Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 3, May 2025, Page 581-590
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i3.5911

Determining the Shortest Route for Eid Homecoming Route Using the Haversine Formula Method and A Star Algorithm

Finsa Nurpandi*, Diny Syarifah Sany

Informatics Engineering Program, Suryakancana University, Cianjur, Indonesia
Email: [1,*]finsa@unsur.ac.id, [2]dinysys@unsur.ac.id
Correspondence Author Email: finsa@unsur.ac.id
Submitted: 12/09/2024; Accepted: 31/05/2025; Published: 31/05/2025

Abstract−Eid homecoming, also known as "Mudik" in Indonesia, is an annual tradition that involves the mass movement of people from cities to their hometowns to celebrate the Eid al-Fitr holiday. Based on data from the Ministry of Transportation of the Republic of Indonesia, there are five major regions that serve as the primary destinations for homecoming travelers: West Java, Central Java, East Java, DI Yogyakarta, and the Jabodetabek region. The Central Java region is estimated to receive the largest number of homecoming travelers, with an estimated 61.6 million people. Given the potential for human movement of up to 193.6 million people during this period, it poses a significant traffic burden, and an efficient determination of the shortest route is crucial. To address this challenge, the study utilizes the Haversine Formula, which calculates the distance between two geographic points on the earth's surface by considering the curvature of the earth. This approach provides a more accurate distance estimate compared to traditional linear distance calculations. Additionally, the A* algorithm is employed to determine the shortest path from the starting point to the destination. The A* algorithm combines the distance calculation results from the Haversine Formula as a heuristic component in the search process, effectively optimizing the route selection. The results of the A* algorithm search identified the shortest route for homecoming travelers, which starts from the city of Jakarta, passes through West Karawang, Indramayu, Cirebon, Tegal, Pekalongan, Semarang, and Salatiga, before reaching the final destination of Klaten. This optimized route covers a total distance of 599.4 km, providing an efficient and cost-effective option for travelers during the Eid homecoming period.

Keywords: Eid Homecoming; Shortest Path; Haversine Formula; A Star Algorithm

1. INTRODUCTION

Eid homecoming is an annual social and cultural tradition in Indonesia, where millions of people journey to their hometowns to celebrate Eid al-Fitr with their families. This tradition leads to a significant surge in traffic volume, often resulting in severe congestion and extended travel times. According to the Indonesian Ministry of Transportation, the 2024 Eid homecoming is projected to involve the movement of 193.6 million people [1], equivalent to 71.7% of Indonesia's population. The main destinations for these homecoming travelers include West Java, Central Java, East Java, DI Yogyakarta, and the Jabodetabek area, with Central Java being the largest, estimated at 61.6 million travelers [2]. Furthermore, it is estimated that around 66.54 million people will use private vehicles, including 35.42 million in private cars and 31.12 million on motorbikes. The significant increase in traffic during this period often leads to severe congestion and prolonged travel times, posing significant challenges for travelers and transportation authorities alike. Therefore, determining the shortest and most efficient route becomes crucial to reduce travel time and alleviate the strain on the transportation network, benefiting both the travelers and the overall flow of traffic.

Navigation technology plays a crucial role in helping drivers find the fastest and most efficient routes during the Eid homecoming period. Algorithms like A* have demonstrated their effectiveness in identifying the shortest paths for various navigation applications. The A* algorithm uses a heuristic approach, leveraging the closest distance estimate to the destination, to determine the next optimal location [3], [4]. By combining this heuristic method with a cost-based search, A* can efficiently calculate the most optimal path from the starting point to the destination. The Haversine formula is employed to determine the heuristic value, which enables more accurate distance calculations between two points on the earth's surface, taking into account the curvature of the earth [5]. This makes the Haversine formula particularly well-suited for navigation applications that require cross-geographic distance estimates.

The process of determining the optimal path typically involves calculating the shortest distance between the origin and destination points based on their coordinate data. This coordinate information is then utilized as input parameters for various algorithmic approaches, including converting it into a heuristic value. One such method for obtaining a heuristic value to guide the search for the shortest route is the Haversine Formula calculation. The Haversine Formula is a commonly used method for calculating the great-circle distance, or the shortest distance between two points on the Earth's surface, taking into account the curvature of the Earth. This formula is particularly useful in applications involving geographic coordinates, such as navigation, mapping, and location-based services. Previous research has explored the practical application of the Haversine Formula, as exemplified by the study on the "Landmark Nusantara" educational game, which utilized the formula to determine the shortest distance between user locations and nearby landmarks [6]. Zelvia Ayu Puspita and colleagues, in their research, applied the Haversine Formula to identify the shortest path for locating hospitals in the South Jakarta area [7], with the aim of making it more convenient for the public to access information about

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 3, May 2025, Page 581-590
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i3.5911

hospital locations, routes, and distances. Similarly, Silvia Kartika and colleagues conducted a study that leveraged the Haversine Formula to determine the shortest route on an Android-based donation application [8], which was designed to facilitate the process of donors finding and accessing donation drop-off locations, thereby simplifying the collection of donated items by fundraisers.

