Journal of Mathematical Sciences and Optimization ………… e-ISSN: 3025-3926
Vol. 1. No. 2 (2024), pp. 44-52

Comparison of the Dijkstra's Algorithm and the Floyd-Warshall's Algorithm to Determine the Shortest Path between Hospitals in Several Cities in Lampung Province

Alenia Daynur Putriani[1], Wamiliana[1*], and Siti Laylatul Chasanah[1]

[1]Jurusan Matematika, Universitas Lampung, Bandar Lampung 35145, Indonesia
*Corresponding author's email: Wamiliana.1963@fmipa.unila.ac.id

Abstract
Determining the shortest route from one node to another is a problem we often encounter in everyday life. The routes obtained are intended to minimize costs, travel time or distance. In this study, we compare Dijkstra's and Floyd-Warshall's algorithms to determine the shortest route between hospitals in some cities in Lampung Province. The efficiency of the two algorithms in solving this problem is assessed based on the program running time. The results show that both Dijkstra's algorithm and Floyd-Warshall's algorithm provide the shortest path in the same distance. However, the running time of the Dijkstra's algorithm takes less time than Floyd-Warshall's algorithm.

Keywords: *shortest path, Dijkstra's algorithm, Floyd-Warshall's algorithm, running time*

1. INTRODUCTION

A hospital is an institution that operates in the field of health services. The existence of a hospital is very important because it can help treat people who are sick. The hospital provides outpatient, inpatient, emergency services, a knowledge and technology transfer center and they also function as a referral center. During the Covid-19 pandemic, many people underwent healing treatment in hospitals and there were also those who underwent healing treatment at home by self-isolating because the hospitals they wanted to visit were full occupied. When the hospital visited for treatment is full, the person looks for another available hospital with the fastest route.

One of the problems often encountered in everyday life is finding the shortest path. This problem can be modeled into a weighted graph with values on each edge representing the problem to be solved. In previous research, the Djikstra's algorithm was implemented to determine the shortest path to museums in Jakarta. The research results show that the route in this algorithm is very effective when driven by car and ignoring traffic jams and *odd-even conditions* in Jakarta [1]. Furthermore, research has also been carried out on the comparison of the Floyd-Warshall's algorithm and the Djikstra's algorithm to determine the nearest hospital routes for web-based traffic accident evacuation. It was found that the system was successfully operated with the result that the Djikstra's algorithm used less memory, and the processing time for determining the path was faster than the Floyd-Warshall's algorithm, but the Floyd-Warshall's algorithm produced a shorter path [2].

In this paper we study the determination of the shortest path between hospitals in several cities in Lampung Province using the Djikstra's algorithm and the Floyd-Warshall's algorithm using a weighted graph.

2. THE METHOD

Graph

A graph G is a pair of sets (V, E),, written with the notation $G = (V, E)$, where V is a nonempty set of points and E is a set of lines connecting a pair of points in G. A graph consisting only of a point without sides is called a null graph. A graph in which each line is assigned a value (weight) is called a weighted graph. Points on a graph can be used to represent cities, computers, train stations, airports, etc., while lines connecting two points can represent roads, cables, train tracks, flight routes, etc. The weights on each line can vary depending on the problem being modeled with the graph. Weight can express the distance between two cities, travel time, travel costs, and so on

Shorthest Path Determination

The shortest path problem is one of the most classic algorithmic problems in graph theory, which has the goal of determining the shortest path between two or more points in a path [3]. Determining the shortest path (shortest path problem) is the problem of determining which path to traverse so that the shortest total weight from the starting point to the destination is obtained.The shortest path problem is one of the most classic algorithmic problems in graph theory, which has the goal of determining the shortest path between two or more points in a path [3]. Determining the shortest path (shortest path problem) is the problem of determining which path to traverse so that the shortest total weight from the starting point to the destination is obtained. The graph used in determining the shortest path is a weighted graph. There are several types of shortest path problems, including: shortest path between two points, shortest path between all pairs of points, shortest path from a certain point to all other points and shortest path between points that pass through certain points [4].

Djikstra's algorithm

Djikstra's algorithm is an algorithm that can be used to determine a path connecting two points on a weighted graph, in this case the weight is positive or zero. Disjktra's algorithm can generally be described as follows [5]:

Let $\mathcal{J} \coloneqq$ the set of points where the shortest path has been obtained, and $a \coloneqq$ initial point.

Initiation: $\mathcal{J} = \{a\}$, $d_{aa} = 0$ (the distance from point a to itself is 0), $d_{ai} = c_{ai}$ for all $i \neq a$ (direct distance from point a to point $i \neq a$).

Step A: (Determining the closest point i).
Determine $i \notin \mathcal{J}$ so that $d_{ai} = \min d_j$ for $j \notin \mathcal{J}$.
Add i into $\mathcal{J}$.
If $\mathcal{J}$ contains all points, terminate the process.

Step B: (Updating the shortest distance)
For all points $j \notin \mathcal{J}$
$d_{aj} = \min \{d_{aj}, d_{ai} + c_{ij}\}$ (smallest weight from a to j through point i in $\mathcal{J}$).
Back to Step A.

Floyd-Warshall's algorithm

The Floyd-Warshall's algorithm is a variant of dynamic programming, namely a method that solves problems by viewing the solution to be obtained as an interrelated decision [6]. The steps taken to determine the shortest path using the Floyd-Warshall's algorithm are as follows [7]:

1. Form an adjacent matrix using weight/distance data from the points in G that will be observed.
2. Update the matrix by using each point k in G as an intermediate point, for all $k \in G$, $d_{ij} = \min \{d_{ij}, d_{ik} + d_{kj}\}$; update distances/weights. In this process, the points are observed one by one and the distances/weights are updated to get the shortest path for each pair of points.
3. For each pair of points (i, j), there are two possible cases:
   a. k is not an intermediate point of the shortest path (i, j), so the value d_{ij} does not contain point k.
   b. k is an intermediate point of the shortest path (i, j), then the value $d_{ij} = d_{ik} + d_{kj}$.

