1. Bhakti Yudho SUPRAPTO[1], 2. Suci DWIJAYANTI[1], 3. Muhammad Naufal Ghiffari ISKANDAR[1], 4. Rendyansyah[1], 5. Diah Rahmah DINI[1], 6. Patrick Kusuma WIJAYA[1]

Universitas Sriwijaya (1)
ORCID: 1. 0000-0002-3995-6347; 2. 0000-0003-2060-6408

doi:10.15199/48.2024.08.57

Search for The Best Route on A GPS-Based Autonomous Electric Vehicle Using The A-Star Algorithm

***Abstract**. Route search is critical for autonomous vehicles because the vehicle can decide what path to follow to a destination while driving. Route search can use different algorithms, but the algorithms in previous studies require a long computational time. Therefore, in this study, an autonomous electric vehicle uses the A* algorithm to perform mapping to find the best route with the fastest path to a given destination. The A* algorithm is a shortest route search algorithm that uses a heuristic function to obtain optimal results, and is the most effective algorithm for finding the shortest route using static routing. Based on the results of research conducted on two routes at the Palembang and Inderalaya campuses of Sriwijaya University, the A* algorithm can be used to perform mapping for the best route to a destination using input from the latitude and longitude positions provided by the Global Positioning System (GPS). The shortest route taken in the research results for the Palembang campus is an ABCFG route, while the shortest route on the Inderalaya campus is ABCDGHI. A comparison of the actual distance with the measurement of the distance obtained by the A* algorithm shows a small error of 5.9 m on the Inderalaya campus. These results indicate that the A* algorithm can be used for mapping by autonomous vehicles, enabling the vehicles to determine the best routes.*

***Streszczenie.** Wyszukiwanie trasy ma kluczowe znaczenie w przypadku pojazdów autonomicznych, ponieważ pojazd może podczas jazdy decydować, jaką ścieżką podążać do celu. Wyszukiwanie tras może wykorzystywać różne algorytmy, ale algorytmy z poprzednich badań wymagają długiego czasu obliczeniowego. Dlatego w niniejszym badaniu autonomiczny pojazd elektryczny wykorzystuje algorytm A* do wykonania mapowania w celu znalezienia najlepszej trasy z najszybszą ścieżką do danego miejsca docelowego. Algorytm A* to algorytm wyszukiwania najkrótszej trasy, który wykorzystuje funkcję heurystyczną w celu uzyskania optymalnych wyników i jest najskuteczniejszym algorytmem wyszukiwania najkrótszej trasy przy użyciu routingu statycznego. W oparciu o wyniki badań przeprowadzonych na dwóch trasach w kampusach Palembang i Inderalaya Uniwersytetu Sriwijaya, algorytm A* może zostać wykorzystany do wykonania mapowania najlepszej trasy do miejsca docelowego przy użyciu danych wejściowych z pozycji szerokości i długości geograficznej dostarczonych przez Global Positioning Systemu (GPS). Najkrótsza trasa wybrana w wynikach badań dla kampusu Palembang to trasa ABCFG, natomiast najkrótsza trasa na terenie kampusu Inderalaya to ABCDGHI. Porównanie odległości rzeczywistej z pomiarem odległości uzyskanej za pomocą algorytmu A* wykazuje na terenie kampusu Inderalaya niewielki błąd wynoszący 5,9 m. Wyniki te wskazują, że algorytm A* może być wykorzystany do mapowania przez pojazdy autonomiczne, umożliwiając tym pojazdom wyznaczanie najlepszych tras. (**Wyszukaj najlepszą trasę w autonomicznym pojeździe elektrycznym opartym na GPS, korzystając z algorytmu A-Star**)*

Keywords: A-Star algorithm, Best route, Route search, Unstructured road.
Słowa kluczowe: Algorytm A-Star, najlepsza trasa, wyszukiwanie trasy, droga nieustrukturyzowana.

Introduction

Autonomous vehicles can be classified into several levels, based on the definition of vehicle automation stipulated by the Society of Automotive Engineering, from Level 1, in which a human driver controls most functions, to Level 5, in which the vehicle is fully autonomous [1]. Therefore, an autonomous vehicle may replace some or all human labor required to drive a vehicle via electronic or mechatronic devices [2, 3]. The autonomous vehicle may achieve various mobility gains, such as traffic efficiency, improved mobility rates and mobility patterns, and even a reduction in traffic-related deaths [4]. Autonomous vehicles with automated driving systems include lateral and longitudinal controls, vehicle localization, perception, route planning, and route management. Localization is a crucial function in an autonomous vehicle because the vehicle must be able to locate its position in addition to path finding. Finding the best path is crucial for battery-powered autonomous electric vehicles [5] because the optimal route will optimize battery usage. so that the best route can minimize the total energy consumption and travel time which is a problem in autonomous vehicles[6].