Multiple algorithms are suitable for determining the shortest route, including the A* algorithm. This algorithm evaluates and selects the lowest-cost option at each visited point, incorporating heuristics into the calculation of the cost from the starting point to the current location. Previous research has employed the A* algorithm to identify optimal routes, such as a study by Arief Bramato Wicaksono Putra that examined the shortest path between hospitals in Samarinda [9]. In addition to the A* algorithm, the study also discussed the use of the Floyd-Warshall algorithm, with a comparative analysis of the results. Another relevant study, conducted by Dona Marcelina, focused on the A* algorithm for shortest route determination, utilizing the Euclidean Distance method to establish the necessary heuristic values [3]. This algorithm has been successfully applied in various domains, including transportation, robotics, and video games, where finding the optimal route is crucial. By incorporating heuristic values, the A* algorithm can significantly reduce the search space and computational complexity, making it a popular choice for real-world applications.

This research seeks to integrate heuristic computations employing the Haversine Formula method with the A* algorithm to identify the optimal route for the 2024 Eid homecoming travel. By leveraging real-world geographic data and considering parameters such as distance and heuristics, the aim is to devise a more effective and efficient solution to benefit travelers and alleviate the burden on the transportation network during this peak travel period. The proposed approach aims to provide travelers with the shortest and most optimal routes, reducing travel time and improving the overall flow of traffic during the Eid homecoming tradition. Specifically, this study will utilize the Haversine formula to calculate the accurate distances between various locations, taking into account the Earth's curvature, and then apply the A* algorithm to determine the shortest path that minimizes the total travel time. The integration of these two techniques is expected to result in a more reliable and practical navigation solution for the Eid homecoming travelers, ultimately enhancing their overall experience and contributing to the efficient management of the transportation system during this high-demand period.

2. RESEARCH METHODOLOGY

This research investigates the efficacy and efficiency of the A* Algorithm in identifying optimal routes through directed graph structures. The A* Algorithm is a frequently utilized path-finding technique due to its capacity to integrate heuristic assessments with cost-based search methods. The methodological approach adopted in this study entails the collection and analysis of pertinent geographic data, as well as the implementation and evaluation of the A* Algorithm to determine the most optimal routes for the Eid homecoming travel itinerary.

The research methodology is depicted in a graphical representation provided below, as shown in Figure 1. This figure illustrates the step-by-step process of research implementation, including the literature review, data collection, and the algorithm implementation.

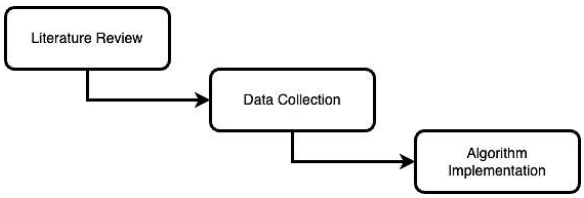

Figure 1. Research Methodology

This paper will provide a comprehensive overview of the research methodology, detailing each step of the process as illustrated in Figure 1. The various components of the methodology, including the application of the A* algorithm and the utilization of the Haversine Formula, will be thoroughly explained to offer readers a clear understanding of the approach taken in this study.

2.1 Literature Review

2.1.1 Heuristic

Heuristics are techniques or methods employed to expedite the process of solving problems by making informed estimates or approximations, rather than conducting an exhaustive search. While these heuristics may not always yield the optimal solution, they can provide a sufficiently good solution within a limited timeframe. In the context of search algorithms, heuristics are used to estimate the lowest cost or distance from a current node to the final destination node. The accuracy of this heuristic significantly impacts the performance of the algorithm, as a more precise heuristic can substantially reduce the number of nodes that must be examined to determine the optimal path. In the domain of path-finding algorithms, such as the A* algorithm, a heuristic function is employed to estimate the lowest cost or distance from a current node to the destination node. To identify the

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 3, May 2025, Page 581-590
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i3.5911

shortest distance solution using the A* algorithm, a heuristic function is required, the value of which does not exceed the total actual cost or distance to the destination [10].

2.1.2 Haversine Formula

The Haversine method is a technique used to compute the great-circle distance, or the shortest distance between two points on the curved surface of the Earth [11], [12]. This formula considers the curvature of the Earth, unlike the simpler Euclidean distance calculation that treats the Earth as flat. By inputting the latitude and longitude coordinates of the two locations, the Haversine formula outputs the most accurate distance between them, accounting for the spherical nature of the planet. This method is particularly useful for longer distances where the Earth's curvature becomes more significant. The Haversine formula was first proposed by James Inman in 1835, building upon the earlier trigonometric approaches developed by James Andrew in 1805 and Josef de Mendoza y Ríos in 1801 for calculating distances on a spherical surface [13].

