Integra: Journal of Integrated Mathematics and Computer Science

Journal Homepage: https://integrajimcs.com/index.php/integra/index

integra
Journal of Integrated Mathematics and Computer Science

Research Paper

The Comparison of the Nearest Neighbor and Modified Sollin Algorithms for Determining the Shortest Tour for BNI Bank Branches in Lampung Province

Assyfa Dwipa Maharani[1], Desiana Putri[1]*, Nonik Mega Sapitri[1], Notiragayu[1], Muslim Ansori[1]

[1]*Department of Mathematics, Faculty of Mathematics and Natural Science, Universitas Lampung, Lampung, 35145, Indonesia*
***Corresponding author**: desianaputri@fmipa.unila.ac.id

Keywords
Travelling Salesman Problem, Nearest Neighbor Heuristic, Algoritma Sollin, Euclidean, Haversine

Abstract
BNI is one of the government -owned banks in Indonesia. At certain times, BNI experiences a significant increase in the number of customers, which can lead to long queues. In such situations, customers generally consider visiting other BNI branches. To optimize time and cost, it is necessary to have information to determine the shortest tour between BNI locations in Lampung Province. The purpose of this study is to determine the shortest tour among 21 BNI locations in Lampung Province. The results show that both the Nearest Neighbor Heuristic Algorithm and the Modified Sollin's Algorithm produce the same solution for Euclidean distance data and Haversine distance data. However, for time-based data, the Modified Sollin's Algorithm performs better than the Nearest Neighbor Heuristic Algorithm.

Received: 12 August 2025, Accepted: 7 November 2025
https://doi.org/10.26554/integra.20252342

1. INTRODUCTION

Bank Negara Indonesia (BNI) is one of the government-owned banks with branches in every province. BNI offers a range of financial products aimed at enhancing service quality for its customer. However, during peak periods, the number of customers visiting the bank increases significantly. As a result, customers may choose to visit alternative BNI branches. Therefore, route optimization is needed to determine the shortest tour among BNI branches in Lampung Province to minimize travel time and cost. This scenario aligns with the Traveling Salesman Problem (TSP), which seeks the shortest possible route that visits each location once.

TSP is an optimization problem aimed at finding the shortest travel route among several locations. Previous studies that examined the TSP problem included research conducted by AlSalibi et al, [1], which found that the NNH algorithm performs very well when the number of cities analyzed is fewer than 50. Similar research was conducted by Sinaga and Marpaung [2], which found that the NNH algorithm proved to be more effective than the CIH algorithm in determining the distribution route for PT JNE Medan. In addition, the TSP also be solved using the Modified Sollin' Algorithm. This algorithm is a development of the Sollin Algorithm and produces a Hamiltonian circuit, representing the shortest tour of the TSP problem. Modified Sollin's Algorithms was investigated by Dinata [3] to find the optimal tour for tourist locations. However, the implementation of the Modified Sollin's Algorithm for solving the TSP remains limited.

TSP is one of the optimization problems in network design that has been extensively investigated. In addition to the Nearest Neighbor Algorithm, other algorithms have been developed and implemented, including Brute Force [4, 5], Cheapest Insertion Heuristics [6, 7, 8, 9], Nearest Neighbor [5, 10, 11, 12, 13], Christofides Algorithm [7, 8, 14, 15], Genetic Algorithm [16, 17, 18, 19, 20, 21], Ant Colony [22, 23], and many others.

In this study, the Nearest Neighbor Heuristic (NNH) Algorithm and Modified Sollin Algorithm were applied to obtain the shortest tour of BNI locations in Lampung Province. There were 21 BNI locations selected. These locations were selected because there were quite a few BNI branches spread across the province, so most of the locations were taken from 10 cities in Lampung Province. These locations were determined by setting BNI UNILA as the center point for other BNI branches with a diameter of no more than 170 km from the center.

Table 1. The List of 21 Branches of BNI Banks

Vertex	Bank	Latitude	Longitude
A	BNI KLN Antasari	-5.40515	105.27835
B	BNI KLN Antasari	-5.44941	105.26565
C	BNI on Laks. Malahayati Road	-5.40842	105.25722
D	Bank BNI 46, Pasir Gintung	-5.41637	105.25412
E	BNI 46, Jl. Kartini, Palapa	-5.4203	105.26027
F	KCP BNI Jl. Majapahit	-5.46526	105.31939
G	BNI Yos Sudarso	-5.3957	105.30024
H	BNI Sukarame	-5.36955	105.24132
I	BNI UNILA	-5.3781	105.21876
J	BNI Kantor Kas Univ. Malahayati	-5.38345	105.27189
K	BNI Way Halim	-5.11447	105.31311
L	BNI Metro A.H. Nasution	-4.83592	104.89457
M	BNI Cabang Kotabumi	-5.35462	104.97975
N	Bank BNI Capem Pringsewu	-5.38065	105.10443
O	BNI Gedong Tataan	-4.9365	105.21168
P	Bank (BNI) Bandar Jaya	-5.17087	105.70187
Q	BNI 45 Way Jepara	-5.72814	105.59384
R	BNI Unit Kalianda	-5.31618	105.19666
S	BNI Natar	-5.58226	105.51818
T	Bank BNI 46 Sidomulyo	-4.63139	105.07508
U	Bank BNI Dayamurni	-5.36383	104.7797

2. THE METHOD

Graph $G(V,E)$ is a mathematical structure that consist of a set V (the set of n vertices, $V \neq \emptyset, V = \{v_1, v_2, \ldots, v_n\}$), and set $E = \{e_{ij} | i, j \in V\}$ of edges that connect vertices in V. E maybe an empty set. A weighted graph is a particular kind of graph in which each edge is given a numerical value known as a weight.

The number of vertices of a graph is called as the order of the graph, and the degree of vertex v_i the number of edges that incident to it.