Previous studies have been conducted to develop algorithms for determining the optimal routes. Among these algorithms, the graph search algorithms, such as the A* algorithm and its various enhanced versions, have been extensively researched and implemented because they are simple and relatively fast [7]. Karova et al. [5] used a genetic algorithm to solve different classes of tasks for optimization to find the shortest routes within a maze, and compared the genetic algorithm with the A* algorithm. The results show that genetic algorithms can find the best routes for vehicles. However, the computation time is longer than that of A*. This shows that genetic algorithms have potential but take a long time to make decisions, making them suboptimal for use in autonomous electric vehicles that require short computation time for decision-making. Furthermore, Liu et al [8] proposed a study for path planning using the Djikstra algorithm for the global path and dynamic window approach for the local path. Jichao et al.[9] use a deep learning algorithm to improve the mobile robot motion route planning and obstacle avoidance. Nazari et al. [10] utilized the A* algorithm as a car navigation system to find the shortest path to destinations. Rachmawati and Gustin [11] compared the Djikstra and the A* in finding the shortest path, with the study results showing that the A* algorithm is faster than the Dijkstra algorithm in search time. Another study performed by Yu et al. [12] also used the A* algorithm for obstacle avoidance but had to improve the safety because the A* algorithm typically follows the path it produces too closely and the path is not smooth. Moreover, the A* algorithm is also used in smart manufacturing because of the ability of the A* algorithm in finding the best route[12]–[15].

In addition to the algorithm, another factor that must be considered in finding the best path is determining the position of the vehicle when it is on the road. This can be done using a navigation system, which is usually equipped with a Global Positioning System (GPS) receiver[16, 17].

Based on the studies mentioned, determining the best route and the position of the vehicle are critical factors in autonomous electric vehicles. However, research on the best route search algorithm is still limited and has not been implemented in automatic electric vehicles. Therefore, in this study, search for the best route was developed using the A* algorithm in an autonomous electric vehicle. In contrast to previous studies, which were implemented as simulations of the studied scenarios, this study investigates the shortest path of an autonomous electric vehicle in real environments, where the path is not completely smooth.

This paper is structured as follows: Section 2 represents the related works of this study, Section 3 introduces the materials and methods used, Section 4 presents the results and discussions, and finally, Section 5 presents the conclusion.

Related Work

The In general, the major components of an autonomous vehicle can be divided into perception, communication, and decision[18]. The path planning relates to the decision components. Autonomous vehicles have adopted techniques implemented in mobile robots and modified them to comply with road networks and driving rules[18]. Some studies have proposed determining the best path planning to avoid obstacles and reduce energy consumption by finding the shortest path. In[19], a well-known A* algorithm was used to obtain trajectory planning on structured road maps. This study demonstrated that the A* algorithm performs well in finding trajectory planning through simulation results. However, it was implemented specifically for structured roads in urban areas. On the other hand, [20] focused on studying the A* algorithm for finding global path planning in off-road autonomous driving vehicles. Nevertheless, the A* algorithm was implemented on a grid-based DEM map. Another study introduced a hybrid A* algorithm that combines the A* search engine with Visibility Diagram planning to find the shortest path for valet parking. In contrast to the aforementioned studies, the focus of this paper is to determine the best route for unstructured roads, which are common in rural areas in Indonesia. Additionally, the A* algorithm is implemented in a real environment, accurately representing the actual road conditions to find the best route towards the final destination.

Method

a. The A* Algorithm

For Using the shortest distance, the A* algorithm plots a path from the starting point to the endpoint. The A* algorithm uses the function of f(n) to estimate the distance between nodes, and the distance from one node to another node can be calculated as follows:

$$(1)\ F(n) = g(n) + h(n)$$

where n is the n-th data, f(n) is the total estimated distance from the starting node to the destination node, g(n) is the distance covered to reach the destination node, and h(n) is the estimated value of the distance from the starting node to the destination node. The h(n) element is a heuristic value that can be estimated using a distance measurement, such as the Manhattan distance or Euclidean distance.

This study uses GPS to measure distance based on latitude and longitude coordinates; thus, the Euclidian distance can be calculated as follows.

$$(2)\quad g(n) = \sqrt{(lat_1 - lat_2)^2 + (long_1 - long_2)^2}$$

In
the Eq. (2), 1 degree of Earth is 111,322 km.

Table 1. The parameters of the sensor

```
A* algorithm
lat: latitude from GPS
long: longitude from GPS
g : distance covered to reach the destination
h : the estimated value of the distance from the starting to the
destination
node (n): x, y, v, parent, f
Input: start(n), goal (n)
Output: path
function AStar(start,goal)
   openSet ←empty set
   closedSet ←empty set
   startNode←createNode(lat,long)
   goalNode ← createNode(lat,long)

   startNode.gCost ← 0
   startNode.hCost ← calculateHeuristic(startNode, goalNode)
   startNode.fCost ←startNode.gCost + startNode.hCost

   openSet.add(startNode)

   while openSet is not empty:
      currentNode ← getNodeWithLowestFCost(openSet)

      if currentNode equals goalNode:
         return constructPath(currentNode)

      openSet.remove(currentNode)
      closedSet.add(currentNode)

      for each neighborNode in getNeighbors(currentNode):
         if neighborNode is in closedSet:
            continue
         tentativeGCost ← currentNode.gCost +
calculateDistance(currentNode, neighborNode)

         if neighborNode is not in openSet or tentativeGCost <
neighborNode.gCost:
            neighborNode.gCost ←tentativeGCost
            neighborNode.hCost
←calculateHeuristic(neighborNode, goalNode)
            neighborNode.fCost ←neighborNode.gCost +
neighborNode.hCost
            neighborNode.parent ←currentNode

            if neighborNode is not in openSet:
               openSet.add(neighborNode)
  return null // No path found
function calculateHeuristic(node, goal):
   // Calculate the heuristic value between node and goal
   // This could be the Manhattan distance or Euclidean
distance
   return heuristicValue
function constructPath(node):
   path ←empty list
   current← node
      while current is not null:
      path.prepend(current)
      current ←current.parent
     return path
```

