

Simulation of Routing Option by Using Two Layers Fuzzy Logic and Dijkstra's Algorithm in MATLAB 7.0

Joni Ganda

Study Program Electrical Engineering, Faculty of Engineering, President University
Jl. Ki Hajar Dewantara, Cikarang Baru, Bekasi-17550, Indonesia
Corresponding author: ganda.joni.11@gmail.com

ABSTRACT

This project is aimed to create simulation software which can give an optimal route based on the condition given by the user. Travelling salesman problem, fuzzy logic and Dijkstra's algorithm will be used as benchmark of the software. The program will be developed in MATLAB 7.0. Fuzzy logic will process the current condition of the map become input for Dijkstra's algorithm. The result of the program will be an optimal route from three points of view; distance, density, and confusion. In order to verify the result of the program, it will be compared with the result of people's opinions in the same problem.

Keywords: Routing, fuzzy logic, Dijkstra's algorithm, Matlab

I. INTRODUCTION

In the year 2010, the car sales in Indonesia is projected to be increased around 15% or approximately 550 thousand units to 600 thousand units compared to the sales in 2009. Based on that, car will become one of main transportation vehicle in Indonesia. Therefore, some additional equipments for car become a concern for car user in Indonesia. One of those equipments is Global Positioning System (GPS).

The basic concept of GPS can be divided into two parts. There are GPS receiver and GPS satellite. A GPS receiver calculates its position through the signal sent by the GPS satellites. There are 24 to 32 satellites around the Earth that send the signal to GPS receiver. The GPS receiver utilizes the messages it receives to determine the transit time of each message and computes the distances to each satellite. Those distances are used to determine the position of the receiver.

In the present time, GPS usually consists of several basic features, such as preloaded maps, automatic rerouting for when you miss a turn, and spoken directions. Besides that, there are several extra features to make a GPS become very useful for the user. There are some of the top extra features that recently offered to the user:

- Spoken Street Names
- Real Time Traffic Reports
- Routing Option
- Detour Feature
- Predictive Data Entry, etc.

Routing Option is one of the top extra features in GPS which set a route for the user from starting point to the destination point. Besides that, it can set the route for

multiple points on one trip before the user start driving. Usually, the route is given into some categories, such as the easiest route, the fastest route, and the economical route. In this feature, GPS consider some aspects in order to give the route option to the user. There are the distance aspect, the economical aspect (the possibility to use highway route or not), and the difficulty aspect (the possibility to use small route or not).

This project is intended to develop a simulation about the routing option feature in GPS. The simulation will provide the information about the route option especially the distance, the density of the route, and the confusion level of the route. It also can set a route for multiple points for the user. Besides that, the user can avoid some points in the route.

The scope of this project will focus in how the program produce the best option route based on the condition given. Therefore, the program will be tested by using an artificial map. The artificial map will be designed similar to the real one in the reality. There are several nodes and icon all over on the map. From one node to another node, it contains information about distance, density, and confusion.

In order to realize this project, MATLAB 7.0 is used as the software. MATLAB 7.0 provides a lot of features that can make the program working well such as, GUI and m-file. By using GUI, the program has some user interface features. M-file will be used as the place where the program is edited and processed.

II. LITERATURE STUDY

After search and analyze every possible theory, algorithm, or method. Thus, there are three main

foundations that are used to support and construct the program in the project:

1. Travelling Salesman Problem (TSP)
2. Dijkstra Algorithm
3. Fuzzy Logic
- 4.

1) Travelling Salesman Problem (TSP)

Nowadays, TSP is a problem in combinatorial optimization studied in operation research and theoretical computer science. The general problem that is usually studied is given a list of cities and their pairwise distance, the task is to find the shortest possible tour that visits each city exactly once.