Data used

The data used are the ones obtained from Google maps with a total of 42 hospitals in several cities in Lampung Province and there are 49 hospital routes. The data on hospitals whose distances were observed are given in the Table

Table 1. Data Rumah Sakit di Beberapa Kota di Provinsi Lampung

Points	Hospitas	Points	Hospitas	Points	Hospitas
v_1	RSUD Dr. H. Abdoel Moeloek	v_{15}	RSUD Bandar Negara Husada	v_{29}	RSU Permata Hati Way Jepara
v_2	RS Tk IV Bandar Lampung	v_{16}	RS Natar Medika	v_{30}	RS AKAMedika Sribhawono
v_3	RS Advent Bandar Lampung	v_{17}	RS Umbul Thoif	v_{31}	RSU Mitra Husada
v_4	RS Urip Sumoharjo	v_{18}	RSUD Bob Bazar	v_{32}	RSU Surya Asih
v_5	RS Bumi Waras	v_{19}	RSU Gladish Medical Center	v_{33}	RSU Wisma Rini Pringsewu
v_6	RSUD Dr. A. Dadi Tjokrodipo	v_{20}	RSUD Pesawaran	v_{34}	RSUD Pringsewu
v_7	RS Bhayangkara	v_{21}	RSU Muhammadiyah Metro	v_{35}	RSU Az-Zahra Kalirejo
v_8	RS Graha Husada	v_{22}	RS Mardi Waluyo	v_{36}	RSU Kartini
v_9	RS Imanuel Way Halim	v_{23}	RSUD Ahmad Yani	v_{37}	RSUD Demang Sepulau Raya
v_{10}	RS Pertamina Bintang Amin	v_{24}	RSU IslamMetro	v_{38}	RS Harapan Bunda
v_{11}	RSU Hermina	v_{25}	RS Azizah	v_{39}	RS Islam Asy-Syifa Bandar Jaya
v_{12}	RS Budi Medika	v_{26}	RSU Permata Hati Metro	v_{40}	RSU Yukum Medical Center
v_{13}	RS Jendral Sudirman	v_{27}	RS Sumber Sari	v_{41}	RS Panti Secanti
v_{14}	RS Airan Raya	v_{28}	RSUD Sukadana	v_{42}	RSUD Batin Mangunang Kota Agung

1. RESULTS AND DISCUSSIONS

Graph Modeling from Data

Data on hospital locations and routes in several cities in Lampung Province can be formed into a weighted graph. The graph is formed from 42 points representing each hospital with the weight being the distance between hospitals, and v_1 (RSUD Dr. H. Abdoel Moeloek) is the starting point. Figure 1 represents the graph form of the 42 points.

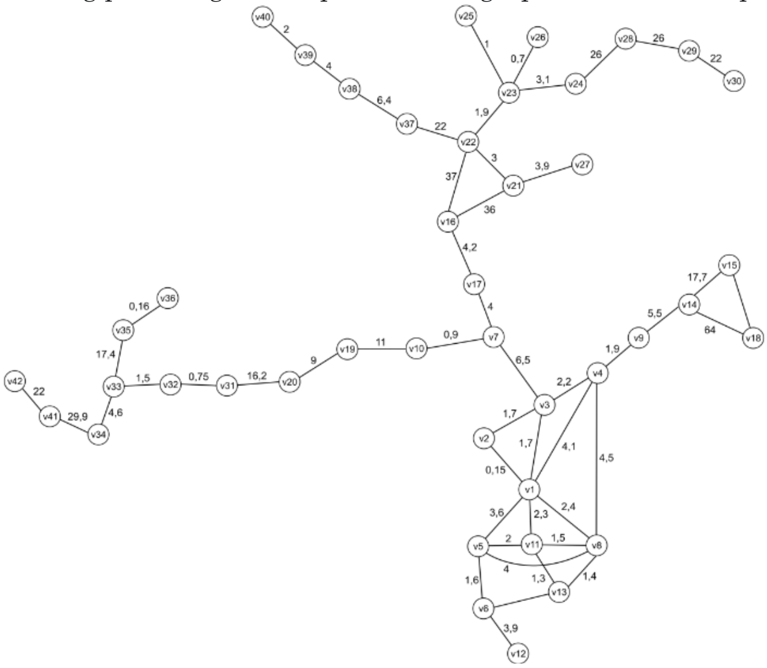

Figure 1. Graph modeling of hospital locations in several cities in Lampung Province

Application of Dijkstra's Algorithm

In this study, we look for the shortest path from RSUD Dr. H. Abdoel Moeloek (v_1) headed to other hospitals in several cities in Lampung Province. Based on the steps of the Dijkstra's algorithm in this case, the iteration process is carried out 41 times and the shortest path from RSUD Dr. H. Abdoel Moeloek headed to 41 other hospitals. Due to space limitations, Table 2 presents the first 4 iterations of the 41 iterations carried out.

