

DESIGN OF OPTIMAL DISTRIBUTION APPLICATION USING FIREFLY ALGORITHM

Ngarap Imanuel Manik¹, Yungghih Nursalim², and Derwin Suhartono³

^{1,2}Mathematics Department, School of Computer Science, Bina Nusantara University

³Computer Science Department, School of Computer Science, Bina Nusantara University
Jln. Kebon Jeruk Raya No.27, Jakarta Barat 11530, Indonesia

¹manik@binus.ac.id; ²yungghih.nursalim@binus.ac.id; ³dsuhartono@binus.edu.

Received: 3rd May 2017/ Revised: 12th August 2017/ Accepted: 14th August 2017

Abstract - The goal of this research was to optimize the distribution of goods and computerize the system. The method consisted of problem identification, analysis, implementation, and evaluation. Firefly algorithm was used as a method for optimizing the distribution of goods. The results are the shortest distribution route of goods in accordance with the existing constraints and low cost of distribution. It can be concluded that the research can be used to optimize the distribution of goods and to minimize distribution cost.

Keywords: firefly algorithm, distribution of goods

I. INTRODUCTION

In this modern age, the economic activities cannot be separated from the distribution of goods. Distribution of goods can be supported by various modes of transportation such as land transportation, water transportation, and air transportation. If the distribution of goods is not done properly, it can increase the cost of goods distribution and also the prices of goods distributed (Onwubiko, 2000; Yang, 2010)

To optimize the distribution of goods, the different methods such as particle swarm optimization, genetic algorithm, or firefly algorithm can be used. The methods used to solve problems optimization of the distribution of goods in this research is firefly. Firefly Algorithm is used to develop discrete versions by updating firefly position with the name of discrete firefly algorithm with edge-based movement (Fister *et al.*, 2013; Jati *et al.*, 2013; Johari *et al.*, 2013; Sayadi *et al.*, 2013).

Discrete firefly algorithm with edge-based movement is used to cope with the distribution of goods by determining the shortest route. The movement in this algorithm also produces better movement than the random movement. The algorithm is in the form of a desktop application to solve the problems of optimization of the goods distribution to the stores. It is expected to distribute customer good easier and produce a more optimal distribution costs. (Pan *et al.*, 2013; Pornsing, 2014; Yesodha, 2015).

II. METHODS

The methodology of this research consists of several steps as follows. First, this step is the identification of the main issues that will be discussed on the distribution cost optimization like how to minimize the cost of goods distribution. Identifying problems is done by the distribution

in Christian Store with interviews and brainstorming. From the available problems, researchers determine the boundary problem that will be the scope of the research. Second, it is literature review. The researchers collect some of there ferences that can help in the research. The literature study is conducted on the books, journals, the Internet and other literature. From literature, it is expected to find the theoretical basis for data processing, model-making, selection of methods and making the program so that it can solve problems of goods distribution in Christian Store. The method is discrete firefly algorithm with edge-based movement and program with a desktop-based application with the Python programming language. Third, by creating a cost from the goods distribution that is identified and based on the literature study in the form of a mathematical model, it can represent the real system. A mathematical model is adjusted with problem in the research. The distribution cost model is as follows.

$$\text{Total Cost of Distribution} = \min (\sum(\text{fixed cost} + \text{variable cost})) \quad (1)$$

Fixed cost is the cost of vehicle maintenance and salary of driver and helper. Meanwhile, variable cost is the total cost of gasoline used.

Fourth, there is data collection. The researchers collect data in accordance with the scope of the problem processed to solve the problems of goods distribution. The relevant data in this research is the number, types, and volume of the vehicles used for distribution; the name and address of the depot and store; the costs distribution of operational expenses such as gasoline, vehicles maintenance, and salary of driver and helper; the distance between the depot and each store and the distance between each store; and other data that support the research. Fifth, it is the application of discrete firefly algorithm with edge-based movement in modeling cost distribution. Once the required data is obtained, the simulation of the calculations can be done. Formulation of a mathematical model created is implemented using discrete firefly algorithm with edge-based movement (Fister *et al.*, 2013). Sixth, the designing and making application for a desktop-based application to optimize the distribution of goods with discrete firefly algorithm with edge-based movement uses the Python programming language. The program is designed with prototyping development methods with the conditions. The program created stores the data with a file-based approach because the data is not many and fast regarding development. Seventh, the application program is tested whether the program is bug-free and in accordance with the objectives of the identified problems. Eighth, the evaluation program is to determine whether it produces optimal results according to the calculations made

as well as to see the suitability of the program to the needs of Christian Store, and the ease of operation. If the results are optimal, it will enter the implementation phase and prepare there ports. If it is no, it will be checked again on the implementation phase of the discrete firefly algorithm with edge-based movement to find optimal results. Ninth, after the research objective is achieved, the program will be implemented, and the process is complete. Then, it is followed by drafting the report. Next, it is the mathematical model. The objective function is in Equation 2.

