Implementation of Traveling Salesman Problem (TSP) based on Dijkstra's Algorithm in Logistics System

M. Firman Aji Sahputra[1], Riri Nada Devita[2], Sherly Allsa Siregar[3], Kartika Candra Kirana[4]
Informatic Engineering
Universitas Negeri Malang
Malang, Indonesia
firman.asbn@gmail.com[1], ririnadadevita@gmail.com[2], sherlyallsasiregar@gmail.com[3], kartika.candra.ft@um.ac.id[4]

***Abstract*—Traveling Salesman Problem (TSP) was defined as a task for finding of the shortest route. The finding of the shortest route influences a price of delivery service and profit of company. Therefore, we proposed an implementation of Traveling Salesman Problem (TSP) based on Dijkstra's Algorithm in Logistics System" for optimizing the finding of shortest route. This algorithm using distance which is extracted from Google Maps. There are 60 routes which are tested. The results show the accuracy of TSP based on Dijkstra's algorithm is 100%. The results can be concluded that the implementation of Dijkstra's algorithm is accurate for finding the shortest route.**

***Keywords*— Traveling Salesman Problem, Dijkstra algorithm, Logistics System.**

I. INTRODUCTION

Delivery services is one of commercial business that is growing rapidly in Indonesia. Transportation plays a critical role in delivery services. Its complexity can take effect only through highly quality management. By means of well-handled transport system, goods could be sent to the right place at right time in order to satisfy customers' demands. It brings efficacy and also it builds a bridge between companies and consumers [1].

One of strategy in delivery services operation is increase the customer satisfaction which can be done with the managing of many data of delivery services accurately, transparantly, and easily. Information system can be used as a solution to manage the data of delivery services. In the information system of delivery services, a determining of a route is an important task. The finding of the shortest route influences the price of delivery service and profit of company. This problem usually was named as Traveling Salesman Problem (TSP). Traveling Salesman Problem (TSP) is classical combinatorial optimization problem for NP-hard problems [2][3]. Many Research have been developed algorithm to solve TSP problem.

Tunon and Lopez (2005) proposed the algorithm of Branch and Bound to solve Traveling Salesman's Problems [2]. Branch and Bound Algorithm has inefficient complexity time because the complexity of Branch and Bound algorithm is O(N!). This algorithm uses a tree diagram for the calculation of the distance.

Additionally, Pragya, Dutta, and Pratyush (2015) develop Dimensional Ant Colony Optimization (DACO) to solve Traveling Salesman's Problems [3]. A disadvantage of this algorithm is random decisions using probability distribution changes by iteration so the convergence of this algorithm is uncertain.

Another research used Dijkstra algorithm. It was performed by [4]. Dijkstra algorithm is an optimization of the greedy algorithm which complexity $O(V^2)$. This algorithm takes the path distances based on the load and calculates the shortest path based on the distance which is taken in every city. Ratnasari (2013) also shows that Dijkstra algorithm has several advantages, in addition to favorable terms of running time, Dijkstra algorithm can resolve some cases finding the shortest path, such as finding the shortest path between two particular node (a pair shortest path), finding the shortest path between every pair of vertices (all pairs shortest path), finding the shortest path from the specific node to all other nodes (single-source shortest path), and finding the shortest path between two vertices through several nodes specific (intermediate shortest path) [5].

From the above description, we proposed *"implementation of traveling salesman problem (TSP) based on Dijkstra in a logistics system"*. This research aims to develop information systems that can provide route recommendations and the sequence the shortest trips between cities of the delivery of goods to be traversed and display the status of goods to be shipped.

II. TRAVELLING SALESMAN PROBLEMS

Traveling Salesman Problem (TSP) is defined as a task for finding of the shortest cycle or path in complete graph of N nodes. It is a classic example of an NP-hard problem. So, the methods of finding an optimal solution involve searching in a solution space that grows exponentially with number of city. Traveling Salesman Problem (TSP) is classical combinatorial optimization problem for NP-hard problems. TSP can be defined as Effective method for solving a NP-hardProblem, because this is heuristic algorithms for obtaining approximate result [6].

