Graphical User Interface for Fleet Monitoring and Control System on Guided Bus

Aulia Hening Darmasti[1], Arief Syaichu R[2], Agung Darmawan[3], Arif Sasongko[4]
Electrical Engineering Major – STEI
Institut Teknologi Bandung
Bandung, Indonesia
[1]ahdarmasti@gmail.com, [2]arief@stei.itb.ac.id, agungdar68@yahoo.com[3], asasongko@gmail.com[4]

***Abstract*—Comfortability and time are the main factors for citizen to choose the transportation option. Meanwhile there is none of transportation option which promise comfort and traffic-free. For those reasons, Fleet Monitoring and Control System (FMCS) for guided bus are designed to fulfilled the absence of convenient, safe, and traffic-free transportation options. FMCS is able to monitor fleet positions throughout the route, displaying physical data of each fleet, maintaining scheduling algorithm, and giving command from control station so that fleet movement remains in accordance with the scheduling. One of important part of FMCS is GUI (Graphical User Interface). GUI serves as an intermediary between systems and operators to interact each other. Through GUI, operators can monitor fleet positions with geographic maps, diagrammatic maps, monitor fleet physical data, manage scheduling, handle emergency situations, and giving command to fleet. This GUI built using Windows Forms in Microsoft Visual Studio and has aa capacity to display up to 40 fleet positions and 10 complete fleet data simultaneously without burdening the PC that used. In addition, GUI is also designed to take into account the convenience of users in its use. In the future, GUI can be developed by reduce the accuracy of segmentation array. Segmenting route display according to the bus length not that urgent for this GUI.**

***Keyword*—Monitoring, Controlling, Interface, UX Design.**

I. INTRODUCTION

In the last few decades, the world has undergone rapid technological developments. This has led to an increase in population mobility and, of course, the increased need for transportation. Bandung city government, in accordance with it, also provides various types of mass transportation for its citizens.

But in fact, the solution provided by government hasn't been good enough yet to be first option for the citizen. Practice and convenience are the main focus for people in choosing transportation option. The main problem that can't be avoided from existing mass transport is the absence of a system that can monitor and control the transportation.

In cooperation with PT. LEN Industry, that now developing the prototype of electrically guided bus, Fleet Monitoring and Control System for guided bus is designed as an effort to overcome the problem. One of the subsystems that needed in FMCS is the GUI that used by operators to perform monitoring.

This paper will explain the detail process start form the design, implementation, testing, and conclusions in the manufacture of GUI for FMCS.

II. GUI FOR FMCS ON GUIDED BUS

This section will explain the entire acknowledge and references used during the work of GUI for FMCS on Guided Bus.

A. Microsoft Visual Studio

Microsoft Visual Studio is a complete suite of software that can be used to develop applications, whether business applications, personal applications, or application components, in the form of console applications, Windows applications, or Web applications. Visual Studio includes compilers, SDKs, Integrated Development Environment (IDE), and documentation (generally an MSDN Library). The compilers included in the Visual Studio package include Visual C ++, Visual C #, Visual Basic, Visual Basic .NET, and more. [1]

B. C# Programming Language

C# (usually known by C-Sharp) is a simple programming language that is used for general purposes, in the sense that this programming language can be used for various functions such as for server-side programming on websites, building desktop or mobile applications, game programming and so on. C # programming (may also apply to some other programming) has 5 basic structure i.e. Resource or Library, Namespace, Class Name, Method Declaration, Method or Command. [2]

C. Object Oriented Programming

OOP (Object Oriented Programming) is a programming method that based on an object. The purpose of OOP was created is to facilitate the development of the program by following the existing models in daily life. Every part of a problem is an object, and the object itself is a composite of some smaller object again. A large object is formed from several smaller objects, then the objects will communicate with each other, and each send a message to another object. OOP has several concepts in the following sections: abstract class, encapsulation, inheritance, and polymorphism. [3]