$$Z = \min \left(\sum_{k=1}^m \sum_{j=0}^n \sum_{i=0}^n \frac{p_k}{r_k} c_{ij} x_{ijk} + \sum_{k=1}^m \sum_{j=1}^n f_k x_{0jk} \right) \quad (2)$$

The barrier function defines each customer node that visits only one time by one vehicle (Equation 3) and defines the number of vehicles entering and exiting the same depot (Equation 4).

$$\begin{aligned} \sum_{i=0}^n \sum_{k=1}^m x_{ijk} &= 1, \quad \forall j = 1, 2, \dots, n \\ \sum_{j=0}^n \sum_{k=1}^m x_{ijk} &= 1, \quad \forall i = 1, 2, \dots, n \end{aligned} \quad (3)$$

$$\sum_{i=0}^n x_{i0k} - \sum_{j=0}^n x_{0jk} = 0, \quad \forall k = 1, 2, \dots, m \quad (4)$$

If the customer visits the vehicle, the vehicle must come out. It uses Equation 5. Then, the number of vehicles can be calculated using Equation 6.

$$\sum_{i=0}^n x_{isk} - \sum_{j=0}^n x_{sjk} = 0, \quad \forall s = 1, 2, \dots, n; \forall k = 1, 2, \dots, m \quad (5)$$

$$\sum_{j=1}^n x_{0jk} \leq 1, \quad \forall k = 1, 2, \dots, m \quad (6)$$

To indicate the relationship between two variables decision, it uses Equation 7. Meanwhile, to show the goods load so it does not exceed the capacity of vehicle, it uses Equation 8.

$$\begin{aligned} \sum_{i=0}^n x_{ijk} &= y_{jk}, \quad \forall j = 0, 1, \dots, n; \forall k = 1, 2, \dots, m \\ \sum_{j=0}^n x_{ijk} &= y_{ik}, \quad \forall i = 0, 1, \dots, n; \forall k = 1, 2, \dots, m \end{aligned} \quad (7)$$

$$\begin{aligned} \sum_{j=1}^n d_j x_{0jk} &\leq Q_k, \quad \forall k = 1, 2, \dots, m \\ \sum_{i=1}^n d_i y_{ik} &\leq \sum_{k=1}^m Q_k, \quad \forall k = 1, 2, \dots, m \end{aligned} \quad (8)$$

The objective function of the model is to minimize the total cost of distribution. One of the ways is by describing the total number of distribution costs. The description of the equation is as follows.

i, j = index of customers
 $i = 1 \dots n$
 $j = 1 \dots n$
0 as depot
 k = index of the vehicles
 $k = 1 \dots m$
 c_{ij} = distance between customer i and j
 d_i = order from customer i
 Q_k = capacity of vehicle k
 p_k = the price of fuel from vehicle k
 r_k = the ratio of the fuel needs of vehicle k
 f_k = maintenance costs as well as driver and helper salaries of vehicle k

Then, the decision variables are

$$x_{ijk} = \begin{cases} 1, & \text{if the vehicle } k \text{ through route from } i \text{ to } j \\ 0, & \text{other} \end{cases}$$

$$y_{ik} = \begin{cases} 1, & \text{if customer } i \text{ visits by vehicle } k \\ 0, & \text{other} \end{cases}$$

Then, the researchers can idealize some of the flashing characteristics of fireflies to develop firefly-inspired algorithms. For simplicity in describing the new firefly algorithm, the researchers use three idealized rules. First, all fireflies are unisex, so one firefly will be attracted to other fireflies regardless of their sex. Second, attractiveness is related to the brightness. Thus, the less bright firefly will move towards the brighter one. The brightness decreases as its distance increases. If there is no brighter firefly than the particular firefly, it will move randomly. Third, the brightness of a firefly is affected or determined by the landscape of the objective function. For maximizing the problem, the brightness can be in line with the value of the objective function. The brightness can also be defined in a similar way to the fitness function in genetic algorithms.

Based on these three rules, the basic steps of firefly algorithm can be summarized as the pseudocode shown in the algorithm below. In a certain sense, there is some conceptual similarity between firefly algorithm and the Bacterial Foraging Algorithm (BFA). In BFA, the attraction among bacteria is based on their fitness and distance. Meanwhile, in firefly algorithm, the attractiveness is linked to their objective function and monotonic decay of the attractiveness with distance. However, the agents in firefly algorithm have adjustable visibility and more versatile in attractiveness variations, which usually leads to higher mobility. Thus, the search space can be explored more efficiently (Yang, 2009).