The pseudocode of the A* algorithm can be seen in Table 1. As shown in the table, the heuristic function is used to determine the heuristic value by calculating the distance between latitude and longitude coordinates, as indicated in Eq. (2). The algorithm iteratively selects the node with the lowest fCost from the open set and explores its neighbors. The fCost is the sum of gCost and hCost, as shown in Eq. (1). Once the goal node is reached, the constructPath function is called to backtrack from the goal node to the start node, following the parent references

b. The System Design

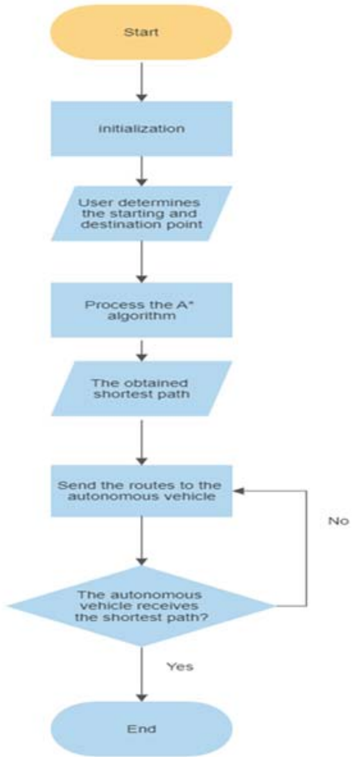


Fig.1. Flowchart of the autonomous vehicle navigation system

Figure 1 presents a flowchart of the system designed in this study and outlines the initial to final stages of the process followed to determine the shortest path. The initial stage in the process performed in this study is to initialize all the devices used, which involves starting the engine of the autonomous vehicle, its computer, and other devices to be used. Next, the user chooses what route to choose and selects a starting and destination point. The computer runs the A* algorithm to generate the shortest route and then sends the generated route to the autonomous vehicle in the form of location coordinates. The autonomous vehicle receives the route from the computer. If the route has been sent and the autonomous vehicle receives the route from the computer, the process is complete. The details of the route generated become inputs for the autonomous vehicle to navigate from the initial position to the destination point. However, if the generated route has not been sent, the system will repeat the computer operation that sends the route to the autonomous vehicle.

c. The System Evaluation

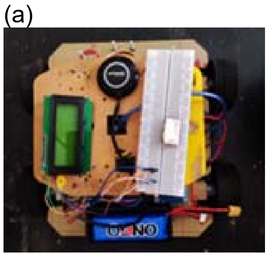
The first step in the test performed in this study was to compare the route generated by the A* algorithm against the route suggested by Google Maps to confirm whether the generated route is the optimal route. The data obtained is then tested using an autonomous electric vehicle. This test evaluates the accuracy and error of the system such that an error value can be obtained from the experiments performed. The error value indicates the error rate of the autonomous electric vehicle in finding the best path and is given as a percentage (%). The experiment was conducted three times, with the initial location and destination point of the autonomous vehicle selected based on the set points generated by the A* algorithm. The experiments were performed at Universitas Sriwijaya in Indonesia. This location was chosen because it has two campuses: Inderalaya and Palembang. These locations were selected for the route generation experiments because both campuses have different characteristics and road network complexities. The autonomous vehicle is operated on campuses

Result and Discussion

a. Road and Data Retrieval

The first route data collection experiment was performed on the Palembang campus of Universitas Sriwijaya, and the data was recorded using a Ublox Neo M8N GPS sensor fitted on the autonomous vehicle prototype (see Figure 2). Data retrieval for the second route was performed using latitude and longitude values from Google Maps. The location for the second data collection experiment was the Indralaya campus. The dataset used in this study takes into account the conditions of the unstructured road as well as the obstacles that can occur on the road.

(a)



(b)

Fig. 2: (a)GPS sensor, and (b)autonomous vehicle prototype

b. Route Design

At the initial stage, seven points of reference were used. These points indicate the route taken by the prototype and are represented by letters A to G, denoting the latitude and longitude positions (Table 2). The position readings were obtained from data read by GPS for routes on the road around the Department of Electrical Engineering, Palembang Campus (the first route). In total, there are seven coordinates that represents the nodes.