Starting 1930s, TSP has become one of the most intensively studied problems in optimization. It has become a point of reference for many optimization methods.

a. TSP as a graph problem

The TSP can be modeled as a graph. The city can be defined as the graph's node; the connection between two cities can be defined as the graph's edge.

b. TSP as asymmetric and as symmetric

In the case of symmetric TSP, the length of connection between two cities is same in every direction. It will form an "undirected graph" which is a graph without orientation. In this case, the numbers of possible solution become half than in a normal TSP problem. On the other hand, in the case of asymmetric TSP, paths may not exist in both directions or the distances might be different. It forms a directed graph which is a graph with orientation to solve it.

c. TSP with metric distance

The intercity distances in this type TSP satisfy the triangle inequality. If we assume there are three points connected and formed a triangle, the direct distance from node A to node B is never longer than the detour via node C which is shown by Eq. (2-1)

$$c_{ij} \leq c_{ik} + c_{kj} \quad (2-1)$$

d. TSP with non-metric distances

This type of TSP does not satisfy the triangle inequality. In many applications, TSP allows cities to be visited twice. Therefore, a symmetric, non-metric instance can be reduced to a metric one. This replaces the original graph with a complete graph in which the inter-city distance c_{ij} is replaced by the shortest path between i and j in the original graph [TSP].

2) Dijkstra's Algorithm

In Graph theory, there are several different algorithms that find a shortest path between two nodes in a weighted map. One of those algorithms is called as Dijkstra's algorithm. Dijkstra's algorithm is discovered by the Dutch mathematician Edsger Dijkstra in 1959 [DA].

Dijkstra's algorithm is a graph search algorithm that solves the single-source shortest path problem for a graph with nonnegative edge path costs, producing a shortest path tree. Therefore, Dijkstra's algorithm becomes one of the most common algorithms that are used to solve TSP. The actual speed of Dijkstra's algorithm can vary from $O(n \cdot \log(\log(n)))$ to $O(n^2)$ [OSSP].

The basic concept of how the algorithm works can be shown from the figure below:

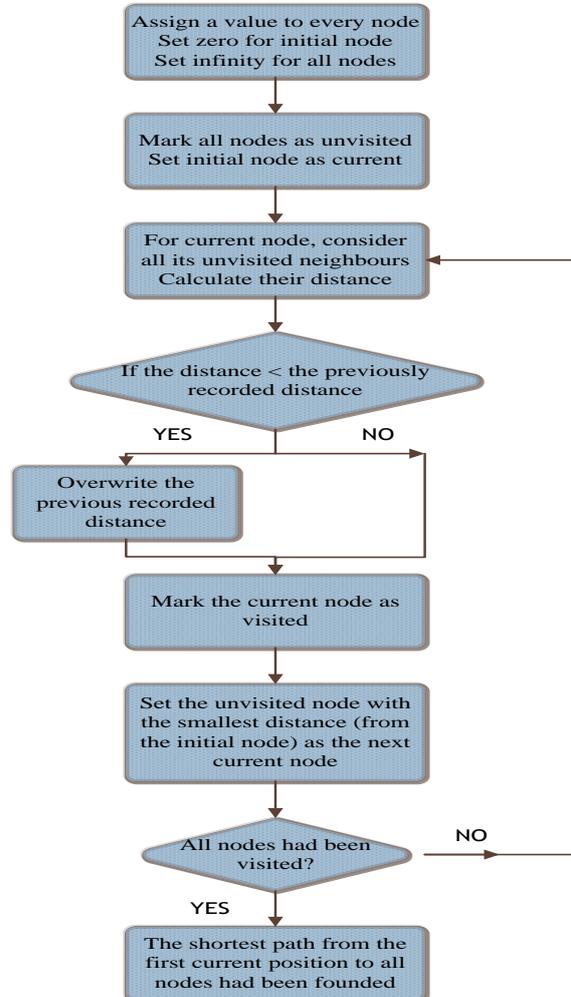


Fig. 1. The basic concept of Dijkstra's algorithm

3) Fuzzy Logic

a. The Concept of Fuzzy Logic

Note that the meaning of fuzzy is "not clearly thought out", "blurry", or "vague", the fuzzy theory can be defined as a mathematical theory that has fuzziness as one of the aspect. In this case, fuzziness is the ambiguity that can be found in the definition of a concept or the meaning of a word. For example, the uncertainty in expressions like "delicious", "high temperature", or "long distance" can be called fuzziness.