Firefly Algorithm

Objective function $f(x)$, $x = (x_1, \dots, x_d)^T$

Generate initial population of fireflies x_i ($i=1, 2, \dots, n$)

Light intensity I_i at x_i is determined by $f(x)$

Define light absorption coefficient γ

While ($t < \text{MaxGeneration}$)

For $i=1: n$ all n fireflies

For $j=1: i$ all n fireflies

If ($I_j > I_i$), Move firefly i towards j in d -dimension; end if

Attractiveness varies with distance r via $\exp[-\gamma r]$

Evaluate new solutions and update light intensity

end for j

end for i

Rank the fireflies and find the current best

end while

Postprocess results and visualization

III. RESULTS AND DISCUSSIONS

In designing this program, the researchers use four UML diagrams for defining the structure of the program. Those are use case diagram, activity diagram, class diagram, and sequence diagram. The use case diagram of the application to be developed is shown in Figure 1.

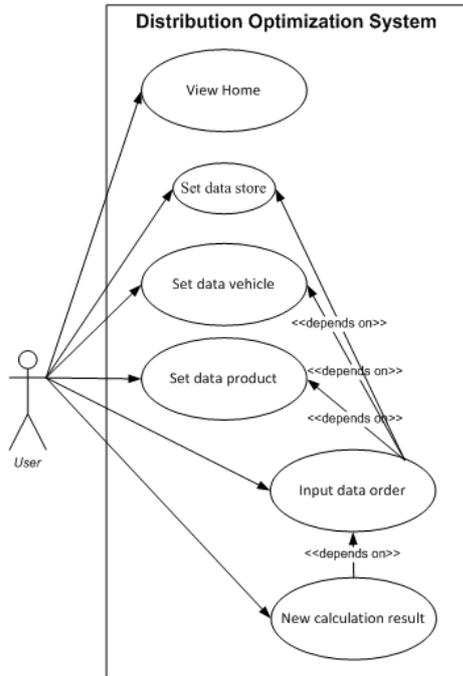


Figure 1 Use Case diagram

An activity diagram is another important diagram in UML to describe the dynamic aspects of the system. It is a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. Then, the control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

Activity diagram designed in this research consists of six activity diagram. Those are the home activity diagram, store activity diagram, vehicle activity diagram, product activity diagram, order activity diagram, and calculate activity diagram. Home activity diagram is the activity that occurs when user run the application and choose the Home menu. When the user presses the Home menu, the system displays the Home page on the screen. Meanwhile, store activity diagram is an activity when the user chooses the Store menu. When the user presses the Store menu, the system shows Store page on screen. On the Store page, the user needs to enter a name and the address of depot, name, and address of customer shops, and the distance data between the shop and shop depot before pressing the Save button to store data.

Similarly, vehicle activity diagram is when the user chooses the Vehicle menu. When the user presses the Vehicle menu, the system shows the Vehicle page on the screen. On the Vehicle page, the user needs to enter the name, volume, fuel ratio (Km/L), fuel price (Rp/L), maintenance cost (Rp) per one road of the vehicle, and salary of the driver and helper per one road (Rp) before pushing Save button to save data. Then, product activity diagram when the user selects

the Product menu. Pressing the Product menu, the users can see the Product page on the screen. On Product page, users need to enter the name of goods and volume of goods before pressing the Save button to save data. Meanwhile, order activity diagram is an activity when the user selects the Order menu. When the user presses the Order menu, the system will check whether there is stored data of vehicles and goods to display the Order page on the screen. On the Order page, the user needs to select the store and enters the order amount of each item. Then, the user can press OK and continue with the next store until all orders are entered. Activity calculation diagram is when the user chooses to press Calculate on the order menu. The system will display the calculation result on the screen.

Next, class diagram models the static structure of a system. It shows the relationships between classes, objects, attributes, and operations. A brief description of each class is shown in Figure 2.

