

Implementation of A * Algorithm in Shortest Route Search of Motorcycle Workshop Using Global Positioning System

Hery Sunandar

Informatics Engineering Program
College of Informatics and Computer Management Budi
Darma Medan
Street Sisingamangaraja No. 338 Simpang Limun-Medan,
North Sumatera, Indonesia

Pristiwanto

Informatics Engineering Program
College of Informatics and Computer Management Budi
Darma Medan
Street Sisingamangaraja No. 338 Simpang Limun-Medan,
North Sumatera, Indonesia

Kristian Siregar

Informatics Engineering Program
College of Informatics and Computer Management Budi Darma Medan
Street Sisingamangaraja No. 338 Simpang Limun-Medan, North Sumatera, Indonesia

Abstract - Authorized workshop from Astra Honda Motor supported by experienced technician become the appeal given to the public then it is normal that motorcycle users come to do maintenance and repair of motorcycle. Not many people in the city of Medan know the nearest motorcycle repair shop from where he lived. The need for a nearest garage search journey requires the selection of the shortest path in order to make the time efficient. The A * algorithm is one of the algorithms used to solve problems in the shortest path search. This algorithm is the development of Dijkstra's Edsger and Greddy Best First Search Algorithms. With the application is expected to display a map of the location of honda motorcycle workshop supported by GPS technology (Global Positioning System) will facilitate its users in determining the closest route to honda motorcycle workshop "AHASS" from the residence of consumers.

Keywords: Shortest route, Motorcycle Workshop, A* Algorithm, GPS.

I. INTRODUCTION

A motorcycle repair shop as a motorcycle service service should pay attention to the needs of the community. Motorcycle repair shops serve problems related to motorcycle damage such as oil change, engine repair, and others. The number of motorcycle repair shops located in a city leads consumers selectively to choose a motorcycle repair shop for the repair and maintenance of his motorcycle.

Honda motorcycle workshop "AHASS" is growing rapidly in the service of problems related to motorcycle damage, so it is natural that users of motorcycles from their respective homes come for maintenance and repair of motorcycles to honda motorcycle workshop "AHASS" that exist in the city of Medan, other than that not many people in the city of Medan who knows the existence or address of honda motorcycle repair shop nearest from his residence, other than just know the address of honda motorcycle workshop hence required the closest path that must be taken to get to honda motorcycle repair shop "AHASS" Closest. The need for this journey requires the selection of the shortest path from a

locomotive motorcycle rental workshop so that it can make the efficiency of distance and time.

The A * algorithm is the method used to solve the problem in the shortest route search. This algorithm is the development of Dijkstra's Edsger Algorithm for the shortest route search with the addition of heuristic function to achieve better performance [9]. Therefore, through the design of this application is expected to display a map of the location of honda motorcycle workshop so that more interesting by the public. GPS technology will facilitate users in determining the closest route to honda motorcycle workshop "AHASS".

Based on the background that has been put forward, then the problem can be formulated is how to design a search application for the nearest route honda motorcycle repair shop (AHASS) by using A * based GPS Algorithm.

The problems formulated in the formulation of the above problem, are limited by several things as are: 1). The roads used are protocol roads, not including aisles and aisles, 2). The result data shows the route (path) that is traversed by the closest distance which is completed with the detail of the road and the distance of each road, 3). Application uses GPS to determine user location, 4). Google Maps app to display maps. 5) Applications designed to be created using java programming language.

II. BASIC THEORY

A. Shortest Path

The shortest path is one of the optimization issues. The graph used in the shortest path search is a weighted graph, a graph whose sides are assigned values or weights. The weights on the side of the graph can indicate the distance between cities, the timing of message delivery, the cost of development, and so on. The assumption used is that all weights are positive. The word "teroptimal" is different in meaning depending on the typical matter of minimizing the weights on a path in graph [3].

There are several kinds of shortest path problems, among others:

- The shortest path between two specific nodes.
- The shortest path between all pairs of vertices.
- The shortest path from a particular node to all the other nodes.
- The shortest path between two vertices passing through certain vertices.