2.1 The Travelling Salesman Problem

The Traveling Salesman Problem (TSP) is one of the combinatorial optimization concepts that often becomes a problem in business and everyday life. TSP is described as a tour that must be taken by a salesman in distributing goods from the city of origin to several destination cities and then returning to the city of origin with the condition that the destination cities are visited exactly once and the tour obtained can minimize the total distance traveled and the costs that must be incurred.

2.2 Google Maps

Google Maps is a service developed by Google that provides maps with road information and visual representations in the surrounding area [24]. The Google Maps service can be easily accessed via `https://maps.google.com`. The functions of google maps that users can obtain include: facilitating location searches and map storage, finding unfamiliar locations, easy access anywhere, and up-to-date information [25].

2.3 Nearest Neighbor Heuristic

The procedure for using the Nearest Neighbor Heuristic Algorithm is presented in the following steps [4]: The Christofides Algorithm procedure is given in the following steps:

Step 1 : Select a vertex that will be the origin and also the terminal of the tour, and labelled it as visited. vertex, and other vertices as unvisited.
Step 2 : Starting from this vertex, identify the edges that connect the visited and unvisited vertices.
Step 3 : Choose the smallest edge that connects the visited vertices to the unvisited vertices.
Step 4 : The selected vertex will be the starting vertex for moving to another vertex, then mark the vertex as visited.
Step 5 : Repeat Steps (2) to (4) until all vertices are marked as visited.

2.4 Modified Sollin Algorithm

The Sollin (also known as Borůvka) algorithm is used to determine the Minimum Spanning Tree. The core principle of the Sollin or Boruvka algorithm is to connect every vertex with the smallest edge incident to it so that several trees are formed. If a single tree remains, the MST has been successfully constructed. Otherwise, the trees are connected using minimum-weight intercomponent edges [26]. To adapt the Sollin algorithm for solving the Traveling Salesman Problem (TSP), modifications are introduced during the component merging phase, a check is performed for vertices with degree greater than 2. The following is the procedure for Modified Sollin's algorithm for TSP.

Step 1 : Connect each vertex in G with the shortest edge incident to it.
Step 2 : Check vertices that have degree greater than two, suppose the vertex is v_i, so that the degree of v_i is 2
Step 3 : Remove some edges incident to v_i so that the degree of v_i is 2.
Step 4 : Connect the isolated vertex (formed after removing process in Step 3), to another vertex whose degree one with the smallest edge.
Step 5 : Combine the formed graph components by connecting vertices with a degree of one with the smallest weight.

2.5 Euclidean Distance

Euclidean distance is one of the most commonly used methods for measuring the distance between two points. It works based on the concept of direct distance without obstacles, similar to calculating the length of a triangle's diagonal. This means that the distance between point A and point B is equal to the distance between point B and point A [27].

Table 2. Euclidean Distance (in km) of 21 Locations of BNI Branches

From/To	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
A	0	5.1	2.4	3	2.6	8.1	2.7	5.7	7.3	2.5	32.6	76.4	33.7	19.6	52.7	53.9	50.3	13.4	33.2	89.1	55.7
B	5.1	0	4.7	3.9	3.3	6.2	7.1	9.3	9.5	7.4	37.7	79.8	33.5	19.5	57.4	57.6	47.9	16.7	31.8	93.5	54.9
C	2.4	4.7	0	0.9	1.4	9.4	5	4.7	5.5	3.2	33.3	75.4	31.5	17.3	52.8	56.1	51.7	12.3	34.9	88.8	53.4
D	3.0	3.9	0.9	0	0.8	9.1	5.6	5.4	5.8	4.2	34.2	76	31.3	17.1	53.6	56.8	51.3	12.9	34.7	89.6	53.1
E	2.6	3.3	1.4	0.8	0	8.3	5.2	6	6.6	4.3	34.6	76.7	32.1	17.9	54.1	56.5	50.5	13.6	33.9	90.2	53.9
F	8.1	6.2	9.4	9.1	8.3	0	8	13.7	14.8	10.5	39.1	84.5	39.8	25.7	60.1	53.7	42.3	21.5	25.7	96.7	61.1
G	2.7	7.1	5.0	5.6	5.2	8.0	0	7.2	9.3	3.4	31.3	77	36	21.9	52.1	51.2	49.4	14.5	31.9	88.7	58.1
H	5.7	9.3	4.7	5.4	6.0	13.7	7.2	0	2.7	3.7	29.5	70.8	29.2	15.3	48.3	55.8	56	7.7	38.9	84.2	51.4
I	7.3	9.5	5.5	5.8	6.6	14.8	9.3	2.7	0	5.9	31.2	70.3	26.7	12.7	49.2	58.5	57.1	7.3	40.3	84.7	48.9
J	2.5	7.4	3.2	4.2	4.3	10.5	3.4	3.7	5.9	0	30.3	74	32.7	18.6	50.2	53.4	52.5	11.2	35.2	86.5	54.8
K	32.6	37.7	33.3	34.2	34.6	39.1	31.3	29.5	31.2	30.3	0	56	45.7	37.7	22.8	43.7	75.1	25.9	56.9	60	65.5
L	76.4	79.8	75.4	76.0	76.7	84.5	77.0	70.8	70.3	74.0	56.0	0	58.5	65	37	97.3	126.2	63.2	108.3	30.4	60.1
M	33.7	33.5	31.5	31.3	32.1	39.8	36.0	29.2	26.7	32.7	45.7	58.5	0	14.2	53.2	82.9	80	24.5	65.1	81.2	22.3
N	19.6	19.5	17.3	17.1	17.9	25.7	21.9	15.3	12.7	18.6	37.7	65.0	14.2	0	50.9	70.5	66.8	12.5	51.2	83.5	36.2
O	52.7	57.4	52.8	53.6	54.1	60.1	52.1	48.3	49.2	50.2	22.8	37.0	53.2	50.9	0	60.5	97.9	42.3	79.6	37.2	67.6
P	53.9	57.6	56.1	56.8	56.5	53.7	51.2	55.8	58.5	53.4	43.7	97.3	82.9	70.5	60.5	0	63.2	58.5	50.2	92.1	104.9
Q	50.3	47.9	51.7	51.3	50.5	42.3	49.4	56.0	57.1	52.5	75.1	126.2	80.0	66.8	97.9	63.2	0	63.7	18.3	135.1	99.3
R	13.4	16.7	12.3	12.9	13.6	21.5	14.5	7.7	7.3	11.2	25.9	63.2	24.5	12.5	42.3	58.5	63.7	0	46.5	77.4	46.7
S	33.2	31.8	34.9	34.7	33.9	25.7	31.9	38.9	40.3	35.2	56.9	108.3	65.1	51.2	79.6	50.2	18.3	46.5	0	116.8	85.7
T	89.1	93.5	88.8	89.6	90.2	96.7	88.7	84.2	84.7	86.5	60.0	30.4	81.2	83.5	37.2	92.1	135.1	77.4	117	0	87.9
U	55.7	54.9	53.4	53.1	53.9	61.1	58.1	51.4	48.9	54.8	65.5	60.1	22.3	36.2	67.6	104.9	99.3	46.7	85.7	87.9	0