D. UX Design

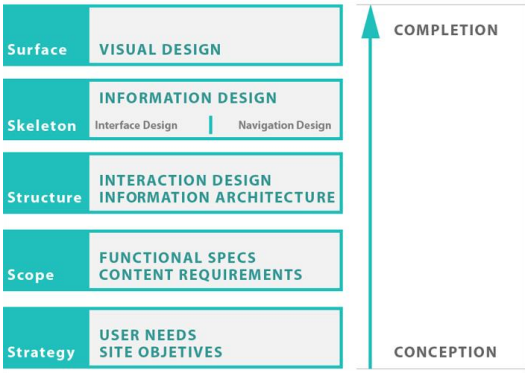

Figure 2. 1 UX Design Hierarchy

User Experience concentrates on how a product feels and whether it solves a problem for the user. UX Design is a process of creating a website or application that would be easy to use and not confusing when used by the user. The five main elements of User Experience Design by Jasse James Garret are Strategy, Scope, Structure, Skeleton, and Surface.

The main elements that must be considered in UX Design is the needs of users, functional needs and delivery of an information/content, information architecture, and to layout and harmonize the display. [4]

III. DESIGN AND IMPLEMENTATION

FMCS on guided bus consists of three main subsystems, namely fleet hardware, server, and GUI. These three subsystems are integrated into one unity. The specifications of GUI designed for FMCS on guided bus are as follows:

1. Able to display data in the form of position, vehicle rpm, fault condition, and energy level of each fleet battery,
2. Able to process data received with scheduling algorithm,
3. Able to send operational orders to the fleet every time a command change occurs.

And an additional user friendly features.

GUI generally serves to display data that sent from fleet hardware through the server. In short, the GUI system can be described in the following block diagram:

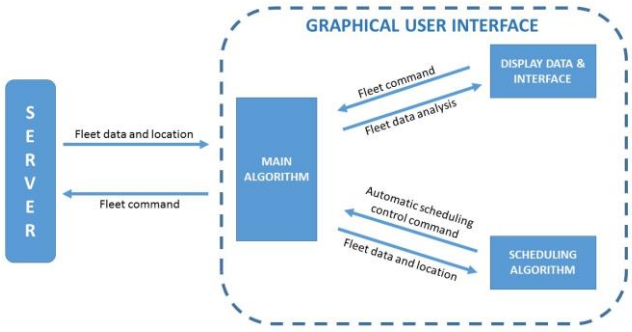

Figure 3. 1 GUI Block Diagram

This GUI had three main functions to run FMCS. Those are main algorithm, display data & interface, and scheduling algorithm.

Figure 3. 2 Main Algorithm Block Diagram

Main algorithm receives 3 types of data. The first is fleet data and the positions from the server, scheduling command from the scheduling algorithm, and the fleet command from operator. All or this data is processed in the main algorithm and issued in the form of analysis of fleet data, fleet data, and fleet command that will be given to every hardware through the server.

The following part will describe the implementation procedure for displaying fleet positions on the geographical map, on the diagrammatical map, the main algorithm of GUI, and also the UX Design procedure for GUI.

A. Geographical Map

In FMCS general specification, a display that can handle GPS data aberrations is needed. This is the main reason of the making of the array of geographical coordinate segmentation. If there is a missed coordinate data from GPS, the marker can still be displayed on the correct route.

The first steps taken is to find the straight line coordinates on the Google Map, then extracting them to get the straight line coordinates, then input those coordinates to an enumerator file to be processed in the making of 12 meter-interval array of coordinates.

Within the enumerator file, first, a method that used to find the closest distance between two coordinate points is using **Haversine Formula**, which has a formula as a following:

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

$\Delta\, lat$ is the difference between first latitude and second latitude. Meanwhile, $\Delta\, long$ is the difference between first longitude and second longitude. R is the radius of earth.

Next, we will look for Azimuth by using bearing formula. Azimuth is the angle formed between the north point of the map with a target point that can be searched by using the following formula:

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

After obtaining the azimuth and distance, we'll look for latitude and longitude of the calculation result. Output of this enumerator file is an array of coordinates ever 12 meters along the Soekarno-Hatta route with the position right in the middle of the road. This method can be applied for any route.

B. Diagrammatical Map

Diagrammatical map is a common map that used to show whether fleet has been passed some shelter or yet. Diagrammatical map creation aims to make it easier for users to monitor vehicle position with a view that's easier to understand than geographical map view.