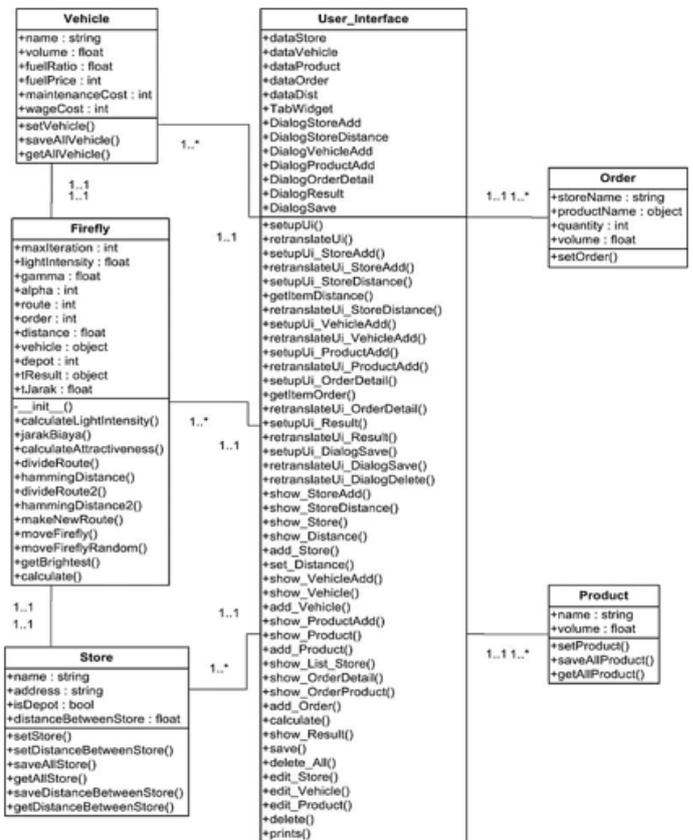


Figure 2 Class Diagram

Sequence diagrams describe the interactions among classes regarding an exchange of messages over time. It is also called as event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. It can help to predict how a system will function and to discover responsibilities a class may need to have in the process of modeling a new system. Like activity diagram, the sequence diagram designed consists of six sequence diagrams. Those are home sequence diagram, store sequence diagram, vehicle sequence diagram, product sequence diagram, order sequence diagram, and calculate sequence diagram. The example is shown in Figure 3.

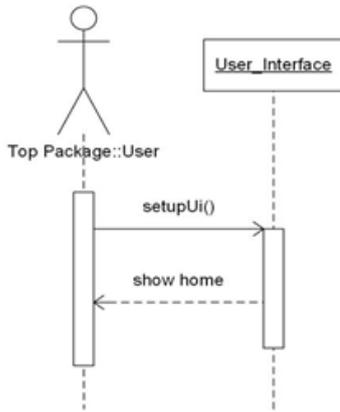


Figure 3 Sequence Diagram of SetupUi

Figure 3 shows the sequence that occurs when the user runs `setUpUi` through `User_Interface`. Then, `User_Interface` displays the home to the user.

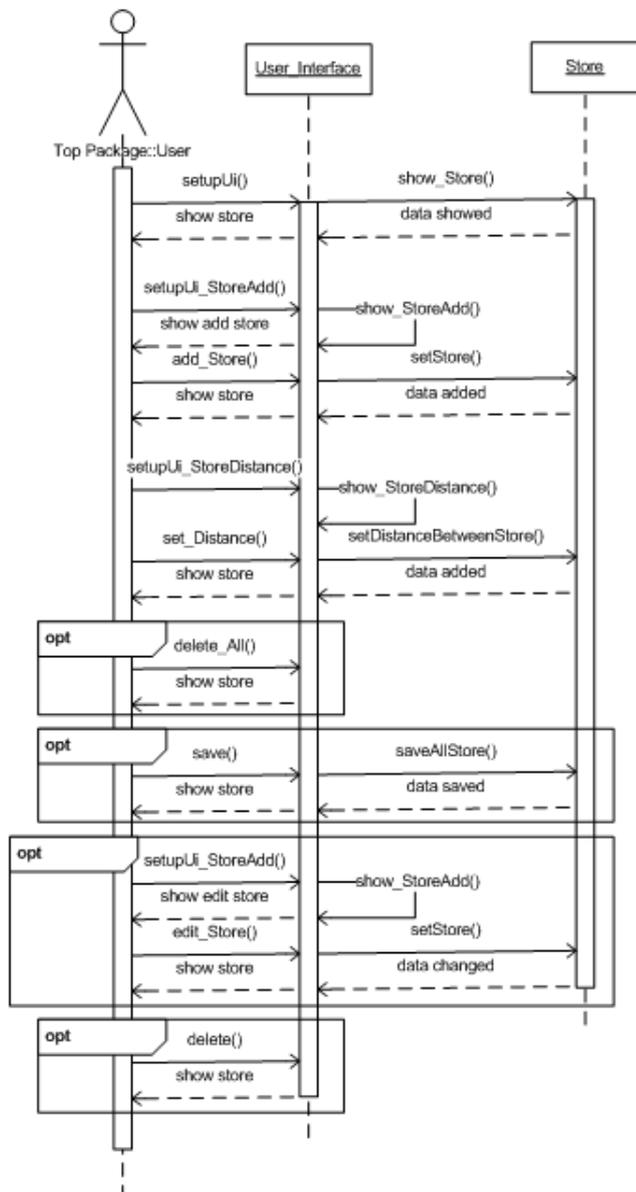


Figure 4 Sequence Diagram of Show_Store

Figure 4 shows the sequence when the user runs the `setUpUi` via `User_Interface`. `User_Interface` invokes the `show_store` and displays the store along with the data from the `Store`. Users enter data store through `User_Interface` followed by `User_Interface` run set store in `Store`. Users also enter a store distance through `User_Interface`, and `User_Interface` runs the distance between stores in the `Store`. There are also options for delete, edit, delete all, and save that can be done by the users on `User_Interface`.

