ISSN 2087-3336 (Print) | 2721-4729 (Online)

TEKNOSAINS: Jurnal Sains, Teknologi dan Informatika
Vol. 11, No. 1, January 2024, page. 103-111
http://jurnal.sttmcileungsi.ac.id/index.php/tekno
DOI: 10.37373

Travel application itinerary using the traveling salesman problem method and the held-karp algorithm

Ade Zaldi Eureka Zendar, Syarifah Putri Agustini Alkadri*, Izhan Fakhruzi

* Informatics Engineering Study Program, Faculty of Engineering and Computer Science, Muhammadiyah University, Pontianak, Indonesia

*✉ agustini.putri@unmuhpnk.ac.id

Submitted: 10/07/2023 Revised: 28/09/2023 Accepted: 04/10/2023

ABSTRACT

The Kapuas Hulu Regency is home to several fascinating tourist spots, yet finding directions inside the regency can be challenging for visitors. The Held-Karp Algorithm can be used to find the fastest route between any two cities with only one stop, which is known as the Traveling Salesman Problem (TSP). Bellman, Held, and Karp created this dynamic program in 1962 with the goal of minimizing travel time and expenses. The number of cities to be visited (forming nodes in the graph), the point of origin (the starting node in the graph), and the distances between the cities (weights between the nodes) must all be specified in the model of the TSP problem before the Held-Karp Algorithm can be applied. The focus of this study will be tourist destinations in West Kalimantan's Kapuas Hulu Regency. A system that uses the Held-Karp Algorithm to find the shortest paths between different tourist spots in Kapuas Hulu Regency will be built as part of this project. As a result, this technology will help tourists plan their trips effectively and minimize their expenditures for both time and transportation

Keyword: Travelling salesman problem; held-karp algorithm; itinerary; kapuas hulu tour

1. INTRODUCTION

One of West Kalimantan Province's Level II Regions is Kapuas Hulu Regency. Putussibau, the capital of the Kapuas Hulu district, is 846 km away by river from Kapuas, 814 km away by land, and 814 km away by air from Pontianak via Pangsuma Airport in small-body aircraft. With a population of 252,609, it occupies 20% of West Kalimantan's total area of 29,842.03 km².

A region's identity and distinguishing feature emerges from its tourism industry as it grows. The Kapuas Hulu Youth, Sports, and Tourism Office reports that there are 105 tourism destinations in the Kapuas Hulu Regency at the moment [1].

The TSP problem is specifically described as follows: a traveler wishes to visit n locations, but he can only visit each city once before returning to his hometown in order to minimize the overall distance traveled. The Traveling Salesman Problem is the name given to this idea (TSP) [2].

When figuring out the shortest path in problems like the Traveling Salesman Problem (TSP) or itinerary planning, the Held-Karp algorithm is frequently employed [3][4]. It can be observed that Held-Karp generates more optimal routes than the Iterative Deepening Search method when comparing the two algorithms. The Held-Karp algorithm has a complexity of O(2nn2) [5][6].

2. METHODS

This research was conducted using the following research methods:

a. Problem Identification: The first step in conducting research is to identify the issue that will be investigated. Identify the research's goals as well as the problem's formulation, background, and constraints.
b. Data Collection: Gathering data is the next step. The information in question is information that lends credence to the problem's formulation as study material for literary works. Interviews and direct observation are the techniques employed.
c. Needs Analysis: After determining the system's performance limitations, a needs analysis is conducted to ascertain the function of the system's users (admin, community).
    1) User Needs: this involves examining the requirements of system users across all roles.
    2) Software Requirements: at this point, examine the technologies employed in the creation of the system under design [7].
    3) System Requirements: The last phase of needs analysis is analyzing what the system needs in order to accomplish its objectives.
d. Design and Research Methods: The system's design is completed, and the computation techniques that will be applied to it are also completed.
e. After the design of the system to be developed is finished, the system is implemented.
f. Testing happens when the developed design is put into practice and then the system's output is examined. At this point, conclusions are also reached by examining the outcomes of system testing in Figure 1.