This diagrammatic map is created by eliminating unneeded information and tends to use symbols that can provide addition information such as small shelters, large shelters, interchange, and shelter names.

Because it aims to make user easier to understand the route, the diagrammatic map is depicted in a straight line. This project used orthogonal display because it's more user-friendly to see.

Modelling the **diagrammatic map** with this straight line causes the map to be uncharted (hence the diagrammatic map usually known as un-scaled map), since the drawing is done only from the basis of the approach and priorities the aesthetic of the view. Diagrammatic maps are designed manually using Adobe Illustrator and then the results are imported into the visual studio.

Figure 3. 3 Diagrammatic Map

C. The Main Algorithm of GUI

GUI consists of one main algorithm, composed of several functions and methods in it. The main algorithm is the whole unity of integration of data reception, data extraction, data processing with scheduling algorithm, displaying data that has been processed, and the process of sending feedback to fleet hardware. This big system can be described in the following flowchart:

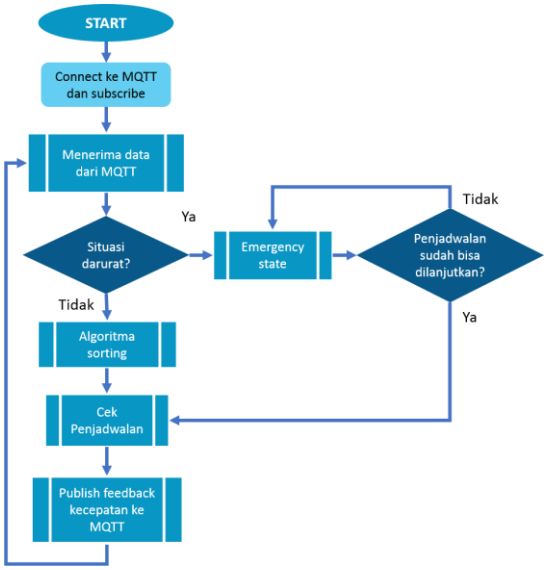

Figure 3. 4 GUI Main Algorithm Flowchart

Besides one main process, there are several IRQs (Interrupt ReQuest) in the form of timers that will interrupt the main process every time they're called. The three IRQs is scheduling algorithm, the display positioning algorithm, and the data display algorithm.

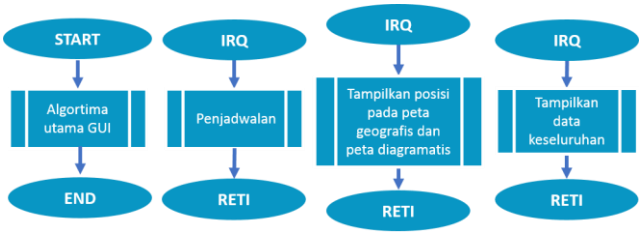

Figure 3. 5 Flowchart Entire GUI System

The following sentences will describe further on sub-processes in the GUI system.

1. Extracting Data from MQTT

GUI connected to MQTT using subscribe method. Subscribe is a method to keep update on one topic. So, every time there is a client who publish on a similar topic, the subscriber will receive the message.

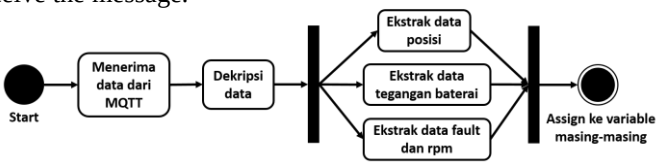

Figure 3. 6 Activity Diagram Data Received

Data that has been decoded then classified according to the type of information respectively, such as data position, battery voltage, fault, and rpm. The data then assigned to the variable in arrays that have been prepared.

2. Sorting Algorithm

Segmentation system in geographical map implemented with finding the nearest distance on the array from coordinate that sent from MQTT. Sorting algorithm processing helped with haversine function, will calculate distance from each coordinate in array of coordinate to that received coordinate from MQTT. The detailed activity diagram of sorting algorithm is as following:

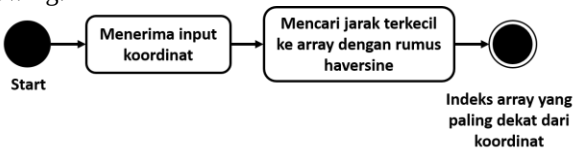

Figure 3. 7 Activity Diagram Sorting Algorithm

In this sorting algorithm, haversine function that used is different with the use of haversine function in array coordinate segmentation. Haversine function that used in this function not including azimuth and bearing in calculation process. This is because the unit in the sorting algorithm only ranges in meters so that the effect of the curvature or the earth can be ignored.

