Shortest Route at Dynamic Location with Node Combination-Dijkstra Algorithm

*Note: Sub-titles are not captured in Xplore and should not be used

line 1: 1st Achmad Fitro
line 2: *Magister Program of Information System*
line 3: *School of Postgraduate Studies, Diponegoro University*
line 4: Semarang, Indonesia
line 5: afi.subarjo@gmail.com

line 1: 2nd Suryono
line 2: *Lecture Magister Program of Information System*
line 3: *Faculty of Information System*
line 4: Semarang, Indonesia
line 5: suryonosur@gmail.com

line 1: 3rd Retno Kusumaningrum
line 2: *Lecture Magister Program of Information System*
line 3: *Faculty of Information System*
line 4: Semarang, Indonesia
line 5: retno@live.undip.ac.id

Abstract*— Online transportation has become a basic requirement of the general public in support of all activities to go to work, school or vacation to the sights. Public transportation services compete to provide the best service so that consumers feel comfortable using the services offered, so that all activities are noticed, one of them is the search for the shortest route in picking the buyer or delivering to the destination. Node Combination method can minimize memory usage and this methode is more optimal when compared to A* and Ant Colony in the shortest route search like Dijkstra algorithm, but can't store the history node that has been passed. Therefore, using node combination algorithm is very good in searching the shortest distance is not the shortest route. This paper is structured to modify the node combination algorithm to solve the problem of finding the shortest route at the dynamic location obtained from the transport fleet by displaying the nodes that have the shortest distance and will be implemented in the geographic information system in the form of map to facilitate the use of the system.

***Keywords*— Shortest Path, Algorithm Dijkstra, Node Combination, Dynamic Location** *(key words)*

I. Introduction

In the world of technology which is very rapid develop-ment, a lot of information contains in an internet both spatial and textual information. The information can be combined into spatial queries or called Geographic In-formation Systems (GIS) [1]. Geographic Information Systems are widely used in information technology such as information on flood-prone areas, land match-ing to plants, tourism, searching for company location or searching for a person's location [2].

Geographic Information System is a very useful tool for spatial analysis that provides functionality for stor-ing, capturing, analyzing and displaying geographic information [3]. However, GIS can also manipulate and display all forms of geo-referenced information efficient-ly so that it can give a decision to the user [4]. One ex-ample is the information used by the user in showing the decision to guide the trip by showing which route or path to go through to reach the destination by display-ing information through the website or mobile device [5]. There are two points to be considered in the process of making decisions about the route of the road on GIS [6], namely: location type and the shortest route search.

A static location is a location that has unchanging coordinates such as gas station locations, hotels, hospi-tals and companies. While the dynamic location is the location where the coordinates may change at any time such as the car or the location of someone who is al-ways on the move [7]. One company that needs this de-cision is a public transport company like a taxi, where the administrator can monitor the fleet and guide the route to the taxi customer. In addition, the shortest route search is now also very popularly used by companies to assist in the delivery of products to consumers [8], con-struction of apartments to regulate the exit points of de-velopment materials [9] as well as a private person who travels.

With the increasing number of users search the shortest route then some researchers use algorithms to overcome the problem including algorithms A*, Dijkstra algorithm and Ant Colony algorithm. Based on existing shortest path search algorithms, the researchers com-pete each other in providing the shortest route to static locations, especially dynamic locations, where the algo-rithms are tested by detailing the shortest route infor-mation with the fastest time and accuracy in the search for the shortest route where the destination location moves constantly.

The shortest route search by using Dijkstra algo-rithm is very popular than other algorithms where Dijkstra algorithm uses greedy principle, that is the steps chosen with minimum weights that will connect between the selected vertices with the other node [10] so that it can provide information the fastest and shortest route. The previous research concludes that Dijkstra's algorithm provides the shortest path information with faster search than the A* algorithm and the Ant Colony algorithm [11],[12],[13],[14].