The Haversine formula simplifies the Earth's shape as a perfect sphere when computing the shortest distance between two points, disregarding the planet's actual ellipsoidal geometry and local terrain features such as hills and valleys [14]. Nevertheless, for most navigation applications, this level of approximation is adequate, and the formula provides a reasonably accurate distance estimate. The mathematical expression for the Haversine formula is as follows [15], [16] :

$$a = sin^2\left(\frac{\Delta lat}{2}\right) + cos(lat_1) \times cos(lat_2) \times sin^2\left(\frac{\Delta lng}{2}\right) \quad (1)$$

$$c = 2 \times arctan2\left(\sqrt{a}, \sqrt{1-a}\right) \quad (2)$$

$$d = R \times c \quad (3)$$

Variable a represents the square of half the chord length between two points on the surface of a sphere. Δlat denotes the difference in latitude between two points, obtained by substracting the latitude value at the initial point from the latitude value at the destination point, which is used to calculate $(lat_2 - lat_1)$. Similarly, Δlng represents the difference in longitude beetween two points, $(lng_2 - lng_1)$. In determining the value of variable a, the formula $sin^2\left(\frac{\Delta lat}{2}\right)$ is employed to measure the contribution of the latitude difference to the distance. Meanwhile, the formula $cos(lat_1) \times cos(lat_2) \times sin^2\left(\frac{\Delta lng}{2}\right)$ is used to measure the contribution of the longitude difference to the distance, taking into account the influence of latitude.

Variable c is the central angle between two points on the surface of the sphere, expressed in radians. This angle represents the angular distance between the two points, which is subsequently converted into a physical distance using the radius of the sphere. The square root of a represents half the chord length, while the complement of the square root of $\sqrt{a}$, denoted as $\sqrt{1-a}$, assists in determining the central angle in a numerically stable manner. The $arctan2\left(\sqrt{a}, \sqrt{1-a}\right)$ function is utilized to compute the angle based on the ratio of these two values. The variable d represents the physical distance between the two points, obtained by multiplying c by the radius of the sphere (R), corresponding to the Earth's radius, which is approximately 6.371 kilometers.

2.1.3 A* Algorithm

The A* algorithm, first introduced in 1968 by researchers Peter Hart, Nils Nilsson, and Bertram Raphael [17], is a best-first search algorithm that leverages a heuristic function in conjunction with actual cost to efficiently determine the minimum total path cost and identify the optimal solution [18]. The core components of the A* algorithm include the initial state, goal state, nodes, open list, closed list, and associated costs [19]. The notation used in the A* algorithm is as follows:

$$f(\mathrm{n}) = g(n) + h(n) \quad (4)$$

The variable $f(\mathrm{n})$ represents the total estimated cost from the starting point to the current node n. The value $g(n)$ corresponds to the actual distance or cost traveled from the starting point to the current node n. Meanwhile, the variable $h(n)$ denotes a heuristic value, which estimates the cost from the current node n to the final destination point. This heuristic value is often calculated as the straight-line distance between the current node and the destination.

The A* algorithm employs two primary lists to organize and manage the nodes during the search process. The OPEN list stores the generated nodes whose estimated costs have been calculated but have not yet been selected as the optimal node. This list contains the potential candidates for the optimal path. Conversely, the CLOSED list holds the nodes that have already been generated and chosen as the best node. These nodes will not be reconsidered during the search, as the algorithm has determined them to be part of the optimal path. By maintaining these two lists, the A* algorithm can efficiently explore the search space, tracking both the unexplored and explored nodes, ultimately identifying the shortest route from the starting point to the destination.

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 3, May 2025, Page 581-590
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i3.5911

Table 1. The pseudo code of the A* algorithm [20]:

```
function A* (problem) returns solution
    OPEN ← S
    CLOSED ← empty array
loop until a goal is found or until there are no nodes in OPEN
    if OPEN = empty then
        Fail
    else
        BestNode = the noe that is in OPEN with f minimum
        Move the best node from OPEN to CLOSED
        if BestNode = goal then
                Success
        else
            Generate all Best Node successors but do not create pointers
            For each successor do:
            Calculate g(cuccessor) = g(BestNode) + actual cost(from Best Node to
successor)
            {Check successor}
            if successor is in OPEN then {has been raised but not yet processed}
                OLD = the node in the same OPEN as the successor
                Add OLD as Best Node successor
                Create a pointer from OLD the Best Node
                Compare the value of g(OLD) with g(suksesor)
                if g(OLD) rather then
                    Change parent OLD to Best Node
                    Change value g and f which is on OLD
                end
            else
                if successor is in CLOSED
                then {has been raised and processed}
                    OLD = the node is the same CLOSED as the successor
                    Add OLD as BestNode successor, compare the value of g(OLD) with
g(successor)
                    if g(OLD) rather then
                        Change parent OLD to BestNode, change the value of g and f
which is on OLD
                        Propagation to all OLD successors by DFS search with the rule:
                        loop until successor node is not in OPEN or node has no successor
                            if successor is in OPEN then
                                Propagation continues
                            else
                                if value og g via successor is better then
                                    Propagation continues
                                else
                                    Propagation stopped
                                end
                            end
                    end
                else {successor is not in OPEN or CLOSED}
                    Enter successor to OPEN, add the successor as the successor of Best
        Node
                    Calculate f=g(successor)+h(successor)
                end
            end
        end
    end
```

2.2 Data Collection

This research study employed a two-pronged approach for data collection. The observational component involved identifying potential travel routes and gathering information on the associated costs for each point along the route. Additionally, the observation process entailed capturing the coordinate points for each city

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 3, May 2025, Page 581-590
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i3.5911

traversed, from the origin city of Jakarta to the destination city of Klaten, utilizing resources provided by the Google Maps application to accurately gather the necessary geographic data. Conversely, the literature review focused on gathering relevant information and background knowledge to support the analysis and implementation of the A* Algorithm in determining the optimal route for the Eid homecoming travel itinerary.