3. Emergency State Handling

In GUI, provided an algorithm to handle emergency state. In this case, fleet hardware that having a emergency situation will send their fleet number to "fleetem" topic with MQTT. Everywhen GUI received this topic, automatically fleet will be instructed to stop in the nearest shelter, and system of GUI will stop for a while. Here is the activity diagram of the emergency algorithm:

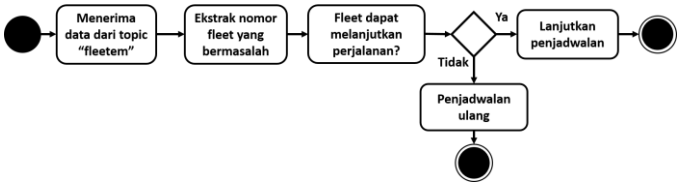

Figure 3. 8 Activity Diagram Emergency State

The operator's task is to ensure whether the problematic fleet can continue its journey or not. If so, the scheduling algorithm will run for the same number fleet. But if not, the scheduling algorithm will be reset for the number of n-1 fleet.

4. Displaying position and fleet data

The main screen of GUI is the screen that show geographical maps. This screen will show the position of all fleet simultaneously in one screen. In this screen, provided several features such as sign marker. On the normal situation, marker will be colored green. But if the fleet on the emergency situation, the marker will be colored red.

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 menjadiree merah, sementara dalam kondisi normal akan berwarna hijau. The diagram activity to show markers on geographical maps is as following:

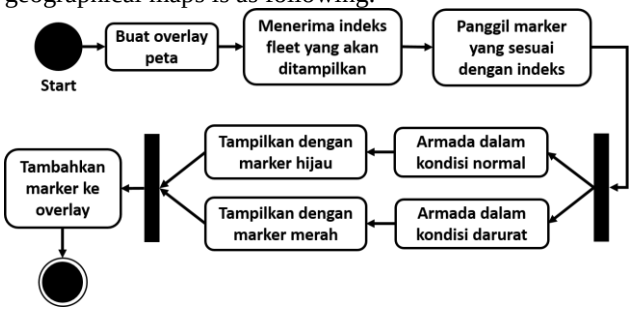

Figure 3. 9 Activity Diagram Geographical Marker

The next screen shows display in diagrammatic maps. Display process in diagrammatical map correlated with geographical maps. First, this algorithm will receive input in the form of geographical array index, then the process of marker display in diagrammatic map will be done manually by using picturebox and x-y point segmentation. The segmentation process until the multiplier coefficients of x and y points are obtained can be seen in this following table:

dx	dy	Halte dan Belokan	x	y	dxy	fdxy	xi	fxi	i-ke	kx	ky
		Elang	102	174							
51	52	Pasirkoja	153	226	72,83543094	73	511	511	511	0,099804305	0,10176125
1	62	**Belokan1**	**154**	**288**	62,00806399	62	411,821	**412**	**923**	0,002427184	0,15048544
42	44	Kopo	196	332	60,82697428	61	405,179	405	1328	0,103753086	0,10859259
23	25	**Belokan2**	**219**	**357**	33,97176475	34	163,348	**163**	**1491**	0,140981595	0,15349693
12	0	Cibaduyut	231	357	12	12	57,6522	58	1549	0,206896552	0
82	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

Figure 3. 10 Data Table Coefficient

With

dx : distance between two x points

dy : distance between two y points

x : pole point x

y : pole point y

kx : x multiplier coefficient

ky : y multiplier coefficient

xi : the length of array range between two points

fxi : array range between two points that has been rounded off

dxy : distance between to coordinates

fdxy : distance between to coordinates that has been rounded off

Activity diagram to display diagrammatical marker is as following:

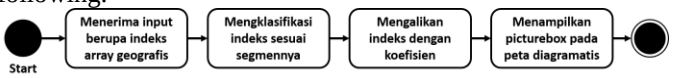

Figure 3. 11 Activity Diagram Diagrammatical Marker

Besides displaying all positions simultaneously, GUI can also display all the data received. The activity diagram is as follows:

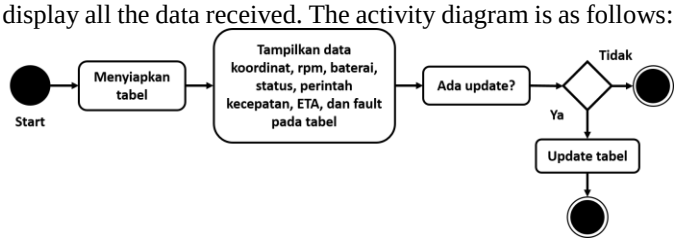

Figure 3. 12 Activity Diagram Table Update

5. Sending Command Speed

As explained, one of the GUI specifications is to send feedback in the form of speed instructions to fleet hardware. The activity diagram is as follows:

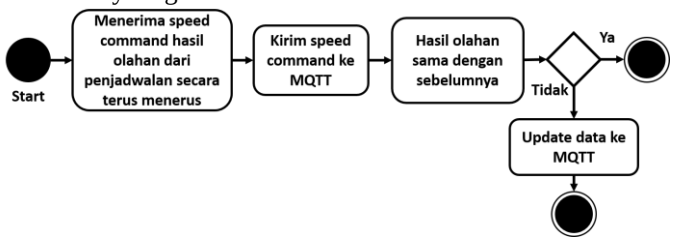

Figure 3. 13 Diagram Activity Speed Command

D. GUI Display (UX Design)

GUI display is made with triadic color scheme. The **triadic color scheme** brings a high contrast color scheme but keep maintaining the same tone[5]. The creation of triadic scheme is made by selecting three colors with lines with the same radius in the color wheel. GUI color scheme is selected with 65% radius of the color wheel in the purpose to bring a **fresh, vibrant, and not boring impression.**

The default screen on GUI is placed on the geographical map screen because displaying the fleet position is the main purpose of GUI making. On the left bar, information personal fleet data is provided. Mousehover on each marker will bring up the tooltip of the marker number, and the mouseclick on each marker will change the information that displayed on the left sidebar, depending on the marker that been clicked. I.e. the operator clicks on the marker fleet 4, the left sidebar will display information about fleet 4.

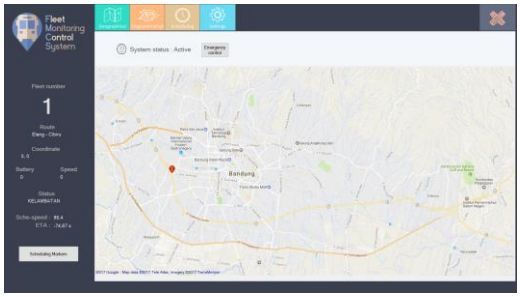

Figure 3. 14 Geographical Map Screen

The scheduling marker button at the bottom of the sidebar can make the geographical maps show all markers of the scheduling range that the fleet must follow. Therefore, this scheduling marker is placed as a toggle button so that the scheduling marker not ruins up the geographic map view.

The selection of contrasting colors on diagrammatic map arrangement aims to make the operator to clearly see the difference between one route and another. The selection of color schemes for this metro map has been widely used in the general metro map.

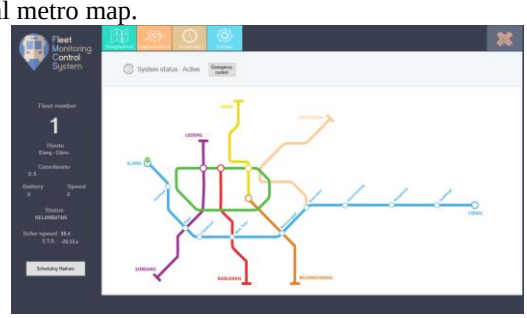

Figure 3. 15 Diagrammatical Map Screen

Meanwhile on the scheduling page, in the left column, operator can change the input speed of the scheduling, the distance between buses, the stop duration at the shelter. Basically, these numbers are already determined by the system, but the operator can change them if needed. While in the right column, there is an overall monitoring table. In this table the operator can observer all data received from the server.