Dijkstra's algorithm is very fast in searching the shortest route on the undigraph model (not directed) [15]. Dijkstra's algorithm, however, has algorithmic changes for some improvements in optimizing the search for the diagraph model, including: 1. increasing the effectiveness of the diagraph output to avoid infinite looping 2. the algorithm will notice adjacent nodes in the shortest path search, and 3. considering many vertices that can separate labels simultaneously [16]. However, this algorithm does not discuss the memory used in the shortest route search so that researchers have an alternative in an effective and efficient search by com-bining Dijkstra algorithm with Node

Combination algo-rithm, where Node Combination is an algorithm that has easy to understand steps and smaller memory us-age compared to Dijkstra's algorithm, and the Node Combination algorithm can be applied to Dijkstra's al-gorithm [17].

Based on the background that has been described, the researcher will modify the node combination algo-rithm to solve the problem in the storage of the nodes passed in the search for the shortest route.

A. Geolocation

Geolocation is a technique of identifying geograph-ical location in the real world that comes from an inter-net-connected object, such as a mobile phone, radar or computer. [18],[19]. In geolocation itself there is the simplest form, where geolocation involves a set of geo-graphical coordinates that are closely related to the use of the system in positioning such as a street address at a location [20].

For geolocation or position, search engines often use radio frequency location (RF) methods, such as Time Difference Of Arrival (TDOA) for precision. TDOA sys-tems often use mapping or other geographic infor-mation systems. If GPS signals are not available, geolo-cation apps may use information from cell towers to triangulate position estimates, methods that are not as accurate as GPS but have improved considerably in re-cent years. This is different from previous location radio technology, such as Direction Finding where the bear-ing line to the transmitter is reached as part of the pro-cess. The word geolocation also refers to the latitude and longitude coordinates of certain locations standard-ized by ISO / IEC 19762-5: 2008 [21].

B. Google Maps API

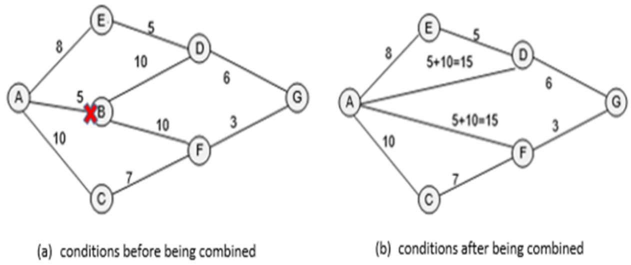
Google Map API (Application Programing Interface) is one of the advantages provided by google to access data from google map and google local search. Application Programing Interface is a documentation that consists of interface, function, class, structure and so on to build a software. With this API, it allows the programmer to develop a software to be integrated with other software. API can be said to link an application with other applications that allow programmers to use the system function. This process is managed through the operating system. The advantage of this is to allow an application with other applications to interact and interact. The programming language used by google map consisting of html, javascript, ajax, and xml, allows to display google maps on web pages.

By using the google map API, we can display the digital map on its own website page. In order for this API application to appear, it is necessary to have a key API that is a unique code that is generalized by google map to recognize a number of utilities for manipulating maps and adding content in maps through various services, allowing the creation of powerful map applications on web pages [22],[23].

C. Node Combination Algorithm

Node combination has a search system by combining all nodes and still maintaining a labeling set such as Dijkstra's algorithm which will be depicted in “Fig. 1” [24],[17] Suppose all nodes are connected to a rope and will be pulled slowly, over time all nodes will converge into one where the string will see the shortest strap.