The literature review component of the research methodology entailed a comprehensive search and analysis of relevant textual information and references pertaining to the research topic. These references were predominantly sourced from academic journal articles associated with the subject matter, as well as from official government publications and pertinent books. The purpose of the literature review was to accumulate the necessary background knowledge and provide support for the analysis and implementation of the A* Algorithm in identifying the optimal route for the Eid homecoming travel itinerary.

2.3 Algorithm Implementation

2.3.1 Graph Representation

Graphs are commonly used to represent transportation networks, where nodes represent locations, such as cities, and edges represent the connections or routes between them [17], [21]. In this research, the graph structure is utilized to model the Eid homecoming travel itinerary. Each node in the graph corresponds to a city that travelers may pass through, and the edges connecting these nodes represent the roads or routes between the cities. The edges are assigned weights that correspond to the costs or distances associated with traveling between the connected cities. By representing the travel network as a graph, the A* algorithm can be employed to efficiently determine the shortest route from the starting point, Jakarta, to the destination, Klaten, taking into account the geographic data collected through the observational component of the study.

2.3.2 Heuristic Function

The heuristic function $h(n)$ employed in this research calculates the distance using the Haversine Formula method, which accounts for the curvature of the Earth's surface when working with 2D graphs based on latitude and longitude coordinates. This method provides a more accurate distance estimate compared to simpler Euclidean calculations. To generate the heuristic values, the function requires two input points: the current search point and the final destination point. This heuristic function plays a vital role in guiding the A* algorithm efficiently towards the target by offering a realistic estimate of the remaining distance to the goal. This allows the algorithm to prioritize the most promising paths and explore the search space more effectively.

2.3.3 Searching Process

The A* algorithm initiates its search from the starting node and utilizes a priority queue to evaluate the nodes based on the value of the total cost function. This total cost function is calculated by summing the actual cost of traveling from the starting node to the current node, together with the heuristic value determined through the application of the Haversine Formula. The Haversine Formula offers a more accurate estimate of the remaining distance to the final destination, accounting for the curvature of the Earth's surface, rather than relying on a simpler Euclidean calculation. By incorporating this heuristic value into the A* algorithm's cost function, the search process can prioritize the most promising paths and explore the search space more efficiently towards the target destination.

3. RESULT AND DISCUSSION

3.1 Data Collection

The data collection phase involved utilizing the Google Maps application to gather relevant information. This included identifying the origin city with the highest number of homecoming travelers, which was determined to be Jakarta City. The destination city was selected from the province with the largest percentage of homecoming travelers, namely Central Java Province. Consequently, Klaten City was chosen as a representative sample of the primary destination for the homecoming journey. The collected data encompassed details such as the cities likely to be traversed, the distances between cities, and the latitude and longitude coordinates of the studied locations. The collected data, which provides the city names and corresponding geographic coordinates, are summarized in Table 1 below. The geographic coordinates for each city, including latitude and longitude, are provided in this tabular format to facilitate the subsequent calculations using the Haversine formula and the implementation of the A* algorithm.

Table 1. City Data

No	Code	City	Latitude	Longitude
1	A	Jakarta	-6,175345	106,827269
2	B	Bogor	-6,59849	106,799440
3	C	Sukabumi	-6,921367	106,925354

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 3, May 2025, Page 581-590
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i3.5911

No	Code	City	Latitude	Longitude
4	D	Cianjur	-6,821235	107,140091
5	E	Karawang Barat	-6,322829	107,306323
6	F	Purwakarta	-6,555815	107,442059
7	G	Bandung	-6,902531	107,618538
8	H	Subang	-6,5715318	107,761403
9	I	Indramayu	-6,325797	108,323551
10	J	Sumedang	-6,83927	107,930854
11	K	Cirebon	-6,738232	108,549780
12	L	Nagrek	-7,012668	107,873783
13	M	Garut	-7,216231	107,901425
14	N	Tasik	-7,326328	108,224213
15	O	Banjar	-7,369402	108,541486
16	P	Tegal	-6,86748	109,137881
17	Q	Slawi	-6,979546	109,139211
18	R	Purwokerto	-7,424366	109,230235
19	S	Banjarnegara	-7,442047	109,547465
20	T	Cilacap	-7,727872	109,009478
21	U	Kebumen	-7,668779	109,651663
22	V	Pekalongan	-6,88329	109,665917
23	W	Wonosobo	-7,358523	109,903036
24	X	Purworejo	-7,712768	110,008625
25	Y	Magelang	-7,477176	110,217984
26	Z	Yogyakarta	-7,805336	110,364139
27	AA	Semarang	-6,99032	110,422965
28	AB	Salatiga	-7,330232	110,499512
29	AC	Klaten	-7,740223	110,664498

To complement the geographic data collection, detailed information on the travel distances between the cities along the Eid homecoming route was necessary. This data was obtained by searching for non-toll road routes on Google Maps from the origin city of Jakarta to the destination city of Klaten. Only the primary roads suitable for standard four-wheeled vehicles were considered. The resulting data on the intercity distances, measured in kilometers, has been represented in a two-dimensional graph format to enable the application of the A* Algorithm in determining the optimal travel route. A graphical representation of the complete non-toll road route from Jakarta to Klaten, including all the intermediate cities along the optimal path, is provided below.

