

Implementation of Dijkstra Algorithm and Welch-Powell Algorithm for Optimal Solution of Campus Bus Transportation

Nurwan¹, Widya Eka Pranata^{2*}, Muhammad Rezky Friesta Payu³, Nisky Imansyah Yahya⁴

^{1,2,3,4}Department of Mathematics, Universitas Negeri Gorontalo, Gorontalo, Indonesia

Article history:

Received Sep 28, 2020

Revised Feb 19, 2021

Accepted May 30, 2021

Kata Kunci:

Rute terpendek,
Jadwal optimal,
Algoritma Dijkstra,
Algoritma Welch-Powell

Keywords:

Shortest route,
Optimal schedule,
Dijkstra algorithm,
Welch-Powell algorithm

Abstrak. Penelitian ini membahas tentang penerapan algoritma Dijkstra dan algoritma Welch-Powell pada masalah transportasi bus kampus. Tujuan penelitian ini adalah untuk menentukan rute terpendek dan jadwal optimal untuk jalur transportasi bus kampus UNG. Dalam menentukan rute terpendek, setiap persimpangan direpresentasikan sebagai simpul dan jalur yang dilalui direpresentasikan sebagai sisi. Lintasan terpendek diperoleh $V_1 - V_2 - V_5 - V_8 - V_9 - V_{10} - V_{13} - V_{16}$. Dalam menentukan jadwal optimal, jumlah bus merepresentasikan simpul dan waktu merepresentasikan sisi yang menghubungkan setiap simpul. Jadwal optimal bus dimulai pukul 06.30 pagi sampai pukul 17.00 sore. Setiap bus mendapatkan 4 (empat) sesi keberangkatan dan 4 (empat) sesi kepulangan dengan waktu tempuh masing-masing sesi 60 menit.

Abstract. This research deals with applying the Dijkstra algorithm and Welch-Powell algorithm to on-campus bus transportation problems. This research aims to determine the optimal solution of campus bus transportation routes in the shortest routes and schedules. In determining the shortest route, each intersection represented as a node and the path described as the sides. The shortest path obtained $V_1 - V_2 - V_5 - V_8 - V_9 - V_{10} - V_{13} - V_{16}$. In determining the optimal schedule, the number of buses represents the vertices, and the time expresses the side that connects each node. The optimal program of the bus starts from 06.30 am to 5.00 pm. Every bus gets four sessions of departure and four sessions return with travel time each session is 60 minutes.

How to cite:

Nurwan, W. E. Pranata, M. R. F. Payu, and N. I. Yahya, "Implementation of Dijkstra Algorithm and Welch-Powell Algorithm for Optimal Solution of Campus Bus Transportation", *J. Mat. Mantik*, vol. 7, no. 1, pp. 31-40, May 2021.

CONTACT:

Widya Eka Pranata,  widyapranata983@gmail.com  Department of Mathematics, Universitas Negeri Gorontalo, Kota Gorontalo, Gorontalo 96128, Indonesia



1. Introduction

The development of a new campus of Gorontalo State University (UNG) at Bone Bolango Regency is approximately 15 km from the center of Gorontalo City. Campus transfer impacts academic activities or activities as many as 4 (four) faculties, with about 8643 students. The UNG campus transfer from the center of Gorontalo city to Bone Bolango regency impacts transportation availability. The availability of transport is crucial for students to support activities in the new UNG campus. Transportation equipment that serves the route of Gorontalo City to the new campus Bone Bolango is a bus provided by the Gorontalo Porvinsi Transportation Agency and the Bone Bolango Regency Transportation Agency. Gorontalo Provincial Government prepares public transportation services in the form of Bus Rapid Transit (BRT). BRT is a mass transportation facility for the people, including UNG students, to the UNG Bone Bolango campus, even with a limited number. BRT does not prioritize the needs of UNG students. Therefore, it has an impact on the delay of students in eroding lectures. The issue of distance and the discovery of optimal routes is the most crucial thing in transportation problems when students go to Bone Bolango's new campus. This situation is to reduce student delays in participating in academic activities.

The above conditions required a solution to get the optimal bus route that serves students from the center of Gorontalo City to the New Campus of UNG Bone Bolango Regency and the optimal schedule of companies or transportation service providers. Transportation problems can be solved using graph theory to describe the pain to make it easier to solve. One of the ideas developed in graph theory is coloring. There are three kinds of color in graph theory: vertex coloring, face coloring, and region color [1].

