

## Application of A\* Algorithm as a Solution for Finding the Shortest Route

Sunjana<sup>1</sup>

<sup>1</sup>Computer Science, Faculty of Engineering, Widyatama University Jln. Cikutra 20124 A, Bandung 40125, INDONESIA

<sup>1</sup>sunjana@widyatama.ac.id

**Article History:** Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

**Abstract:** The application of the path finding algorithm is one way to solve various problems related to the search for the optimal route without obstacles or obstacles. To be able to provide solutions related to the optimal route problems, an algorithm can be used, namely the A\* algorithm. A\* is a route search algorithm that is often used to get the optimal route. By using the A\* algorithm, the optimal route finding problem solving can be done quickly and easily. Algorithm A\* can not only find one or two solutions, but provides the best solution and the optimal route. The simulation was carried out using the Visual Basic programming language. The purpose of this simulation is to find out how the A\* algorithm works in the optimal route search, which is assumed to be the condition of someone looking for a route in a road condition that tends to be congested..

**Keywords:** Optimized route, A Star, Path Finding

### 1. Introduction

Algorithm A\* is one of the methods included in the heuristic search which is commonly used in finding the shortest route. This algorithm is considered as an algorithm that has a high level of efficiency and is optimal compared to other path finding algorithms. This algorithm is a combination of the best first search algorithm (Best First Search) and the Branch and Bound algorithm. Algorithm A\* searches for the fastest distance route from a starting point to an end point. Algorithm A\* combines the value of  $g(n)$  and the value of  $h(n)$  in order to find the lowest weight in obtaining a solution. Algorithm A\* uses the shortest estimated distance to reach the solution and has a heuristic value that is used as a basis for consideration.

To be able to find the end point, you have to try every possible route. The method of resolution will be very easy for humans to solve for simple cases. However, for complex cases it will take a long time to find a solution.

### 2. Theoretical basis

#### A. Path Finding

Path Finding or route search is one of the vital subjects in Artificial Intelligence. Path Finding is widely used to solve a problem in graph. Graph itself is a set of points or nodes that are connected by edges.

This edge is a vector that has a certain magnitude and direction.

To be able to find the route from the initial node to the end node, it is necessary to search for each edge of the graph. Tracing is carried out following the direction of the edge that connects between the nodes of the graph.

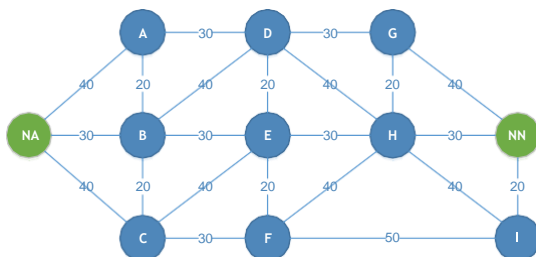


Figure 1. Graph

Figure 1 is an example of a graph with an NA starting node and an NN ending node. NA and NN are connected to each other by edges of different magnitudes. If tracing is performed, it will generate many route combinations that can be traversed to reach the end node. It can be concluded from the graph, each node will provide various solutions to the end node.

#### B. Shortest Path

Shortest Path is an optimization way for path finding. Each route that is produced on path finding will search for the shortest route. The shortest route search is obtained based on the accumulated magnitude of each edge traversed to reach the end node. The accumulation of quantities that have the smallest value is the shortest route from the graph.

### C. Algorithm A\*

The A\* or AStar algorithm is one of the many search algorithms that analyzes input, comparing a number of possible routes to get a solution. This algorithm is widely used in graph traversal and route discovery and route planning processes that can be traversed efficiently around points called nodes.

The basic terminology contained in this algorithm includes the starting point, node, end point, open list, closed list, cost, and obstacle. The basic principle of this algorithm is to find the smallest accumulated value of each route passed to reach the end point or solution (Indartono & Hamidy, 2019).

Algorithm A\* uses the route with the lowest magnitude of each route passed to the node which makes it the best first value search algorithm or best first search.

The basic formula of this algorithm:

$$f(x) = g(x) + h(x)$$

information:

- $f(x)$  is the total cost from the start point to the end point.
- $h(x)$  is a heuristic function used to estimate the distance from the current node to the destination node.
- $g(x)$  is the total distance from the initial node to the current node.

Algorithm A\* uses open list and closed list to locate each node that has been tested. There are three conditions for each successor to be raised, the first is on the open list, the second is on the closed list, and the third is not in between.

In these three conditions, different treatments will be given. If the successor is already on the open list, it is necessary to check whether it is necessary to change the parent or not. The change in the parent depends on the resulting  $g$  value, if the value of  $g$  through the new parent gives a smaller value than the value of the old parent then the parent changes. If the change in parent is made, then update the values  $g$  and  $f$  in the successor. With this update, the successor is more likely to be selected as the best node.