The working of the TSP is with metric distances. In the metric TSP, known as delta-TSP, the intercity distances

satisfy the triangle inequality. This can be understood as "no shortcuts", in the sense that the direct connection from A to B is never longer than the tour via C which can use the formula shown in Equation 1 [7].

$$C_{ij} \leq C_{ik} + C_{kj,} \quad (1)$$

where *C* is the distance. *I*, *j*, and *k* are the node of path. Equation 1 shows that the distance of *i,j* have to less than the sum of the distance of *i,k* and the distance of *k,j*.

Some algorithm have been developed some algorithm to solve TSP. Adewole (2011) have been proposed Genetic Algorithm for solving Traveling Salesman Problem [8]. Besided that, Dweepna Garg and Saurabh Shah (2011) used Ant Colony Optimization for Solving Traveling Salesman Problem" [9]. Tunon and Lopez (2005) proposed the algorithm of Branch and Bound to solve Traveling Salesman's Problems [2]. Additionally, Pragya, Dutta, and Pratyush (2015) develop Dimensional Ant Colony Optimization (DACO) to solve Traveling Salesman's Problems [3]. Another research used Dijkstra algorithm. It was performed by [4] and [5].

III. Dijkstra's Algorithm

Dijkstra's algorithm is often known as single source shortest path algorithm. Dijkstra algorithm used the method of increasing node by node to get a shortest path tree which makes the starting point as its root [10]. In the Weighted directed graph, the shortest path node which starts from the starting point s and reaches the earliest must be the smallest point where all the nodes adjacent to *s* and its length of arc is chord length. If the vertices of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities if the vertices from the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities [10][11].

The pseucode of Dijktra Algorithm that can be seen in Figure 1. The first steps of the Dijkstra's algorithm is initialize the graph *G* with list of vertices *V* end list of edges *E*. The second step is the assignment of distance for every node and setting 0 value to current node and setting ∞ to another unvisited node. The third step is calculation the distance from current node to its entire neighbor node and marking as visited node. The visited node cannot be used again. The next process is setting the unmarked node with less distance as current node and then continue from previous step [10]. The relaxation process updates the costs of all the vertices, *v*, connected to a vertex, *u*, if we could improve the best estimate of the shortest path to v by including *(u,v)* in the path to *v* [11][12].

```
initialise_single_source( Graph g, Node s )
   for each vertex v in Vertices( g )
       g.d[v] := infinity
       g.pi[v] := nil
   g.d[s] := 0;

relax( Node u, Node v, double w[][] )
    if d[v] > d[u] + w[u,v] then
       d[v] := d[u] + w[u,v]
       pi[v] := u

shortest_paths( Graph g, Node s )
    initialise_single_source( g, s )
    S := { 0 }        /* Make S empty */
    Q := Vertices( g ) /* Put the vertices in a PQ */
    while not Empty(Q)
        u := ExtractCheapest( Q );
        AddNode( S, u ); /* Add u to S */
        for each vertex v in Adjacent( u )
            relax( u, v, w )
```

Fig. 1. Pseudocode of Dijktra Algorithm [12][13]

IV. Logistics System

Delivery services is the process of transporting the goods, delivered through the transport network. Payload (physical goods) delivered by road using train, by sea using ship and by air using airlines. Therefore, it can be concluded that logistics system is a procedure where a series of data about results and work to deliver the goods to the person who processed into useful information for the company management [14]. The operation of transportation determines the efficiency of moving products. The progress in techniques and management principles improves the moving load, delivery speed, service quality, operation costs, the usage of facilities and energy saving [1].

One of strategy in delivery services operation is increase the customer satisfaction which can be done with the managing of many data of delivery services accurately, transparantly, and easily. An information system can be used as a solution to manage the data of delivery services. The information system is a collection of computer hardware and software as well as human device that will process the data using hardware and software [15]. Information system also is defined as the software that helps organize and analyze data. So, the purpose of an information system is to turn raw data into useful information that can be used for decision making in an organization. [16].

V. Methodology

A. Data Research

Dataset of this research are derived from the Google Map. The variables which was taken, is included by origin shipment, destination shipment, shipment route, and route's distance. In this study, the amount of data are 60 routes. Graph of city which is used in this study, can be seen in Figure 2.

B. Design of Information System

Design of information system is shown by activity diagrams in Figure 3. Activity diagram illustrate the flow of work or activity of a system or business process.[17].