Dijkstra's algorithm can also be called a greedy algorithm. It is one of the algorithms used to solve the shortest path. It does not have a negative cost [2]. The optimal route is completed using the Dijkstra algorithm to get the optimal bus schedule used welch-Powell algorithm. The working principle of algorithm Dijkstra searches for the two smallest trajectories, so this algorithm is advantageous in determining the shortest course from one point to another [3]. Dijkstra algorithms often search for the shortest routes, using nodes on a simple road network [3]. The Dijkstra algorithm's use to determine the shortest route of a graph will result in the best route, namely selecting and analyzing the unselected node's weight, then selecting the node with the most negligible weight [4].

The Dijkstra algorithm's application in determining the shortest route, among others, finds an effective route to avoid traffic jams during rush hour [5]. Dijkstra algorithm is used to calculate the closest distance from one point to the museum chosen to be the destination [6]. Implementation of Dijkstra algorithms on urban rail transit networks [7]. Determination of the Shortest Route with Using Dijkstra's Algorithm on the Path School bus [8]. One of the concepts of graphs to solve transportation scheduling problems is the concept of graph coloring. Graph coloring is the coloration represented by the sorted number [9] [10]. Use by coloring vertices based on the highest degree of all vertices [11]. The welch-Powell algorithm invented by Welch and Powell is very useful in scheduling. The application of graph coloring uses the Welch-Powell algorithm in determining student guidance schedules [12].

Another study about applying the Dijkstra algorithm was carried for selecting the route to reduce traffic congestion in Purwokerto. This study aims to solve congestion by determining alternative routes that are more effective and efficient. Application of Dijkstra's Algorithm utilizing determine the most negligible weight of each road segment. From this research, the rider can choose alternative routes to avoid congestion [13]. They are searching for the shortest route with Dijkstra's Algorithm. This research aims to simulate shortest path search using The Dijkstra algorithm to help find the shortest route [14]. Application of Dijkstra's Algorithm in the Bus Route Search Application Trans

Semarang. This researcher proposes a digital application solution to search for Trans Semarang Bus routes using the Dijkstra Algorithm [15].

Several studies related to Dijkstra's algorithm in transportation problems only focus on determining the shortest route without optimal scheduling. To optimize students' transportation routes to the UNG Bone Bolango campus and the opposite, the researchers solved two problems transportation: shortest route and the optimal schedule. Therefore, the researcher uses two different algorithms, namely the Dijkstra algorithm and the Welch Powell algorithm.

The background presented above is needed for optimal bus transportation that serves students from campus 1 in Gorontalo city to Bone Bolango campus. Researchers applied Dijkstra algorithms to determine the shortest routes and Welch-Powell algorithms to design schedules. This study will find the shortest route and planned bus departure schedule from campus 1 to Bone Bolango campus and scheduled return from Bone Bolango campus to Campus 1 Gorontalo City.

2. Methods

This study aims to find the optimal route solution of the bus by using the Dijkstra algorithm and set the bus optimal schedule solution by using the Welch-Powell algorithm with the following stages:

- Take a screenshot on google maps in the form of an image.
- Determines several routes from campus 1 to the Bone Bolango campus.
- Specify a starting point.
- Specify a destination point.
- Specify multiple intersections as nodes in the graph.
- Create a straight line from node to node as a side on a graph.
- We create a weighted graph by connecting the vertices using the contents and giving weight according to the distance.
- Analysis of data using Dijkstra and Welch-Powell algorithms.
- Make conclusions.

3. Results and Discussion

3.1 Bus Shortest Route to UNG Bone Bolango Campus

Researchers take screenshots on google maps based on previous research methods and represented them in the figure's form. The resulting image is then graphed, as shown in Figure 1.