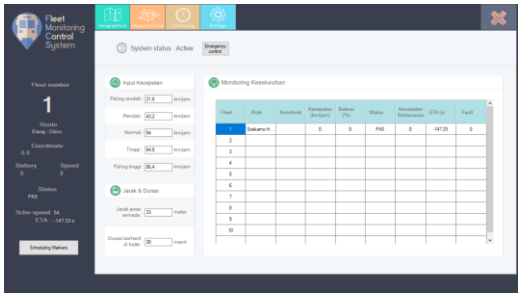

Figure 3. 16 Scheduling Screen

In the main panel (beside system status), there is an emergency control button. If emergency state happened, this button will be colored red. Operator have to confirmed the sustainability of fleet scheduling with this button.

IV. TESTING

A. *Testing of GUI Functional Capabilities*

In this testing, will be tested the capabilities of functional specs of GUI. Requirements that must be met is the internet connection is good enough and PC that used is not running another program except the GUI.

First stage testing is done by sending 10 complete data of fleet consists coordinates, rpm, fault, and percentage of battery from dummy data. Fleet status, speed command, and ETA are processed results from scheduling algorhimgs that have been integrated with GUI. The results are as follows:

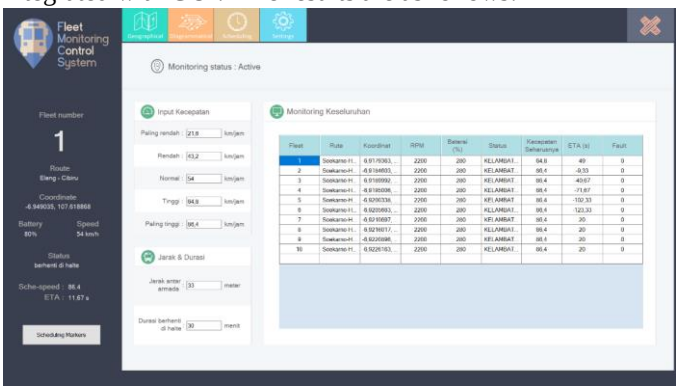

Figure 4. 1 Full Data Testing

From observation during the data processing, GUI can display all data well without causing the application lag/error.

Second stage testing is done by sending 40 fleet position data from dummy data and display it on geographical map.

The test result is as following:

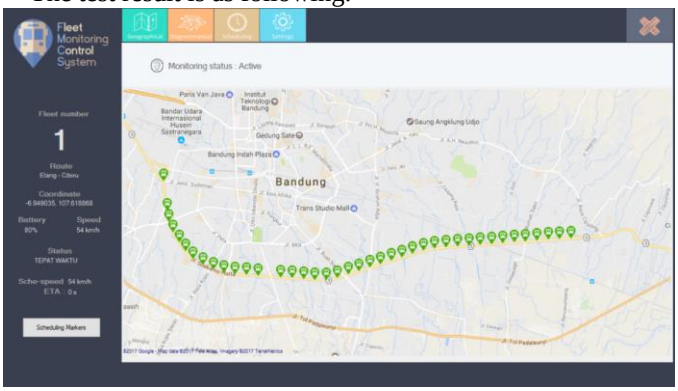

Figure 4. 2 40 Fleet Position Testing

From observation during the data processing, GUI can display 40 data fleet position simultaneously well without causing the application lag/error

B. *UX Design and UI Design Testing*

In this testing, specification area that will be tested is GUI feature, which is whether the application having a good UX design that user-friendly for user. Testing subjects will be given a short explain about GUI and then asked to operate GUI and fill a questionnaire.

The requirement that need in this testing is that testing subjects haven't known the system from GUI before, and assumed that subjects understand simple English language, simple operating language, and can operate PC.