Table 2. The first four iterations of Dijkstra's algorithm

Iteration	V	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	$v1_2$	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}	v_{18}	v_{19}	v_{20}	v_{21}	v_{22}
0	{1}	15	1,7	4,1	3,6	∞	∞	2,4	∞	∞	2,3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	{1, 2}	15	1,7	4,1	3,6	∞	∞	2,4	∞	∞	2,3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	{1, 2, 3}	15	1,7	3,9	3,6	∞	8,2	2,4	∞	∞	2,3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
3	{1, 2, 3, 11}	15	1,7	3,9	3,6	∞	8,2	2,4	∞	∞	2,3	∞	3,6	∞	∞	∞	∞	∞	∞	∞	∞	∞
4	{1, 2, 3, 8, 11	15	1,7	3,9	3,6	∞	8,2	2,4	∞	∞	2,3	∞	3,6	∞	∞	∞	∞	∞	∞	∞	∞	∞

Iteration	V	v_{23}	v_{24}	v_{25}	v_{26}	v_{27}	v_{28}	v_{29}	v_{30}	v_{31}	v_{32}	v_{33}	v_{34}	v_{35}	v_{36}	v_{37}	v_{38}	v_{39}	v_{40}	v_{41}	v_{42}
0	{1}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	{1, 2}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	{1, 2, 3}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
3	{1, 2, 3,11}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
4	{1, 2, 3, 8, 11	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

Notes:

: Selected points and weights

Applications of Floyd-Warshall's Algorithm

The Floyd-Warshall's algorithm is also an algorithm that has the same goal as the Dijkstra's algorithm, namely finding the shortest path from the starting point to the destination point. In this case, the shortest route from RSUD Dr. H. Abdoel Moeloek goes to other hospitals in several cities in Lampung Province. Based on the steps of the Floyd-Warshall's algorithm, a 42×42 matrix is formed, then an iteration process is carried out starting from the 1st to 42nd iteration and the final matrix is obtained, which is the smallest weight among all hospitals as shown in Table 3.

Table 3. 42nd iteration of Floyd-Warshall

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	0,15	1,7	3,9	3,6	5,2	8,2	2,4	5,8	9,1	2,3	9,1	3,6	11,3	29	16,4	12,2	75,3	20,1	29,1	52,4
2	0,15	0	1,7	3,9	3,75	5,35	8,2	2,55	5,8	9,1	2,45	9,25	3,75	11,3	29	16,4	12,2	75,3	20,1	29,1	52,4
3	1,7	1,7	0	2,2	5,3	6,9	6,5	4,1	4,1	7,4	4	10,8	5,3	9,6	27,3	14,7	10,5	73,6	18,4	27,4	50,7
4	3,9	3,9	2,2	0	7,5	9,1	8,7	4,5	1,9	9,6	6,2	13	5,9	7,4	25,1	16,9	12,7	71,4	20,6	29,6	52,9
5	3,6	3,75	5,3	7,5	0	1,6	11,8	3,5	9,4	12,7	2	5,5	3,3	14,9	32,6	20	15,8	78,9	23,7	32,7	56
6	5,2	5,35	6,9	9,1	1,6	0	13,4	5,1	11	14,3	3,6	3,9	4,9	16,5	34,2	21,6	17,4	80,5	25,3	34,3	57,6
7	8,2	8,2	6,5	8,7	11,8	13,4	0	10,6	10,6	0,9	10,5	17,3	11,8	16,1	33,8	8,2	4	80,1	11,9	20,9	44,2
8	2,4	2,55	4,1	4,5	3,5	5,1	10,6	0	6,4	11,5	1,5	9	1,4	11,9	29,6	18,8	14,6	75,9	22,5	31,5	54,8
9	5,8	5,8	4,1	1,9	9,4	11	10,6	6,4	0	11,5	8,1	14,9	7,8	5,5	23,2	18,8	14,6	69,5	22,5	31,5	54,8
10	9,1	9,1	7,4	9,6	12,7	14,3	0,9	11,5	11,5	0	11,4	18,2	12,7	17	19,4	9,1	4,9	65,7	11	20	45,1
11	2,3	2,45	4	6,2	2	3,6	10,5	1,5	8,1	11,4	0	7,5	1,3	13,6	31,3	18,7	14,5	77,6	22,4	31,4	54,7
12	9,1	9,25	10,8	13	5,5	3,9	17,3	9	14,9	18,2	7,5	0	8,8	20,4	38,1	25,5	21,3	84,4	29,4	38,4	61,5
13	3,6	3,75	5,3	5,9	3,3	4,9	11,8	1,4	7,8	12,7	1,3	8,8	0	13,3	31	20	15,8	77,3	23,7	32,7	56
14	11,3	11,3	9,6	7,4	14,9	16,5	16,1	11,9	5,5	17	13,6	20,4	13,3	0	17,7	33,3	29,1	64	28	37	69,3
15	29	29	27,3	25,1	32,6	34,2	33,8	29,6	23,2	19,4	31,3	38,1	31	17,7	0	51	46,8	81,7	45,7	54,7	87
16	16,4	16,4	14,7	16,9	20	21,6	8,2	18,8	18,8	9,1	18,7	25,5	20	33,3	51	0	4,2	97,3	20,1	29,1	36
17	12,2	12,2	10,5	12,7	15,8	17,4	4	14,6	14,6	4,9	14,5	21,3	15,8	29,1	46,8	4,2	0	93,1	15,9	24,9	40,2
18	75,3	75,3	73,6	71,4	78,9	80,5	80,1	75,9	69,5	65,7	77,6	84,4	77,3	64	81,7	97,3	93,1	0	92	101	133,3
19	20,1	20,1	18,4	20,6	23,7	25,3	11,9	22,5	22,5	11	22,4	29,4	23,7	28	45,7	20,1	15,9	92	0	9	51,9
20	29,1	29,1	27,4	29,6	32,7	34,3	20,9	31,5	31,5	20	31,4	38,4	32,7	37	54,7	29,1	24,9	101	9	0	65,1
21	52,4	52,4	50,7	52,9	56	57,6	44,2	54,8	54,8	45,1	54,7	61,5	56	69,3	87	36	40,2	133,3	51,9	65,1	0