Table 2: First route: Latitude and longitude

Coordinate name	Latitude	Longitude
A	−2.984024	104.734252
B	−2.984283	104.734275
C	−2.984644	104.734352
D	−2.984724	104.734359
E	−2.984707	104.734268
F	−2.984658	104.734252
G	−2.984705	104.734169

The latitude and longitude readings are then plotted on a map image, such that the route is traced out, as shown in Figure 3. Each node is then connected, with seven node connections.

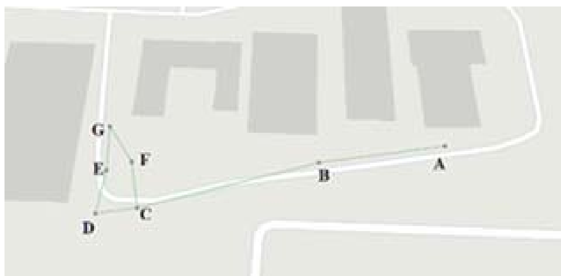


Fig. 3: Mapping of the first route

The second route has its location on the Indralaya campus, with a path from the Faculty of Engineering to the Faculty of Social and Political Sciences. Latitude and longitude data were obtained from Google Maps. Table 3 presents the latitude and longitude for each node of the second route. The mapping of this route is shown in Figure 4. Each set of coordinates are matched to derive the nodes. In total, there were 27 nodes.

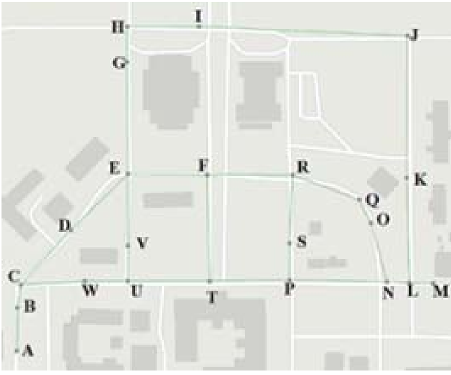


Fig. 4: Mapping of the second route

Table 3: First route: Latitude and longitude

Coordinate name	Latitude	Longitude
A	104.646508	−3.217456
B	104.646513	−3.216977
C	104.646551	−3.216721
D	104.647109	−3.216112
E	104.647742	−3.215489
F	104.648622	−3.215500
G	104.647729	−3.214244
H	104.647735	−3.213909
I	104.648532	−3.213851
J	104.650844	−3.213951
K	104.650835	−3.215532
L	104.650866	−3.216698
M	104.651128	−3.216704
N	104.650618	−3.216689
O	104.650439	−3.216036
P	104.649535	−3.216669
Q	104.650308	−3.215775
R	104.649569	−3.215501
S	104.649535	−3.216258
T	104.648646	−3.216682
U	104.647740	−3.216681
V	104.647742	−3.216287
W	104.647264	−3.216681

c. Route Search

After all the connection points are determined, the next step is to determine the value of the distance between the nodes. These values can be derived using the Euclidean distance equation, expressed as Eq. (2). The calculated distances were then compared to the distances obtained from Google Maps. The distances between each set of connected nodes in the first and second routes are presented in Table 4 and 5, respectively

Table 4: Distances between connected nodes for the first route

Connection	Euc. dis (m)	Man. dis. (m)	Google maps (m)
A→B	28.94586073	31.392804	29
B→C	41.09124082	48.759036	41
C→D	8.939787356	9.685014	8
D→E	11.24076617	12.690708	11
C→F	10.30555561	12.022776	10
E→G	11.0231267	11.243522	11
F→G	10.61827494	14.47186	11

As can be seen in Table 4, the calculated distances obtained using Euclidean distance are remarkably close to the distance readings obtained from Google Maps. However, the distances calculated using the Manhattan distance are longer than the distance readings from Google Maps compared to those obtained using the Euclidean distance with the difference between the Euclidean distance and the distance readings from Google Maps being slight. Similar results are obtained for the second route: the Manhattan distances are longer than the Google Maps distances and are not very accurate when compared to the Euclidean distances, which are closer to the Google Maps distances and more accurate.

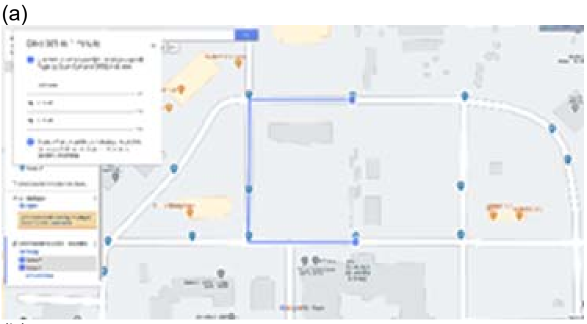
Table 5: Distances between connected nodes for the second route