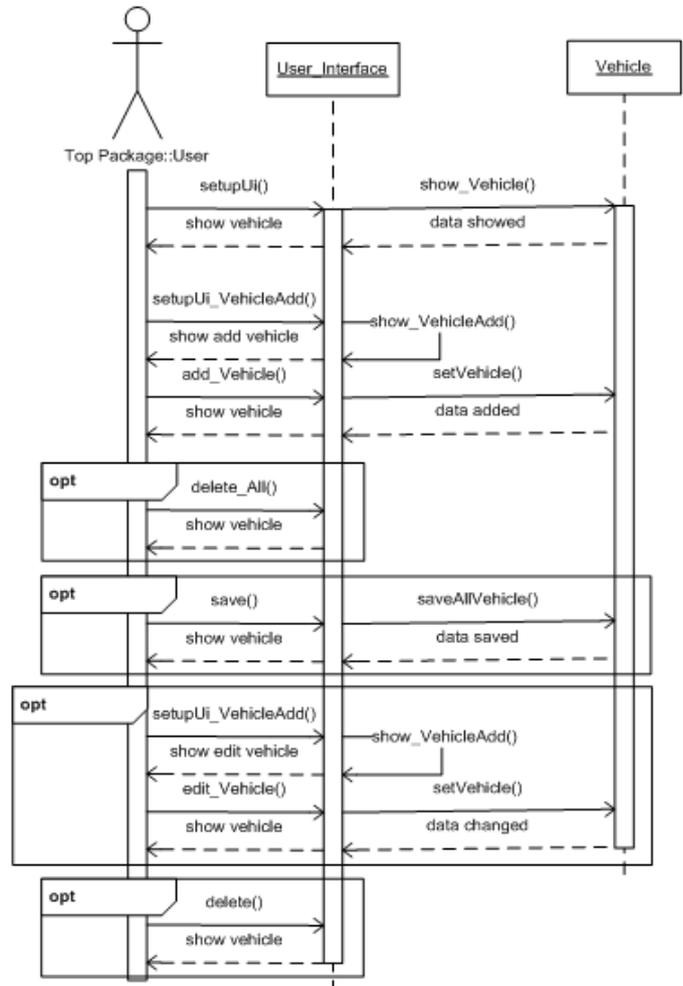


Figure 5 Sequence Diagram of Show_Vehicle

Figure 5 shows the sequence when the user runs the `setUpUi` via `User_Interface`. `User_Interface` shows `show_vehicle` and displays the vehicle along with the data from `Vehicle`. Users enter the vehicle data through `User_Interface`. Then, it input the vehicle set on the `Vehicle`. There are also options for delete, edit, delete all, and save that can be done by the users on `User_Interface`.

Figure 6 shows the sequence when the user runs `setUpUi` through `User_Interface`. Then, `User_Interface` invokes `show_List_Store` and displays the order along with the data from the store. The user selects the data stored on `User_Interface` followed by `User_Interface` invokes the `show_order` product and displays the detailed order along with the data from the product. Last, users enter data order through `User_Interface` by clicking the order set on `Order`. The order delivers the data to the `User_Interface` followed by displaying the return order.