22	53,4	53,4	51,7	53,9	57	58,6	45,2	55,8	55,8	46,1	55,7	62,5	57	70,3	88	37	41,2	134,3	52,9	66,1	3
23	55,3	55,3	53,6	55,8	58,9	60,5	47,1	57,7	57,7	48	57,6	64,4	58,9	72,2	89,9	38,9	43,1	136,2	54,8	68	4,9
24	58,4	58,4	56,7	58,9	62	63,6	50,2	60,8	60,8	51,1	60,7	67,5	62	75,3	93	42	46,2	139,3	57,9	71,1	8
25	56,3	56,3	54,6	56,8	59,9	61,5	48,1	58,7	58,7	49	58,6	65,4	59,9	73,2	90,9	39,9	44,1	137,2	55,8	69	5,9
26	56	56	54,3	56,5	59,6	61,2	47,8	58,4	58,4	48,7	58,3	65,1	59,6	72,9	90,6	39,6	43,8	136,9	55,5	68,7	5,6
27	56,3	56,3	54,6	56,8	59,9	61,5	48,1	58,7	58,7	49	58,6	65,4	59,9	73,2	90,9	39,9	44,1	137,2	55,8	69	3,9
28	84,4	84,4	82,7	84,9	88	89,6	76,2	86,8	86,8	77,1	86,7	93,5	88	101,3	119	68	72,2	165,3	83,9	97,1	34
29	110,4	110,4	108,7	110,9	114	115,6	102,2	112,8	112,8	103,1	112,7	119,5	114	127,3	145	94	98,2	191,3	109,9	123,1	60
30	132,4	132,4	130,7	132,9	136	137,6	124,2	134,8	134,8	125,1	134,7	141,5	136	149,3	167	116	120,2	213,3	131,9	145,1	82
31	45,3	45,3	43,6	45,8	48,9	50,5	37,1	47,7	47,7	36,2	47,6	54,6	48,9	53,2	70,9	45,3	41,1	117,2	25,2	16,2	81,3
32	46,05	46,o5	44,35	46,55	49,65	51,25	37,85	48,45	48,45	36,95	47,95	55,35	49,65	53,95	71,65	46,05	41,85	117,95	25,95	16,95	82,05
33	47,55	47,55	45,85	48,05	51,15	52,75	39,35	49,95	49,95	38,45	49,45	56,85	51,15	55,45	73,15	47,55	43,35	119,45	27,45	18,45	83,55
34	52,15	52,15	50,45	52,65	55,75	57,35	43,95	54,55	54,55	43,05	54,05	61,45	55,75	60,05	77,75	52,15	47,95	124,05	32,05	23,05	88,15
35	64,95	64,95	63,25	65,45	68,55	70,15	56,75	67,35	67,35	55,85	66,85	74,25	68,55	72,85	90,55	64,95	60,75	136,85	44,85	35,85	100,95
36	65,11	65,11	63,41	65,61	68,71	70,31	56,91	67,51	67,51	56,01	67,01	74,41	68,71	73,01	90,71	65,11	60,91	137,01	45,01	36,01	101,11
37	75,4	75,4	73,7	75,9	79	80,6	67,2	77,8	77,8	68,1	77,,7	84,5	79	92,3	110	59	63,2	156,3	74,9	88,1	25
38	81,8	81,8	80,1	82,3	85,4	87	73,6	84,2	84,2	74,5	84,1	90,9	85,4	98,7	116,4	65,4	69,6	162,7	81,3	94,5	31,4
39	85,8	85,8	84,1	86,3	89,4	91	77,6	88,2	88,2	78,5	88,1	94,9	89,4	102,7	120,4	69,4	73,6	116,7	85,3	98,5	35,4
40	87,8	87,8	86,1	88,3	91,4	93	79,6	90,2	90,2	80,5	90,1	96,9	91,4	104,7	122,4	71,4	75,6	168,7	87,3	100,5	37,9
41	82,05	82,05	80,35	82,55	85,65	87,25	73,85	84,45	84,45	72,95	83,95	91,35	85,65	89,95	107,65	82,05	77,85	153,95	61,95	52,95	118,05
42	104,05	104,05	102,35	104,55	107,65	109,25	95,85	106,45	106,45	94,95	105,95	113,35	107,65	111,95	129,65	104,05	99,85	175,95	83,95	74,95	140,05