Based on that, how the fuzzy logic approach a problem is very simple. It uses some IF-THEN rules. For example, if the temperature is high then the speed of the fan is increasing. Therefore, fuzzy logic maps the input of the problem into the output by using IF-THEN rule. The

mapping process is done in the system that is called Fuzzy Interface System (FIS). FIS will process and evaluate every rule in order to get the output.

i. Fuzzy Set

Fuzzy set is a set which the membership of every element does not have a clear boundary. This type of set is very contrast with the classic type. For example, if the classic type knows only “white” and “black” then the fuzzy type knows “grey” between “white” and “black”. The mathematical model of the fuzzy set can be defined as in Eq. (2-2):

$$\mu(x) \in [0 \ 1] \quad (2-2)$$

It means that the value of $\mu(x)$ can vary from 0 to 1 depends on the value of x .

ii. Fuzzy Membership Function

Membership function is a function that maps the input into a degree of membership. There are several points that to be considered when the membership function is built:

- Fuzzy set intend to keep the concept of fuzziness
- Fuzzy set allow a membership from one function to be partial membership of another function as long as the value of the membership is below 1
- A degree of membership in fuzzy set is ranged from 0 to 1
- Every membership function μ associates with other functions and maps an input into corresponded degree of memberships.

iii. Fuzzy Logic Operator

The logic operator like AND, OR, and NOT in Boolean logic will have their correspondent operators in fuzzy logic. The difference of those logic operators for both of them is located in the degree of membership that is owned in fuzzy logic.

Min-Max Method

for $(\mu_A(x), \mu_B(y))$:

$$\bullet \text{ NOT } \mu_A(x) = 1 - \mu_A(x) \quad (2-3)$$

$$\bullet \mu_A(x) \text{ AND } \mu_B(y) = \min(\mu_A(x), \mu_B(y)) \quad (2-4)$$

$$\bullet \mu_A(x) \text{ OR } \mu_B(y) = \max(\mu_A(x), \mu_B(y)) \quad (2-5)$$

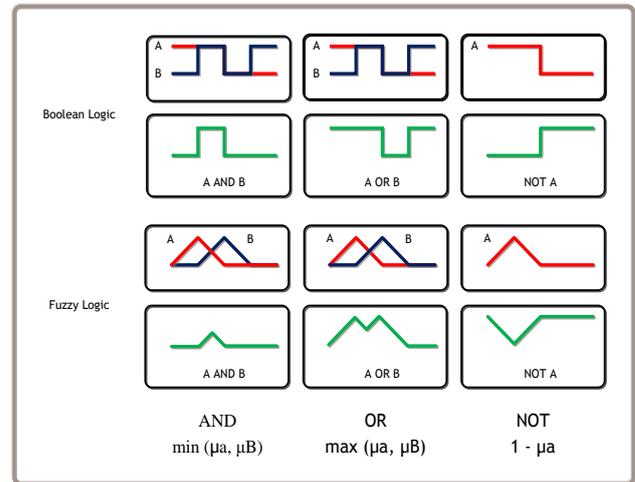


Fig. 2. The graphic of logic operation in Boolean logic and fuzzy logic using Min-Max method [9]

iv. IF-THEN Rule

Fuzzy logic works based on rules that are set in the form of IF-THEN statement. In order to interpret a IF-THEN statement, there are three parts that must be done:

1. Fuzzification – first of all, the input had to be defined the degree of membership.
2. Fuzzy Logic Operation – after the fuzzification process, the selected fuzzy logic operator will process the operation.
3. Implication – finally, the result of fuzzy logic operation will be applied in order to get the conclusion for the output.

b. Fuzzy Interface System

Based on the concept of fuzzy logic, the problem will be processed by a system that will map input into output according to the given IF-THEN rules. This fuzzy system is known as Fuzzy Interface System (FIS).

i. Type of FIS

Up to now, there are two types of FIS. They are Mamdani type and Sugeno type. Both methods have their own ways to determine the output of FIS.

Mamdani type is the common used type in FIS. In Mamdani method, an output of an IF-THEN rule is calculated by measure the area below the output curve of the fuzzy set. After that, Mamdani method calculate the central gravity of the area that consist of every output from fuzzy set based on all applied rule. The result will be the output of FIS.