Fig. 1. An example of a combination of nodes



The shortest path search steps using a combination node can be explained by pseudocode to be described in Table 1

TABLE I. PSEUDOCODE ALGORITMA NODE COMBINATION

Algoritma Node Combination (G,s)
1 W[s,u] := 0, vu := vs, V := V – {s} /* Initialization */
2 while W[s,u] < ∞ and |V| > 0
3 V := V – {u} /* Node Combination */
4 for each j in V
5 W[s,j] := min {W[s,j], W[s,u] + W[u,j]} /* updating edge weights */
6 vu := the nearest neighbor of s in V /* finding the nearest neighbor */
/* at the end of the algorithm, the sth row in W contains the corresponding distances */

D. Node Combination-Dijkstra Algorithm

Node Combination-Dijkstra algorithm is algorithm developed by writer in search of shortest route by adopting Node Combination search method that is combining one node with other node and modification in case of storage node which will be explained by pseudocode in Table 2.

TABLE II. PSEUDOCODE ALGORITMA NODE COMBINATION-DIJSKTRA

Algoritma Node Combination-Dijkstra (G,s)
1. $W[s,u] := 0$, $v_u := v_s$, $P[u] := [s]$, $V := V - \{s\}$/* Initialization */
2. while $W[s,u] < \infty$ and $\|V\| > 0$
3. $V := V - \{u\}$ /* Node Combination */
4. for each j in V
5. If $W[s,u] + W[u,j] < W[s,j]$
6. Then $P[j] := P[u] \cup \{j\}$, /* updating path */
7. $W[s,j] := W[s,u] + W[u,j]$ /* updating edge weights */
8. *vu := the nearest neighbor of s in V /* finding the nearest neighbor */*
/* at the end of the algorithm, the sth row in W contains the corresponding distances */

II. Data

The data used for this study is the location of places in the city surabaya which is interconnected between one point with another point with information weight / distance in meters, latitude longitude coordinates, and address which will be explained in Table 3.

TABLE III. CORDINAT POINTS

No	From	To	Coordinat	Weight	Addres
1	A	B	"-7.274982997930746, 112.72668129821636"	1692.91	Jl. Pasar Kembang Kec. Sawahan, Kota SBY, Jawa Timur
		D		954.0145	
		E		728.3023	
		G		2103.285	
2	B	A	"-7.272939639599, 112.741873330076"	1692.91	Jl. Basuki Rahmat Embong Kaliasin, Genteng, Kota Surabaya
		C		826.8011	
		M		2451.245	
		J		1692.327	
3	K	I	"-7.283092639519452, 112.75612122436382"	785.2675	Bank Antardaerah. PT - Pucang Anom Timur Jl. Pucang Anom Tim. No.19, Surabaya
		J		483.4498	
		N		1374.426	
4	L	M	"-7.269129757840544, 112.77423149963238"	1624.072	Jl. Raya Dharma Husada Indah Mulyorejo, Kota SBY, Jawa Timur
		I		1759.133	
5	M	B	"-7.262233296342604, 112.76127106567242"	2451.245	Jl. Kedung Tarukan No.86C Pacar Kembang, Tambaksar
		L		1624.072	
6	N	K	"-7.290867201464144, 112.74645163518085"	1374.426	STIE YOUTH SURABAYA JL Bung Tomo, No. 08, Kavling 8 Surabaya
		G		1388.481	
		O		727.7807	
7	O	N	"-7.297337545443966, 112.74739577275409"	727.7807	Jl. Ngagel Rejo Kidul 54-38 Ngagelrejo, Wonokromo, Kota Surabaya
		P		1439.804	
8	P	O	"-7.309256355149782, 112.74233176213397"	1439.804	Bendul Merisi Wonocolo, Kota Surabaya
		Q		760.3496	
9	Q	P	"-7.306617217538221, 112.7359802911867"	760.3496	Jl. Raya Malang - Surabaya Wonokromo, Kota Surabaya
		H		790.3205	
10	C	B	"-7.266213519414126, 112.73869759460308"	826.8011	Jl. Tegalsari No.25 Tegalsari, Kota Surabaya
		D		1670.288	
11	D	C	"-7.26697979135388, 112.7235913934312"	1670.288	Jl. Petemon No.5 Petemon, Kec. Sawahan, Kota Surabaya
		A		954.0145	
12	E	A	"-7.278729130708208, 112.72127396484234"	728.3023	Jl. Putat Jaya 30 Putat Jaya, Kec. Sawahan, Kota Surabaya
		F		1539.035	
13	F	E	"-7.292266031440466, 112.72410637756207"	1539.035	Jl. Mayjen Sungkono 75-143 Gn. Sari, Dukuh Pakis, Kota Surabaya
		G		1090.059	
14	G	A	"-7.292436304316622, 112.73397690673687"	2103.285	Bubur Ayam Rasa Darmo, Wonokromo, Surabaya
		F		1090.059	
		N		1388.481	
		H		807.5721	
15	H	Q	"-7.299672841676093, 112.73449189086773"	790.3205	Jl. Joyoboyo Sawunggaling, Wonokromo, Kota Surabaya
		G		807.5721	
16	I	L	"-7.2796019598365245, 112.76230103393414"	1759.133	Jl. Raya Menur Manyar Sabrangan, Mulyorejo, Kota Surabaya
		J		698.3279	
		K		785.2675	
17	J	B	"-7.278750570426501, 112.75603539367535"	1692.327	Jl. Dharmawangsa No.97 Kertajaya, Gubeng, Kota Surabaya
		I		698.3279	
		K		483.4498	