	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
1	53,4	55,3	58,4	56,3	56	56,3	84,4	110,4	132,4	45,3	46,05	47,55	52,15	64,95	65,11	75,4	81,8	85,8	87,8	82,05	104,05
2	53,4	55,3	58,4	56,3	56	56,3	84,4	110,4	132,4	45,3	46,05	47,55	52,15	64,95	65,11	75,4	81,8	85,8	87,8	82,05	104,05
3	51,7	53,6	56,7	54,6	54,3	54,6	82,7	108,7	130,7	43,6	44,35	45,85	50,45	63,25	63,41	73,7	80,1	84,1	86,1	80,35	102,35
4	53,9	55,8	58,9	56,8	56,5	56,8	84,9	110,9	132,9	45,8	46,55	48,05	52,65	65,45	65,61	75,9	82,3	86,3	88,3	82,55	104,55
5	57	58,9	62	59,9	59,6	59,9	88	114	136	48,9	49,65	51,15	55,75	68,55	68,71	79	85,4	98,4	91,4	85,65	107,65
6	58,6	60,5	63,6	61,5	61,2	61,5	89,6	115,6	137,6	50,5	51,25	52,75	57,35	70,15	70,31	80,6	87	91	93	87,25	109,25
7	45,2	47,1	50,2	48,1	47,8	48,1	76,2	102,2	124,2	37,1	37,85	39,35	43,95	56,75	56,91	67,2	73,6	77,6	79,6	73,85	95,85
8	55,8	57,7	60,8	58,7	58,4	58,7	86,8	112,8	134,8	47,7	48,45	49,95	54,55	67,35	67,51	77,8	84,2	88,2	90,2	84,45	106,45
9	55,8	57,7	60,8	58,7	58,4	58,7	86,8	112,8	134,8	47,7	48,45	49,95	54,55	67,35	67,51	77,8	84,2	88,2	90,2	84,45	106,45
10	46,1	48	51,1	49	48,7	49	77,1	103,1	125,1	36,2	36,95	38,45	43,05	55,85	56,01	68,1	74,5	78,5	80,5	72,95	94,95
11	55,7	57,6	60,7	58,6	58,3	58,6	86,7	112,7	134,7	47,6	47,95	49,45	54,05	66,85	67,01	77,7	84,1	88,1	90,1	83,95	105,95
12	62,5	64,4	67,5	65,4	65,1	65,4	93,5	119,5	141,5	54,6	55,35	56,85	61,45	74,25	74,41	84,5	90,9	94,9	96,9	91,35	113,35
13	57	58,9	62	59,9	59,6	59,9	88	114	136	48,9	49,65	51,15	55,75	68,55	68,71	79	84,5	89,4	91,4	85,65	107,65
14	70,3	72,2	75,3	73,2	72,9	73,2	101,3	127,3	149,3	53,2	53,95	55,45	60,05	72,85	73,01	92,3	98,7	102,7	104,7	89,95	111,95
15	88	89,9	93	90,9	90,6	90,9	119	145	167	70,9	71,65	73,15	77,75	90,55	90,71	110	116,4	120,4	122,4	107,65	129,65
16	37	38,9	42	39,9	39,6	39,9	68	94	116	45,3	46,05	47,55	52,15	64,95	65,11	59	65,4	69,4	71,4	82,05	104,05
17	41,2	43,1	46,2	44,1	43,8	44,1	72,2	98,2	120,2	41,1	41,85	43,35	47,95	60,75	60,91	63,2	69,6	73,6	75,6	77,85	99,85
18	134,3	136,2	139,3	137,2	136,9	137,2	165,3	191,3	213,3	117,2	117,95	119,45	124,05	136,85	137,01	156,3	162,7	166,7	168,7	153,95	175,95
19	52,9	54,8	57,9	55,8	55,5	55,8	83,9	109,9	131,9	25,2	25,95	27,45	32,05	44,85	45,01	74,9	81,3	85,3	87,3	61,95	83,95
20	66,1	68	71,1	69	68,7	69	97,1	123,1	145,1	16,2	16,95	18,45	23,05	35,85	36,01	88,1	94,5	98,5	100,5	52,95	74,95
21	3	4,9	8	5,9	5,6	3,9	34	60	82	81,3	82,05	83,55	88,15	100,95	101,11	25	31,4	35,4	37,9	118,05	140,05
22	0	1,9	5	2,9	2,6	6,9	31	57	79	82,3	83,05	84,55	89,15	101,95	102,11	22	28,4	32,4	34,4	119,05	141,05
23	1,9	0	3,1	1	0,7	8,8	29,1	55,1	77,1	84,2	84,95	86,45	91,05	103,85	104,01	23,9	30,3	34,3	36,3	120,95	142,95
24	5	3,1	0	4,1	3,8	11,9	26	52	74	87,3	88,05	89,55	94,15	106,95	107,11	27	33,4	37,4	39,4	124,05	146,05
25	2,9	1	4,1	0	1,7	9,8	30,1	56,1	78,1	85,2	85,95	87,45	92,05	104,85	105,01	24,9	31,3	35,3	37,3	121,95	143,95
26	2,6	0,7	3,8	1,7	0	9,5	29,8	55,8	77,8	84,9	85,65	87,15	91,75	104,55	104,71	24,6	31	35	37	121,65	143,65
27	6,9	8,8	11,9	9,8	9,5	0	37,9	63,9	85,9	85,2	85,95	87,45	92,05	104,85	105,01	28,9	35,3	39,3	41,3	121,95	143,95
28	31	29,1	26	30,1	29,8	37,9	0	26	48	113,3	114,05	115,55	120,15	132,95	133,11	53	59,4	63,4	65,4	150,05	172,05
29	57	55,1	52	56,1	55,8	63,9	26	0	22	139,3	140,05	141,55	146,15	158,95	159,11	79	85,4	89,4	91,4	176,05	198,05
30	79	77,1	74	78,1	77,8	85,9	48	22	0	161,3	162,05	163,55	168,15	180,95	181,11	101	107,4	111,4	113,4	198,05	220,05
31	82,3	84,2	87,3	85,2	84,9	85,2	113,3	139,3	161,3	0	0,75	2,25	6,85	19,65	19,81	104,3	110,7	114,7	116,7	36,75	58,75
32	83,05	84,95	88,05	85,95	85,65	85,95	114,05	140,05	162,05	0,75	0	1,5	6,1	18,9	19,06	105,05	111,45	115,45	117,45	36	58
33	84,55	86,45	89,55	87,45	87,15	87,45	115,55	141,55	163,55	2,25	1,5	0	4,6	17,4	17,56	105,55	111,95	115,95	117,95	34,5	56,5
34	89,15	91,05	94,15	92,05	91,75	92,05	120,15	146,15	168,15	6,85	6,1	4,6	0	22	22,16	111,15	117,55	121,55	123,55	29,9	51,9
35	101,95	103,85	106,95	104,85	104,55	104,85	132,95	158,95	180,95	19,65	18,9	17,4	22	0	0,16	123,95	130,35	134,35	136,35	51,9	73,9