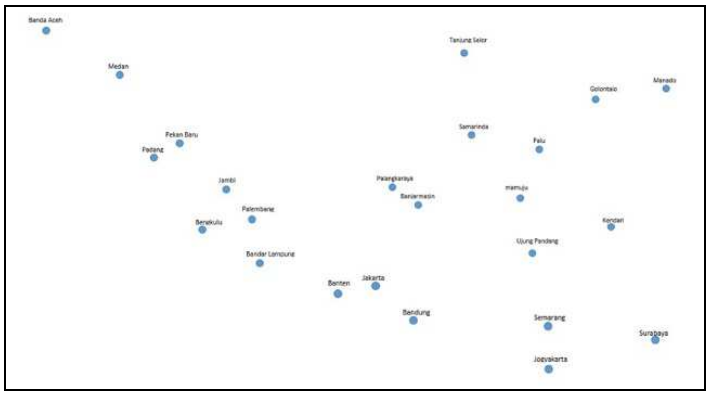

Fig. 2. Node of Town

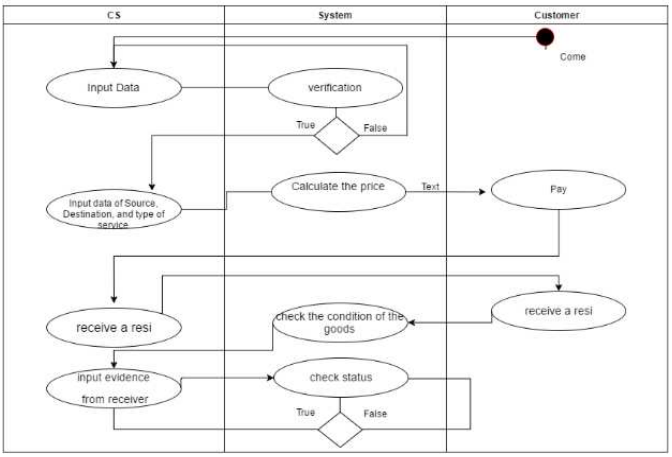

Fig. 3. Activity Diagram

Based on Figure 3, after the customer service input the transactions of goods delivery, the customer will receive a tracking code that will be used to check the status of the goods delivered. Once customer input tracking code, the system will display information, such as origin shipment, destination shipment, the weight of goods shipped, delivery mileage, travel route and delivery status of goods. In this system, Dijkstra's algorithm is used to finding shortest travel route. Every town which are tracked in travel route will be synchronized with delivery status of goods.

C. Dijkstra Algorithm Development Methods

In this study, Dijkstra algorithm is used to find the shortest path. This algorithm using distance of each city for determining the shortest path because it is fixed. This is different with other variables that have a value which is changes every day, such as time which is influenced by the level of congestion. Therefore, this system only considers the shortest route based on distance route search.

Flowchart of Dijkstra algorithm is show in Figure 4. Based on Figure 4, the determination is done by inserting the graph and initializing a vertex where each city is located. Furthermore, the distance of each city has been initialized. After initializing the vertex of each city and determining the distance of each vertex, the next step is determining of a initial vertex (s) and a vertex of interest (t). Then *permanent label = 0* will be given to the initial vertex (s) and *temporary label* = ∞ will be given to another vertex. For each vertex V that has not got a permanent label, they would got *temporary label = min.* After that, it will find the minimum value among all vertices still labeled with the temporary label, and make the minimum vertex while vertices labeled with the permanent label. If there is more than one minimum vertex, it will select one of them. It will be repeated until the system reached the goal. Then, the system will publish the path route [17].