Table 3. Haversine Distance (in km) of 21 Locations of BNI Branches

From/To	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
A	0	5.1	2.7	2.3	2.3	4.9	7.1	5.7	7.3	2.5	32.6	76.3	33.7	19.5	52.6	53.9	50.2	13.4	33.2	89	55.6
B	5.1	0	4.7	3.9	6.8	4.3	6.6	10.3	7.9	9	55.6	75.3	20.8	9.7	74.9	57.6	47.9	16.7	16.6	93.4	54.9
C	2.7	4.7	0	1	1.4	9.4	5	4.6	5.5	3.2	33.3	75.3	31.5	17.3	52.7	56	51.6	12.3	34.9	88.7	53.4
D	2.3	3.9	1	0	0.8	9.1	5.6	5.3	5.8	4.1	34.2	75.9	31.3	17.1	53.5	56.8	51.3	12.8	34.7	89.5	53.1
E	2.3	6.8	1.4	0.8	0	8.3	5.2	6	6.6	4.3	34.5	76.6	32.1	17.9	54	56.4	59.9	38.8	19.4	90.1	53.8
F	4.9	4.3	9.4	9.1	8.3	0	8.1	13.7	14.8	10.5	39.1	84.4	39.7	25.7	60	53.7	42.2	21.5	25.6	96.7	61.1
G	7.1	6.6	5	5.6	5.2	8.1	0	7.2	9.3	3.5	31.3	76.9	36	21.9	52	51.2	49.3	14.5	31.9	88.6	58
H	5.7	10.3	4.6	5.3	6	13.7	7.2	0	2.7	3.7	29.5	70.8	29.2	15.3	48.3	55.8	55.9	7.8	38.8	84.2	51.4
I	7.3	7.9	5.5	5.8	6.6	14.8	9.3	2.7	0	6	31.2	70.3	26.7	12.7	49.1	58.5	38.9	7.4	40.3	84.6	48.8
J	2.5	9	3.2	4.1	4.3	10.5	3.5	3.7	6	0	30.3	73.9	32.6	18.6	50.1	53.4	52.4	11.2	35.2	86.5	54.8
K	32.6	55.6	33.3	34.2	34.5	39.1	31.3	29.5	31.2	30.3	0	55.9	45.7	37.6	22.7	43.7	75	25.9	56.8	59.9	65.5
L	76.3	75.3	75.3	75.9	76.6	84.4	76.9	70.8	70.3	73.9	55.9	0	58.5	64.9	37	97.2	126	63.1	108.1	30.3	60.1
M	33.7	20.8	31.5	31.3	32.1	39.7	36	29.2	26.7	32.6	45.7	58.5	0	14.2	53.2	82.9	79.9	24.5	65	81.2	22.3
N	19.5	9.7	17.3	17.1	17.9	25.7	21.9	15.3	12.7	18.6	37.6	64.9	14.2	0	50.8	70.4	66.7	12.5	51.2	83.4	36.2
O	52.6	74.9	52.7	53.5	54	60	52	48.3	49.1	50.1	22.7	37	53.2	50.8	0	60.4	97.7	42.2	79.5	37.2	67.6
P	53.9	57.6	56	56.8	56.4	53.7	51.2	55.8	58.5	53.4	43.7	97.2	82.9	70.4	60.4	0	63.1	58.5	50.1	92	104.8
Q	50.2	47.9	51.6	51.3	59.9	42.2	49.3	55.9	38.9	52.4	75	126	79.9	66.7	97.7	63.1	0	63.6	18.2	134.9	99.2
R	13.4	16.7	12.3	12.8	38.8	21.5	14.5	7.8	7.4	11.2	25.9	63.1	24.5	12.5	42.2	58.5	63.6	0	46.4	77.3	46.7
S	33.2	16.6	34.9	34.7	19.4	25.6	31.9	38.8	40.3	35.2	56.8	108.1	65	51.2	79.5	50.1	18.2	46.4	0	116.7	85.6
T	89	93.4	88.7	89.5	90.1	96.7	88.6	84.2	84.6	86.5	59.9	30.3	81.2	83.4	37.2	92	134.9	77.3	116.7	0	87.9
U	55.6	54.9	53.4	53.1	53.8	61.1	58	51.4	48.8	54.8	65.5	60.1	22.3	36.2	67.6	104.8	99.2	46.7	85.6	87.9	0