36	102,11	104,01	107,11	105,01	104,71	105,01	133,11	159,11	181,11	19,81	19,06	17,56	22,16	0,16	0	124,11	130,51	134,51	136,55	52,06	74,06
37	22	23,9	27	24,9	24,6	28,9	53	79	101	104,3	105,05	105,55	111,15	123,95	124,11	0	6,4	104	10,6	145,05	167,05
38	28,4	30,3	33,4	31,3	31	35,3	59,4	85,4	107,4	110,7	111,45	111,95	117,55	130,35	130,51	6,4	0	4	6	147,05	169,45
39	32,4	34,3	37,4	35,3	35	39,3	63,4	89,4	111,4	114,7	115,45	115,95	121,55	134,35	134,51	104	4	0	2	151,45	173,45
40	34,4	36,3	39,4	37,3	37	41,3	65,4	91,4	113,4	116,7	117,45	117,95	123,55	136,35	136,55	10,6	6	2	0	153,45	175,45
41	119,05	120,95	124,05	121,95	121,65	121,95	150,05	176,05	198,05	36,75	36	34,5	29,9	51,9	52,06	145,05	147,05	151,45	153,45	0	22
42	141,05	142,95	146,05	143,95	143,65	143,95	172,05	198,05	220,05	58,75	58	56,5	51,9	73,9	74,06	167,05	169,45	173,45	175,45	22	0

From the application of the Dijkstra's algorithm and the Floyd-Warshall's algorithm, the following results are obtained:

1. Shortest path from $v_1$to v_2 is 0,15 km.
2. Shortest path fromv_1 ke v_3 yaitu 1,7 km.
3. Shortest path from v_1 to v_4 is 3,9 km with paths $v_1 - v_3 - v_4$.
4. Shortest path from dari v_1 to v_5 is 3,6 km.
5. Shortest path from v_1 to v_6 is 5,2 km with paths $v_1 - v_5 - v_6$.
6. Shortest path from v_1 to v_7 is 8,2 km with paths $v_1 - v_3 - v_7$.
7. Shortest path from v_1 to v_8 is 2,4 km.
8. Shortest path from v_1 to v_9 is 5,8 km with paths $v_1 - v_3 - v_4 - v_9$.
9. Shortest path from v_1 to v_{10} is 9,1 km with paths $v_1 - v_3 - v_7 - v_{10}$.
10. Shortest path from v_1 to v_{11} is 2,3 km.
11. Shortest path from v_1 to v_{12} is 9,1 km with paths $v_1 - v_5 - v_6 - v_{12}$.
12. Shortest path from v_1 to v_{13} is 3,6 km with paths $v_1 - v_{11} - v_{13}$.
13. Shortest path from v_1 to v_{14} is 11,3 km with paths $v_1 - v_3 - v_4 - v_9 - v_{14}$.
14. Shortest path from v_1 to v_{15} is 29 km with paths $v_1 - v_3 - v_4 - v_9 - v_{14} - v_{15}$.
15. Shortest path from v_1 to v_{16} is 16,4 km with paths $v_1 - v_3 - v_7 - v_{17} - v_{16}$.
16. Shortest path from v_1 to v_{17} is 12,2 km with paths $v_1 - v_3 - v_7 - v_{17}$.
17. Shortest path from v_1 to v_{18} is 75,3 km with paths $v_1 - v_3 - v_4 - v_9 - v_{14} - v_{18}$.
18. Shortest path from v_1 to v_{19} is 20,1 km with paths $v_1 - v_3 - v_7 - v_{10} - v_{19}$.
19. Shortest path from point v_1 to v_{20} is 29.1 km with paths $v_1 - v_3 - v_7 - v_{10} - v_{19} - v_{20}$.
20. Shortest path from point v_1 to v_{21} is 52.4 km with paths $v_1 - v_3 - v_7 - v_{17} - v_{16} - v_{21}$.
21. Shortest path from point v_1 to v_{22} is 53.4 km with paths $v_1 - v_3 - v_7 - v_{17} - v_{16} - v_{22}$.
22. Shortest path from point v_1 to v_{23} is 55.3 km with paths $v_1 - v_3 - v_7 - v_{17} - v_{16} - v_{22} - v_{23}$.
23. Shortest path from point v_1 to v_{24} is 58.4 km dengan ute $v_1 - v_3 - v_7 - v_{17} - v_{16} - v_{22} - v_{23} - v_{24}$.
24. Shortest path from point v_1 to v_{25} is 56.3 km with paths $v_1 - v_3 - v_7 - v_{17} - v_{16} - v_{22} - v_{23} - v_{25}$.
25. Shortest path from point v_1 to v_{26} is 56 km with paths $v_1 - v_3 - v_7 - v_{17} - v_{16} - v_{22} - v_{23} - v_{26}$.
26. Shortest path from point v_1 to v_{27} is 56.3 km with paths $v_1 - v_3 - v_7 - v_{17} - v_{16} - v_{21} - v_{27}$.
27. Shortest path from point v_1 to v_{28} is 84.4 km with paths $v_1 - v_3 - v_7 - v_{17} - v_{16} - v_{22} - v_{23} - v_{24} - v_{28}$.
28. Shortest path from point v_1 to v_{29} is 110.4 km with paths $v_1 - v_3 - v_7 - v_{17} - v_{16} - v_{22} - v_{23} - v_{24} - v_{28} - v_{29}$.
29. Shortest path from point v_1 to v_{30} is 132.4 km with paths $v_1 - v_3 - v_7 - v_{17} - v_{16} - v_{22} - v_{23} - v_{24} - v_{28} - v_{29} - v_{30}$.
30. Shortest path from point v_1 to v_{31} is 45.3 km with paths $v_1 - v_3 - v_7 - v_{10} - v_{19} - v_{20} - v_{31}$.
31. Shortest path from point v_1 to v_{32} is 46.05 km with paths $v_1 - v_3 - v_7 - v_{10} - v_{19} - v_{20} - v_{31} - v_{32}$.
32. Shortest path from point v_1 to v_{33} is 47.55 km with paths $v_1 - v_3 - v_7 - v_{10} - v_{19} - v_{20} - v_{31} - v_{32} - v_{33}$.
33. Shortest path from point v_1 to v_{34} is 52.15 km with paths $v_1 - v_3 - v_7 - v_{10} - v_{19} - v_{20} - v_{31} - v_{32} - v_{33} - v_{34}$.
34. Shortest path from point v_1 to v_{35} is 64.95 km with paths $v_1 - v_3 - v_7 - v_{10} - v_{19} - v_{20} - v_{31} - v_{32} - v_{33} - v_{35}$.
35. Shortest path from point v_1 to v_{36} is 65.11 km with paths $v_1 - v_3 - v_7 - v_{10} - v_{19} - v_{20} - v_{31} - v_{32} - v_{33} - v_{35} - v_{36}$.
36. Shortest path from point v_1 to v_{37} is 75.4 km with paths $v_1 - v_3 - v_7 - v_{17} - v_{16} - v_{22} - v_{37}$.