Sugeno type is also known as Takagi-Sugeno-Kang type. The main difference between Sugeno and Mamdani is the output membership function. Sugeno type uses linear or constant output membership function. Thus the output of this FIS will be very simple. It uses Eq. (2-22):

$$\text{Output} = \frac{\sum_{i=1}^N w_i z_i}{\sum_{i=1}^N w_i} \quad (2-22)$$

where W_i is the result of fuzzy logic operation and Z_i is the output of rule number i . Thus, Sugeno type FIS normally uses to model the non-linear system [NA09].

ii. The Process of FIS

The process of FIS can be divided into five parts:

1. Fuzzification input

In this part, FIS take the input and determine the degree of membership in all fuzzy sets using every membership function.

2. Fuzzy logic operation

Fuzzy logic operation processes the fuzzification result as the input become a single output. The output will be a truth value from the fuzzy logic operation such as AND operation or OR operation. This output can be called as “the degree of antecedent truth”.

3. Implication

By using the degree of antecedent truth, the output of an IF-THEN rule can be got. Usually the user can set the weight of every IF-THEN rule. The weight varies from 0 to 1.

4. Accumulation

Every output of IF-THEN rule will be combined into a single fuzzy set. This process normally use OR operation to combine all outputs of IF-THEN rule.

5. Defuzzification

In this part, the input is a fuzzy set which is resulted from the accumulation process and the output is a single value.

III. PROGRAM DEVELOPMENT

This program is composed of fuzzy logic operation, software development, and interface design. The overall process of the program can be shown in the figure below:

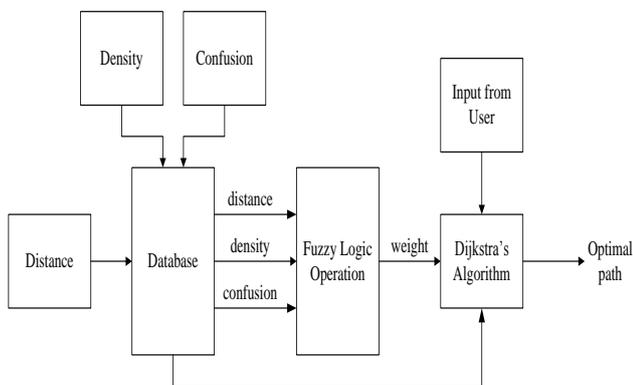


Fig. 3. The overall process of the program

1. Fuzzy Logic Operation

This is the first step in program development. Note that Dijkstra's algorithm can only process a value from one point of view between two nodes; it becomes a handicap of

Dijkstra's algorithm. Because in this case, the value between two nodes will be composed of three points of view, the fuzzy logic operation is needed. The fuzzy logic operation processes the three values from every point of view become one single value.

The “density” point of view covers a number of vehicles in the path between two nodes, the width of the path, and the building around the path. If a path between node A and B has a department store and a school then it will have a higher “density” value than a path between node C and D that has only a school. In this problem, the width of the path and a number of vehicles in the path are ignored. The building around the path is the main factor that affects the value of “density”.

The “confusion” point of view means the difficulty of the path between two nodes. It is composed of how many turns, how many junctions, and the type of the path (one way or two ways). Note that, people tend to choose the simple path rather than the complicated path, the path with no turn will have a smaller “confusion” value than the path with a lot of turns. Besides that, in a path with a lot of turns, the velocity of the vehicle will be slowed down every time it wants to take turn. Therefore, “density” and “confusion” can be combined to replace “time” in this case.

The “distance” point of view in fuzzy logic operation will be represented in the value from 0 to 1. Before the value of “distance” from the database is processed by fuzzy logic operation, there is a variable “edge” which will find the maximum value of “distance” in the selected database. The value from “edge” will become the divider. Thus, the value of “distance” in fuzzy logic operation can be got by dividing the value of “distance” from the selected data with the value of “edge”.