Table 4. Time Travel (in Minute) of 21 Locations of BNI Branches

From/To	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
A	0	13	9	12	10	18	6	18	22	9	60	154	65	40	70	106	77	28	57	124	105
B	13	0	12	10	11	15	22	24	24	21	76	161	66	41	89	112	76	34	56	173	106
C	9	12	0	7	7	22	14	12	16	9	65	149	61	37	80	110	82	23	61	165	95
D	12	10	7	0	5	20	15	14	16	11	67	151	60	35	82	112	80	25	60	164	95
E	10	11	7	5	0	18	14	18	21	15	72	155	65	40	86	112	80	30	59	168	102
F	18	15	22	20	18	0	14	32	35	23	77	165	80	54	82	101	64	40	45	176	116
G	6	22	14	15	14	14	0	20	24	11	60	155	68	42	71	81	56	30	55	168	106
H	18	24	12	14	18	32	20	0	9	10	60	142	57	30	76	109	89	16	69	155	91
I	22	24	16	16	21	35	24	9	0	18	63	143	57	29	60	116	98	18	76	117	91
J	9	21	9	11	15	23	11	10	18	0	62	149	63	38	66	102	85	23	64	161	100
K	60	76	65	67	72	77	60	60	63	62	0	111	89	75	44	59	114	50	109	124	127
L	154	161	149	151	155	165	155	142	143	149	111	0	134	162	73	170	231	132	210	58	157
M	65	66	61	60	65	80	68	57	57	63	89	134	0	29	106	162	149	62	129	182	37
N	40	41	37	35	40	54	42	30	29	38	75	162	29	0	88	137	120	34	98	166	62
O	70	89	80	82	86	82	71	76	60	66	44	73	106	88	0	100	162	63	141	75	142
P	106	112	110	112	112	101	81	109	116	102	59	170	162	137	100	0	116	120	115	172	197
Q	77	76	82	80	80	64	56	89	98	85	114	231	149	120	162	116	0	83	29	242	164
R	28	34	23	25	30	40	30	16	18	23	50	132	62	34	63	120	83	0	78	142	92
S	57	56	61	60	59	45	55	69	76	64	109	210	129	98	141	115	29	78	0	139	159
T	124	173	165	164	168	176	168	155	117	161	124	58	182	166	75	172	242	142	139	0	207
U	105	106	95	95	102	116	106	91	91	100	127	157	37	62	142	197	164	92	159	207	0

The Euclidean distance formula is as follow:

$$\text{distance} = \sqrt{((lat_1 - lat_2)^2 + ((lon_1 - lon_2)^2)} \quad (1)$$

where

lat_1 : latitude 1 lon_1 : longitude 1

lat_2 : latitude 2 lon_2 : longitude 2

2.6 Haversine Distance

Haversine is an equation used to calculate the shortest distance between two locations on the spherical surface of the Earth.

Figure 1. The Tour Obtained with Euclidean Distance Data using NNH

Figure 2. The Tour Obtained with Haversine Distance Data using NNH

This formula takes into account the curvature of the Earth using latitude and longitude coordinates [28].

The Haversine distance is as follow:

$$x = lon_2 - lon_1 \cdot \cos\left(\frac{lat_1 + lat_2}{2}\right)$$

$$y = lat_2 - lat_1$$

$$\text{distance} = \sqrt{x \cdot x + y \cdot y}\, R$$

where:

x = longitude

y = latitude

R = the average radius of the earth (6371 km)

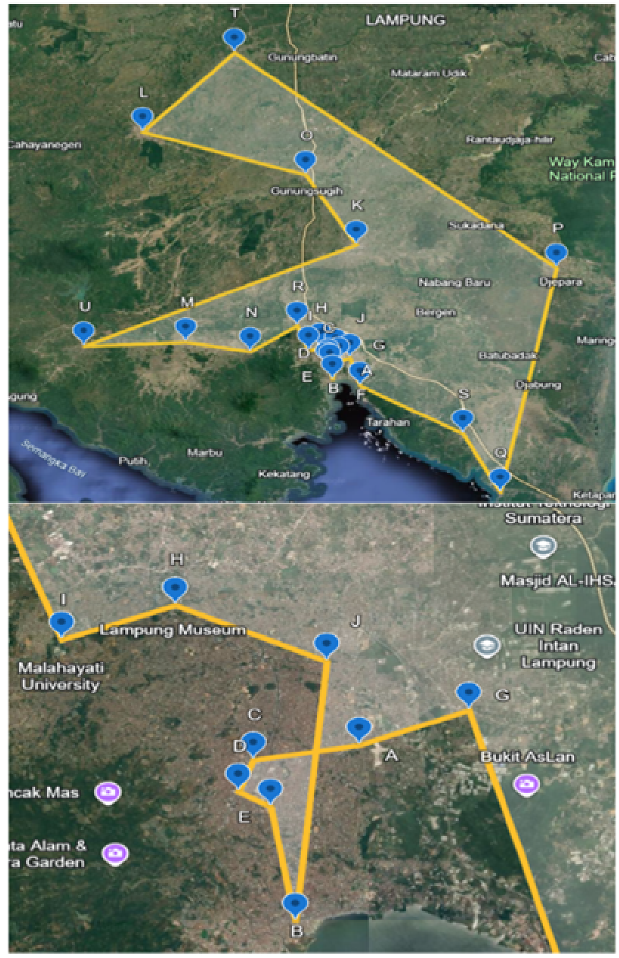


Figure 3. The Tour Obtained with Travel Time Data using NNH

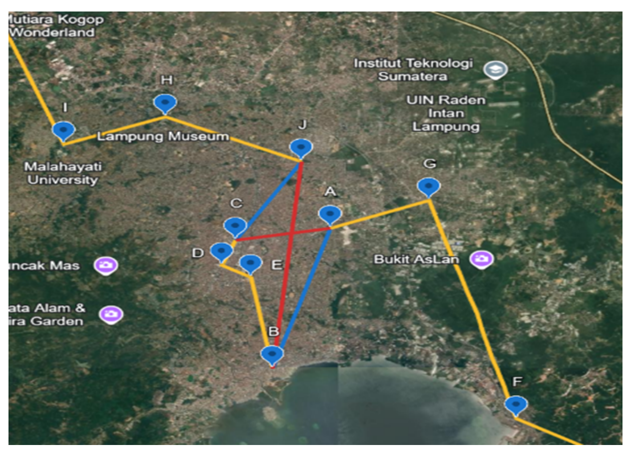


Figure 4. The Revised Tour

3. RESULTS AND DISCUSSION

3.1 The Data

This study uses distance and time data for Bank Negara Indonesia locations in Lampung Province, comprising 21 BNI locations

Figure 5. The Tour Obtained with Travel Time Data using NNH

with a diameter of no more than 170 km (kilometers) between BNI UNILA as the central point. Table 1 shows the list of 21 BNI branches, Table 2 displays the Euclidean distance for 21 BNI branches, Table 3 shows the Haversine distance for those BNI branches, and Table 4 displays the time needed to go from one branch to another branches.