37. Shortest path from point v_1 to v_{38} is 81.8 km with paths $v_1 - v_3 - v_7 - v_{17} - v_{16} - v_{22} - v_{37} - v_{38}$.
38. Shortest path from point v_1 to v_{39} is 85.8 km with paths $v_1 - v_3 - v_7 - v_{17} - v_{16} - v_{22} - v_{37} - v_{38} - v_{39}$.
39. Shortest path from point v_1 to v_{40} is 87.8 km with paths $v_1 - v_3 - v_7 - v_{17} - v_{16} - v_{22} - v_{37} - v_{38} - v_{39} - v_{40}$.
40. Shortest path from point v_1 to v_{41} is 82.05 km with paths $v_1 - v_3 - v_7 - v_{10} - v_{19} - v_{20} - v_{31} - v_{32} - v_{33} - v_{34} - v_{41}$.
41. Shortest path from point v_1 to v_{42} is 104.05 km with paths $v_1 - v_3 - v_7 - v_{10} - v_{19} - v_{20} - v_{31} - v_{32} - v_{33} - v_{34} - v_{41} - v_{42}$.

Running Time of Algorithm

Tabel 4. Running Time of Dijkstra's algorithm and Floyd-Warshall's algorithm

Implementation of the	Dijkstra's algorithm (seconds)	Floyd-Warshall's algorithm (seconds)
1	0.0644	0.0762
2	0.0704	0.0688
3	0.0695	0.0829
4	0.0617	0.0734
5	0.0606	0.0866
6	0.0677	0.0712
7	0.0754	0.0833
8	0.0684	0.0899
9	0.0581	0.0740
10	0.0729	0.0685
11	0.0732	0.0675
12	0.0603	0.0833
13	0.0708	0.0686
14	0.0607	0.0881
15	0.0604	0.0733
16	0.0682	0.0840
17	0.0603	0.0806
18	0.0704	0.0845
19	0.0688	0.0760
20	0.0686	0.0767
21	0.0723	0.0757
22	0.0678	0.0707
23	0.0623	0.0870
24	0.0591	0.0806
25	0.0730	0.0820
26	0.0737	0.0736
27	0.0690	0.0697
28	0.0730	0.0700
29	0.0716	0.0731
30	0.0714	0.0877
Average	**0.0674**	**0.0775**

In the process of completing the comparison of the Dijkstra's algorithm and the Floyd-Warshall's algorithm to determine the shortest path between hospitals in several cities in Lampung Province, Python software was also used. By using Python software, the running time of the two algorithms used is obtained, namely for the Dijkstra's algorithm, the average time is 0.0674 seconds, while for the Floyd-Warshall's algorithm, the average time is 0.0775 seconds, as shown in Table 4. Figure 2 and Figure 3 give the output of Dijkstra's algorithm and the Floyd-Warshall's algorithm.