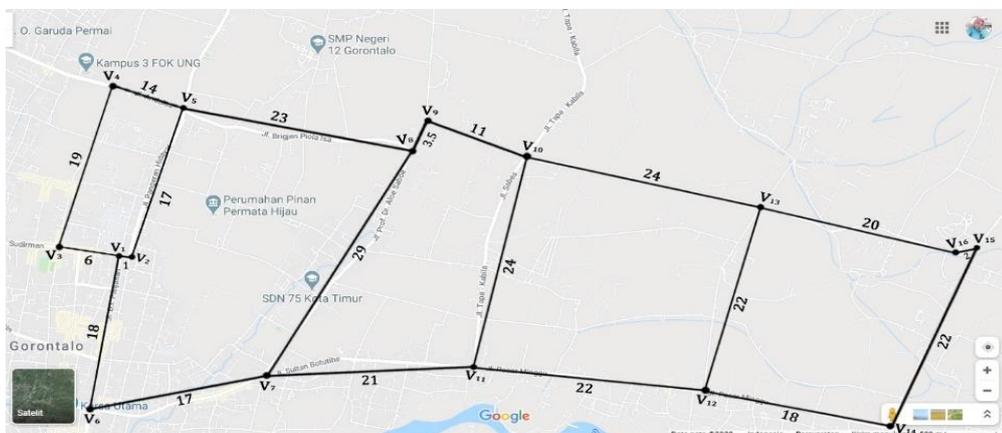


Figure 1. Weighted Graph

Figure 1 is a route that a bus can take from the starting point or departure point located at campus 1 UNG Gorontalo city to the endpoint located at the campus UNG Bone Bolango. The route in this study represents the side that connects each node. We can see the definition of nodes in Figure 1 in Table 1.

Table 1. Image Copyright Getty Images Image Caption The 1st graph

No	Node	Name (intersection)
1	V_1	Campus 1 UNG
2	V_2	The intersection of three Sentra Media (Jendral Sudirman street–Pangeran Hidayat street)
3	V_3	The intersection of four Junior high school 6 Gorontalo (Jendral Sudirman street– Jaksa Agung Suprpto street– Arif Rahman Hakim street)
4	V_4	The intersection of four Darul Muhtadin mosque (Arif Rahman Hakim street –Prof Jhon Aryo Katili street – Bj. Pola Isa street)
5	V_5	The intersection of four Public health center (Pangeran Hidayat street – Rusli Datau street – Prof. Aryo Katili street – Bj. Pola Isa street)
6	V_6	The intersection of four Baiturahim mosques (Nani Wartabone street– Raja Eyato street– Sultan Botutihe steer)
7	V_7	The intersection of four Moodu Market (Sultan Botutihe street– AloeI Saboe steer– Matolodula street)
8	V_8	The intersection of three UBM (Bj. Pola Isa street–AloeI Saboe street– Tinaloga street)
9	V_9	The intersection of three Tinaloga gas station (Tinaloga steer–Toto Tengah street)
10	V_{10}	The intersection of 4 Bypass Kabila (Toto Tengah street–B.J Habibie street– Sabes street– Noho Hudji street)
11	V_{11}	The intersection of three Police office of Kabila (Pasar Minggu street– Tapa Kabila street)
12	V_{12}	The intersection of three Al Munawarah mosque (Pasar Minggu street– Muh. Van Gobel street)
13	V_{13}	The intersection of four Darul Muhaimin mosque (B.J Habibie street– Muh. Van Gobel street – El Madinah Road street)
14	V_{14}	The intersection of three Indomaret (Pasar Minggu street–Jembatan Merah street)
15	V_{15}	The intersection of three Adipura Monument of Bone Bolango (Jembatan Merah street–B.J Habibie street)
16	V_{16}	UNG Bone Bolango campus

Table 1 data is used to determine the shortest trajectory using the Dijkstra algorithm with steps: 1). Node label with $\lambda(s) = 0$, and for each v node in G other than s , v node label with $\lambda(v) = \infty$. 2). Suppose $u \in T$ with a minimum $\lambda(u)$. 3). If $u = t$, stop, then the shortest trajectory from s to t is $\lambda(t)$. 4). For each side $e = uv$, $v \in T$; replace label v with $\lambda(v) = \text{minimum } \lambda(v), \lambda(u) + w(e)$. 5). $T = T - u$, and return to iteration 2 [16].

The shortest trajectory problem in all knot pairs is to find the shortest trajectory between knot pairs. $V_i, V_j \in v$ in such a way that $i \neq j$ [17]. Matrices of agility formed from graph weights with steps: 1). The distance of V_i node with V_j if there is a connecting side, then in writing with the weight, 2) 0 if the V_i node is connected to V_i And 3) if no side connects the V_i node with V_j After looking at the steps above then fill the matrix of agility with the weight on the graph. The results of graph representations weighted into the matrices of the neighboring show in Table 2.