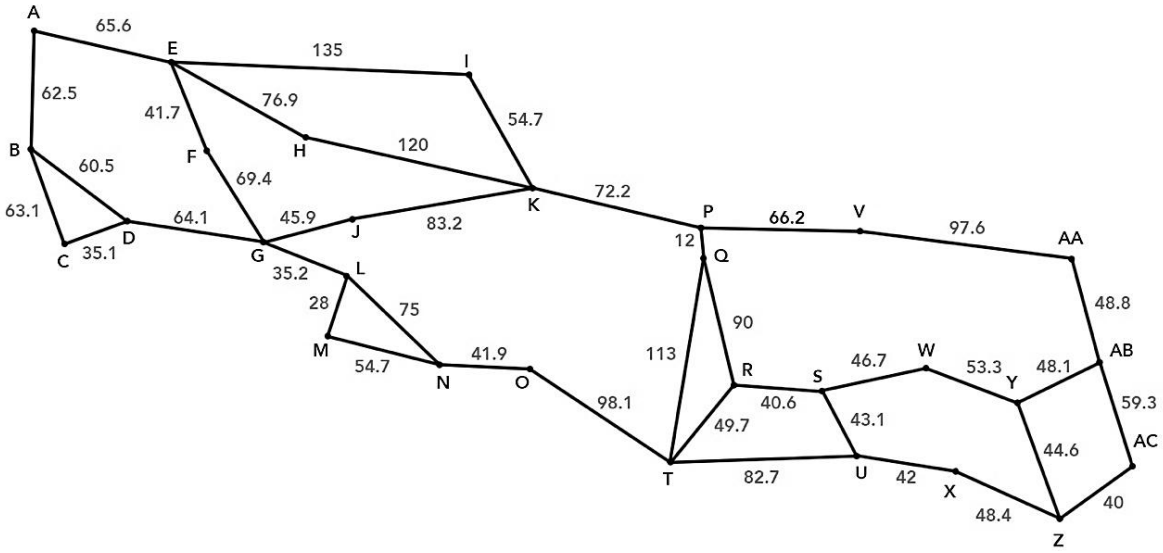

Figure 2. Graph Representation

3.2 Haversine Method Calculation

The Haversine Formula is utilized to calculate the distance between two cities, which in turn generates a Heuristic value for subsequent application in the A* algorithm. The Heuristic value represents the straight-line distance from the origin city to the destination city. To illustrate the Haversine Formula calculation, an example will be provided using the cities of Jakarta as the origin and Klaten as the destination.
Jakarta coordinate points:

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 3, May 2025, Page 581-590
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i3.5911

$lat_1 = -6{,}175345$

$long_1 = 106{,}827269$

Klaten coordinate points (*goal state*):

$lat_2 = -7{,}740223$

$long_2 = 110{,}664498$

Changing the degree of radiance:

$lat_1 = -6{,}175345 \times \frac{\pi}{180} = -6{,}175345 \times 0.0174532925 = -0{,}1077801$ radian

$lat_2 = -7{,}740223 \times \frac{\pi}{180} = -7{,}740223 \times 0.0174532925 = -0{,}1350924$ radian

$long_1 = 106{,}827269 \times \frac{\pi}{180} = 106{,}827269 \times 0.0174532925 = 1{,}86448757$ radian

$long_2 = 110{,}664498 \times \frac{\pi}{180} = 110{,}664498 \times 0.0174532925 = 1{,}93145985$ radian

Finding the values of Δlat and Δlng:

$\Delta lat = (lat_2 - lat_1) = (-0{,}1350924) - (-0{,}1077801) = -0{,}0273122$

$\Delta lng = (lng_2 - lng_1) = 1{,}93145985 - 1{,}86448757 = 0{,}0669723$

Finding the value of a (haversine):

$a = sin^2\left(\frac{\Delta lat}{2}\right) + cos(lat_1) \times cos(lat_2) \times sin^2\left(\frac{\Delta lng}{2}\right)$

$a = sin^2\left(\frac{-0{,}0273122}{2}\right) + cos(-0{,}1077901) \times cos(-0{,}1350924) \times sin^2\left(\frac{0{,}0669723}{2}\right) = 0{,}0012907$

Finding the value of c (central angle):

$c = 2 \times arctan2\left(\sqrt{a}, \sqrt{1-a}\right) = 2 \times arctan2\left(\sqrt{0{,}0012907}, \sqrt{1-0{,}0012907}\right) = 0{,}07186874$

Finding the value of d (distance):

$d = R.c = 6371 \times 0{,}07186874 = 457{,}88\ km$

The Haversine Formula calculation yielded a heuristic value of 457.88 kilometers, representing the straight-line distance between the origin city of Jakarta and the destination city of Klaten. In table 2 below, the complete set of heuristic values for each city, calculated using the Haversine Formula, is provided in kilometers. This table presents the heuristic distance estimates from each city to the final destination of Klaten.