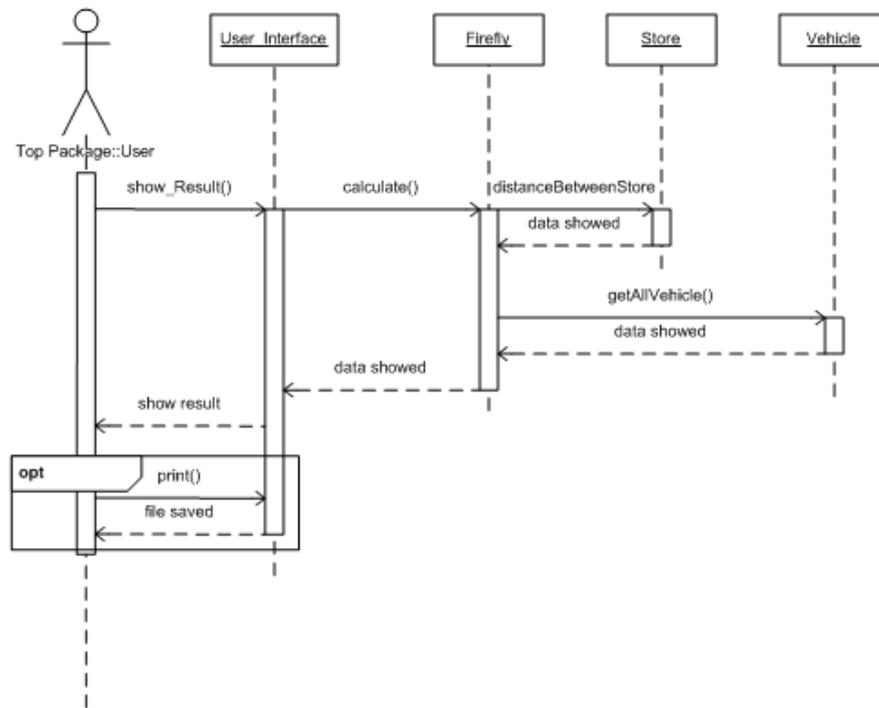


Figure 6 Sequence Diagram of Show Result

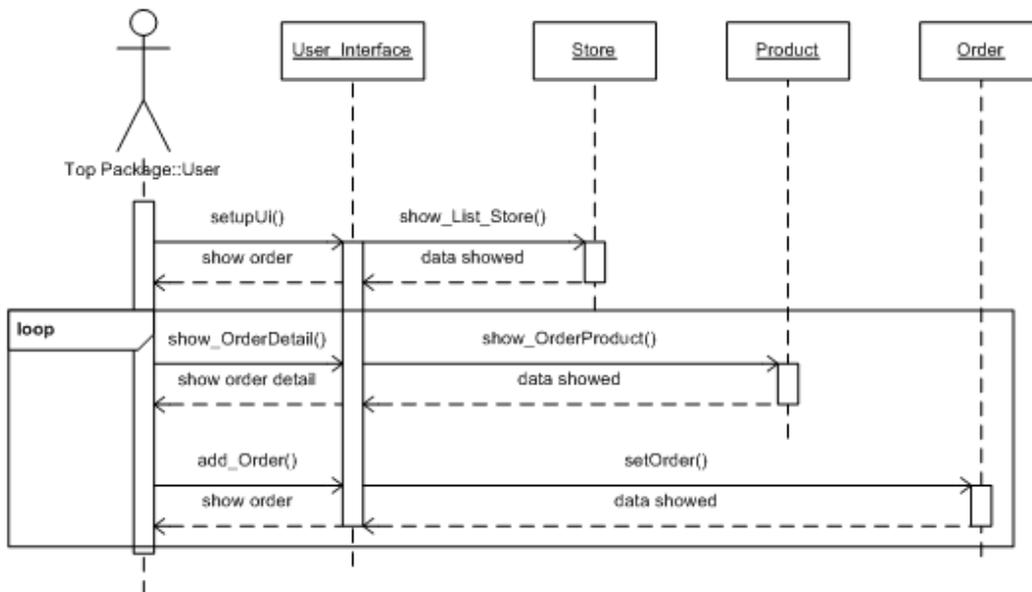


Figure 7 Sequence Diagram of Show_Order Product

Figure 7 shows the sequence when the user runs show_result to see the calculation result through User_Interface. User_Interface invokes calculate on firefly. Then, firefly runs distance between store on store. The store will give data to firefly, followed by running firefly to get all vehicles. The vehicle delivers the data to firefly. Data that has been processed is sent from firefly to User_Interface, which User_Interface. It shows the result to user. There is also an option to print and save in the form of PDF that can be done by the users on User_Interface. The applications are created with five main menus. Home is to see the initial page containing how to use the program. Then, Store is to set the store and depot data. Vehicle is to set the vehicle data and

Product is to organize product data. Last, Order is for users to enter the order data which is followed by calculations on the program that generates the data distribution costs. These select routes and distances.

For the simulation, the result uses a case solved using the firefly algorithm. There are five randomly selected customers from ten customer data. Table 1 is the name of the customer and order quantities of each type of goods. Then, Table 2 is the distance from one store to another. Table 3 is the volume of goods ordered in every store. After the data in the simulation is calculated, the cost of the vehicle and volume of goods are known. Those are shown in Table 4 and Table 5.