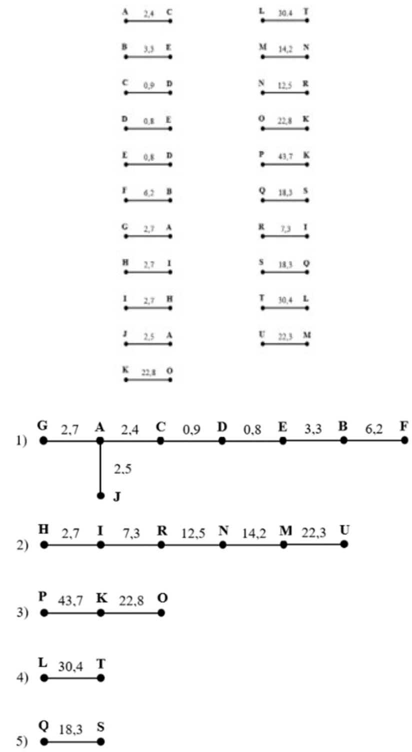
3.2 Solving TSP using Nearest Neighbor Algorithm

Based on the solution steps using the NNH described above, 21 tours can be formed. Tables 5, 6, and 7 show the result of solving the TSP using NNH, where in Table 5 presents results using Euclidean distance data, Table 6 uses Haversine distance data, and Table 7 uses travel time data. Figure 1 displays the best tour obtained using NNH implemented in Euclidean data, Figure 2 displays the best tour implemented on Haversine data, and Figure 3 implemented on travel time data. All results obtained the same starting and terminal vertex, which starts and end at vertex Q.

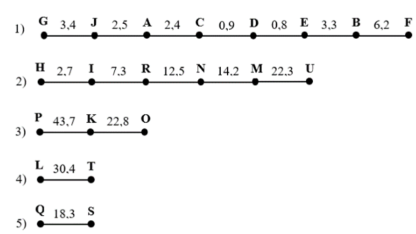
Note that on Figure 3 the tour obtained contained crossing path (the bottom picture). It indicates that we can get better solution by refining the path. Figure 4 shows the removed edge $e_{(J,B)}$ (red color) and added edge $e_{(J,C)}$ (green color) with the weight reduction $-21 + 9 = -12$, and also removed edge $e_{(A,C)}$ (red color) and added edge $e_{(A,B)}$ (green color) with the weight reduction $-9 + 13 = 4$. Therefore, the new tour has total weight of $874 - 12 + 4 = 866$ minutes. Figure 5 show the new tour after revision.

3.3 Solving TSP using Modified Sollin Algorithm Implemented on Euclidean Distance

1. Connecting every vertex with the smallest edge incident to it.
   Connect the same labelled vertices, so we get five components as follow:

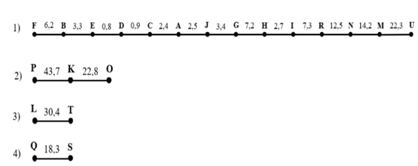


2. Check vertices that have degree greater than two. Note that component 1 has one vertex with degree 3 which is vertex A which adjacent with vertex G, J, and C with edges $e_{AC} = 2.4$, $e_{AJ} = 2.5$, and $e_{AG} = 2.7$.
3. Since the weight e_{AG} is the largest, remove it. This process make G becomes isolated.
4. Connect G with to another vertex whose degree one with the smallest edge, which is with vertex J with $e_{GJ} = 3.4$, so we get the components as follow:



5. Combine the formed graph components by connecting vertices with a degree of one with the smallest weight.

By connecting component (1) and (2) we get the following:



Do the same process, we get the following tour with total weight 414.7 km.

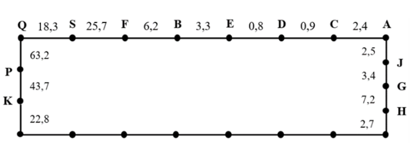
We do the similar process implemented on Haversine and travel time data. The results are as following:

Table 5. The Tour Obtained Using NN Algorithm Implemented on Euclidean Distance Data

