

**BASIS PATH TESTING OF ITERATIVE DEEPENING SEARCH
AND HELD-KARP ON PATHFINDING ALGORITHM**

^aI Gede Surya Rahayuda, ^bNi Putu Linda Santiari

^{a,b}STMIK STIKOM Bali, Jl. Raya Puputan Renon No. 86 Denpasar, 80234

E-mail: surya_rahayuda@stikom-bali.ac.id

Abstract

This research is a continuation of previous research, where in previous research discussed about the implementation of Iterative Deepening Search and Held-Karp Method. Both methods are compared based on the results obtained. While in the current study, research is more focused on the analysis of the program flow that has been made. Evaluation is done by using basis path method, there are several processes performed on the method, such as: flowgraph, independent path, cyclomatic complexity and graph matrix. In addition to the evaluation of program flow, evaluation is also done based on program performance. Performance tests are based, time, cpu and memory. Based on the evaluation using the base path, obtained flowgraph structure and independent path different, but obtained the result of cyclomatic complexity and graph matrix calculation of the same between Iterative Deepening Search and Held-Karp method is 4. Based on evaluation in terms of performance, process the program from entering data and until getting the result, the Held-Karp method takes a longer time than the Iterative Deepening Search method. The Iterative Deepening Search method takes 2.7 seconds while the Held-Karp method takes 2.8 seconds.

Keywords: Held-Karp, Iterative Deepening Search, Basis Path, Cyclomatic complexity, Graph Matrix

INTRODUCTION

This research is extended research from previous research, where in previous research more discussed about implementation of Iterative Deepening Search and Held-Karp method [1][2]. In that research also discussed about the comparison of the results of both methods. In this research will be more about the evaluation or analysis of the script program that has been made.

Of the two methods that have been implemented into the program in previous research, will be evaluated using the basis path method [3][4]. The basis path method is one of the white box testing methods, where in the testing process needed to create a flowgraph of the script program and also determine the value of cyclomatic complexity. This test aims to analyze the correctness of the structure of the program created and the performance of the program [5][6].

In addition to testing using the basis path method, evaluation is also done based on the speed of processing the program to get the results, the processor and memory performance required during the program.

Of all the tests performed, then the results of the evaluation data are then compared in the form of tables, so that later can be analyzed about the differences of both methods based on various tests that have been done.

Regarding similar research that has been done, the author will begin to explain about previous research that has been done by the author, where the paper has been published in national seminars and journals, research ever conducted on path search method, the method of Iterative Deepening Search and Held-Karp.

My first research on path search methods, the study was titled "Penerapan Pemrograman Dinamis pada Manajemen Pengiriman Produk Menggunakan Metode Held-Karp". The research is my preliminary study on the use of path search algorithms. In the field of informatics science, various path search methods are called dynamic programming. In that study I chose to use the Held-Karp method used to determine the closest route of a product delivery trip. The method is very appropriately applied to the case of product delivery. Because the Held-Karp method find for the shortest path from the starting point to all of the route point and then back to the starting point. In contrast

to other path search algorithms, where generally the method only looks for the shortest path from the starting point to the destination point. In that research developed Held-Karp method using visual basic language in console based desktop program [7].

Publication of the second phase of research on pathfinding method entitled "Implementasi dan Perbandingan Metode Iterative Deepening Search dan Held-Karp pada Manajemen Pengiriman Produk". The research is a continuation of previous research, whereas in this research developed Held-Karp method on visual based desktop program [8]. In addition to that research also use other pathfinding method, Iterative Deepening Search method, Iterative Deepening Search method is the development of Depth First Search method, where the Iterative Deepening Search method applied a recursive process or repetition if in a case the path is not found, the process will then repeated and done until the paths are actually found. But if the method is applied to the product delivery will have some deficiency, because the Iterative Deepening Search method only finds the path from the starting point to the destination point. To overcome this the authors add some additional scripts for the path from the end point to the starting point, so that will get the total results of travel distance from the starting point to all points and back to the starting point. Both programs are compared by inputting the same test data, with the same test data, the two programs produce different routes, when compared to the results obtained, the Held-Karp method gets better route results than the Iterative Deepening Search method.

PATH FINDING USING ITERATIVE DEEPENING SEARCH AND HELD-KARP METHOD

In the previous research has been discussed and solved two desktop pathfinding programs using Iterative Deepening Search and Held-Karp Method. In the program as depicted in Figure 1, provides 5 pieces of combobox to determine which subscription stores that order gas to be delivered. It is assumed that in the case of product shipments there are 5 orders. Administrator will determines the delivery route of the product using the program and

sends the product according to the route specified in the program, as seen in Figure 2.

The same program is done using the Held-Karp method, as shown in Figure 3. In some cases of product delivery Held-Karp and Iterative Deepening Search get different results. The search route can be seen in Figure 4.

The use of both programs, tested in some cases of product delivery, is the comparison table from the previous study, shown in Table 1. There are 20 product shipments, where each shipment contains 5 product shipping locations.