From the three points of view, fuzzy logic operation combines them into one single value. Based on the research of trial and error, the best combination is using 2 layers fuzzy logic operation. First, the fuzzy logic operation of “distance” and “density” will result a value called “weight1”. On the other hand, the fuzzy logic operation of “distance” and “confusion” will result a value called “weight2”. At the end, “weight1” and “weight2” will become input for the second layer fuzzy logic to get the main output that is “totalweight” from those three points of view.

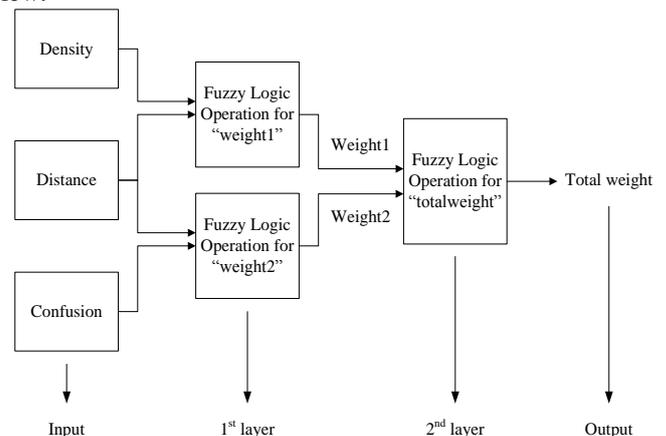


Fig. 4. The general process of fuzzy logic operation in this program

The operation of fuzzy logic will be done by MATLAB 7.0. To make the fuzzy operation in MATLAB 7.0, it uses FIS Editor. Therefore, the FIS editor of fuzzy logic operation for weight 1 is named “tsp_weight1”. For FIS editor of fuzzy logic operation for weight 2 is named “tsp_weight2”. FIS editor for the second layer of fuzzy logic operation is named “tsp_totalweight”.

a. Fuzzy Interface System (FIS)

The characteristics for those three FIS in general are same, such as:

- FIS type = Mamdani
- AND method = min
- OR method = max
- Implication = min
- Aggregation = max
- Defuzzification = centroid

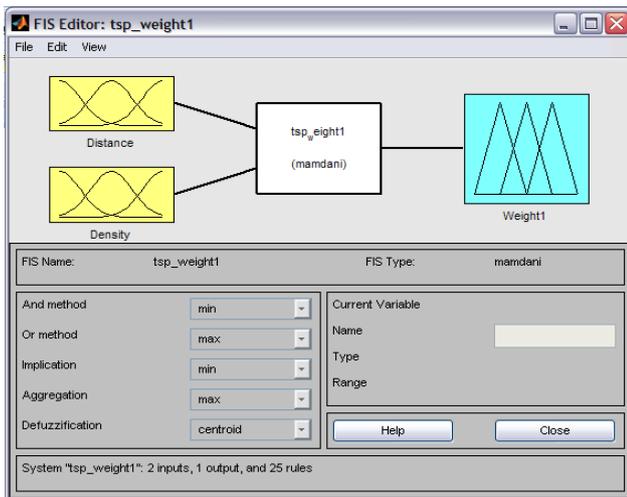


Fig. 5. FIS editor of “tsp_weight1”

b. Membership Function

Based on the condition of the fuzzy logic operation in this program, every fuzzy logic operation will have two inputs and one output. There are three main inputs that had been described early. There are “distance”, “density”, and “confusion”. The characteristic of “distance” will same for two fuzzy logic operations. Besides that, the results from first layer fuzzy logic operations (“weight1” and “weight2”) will be converted into a different membership function in the second layer fuzzy logic operation. Both “weight1” and “weight2” in first layer fuzzy logic operations will have 7 membership functions. It will be converted into 5 membership functions when they act as input in the second layer fuzzy logic operation. The detail of every variable will be shown by the following table:

Table 1. The detail of every variable in fuzzy logic operation.