Table 2. Heuristic data for each City

No	Code	City	Heuristic	No	Code	City	Heuristic
1	A	Jakarta	457,88	16	P	Tegal	194,34
2	B	Bogor	444,9	17	Q	Slawi	188,27
3	C	Sukabumi	422,3	18	R	Purwokerto	161,94
4	D	Cianjur	401,94	19	S	Banjarnegara	127,51
5	E	Karawang Barat	402,71	20	T	Cilacap	182,36
6	F	Purwakarta	379,14	21	U	Kebumen	111,89
7	G	Bandung	348,6	22	V	Pekalongan	145,63
8	H	Subang	345,65	23	W	Wonosobo	94,06
9	I	Indramayu	302,45	24	X	Purworejo	72,33
10	J	Sumedang	317,71	25	Y	Magelang	57,25
11	K	Cirebon	258,51	26	Z	Yogyakarta	33,87
12	L	Nagrek	318,2	27	AA	Semarang	87,54
13	M	Garut	310,15	28	AB	Salatiga	49,08
14	N	Tasik	272,91	29	AC	Klaten	0
15	O	Banjar	237,62				

3.3 A* Algorithm Implementation

The A* Algorithm calculation is performed by summing the actual cost of traveling from the origin point to the current point, and the heuristic value calculated using the Haversine Formula method. During each search iteration, the option with the smallest total cost is selected and used as the starting point for the next search. The

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 3, May 2025, Page 581-590
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i3.5911

search commences from the city of Jakarta, represented as Node A, which is the origin for the travelers. The search will terminate when the destination city of Klaten, represented as Node AC, has the lowest value among the remaining nodes in the OPEN List. Examining the graph in Figure 1, Node A is connected to Nodes E and B. The following explanation outlines the first iteration of the search process.
Calculation from Node A to Node B

$g(B) = 62{,}5$

$h(B) = 444{,}9$

Finding value of $f(B)$

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

$f(B) = 62{,}5 + 444{,}9$

$f(B) = 507{,}4$

Calculation from Node A to Node E

$g(E) = 65{,}6$

$h(E) = 402{,}71$

Finding value of $f(E)$

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

$f(E) = 65{,}6 + 402{,}71$

$f(E) = 468{,}31$

In the first iteration, Nodes B and E are added to the OPEN list as they have not been examined and will be searched in subsequent steps. Meanwhile, Node A is added to the CLOSED list since no further searches will be performed on it. The calculations for Nodes B and E show the smallest value is for Node E, with a cost of 468.31. Therefore, the next search will commence from Node E. The table below presents the overall results of the A* algorithm calculation.

Table 3. A* algorithm calculation results

Iteration	Initial Node	Destination Node	*g*(n)	*h*(n)	*f*(n)	OPEN	CLOSED
1	A	B	62,5	444,9	507,4	B, E	A
		E	65,6	402,71	468,31		
2	E	F	107,3	379,14	486,44	B, F, H, I	A, E
		H	142,5	345,65	488,15		
		I	200,6	302,45	503,05		
3	F	G	176,7	348,6	525,3	B, G, H, I	A, E, F
4	H	K	262,5	258,51	521,01	B, G, I, K	A, E, F, H
5	I	K	255,3	258,51	513,81	B, G, K	A, E, F, H, I
6	B	C	125,6	422,3	547,9	C, D, G, K	A, B, E, F, H, I
		D	123	401,94	524,94		
7	K	J	338,5	317,71	656,21	C, D, G, J, P	A, B, E, F, H, K, I
		P	327,5	194,34	521,84		
8	P	Q	339,5	188,27	527,77	C, D, G, J, Q, V	A, B, E, F, H, I, K, P
		V	393,7	145,63	539,33		
9	G	J	222,6	317,71	540,31	C, D, J, L, Q, V	A, B, E, F, H, I, G, K, P
		L	211,9	318,2	530,1		
10	D	C	125,6	422,3	547,9	C, J, L, Q, V	A, B, D, E, F, G, H, I, K, P
		G	176,7	348,6	525,3		
11	Q	T	452,2	182,36	634,59	C, J, L, V, R, T	A, B, D, E, F, G, H, I, K, P,
		R	429,5	161,94	591,44		Q, T
12	L	M	239,9	310,15	550,05	C, J, M, N, R, T,	A, B, D, E, F, G, H, I, K, L,
		N	286,9	272,91	559,81	V	P, Q, T
13	C	D	123	401,94	524,94	J, M, N, R, T, V	A, B, C, D, E, F, G, H, I, K, L, P, Q, T
14	V	AA	491,3	87,54	578,84	J, M, N, R, T, AA	A, B, C, D, E, F, G, H, I, K, L, P, Q, T, V
15	J	K	255,3	258,51	513,81	M, N, R, T, AA	A, B, C, D, E, F, G, H, I, J,