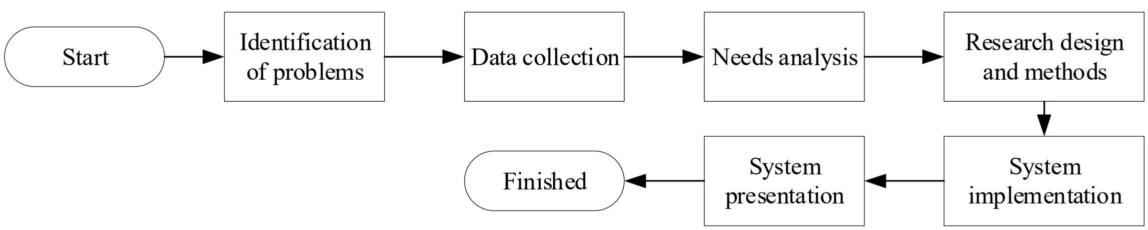

Figure 1. Research method flowchart.

3. RESULT AND DISCUSSION

3.1 Implementation of the Held-Karp Algorithm

The location and distance information that will be utilized to put the Held-Karp algorithm into practice is listed below. Names of tourist destinations and the separations between them were calculated using a variety of information from the Kapuas Hulu Tourism Office Table 1.

Table 1. Names and distances between tourist locations

Starting Point	Destination Point	Distance (km)
Ke – 0 Lubuk Mantuk	Ke – 1 Danau Empangau	60 km
Ke – 0 Lubuk Mantuk	Ke – 2 Betung Karihun	40 km
Ke – 0 Lubuk Mantuk	Ke – 3 Serau Berunyau	70 km
Ke – 1 Danau Empangau	Ke – 0 Lubuk Mantuk	60 km
Ke – 1 Danau Empangau	Ke – 2 Betung Karihun	100 km
Ke – 1 Danau Empangau	Ke – 3 Serau Berunyau	40 km
Ke – 2 Betung Karihun	Ke – 0 Lubuk Mantuk	40 km
Ke – 2 Betung Karihun	Ke – 1 Danau Empangau	100 km
Ke – 2 Betung Karihun	Ke – 3 Serau Berunyau	110 km
Ke – 3 Serau Berunyau	Ke – 0 Lubuk Mantuk	70 km
Ke – 3 Serau Berunyau	Ke – 1 Danau Empangau	40 km
Ke – 3 Serau Berunyau	Ke – 2 Betung Karihun	110 km

You can view the graph to make it simpler to set up connections and routes between locations in accordance with the above Table 1 and Figure *2*

ISSN 2087-3336 (Print) | 2721-4729 (Online) 105
DOI 10.37373/tekno.v11i1.682

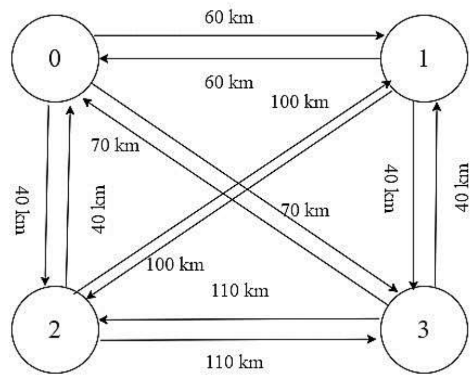

Figure 2. Route and distance graph in tourism data.

Subsequently, the problem's initial modeling will produce a 2-dimensional matrix with the subsequent layout:

Table 2. Distance between points

	0	1	2	3
0	0	60	40	70
1	60	0	100	40
2	40	100	0	110
3	70	40	110	0

We will move on to the first phase of problem solving after this modeling. Finding every subset of the tourist attraction set that is mapped to a set of points should be your first step. Following that, you'll get:

Subset:

Empty set,
{1}, {2}, {3},
{1, 2}, {2, 3}, {1, 3},
{1, 2, 3}

Table 3 it will map the search tree into a table (the empty set will be symbolized 'empty'):

Table 3. Travel route distance based on points.

	Cost	Parent node
[1, empty]	60	0
[2, empty]	40	0
[3, empty]	70	0
[2, {1}]	160	1
[3, {1}]	100	1
[2, {1}]	140	2
[3, {2}]	150	2
[1, {3}]	110	3
[2, {3}]	180	3
[3, {1, 2}]	180	1
[1, {2, 3}]	190	3
[2, {1, 3}]	210	1
[0, {1, 2, 3}]	250	3

The next step is to fill in all the array elements containing form elements, such as [1, empty], after the search tree mapping results are shown. This element's meaning is that travelers follow the entire path from the starting point back to the starting point.