Fuzzy logic operation	Name of variable	Type of variable	Range of variable	Number of membership functions
Fuzzy logic operation of “tsp_weight 1”	Distance	Input	0 to 1	5
	Density	Input	0% to 100 %	5
	Weight1	Output	0 to 1	7
Fuzzy logic operation of “tsp_weight 2”	Distance	Input	0 to 1	5
	Confusion	Input	0% to 100 %	3
	Weight2	Output	0 to 1	7
Fuzzy logic operation of “tsp_totalweight”	Weight1	Input	0 to 1	5
	Weight2	Input	0 to 1	5
	Totalweight	Output	0 to 1	11

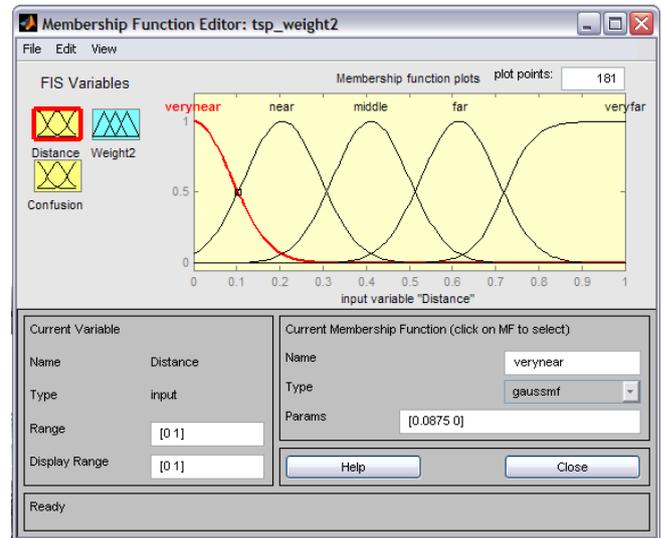


Fig. 6. Membership Function Editor: tsp_weight2 – “distance”

c. Rule

For this case, every rule in three fuzzy logic operations has same format:

“**IF** (input 1) **AND** (input 2) **THEN** (output)”

In order to get the main output, this 2-layer fuzzy logic operation uses 65 IF-THEN rules. Fuzzy logic operation of weight1 uses 25 IF-THEN rules. Fuzzy logic operation of weight2 uses 15 IF-THEN rules. And at last, the second

layer fuzzy logic operation for total weight uses 25 IF-THEN rules.

2. Software Development

In MATLAB 7.0, a complex program will be built in the form of many functions. Some function that is often called in the main function will have its own page of m-file. It is because MATLAB 7.0 only can call a function once in one page of m-file and it cannot be called by another function in the same page of m-file. Thus, this type of function is usually named as user defined function which has same characteristic with built-in function in MATLAB 7.0. Thus, it can be called many times in different functions in same page of m-file.

The software for this case will have four pages of m-files. Every page contains different function. There are main function, function dijkspath, function dijkstra, and function printpath. The figure below shows the relation of every function.

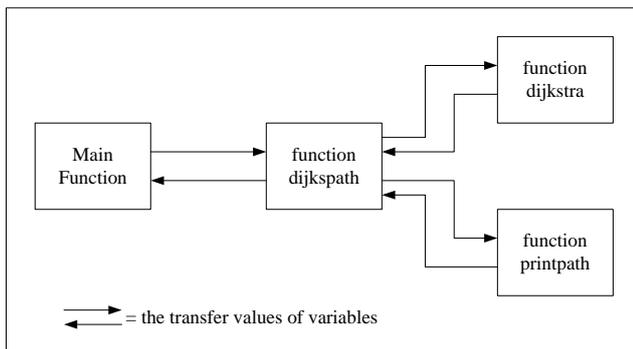


Fig. 7. The relation of four functions in the software

a. Link of Fuzzy Logic Operation with the Main Function

To combine the fuzzy logic operation into the main function, the built-in function from MATLAB 7.0 will be used. The built-in function that will be used is “readfis” function and “evalfis” function. “readfis” function is used to load a FIS from disk. “evalfis” function will perform fuzzy inference calculations in m-file.

b. Dijkstra’s Algorithm in MATLAB 7.0

In this software, Dijkstra’s algorithm will be called more than one time. Therefore the software about Dijkstra’s algorithm is built in one independent m-file page as the user defined function. Because Dijkstra’s algorithm is built as the user defined function which is named as “dijkstra” function, it will have inputs and outputs. The starting node, the number of total node in the map, and the weights between two nodes are the inputs for this function. The total weight from starting node to other nodes, the recorded previous path, and the data of node that had been visited are outputs for this function.