Table 1 Simulation of Order

Customer store	Larutan Cap Kaki Tiga	Pocari Sweat	Kratingdaeng
Tk. Aulia	30 box	30 box	20 box
Tk. H. Endin	20 box	20 box	30 box
Tk. H. Mahmud	25 box	30 box	20 box
Tk. Sinar Priangan	30 box	30 box	30 box
Tk. Hamim	30 box	30 box	30 box

Table 2 Simulation of Distance between Stores

Store	DEPOT	Tk. Aulia	Tk. H. Endin	Tk. H. Mahmud	Tk. Sinar Priangan	Tk. Hamim
DEPOT	0 km	2,6 km	3,1 km	5,9 km	3,5 km	3,1 km
Tk.Aulia	2,6 km	0 km	1,2 km	3,3 km	0,9 km	1,3 km
Tk. H. Endin	3,1 km	1,2 km	0 km	3,2 km	0,9 km	2,4 km
Tk. H. Mahmud	5,9 km	3,3 km	3,2 km	0 km	2,4 km	4,7 km
Tk. Sinar Priangan	3,5 km	0,9 km	0,9 km	2,4 km	0 km	2,3 km
Tk. Hamim	3,1 km	1,3 km	2,4 km	4,7 km	2,3 km	0 km

Table 3 Simulation of Volume of Goods Ordered by Stores

Customer store	The volume of goods ordered
Tk. Aulia	2,86 m ³
Tk. H. Endin	2,44 m ³
Tk. H. Mahmud	2,68 m ³
Tk. Sinar Priangan	3,18 m ³
Tk. Hamim	3,18 m ³

Table 4 Cost of Vehicle

Cost per vehicle	Toyota Innova	Toyota Dyna
Fixed Cost		
Maintenance costs	Rp6.000,00	Rp35.000,00
Driver and helper salary	Rp109.000,00	Rp317.000,00
The price of fuel	Rp7.350,00/L	Rp5.150,00/L
The ratio of the fuel needs of vehicle	13,23 km /L	5 km /L
Capacity of vehicle	2,9 m ³	14,4 m ³
Average use of gasoline to five customers in random and manual calculation	Rp17.250,00	Rp71.000,00

Table 5 Volume of Goods

Goods	Volume
Larutan Cap Kaki Tiga	0,036 m ³
Pocari Sweat	0,038 m ³
Kratingdaeng	0,032 m ³

The total volume of the delivered goods does not exceed the capacity of the vehicle (14:34<17,3). Thus, it can be sent. If it uses a route optimization with firefly algorithm edge-based movement, it will display the results as follows.

Assumption 1: Tk. Aulia, 2: Tk. H. Endin, 3: Tk. H. Mahmud, 4: Tk. Sinar Priangan, and 5: Tk. Hamim created 3 firefly and iteration 1, and generated a minimum distance. Following a random path and the result of distance, route, and optimal cost.

Iteration 0:

Firefly A:1>5> 2>3>4(legal)
 distance: 15.4km, light intensity: 2.7184
 Firefly B:2> 4>3>1>5(legal)
 distance: 14.1km, light intensity: 2.7283
 Firefly C:3> 2>5>1>4(legal)
 distance: 17.2km, light intensity: 2.7048 attractiveness: 2.7283, while the best path:
 2>4> 3> 1> 5
 distance: 14.1km, cost: Rp366.523,00

Iteration 1:

FireflyA1a:1>5>2> 4> 3(legal) FireflyA1b:1> 5>3>4> 2(legal) distance: 15km, light intensity: 2.7215
 FireflyB1:2>3>4>1> 5(legal)

distance: 14 km, light intensity: 2.7291
 FireflyC1a: 3>2> 4>5>1(legal) FireflyC1b:
 3>4>2> 5> 1(legal) FireflyC1c: 3>5> 1>2>4(legal)
 FireflyC1d:3>5>1> 4> 2(legal)
 distance: 15.5km, light intensity: 2.7176 attractiveness: 2.7291, while the best path:
 2>3> 4> 1> 5
 distance: 14km, cost: Rp366.420,00

To determine the legitimate or illegitimate route, it should be seen from the volume of each store's goods and the capacity of each vehicle. The capacity of the first chosen vehicle should be smaller and so on. The volume of goods is added to the first vehicle until it can not be added anymore because it exceeds the capacity of the vehicle. Then, the goods will be added to the next vehicle.

The next example uses iteration 0, and firefly A:1>5>2>3>4. The first selected vehicle is Toyota Dyna with a capacity of 14,4 m³. Then, Tk. Aulia has goods about 2,86 m³, so it still has capacity of 11,54 m³ to Tk. Hamim. Next, it carries goods about 3,18 m³ from Tk. Hamim to Tk. H. Endin. The available capacity is 8,36 m³. Then, it will carry goods about 2,44 m³ to Tk. H. Mahmud. The available capacity is 5,92 m³. It brings 2,68 m³ volume of goods too. The last is to Tk. Sinar Priangan with 3,18 m³. Thus, the available capacity is 0,06 m³ and it uses only Toyota Dyna.