Connection	Euc. dis (m)	Man. dis. (m)	Google maps (m)
A→B	53.32614298	53.879848	53
B→C	28.81068418	32.728668	29
C→D	91.949883	129.912774	92
C→W	79.49739335	83.825466	81
D→E	98.87110919	139.820432	106
E→F	97.97101309	99.187902	97
E→G	138.6034454	140.043076	138
E→V	88.834956	88.834956	89
F→R	105.4191518	105.533256	105
F→T	131.6097253	134.254332	331(/)
G→H	37.29784584	37.960802	37
H→I	88.95825929	95.18031	89
I→J	257.6170998	268.508664	259
J→K	176.0029337	177.00198	176(182)
K→L	128.9570596	132.361858	128
N→L	29.18760093	30.279584	29
M→L	27.52294077	28.61363836	27
N→P	120.6897017	122.8956036	118
N→O	75.0815056	92.39337564	74(85)
O→Q	31.91390311	42.970292	34
P→S	44.751444	44.751444	46
P→T	98.99555699	101.414342	101
Q→R	87.73960503	112.769186	92
R→S	84.35571001	88.055702	85
T→U	100.8577934	100.969054	100
U→V	43.86143308	44.083512	45
U→W	52.989272	52.989272	53

As can be seen in Table 5, there are several connected points with different distances: Node J to Node K (J→K), and Node N to Node O (N→O). This is because Google Maps automatically records the distance of the route traveled, including at turns, while the maps generated in the experiments are made using Mapbox, with the distance calculation based only on the distance between nodes. The distances at turns still have to be entered into the nodes if you want to calculate.

In addition, Google Maps cannot detect the presence of a path between a pair of connected nodes, i.e., Node F and Node T (see Figure 5a). Hence, the Google Maps distance readings for Node F to Node T is significantly large, at 331 m. This is rather different from the value in the Mapbox map, with a distance of only 131.609 m, due to the assumption in Mapbox that there is a path for a route from Node F to Node T (see Figure 5b). In fact, there is actually a building between Node F and Node T node, which is shown in Figure 5c, but it provides a path between the two nodes.

(a)



(b)

(c)

Fig. 5: (a) Unreadable road on Google Maps, (b) Road in Mapbox, and (c) Actual route from Node F to Node T

The user of an autonomous vehicle can be relieved of the process of finding the best route by using a website. Users can choose the initial location and their destination on a website created for this purpose, and a map showing the best route is presented through the website. The initial display of the website is shown in Figure 6, with the initial display presenting two options in the form of a button to select the first route (Palembang Campus) or the second route (Inderalaya Campus).

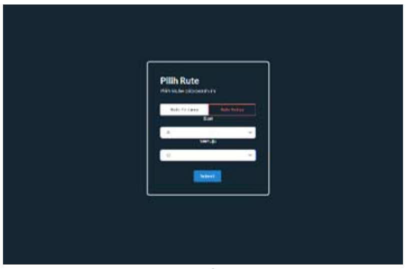


Fig. 6: Initial view of the route selection website

After the user selects the desired route, the user immediately connects to the next page, which is shown in Figure 7 The route presented on this page is the closest route, based on the location selected in the initial display after pressing the submit button. The route is presented on a map, with a blue color denoting the initial node, traversed nodes are indicated by a yellow color, and the red node is the final node. The blue arrow represents the direction of the intended route.

Fig. 7: Generated map of the selected route

d. First Route Evaluation

After the mapping of each route is obtained, the next stage is the evaluation stage of the first route selected. This test was conducted using 8 samples, with the start point and destination point selected to test whether the A* algorithm can find the best route, and whether the results can be presented as a map. The results of the tests for the first route at Palembang Campus are presented in Table 6 using Euclidean distance and Manhattan distance.

Table 6: Evaluation of the shortest path using A* for the first route

Selected route	Acquired routes	Euc. dis (m)	Man. dis. (m)
A→G	A→B→C→F→G	91.84	107.1
B→G	B→C→F→G	62.94	75.7
C→G	C→F→G	21.84	27
D→G	D→E→G	21.3	23.2
F→G	F→G	10.6	14.4
E→G	E→G	11	11.2
D→F	D→C→F	20.14	22.3
A→E	A→B→C→D→E	89.2	101.8

From Table 6, it can be seen that when the point of origin is A and the destination point is G, the best route that can be traveled according to the A* algorithm is A→B→C→D→F→G, with a total distance of 91.84 m. However, when the route chosen is A→B→C→D→E→G, the path covers a longer distance, with the A→B→C→D→E portion of the route covering a distance of 89.2 m and the E→G portion 11 m, such that the total distance for the A→B→C→D→E→G route is 100.2 m. The results of the tests conducted show that the A* algorithm can determine the optimal distance from the point of origin to the destination point. For the route testing using Manhattan distance, the route results are the same as those for the Euclidean distance, but the total distances for Manhattan distance are longer than those for the Euclidean distance.

e. Second Route Evaluation

The second stage of testing route selection was conducted to select the starting point and destination point on the Inderalaya campus area. This test was performed with as many as 10 samples, and the best route obtained was presented as a map. The results of the second route test are presented in Table 7.