First stage testing is UX Design testing that done to 12 subjects. Meanwhile the second testing done virtually to 11 subjects by online questionnaire. This data testing result will be

obtained from questionnaire that filled by subjects after operate GUI. The data that will be observed is:

- The delivery of an information/content in UX Design
- The effectiveness in GUI (UI Design)
- User's comfortability in using GUI (UX Design)
- The aesthetic from User Interface

From questionnaire result in the first stage testing, for UI assessment, 91.7% subjects are comfort with GUI design, 91.7% are easy to understand GUI, and 100% agree that GUI is effective enough. While questionnaire result in the second stage testing, obtained 90.9% subjects are comfort with GUI design, 90.9% are easy to understand GUI, and 100% agree that GUI is effective enough.

From all data that obtained from UX and UI Design with average score above 90%, we can conclude that GUI have been user-friendly enough for user.

C. *Field Testing*

This testing held in Soekarno-Hatta route with segmentation start from Cijagra road until Buah Batu intersection. Test done by bring the fleet hardware that has been switched on using motorcycle and sending GPS coordinate data.

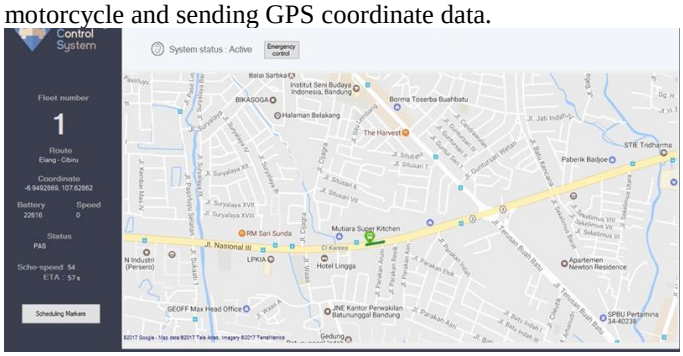

Figure 4. 3 Field Testing

From the picture above, can be concluded that GUI can show fleet1 marker with its scheduling marker. Beside that, personal fleet data is also can be seen on the sidebar.

In this testing, we also recorded two GPS coordinates route. The first is a route that plotted from GPS coordinates that sent by one of the member's smartphone and the second is the route that GUI shows. The result testing is as following:

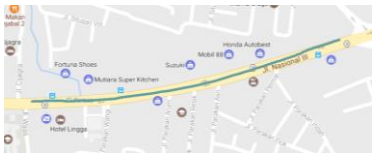

Figure 4. 4 Route Coordinate Smartphone GPS

Figure 4. 5 Route Coordinate Displayed in GUI

From the comparison of the two routes above, it can be concluded that the GUI can handle the deviations sent by GPS. Although there are deviations on the geographic map, the location displayed on the GUI remains on the centerline of the route it should be.

D. *Command Fleet Testing*

Testing is conducted simultaneously with field testing. Data sent fleet hardware in the form of position data will be processed by GUI scheduling algorithm. The GUI then sends the speed instructions for each change in the instruction speed. The test results fleet hardware is able to receive instructions speed from the GUI.

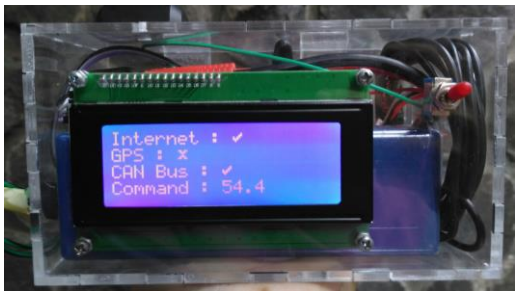

Figure 4. 6 Fleet Command Testing

V. CONCLUSION AND SUGGESTION

From the test results that have been done, FMCS GUI is able to display 40 fleet positions simultaneously and 10 complete data fleet in the form of coordinates, rpm, fault, and battery percentage simultaneously and without causing lag / error on the PC. In addition, FMCS GUI also has UX and UI Design that already meet user needs so it can be concluded GUI FMCS has a view that is quite user friendly.

Suggestions for development, segmentation in GUI geographical map, accuracy is not necessary until it reaches 12 meters. Taking into account the bus length of 12 meters and the bus is moving, the accuracy can be reduced to 15 meters (12 meters bus length + 6 meters and half bus length). Array cultivated to be minimized so that the calculation process faster and the system getting lighter.

REFERENCES

[1] Wikipedia, 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