To determine the light intensity, it uses the objective function (Equation 1 and 2). After the iteration is complete, it can be seen that the optimal route is the Christian Store> Tk. H. Endin> Tk. H. Mahmud> Tk. Sinar Priangan> Tk. Aulia> Tk. Hamim> Christian Store with the distance of 14 km and costs about Rp366.420,00. The costs of Rp366.420,00 is obtained from the following calculation.

$$z = \left(\frac{Rp\ 5.150,00/L}{5\ L/km} 14km + Rp317.000,00 + Rp35.000,00 \right)$$

$$= Rp366.420,00$$

Christian Store is advised to use this route from Christian Store> Tk. H. Endin> Tk. H. Mahmud> Tk. Sinar Priangan> Tk. Aulia> Christian Stores with a distance of 14 km using a Toyota Dyna. It has the smallest cost of Rp366,420.00. Meanwhile, the manual calculation done by Christian Stores for five customers uses Equation 9. If the case study shows that the results of manual calculations are Rp423.000,00 with assumption as Rp352.000,00 + Rp71.000,00 and the calculations with firefly algorithm are Rp366.420,00. The Christian Store can save about Rp56.580,00. The example of the calculation in the program can be seen in Figure 8.

$$Total\ Cost = f_k + Average\ the\ use\ of\ gasoline\ on\ manual\ calculation \quad (9)$$

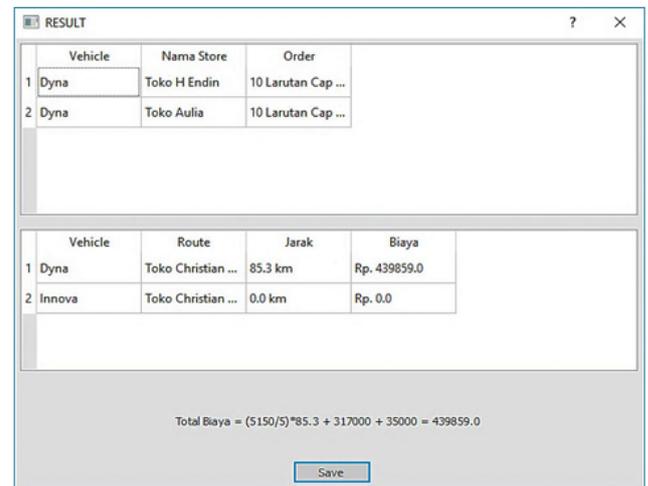


Figure 8 Example of the Calculation in the Program

IV. CONCLUSIONS

The processing time by using the discrete firefly algorithm with edge-based movement on vehicle routing problem solve the problem faster than with the manual process. It can be obtained by optimal route in distribution of goods. This application can help shops to solve problems in the number of customers, requests for different item search customer, the difference in volume, the ratio of fuel, the price of fuel, maintenance of each vehicle, salaries of driver and helper, and the distance between the customer with the depot. This application can help the store to determine vehicles and routes taken as the minimum total distribution costs. Based on the theories that have been discussed and the test results in this application, some suggestions can be used to increase a further development of the application. It is recommended to add features in the programs, so it can enter distance data more easily, display the estimated travel time and the route in detail, and use maps.

REFERENCES

Fister, I., Fister Jr, I., Yang, X. S., & Brest, J. (2013). A comprehensive review of firefly algorithms. *Swarm and Evolutionary Computation*, 13, 34-46.

- Jati, G. K., Manurung, R., & Suyanto, S. (2013). Discrete firefly algorithm for traveling salesman problem: A new movement scheme. *Swarm Intelligence and Bio-Inspired Computation*, 13, 295-312.
- Johari, N. F., Zain, A. M., Mustaffa, N. H., & Udin, A. (2013). Firefly algorithm for optimization problem. *Applied Mechanics and Materials*, 421, 512-517.
- Onwubiko, C. (2000). *Introduction to engineering design optimization*. Upper Saddle River, NJ: Prentice Hall.
- Pan, F., Ye, C., Wang, K., & Cao, J. (2013). Research on the vehicle routing problem with time windows using firefly algorithm. *Journal of Computers*, 8(9), 2256-2261.
- Pornsing, C. (2014). *A particle swarm optimization for the vehicle routing problem* (Dissertations). University of Rhode Island.
- Sayadi, M. K., Hafezalkotob, A., & Naini, S. G. (2013). Firefly-inspired algorithm for discrete optimization problems: An application to manufacturing cell formation. *Journal of Manufacturing Systems*, 32, 78-84
- Yang, X. S. (2009). Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms* (pp. 169-178). Springer, Berlin, Heidelberg.
- Yang, X. S. (2010). *Engineering optimization: An introduction with meta heuristic applications*. New Jersey: John Wiley & Sons, Inc.
- Yesodha, R., & Amudha, T. (2015). A study on bio-inspired meta heuristics for solving vehicle routing problem. *Indian Journal of Science and Technology*, 8(25), 85-93.