If the point of origin is at a point A and the destination is a point J, then according to the A* algorithm, the best route that can be taken is A→B→C→D→E→H→I→J, with a total distance of 795.2 m (Table 6), which is a more optimal path than the A→B→C→W→U→T→P→N→L→K→J route with a distance of approximately 867.3 m. The distance covered by the latter route is a combination of the A→B→C→W→U→T→P→N→L→K route distance of 691.3 m and the 176 m distance of K→J. The test results for this second route show that the A* algorithm can determine the best route that an autonomous vehicle can travel. From Table 7, it can be seen that the distances covered by the routes are derived using the heuristic Euclidean distance and the Manhattan distance. For the fourth test, which is for Node A to Node J, the route result is A→B→C→D→E→H→I→J. Using the Euclidean distance, the distance from Node A to Node J is 795.2 m. However, using the Manhattan distance, the route result is A→B→C→W→ U→T→P→N→L→K→J, with a total distance of 888.2 m. It can be seen that the route result can be different using the Manhattan distance because the distance derived using the Manhattan distance is very different from the actual distance.

Table 7: Evaluation of the shortest path using A* for the second route

Selected route	Acquired routes	Euc. dis (m)	Man. dis. (m)
A→W	A→B→C→W	161.6	170.4
A→V	A→B→C→W→ U→V	258.3	267.3
A→M	A→B→C→W→U →T→P→N→L→ M	591.4	607.3
A→J	A→B→C→D→E →G→H→I→J	795.2	888.2
A→I	A→B→C→D→E →G→H→I	537.6	629.1
A→Q	A→B→C→D→E →F→R→Q	563.8	671
A→R	A→B→C→D→E →F→R	476.1	558.3
A→K	A→B→C→W →U→T→P →N→L→K	691.3	711.2
A→S	A→B→C→W→ U→T→P→S	459.9	459.8
K→J	K→J	176	177

f. Comparison of the best routes of the A* algorithm against Google maps

The next test compares the routes generated by the A* algorithm and the routes recommended on Google Maps. The second route is used for this test, i.e., the route on the Indralaya campus. This test aims to determine whether the results for all nodes traversed by the A* algorithm are the same. In addition, this test also compares the distances for the routes traversed using the A* algorithm and Google Maps. The results of the distance comparison for the second route are presented in Table 8. As can be seen in Table 8, the routes generated by the A* algorithm and the routes recommended on Google Maps are the same, with identical nodes along the length of each route. This shows that the A* algorithm can determine the best route, equivalent to that recommended on Google Maps. Furthermore, the mileages obtained by the A* algorithm and Google Maps have an average difference of 5.95 m. This result is obtained by summing all errors and dividing them by the total number of experiments performed. From Table 7, the A* algorithm has a value that is close to the distance obtained using Google Maps. However, the accuracy achieved using the A* algorithm is more precise, with its ability to calculate distances up to one decimal point. However, the A* algorithm has a disadvantage when the distance travelled includes a turn. For example, for the route from Node A to Node I, the distance based on the A* algorithm is 537.6 m, while the distance on Google Maps is 550 m. This is because Google Maps automatically measures the distance for the route taken, including the turns. However, the map of the route generated by the A* algorithm is made using a Mapbox, in which the distance calculation is based on the distance between nodes, such that the distance at turns must also be entered manually for each node pair if you want it included in the calculations.

Table 8: Comparison of the best route generated by the A* algorithm and that recommended by Google Maps

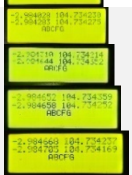
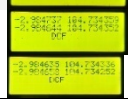
Selected route	Routes traveled	Google maps	A* (m)	Google maps (m)	Diff. (m)
A→W	A-B-C-W	A-B-C-W	161.6	163.1	1.5
A→V	A-B-CW-V	A-B-CW-V	258.3	261	2.7
A→M	A-B-C-W-U-T-P-N-L-M	A-B-C-W-U-T-P-N-L-M	591.4	592	0.6
A→J	A-B-C-D-E-G-H-I-J	A-B-C-D-E-G-H-I-J	795.2	809	13.8
A→I	A-B-C-D-E-G-H-I	A-B-C-D-E-G-H-I	537.6	550	12.4
A→Q	A-B-C-D-E-F-R-Q	A-B-C-D-E-F-R-Q	563.8	574	10.2
A→R	A-B-C-D-E-F-R	A-B-C-D-E-F-R	476.1	482	5.9
A→K	A-B-C-W-U-T-P-N-L-K	A-B-C-W-U-T-P-N-L-K	691.3	691	0.3
A→S	A-B-C-W-U-T-P-S	A-B-C-W-U-T-P-S	459.9	466	6.1
K→J	K-J	K-J	176	182	6

When distance comparison testing was conducted for the first route (Palembang Campus), Google Maps was unable to determine the shortest route and the distance for the route. This is because Google Maps is not able to read the distance for paths covered by buildings or routes that are not on roads that have been mapped by Google Maps (see Figure 5). These results show that the A* algorithm approach is better than Google Maps for measuring and determining the shortest route for routes with portions of the path covered by a building.

g. General System Testing

General system testing was also conducted to determine whether the route obtained is the best route. The test experiment was performed on the Palembang and Indralaya campuses. 'A testing phase was conducted to determine the performance of the A* algorithm and the acceptability of the data generated by the prototype. The results are presented in Table 9.