Table 1. Test Results on Some Data Test

pac..	Start Rute	IDS Rute	dist..	HK Rute	dist..
P1	Jl.Gatot Subroto	Jl.Gatot Subroto	41 Km	Jl.Gatot Subroto	60 Km
	Jl.Mertasari	Jl.Katrangan		Jl.Katrangan	
	Jl.Antasura	Jl.Kebudayaan		Jl.Kebudayaan	
	Jl.Kebudayaan	Jl.Mertasari		Jl.Antasura	
	Jl.Katrangan	Jl.Antasura		Jl.Mertasari	
P2	Jl.Gatot Subroto	Jl.Gatot Subroto	63 Km	Jl.Gatot Subroto	70 Km
	Jl.Pratama	Jl.Mahendradatta		Jl.Mahendradatta	
	Jl.Wr. Supratman	Jl.Pratama		Jl.Raya Kesambi	
	Jl.Raya Kesambi	Jl.Raya Kesambi		Jl.Wr. Supratman	
	Jl.Mahendradatta	Jl.Wr. Supratman		Jl.Pratama	
P3	Jl.Gatot Subroto	Jl.Gatot Subroto	33 Km	Jl.Gatot Subroto	48 Km
	Jl.Tirtanadi	Jl.Merdeka		Jl.Merdeka	
	Jl.Diponegoro	Jl.Tirtanadi		Jl.Kebo Iwa	
	Jl.Kebo Iwa	Jl.Diponegoro		Jl.Diponegoro	
	Jl.Merdeka	Jl.Kebo Iwa		Jl.Tirtanadi	
P4	Jl.Gatot Subroto	Jl.Gatot Subroto	32 Km	Jl.Gatot Subroto	44 Km
	Jl.Katrangan	Jl.Pemuda		Jl.Pemuda	
	Jl.Palapa	Jl.Palapa		Jl.Suradipa	
	Jl.Suradipa	Jl.Katrangan		Jl.Palapa	
	Jl.Pemuda	Jl.Suradipa		Jl.Katrangan	
P5	Jl.Gatot Subroto	Jl.Gatot Subroto	81 Km	Jl.Gatot Subroto	93 Km
	Jl.Mahendradatta	Jl.Siligita		Jl.Siligita	
	Jl.Mertasari	Jl.Mertasari		Jl.Antasura	
	Jl.Antasura	Jl.Antasura		Jl.Mertasari	
	Jl.Siligita	Jl.Mahendradatta		Jl.Mahendradatta	
P6	Jl.Gatot Subroto	Jl.Gatot Subroto	57 Km	Jl.Gatot Subroto	78 Km
	Jl.Merdeka	Jl.Pendidikan		Jl.Pendidikan	
	Jl.Pratama	Jl.Pratama		Jl.Wr. Supratman	
	Jl.Wr. Supratman	Jl.Merdeka		Jl.Pratama	
	Jl.Pendidikan	Jl.Wr. Supratman		Jl.Merdeka	
P7	Jl.Gatot Subroto	Jl.Gatot Subroto	37 Km	Jl.Gatot Subroto	62 Km
	Jl.Pemuda	Jl.Sidakarya		Jl.Sidakarya	
	Jl.Tirtanadi	Jl.Tirtanadi		Jl.Diponegoro	
	Jl.Diponegoro	Jl.Diponegoro		Jl.Tirtanadi	
	Jl.Sidakarya	Jl.Pemuda		Jl.Pemuda	
P8	Jl.Gatot Subroto	Jl.Gatot Subroto	55 Km	Jl.Gatot Subroto	75 Km
	Jl.Siligita	Jl.Raya Pemogan		Jl.Raya Pemogan	
	Jl.Katrangan	Jl.Siligita		Jl.Palapa	
	Jl.Palapa	Jl.Palapa		Jl.Katrangan	
	Jl.Raya Pemogan	Jl.Katrangan		Jl.Siligita	
P9	Jl.Gatot Subroto	Jl.Gatot Subroto	38 Km	Jl.Gatot Subroto	51 Km
	Jl.Pendidikan	Jl.Kebudayaan		Jl.Kebudayaan	
	Jl.Mahendradatta	Jl.Mertasari		Jl.Mertasari	
	Jl.Mertasari	Jl.Pendidikan		Jl.Mahendradatta	
	Jl.Kebudayaan	Jl.Mahendradatta		Jl.Pendidikan	
P10	Jl.Gatot Subroto	Jl.Gatot Subroto	60 Km	Jl.Gatot Subroto	68 Km
	Jl.Sidakarya	Jl.Raya Kesambi		Jl.Raya Kesambi	
	Jl.Merdeka	Jl.Pratama		Jl.Pratama	
	Jl.Pratama	Jl.Sidakarya		Jl.Merdeka	
	Jl.Raya Kesambi	Jl.Merdeka		Jl.Sidakarya	
P11	Jl.Gatot Subroto	Jl.Gatot Subroto	37 Km	Jl.Gatot Subroto	66 Km
	Jl.Raya Pemogan	Jl.Kebo Iwa		Jl.Kebo Iwa	
	Jl.Pemuda	Jl.Pemuda		Jl.Tirtanadi	
	Jl.Tirtanadi	Jl.Tirtanadi		Jl.Pemuda	
	Jl.Kebo Iwa	Jl.Raya Pemogan		Jl.Raya Pemogan	
P12	Jl.Gatot Subroto	Jl.Gatot Subroto	59 Km	Jl.Gatot Subroto	66 Km
	Jl.Kebudayaan	Jl.Suradipa		Jl.Suradipa	
	Jl.Suradipa	Jl.Katrangan		Jl.Katrangan	
	Jl.Siligita	Jl.Kebudayaan		Jl.Siligita	
	Jl.Suradipa	Jl.Siligita		Jl.Kebudayaan	
P13	Jl.Gatot Subroto	Jl.Gatot Subroto	40 Km	Jl.Gatot Subroto	43 Km
	Jl.Raya Kesambi	Jl.Antasura		Jl.Antasura	
	Jl.Pendidikan	Jl.Raya Kesambi		Jl.Mahendradatta	
	Jl.Mahendradatta	Jl.Mahendradatta		Jl.Pendidikan	
	Jl.Antasura	Jl.Pendidikan		Jl.Raya Kesambi	
P14	Jl.Gatot Subroto	Jl.Gatot Subroto	28 Km	Jl.Gatot Subroto	28 Km
	Jl.Kebo Iwa	Jl.Wr. Supratman		Jl.Wr. Supratman	
	Jl.Sidakarya	Jl.Merdeka		Jl.Merdeka	
	Jl.Merdeka	Jl.Sidakarya		Jl.Sidakarya	