c. Additional Features in the Software

To make this program different and attractive, there are some additional features which had been added in the software, such as:

1. Open Map
2. Detail Info of the Path
3. Preference Node and Banned Node
4. Visit Some Nodes and Back to the Starting Node

3. Interface Design

This program has a user interface so that the user can communicate with the software. By using the components of GUI in MATLAB 7.0, the interface of this software is made. Starting from enter the input until the change of input if there is an error when the user enter the input.

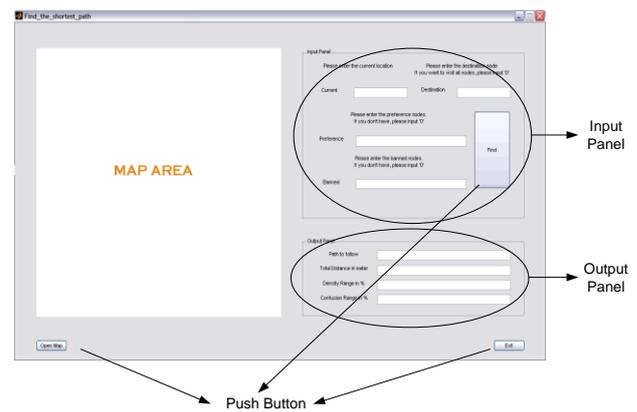


Fig. 8. The interface design of the program

IV. RESULT AND DISCUSSION

1. Artificial Map

The map will be built similar with the real city. There are several special places in the artificial map same with the real ones, such as traditional market, school, hospital, etc. These special places will be shown in the artificial map as an icon. Each special place has a standard length and width regardless the size of the icon in the map. This artificial map will be built in the length 8800 meters and width 7800 meters.

This artificial map will be named Berk City. The map of Berk City can be seen in the figure below:

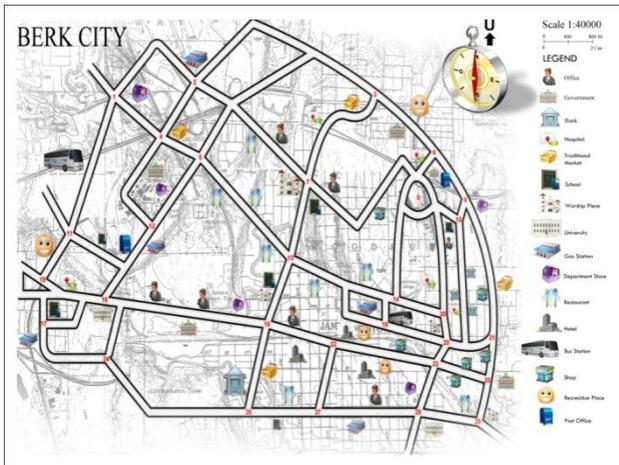


Fig. 9. The map of Berk City

Berk City has sixteen types of special places and six types of roads in this case. In normal condition, each road in Berk City will has 10% of density and 10% of confusion.

2. Cases

Based on what had been stated in the briefing this chapter, there are five cases to be asked to the people. The number of people’s answer that had been collected is 55 answers.

The objective of this questioner is checking the result of the program with the answer from the people. The result of the program can be said correct if the result become the first majority or second majority in the collected routes based on the variation of the route.

There are five cases that will be asked to the people:

- **Case 1:**

The fastest and easiest route from node 1 to node 29

- **Case 2:**

The fastest and easiest route from node 3 to node 24

- **Case 3:**

The fastest and easiest route from node 11 to node 10 through node 18

- **Case 4:**

The fastest and easiest route from node 15 to node 6 without through node 3, node 7, and node 8

- **Case 5:**

The fastest and easiest route from node 1 to node 29 through node 8, node 16, and node 23

In this journal, it will show the result for case 1 and case 5.

a. Case 1

The fastest and easiest route from node 1 to node 29.