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 3, May 2025, Page 581-590
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i3.5911

Iteration	Initial Node	Destination Node	*g*(n)	*h*(n)	*f*(n)	OPEN	CLOSED
		G	176,7	348,6	525,3		K, L, P, Q, T, V
16	M	N	286,9	272,91	559,81	N, R, T, AA	A, B, C, D, E, F, G, H, I, J, K, L, M, P, Q, T, V
17	N	O	328,8	237,62	566,42	R, T, O, AA	A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, T, V
18	O	T	426,9	182,36	609,26	R, T, AA	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, T, V
19	AA	AB	540,1	49,08	589,18	R, T, AB	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, T, V, AA
20	AB	AC	599,4	0	599,4	R, T, AC	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, T, V, AA, AB
21	R	S	470,1	127,51	597,61	S, T, AC	A, B, C, D, E, F, G, H, I, J,
		T	452,5	182,36	634,86		K, L, M, N, O, P, Q, R, T, V, AA, AB
22	AC	AC	599,4	0	599,4	S, T	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, T, V, AA, AB, AC

The A* algorithm calculation, as presented in Table 3, identifies the optimal route for travelers journeying from Jakarta City to Klaten City. This calculation was performed over the course of 22 iterations, which allowed for the determination of the most efficient path for the Eid homecoming journey. The shortest path is determined to be via the sequence of nodes A-E-I-K-P-V-AA-AB-AC, or the cities of Jakarta - West Karawang - Indramayu - Cirebon - Tegal - Pekalongan - Semarang - Salatiga - Klaten, with a total distance of 599.4 km.

4. CONCLUSION

The findings of this study demonstrate that the A* algorithm, combined with heuristic values calculated using the Haversine Formula, can effectively determine the optimal route for the 2024 Eid homecoming journey. The analysis identified the sequence of cities that travelers can traverse, starting from the origin city of Jakarta and culminating in the destination city of Klaten. Importantly, the determined intercity route is based on actual non-toll roads, providing a cost-effective and efficient option for travelers during the Eid homecoming period. By leveraging the strengths of the A* algorithm and the Haversine Formula, this study presents a practical approach to planning the optimal journey that considers both distance and travel time factors, ultimately enhancing the overall experience for homecoming travelers. The results of the A* algorithm and Haversine Formula analysis identified the optimal route for the Eid homecoming journey as commencing in the origin city of Jakarta, traversing through the intermediate cities of West Karawang, Indramayu, Cirebon, Tegal, Pekalongan, Semarang, and Salatiga, before reaching the final destination of Klaten. This determined route covers a total distance of 599.4 kilometers. Future research could explore the comparison of alternative heuristic value calculation methods, such as the Manhattan Distance or Euclidean Distance, and the utilization of different optimization algorithms, like Dijkstra's Algorithm or Genetic Algorithms, to further enhance the accuracy and efficiency in determining the optimal route for the Eid homecoming journey.

REFERENCES

[1] K. P. R. I. Biro Komunikasi dan Informasi Publik, "Potensi Pergerakan Masyarakat Selama Lebaran 2024 Mencapai 193,6 juta Orang, Pemerintah Terapkan Kebijakan Efektif Untuk Antisipasi," *Kementrian Perhubungan Republik Indonesia*, Jakarta, Mar. 12, 2024. Accessed: Jun. 06, 2024. [Online]. Available: https://dephub.go.id/post/read/potensi-pergerakan-masyarakat-selama-lebaran-2024-mencapai-193,6-juta-orang,-pemerintah-terapkan-kebijakan-efektif-untuk-antisipasi#:~:text=Potensi%20Pergerakan%20Masyarakat%20Selama%20Lebaran,Antisipasi%20Kementerian%20Perhubungan%20Republik%20Indonesia

[2] Tasya Natalia, "5 Provinsi Tujuan Mudik Paling Banyak, Jawa Tengah Teratas!," *CNBC Indonesia*, Jakarta, Apr. 06, 2024. Accessed: Jun. 06, 2024. [Online]. Available: https://www.cnbcindonesia.com/research/20240405010837-128-528505/5-provinsi-tujuan-mudik-paling-banyak-jawa-tengah-teratas

[3] D. Marcelina and E. Yulianti, "Aplikasi Pencarian Rute Terpendek Lokasi Kuliner Khas Palembang Menggunakan Algoritma Euclidean Distance dan A*(Star)," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 9, no. 2, pp. 195–202, Jun. 2020, doi: 10.32736/sisfokom.v9i2.827.

ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 6, No. 3, May 2025, Page 581-590
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v6i3.5911

[4] R. Hidayati and N. Mutiah, "Penerapan Metode Haversine Formula Pada Pencarian Lokasi Fasilitas Kesehatan Terdekat," *Jurnal Media Informatika Budidarma*, vol. 6, no. 1, p. 278, Jan. 2022, doi: 10.30865/mib.v6i1.3445.