Based on the previous explanation, it can be seen that the pair [3, {1,2}] tourists can interpret it as the total distance traveled from the starting point to the 3rd point while traveling to the 1st and 2nd points. early So what is the sequence of the route taken by this couple? The distance traveled is the distance with the smallest total distance, i.e. all permutations between 1 and 2 that make the distance from the starting point to the third point minimal. A real example is when tourists go from the starting point to the 3rd point and pass the 1st and 2nd points, then the order of rides has different options, such as: 0-1-2-3 or 0-2-1-3. The route chosen from 2 possible routes is the route that provides the smallest total distance.

Therefore, tourists can understand how to interpret the pair [0, {1,2,3}], namely tourists are looking for the total route from the starting point back to the starting point by passing 1, 2, 3 to at least kilometers ahead. What can be concluded from this description? Travelers can see the meaning pattern of each pair, namely [0, {1,2,3}] which is the final answer to the TSP problem when using this dynamic program. The final answer is the total distance traveled and the order of cities passed.

Now let's start filling in the table with the understanding obtained and referring to the adjacency matrix obtained previously (only a few examples of table elements will be discussed).

For the pair [1, empty]: The total distance traveled from the starting point to the 1st point without passing any point is the value at coordinates (0, 1) in the previous adjacency matrix, namely 60. The value of the Parent node is the point before the traveler up to the 1st point, namely the 0th point. So, Cost = 60 and Parent node = 0.

For the pair [2, empty]: The total distance traveled from the initial point to the 2nd point without passing any point is the value at coordinates (0, 2) in the previous adjacency matrix, namely 40. The value of the Parent node is the point before the traveler up to the 2nd point, namely point 0. So, Cost = 40 and Parent node = 0.

For the pair [3, empty]: The total distance traveled from the starting point to the 3rd point without passing any point is the value at coordinates (0, 3) in the previous adjacency matrix, namely 70. The value of the Parent node is the point before the traveler up to the 3rd point, namely point 0. So, Cost = 70 and Parent node = 0.

For the pair [2, {1}]: The total distance traveled from the initial point to the 2nd point and previously passing the 1st point is searched from the possible routes that can be passed. In this case there is only one possible route, namely 0-1-2. The distance traveled from the 0th point to the 1st point is 60 and the distance from the 1st point and the 2nd point is 100. So, the total is 160 and the point before reaching the 2nd point is the 1st point. So, Cost = 160 and Parent node = 1.

For the pair [3, {1}]: The total distance traveled from the starting point to the 3rd point and previously passing the 1st point is searched from the possible routes that can be passed. In this case there is only one possible route, namely 0-1-3. The distance traveled from point 0 to point 1 is 60 and the distance from point 1 and point 3 is 40. So, the total is 100 and the point before reaching point 3 is point 1. So, Cost = 100 and Parent node = 1.

For the pair [1, {2}]: The total distance traveled from the starting point to the 1st point and previously passing the 2nd point is searched from the possible routes that can be passed. In this case there is only one possible route, namely 0-2-1. The distance traveled from the 0th point to the 2nd point is 40 and the distance from the 2nd point and the 1st point is 100. So, the total is 140 and the point before reaching the 1st point is the 2nd point. So, Cost = 140 and Parent node = 2.

For the pair [3, {2}]: The total distance traveled from the starting point to the 3rd point and previously passing the 2nd point is searched from the possible routes that can be passed. In this case there is only one possible route, namely 0-2-3. The distance traveled from the 0th point to the 2nd point is 40 and the distance from the 2nd point and the 3rd point is 110. So, the total is 150 and the point before reaching the 3rd point is the 2nd point. So, Cost = 150 and Parent node = 2.

For the pair [1, {3}]: The total distance traveled from the starting point to the 1st point and previously passing the 3rd point is searched from the possible routes that can be passed. In this case there is only one possible route, namely 0-3-1. The distance traveled from point 0 to point 3 is 70 and the distance from point 3 and point 1 is 40. So, the total is 110 and the point before reaching point 1 is point 3. So, Cost = 110 and Parent node = 3.

For the pair [2, {3}]: The total distance traveled from the starting point to the 2nd point and previously passing the 3rd point is searched from the possible routes that can be passed. In this case there is only one possible route, namely 0-3-2. The distance traveled from the 0th point to the 3rd point is 70 and the distance from the 3rd point and the 2nd point is 110. So, the total is 180 and the point before reaching the 2nd point is the 3rd point. So, Cost = 180 and Parent node = 3.