Table 2. Representation of Graphs in The Matrices of Neighboring

V	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	V_{11}	V_{12}	V_{13}	V_{14}	V_{15}	V_{16}
V_1	0_{v1}	1_{v1}	6_{v1}	∞	∞	18_{v1}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
V_2	1_{v2}	0_{v2}	∞	∞	17_{v2}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
V_3	6_{v3}	∞	0_{v3}	19_{v3}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
V_4	∞	∞	19_{v4}	0_{v4}	14_{v4}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
V_5	∞	17_{v5}	14_{v5}	0_{v5}	∞	∞	∞	23_{v5}	∞							
V_6	18_{v6}	∞	∞	∞	∞	0_{v6}	17_{v6}	∞								
V_7	∞	∞	∞	∞	∞	17_{v7}	0_{v7}	29_{v7}	∞	∞	21_{v7}	∞	∞	∞	∞	∞
V_8	∞	∞	∞	∞	23_{v8}	∞	29_{v8}	0_{v8}	3.5_{v8}	∞						
V_9	∞	3.5_{v9}	0_{v9}	11_{v9}	∞	∞	∞	∞	∞	∞						
V_{10}	∞	11_{v10}	0_{v10}	24_{v10}	∞	24_{v10}	∞	∞	∞							
V_{11}	∞	∞	24_{v11}	0_{v11}	22_{v11}	∞	22_{v11}	∞	∞							
V_{12}	∞	∞	∞	22_{v12}	0_{v12}	22_{v12}	∞	∞	∞							
V_{13}	∞	∞	24_{v13}	∞	∞	0_{v13}	∞	∞	20_{v13}							
V_{14}	∞	∞	∞	22_{v14}	∞	∞	0_{v14}	22_{v14}	∞							
V_{15}	∞	∞	∞	∞	∞	∞	22_{v15}	0_{v15}	2_{v15}							
V_{16}	∞	∞	∞	∞	∞	20_{v16}	14	20_{v16}	0_{v16}							

Weight change of each iteration based on algorithm Dijkstra's steps obtained the shortest route from each node V_i to the V_j as in Table 3.

Table 3. Results of the shortest route in the matrices of neighbouring

V	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	V_{11}	V_{12}	V_{13}	V_{14}	V_{15}	V_{16}
V_1	0_{v1}	1_{v1}	6_{v1}	∞	∞	18_{v1}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
V_2		1_{v1}	6_{v1}	∞	18_{v2}	18_{v1}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
V_3			6_{v1}	25_{v3}	18_{v2}	18_{v1}	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
V_4				25_{v3}	18_{v2}	18_{v1}	∞	41_{v5}	∞	∞	∞	∞	∞	∞	∞	∞
V_5				25_{v3}		18_{v1}	35_{v4}	41_{v5}	∞	∞	∞	∞	∞	∞	∞	∞
V_6				25_{v3}			35_{v6}	41_{v5}	∞	∞	∞	∞	∞	∞	∞	∞
V_7							35_{v6}	41_{v5}	∞	∞	56_{v7}	∞	∞	∞	∞	∞
V_8								41_{v5}	44.5_{v8}	∞	56_{v7}	∞	∞	∞	∞	∞
V_9									44.5_{v8}	55.5_{v9}	56_{v7}	∞	∞	∞	∞	∞
V_{10}										55.5_{v9}	56_{v7}	∞	79.5_{v10}	∞	∞	∞
V_{11}											56_{v7}	78_{v11}	79.5_{v10}	∞	∞	∞
V_{12}												78_{v11}	79.5_{v10}	96_{v12}	∞	∞
V_{13}													79.5_{v10}	96_{v12}	∞	∞
V_{14}														96_{v12}	118_{v14}	99.5_{v1}
V_{15}															101.5_{v1}	99.5_{v1}
V_{16}																101.5_{v1}

Representation of the neighbouring matrix in Table 3 results in the shortest route of the bus with the following details:

- a. The V_1 is connected to V_1, V_3, V_6 with a weight of $V_1 - V_2 = 1, V_1 - V_3 = 6, V_1 - V_6 = 18$. Because V_2 the smallest weight choose and colour, it then lowered the V_2 .
- b. V_2 connected with V_1 and V_5 with a $V_2 - V_5 = 17$. Added with colored weight ($17 + 1 = 18$). Because V_3 the smallest weight, then the V_3 selected and coloured, then lowered the V_3 .