Table 9: System evaluation for Palembang campus

Route sent	Route view	Act. dis.(m)	A* (m)	Diff. (m)
A→B→C→ D→E		91.6	89.2	2.4
A→B→C→ F→G		92.8	91.8	1
D→C→F		27.9	20.14	7.76

As can be seen in Table 9, the display on the LCD comprises the vehicle position and the transmitted route. This test reveals the difference between the actual distance and the distance of the measurement results of the A* algorithm. The average difference between the actual distance and the distance calculated using the A* algorithm is 3.72 m, which is due to a turn in the actual road taken. However, the distance obtained from the calculation based on A* is close to the actual distance, confirming that the A* algorithm can generate a map with the best route. The routes chosen for this test on the Inderalaya Campus are presented in Fig. 8, in which the red waypoint is obtained directly from GPS sensors, and the blue waypoint was obtained earlier from Google Maps.

Fig. 8: Route chosen for testing on Inderalaya campus

Table 10: System evaluation for Indralaya campus

Route sent	Route view	Act. dis.(m)	A* (m)	Diff. (m)
A→B→C →D→E		310.5	277.5	33
A→B→C →W→U		242.8	237.6	5.2

This test reveals the difference between the actual distance and the distance derived using the measurement results from the A* algorithm. The average difference between the actual distance and the distance calculated based on the A* algorithm is 19.1 m. This is due to a turn in the actual road. However, the distance from the calculations based on A* is close to the actual distance, confirming that the A* algorithm can generate a map with the best route.

h. Avoiding Obstacles and Finding an Alternate Route

```
Rute Yang Diterima : ABCFG
Distance:21CM
Rute Yang Diterima : ABCFG
Distance:20CM
Switching Route
Obstacle in C to F Node
[WiFiEsp] Connecting to siapbot1.com
Sukses
[WiFiEsp] Connecting to siapbot1.com
CDEG
Rute Yang Diterima : CDEG
```

Fig. 9: Alternative routes obtained

Table 11: System evaluation for Indralaya campus

No	Routes requested	Paths obtained	Obstacle	Path obtained after obstacle detection
1	A→G	A→B→C→F →G	C→F	C→D→E→G
2	A→E	A→B→C→ D→E	D→E	D→C→F→G→E
3	A→F	A→B→C→F	C→F	C→D→G→E →F

The A* algorithm is also evaluated to investigate its performance in avoiding obstacles. Fig. 9 shows that the initial generated route was A→B→C→F→G. However, after an obstacle was detected at a distance of 20 cm from the autonomous electric vehicle, it automatically requested a new path beginning at the blocked point. The obstacle was between Node C and Node F, and the alternative path generated for the blocked point onward was C→D→E→G. Three route results with obstacles are presented in Table 11.

Conclusions

Based on the research conducted in this study, it can be concluded that the A* algorithm manages to generate the fastest route, and it can be implemented in autonomous electric vehicles. This is demonstrated by the ability of the A* algorithm to detect routes on the Palembang and Indralaya campuses of Sriwijaya University. The shortest route taken on the Palembang campus is an ABCFG route, with a distance 3,72 meters, while the shortest route on the Indralaya campus is ABCDEGHIJ. Furthermore, the shortest route taken on the Palembang campus for the general testing is an ABCFG route, while the shortest route on the Indralaya campus is ABCWU. This study shows that the distance covered using the A* algorithm compared to the readings on Google Maps has an average error of 6.4 m for the Indralaya campus.

Conclusions

This study was funded through the Directorate of Research, Technology, and Community Service Directorate General of Higher Education, Research, and Technology According to the Research Contract Number: 142/E5/PG.02.00.PT/2022

***Authors**: Dr Bhakti Yudho Suprapto, Electrical Engineering, Jl Raya Palembang-Prabumulih KM 32, Inderalaya, Ogan Ilir, South of Sumatera, Indonesia, E-mail: bhakti@ft.unsri.ac.id (Corresponding Author); Dr Eng. Suci Dwijayanti, Electrical Engineering, Jl Raya Palembang-Prabumulih KM 32, Inderalaya, Ogan Ilir, South of Sumatera, Indonesia, E-mail: sucidwijayanti@ft.unsri.ac.id; Muhammad Naufal Ghiffari Iskandar, Electrical Engineering, Jl Raya Palembang-Prabumulih KM 32, Inderalaya, Ogan Ilir, South of Sumatera, Indonesia, E-mail: nagihifari@gmail.com; Rendyansyah, Electrical Engineering, Jl Raya Palembang-Prabumulih KM 32, Inderalaya, Ogan Ilir, South of Sumatera, Indonesia, E-mail: rendyansyah@ft.unsri.ac.id; Diah Rahmah Dini, Electrical Engineering, Jl Raya Palembang-Prabumulih KM 32, Inderalaya, Ogan Ilir, South of Sumatera, Indonesia, E-mail: diahrahmadini13@gmail.com; Patrick Kusuma Wijaya, Electrical Engineering, Jl Raya Palembang-Prabumulih KM 32, Inderalaya, Ogan Ilir, South of Sumatera, Indonesia, E-mail: patrick_cluvert@yahoo.com;*

REFERENCES

[1] SAE J3016:JAN2014, “Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems,” *Soc. Autom. Eng.*, 2014.