III. Result and Discussion

A. Dynamic Location Search

Determination of coordinates on the built system uses a geolocation where the identification or estimation of the real-world geographical location of an object such as a mobile phone. Determination of coordinates on this system is the basic system built to sync the object coordinates with the coordinates that already exist in the system.

Locations generated from geolocation are not all coordinates listed on the location data in this shortest route search information system so that the determination of coordinates generated from the geolocation with the coordinates present in the system will be calculated for the closest search of the coordinates to the coordinates present in system.

The algorithm for calculating the coordinate distance of one with the other coordinates will be written in the Table 4 about pseudocode distance calculation.

TABLE IV. PSEUDOCODE DISTANCE CALCULATION

```
function jarak ($lat1, $lon1, $lat2, $lon2, $hasil)
$kordinat= $lon1 - $lon2;
$jarak = sin(deg2rad($lat1)) * sin(deg2rad($lat2)) + cos(deg2rad($lat1)) * cos(deg2rad($lat2)) * cos(deg2rad($kodinat));
$ jarak = acos($jarak);
$ jarak = rad2deg($jarak);
$mili = $ jarak * 60 * 1.1515;
$hasil = strtoupper($hasil);
 if ($hasil == "K") {
 return ($mili * 1.609344)
 } else if ($hasil == "N") {
 return ($mili * 0.8684);
} else {
return $mili; }
end
```

After the calculation of the algorithm it will get the distance between the coordinates of one with the coordinates in the system then, the location is given the coordinates that have the closest distance so that the location will be initialized as a starting point in the search for the shortest route. “Fig. 2” is an example of the position of the geolocation represented by the letter “P” with the graph contained in the system.

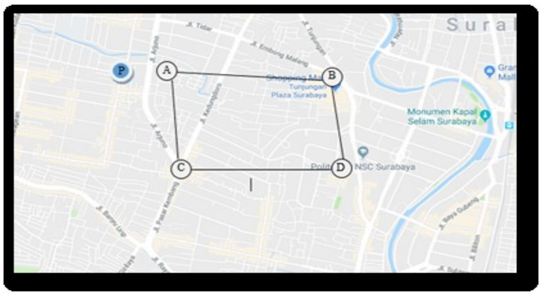


Fig. 2. Position of geolocation with graph.

From “Fig. 2” we can conclude that the position of the geolocation is initialized starting point is at point “A” because that point has the most shortest distance.

B. Implementation of Node Combination-Dijsktra Algorithm

The Node Combination-Dijkstra algorithm has a base that the initial node will find the closest vertices to the destination node, when it finds the nearest node, the node will be joined with the next node until the end result of this system will be a node that informs the shortest distance between the initial node with the destination node along with the points that have been passed.