- c. V_3 connected with V_1 and V_4 nodes with a weight of $V_3 - V_4 = 19$ Added with a coloured weight ($19+6 = 25$). Furthermore, two nodes have the same weight (smallest), namely V_5 and V_6 , selected V_5 , then lowered the V_5 .
- d. V_5 connected with V_2 , V_4 , V_8 with a weight of $V_5 - V_4 = 14$, Added with a coloured weight ($14 + 18 = 32$). Because the weight is greater than the previous weight, the previous weight is lowered. Next $V_5 - V_8 = 23$ then add with the coloured weight ($23+18 = 41$). Because V_6 the smallest weight, then select and colour, then lowered the V_6 .
- e. V_6 connected with V_1 and V_7 nodes with a $V_6 - V_7 = 17$, then add with the coloured weight ($17 + 18 = 35$). Because V_4 smallest weight choose and colour then lowered the V_4 .
- f. V_4 connected with the V_3 and V_5 because the V_3 and V_5 already coloured, then further lose the smallest weight. Since V_7 smallest weight choose and colour, it is also lowered V_7 .
- g. V_7 connected with V_8 and V_{11} with a weight of $V_7 - V_8 = 29$, then add with the coloured weight ($29 + 35 = 64$) because the result is 64, greater than the previous weight, then lower the previous weight. Next $V_7 - V_{11} = 21$ then add with the colored weight ($21 + 35 = 56$). Since the smallest weighted V_8 select and colour, the next lowered V_8 .
- h. V_8 is connected to V_7 and V_9 with a weight of $V_8 - V_9 = 3.5$, then add with the coloured weight ($3.5 + 41 = 44.5$). Because V_9 smallest weight choose and colour then lowered the V_9 .
- i. V_9 is connected to V_8 , and V_{10} with a weight of $V_9 - V_{10} = 11$, then add with the coloured weight ($11 + 44.5 = 55.5$). Since V_{10} smallest weight choose and colour, the knot is further lowered V_{10} .
- j. V_{10} connected with V_9 , V_{11} , V_{13} with a weight of $V_{10} - V_{11} = 24$ then add with the weight already coloured ($24 + 55.5 = 79.5$) because the result is 79.5, greater than the previous weight, then lower the previous weight. Next $V_{10} - V_{13} = 24$ then add with the colored weight ($24 + 55.5 = 79.5$). Because V_{11} smallest weight choose and colour then lowered the V_{11} .
- k. V_{11} connected with V_7 , V_{10} , V_{12} with a weight of $V_{11} - V_{12} = 22$, then added with a coloured weight ($22 + 56 = 78$). Because V_{12} smallest weight choose and colour then lowered the V_{12} .
- l. V_{12} is connected with V_{11} , V_{13} , V_{14} with a weight of $V_{12} - V_{13} = 22$, then add with the coloured weight ($22 + 78 = 100$) because the result is 100, greater than the previous weight, then lower the previous weight. Next $V_{12} - V_{14} = 18$ then add the colored weight ($18+78 = 96$). Because V_{13} smallest weight choose and colour then lowered the V_{13} .
- m. V_{13} is connected with V_{10} , V_{12} , V_{16} with a weight of $V_{13} - V_{16} = 20$ then add with the colored weight ($20 + 79.5 = 99.5$). Because V_{14} smallest weight choose and colour then lowered the V_{14} .
- n. V_{14} is connected V_{12} , V_{15} with a weight of $V_{14} - V_{15} = 22$ then 27 add with colored weight ($22 + 96 = 118$). Because of the smallest weighted V_{16} select and colour, the next lowered V_{16} .
- o. V_{16} is connected to the V_{13} , and V_{15} with a weight of $V_{16} - V_{15} = 2$, then add with the coloured weight ($2+99.5 = 101.5$) because the result is 101.5, smaller than the previous weight, then choose and colour. Next, I lowered the V_{15} .
- p. V_{15} connected to the V_{14} and V_{16} because the V_{14} and V_{16} has been coloured, then it is finished.