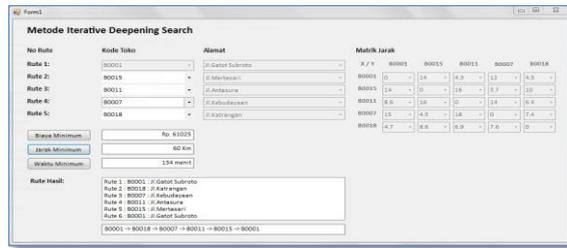


Figure 1. Iterative Deepening Search Program

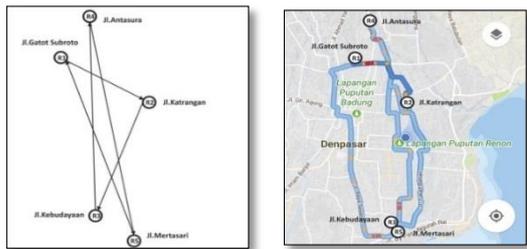


Figure 2. Iterative Deepening Search Route of P1 Delivery on Map

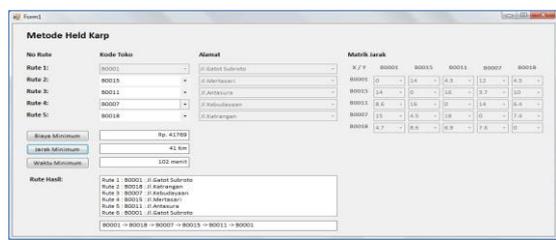


Figure 3. Held-Karp Program

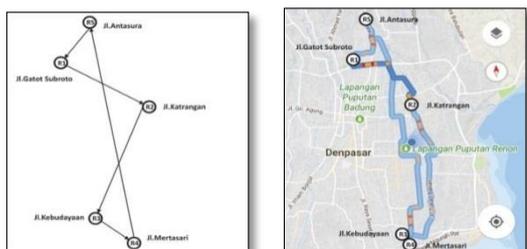


Figure 4. Held-Karp Route of P1 delivery on Map

	Jl.Wr. Supratman	Jl.Kebo Iwa		Jl.Kebo Iwa
	Jl.Gatot Subroto	Jl.Gatot Subroto		Jl.Gatot Subroto
P15	Jl.Gatot Subroto	Jl.Gatot Subroto	35 Km	Jl.Gatot Subroto 35 Km
	Jl.Suradipa	Jl.Diponegoro		Jl.Diponegoro
	Jl.Raya Pemogan	Jl.Pemuda		Jl.Pemuda
	Jl.Pemuda	Jl.Raya Pemogan		Jl.Raya Pemogan
	Jl.Diponegoro	Jl.Suradipa		Jl.Suradipa
	Jl.Gatot Subroto	Jl.Gatot Subroto		Jl.Gatot Subroto
P16	Jl.Gatot Subroto	Jl.Gatot Subroto	65 Km	Jl.Gatot Subroto 69 Km
	Jl.Antasura	Jl.Palapa		Jl.Palapa
	Jl.Kebudayaan	Jl.Kebudayaan		Jl.Siligita
	Jl.Siligita	Jl.Siligita		Jl.Kebudayaan
	Jl.Palapa	Jl.Antasura		Jl.Antasura
	Jl.Gatot Subroto	Jl.Gatot Subroto		Jl.Gatot Subroto
P17	Jl.Gatot Subroto	Jl.Gatot Subroto	42 Km	Jl.Gatot Subroto 42 Km
	Jl.Wr. Supratman	Jl.Mertasari		Jl.Mertasari
	Jl.Raya Kesambi	Jl.Pendidikan		Jl.Pendidikan
	Jl.Pendidikan	Jl.Raya Kesambi		Jl.Raya Kesambi
	Jl.Mertasari	Jl.Wr. Supratman		Jl.Wr. Supratman
	Jl.Gatot Subroto	Jl.Gatot Subroto		Jl.Gatot Subroto
P18	Jl.Gatot Subroto	Jl.Gatot Subroto	27 Km	Jl.Gatot Subroto 37 Km
	Jl.Diponegoro	Jl.Pemuda		Jl.Pemuda
	Jl.Kebo Iwa	Jl.Sidakarya		Jl.Sidakarya
	Jl.Sidakarya	Jl.Diponegoro		Jl.Kebo Iwa
	Jl.Pemuda	Jl.Kebo Iwa		Jl.Diponegoro
	Jl.Gatot Subroto	Jl.Gatot Subroto		Jl.Gatot Subroto
P19	Jl.Gatot Subroto	Jl.Gatot Subroto	67 Km	Jl.Gatot Subroto 80 Km
	Jl.Palapa	Jl.Siligita		Jl.Siligita
	Jl.Suradipa	Jl.Palapa		Jl.Raya Pemogan
	Jl.Raya Pemogan	Jl.Raya Pemogan		Jl.Suradipa
	Jl.Siligita	Jl.Suradipa		Jl.Palapa
	Jl.Gatot Subroto	Jl.Gatot Subroto		Jl.Gatot Subroto
P20	Jl.Gatot Subroto	Jl.Gatot Subroto	50 Km	Jl.Gatot Subroto 59 Km
	Jl.Mertasari	Jl.Pendidikan		Jl.Pendidikan
	Jl.Antasura	Jl.Mertasari		Jl.Antasura
	Jl.Kebudayaan	Jl.Kebudayaan		Jl.Kebudayaan