B. Search Method

Many ways can be used to build systems that can solve problems as above [6]. Usually to build the system, 4 things to consider are:

- Define the problem appropriately, This definition includes a description of the problem well.
- Analyze the problem and look for some appropriate problem-solving techniques.
- Represents the necessary knowledge to solve the problem.
- Choose the best problem solving technique.

Of these 4 points, the 1st, 2nd and 3rd points have been discussed earlier. For the fourth point, there are also many problem solving techniques that can be used, such as Searching, Reasoning (reasoning); Planning is breaking the problem into smaller sub-issues, solving sub-problems one by one, then combining the solutions of the sub-problems into a complete solution; and Learning, which is a computer program that automatically able to learn and improve its performance through experience. This section deals only with troubleshooting techniques using searching.

Troubleshooting by using searching will be easier if the object is represented as a graph. Graph representation is done first by making the representation of the problem object as a node in the graph and making a representation of relationships between objects with lines connecting the nodes. In addition, each vertex in the graph is visited systematically (traverse) [2]. Basically searching techniques can be divided into 2 (two) large groups, namely blind search (search and blind search) and heuristic search. To measure the performance of search methods, there are four criteria that can be done are:

- Completeness: Does the method guarantee a solution if the solution exists?
- Time Complexity: How long does it take to find the solution?
- Space Complexity: How much memory is needed to find the best solution?
- Optimality: Does the trick method find the best solution if there are several different solutions?

C. Informed Search Algorithm

Informed Search is called Heuristic Search. The search by this algorithm uses specific knowledge because of the problems encountered in addition to the definition of the problem itself. This method is able to find solutions more efficiently than can be done on the uninformed strategy method [2].

The search uses the Uniform Cost Search method (one part of the Uninformed Search Algorithm), which means comparing the value to an existing path and then expanding

the first time on the path with the smallest value. The path value is usually denoted by $g(n)$. Furthermore, with the search method, the Informed Search Algorithm, will be known value estimation (prediction) from one node to another node. The estimated value is usually denoted by $h(n)$. If n is the goal node, then the value of $h(n)$ is zero.

D. Algorithm A* (A Star)

There are many shortest path search algorithms, Dijkstra's algorithm is one of the algorithms [5]. Using the cost function $g(n)$ of each node, Dijkstra's algorithm checks the feasibility of the costs required to reach a node from another node. This process is repeated until the destination node is checked.

Dijkstra's algorithm ensures optimal path gain, but this algorithm has weaknesses. Examination of the knot will be done in all possible directions which eventually all vertices on a graph will be checked. This causes the algorithm to work so slowly that the time it takes to find a solution will get bigger.

The A* algorithm is an algorithm that combines Dijkstra's algorithm and Greedy Best First Search algorithm. In addition to calculating the costs required for the node from one node to another, the A* algorithm also uses heuristic functions to prioritize checking nodes in the right direction so that the A* algorithm has a good time efficiency without sacrificing the actual cost calculation.

E. A* Algorithm in Shortest Route Search

Some best first search strategies differ only in the form of evaluation function used [5]. In the best first search strategy, the selected node to be developed is the node with the lowest evaluation function value and when two paths lead to the same node, the node with the greatest value will be discarded. Mathematically, the evaluation function of a node in the A* algorithm is given by:

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

with:

$f(n)$ = evaluation function.

$g(n)$ = cost of the initial node (start node) to node n .

$h(n)$ = cost estimation from n node to destination node.

The A* algorithm uses two lists of Open List and Closed List. As with other best-first search the two lists have the same function. At first the Open List contains only one node that is the initial node and the Closed List is empty. Note that each node will store its "parent" instructions so that after the search ends the path will also be obtained. Here are the steps of the A* (A Star) algorithm:

1. Enter the initial node into the Open List.
2. Repeat the following steps until the search ends.
 - A. Find the node n with the lowest $f(n)$ value, in the Open List. This node will be the current node.
 - B. Remove the current node from the Open List and enter it into the Closed List.
 - C. For each successor of the current node do the following:
 - If already in the Closed List, ignore it, otherwise continue.