For the application of Node Combination-Dijkstra algorithm will be described in Table 5 where the authors will calculate into the assessment matrix starting from point I with the goal to point C. with this matrix the writer can show that the Node Combination-Dijkstra algorithm has the same calculation pattern with Node Combination but the algorithm can also save the history passed.

Initials : W[s, u] := 0, vu := vs, P[u] := [s] := [1], V := V – {s}

TABLE V. THE ROUTE SEARCH MATRIX FROM POINT I TO POINT C

Step	Start	From I to C																
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	*I* =698										698 {I,J}	785 {I,K}	1759 {I,L}					
2	*J* =785		698+1692 =2390 {I,J,B}									698+483 =1181 > 785 Then 785 {I,K}	1759					
3	*K* =1759		2390										1759		785+1374 =2159 {I,K,N}			
4	*L* =2159		2390											1759+1624 =3383 {I,L,M}	2159			
5	*N* =2390		2390 {I,J,B}											3383		2159+727 =2886 {I,K,N,O}		
6	*B* =2886	2390+1692 =4083 {I,J,B,A}		2390+826 =3216 {I,J,B,C}										3383		2886		
7	*O* =3216	4083		3216 {I,J,B,C}										3383		4325	2886+1439 =4325 {I,K,N,O,P}	

From Table 5 it can be concluded that the system of finding the shortest route from point I to point C can be reached through the point I-J-B-C with a distance of 3216 meters.

C. *System Analysis*

To prove that the system has chosen the shortest route, a few alternate paths from the starting point are located at point I with the end point at point C. Alternative routes that can be passed will be explained in Table 6.

TABLE VI. ROUTE SEARCH ALTERNATIVES

Nomor Alternatif	Start	End	Rute	Jarak yang ditempuh
Alt. 1	I	C	I-J-B-C	3216 (m)
Alt. 2			I-K-J-B-C	3786 (m)
Alt. 3			I-L-M-B-C	6660 (m)
Alt. 4			I-L-M-B-A-D-C	10150 (m)
Alt. 5			I-K-N-O-P-Q-H-G-A-D-C	11409 (m)

In Table 6 we can see that there are several alternatives in route determination and we can conclude that the shortest route search system using the Node Combination-Dijkstra Algorithm selects the first alternative with a distance of 3216 meters which is the shortest distance among other alternatives, and also, the system can record the history of the nodes passed the

CONCLUSION

This research has presented the shortest route search system by modifying the combination node algorithm to the dynamic location used in general such as mobile phone online transportation. From the test result of the system, the node combination-djikstra algorithm can give optimal result in the shortest route search on the dynamic location applied in surabaya city that shows the route search from point I to point C is 3216 meters for the distance with I-J-B-C route.

REFERENCES

[1] Jie, Z., Jin, P., Zhang, Q., dan Wen, R., 2014, Exploiting location information for web search, Computer in human behavior 30, 378-388

[2] Pugas, D.O., Somantri, M., and Satoto, K.I. 2011. Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra dan Astar (A*) pada SIG Berbasis Web untuk Pemetaan Pariwisata Kota Sawahlunto.

[3] Rikalovic, A., Cosic, I., dan Lazarevic, D., 2014, GIS Based Multi-Criteria Analysis for Industrial Site Selection., Procedia Engineering 69 (2014) 1054-1063.

[4] Church, R.L., 2002, Geographical information systems and location science, Computers & Operations Research 29 (2002) 541-562.

[5] Krichen, S., Faiz, S., Tlili, T., and Tej, K. 2014. Tabu-based GIS for solving the vehicle routing problem. Expert systems with application 41 (2014) 6483-6493.

[6] Bo, H., Liu, N., dan Magesh, C., 2006, A GIS supported Ant algorithm for the linear feature covering problem with distance constraints, decision support system 42, 1063-1075.

[7] Yin, X., Ding, Z., dan Li, J., 2008, A Shortest path algorithm for moving objects in spatial network databases., Progress in Natural Science (2008) 893-899.