	Jl.Pendidikan	Jl.Antasura		Jl.Mertasari
	Jl.Gatot Subroto	Jl.Gatot Subroto		Jl.Gatot Subroto

Table 2. Comparison of Both Method on Some Data Test

pac..	Distance HK	Distance IDS	Average	
			Distance	Percentage
P1	41 Km	60 Km	19 Km	32%
P2	63 Km	70 Km	7 Km	10%
P3	33 Km	48 Km	15 Km	31%
P4	32 Km	44 Km	12 Km	27%
P5	81 Km	93 Km	12 Km	13%
P6	57 Km	78 Km	21 Km	27%
P7	37 Km	62 Km	25 Km	40%
P8	55 Km	75 Km	20 Km	27%
P9	38 Km	51 Km	13 Km	25%
P10	60 Km	68 Km	8 Km	12%
P11	37 Km	66 Km	29 Km	44%
P12	59 Km	66 Km	7 Km	11%
P13	40 Km	43 Km	3 Km	7%
P14	28 Km	28 Km	0 Km	0%
P15	35 Km	35 Km	0 Km	0%
P16	65 Km	69 Km	4 Km	6%
P17	42 Km	42 Km	0 Km	0%
P18	27 Km	37 Km	10 Km	27%
P19	67 Km	80 Km	13 Km	16%
P20	50 Km	59 Km	9 Km	15%

Regarding the difference of path search result from Held-Karp method and Iterative Deepening Search method can be seen in Table 2. In the table it can be seen that there is not much difference between the two methods. The greatest distance difference is seen in the delivery of P11 about 43%. The lowest distance difference is seen in the delivery of P14, P15 and P17, there is a difference of 0, or on the delivery both methods get the same result. The Iterative Deepening Search and Held-Karp

methods determine the same path on the transmission. The mean difference of the two methods based on the distance is 11.35 km. Overall the Held-Karp method gets better results from the Iterative Deepening Search method, see Table 2, the total overall difference of the two methods is 19.66%.

BASIS PATH TESTING

The evaluation was performed using the basis path method [9], the processes performed on the basis path method, as seen in Figure 5, are:

1. Flowgraph
2. Independent Path
3. Cyclomatic Complexity
4. Graph Matrix

In addition to using the base path method, evaluation is also performed based on the performance of the computer when the program is run [10], some performance tests performed are:

1. Speed Performance
2. CPU Performance
3. Memory Performance

After basis path testing and performance then next do the comparison of test results that have been done.