For the pair [3, {1, 2}]: The total distance traveled from the starting point to the 3rd point by previously passing the 1st and 2nd points is searched from the possible routes that can be passed. In this case, there are two possibilities for this route, namely 0-1-2-3 or 0-2-1-3.

For the possibility 0-1-2-3, the distance from the starting point to the 1st point is 60, the distance from the 1st point and the 2nd point is 100, and the distance from the 2nd point and the 3rd point is 110, so the total is 270. The point before getting to the 3rd point is the 2nd point. For the possibility 0-2-1-3, the distance from the starting point to the 2nd point is 40, the distance from the 2nd point and the 1st point is 100, and the distance from the 1st point and the 3rd point is 40, so the total is 180. The point before getting to the 3rd point is the 1st point. From the two routes that have been calculated, tourists take the minimum Cost value, namely min {270, 180} which gives a result of 180. The Cost value of 180 is on route 0-2-1-3 which has a Parent node value of 1. So, Cost = 180 and Parent node = 1.

For the pair [2, {1, 3}]: The total distance traveled from the starting point to the 2nd point by previously passing the 1st and 3rd points is searched from the possible routes that can be passed. In this case, there are two possibilities for this route, namely 0-1-3-2 or 0-3-1-2. For the possibility 0-1-3-2, the distance from the initial point to the 1st point is 60, the distance from the 1st point and the 3rd point is 40, and the distance from the 3rd point and the 2nd point is 110, so the total is 210. The point before getting to the 2nd point is the 3rd point. For the possibility 0-3-1-2, the distance from the starting point to the 3rd point is 70, the distance from the 3rd point and the 1st point is 40, and the distance from the 1st point and the 2nd point is 100, so the total is 210. The point before getting to the 2nd point is the 1st point. From the two routes that have been calculated, tourists take the minimum Cost value, namely min {210, 210} which gives a result of 210. The Cost value of 210 is on route 0-3-1-2 which has a Parent node value of 3. So, Cost = 210 and Parent node = 1.

For the pair [1, {2, 3}]: The total distance traveled from the starting point to the 1st point by previously passing the 2nd and 3rd points is searched from the possible routes that can be passed. In this case, there are two possibilities for this route, namely 0-2-3-1 or 0-3-2-1. For the possibility 0-2-3-1, the distance from the initial point to the 2nd point is 40, the distance from the 2nd point and the 3rd point is 110, and the distance from the 3rd point and the 1st point is 40, so the total is 190. The point before reaching the 1st point is the 3rd point. For the possibility 0-3-2-1, the distance from the starting point to the 3rd point is 70, the distance from the 3rd point and the 2nd point is 110, and the distance from the 2nd point and the 1st point is 100, so the total is 280. The point before reaching the 1st point is the 2nd point. From the two routes that have been calculated, tourists take the minimum Cost value, namely min {31, 20} which gives a result of 20. The Cost value of 20 is on route 0-3-2-1 which has a Parent node value of 2. So, Cost = 190 and Parent node = 3.

For the pair [0, {1, 2, 3}]: The total distance traveled from the starting point to the starting point again by previously passing the 1st, 2nd, and 3rd points is searched for possible routes that can be passed. In this case there are six possibilities for this route, namely 0-1-2-3-0, 0-1-3-2-0, 0-2-1-3-0, 0-2-3-1- 0, 0-3-1-2-0, and 0-3-2-1-0. For cases with quite a large number of possibilities, tourists do not need to calculate the six possibilities one by one, but tourists will utilize the value of the Cost that has been obtained from previous pairs. This is in accordance with the principle of a dynamic program, where tourists save the solutions obtained for use at a later time. The solutions obtained from the previous pairs are stored for use in finding routes with a minimum total distance formed by permutations of a sufficiently large number of points. For example, tourists take point 1 as the point they pass before point 0, so that the form of the route they pass is as follows 0-x-y-1-0. The x and y values can be 2 or 3. Then, the possibility of the x-y-1 value can be found from the Cost value of the pair [1, {2,3}], where the value is 190. Next, the distance from the 1st point to the 0th is 60. So, for the first possibility the Cost value is 250 and the Parent node is 1. Then for the 2nd possibility where the tourist chooses the 2nd point as the point to pass before the 0th point, then the route is 0 -x-y-2-0. The minimum Cost value of the possible outcome x-y-2 is obtained from the pair [2, {1, 3}] which has a value of 210. Furthermore, the distance from the 2nd point to the 0th point is 40. So, for the 2nd possible Cost value is 250 and the Parent node is 2. Finally, for the 3rd possibility where the tourist chooses the 3rd point as the point to pass before the 0th point, then the route is 0-x-y-3-0. The minimum Cost value of the possible outcome x-y-3 is obtained from the pair [3, {1, 2}] which has a value of 180. Furthermore, the distance from the 3rd point to the 0th point is 70. So, for the 3rd possible Cost value is 250 and the Parent node is 3. After getting all the values from the 3 possibilities, the next step is to find the minimum value, namely with the min function {22, 21, 30} which produces a value of 21, which has a Parent node value of 2 .

