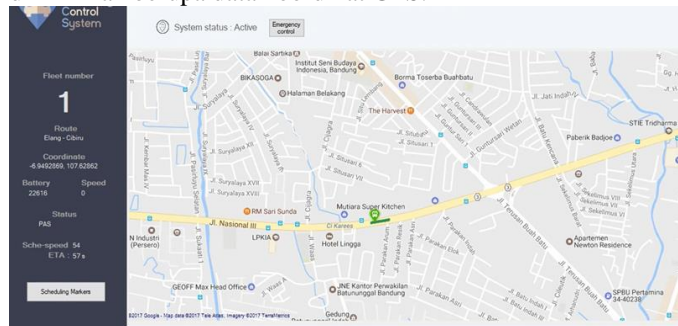
- Ketersampaian konten/informasi dari UX Design
- Keefektifan User Interface
- Kenyamanan user dalam menggunakan GUI
- Estetika tampilan User Interface

Dari hasil kuesioner pada percobaan tahap pertama, untuk penilaian User Interface didapat hasil 91.7% subjek cukup nyaman dengan tampilan GUI, 91.7% mudah memahami GUI dan 100% setuju bahwa GUI cukup efektif. Sementara hasil kuesioner pada percobaan tahap kedua, didapat hasil 90.9% subjek cukup nyaman dengan tampilan GUI, 90.9% mudah memahami GUI dan 100% setuju bahwa GUI cukup efektif.

Dari seluruh hasil pengujian terhadap UX dan UI Design, dengan nilai rata-rata diatas 90% dapat disimpulkan GUI sudah cukup user-friendly bagi pengguna.

#### C. Pengujian di Lapangan

Pengujian dilakukan pada rute Soekarno-Hatta dengan segmen dimulai dari perempatan Jalan Cijagra hingga Perempatan Buah Batu. Test dilakukan dengan cara membawa fleet hardware dengan menggunakan motor. Data yang dikirimkan berupa data koordinat GPS.



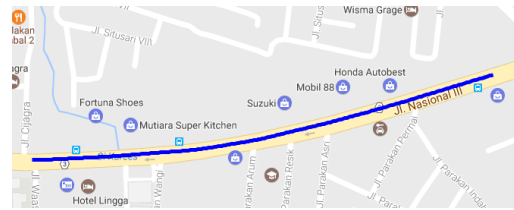
Gambar 4. 3 Pengujian di Lapangan

Dari gambar diatas, dapat disimpulkan GUI sudah dapat menampilkan marker fleet 1 beserta marker range penjadwalannya. Selain itu data-data personal fleet juga sudah dapat ditampilkan pada sidebar.

Pada pengujian ini juga dilakukan dua perekaman rute. Yang pertama adalah rute yang ditampilkan pada GUI dan yang kedua rute yang diplot dari hasil pengiriman koordinat GPS salah satu anggota tim. Hasil pengujian adalah sebagai berikut:



Gambar 4. 4 Rute Koordinat GPS HP



Gambar 4. 5 Rute pada GUI

Dari perbandingan kedua rute diatas, dapat disimpulkan GUI sudah dapat handle simpangan yang dikirim GPS. Meskipun terdapat penyimpangan pada peta geografis, lokasi yang ditampilkan pada GUI tetap berada pada lajur tengah rute seharusnya.

#### D. Pengujian Pengiriman Komando Kecepatan

Pengujian dilakukan bersamaan dengan pengujian di lapangan. Data yang dikirimkan fleet hardware berupa data posisi akan diolah oleh GUI algoritma penjadwalan. GUI kemudian mengirimkan instruksi kecepatan setiap terjadi perubahan pada instruksi kecepatan. Hasil pengujiannya fleet hardware sudah dapat menerima instruksi kecepatan dari GUI.



Gambar 4. 6 Pengujian Pengiriman Komando

## V. KESIMPULAN DAN SARAN

Dari hasil pengujian yang telah dilakukan, GUI FMCS sudah dapat menampilkan 40 posisi fleet secara bersamaan dan 10 data lengkap armada berupa koordinat, rpm, fault, dan persentase baterai secara bersamaan dan tanpa menyebabkan lag/error pada PC. Selain itu GUI FMCS juga memiliki UX dan UI Design yang sudah memenuhi user needs sehingga dapat disimpulkan GUI FMCS memiliki tampilan yang cukup user friendly.

Saran kedepannya, segmentasi di peta geografis GUI, ketelitiannya tidak perlu hingga mencapai 12 meter. Dengan memperhitungkan panjang bis 12 meter dan bis sedang melakukan pergerakan, ketelitian dapat dikurangi hingga 15 meter (12 meter panjang bis + 6 meter setengah panjang bis). Array diusahakan untuk dibuat seminimal mungkin agar proses kalkulasi lebih cepat dan sistem semakin ringan.

#### DAFTAR PUSTAKA

- [1] Wikipedia, [https://id.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://id.wikipedia.org/wiki/Microsoft_Visual_Studio)
- [2] Codepolitan, <https://www.codepolitan.com/pengenalan-bahasa-pemrograman-c-587effa1cb95b>
- [3] Amaliah, Fatiha. <https://fatihamaliah.wordpress.com/2013/04/02/pengertian-konsep-oop-object-oriented-programming/>
- [4] Ardisasmita, Adam. <https://ardisaz.com/2015/03/24/lima-elemen-dari-user-experience-design>
- [5] Multimedia Indonesia, <http://www.multimediaindonesia.net/2016/03/teori-warna-dalam-desain.html>