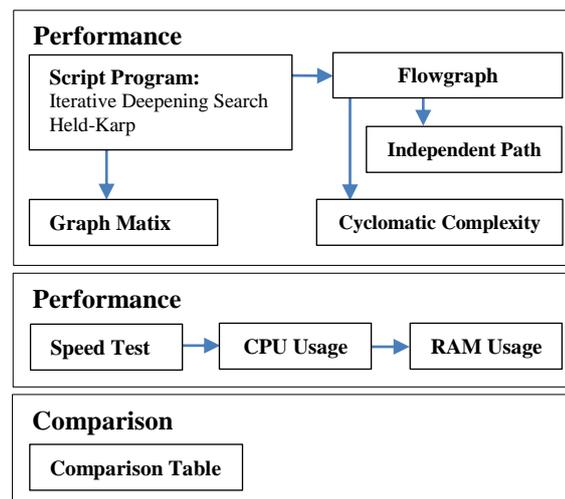


Figure 5. Research Method

RESULT AND DISCUSSION

Iterative Deepening Search

Iterative deepening search or often referred to as iterative deepening depth first search is

one of the pathfinding method which is the development of depth first search method [6][11].

The first version of the depth first search method was discovered around 1900, by the french mathematical scientist Charles Pierre Tremaux as a strategy to overcome the problem of searching the path on the labyrinth. Expressed from several sources, that the logic of the Depth First Search method is to do a preliminary search of the deepest boundary, or searched from the end of a graph, from the last graph sought the smallest number until it is found, then proceeded to the previous boundary and done the same thing, initial limit [12][13].

Iterative deepening search method is the development of depth first search method. Where in the Iterative deepening search method adds several possible repetitions on each search until a route is found. In contrast to the depth first search method, in the depth first search method there is a possibility that the route can not be found, whereas in the Iterative deepening depth first search method if in a certain depth the route is not found, it will be repeated by looking back for the second smallest number at the depth, the next limit, and so on until the route is found [14][15].

Iterative deepening search method has been implemented in previous research [6][16], the flowgraph will be analyzed in order to be evaluated.

Flowgraph Iterative Deepening Search

From the script program is then analyzed and recreated in the form of flowgraph, so that later can be calculated cyclomatic complexity and searched independent path [15][17], program flow from Iterative Deepening Search is:

- a. Initialize the point list as many as the number of points, and enter each point into the list
- b. Initialize the path list according to the available data
- c. Specify the start point of the search path
- d. Perform percountan process starting from the depth or first depth to the limit or maximum limit of depth
- e. For each pre-defined limit, Perform a path search counter through all available points, an explanation of this function will be

described in the following table (points 5a - 5e)

- e.1 Enter the starting point as the first point to look for the path
- e.2 Then enter the other points in sequence from the last point to the first point
- e.3 Initialize the depth or depth variable to know the depth at the time of search path
- e.4 Enter the first point into the path to be discounted
- e.5 Perform percountan as long as there are still points that have not been mounted (points e.5.1 - e.5.4)
 - e.5.1 Enter the next point into the current discount path
 - e.5.2 Count depth at the point being counter, If the depth value is 0, then give the initial value of depth, besides it subtract the depth value by 1
 - e.5.3 If there is a path between the point before and the next point, and all points have been mounted all or the number of points that are being mounted has not exceeded the limit or maximum limit of depth, then check whether all points have been mounted all, If the condition is true, then stop percountan
 - e.5.4 If the above conditions are not met, then do the following percountan (points 5e4a - 5e4g)
 - e.5.4.1 Show error message if the number of points being discount has exceeded the limit or maximum limit of depth
 - e.5.4.2 Show error message if there is no path between point before and point next
 - e.5.4.3 Reduce depth in the position of the counter points for each remaining point that is not mounted
 - e.5.4.4 And for the rest of the point, give the depth value to 0
 - e.5.4.5 For all points with a depth value of 0, Eliminate those points from the path that is being mounted, and re-enter the points into the list of points that have been counter
 - e.5.4.6 Determine the next point with the order of Nearby after the last point is mounted, That is the point with the lowest negative index difference
 - e.5.4.7 After finding the Near point, Sort all uneded points besides point Nearby from point last to first point, Then enter this Nearby point in the first sequence before the sequence

Row 1	: 1 - 1 = 0
Row 2	: 1 - 1 = 0
Row 3	: 1 - 1 = 0
Row 4	: 1 - 1 = 0
Row 5	: 2 - 1 = 1
Row 6	: 1 - 1 = 0
Row 7	: 1 - 1 = 0
Row 8	: 1 - 1 = 0
Row 9	: 1 - 1 = 0
Row 10	: 2 - 1 = 1
Row 11	: 1 - 1 = 0
Row 12	: 1 - 1 = 0
Row 13	: 1 - 1 = 0
Row 14	: 2 - 1 = 1
Row 15	: 1 - 1 = 0
Row 16	: 1 - 1 = 0
Row 17	: 1 - 1 = 0
Row 18	: 1 - 1 = 0
Row 19	: 1 - 1 = 0
Row 20	: 1 - 1 = 0
Row 21	: 1 - 1 = 0
Row 22	: <u>1 - 1 = 0</u>
	3 + 1 = 4
	(cyclomatic complexity)

Held-Karp

The Held-Karp method, also called the Bellman Held-Karp algorithm, is a dynamic programming algorithm proposed in 1962 independently by Bellman and Karp to solve the traveling salesman problem (TSP). TSP is an extension of the Hamiltonian circuit problem [21][22].