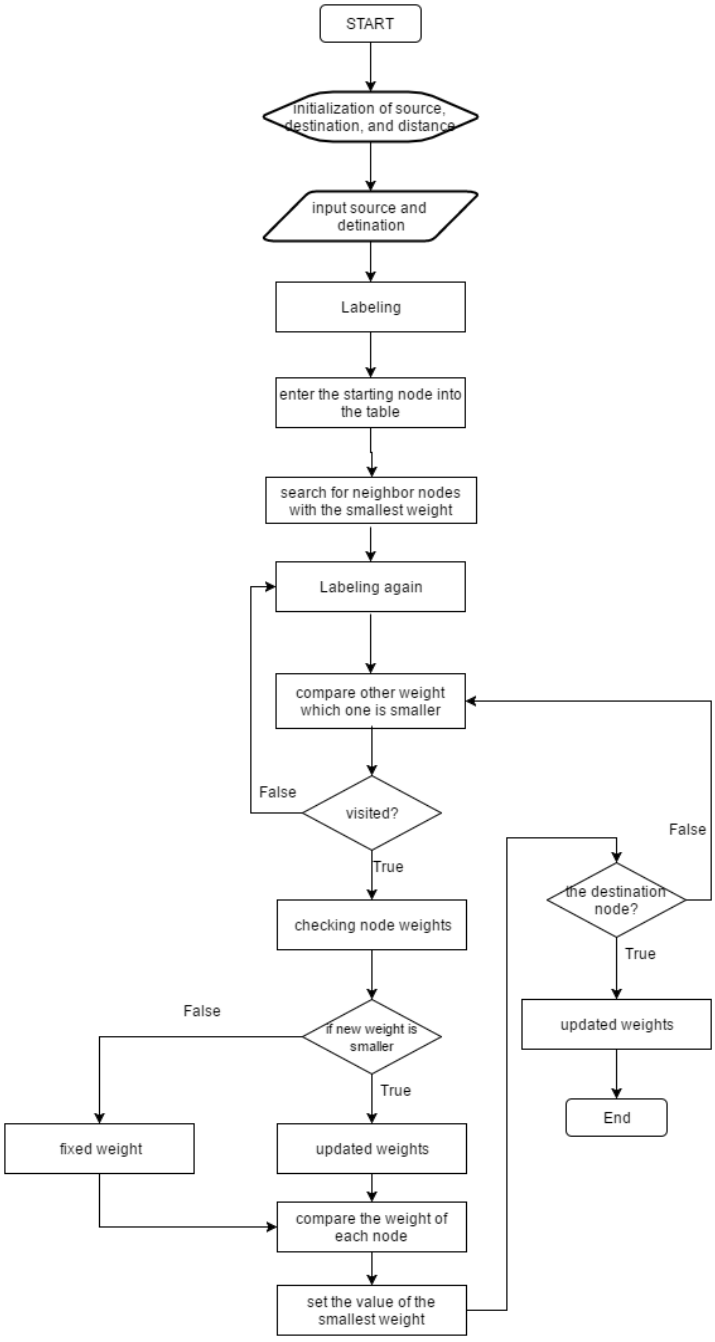

Fig. 4. Flowchart of Dijkstra Algorithm [17]

D. Testing Method

In this study, there are two testing data,namely: route and distance. To calculate the accuracy of these comparisons on a Google map and dijkstra algorithm, can use the formula shown in Equation 2.

$$accuracy = \frac{dA}{n} * 100\% \quad (2)$$

where *dA* is the amount of accurate data which is has a sameness between the Google Map's result and Dijktra algorithm's result. *n* is the total amount of data would be tested.

This research also compared the distance accuration between Google Map's result and Dijktra algorithm's result using Mean Absolute Deviation (MAD). MAD shows the sum of the absolute errors [18]. This method measures accuration of forecasting by averaging the alleged error (absolute value of each error). MAD useful when measuring forecast error in the same units as the original series. MAD can be calculated using the formula which is shown in Equation 3.

$$\text{MAD} = \frac{\sum e}{n} \quad (3)$$

MAD is calculated by divide *e* error with *n* data. *e* is absolute value of decrement between real distance (*rd*) with estimation distance (*ed*) which is shown in Equation 4.

$$e = |\text{rd} - \text{ed}| \quad (4)$$

VI. Results And Discussion

A. Result of Route Search Development on Information Delivery System

In this system, customer services fill out the form of customer request for delivery services. Page of delivery service input is shown in Figure 5. Upon completion, those data will be stored on the database. All of transaction which is shown in Figure 6 can be accessed by customer services. After the customer service fill out the transactions of goods delivery, the customer will receive a tracking code that will be used to check the status of the goods delivered. The information of goods tracking can be accessed by inputting the tracking code on tracking page which is shown in Figure 7. If the tracking code that is entered is valid, the system will show the delivery services description such as origin shipment, destination shipment, the weight of goods shipped, delivery mileage, travel route and delivery status of goods that is shown in Figure 8.