Likewise, successors who are on the closed list need to be tested to determine whether the parent changes or not. If the change in parent is made, the change in the successor's  $g$  and  $f$  values will also apply to all offspring who have been on the open list. This way, all the descendants have a better chance of being selected as the best node.

If the successor is neither on the open list nor on the closed list, then the successor is automatically added to the open list. Then add the successor to the best node successor.

### 3. Research methodology

In this research, we need a guide that will serve as a reference in the decision-making process for solving a problem. The steps in the guide must be structured and systematic so that the resulting solution will be good and clear.

The guidelines are as follows :

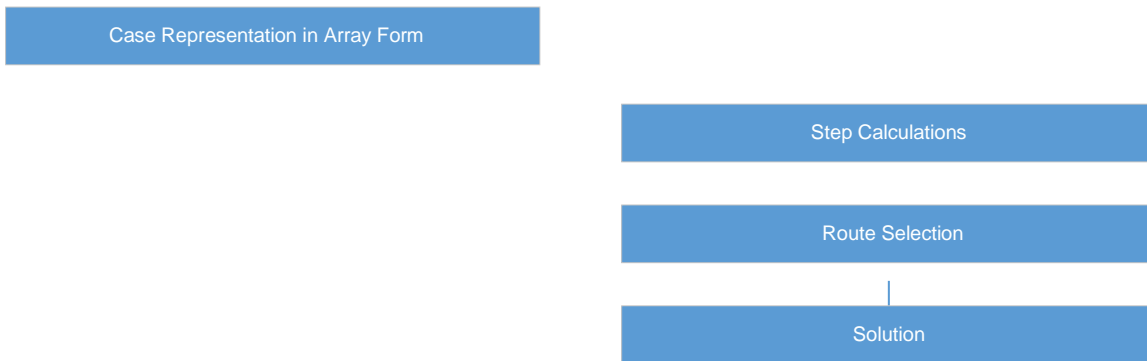
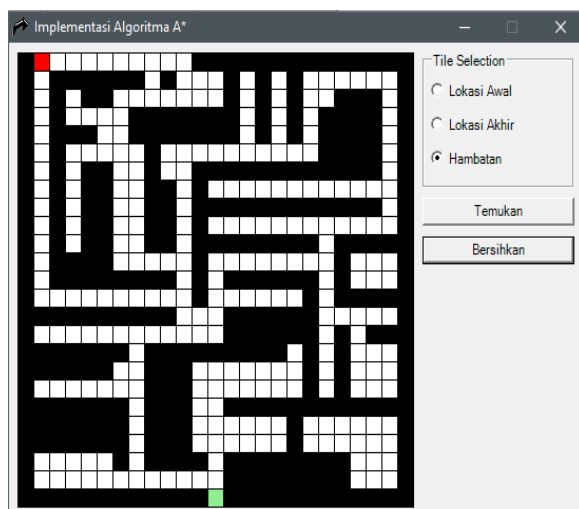


Figure 2. Problem Solving Stage

A. Case Representation in Array Form

A case which is an input in the system to be built will be presented in the form of an array. The representation of the array is done by replacing the elements in a case into a numeric code. In this study the codes used to replace the elements in the case are:

1. Initial Node = 1
2. End Node = 2
3. Obstacle = 0



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25						
1																															
2	40	44	43	42	41	40	39	38																							
3	39	37				35	34	35	36	37	38	39	40	41	35	37	39	40	41	42	43	44									
4	38	36	35	34	33										33	35	37								46						
5	37														32	34	36								47						
6	36	34	33	32	31	30				27	28	29	30	31	32	33	34	35							48						
7	35	35				30	29			26															49						
8	34	36				29	28			25					61	60	59	58	57	56	55	54	53	52	51	50					
9	33	37				28	27			24																51					
10	32	38				27	26			23					63	62	61	60	59	58	57	56	55	54	53	52					
11	31	39				26	25			22																31					
12	30					25	24	23	22	21					23	24	25	26	27	28	29	30									
13	29									20					22											31					
14	28	27	26	25	24	23	22	21	20	19					21	22	23	24	25	26					32						
15															18	19	20								33						
16		20	19	18	17	16	15	14	15	16	17	18	19												34	35	36	37	38		
17															13												35	37			
18															13	12												36	38	39	40
19															17	16	15	14	13	12	11							37	39	40	41
20																															
21																															
22																															
23																															
24																															
25																															