- If not already in Open List, Enter to Open List. Save Current Node as the "parent" of these successors. Save the cost of each node.
 - If already in the Open List, check if the successor's node has a smaller value than the previous successor. If it is smaller, make it a current node and replace "parent" this node.
- D. Although it has reached the destination node, if there is still a successor that has a smaller value, then the node will continue to be selected until the weight is much greater or reaches the final node with a smaller weight than the previous node that has reached the destination node.
- E. In each subsequent node selection, the value of $f(n)$ will be evaluated, and if there is the same $f(n)$ value it will be selected based on the largest value of $g(n)$.

F. Global Positioning System (GPS)

GPS is a satellite navigation system and satellite positioning. Its formal name is NAVSTAR GPS, short for Navigation Satellite Timing and Ranging Global Positioning System. A system that can be used by many people at once in any weather, designed to provide accurate, thorough three-dimensional position, and information about time continuously around the world [7].

Basically GPS consists of three main segments, namely the space segment consists of GPS satellites, the control system segment consists of monitoring stations and satellite controllers, and the user segment consists of GPS users including receivers and signal processing as well as GPS data [10].

III. RESEARCH METHODS

The steps taken to complete this research include:

- Literature review
 Stages to deepen the theory and look for references related to this research. The source of reference comes from articles in the form of journals and books. This stage is very important because it is used to support the next stages in the implementation of this research.
- Needs Analysis
 This stage is to analyze what is the need for this research. Such as data collection, data analysis, and software requirements analysis. This stage is very important to support the design stage of the system.
- System planning
 At this stage, begins the design of the system. ranging from design, system design in order to achieve goals in accordance with the topic of discussion. The results at this stage will continue at the system implementation stage.
- System Creation
 At this stage, the implementation of the design has been prepared in the previous stage according to the concept that has been made. The system can experience a change of concept from the previous design then at this stage will make changes to the system to achieve the expected results by applying the A * algorithm for GPS based location search ..
- System Trial

At this stage checks whether the system has the ability as expected.

- Conclusion Making
 This stage is the final stage after the system has been running as expected to be evaluated and drawing conclusions.

The A * algorithm will be applied to the case analysis of the shortest path to the destination in the application to find the step distance to the destination and the checked node.

In the distance conditions traveled very far and have many obstacles A * algorithm suitable to find the best solution. Any movement done on the status box will be stored in a list. This list will be used to check whether it has ever built that status yet so as not to move the same box repeatedly to the same state. By implementing this strategy, in addition to finding solutions, these algorithms can also find the shortest step to reach a solution. For more details, the workings of the A * algorithm can be seen in figure 1 below.

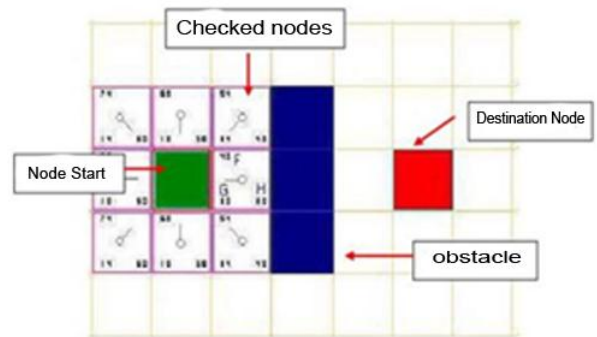


Figure 1. Process Algorithm A *

In figure 1, one (1) green box on the left is the starting node, and the red box on the right is the destination. Three blue squares in the middle are an impassable obstacle. The numbers in each box are the value of f (top left) and h (lower right). The circle and line marks in the center of the box indicate the box parent. For horizontal and vertical movement, its cost is 10, while for diagonal movement cost 14.

In the box to the right of the starting node, g is worth ten and h is 30. The calculation is simple, g is worth ten causes from the initial node to the box only need to move horizontally once. A value of 30 is derived from the number of steps required to reach the goal of the box.

Next, the algorithm will examine the box that has the smallest f -value and develop a path around the box above. With the steps above, this final result is like the figure 2 below.