Fig. 5. Page of Delivery Services Input

Fig. 6. Transaction Page

Fig. 7. Tracking Page

Fig. 8. Output Tracking Page

B. Testing Result for TSP Based on Dijkstra Algorithm

After developing information systems, researchers conducted testing of Dijkstra algorithm performance. There are 60 data taken as a testing data. The conclusion of accuracy of each variable are shown in Table 1. Table 1 shows the travel route of Dijkstra algorithm has reached an accuracy up to 100% and the distance accuration of dijkstra algorithm has reached 4.3.

In this study, we just search the town route based on distance, even though the shortest route also can be determined based on shortest time or shortest cost. Sometimes, the shortest distance does not always bring the best profit. It can be shown from the difference of distance which are calculated by using Dijktra Algorithms and Google Maps. The sample of route distance which is processed by Dijkstra algorithm and Google Map are shown in Tables 2. Based on Table 2, the accuracy of distance which is could not achieve a perfect result with no error is caused by the difference of center distance calculations on the Google map and the Dijkstra algorithm. The distance calculation of Google map does not always pass through the center of town, while the distance calculation of of dijkstra algorithm will start between downtown (per branch).

In some routes, the distance calculations of dijkstra algorithms is not same as the distance calculations of google maps because the dijkstra algorithm will calculate the distance based on the length among the nodes which is the center of town. For example, if we want to calculate from town A to town C and the route is A - B - C, the dijkstra algorithm will calculate the distance from the center of town A to the center of town B and the center of town B to the center of town C, whereas google maps will calculate the distance not just by the center of town. Google maps will calculates all possible paths (through town B) including paths that do not pass through the center of town B. If the distance which is calculated by google maps pass through the center of town, the distance calculations of dijkstra will be same with the distance from google maps.

In addition,, we also do not consider the distance of the city center to the delivery address because we assume that the distance of town center and delivery address is not influenced the conclution of shortest route which is shown in Table 3. However, this is causes the distance calculation is less accurate.

TABLE I. THE RESULTS

No	Testing	Result
1	Route accuracy	100%
2	Distance accuracy (MAD)	4,3

TABLE II. THE COMPARATION OF DISTANCE *

No	Source	Destination	Route Distance	
			Google Map	Dijkstra Algorithm
1	Surabaya	Mamuju	2042	2042
2	Palangka-raya	Tanjung Selor	1321	1321
3	Palembang	Palangkaraya	1997	1998
4	Medan	Surabaya	2042	2042
5	Bandung	Medan	1321	1321

* 5 of 60 sample data for verify the route

TABLE III. THE SAMPLE OF RESULTS*

No	Source	Destination	Travel Route	
			Google Map	Dijkstra Algorithm
1	Surabaya	Mamuju	Surabaya - Ujung pandang - Palu - Mamuju	Surabaya - Ujung pandang - Palu – Mamuju
2	Palangkaraya	Tanjung Selor	Palangkaraya-Banjarmasin-Samarinda-Tanjung Selor	Palangkaraya-Banjarmasin-Samarinda-Tanjung Selor
3	Palembang	Palangkaraya	Palembang-Lampung-Jakarta-Semarang-Palangkaraya	Palembang-Lampung-Jakarta-Semarang-Palangkaraya
4	Medan	Surabaya	Medan - Jambi - Lampung - Jakarta - Semarang - Surabaya	Medan - Jambi - Lampung - Jakarta - Semarang – Surabaya
5	Bandung	Medan	Bandung - Jakarta - Lampung - Palembang - Jambi - Medan	Bandung - Jakarta - Lampung - Palembang - Jambi - Medan

* 5 of 60 sample data for verify the route

VII. CONCLUSION

The results has been reached the accuracy of 100%. Thus, it can be concluded that the dijkstra algorithm accurately in the shortest route search problems. The weakness of this system is

not precisely calculate the distance. It should consider the nodes as the addresses of each locations for delivering the goods, not as the center of town because the differences of distance calculation can affect the cost of goods shipping. In addition, the weaknesses of this system is only consider the distance for searching the shortest route. Sometimes, the shortest distance does not bring the best profit compared with the shortest time, so it takes some combination of variables to improve the accuracy of distance prediction. However, all of the weakness of this research can be used as further research.