Figure 2 The shortest route a bus can take from campus 1 UNG to Bone Bolango's new campus is From V_1 to V_{16} , or vice versa obtained the shortest route begins $V_1 - V_2 - V_5 - V_8 - V_9 - V_{10} - V_{13} - V_{16}$ or Campus 1 UNG - The intersection of three Sentra

Media (Jendral Sudirman street–Pangeran Hidayat street) - The intersection of four Public health center (Pangeran Hidayat street – Rusli Datau street – Prof. Aryo Katili street – Bj. Pola Isa street) - The intersection of three UBM campus (Bj. Pola Isa street–Aloei Saboe street-Tinaloga street) - The intersection of three Tinaloga gas station (Tinaloga steer–Toto Tengah street) - The intersection of four Bypass Kabila (Toto Tengah street–B.J Habibie street– Sabes street– Noho Hudji street) - The intersection of four Darul Muhaimin mosque (B.J Habibie street– Muh. Van Gobel street – El Madinah Road street) – UNG Bone Bolango Campus.



Figure 2. Bus Shortest Route to UNG Bone Bolango Campus

3.2 Bus Schedule to UNG Bone Bolango Campus

The Welch-Powell algorithm is required to color the graph's vertices based on the highest degree of all its nodes. From the data available, there are four buses in operation [18]. These four buses run back and there, represented in a graph. The data of the number of buses be defined as a node on the graph. There are no specific labelling rules, labelled with an index to distinguish which buses operate back and away. Suppose each node with $V_{a,b}$ with the following description:

1. a: A bus, $a \in [1, 4]$
2. b: Return or Departure (1 for departure, 2 for return)

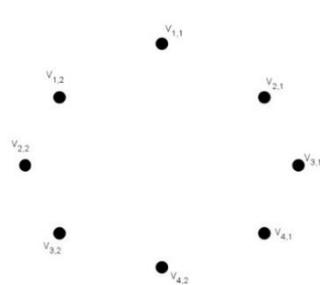


Figure 3. Representation of Bus Data in Graph

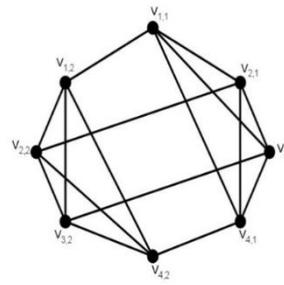


Figure 4. Graph Representation of Bus Schedules

Each node in Figure 3 is then connected and forms aside. The side represents the bus's operational time, and each bus does not operate simultaneously. The arrangement of the vertices' location establishes a circle to make it easier to draw a straight line for the relationship of each node. The node in Figure 4 represents the bus, and the side is a representation of time. Node $V_{1.1}, V_{2.1}, V_{3.1}, V_{4.1}$ is the bus departure node from campus 1 to campus UNG Bone Bolango and node $V_{1.2}, V_{2.2}, V_{3.2}, V_{4.2}$ is the node from campus UNG Bone Bolango to campus 1. what can see see degrees on each node in Table 4.

Table 4. Vertices by degree

No	Node	Neighbors	Degrees
1	$V_{1.1}$	$V_{1.2}, V_{2.1}, V_{3.1}, V_{4.1}$	4
2	$V_{1.2}$	$V_{1.1}, V_{2.2}, V_{3.2}, V_{4.2}$	4
3	$V_{2.1}$	$V_{1.1}, V_{2.2}, V_{3.1}, V_{4.1}$	4
4	$V_{2.2}$	$V_{1.2}, V_{2.1}, V_{3.2}, V_{4.2}$	4
5	$V_{3.1}$	$V_{3.2}, V_{1.1}, V_{2.1}, V_{4.1}$	4
6	$V_{3.2}$	$V_{3.1}, V_{1.2}, V_{2.2}, V_{4.2}$	4
7	$V_{4.1}$	$V_{4.2}, V_{1.1}, V_{2.1}, V_{3.1}$	4
8	$V_{4.2}$	$V_{4.1}, V_{1.2}, V_{2.2}, V_{3.2}$	4

Table 4 shows the same degree on each node, then coloured by not giving the same colour to the neighbouring nodes as in Figure 5. By doing the steps of the welch-Powell algorithm, knot colouring is obtained as in Figure 6.