Final solution: So, after getting the Cost and Parent node values from the pair [0, {1, 2, 3}], it shows that the tourist has got the final solution, where the minimum total distance traveled is 250 and the travel route traversed is 0-2-1-3-0. The travel route is obtained by starting from the Parent node belonging to the pair [0, {1, 2, 3}] namely 3, Parent node belonging to the pair [3, {1, 2}] namely 1, Parent node belonging to the pair [1, {2 }] which is 2, and the Parent node belonging to the pair [1, empty] is 0. So if combined and written backwards it becomes 0-2-1-3-0.

3.2 Database design

In this system, four tables are used, namely the location table to accommodate data regarding tourism, the admin table to accommodate manager data, the tourist table to accommodate user data or users looking for tourist locations and the distance table to accommodate the distance between tourist locations Figure 3.

Figure 3. Database design.

3.3 Black box testing

Testing is carried out using the Blackbox all-pair testing method to test the system's functionality by entering all possible inputs [8][9]. Comparing the results of manual calculations and the results of calculations carried out by the system to test the level of accuracy of the method implementation into the application that has been built [10].

In testing tourism applications, there are several stages that will be carried out by users as a guide during testing and implementation. These stages are as follows:

a. Users enter the tourist website application page, both the general page and the admin page
b. Users carry out experiments using black box testing, namely testing that pays attention to the functionality of the tourist application
c. Each user carries out black box testing from a back-end perspective by processing tourist destination data connected to the database.

The black box testing process is carried out by trying all combinations of input and output that occur, the results on testing the dashboard page and admin page, the system has worked as it should and is in accordance with the design [11][12].

3.4 Accuracy testing

After black box testing is carried out, the next step will be to look for the level of accuracy of the system that has been created, in determining the travel route [13]. This accuracy level calculation will compare the results of manual route calculations with calculations carried out from the system [14]. The results of this calculation are taken from 10 route samples, the results are as follows:

a. Lubuk Mantuk (0) – Empangau Lake (1) – Betung Kerihun National Park (2) – Water Sarai Brunyau Falls (3) System: 0 – 2 – 1 – 3 – 0
Manual: 0 – 2 – 1 – 3 – 0

b. Lake Empangau (0) – Bukit Tekenang Lake Sentarum (1) – Lubuk Mantuk (2) – Bukit Pig (3)
System: 0 – 2 – 3 – 1 – 0
Manual: 0 – 2 – 3 – 1 – 0
c. Lake Empangau (0) – Bukit Tekenang Lake Sentarum (1) – Betung Kerihun National Park (2) – Sarai Brunyau Waterfall (3)
System: 0 – 3 – 2 – 1 – 0
Manual: 0 – 2 – 3 – 1 – 0
d. Babi Hill (0) – Tekenang Lake Sentarum Lake (1) – Lubuk Mantuk (2) – Waterfall Sarai Brunyau (3)
System: 0 – 2 – 3 – 1 – 0
Manual: 0 – 2 – 3 – 1 – 0
e. Sarai Brunyau Waterfall (0) – Lubuk Mantuk (1) – Betung Kerihun National Park (2) – Pig Hill (3)
System: 0 – 3 – 2 – 1 – 0
Manual: 0 – 3 – 2 – 1 – 0
f. Taman Nasional Betung Kerihun (0) – Bukit Babi (1) – Air Terjun Sarai Brunyau (2) – Bukit Tekenang Danau Sentarum (3)
Sistem: 0 – 2 – 3 – 1 – 0
Manual: 0 – 2 – 3 – 1 – 0
g. Tekenang Lake Sentarum Lake (0) – Empangau Lake (1) – Babi Hill (2) – Park Betung Kerihun National (3)
System: 0 – 1 – 3 – 2 – 0
Manual: 0 – 1 – 3 – 2 – 0
h. Lubuk Mantuk (0) – Sarai Brunyau Waterfall (1) – Empangau Lake (2) – Babi Hill (3)
h. System: 0 – 3 – 2 – 1 – 0
Manual: 0 – 3 – 2 – 1 – 0
i. Babi Hill (0) – Betung Kerihun National Park (1) – Tekenang Lake Sentarum Hill (2) – Lubuk Mantuk (3)
System: 0 – 1 – 3 – 2 – 0
Manual: 0 – 1 – 3 – 2 – 0
j. Sarai Brunyau Waterfall (0) – Betung Kerihun National Park (1) – Tekenang Hill Lake Sentarum (2) – Lubuk Mantuk (3)
System: 0 – 2 – 1 – 3 – 0
Manual: 0 – 1 – 3 – 2 – 0