The problem can be described as finding a city tour of N in a country (assuming all the cities to be visited within reach), the tour must visit every city only once, the tour must return to the starting point and the tour should get the minimum distance. In general, TSP is classified as a symmetrical travel problem (sTSP), asymmetric traveling salesman problem (aTSP), and multi-travel salesman problem (mTSP). MTSP is generally treated as a matter of quiet vehicle routing or not too high [23][24].

Just like the Iterative Deepening Search method, Held-Karp method has also been applied and implemented in previous research [20][25], the flowgraph from the Held-Karp method will be analyzed in order to be evaluated.

Flowgraph Held-Karp

The program flow from Held-Karp is:

- a. Initialize the point list as many as the number of points, and enter each point into the list
- b. Initialize the path list according to the available data
- c. Specify the start point of the search path
- d. Determine whether the invitations that are made only until all the points have been passed, or until the return to the starting point, If the input is yes, then the path sought must be back to the starting point, If the input is not, then the path searched only to all points already passed
- e. Perform a search for the path with the lowest distance, Explanation of the function will be explained in percoutan below (points e.1 - e.3)
 - e.1 Initialize the initial point list alongside the initial point, then enter all the points alongside the initial point into the list
 - e.2 Perform a reply to the lowest distance search on all points, The explanation of the function will be explained in the following table (points e.2.1 - e.2.4)
 - e.2.1 Perform sums on all point purposes
 - e.2.2 Add a branch of all point purposes at the starting point
 - e.2.3 Determine distance path awal dari point awal ke point purpose
 - e.2.4 Determine the initial distance path from the starting point to point purpose
 - e.2.4.1 Create a new point list for all points beside starting point and point purpose
 - e.2.4.2 If the distance path on the point to the new point is more than 0, then add that distance to the initial distance
 - e.3 Find the path of points with the lowest distance that has been found before, Explanation of the function will be explained in percoutan below
- f. Record the final result for each line answer found, Then show the total distance path

Step count number and flowgraph can be seen in Table 5 and Figure 7.

Independent Path Held-Karp

Similar to what happens with Iterative Deepening Search methods, in the Held-Karp

method there are also 4 independent paths was formed [26][27], as:

- Path 1 : 1, 2, 3, 4, 5, 6, 14, 15
- Path 2 : 1, 2, 3, 4, 5, 7, 8, 9, 10, 15
- Path 3 : 1, 2, 3, 4, 5, 7, 11, 12, 15
- Path 4 : 1, 2, 3, 4, 5, 7, 11, 13, 15

Cyclomatic Complexity Held-Karp

After creating the flowgraph and determining the independent path, then cyclomatic complexity can be searched [28]. Here is the process to find the cyclomatic complexity value of the method.

VG : 4
 VG : 17 edge - 15 + 2 = 4
 VG : 3 predicate node + 1 = 4

Graph Matrix Held-Karp

The search for cyclomatic complexity values can also be done by creating a graph matrix [25]. Here is the process of calculating the matrix graph of Iterative Deepening Search algorithm, as seen in Table 6.

Table 5. Held-Karp Program Flow and Point List

a = 1	e = 5	e.2.2 = 9	e.2.4.2 = 13
b = 2	e.1 = 6	e.2.3 = 10	e.3 = 14
c = 3	e.2 = 7	e.2.4 = 11	f = 15
d = 4	e.2.1 = 8	e.2.4.1 = 12	

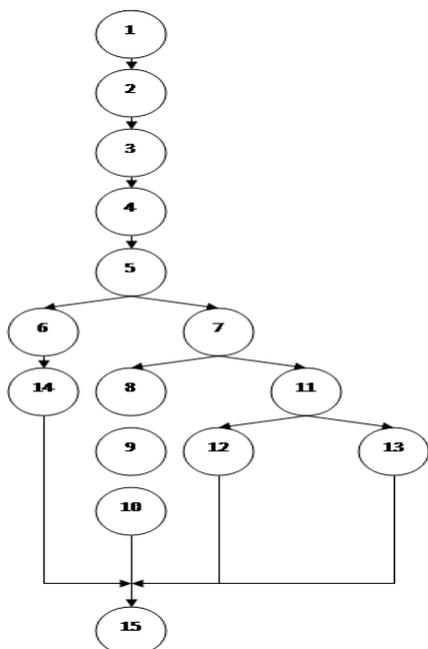


Figure 7. Flowgraph Held-Karp

Table 6. Graph Matrix Held-Karp

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1		1													
2			1												
3				1											
4					1										
5						1	1								
6														1	
7								1			1				
8									1						
9										1					
10															1
11											1	1			
12															1
13															1
14															1
15															