The starting and terminal vertex	The Tour	Total distance (km)
A	A - C - D - E - B - F - G - J - H - I - R - N - M - U - L - T - O - K - P - S - Q - A	400.7
B	B - E - D - C - A - J - G - H - I - R - N - M - U - L - T - O - K - F - S - Q - P - B	433.9
C	C - D - E - A - J - G - B - F - H - I - R - N - M - U - L - T - O - K - P - S - Q - C	410.6
D	D - E - C - A - J - G - B - F - H - I - R - N - M - U - L - T - O - K - P - S - Q - D	410.5
E	E - D - C - A - J - G - B - F - H - I - R - N - M - U - L - T - O - K - P - S - Q - E	409.2
F	F - B - E - D - C - A - J - G - H - I - R - N - M - U - L - T - O - K - P - S - Q - F	390.7
G	G - A - C - D - E - B - F - J - H - I - R - N - M - U - L - T - O - K - P - S - Q - G	401.6
H	H - I - C - D - E - A - J - G - B - F - R - N - M - U - L - T - O - K - P - S - Q - H	420.9
I	I - H - J - A - C - D - E - B - F - G - R - N - M - U - L - T - O - K - P - S - Q - I	413.8
J	J - A - C - D - E - B - F - G - H - I - R - N - M - U - L - T - O - K - P - S - Q - J	405.5
K	K - O - L - T - R - I - H - J - A - C - D - E - B - F - G - N - M - U - S - Q - P - K	474.7
L	L - T - O - K - R - I - H - J - A - C - D - E - B - F - G - N - M - U - S - Q - P - L	477
M	M - N - R - I - H - J - A - C - D - E - B - F - G - K - O - L - T - U - S - Q - P - M	524
N	N - R - I - H - J - A - C - D - E - B - F - G - K - O - L - T - M - U - S - Q - P - N	513
O	O - K - R - I - H - J - A - C - D - E - B - F - G - N - M - U - L - T - P - S - Q - O	493.9
P	P - K - O - L - T - R - I - H - J - A - C - D - E - B - F - G - N - M - U - S - Q - P	474.7
Q	Q - S - F - B - E - D - C - A - J - G - H - I - R - N - M - U - L - T - O - K - P - Q	387.1
R	R - I - H - J - A - C - D - E - B - F - G - N - M - U - L - T - O - K - P - S - Q - R	422.6
S	S - Q - F - B - E - D - C - A - J - G - H - I - R - N - M - U - L - T - O - K - P - S	390.7
T	T - L - O - K - R - I - H - J - A - C - D - E - B - F - G - N - M - U - S - Q - P - T	417.6
U	U - M - N - R - I - H - J - A - C - D - E - B - F - G - K - O - L - T - P - S - Q - U	468.2

Table 6. Time Travel (in Minute) of 21 Locations of BNI Branches

The tarting and terminal vertex	The Tour	Total distance (km)
A	A - C - D - E - B - F - G - J - H - I - R - N - M - U - L - T - O - K - P - S - Q - A	400.6
B	B - E - D - C - A - J - G - H - I - R - N - M - U - L - T - O - K - F - S - Q - P - B	433.7
C	C - D - E - A - J - G - B - F - H - I - R - N - M - U - L - T - O - K - P - S - Q - C	410.4
D	D - E - C - A - J - G - B - F - H - I - R - N - M - U - L - T - O - K - P - S - Q - D	410.3
E	E - D - C - A - J - G - B - F - H - I - R - N - M - U - L - T - O - K - P - S - Q - E	409.1
F	F - B - E - D - C - A - J - G - H - I - R - N - M - U - L - T - O - K - P - S - Q - F	390.5
G	G - A - C - D - E - B - F - J - H - I - R - N - M - U - L - T - O - K - P - S - Q - G	401.3
H	H - I - C - D - E - A - J - G - B - F - R - N - M - U - L - T - O - K - P - S - Q - H	420.6
I	I - H - J - A - C - D - E - B - F - G - R - N - M - U - L - T - O - K - P - S - Q - I	413.5
J	J - A - C - D - E - B - F - G - H - I - R - N - M - U - L - T - O - K - P - S - Q - J	405.3
K	K - O - L - T - R - I - H - J - A - C - D - E - B - F - G - N - M - U - S - Q - P - K	474.4
L	L - T - O - K - R - I - H - J - A - C - D - E - B - F - G - N - M - U - S - Q - P - L	476.7
M	M - N - R - I - H - J - A - C - D - E - B - F - G - K - O - L - T - U - S - Q - P - M	523.8
N	N - R - I - H - J - A - C - D - E - B - F - G - K - O - L - T - M - U - S - Q - P - N	512.7
O	O - K - R - I - H - J - A - C - D - E - B - F - G - N - M - U - L - T - P - S - Q - O	493.5
P	P - K - O - L - T - R - I - H - J - A - C - D - E - B - F - G - N - M - U - S - Q - P	474.4
Q	Q - S - F - B - E - D - C - A - J - G - H - I - R - N - M - U - L - T - O - K - P - Q	386.9
R	R - I - H - J - A - C - D - E - B - F - G - N - M - U - L - T - O - K - P - S - Q - R	422.4
S	S - Q - F - B - E - D - C - A - J - G - H - I - R - N - M - U - L - T - O - K - P - S	390.5
T	T - L - O - K - R - I - H - J - A - C - D - E - B - F - G - N - M - U - S - Q - P - T	417.3
U	U - M - N - R - I - H - J - A - C - D - E - B - F - G - K - O - L - T - P - S - Q - U	467.9



All the tours obtained contain crossing path, therefore we revised the tour with the similar process as on Figure 4. Note that the tour for Euclidean dan Haversine data are the same. Figure 2 shows the revision tour implemented on Euclidean distance data and using Haversine distance data, while Figure 2 displays the