In comparing the results above, it can be seen that there are 2 samples that have different results, namely sample-3 and also sample-10. From this acquisition, there were 8 samples that had the same results as the system, and there were also 2 samples that had different results from the system. Therefore, the level of accuracy of the system can be calculated as follows:

$$\text{Level of accuracy} = \frac{\textit{Number of samples that have the same results}}{\textit{Number of Samples}} \times 100 \quad (1)$$

$$= \frac{8}{10} x\ 100$$

$$= 80\%$$

From the calculation above, the accuracy value of the tourism application is 80%.

3.5 Usability testing

This test is carried out by asking respondents to fill out a questionnaire. The purpose of this testing is to evaluate the system whether all its functions work correctly and according to the application objectives [15]. Hulu Kapuas tourist guests undergo usability testing. Respondents were asked to open the route website, create a route and test the available features without a specific scenario. Respondents were then asked to fill in feedback via a questionnaire and enter the values assessed by 51 respondents. Below is a Table 4 of usability test results:

Table 4. Website testing results for users.

No	Question	Answer Frequency				
		STS	TS	N	S	SS
1	Can the website provide a tour itinerary?	6	2	3	19	21
2	Can the website display tourist locations on a map?	5	1	3	22	20
3	Can the website provide directions to each location?	5	1	4	23	18
4	Can the website display tourist itineraries?	4	3	2	23	19
5	Can the website display detailed information for each location?	6	1	4	20	19
Value Weight		26	16	48	428	485

Information:

STS = Strongly Disagree
TS = Don't agree
N = Neutral
S = Agree
SS = Strongly agree
Maximum value = SS value x number of questions x total respondents
= 5 x 5 x 51
= 1.275
Questionnaire value = 26 + 16 + 48 + 428+ 485
= 1.003
Final score = 1.003/1.275 x 100%
= 78,67%

4. CONCLUSION

In this study, we address the issue of determining the quickest path for visitors wishing to visit Kapuas Hulu Regency's tourist destinations. The Held-Karp method, which is known to make it simpler for visitors to find the shortest route when visiting Kapuas Hulu Regency tourism, was used in the creation of the application. Black box techniques, manual experiments, and usability testing are all used to test applications. The application was operating as it should from the admin side and the user side, according to testing conducted with the black box method. An 80% system success rate was achieved in tests comparing the system with manual calculations. Furthermore, according to survey data on usability testing, 78.67% of users were satisfied with the application's features

REFERENCE

[1] S. Siahaan, F. Mizwar, and R. S. Wulandari, “Penilaian Potensi Objek Wisata Air Terjun Saray Brunyau Desa Riam Piyang Kecamatan Bunut Hulu Kabupaten Kapuas Hulu,” *J. Belantara*, vol. 5, no. 1, pp. 96–105, 2022, doi: 10.29303/jbl.v5i1.862.
[2] A. Yumalia, “Minimasi Biaya Distribusi Dengan Menggunakan Metode Traveling Salesman Problem (TSP),” *J. UMJ*, no. November 2017, pp. 1–8, 2017, [Online]. Available: jurnal.umj.ac.id/index.php/semnastek
[3] F. M. Gurnitowati, Rochmad, and Supriyono, “Penerapan Algoritma Branch and Bound Untuk Menentukan Rute Objek Wisata Di Semarang,” *Unnes J. Math.*, vol. 3, no. 1, 2014.
[4] M. Irfan, “Penyelesaian Travelling Salesman Problem (TSP) Menggunakan Algoritma Hill

Climbing dan MATLAB," *Matematika*, vol. 17, no. 1, pp. 13–20, 2018, doi: 10.29313/jmtm.v17i1.3090.