- Row 1 : 1 - 1 = 0
- Row 2 : 1 - 1 = 0
- Row 3 : 1 - 1 = 0
- Row 4 : 1 - 1 = 0
- Row 5 : 2 - 1 = 1
- Row 6 : 1 - 1 = 0
- Row 7 : 2 - 1 = 1
- Row 8 : 1 - 1 = 0
- Row 9 : 1 - 1 = 0
- Row 10 : 1 - 1 = 0
- Row 11 : 2 - 1 = 1
- Row 12 : 1 - 1 = 0
- Row 13 : 1 - 1 = 0
- Row 14 : 1 - 1 = 0
- Row 15 : 1 - 1 = 0
- Row 16 : 1 - 1 = 0
- Row 17 : 1 - 1 = 0
- Row 18 : 1 - 1 = 0
- Row 19 : 1 - 1 = 0
- Row 20 : 1 - 1 = 0
- Row 21 : 1 - 1 = 0
- Row 22 : 1 - 1 = 0

3 + 1 = 4
 (cyclomatic complexity)

Comparison Result

After all the evaluation process is done, then Iterative Deepening Search and Held-Karp methods are also evaluated based on the performance of programs that have been implemented.

Evaluation based on program performance is done by testing, Speed Process, CPU Usage and RAM or Memory Usage. Testing is done by running programs that have been made, with the same sample data input value as the sample data used at the time of implementation.

All data on speed, cpu and ram recorded starting from program run until get result. Here is the comparison result of the test that has been done as seen in Table 7.

Table 7. Comparison of Iterative Deepening Search and Held-Karp Evaluation

	IDS	HK
Independent Path	4	4
C. Complexity	4 (relevan)	4 (relevan)
Graph Matrix	4	4
Open Program		
Time	5,2 s	3,1 s
CPU Usage	15 %	25 %
Memory Usage	6 Mb	1 Mb
Processing Program		
Time	2,7 s	2,8 s
CPU Usage	58 %	65 %
Memory Usage	1 Mb	1 Mb

CONCLUSION

From the evaluation that has been done, it was found that Iterative Deepening Search and Held-Karp method, have different flowgraph images, as well as independent paths, each method has a different path, but when calculated about cyclomatic complexity and Graph Matrix obtained the same result between Iterative Deepening Search and Held-Karp method. Both methods produce the relevant cyclomatic complexity. Based on the

REFERENCES

- [1] I. G. S. Rahayuda and N. P. L. Santiari, "Implementasi dan Perbandingan Metode Iterative Deepening Search dan Held-Karp pada Manajemen Pengiriman Produk," *SISFO*, vol. 7 No. 2, 2018.
- [2] D. Zai, "Simulasi Rute Terpendek Lokasi Pariwisata Di Nias Dengan Metode Breadth First Search Dan Tabu Search," *J InFact*, vol. 1, pp. 30–41, 2016.
- [3] S. Sukiswo, "Evaluasi Kinerja Algoritma Penjadwalan Weighted Round Robin Pada Wimax," *TRANSMISI*, vol. 10, no. 2, pp. 58–64, 2008.
- [4] T. S. Aziz and I. Alfi, "Penerapan Algoritma Alpha Beta Pruning Sebagai Kecerdasan Buatan Pada Game Pawn Battle," Institut Teknologi Sepuluh Nopember, 2016.
- [5] S. R. Wulandari, Y. Purwanto, and B. Irawan, "Evaluasi Algoritma Pencarian Jalur Pada Aplikasi e-iTRIP Guna Menentukan Rute Pariwisata Kota Bandung Berbasis Perangkat Mobile Android," in *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, 2012.
- [6] J. Sophia and N. Rama, "Improving the Proactive Routing Protocol using Depth First Iterative Deepening Spanning Tree in Mobile Ad Hoc Network," *Int. J. Electr. Comput. Eng. IJECE*, vol. 7, no. 1, pp. 316–323, 2017.
- [7] I. G. S. Rahayuda and N. P. L. Santiari, "Penerapan Pemrograman Dinamis Pada Manajemen Pengiriman Produk Menggunakan Metode Held Karp," *E-Proc. KNSI STIKOM Bali*, pp. 513–518, 2017.
- [8] I. G. S. Rahayuda and N. P. L. Santiari, "Implementasi dan Perbandingan Metode Iterative Deepening Search dan Held-Karp pada Manajemen Pengiriman Produk," *J SISFO Inspirasi Prof Sist Inf Inst. Teknol Sepuluh Nop*, vol. 7, 2017.

performance test conducted obtained a fairly varied results, at the time the program starts to run Iterative Deepening Search method takes a longer time and larger memory. Iterative Deepening Search takes 5.2 seconds while Held-Karp method takes 3.1 seconds. If analyzed based on the program process of entering data and until the program gets results, the Held-Karp method takes a longer time than the Iterative Deepening Search method. The Iterative Deepening Search method takes 2.7 seconds while the Held-Karp method takes 2.8 seconds.