[8] Soltani, A.R., Tawfik, J.Y., Goulermas., dan Fernando, T., 2002, Path planning in construction sites : performance evaluation of the Dijkstra, A*, and GA search algorithm, Advanced engineering informatics 16, 291-303.

[9] Aleksandar, R., Ilija, C., Djordje., dan Lazarevic., 2013, GIS Based Multi-Criteria Analysis for Industrial Site Selection, Procedia Engineering 69, 1054-1063.

[10] Peyer, S., Rautenbach, D., Vygen, J., 2009, A generalization of Dijkstra's shortest path algorithm with applications to VLSI routing, Journal of Discrete Algorithms 7 (2009) 377–390.

[11] Choubey, N., Bhupesh., dan Gupta, K.R., 2013, Analysis of Working of Dijkstra and A* to Obtain Optimal Path, International Journal of Computer Science and Management Research Vol 2 Issue 3 March 2013 ISSN 2278-733X.

[12] Dramski, M.,2012, A comparison between Dijkstra algorithm and simplified ant colony optimization in navigation, Scientific Journals 29 (101), pp. 25–29.

[13] Pratiarso, A., Hadi, M.Z.S., Yuliana, M., and Wahyuningdiyah, N. 2010. Perbandingan Metode Ant Colony Optimization dan Dijkstra untuk Pengembangan Sistem Pengiriman Barang di Kantor Pos Area Surabaya Timur Berbasis J2ME. ISSN: 2088-0596 (2010). EEPIS.

[14] Santoso, L.W., Setiawan, A., dan Prajogo, A.K., 2012, Performance Analysis of Dijkstra, A* and Ant Algorithm for Finding Optimal Path Case Study: Surabaya City Map, Informatics Department, Faculty of Industrial Engineering.

[15] Chen, Y.Z., Shen, S., Chen, T., dan Yang, R., 2014, Path Optimization Study for Vehicles Evacuation Based on Dijkstra algorithm, Procedia Engineering 71 159 – 165.

[16] Shu-Xi, W., 2012, The Improved Dijkstra's Shortest Path Algorithm and Its Application, Procedia Engineering 29 , 1186 – 1190.

[17] Lu, X., dan Camitz, M.., 2011, Finding the shortest paths by node combination, Applied Mathematics and Computation 217, 6401–6408.

[18] Ciavarrini, G., Luconi, V., and Vecchio, A. 2017. Smartphone-based geolocation of internet hosts. Computer Networks 116 (2017) 22-32.

[19] Cui, Y., An, R., Kartik, B, and Ariyur. 2015. Cellphone Geolocation Via Magnetic Mapping. Automatica (2015) 70-79.

[20] Gustafsson, F. Geolocation-Maps, Measeurements, Models, and Methods. Division of Automatic Control, Department of Electrical Engineering, Link&Ping University, Sweden.

[21] Dodsworth, E and Nicholson, A. 2012. Academic Uses of Google Earth and Google Maps in a Library Setting. Information Technology And Libraries.

[22] Kiraly, A and Abonyi, J. 2015. Redesign of the supply of mobile mechanics based on a novel genetic optimization algorithm using Google Maps API. Engineering Applications of Artificial Intelligence 38 (2015) 122-130.

[23] Yang, S.Y and Hsu, C.L., 2015. A location-based services and Google maps-based information master system for tour guiding, Computers and Electrical Engineering 000 (2015) 1-19.

[24] Cantone, D., and Faro, S. 2004. Two-Levels-Greedy: a generalization of Dijkstra's shortest path algorithm. Electronic Notes in Discrete Mathematics 17 (2004) 81-86.

[25] Fitro, A., Bachri, S.O., Purnomo, A.I.S and Frendianata, I. Shortest Path Finding in Geographical Information Systems using Node Combination and Dijkstra Algorithm, International Journal of Mechanical Engineering and Technology 9 (2), 2018. pp. 755–760.