Table 7. Time Travel (in Minute) of 21 Locations of BNI Branches

The starting and terminal vertex	The Tour	Total distance (km)
A	A - G - J - C - D - E - B - F - H - I - R - N - M - U - K - O - L - T - S - Q - P - A	915
B	B - D - E - C - A - G - J - H - I - R - N - M - U - F - S - Q - K - O - L - T - P - B	948
C	C - D - E - A - G - J - H - I - R - B - F - S - Q - K - O - L - T - N - M - U - P - C	1027
D	D - E - C - A - G - J- H - I - R - B - F - S - Q - K - O - L - T - N - M - U - P - D	1028
E	E - D - C - A - G - J - H - I - R - B - F - S - Q - K - O - L - T - N - M - U - P - E	1028
F	F - G - A - C - D - E - B - J - H - I - R - N - M - U - K - O - L - T - S - Q - P - F	897
G	G - A - C - D - E - B - F - J - H - I - R - N - M - U - K - O - L - T - S - Q - P - G	880
H	H - I - C - D - E - A - G - J - B - F - R - N - M - U - K - O - L - T - S - Q - P - H	935
I	I - H - J - A - G - C - D - E - B - F - R - N - M - U - K - O - L - T - S - Q - P - I	928
J	J - A - G - C - D - E - B - F - H - I - R - N - M - U - K - O - L - T - S - Q - P - J	914
K	K - O - I - H - J - A - G - C - D - E - B - F - R - N - M - U - L - T - S - Q - P - K	888
L	L - T - O - K - R - H - I - C - D - E - A - G - J - B - F - S - Q - P - N - M - U - L	893
M	M - N - I - H - J - A - G - C - D - E - B - F - R - K - O - L - T - S - Q - P - U - M	927
N	N - I - H - J - A - G - C - D - E - B - F - R - K - O - L - T- S - Q - P - M - U - N	925
O	O - K - R - H - I - C - D - E - A - G - J - B - F - S - Q - P - N - M - U - L - T - O	893
P	P - K - O - I - H - J - A - G - C - D - E - B - F - R - N - M - U - L - T - S - Q - P	888
Q	Q - S - F - G - A - C - D - E - B - J - H - I - R - N - M - U - K - O - L - T - P - Q	874
R	R - H - I - C - D - E - A - G - J - B - F - S - Q - K - O - L - T - N - M - U - P - R	1028
S	S - Q - B - A - C - D - E - B - F - J - H - I - R - N - M - U - K - O - L - T - P - S	887
T	T - L - O - K - R - H - I - C - D - E - A - G - J - B - F - S - Q - P - N - M - U - T	941
U	U - M - N - I - H - J - A - G - C - D - E - B - F - R - K - O - L - T - S - Q - P - U	927

Table 8. Time Travel (in Minute) of 21 Locations of BNI Branches

Data Types	Formed Tour	Total Weight
Euclidean	Q - S - F - B - E - D - C - A - J - G - H - I - R - N - M - U- T - L - O - K - P - Q	414.7
Haversine	Q - S - F - B - E - D - C - A - J - G - H - I - R - N - M - U - T - L - O - K - P - Q	414.5
Time	Q - S - F - G - A - J - C - D - E - B - R - H - I - N - M - U - T - L - O - K - P - Q	846

Table 9. Time Travel (in Minute) of 21 Locations of BNI Branches

Data Types	Formed Tour	Total Weight
Euclidean	Q - S - F - B - E - D - C - A - J - G - H - I - R - N - M - U- L - T - O - K - P - Q	387.1
Haversine	Q - S - F - B - E - D - C - A - J - G - H - I - R - N - M - U - L - T - O - K - P - Q	386.9
Waktu	Q - S - F - G - A - J - C - D - E - B - I - H - R - N - M - U - L - T - O - K - P - Q	793

revision tour implemented on travel time data. Table 9 shows the new tour after revision for all implemented data.

4. CONCLUSIONS

Based on the results and discussion in this study, manual TSP calculations using the Nearest Neighbor Heuristic Algorithm produced distances of 387.1 km for the Euclidean data, 386.9 km for the Haversine data, and 874 minutes for the time data. Because there was intersection in the tour with travel time data, revision of the tour was made, resulting in a total time of 866 minutes. Meanwhile, the Modified Sollin's Algorithm with Euclidean distance data yielded results of 414.7 km, 414.5 km with Haversine distance data, and 846 minutes for travel time data. The tours also had intersection edges. After corrections were made, the best tours for each of the Euclidean distance, Haversine distance, and time data sets were 387.1 km, 386.9 km, and 793 minutes, respectively. Thus, it can be concluded that the Nearest Neighbor Heuristic Algorithm and the Modified Sollin's Algorithm both produced the same solution for Euclidean distance data and Haversine distance data. Meanwhile, for time data, the Modified Sollin Algorithm is better than the Nearest Neighbor Heuristic Algorithm.

5. ACKNOWLEDGEMENT

We would like to express our sincere gratitude to Operations Research team at Study Program of Mathematics and Mathematics and Statistics Research Laboratory Universitas Lampung for the support given.

REFERENCES

[1] Besan A AlSalibi, Marzieh Babaeian Jelodar, and Ibrahim Venkat. A Comparative Study between the Nearest Neighbor and Genetic Algorithms: A Revisit to the Traveling Salesman Problem. *International Journal of Computer Science and Electronics Engineering (IJCSEE)*, 1(1):110–123, 2013.

[2] Rizki Putra Sinaga and Faridawaty Marpaung. Perbandingan Algoritma Cheapest Insertion Heuristic dan Nearest Neighbor Dalam Menyelesaikan Traveling Salesman Problem. *Jurnal Riset Rumpun Matematika dan Ilmu Pengetahuan Alam*, 2(2):238–247, 2023.

[3] Abdi Restu Dinata, Wamiliana Wamiliana, Muslim Ansori, Fitriani Fitriani, and Notiragayu Notiragayu. Determining the Shortest Tour Location of Tourist Attractions in Bandar Lampung Using Cheapest Insertion Heuristic (CIH) and Modified Sollin Algorithm. *Jurnal Pepadun*, 6(1):92–102, 2025.

[4] Meli Amelia, Ilma Isyahna Sholeha, Yanda Rico Revangga, and Wamiliana Wamiliana. The Comparison of Brute Force, Cheapest-Insertion, and Nearest-Neighbor Heuristics for Determining the Shortest Tour for Visiting Malls in Bandar Lampung. *EXPERT*, 14(1):51–54, 2024.

[5] Sriyani Violina. Analysis of Brute Force and Branch & Bound Algorithms to Solve the Traveling Salesperson Problem (TSP). *Turkish Journal of Computer and Mathematics Education*, 12(8):1226–1229, 2021.

[6] Annisa Salsabila, Nadhir Rotun Nikmah, Rafif Syadid Bakhtiananda, Micelle Yap Aswin, and Dina Eka Nurvazly. Products Distribution from Suppliers to Retailers in Bandarlampung City (Case Study: Retailers location in Teluk Betung). *Integra: Journal of Integrated Mathematics and Computer Science*, 1(1):6–12, 2024.