Figure 3. Example Case

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	0	1								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0		0		0	0		0							0	0	0	0							0
3	0		0		0	0										0	0	0				0	0	0	0
4	0		0					0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0
5	0		0	0				0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0
6	0		0					0	0												0	0	0	0	0
7	0		0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0		0	0				0	0	0	0										0	0	0	0	0
9	0		0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0		0	0				0	0	0	0	0									0	0	0	0	0
11	0		0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0		0	0				0	0	0	0	0									0	0	0	0	0
13	0		0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0		0	0				0	0	0	0	0									0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0
16	0		0	0				0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0		0	0				0	0	0	0	0									0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0									0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0									0	0	0	0	0
23	0		0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0		0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4. Case Representation in Array

Step selection starts from the initial node until it reaches the end node. In selecting a step, the route that cannot be traversed will be ignored so that it will only calculate the possible routes that can be traversed. In Figures 3 and 4, there are two possible routes that can be followed, namely the southern route and also the eastern route.

B. Path Counting

Path counting or step calculation is done by calculating the value of  $h(x)$  and  $g(x)$  on neighboring nodes. The value  $h(x)$  is the estimated cost required from the current node to the end node, while  $g(x)$  is the cost required from the current node to the next node..

Figure 5. Value Representation  $h(x)$

In the first step, two possible steps can be generated, namely node AB and node AC, because the cost at node AB is smaller than the cost at node AC, a route through node AB is chosen. In the next step, the neighbor node from AB which has the smallest cost is selected. The second step has only 1 possibility which is BA. To select the node that has the closest distance, it is necessary to calculate the value of  $f(x)$  for node AA and node BB.

Then the calculation is as follows:

$$f(AA) = g(BA) + h(AB)$$

$$f(AA) = 19 + 21$$

$$f(AA) = 40$$

$$f(BB) = g(BA) + h(AC)$$

$$f(BB) = 19 + 25$$

$$f(BB) = 44$$

From the two results above, it can be concluded that which node will be the route, the node with the smallest cost will be selected. In this case node AB is chosen because it has the smallest cost so Node AB will be placed in the open list and node AC will be placed in the closed list. The count of nodes adjacent to the previously selected node will be continued until it reaches the destination node or a deadlock condition occurs.

If you meet a deadlock condition, the search will return to the last node that was registered in the open list and then place it in the closed list. This process will continue until it reaches the destination node.

Figure 6 is the result of the above case calculation using the A\* algorithm.

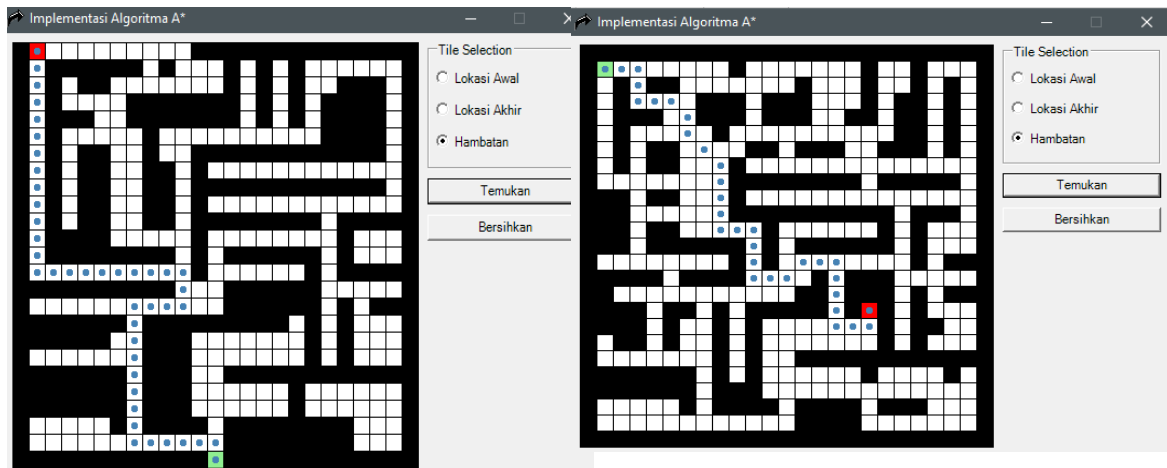


Figure 6. Shortest Route Search Results

#### 4. Testing

This test was carried out using three different case samples with a size of  $25 \times 25$  each. The following are the results of tests carried out on these three samples.