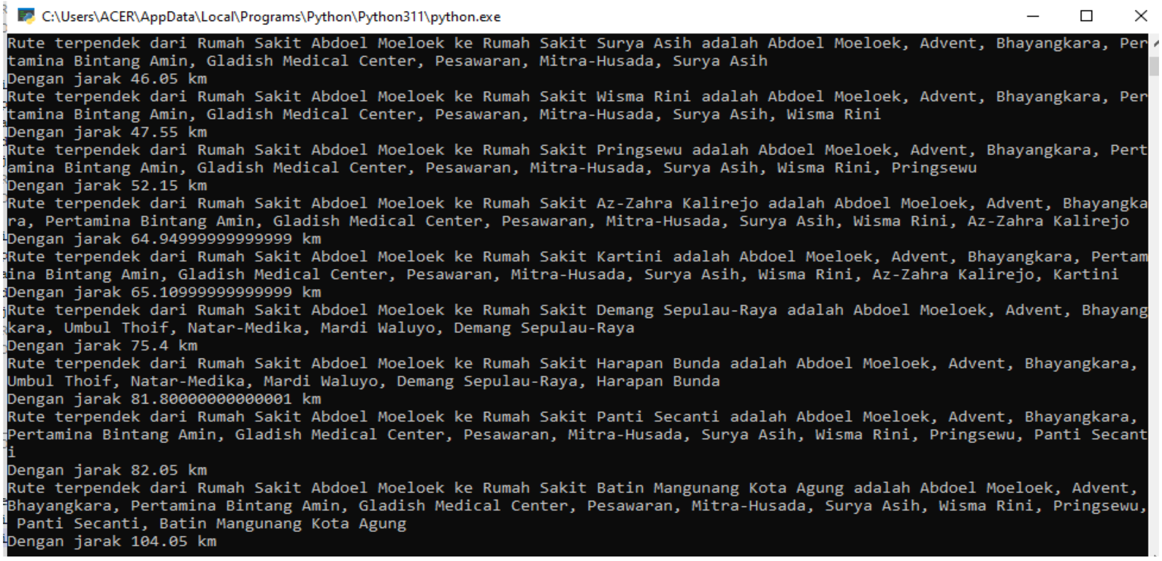

Figure 2. Output of Dijkstra's algorithm

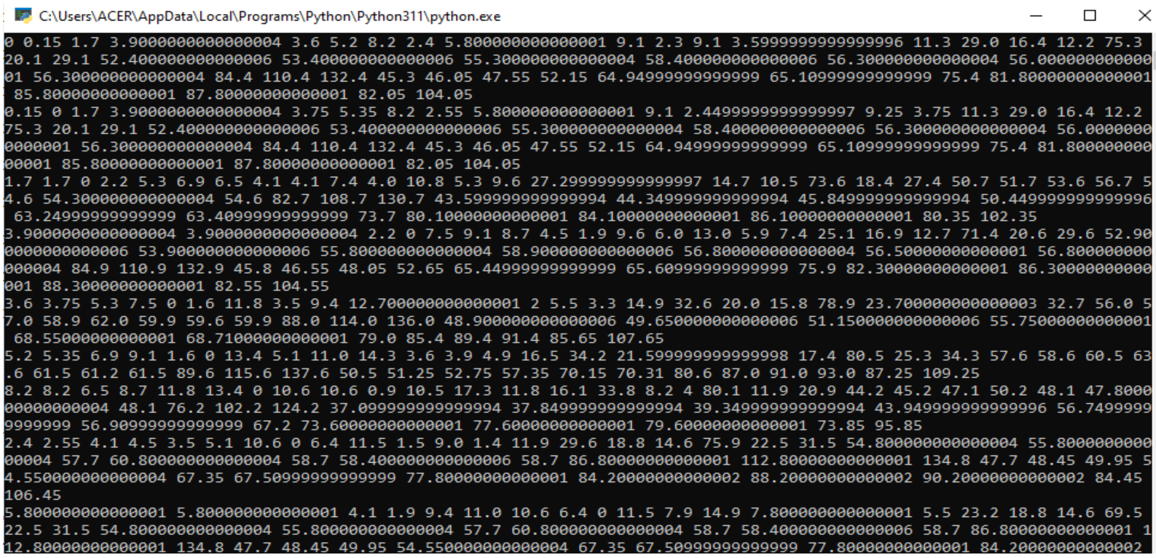

Figure 3. Output of Dijkstra's algorithm

5. CONCLUSION

Based on the results and discussion above, it can be concluded that determining the shortest path between hospitals in several cities in Lampung Province using the Dijkstra's algorithm and the Floyd-Warshall's algorithm produces the same paths and weights. However, using the Dijkstra Algorithm is faster because it does not require too many mechanisms compared to the Floyd-Warshall Algorithm because in the Dijsktra Algorithm you can only determine the shortest path from the starting point to the other points, whereas in the Floyd-Warshall Algorithm you get the shortest path for each pair of points. The running time of the program for the two algorithms used is also obtained using Python software, namely an average of 0.0674 seconds for the Dijkstra's algorithm and an average of 0.0775 seconds for the Floyd Warshall's algorithm. The Floyd-Warshall's algorithm provides more information because it displays the final result in the form of the distance between all hospitals.

REFERENCES

1. Cantona, A., Fauziah, and Winarsih. Impelementasi Algoritma Dijkstra pada Pencarian Lintasan Terpendek ke Museum di Jakarta, 2020, *Jurnal Teknologi and Manajemen Informatika*, **Volume:** 6 pp 27–34.
2. Rudiyanto, A. D., Wahyuddin, M. I., and Andrianingsih, 2020, Perbandingan Algoritma Floyd-Warshall and Dijkstra untuk Menentukan Lintasan Rumah Sakit Terdekat Lintasan Evakuasi Kecelakaan Lalu Lintas Berbasis Web. *Journal of Information Technology and Computer Science*, **Volume**: 3 pp 336–345.
3. Wang, X. Z. The Comparison of Three Algorithms in Shortest Path Issue, 2018, *Journal of Physics Conference Series*, **Volume:** 1087 pp 164–170.
4. Harahap, M. K., and Khairina, N. Pencarian Lintasan Terpendek dengan Algoritma Dijkstra, 2017, *Jurnal & Penelitian Teknik Informatika*, **Volume:** 2 pp 18–23.
5. Aulia, S. R, Wamiliana, Asmiati, and Notiragayu, 2023, Perbandingan Algoritme Dijkstra and Algoritme A * (A-Star) dalam Penentuan Lintasan Terpendek dari Dinas Pendidikan Provinsi Lampung ke Beberapa Sekolah Menengah Atas (SMA) Negeri di Provinsi Lampung. *Jurnal Pepadun*, **Volume**: 4 pp 83 – 190.
6. Darnita, Y., Toyib, R., and Rinaldi, 2017, Implementasi Algoritma Floyd-Warshall untuk Menentukan Letak and Lokasi Perusahaan Travel/Rental Mobil di Kota Bengkulu. *Jurnal Pseudocode*, **Volume:** 4 pp 144–155.
7. https://cp-algorithms.com/graph/all-pair-shortest-path-floyd-warshall.html, accessed in 20 Nov 2023.