[7] Nur Wafiqoh Hadi, Rehsya Nurfabella, Mira Mustika, et al. Comparative Analysis of CIH and Christofides Algorithms for Optimal Tourist Route Planning in West Java. *Integra: Journal of Integrated Mathematics and Computer Science*, 2(2):56–62, 2025.

[8] Micelle Yap Aswin, Wamiliana Wamiliana, Fitriani Fitriani, Muslim Ansori, and Notiragayu Notiragayu. Perbandingan Cheapest Insertion Heuristic dan Algoritma Christofides Untuk Menentukan Tour Pasar Tradisional di Kota Bandar Lampung. *Jurnal Pepadun*, 5(2):182–194, 2024.

[9] Farid Fargiana, Respitawulan Respitawulan, Yusuf Fajar, Didi Suhaedi, and Erwin Harahap. Implementation of Cheapest Insertion Heuristic Algorithm in Determining Shortest Delivery Route. *International Journal of Global Operations Research*, 3(2):37–45, 2022.

[10] Sameer A Nene and Shree K Nayar. A Simple Algorithm for Nearest Neighbor Search in High Dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):989–1003, 2002.

[11] Md Ziaur Rahman, Sakibur Rahamn Sheikh, Ariful Islam, and Md Azizur Rahman. Improvement of the Nearest Neighbor Heuristic Search Algorithm for Traveling Salesman Problem. *Journal of Engineering Advancements*, 5(01):19–26, 2024.

[12] Anie Lusiani, Siti Samsiyah Purwaningsih, and Euis Sartika. TSP Method Using Nearest Neighbor Algorithm at PT. J&T Express in Bandung. *Jurnal Lebesgue: Jurnal Ilmiah Pendidikan Matematika, Matematika dan Statistika*, 4(3):1560–1568, 2023.

[13] Stefan Hougardy and Mirko Wilde. On the Nearest Neighbor Rule for the Metric Traveling Salesman Problem. *Discrete Applied Mathematics*, 195:101–103, 2015.

[14] Dwi Rizka Amelia Putri, Niken Sabella Oktavia, Siti Laelatul Chasanah, Riza Sawitri, and Felicia Andrade Paskalia. Implementation of Christofides Algorithm to Determine the Shortest Tour of Some Hospitals in Palembang City. *Integra: Journal of Integrated Mathematics and Computer Science*, 2(1):15–19, 2025.

[15] Zhou Xu and Brian Rodrigues. An Extension of the Christofides Heuristic for the Generalized Multiple Depot Multiple Traveling Salesmen Problem. *European Journal of Operational Research*, 257(3):735–745, 2017.

[16] Noraini Mohd Razali, John Geraghty, et al. Genetic Algorithm Performance with Different Selection Strategies in Solving TSP. In *Proceedings of the World Congress on Engineering*, volume 2, pages 1–6. International Association of Engineers Hong Kong, China, 2011.

[17] Darius Bethel and Hakki Erhan Sevil. Revisiting Traveling Salesman Problem (TSP): Analysis of GA and SA Based Solutions. *International Journal of Recent Contributions from Engineering, Science & IT*, 9:44–56, 2021.

[18] Ulviye Hacizade and I Kaya. GA Based Traveling Salesman Problem Solution and Its Application to Transport Routes Optimization. *IFAC-PapersOnLine*, 51(30):620–625, 2018.

[19] AB Doumi, BA Mahafzah, and H Hiary. Solving Traveling Salesman Problem Using Genetic Algorithm Based on Efficient Mutation Operator. *Journal of Theoretical and Applied Information Technology*, 99(15):3768–3781, 2021.

[20] David B Fogel. An Evolutionary Approach to the Traveling Salesman Problem. *Biological Cybernetics*, 60(2):139–144, 1988.

[21] Mohammad Asim, Ritika Gopalia, and Shivalika Swar. Traveling Salesman Problem Using Genetic Algorithm. *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, 3(3):183–190, 2014.

[22] Xuan-Shi Yao, Yun Ou, and Kai-Qing Zhou. TSP Solving Utilizing Improved Ant Colony Algorithm. In *Journal of Physics: Conference Series*, volume 2129, page 012026. IOP Publishing, 2021.

[23] Marco Dorigo and Luca Maria Gambardella. Ant Colonies for the Travelling Salesman Problem. *Biosystems*, 43(2):73–81, 1997.

[24] Eka Ardhianto, Endang Lestariningsih, Widiyanto Tri Handoko, and R Soelistijadi. Pemanfaatan Peta Google Sebagai Sarana Promosi Melalui Penandaan Lokasi Usaha Rumahan Di Paguyuban Warga Lentera Prigi Mranggen Demak. *Jur-*

nal Pengabdian Masyarakat Intimas (Jurnal INTIMAS): Inovasi Teknologi Informasi dan Komputer Untuk Masyarakat, 2(2):78–83, 2022.

[25] Ricky Ramadhan Harahap and Hidayatullah Hidayatullah. Sistem Informasi Google Maps Dengan Menggunakan VB. NET. *Journal of Science and Social Research*, 1(1):36–41, 2018.

[26] Wamiliana Wamiliana. Minimum Spanning Tree dan Desain Jaringan, 2022.

[27] Canggih Ajika Pamungkas. Aplikasi Penghitung Jarak Koordinat Berdasarkan Latitude Dan Longitude Dengan Metode Euclidean Distance Dan Metode Haversine. *Jurnal Informa: Jurnal Penelitian dan Pengabdian Masyarakat*, 5(2):8–13, 2019.

[28] Yulianto Yulianto, Ramadiani Ramadiani, and Awang Harsa Kridalaksana. Penerapan Formula Haversine Pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal. *Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer*, 13(1):14, 2018.