ACKNOWLEDGMENT

Thank to my lovely wife and daughter, my family and all of my friends who always give support from the start process of the reseach and until writing articles to publish. Also thank to my institution, STMIK STIKOM Bali, which always give motivation and funding support, thus make me always get spirit to learn on research, and make good publication.

- [9] Y. I.-U. M. Kudus, "Pengujian Sistem Informasi Pengelolaan Pelatihan Kerja Upt. BLK Kabupaten Kudus dengan Metode Whitebox Testing," *Speed-Sentra Penelit. Eng. Dan Edukasi*, vol. 9, no. 3, 2017.
- [10] Selvia Lorena Br Ginting, "Evaluasi Algoritma CB* Untuk Konstruksi Struktur Bayesian Network Dalam Data Mining," presented at the Konferensi Nasional Sistem dan Informatika 2008, Bali, 15 November, vol. Vol. 9, No. 3, pp. 233–239.
- [11] S. Tanaka and K. Tierney, "Solving real-world sized container pre-marshalling problems with an iterative deepening branch-and-bound algorithm," *Eur. J. Oper. Res.*, vol. 264, no. 1, pp. 165–180, 2018.
- [12] M. Naumov, A. Vrieling, and M. Garland, "Parallel Depth-First Search for Directed Acyclic Graphs," in *Proceedings of the Seventh Workshop on Irregular Applications: Architectures and Algorithms*, 2017, p. 4.
- [13] M. M. Asadi, H. Mahboubi, J. Habibi, A. G. Aghdam, and S. Blouin, "Connectivity Assessment of Random Directed Graphs with Application to Underwater Sensor Networks," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 4, pp. 1457–1464, 2017.
- [14] T. S. Aziz and I. Alfi, "Penerapan Algoritma Alpha Beta Pruning Sebagai Kecerdasan Buatan Pada Game Pawn Battle," Institut Teknologi Sepuluh Nopember, 2016.
- [15] J. Pinto and A. Fern, "Learning partial policies to speedup MDP tree search via reduction to IID learning," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 2179–2213, 2017.
- [16] H. Vamja, "Comparative Analysis of Different Path Finding Algorithms to Study Limitations and Progress," *Int J Emerg Technol Adv Eng*, vol. 7, no. 9, pp. 68–75, 2017.
- [17] H. Honainah, "Optimasi Neural Network Untuk Mendeteksi Penyakit Diabetes Dengan Algoritma Genetika," *J. Tek. Inform.*, vol. 6, no. 01, 2017.
- [18] G. Goisque and C. Rapine, "Polynomial time algorithm for multi-level in series capacitated lot-sizing problems with constant batch size," in *International Workshop on Lot Sizing 2016*, 2016.
- [19] R. Palani Kumar, "Internal and External Measures of Code Quality Impact on Refactoring," *SSRG Int. J. Civ. Eng. – ICCREST'17*, pp. 70–72, 2017.
- [20] K. Maheswaran and A. Aloysius, "An Analysis of Object Oriented Complexity Metrics," 2017.
- [21] A. Becker, E. Fox-Epstein, P. N. Klein, and D. Meierfrankenfeld, "Engineering an Approximation Scheme for Traveling Salesman in Planar Graphs," in *LIPICs-Leibniz International Proceedings in Informatics*, 2017, vol. 75.
- [22] C. Chekuri, "Approximating the Held-Karp Bound for Metric TSP in Nearly-Linear Time," *ArXiv Prepr. ArXiv170204307*, 2017.
- [23] M. Zia, Z. Cakir, and D. Z. Seker, "A New Spatial Approach for Efficient Transformation of Equality-Generalized TSP to TSP," in *Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings*, 2017, vol. 17, p. 5.
- [24] D. J. Moylett, N. Linden, and A. Montanaro, "Quantum speedup of the traveling-salesman problem for bounded-degree graphs," *Phys. Rev. A*, vol. 95, no. 3, pp. 1–12, 2016.
- [25] T. Moore, "Implementing the Held-Karp Lower Bound Algorithm in Python," 2015.
- [26] Rohit and Purnendra Kumar, "A Survey for the Quality Assessment of Object Oriented Software," *Int. J. Inf. Futur. Res.*, vol. Vol. 4, No. 9, pp. 7700–7705, 2017.
- [27] C. Chekuri and K. Quanrud, "Approximating the Held-Karp Bound for Metric TSP in Nearly-Linear Time," *ArXiv Prepr. ArXiv170204307*, 2017.
- [28] Niveth Vijay K and Vipin Kumar K S, "Automated Test Path Generation using Genetic Algorithm," *Int. J. Eng. Res. Technol. IJERT*, vol. 6, no. 7, 2017.