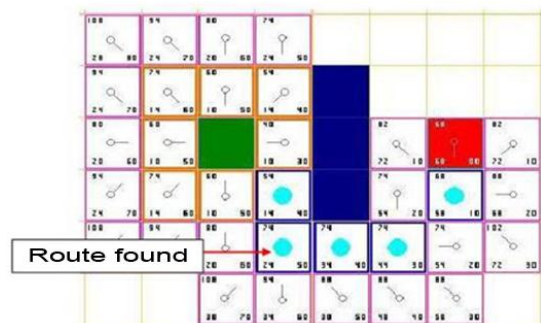


Figure 2. Final Result of algorithm A *

In Figure 2 the route is illustrated with a blue-rectangular square with a light blue circle in the center. So if implemented in the Application pathfinding of a character will pass obstacles and will eventually find the shortest route to the destination.

IV. DISCUSSION

The search application will be boxes with the order X x Y, so that the user can decide for himself the desired order. In the order X. Y box will be used by the starting point and the destination point, the rest to generate a path to look the path that will ultimately determine the shortest path to the destination point.

The maximum barrier to be attached to other orders is as much as the order minus two to place the starting point and destination point. In Figure 3 shows a space (map) with the order 3 x 3 in the Application to be built. Each box represents a node. Each square is connected to the eight closest boxes, meaning each node is connected to another node on the right, left, top-right, bottom-right, bottom-left, and top-left side of the node. The orange and green color boxes are implemented as a barrier, a box that can not be passed by the starting point. Now, it will find the shortest path from the starting point position to the destination point position. Since point A is not directly connected to the point T, it is necessary to pass certain nodes that will eventually deliver to point T with the minimum distance possible.

A. Problem Solving Analysis with Algorithm A* (A Star)

There are several things that need to be defined first in case of Pathfinding Application with the application of A* algorithm. The terms that will be discussed are path, open list, closed list, f, g and n values.

The A* algorithm uses two listing ie OPEN and CLOSED. OPEN is a list used to store the nodes that have been raised and the heuristic value has been calculated but not selected as best node in other words, OPEN contains nodes still have a chance to be selected as the best node, while CLOSED is a list to store the nodes that have been raised and have been selected as the best node. That is, CLOSED contains nodes that may not be selected as the best node (the chances for being selected are already closed).

- OPEN LIST is a list that stores possible paths to be checked. OPEN LIST is sorted by value f. OPEN LIST is used to selectively specify (based on the value of f) a path that is thought to be closer to the destination path. OPEN contains nodes that still have the chance to be selected as the best node.
- CLOSED is a list to store the nodes that have been raised and have been selected as the best node or list that stores the checked path from the open list. This means that CLOSED contains nodes that may not be selected as the best node (the chance to be selected is closed). The two lists (OPEN LIST and CLOSED LIST) are also intended to avoid repeated searching of the identified routes log back into OPEN LIST.
- The F value is the approximate cost of an identified path. The F value is the result of f (n).
- The G value of the function g (n), is the number of steps required to go to the current path.

- Each node must have h (n) value information, ie the estimated value of the node is calculated from the destination node whose result becomes the H value. The function f as the estimation of the evaluation function on the node can be written:

$$F(n) = g(n) + h(n)$$

with:

- f (n) = evaluation function (number of g (n) with h (n))
- g (n) = coordinate distance to a particular destination
- h (n) = cost estimation between coordinates (goal node)

B. Sample Case Algorithm A* (A Star)

The graph used in the shortest path search is a weighted graph, a graph whose sides are assigned values or weights. The weights on the graph can represent the distance between cities. The assumption used is that all weights are positive. To clarify about the A* algorithm, we will give an example problem in Figure 3 below:

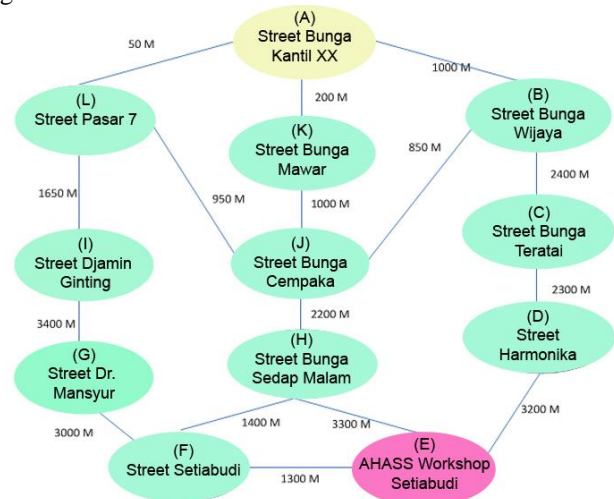


Figure 3. Sample Case Algorithm A* (A Star)

By using the Euclidean distance function (straight line distance from vertex n to vertex G), we get h (n) each vertex in meters (m).

The following is a sequence of steps of the shortest path search process from "Kantil XX Flower Road to AHASS Setia Budi Workshop" using the a* algorithm.

- The first step
 Because on the OPEN list there is only one vertex, namely A. Then A is selected as bestnode and moved to the CLOSED list. Then generated all successors A, namely: B, K and L. Because the three successors are not in the list OPEN maupun CLOSED list, then all three are entered into the list OPEN. This first step produces list OPEN = [B, K, L] and CLOSED = [A].
 Evaluation Function:
 $f(B) = g(A \text{ to } B) + h(A) = 1000 + 0 = 1000m$
 $f(K) = g(A \text{ to } K) + h(A) = 50 + 0 = 50m$
 $f(L) = g(A \text{ to } L) + h(A) = 200 + 0 = 200m$
 From the above calculation can be obtained with the results Algorithm A* The first step with the smallest distance is 50meter.
- Step Two

Next, L with the smallest distance, which is 50 m selected as bestnode and moved to CLOSED. Then, all successors L are generated, namely: J, I and C, since J, I and C have not been in the OPEN list or CLOSED list, then J, I, and C are included in the OPEN list. This second step produces OPEN = [J, I, C] and CLOSED = [A to B, A To L].

Evaluation function:

$$f(J) = g(L \text{ to } J) + h(A \text{ to } L) = 950 + 50 = 1000m$$

$$f(I) = g(L \text{ to } I) + h(A \text{ to } L) = 1650 + 50 = 1700m$$

$$f(C) = g(B \text{ to } C) + h(A \text{ to } B) = 2400 + 1000 = 3400m$$

From the above calculation can be obtained with A* Algorithm is the second step with the smallest distance is 1000 meters.

• Third step

Furthermore, J with the smallest distance, which is 1000 m was selected as bestnode and moved to CLOSED. Then, all J successors are generated, namely: H, G and D, since H, G and D have not been in the OPEN list or CLOSED list, then H, G and D are included in the OPEN list. This third step produces OPEN = [H, G, D] and CLOSED = [L to J, L to I, B to C].

Evaluation function:

$$f(H) = g(J \text{ to } H) + h(L \text{ to } J) = 2200 + 1000 = 3200m$$

$$f(G) = g(I \text{ to } G) + h(L \text{ to } I) = 3400 + 1700 = 5100m$$

$$f(D) = g(C \text{ to } D) + h(B \text{ to } C) = 2300 + 3400 = 5700m$$

From the above calculation can be obtained with the result of A* algorithm is the third step with the smallest distance is 3200 meters.

• Step Four

Next, H with the smallest distance, which is 3200 m was selected as bestnode and moved to CLOSED. Then, all H successors are raised, namely: F, since F has not been on the OPEN list or the CLOSED list, then F is added to the OPEN list. This fourth step produces OPEN = [F] and CLOSED = [I to G, J to H].

Evaluation Function:

$$f(F) = g(G \text{ to } F) + h(I \text{ to } G) = 3000 + 5100 = 8100m$$

$$f(H) = g(H \text{ to } F) + h(J \text{ to } H) = 1400 + 3200 = 4600m$$

From the above calculation can be obtained with the results of Algorithm A* is the fourth step with the smallest distance is 4600 meters.

• Step Five

Next, F with the smallest distance, which is 4600 m was selected as bestnode and moved to CLOSED. Then, all successor F is generated, ie: E, since E has never been in OPEN list or CLOSED list, then E is inserted into OPEN list. This fifth step produces OPEN = [E] and CLOSED = [J to H, H to F].