REFERENCES

[1] Y.Y. Sheng, "The Role of Transportation in Logistics Chain", Proceedings of the Eastern Asia Society for Transportation Studies, Vol. 5, pp. 1657 - 1672, 2005

[2] M.I.C., Tunon and M.R. Lopez."Design and Use of the CPAN Branch and Bound for The Solution of the Travelling Salesman Problem (TSP)" International Conference on Electronics, Communication and Computers, Conielecomp, 1 August 2005.

[3] Pragya, M.Dutta, and Pratyush, "TSP Solution Using Dimentional Ant Colony Optimization", International Conference on Advanced Computing & Communication Technologies, 6 April 2015.

[4] M.M. Mena, R.H. Ucan, V.C. Vetina, and F.M. Ramirez, "Web Services Compotition using the Bidirectional Dijktra Algorithm" IEEE Latin America Transaction, vol 14, issue 5, May 2016

[5] A. Ratnasari, F. Ardiani, F. Nurvita. "Penentuan Jarak Terpendek dan Jarak Terpendek Alternatif menggunakan Algoritma Dijkstra serta Estimasi Waktu Tempuh". Universitas Islam Indonesia. ISBN: 979-26-0266-6. Yogyakarta 2013

[6] E. Kasturi and S. Lakshmi Narayanan, "A Novel Approach to Hybrid Genetic Algorithms to Solve Symmetric TSP", Volume 2, Issue 2, February 2014, ISSN: 2321-7782

[7] Mohd. Junedul Haque and Khalid. W. Magld. "Improving the Solution of Traveling Salesman Problem Using Genetic, Memetic Algorithm and Edge assembly Crossover", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 7, 2012

[8] Adewole Philip, Akinwale Adio Taofiki and Otunbanowo Kehinde, "A Genetic Algorithm for Solving Travelling Salesman Problem" (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No.1, January 2011

[9] Dweepna Garg and Saurabh Shah, "Ant Colony Optimization For Solving Traveling Salesman Problem", International Journal of Computer Science and System Analysis Vol. 5, No. 1, January-June 2011, pp. 23-29, ISSN 0973-7448

[10] Vaibhavi Patel and Prof. ChitraBaggar, "A Survey Paper of Bellman-Ford Algorithm and Dijkstra Algorithm For Finding Shortest Path in GIS Application", International Journal of P2P Network Trends Technology(IJPTT), Vol. 5, February 2014

[11] Ni Kai, Zhang Yaoting, Ma Yuepeng, "Shortest Path Analysis Based on Dijkstra's Algorithm in Emergency Response System" Indonesian Journal of Electrical Engineering and COmputer Science, vol. 12, no. 5, pp 133-139, 2015

[12] S.S. Biswas, B. Alam and M.N. Doja, "Generalization of Dijkstra's Algorithm for Extraction ff Shortest Paths In Directed Multigraphs " Journal of Computer Science, vol. 9, no.3, pp. 377-382, 2013

[13] Muhamad Ibrahim, Hadi Muhammad Diar and Gandi Suryanto, "Perancangan Sistem Rute Terpendek Menggunakan Algoritma Greedy". Universitas Komputer Indonesia, January 2014

[14] Hendy Setiady and Yulistia. "Sistem Informasi Pemesanan dan Penjualan Berbasis Web pada Dewi Florist". STMIK GI MDP, August 2012

[15] Andri Kristanto, "Perancangan Sistem Informasi dan Aplikasinya". Yogyakarta: Gava Media, 2008

[16] G.G Gable; D. Sedera, T. Chan,"Re-conceptualizing Information System Success: The IS-Impact Measurement Model" Journal of the Association for Information Systems, vol. 9, no. 7, pp. 377-408. 2008

[17] Rahayu Ningati, "Aplikasi Pencarian Rute Terpendek Daerah Wisata Kota Kediri menggunakan Algoritma Dijkstra". Universitas Nusantara PGRI Kediri, 2014

[18] D. Larasati, B.S. Yuusufa, and K.C. Kirana, "Sistem Pendukung Keputusan Produksi Berbasis Fuzzy Pada Sistem Informasi Jual Beli Mobil Bekas", vol. 3 , no. 10, pp. 29-36. 2016