[2] S. E. Shladover, “Connected and automated vehicle systems: Introduction and overview,” *J. Intell. Transp. Syst. Technol. Planning, Oper.*, vol. 22, no. 3, pp. 190–200, 2018, doi: 10.1080/15472450.2017.1336053.

[3] I. s. R. V. Tirumalapudi, Raviteja, “No Title,” *J. Crit. Rev.*, vol. 7, no. 13, pp. 196–202, 2020, doi: 10.31838/jcr.07.13.33.

[4] M. Martínez-Díaz and F. Soriguera, “Autonomous vehicles: Theoretical and practical challenges,” *Transp. Res. Procedia*, vol. 33, pp. 275–282, 2018, doi: 10.1016/j.trpro.2018.10.103.
[5] M. Karova, I. Penev, and N. Kalcheva, “Comparative analysis of algorithms to search for the path in a maze,” *2016 IEEE Int. Black Sea Conf. Commun. Networking, BlackSeaCom 2016*, pp. 0–3, 2017, doi: 10.1109/BlackSeaCom.2016.7901597.
[6] W. Zhou and L. Wang, “The Energy-Efficient Dynamic Route Planning for Electric Vehicles,” *J. Adv. Transp.*, vol. 2019, p. 2607402, 2019, doi: 10.1155/2019/2607402.
[7] S. Erke, D. Bin, N. Yiming, Z. Qi, X. Liang, and Z. Dawei, “An improved A-Star based path planning algorithm for autonomous land vehicles,” *Int. J. Adv. Robot. Syst.*, vol. 17, no. 5, pp. 1–13, 2020, doi: 10.1177/1729881420962263.
[8] L. S. Liu *et al.*, “Path Planning for Smart Car Based on Dijkstra Algorithm and Dynamic Window Approach,” *Wirel. Commun. Mob. Comput.*, vol. 2021, 2021, doi: 10.1155/2021/8881684.
[9] J. Cui and G. Nie, “Motion Route Planning and Obstacle Avoidance Method for Mobile Robot Based on Deep Learning,” *J. Electr. Comput. Eng.*, vol. 2022, p. 5739765, 2022, doi: 10.1155/2022/5739765.
[10] S. Nazari, M. R. Meybodi, M. A. SalehiGh, and S. Taghipour, “An advanced algorithm for finding shortest path in car navigation system,” in *Proceedings - The 1st International Conference on Intelligent Networks and Intelligent Systems, ICINIS 2008*, 2008, pp. 671–674, doi: 10.1109/ICINIS.2008.147.
[11] D. Rachmawati and L. Gustin, “Analysis of Dijkstra’s Algorithm and A∗ Algorithm in Shortest Path Problem,” *J. Phys. Conf. Ser.*, vol. 1566, no. 1, 2020, doi: 10.1088/1742-6596/1566/1/012061.
[12] J. Yu, J. Hou, and G. Chen, “Improved Safety-First A-Star Algorithm for Autonomous Vehicles,” pp. 2–6, 2020.
[13] K. and S. D. B. Y. Suprapto, A. F. Aristz, E. Sean, “Smart manufacturing workplace safety with virtual training, AR and haptic technologies,” in *Human Machine Collaboration and Interaction for Smart Manufacturing*, London: The Institution of Engineering and Technology, 2022.
[14] T. Zheng, Y. Xu, and D. Zheng, “AGV path planning based on improved A-star algorithm,” in *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2019, pp. 1534–1538.
[15] G. Tang, C. Tang, C. Claramunt, X. Hu, and P. Zhou, “Geometric A-Star Algorithm: An Improved A-Star Algorithm for AGV Path Planning in a Port Environment,” *IEEE Access*, vol. 9, pp. 59196–59210, 2021, doi: 10.1109/ACCESS.2021.3070054.
[16] R. Maddison and C. Ni Mhurchu, “Global positioning system: a new opportunity in physical activity measurement,” *Int. J. Behav. Nutr. Phys. Act.*, vol. 6, no. 1, pp. 1–8, 2009.
[17] F. Rovira-Más, “Vulnerability of GPS to provide vehicle states in real time,” *IFAC Proc. Vol.*, vol. 46, no. 18, pp. 207–212, 2013.
[18] D. González, J. Pérez, V. Milanés, and F. Nashashibi, “A Review of Motion Planning Techniques for Automated Vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, 2016, doi: 10.1109/TITS.2015.2498841.
[19] Z. Boroujeni, D. Goehring, F. Ulbrich, D. Neumann, and R. Rojas, “Flexible unit A-star trajectory planning for autonomous vehicles on structured road maps,” *2017 IEEE Int. Conf. Veh. Electron. Safety, ICVES 2017*, pp. 7–12, 2017, doi: 10.1109/ICVES.2017.7991893.
[20] Q. Liu, L. Zhao, Z. Tan, and W. Chen, “Global path planning for autonomous vehicles in off-road environment via an A-star algorithm,” *Int. J. Veh. Auton. Syst.*, vol. 13, no. 4, pp. 330–339, 2017, doi: 10.1504/IJVAS.2017.087148.