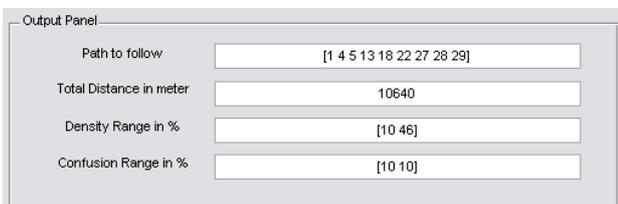


Fig. 10. The fastest and easiest route from node 1 to node 29

The two top majority from the questioner answer.

Table 2. The result of case 1 based on the variation of route options.

No	Type of Route	Quantity	Percentage
1	1-11-16-24-26-27-28-29	11	20%
2	1-4-5-13-18-22-27-28-29	7	13%

b. Case 5

The fastest and easiest route from node 1 to node 29 through node 8, node 16, and node 23.

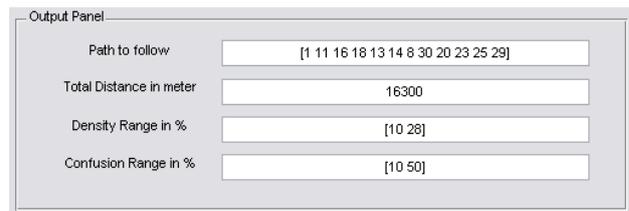


Fig. 11. The fastest and easiest route from node 1 to node 29 through node 8, node 16, and node 23

The two top majority from the questioner answer.

Table 3. The result of case 5 based on the variation of route options

No	Type of Route	Quantity	Percentage
1	1-11-16-18-13-14-8-30-20-23-25-29	12	22%
2	1-11-16-18-13-7-8-30-20-23-28-29	7	13%

V. CONCLUSION AND FUTURE WORK

1. Conclusion

Routing option is one of top features in GPS. There are a lot of aspects that can be improved in this feature. This project is one of examples of the improvement for routing option by adding density factor and confusion factor in order to determine the route.

This project is a simple simulation about routing option. It combines fuzzy logic operation and Dijkstra’s algorithm in MATLAB 7.0 as the programming language. Fuzzy logic processes the three inputs, “distance”, “density”, and “confusion” become one single value that is “totalweight”. By using two layers fuzzy logic, the operation will have three fuzzy logic operations. It is used to control the weighting between “distance” – “density” combination and “distance” – “confusion” combination. Dijkstra’s algorithm will compute the database which has been inputted from

output of fuzzy logic operation in order to get the ideal route for the chosen node.

Based on the questioner, the result from the program is good because it becomes the first majority or the second majority in several different cases. The feature of preference nodes and banned nodes make the program become more complete.

There are some limitations in this project, such as:

- The program cannot read the map and create the database automatically. Thus the database of every map must be inserted first if the program wants to be run.
- Due to the limitation of time, the program will not be equipped with time identifier. By using the time identifier, the density from one node to another node can be varying through time. In example, from node A to node B, if the user passes it at 11 am, it will get heavier density than the user passes it at 11 pm.

2. Future Work

There are several aspects that can be used as the improvement of this project, such as:

- The database of the map's detail has to be inputted first.
- The improvement that can be done is having a feature that can read a map and input the data of the map directly to the database without the user involvement.
- The output of this program just presents a detail data of the ideal route. It can be improved by adding a visualization of the ideal route in the map.
- The possibility to use another algorithm and programming language in order to improve the quality of the feature.

REFERENCES

- [TSP] *Travelling salesman problem*, http://en.wikipedia.org/wiki/Travelling_salesman_problem
- [DA] *Dijkstra's algorithm*, http://en.wikipedia.org/wiki/Dijkstra's_algorithm.
- [OSSP] *One Source Shortest Path: Dijkstra's Algorithm*, <http://compprog.wordpress.com/2007/12/01/one-source-shortest-path-dijkstras-algorithm/>
- [NA09] Naba, Agus, *Belajar Cepat Fuzzy Logic Menggunakan MATLAB*, ANDI Yogyakarta, 2009.

Received: 25 January 2016

Revised: 18 March 2016

Accepted: 16 May 2016