[5] I. G. S. Rahayuda and N. P. L. Santiari, "Basis Path Testing of Iterative Deepening Search and Held-Karp on Pathfinding Algorithm," *Kursor*, vol. 9, no. 2, pp. 39–48, 2018, doi: 10.28961/kursor.v9i2.129.

[6] J. Ilmiah, I. Komputa, E. Volume, and B. Issn, "IMPLEMENTASI ALGORITMA ITERATIVE DEEPENING A * DAN METODE PRUNING PADA SOLUSI PERMAINAN PUZZLE FLOW Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)".

[7] M. Nur Ichsanudin, M. Yusuf, Uminingsih, and Suraya, "Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula Info Artikel Abstrak," *STORAGE - J. Ilm. Tek. dan Ilmu Komput.*, vol. 1, no. 2, pp. 1–8, 2022, [Online]. Available: https://journal.literasisains.id/index.php/storage/article/view/270

[8] H. Raihan and A. Voutama, "Pengujian Black Box Pada Aplikasi Database Perguruan Tinggi dengan Teknik Equivalence Partition," *Antivirus J. Ilm. Tek. Inform.*, vol. 17, no. 1, pp. 1–18, 2023, doi: 10.35457/antivirus.v17i1.2501.

[9] M. Zidan, S. Nur'aini, N. C. H. Wibowo, and M. A. Ulinuha, "Black Box Testing pada Aplikasi Single Sign On (SSO) di Diskominfostandi Menggunakan Teknik Equivalence Partitions," *Walisongo J. Inf. Technol.*, vol. 4, no. 2, pp. 127–137, 2022, doi: 10.21580/wjit.2022.4.2.12135.

[10] H. Nurfauziah and I. Jamaliyah, "Perbandingan Metode Testing Antara Blackbox Dengan Whitebox Pada Sebuah Sistem Informasi," *J. Vis.*, vol. 8, no. 2, pp. 105–113, 2022, [Online]. Available: https://jurnas.saintekmu.ac.id/index.php/visualika/article/view/24

[11] B. B. Sasongko, F. Malik, F. Ardiansyah, A. F. Rahmawati, F. D. Adhinata, and D. P. Rakhmadani, "Pengujian Blackbox Menggunakan Teknik Equivalence Partitions pada Aplikasi Petgram Mobile," *Fak. Inform. Inst. Teknol. Telkom Purwokerto*, vol. 2, no. 1, pp. 10–16, 2021, [Online]. Available: https://ejurnal.teknokrat.ac.id/index.php/ictee/article/view/1012

[12] D. Triady, Izmy Alwiah Musdar, and Hendra Surasa, "Pengujian Blackbox Pada Website Worker'S Menggunakan Metode Equivalence Partitioning," *KHARISMA Tech*, vol. 18, no. 1, pp. 84–98, 2023, doi: 10.55645/kharismatech.v18i1.292.

[13] A. Ariansa, A. N. Apriyand, R. Ryan, and A. Saifudin, "BOX TESTING DALAM MENENTUKAN RUTE PERJALANAN," vol. 1, no. 1, pp. 66–69, 2022.

[14] A. Fajri, E. P. Purwandari, and F. F. Coastera, "METODE ALGORITMA DIJKSTRA (Studi Kasus : Kelurahan Kandang Limun dan Beringin Raya)," vol. 6, no. 1, pp. 69–79, 2018.

[15] U. U. Sufandi, M. Priono, D. A. Aprijani, B. A. Wicaksono, and D. Trihapningsari, "Uji Usability Fungsi Aplikasi Web Sistem Informasi Dengan Use Questionnaire (Studi Kasus : Aplikasi Web Sistem Informasi Tiras dan Transaksi Bahan Ajar)," *J. Pendidik. Teknol. dan Kejuru.*, vol. 19, no. 1, p. 25, 2022.