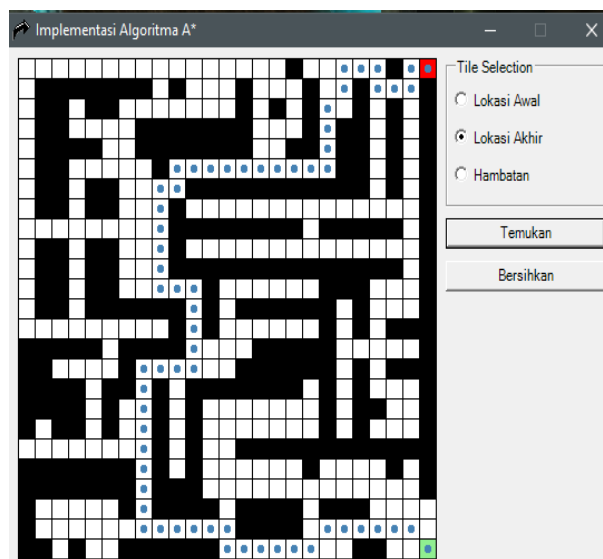


Figure 7. Testing 1

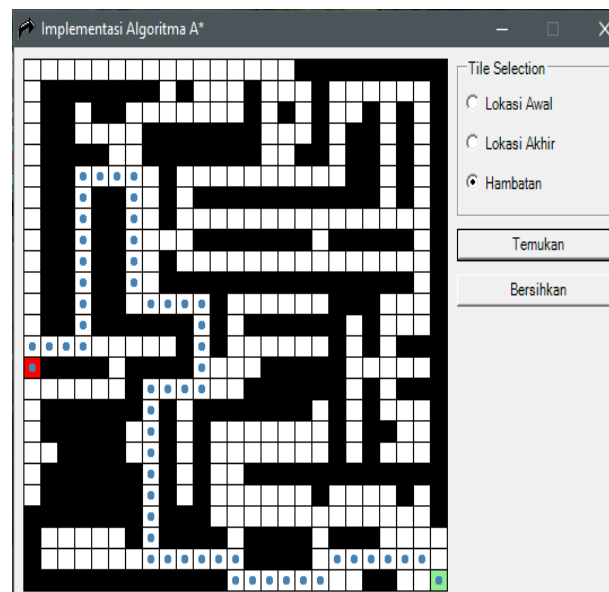


Figure 8. Testing 2

Figure 9. Testing 3

## 5. Conclusion

Based on the results of the analysis and testing that has been carried out on the A \* algorithm, it can be concluded:

1. Algorithm A \* can be applied to various problems such as determining the route to be followed.
2. This test can determine the best route with the obstacles given for each route. From the test results above, the route found is the best route with the smallest  $f(n)$  value compared to other routes.
3. For further development it is recommended to compare the efficiency of the A \* algorithm with other algorithms that are considered more optimal in determining the fastest route.

## 6. References

1. R. Kuniawan, Penerapan Algoritma A\* Sebagai Solusi Rute Terpendek, Medan : Universitas Islam Sumatera Utara, 2016.
2. E. S. Lubis, Sistem Pengantaran Makanan dengan Pendayagunaan Vehicle Menggunakan Geographical Information System (GIS) dan Algoritma A Star (A\*), Medan, 2016.
3. M. E. Ernawati, Implementasi Algoritma A\* Pada Sistem Pendukung Keputusan Untuk Mendiagnosa Penyakit Pada Tanaman Jeruk, Bengkulu : Universitas Bengkulu.
4. Y. Syukriyah and Falahah, Penerapan Algoritma A\* Untuk Mencari Rute Tercepat. :, Bandung: Universitas Widyatama, 2016.
5. M. Yamin and M. Bandrigo, Aplikasi Pencarian Jalur Terpendek Pada Rumah Sakit Umum Bahteramas Menggunakan Algoritma A\*, Kendari: Universitas Halu Uleo.
6. Jabarullah, N. H., Geetha, E., Arun, M., & Vakhnina, V. (2020). Design, analysis, and implementation of a new high step-up DC–DC converter with low input current ripple and ultra-high-voltage conversion ratio. *IET Power Electronics*, 13(15), 3243-3253.
7. H. A. D. K. P. Akshay Kumar Guruji, "Time-Efficient A\* Algorithm for Robot Path Planning," in 3rd International Conference on Innovations in Automation and Mechatronics Engineering, ICIAME 2016, Madhya Pradesh, India, 2016.
8. S. K. Sharma and B.L.Pal, "Shortest Path Searching for Road Network using A\* Algorithm," *JCSMC*, vol. 4, no. 7, pp. 513-522, 2015.
9. E. Dere and A. Durdu, "Usage of the A\* Algorithm to Find the Shortest Path in Transportation Systems," in International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES 2018), Safranbolu – Karabuk, Turkey, 2018.
10. Indartono, S., & Hamidy, A. (2019). The Contribution of Test Type and Curriculum Difference on the Effect of the National Test Score at International Mathematic Test Score: The Challenge of IR 4.0 Curriculum. *Eurasian Journal of Educational Research*, 82, 191-201.