[5] M. Kisanrao Nichat, N. RChopde, and M. K. Nichat, "Landmark based shortest path detection by using A* Algorithm and Haversine Formula Landmark Based Shortest Path Detection by Using A* and Haversine Formula," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, no. 2, Apr. 2013, [Online]. Available: www.ijircce.com

[6] H. Rahmania Hatta, M. Hadi Suroso, I. F. Astuti, D. M. Khairina, and S. Maharani, "Application of Haversine Formula in Education Game 'Landmark Nusantara,'" in *Proceedings of the 2nd Borobudur International Symposium on Science and Technology (BIS-STE 2020)*, Magelang: Atlantis Press, Aug. 2021. doi: 10.2991/aer.k.210810.039.

[7] Z. A. Puspita, F. Fauziah, and I. D. Sholihati, "Metode Haversine Formula Pada Pencarian Rumah Sakit di Wilayah Jakarta Selatan Berbasis Android," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 8, no. 4, pp. 1142–1153, Nov. 2023, doi: 10.29100/jipi.v8i4.3962.

[8] S. Kartika and dan Raissa Amanda Putri, "Sistem Pencarian Lokasi dan Rute Terdekat Menggunakan Metode Haversine Formula Pada Aplikasi Donatur Pakaian Berbasis Android," *Al Ulum Sains dan Teknologi*, vol. 7, no. 1, pp. 14–20, Nov. 2021, doi: 10.31602/ajst.v7i1.5678.

[9] A. Bramato Wicaksono Putra, A. Aulia Rachman, and A. Santoso, "Perbandingan Hasil Rute Terdekat Antar Rumah Sakit di Samarinda Menggunakan Algoritma A*(star) dan Floyd-Warshall," *Sistem Informasi dan Komputer*, vol. 09, no. 1, pp. 59–68, Mar. 2020, doi: 10.32736/sisfokom.v9.i1.685.

[10] S. Russell and P. Norvig, *Artificial Intelligence A Modern Approach Third Edition*, 3rd ed. Upper Saddle River: Pearson Education Limited, 2016.

[11] Y. Miftahuddin, S. Umaroh, and F. R. Karim, "Perbandingan Metode Perhitungan Jarak Euclidean, Haversine, dan Manhattan Dalam Penentuan Posisi Karyawan," *Jurnal Tekno Insentif*, vol. 14, no. 2, pp. 69–77, Aug. 2020, doi: 10.36787/jti.v14i2.270.

[12] R. H. D. Putra, H. Sujiani, and N. Safriadi, "Penerapan Metode Haversine Formula Pada Sistem Informasi Geografis Pengukuran Luas Tanah," *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, vol. 4, no. 1, Mar. 2016.

[13] D. Malik, V. Rosalina, J. Raya Serang, and T. Drangong Serang, "Sistem Pemesanan Makanan Tradisional Berbasis Android Menggunakan Metode Haversine Formula," *Jurnal Sistem Informasi (JSiL)*, vol. 6, no. 1, pp. 12–19, Mar. 2019.

[14] I. Irwan and D. Atmajaya, "Sistem Informasi pencarian lokasi Perguruan Tinggi di Makassar," *ILKOM Jurnal Ilmiah*, vol. 10, no. 2, pp. 232–236, Aug. 2018, doi: 10.33096/ilkom.v10i2.251.232-236.

[15] M. F. Mahatmi, T. Hasanuddin, and F. Umar, "Implementasi Metode Haversine Formula Untuk Menentukan Jarak Terdekat Pada Pengantaran Air Galon Depot Anantama Berbasis Android," *Buletin Sistem Informasi dan Teknologi Islam*, vol. 3, no. 1, pp. 69–78, Feb. 2022, doi: 10.33096/busiti.v3i1.1098.

[16] M. Kisanrao Nichat, N. RChopde, and M. K. Nichat, "Landmark based shortest path detection by using A* Algorithm and Haversine Formula Landmark Based Shortest Path Detection by Using A* and Haversine Formula," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, no. 2, Apr. 2013, [Online]. Available: www.ijircce.com

[17] I. B. G. W. A. Dalem, "Penerapan Algoritma A* (Star) Menggunakan Graph Untuk Menghitung Jarak Terpendek," *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, vol. 1, no. 1, pp. 41–47, Apr. 2018, doi: 10.31598/jurnalresistor.v1i1.253.

[18] M. Ardiansyah Muktadir Gasba, U. Sumoharjo Km, and S. Selatan, "Implementasi Algoritma A* (A Star) dalam Menentukan Jarak Terpendek Menuju Rumah Sakit Rujukan Covid-19," *Buletin Sistem Informasi dan Teknologi Islam*, vol. 3, no. 3, pp. 203–212, 2022, doi: 10.31598/jurnalresistor.v1i1.253.

[19] Y. Mananoma, S. R. Sentinuwo, and A. M. Sambul, "Waste Transportation Route Optimization in Manado using A-Star Algorithm (A*)," *Jurnal Teknik Informatika*, vol. 16, no. 3, Sep. 2021, doi: 10.35793/jti.v16i3.34193.

[20] Suyanto, *Artificial Intelligence Searching, Reasoning, Planning, dan Learning*, 2nd ed. Bandung: Informatika Bandung, 2014.

[21] Zakaria, *Teknologi Informasi dan Komunikasi*. Jakarta: Arya Duta, 2016.