Evaluation Function:

$$f(E) = g(H \text{ to } E) + h(J \text{ to } H) = 3300 + 3200 = 6500m$$

$$f(F) = g(F \text{ to } E) + h(H \text{ to } F) = 1300 + 4600 = 5900m$$

f with the smallest cost, which is 5900 m selected as bestnode. Because the bestnode is the same as the goal, the solution has been found. In the above case A* generates and stores 9 (Nine) vertices of 12 (twelve) vertices present in the graph. Search results from the above algorithm obtained that the shortest route traversed from Street Bunga Kantil XX to Bengkel AHASS Setia Budi is through Street Pasar 7 - Street

Bunga Cempaka - Street Bunga Sedap Malam- Street Setia Budi - AHASS Setia Budi Workshop, as in figure 4 below:

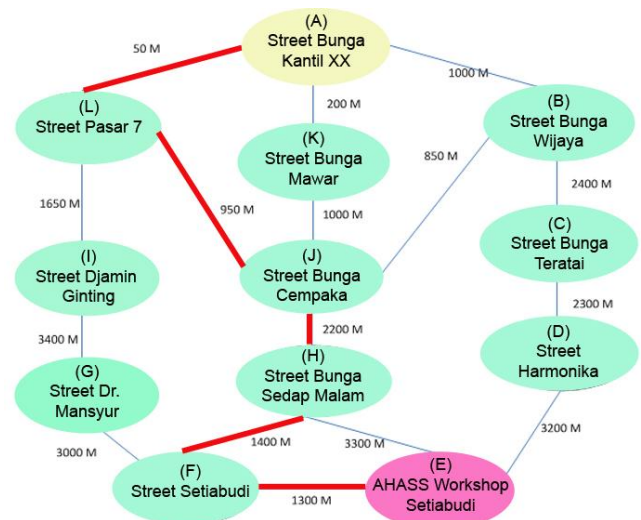


Figure 4. Graph of the Shortest Step Search results

V. CONCLUSIONS AND RESULTS

After applying the A* algorithm on the shortest route search application of AHASS workshop in Medan city, it is concluded as follows:

1. The shortest route search application AHASS repair shop is an application that is used to find the nearest location AHASS motorcycle repair shop located in Medan city.
2. By applying the A* (A Star) algorithm for the shortest route search application AHASS workshop in Medan city is a search method to facilitate society in mencaisolusi best with GPS technology (Global Positioning System) there is distance from point of origin to point of destination.

Application of the shortest route search for AHASS workshop is still able to be developed more system that must be fulfilled in achieving higher system performance stage and better system performance that can use algorithm other than A* which is more optimal in terms of time and node examined and expected for development then the display interface is more interesting

REFERENCES

- [1] Anhar, Guide Mastering PHP & MySQL Autodidak, Mediakita, Jakarta, 2010.
- [2] Budiharto, Widodo. and Derwin S., Artificial Intelligence Concepts and Implementation, Andi Publisher, Yogyakarta, 2014.
- [3] Damiri, R, The shortest route finder application using android-based dijkstra algorithm for mapping hospital in Bandung, Repository Telkom University, 2013, in Press
- [4] Fathansyah, Database, Informatics Bandung, Bandung, 2012.
- [5] Nainggolan, L, Algorithm A* and Greedy Algorithm In Shortest Path Search. Digital Library UPI, 2011, in Press
- [6] Sutojo, T., Artificial Intelligence, Andi Publisher, Yogyakarta, 2011
- [7] Prahasta, E., Spatial Data and Digital Map, Gava Media, Yogyakarta, 2012.
- [8] Suarga, Algorithms & Programming, Andi Publisher, Yogyakarta, 2006,
- [9] Reciter, Hapsari., Application of A-star Algorithm (A*) To Solve Maze Problem, ITB Library, Bandung, 2011, in Press
- [10] Jaizki Mendizabal; Roc Berenguer; Juan Melendez, GPS and Galileo. McGraw Hill. ISBN 978-0-07-159869-9. (2009).