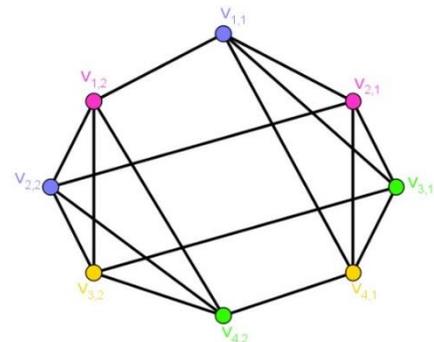
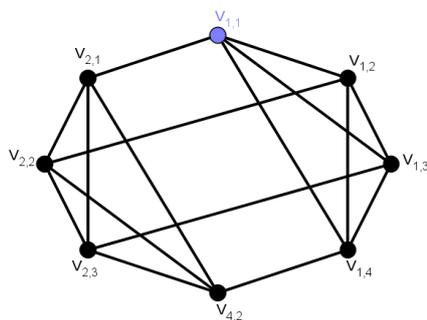


Figure 5. The Colouration of Vertices In Graph

Figure 6. Final Result of Knot Coloring

In Figure 6, four kinds of coloring are obtained. Next, group the buses by chromatic numbers. A graph has a chromatic number denoted by $\chi(G)$ [19] [20]. Zero graphs have a chromatic number of $\chi(G) = 1$, while to color, a complete graph is required n color fruits because all points are interconnected [19] [21]. The chromatic number of the bus schedule representation graph is $\chi(G) = 4$ which means each bus schedule for return or departure is at least four sessions. We can see the results of graph coloring for the campus bus schedule in Table 5.

Table 5. Bus Departure and Return Schedule to/from UNG Bone Bolango Campus

No	Departure Schedule		Schedule of Return	
	Time	Bus	Time	Bus
1	06.30	BUS 1	07.30	BUS 1
2	07.30	BUS 2	08.30	BUS 2
3	08.00	BUS 3	09.00	BUS 3

No	Departure Schedule		Schedule of Return	
	Time	Bus	Time	Bus
4	08.30	BUS 4	09.30	BUS 4
5	09.00	BUS 1	10.00	BUS 1
6	09.30	BUS 2	11.00	BUS 2
7	10.00	BUS 3	12.30	BUS 3
8	11.00	BUS 4	13.00	BUS 4
9	12.30	BUS 1	13.30	BUS 1
10	13.00	BUS 2	14.00	BUS 2
11	13.30	BUS 3	14.30	BUS 3
12	14.00	BUS 4	15.00	BUS 4
13	14.30	BUS 1	15.30	BUS 1
14	15.00	BUS 2	16.00	BUS 2
15	15.30	BUS 3	16.30	BUS 3
16	16.00	BUS 4	17.00	BUS 4

Table 5 the bus departure schedule from campus 1 to UNG Bone Bolango campus consists of 16 departure time sessions with departure time starting from 06:30 and ending at 16:00 with a delay of 60 minutes each departure time. The bus return schedule from UNG Bone Bolango campus to campus 1 consists of 16 sessions by the same bus. The bus departure schedule from UNG Bone Bolango campus to campus starts at 07.30 am and ends at 05.00 pm. The overall departure schedule consists of 32 departure sessions. Each bus gets eight departure sessions per day with a minimum session break of 60 minutes, including travel time.

4. Conclusions

The Dijkstra algorithm's calculation obtained the shortest bus route from campus 1 to campus UNG Bone Bolango is the trajectory $V_1-V_2-V_5-V_8-V_9-V_{10}-V_{13}-V_{16}$ with a distance of 9.95 km. The bus passed by the bus is Campus 1 UNG → The intersection of 3 Sentra Media → The intersection of 4 Public health center → The intersection of 3 UBM campus → The intersection of 3 Tinaloga gas station → The intersection of 4 Bypass Kabila → Intersection of 4 Darul Muhaimin mosque → UNG Bone Bolango Campus. The coloring results using the Welch-Powell algorithm obtained the number of colors on the bus schedule's color to UNG Bone Bolango in four colors. The bus departure and return schedule are 16 sessions each, and the bus operating time starts from 06.30 am to 05.00 pm. Each bus gets four departure sessions and four return sessions per day with a travel time of 60 minutes.

References

- [1] A. Y. Harsya and I. H. Agustin, "Pewarnaan Titik Pada Operasi Graf Sikel dengan Graf Lintasan Pendahuluan Teorema yang Digunakan," in *Journal University Of Jember*, 2014, vol. 1, no. 1.
- [2] M. S. Yusuf, H. M. Az-zahra, and D. H. Apriyanti, "Implementasi Algoritma Dijkstra Dalam Menemukan Jarak Terdekat Dari Lokasi Pengguna Ke Tanaman Yang Di Tuju Berbasis Android (Studi Kasus di Kebun Raya Purwodadi)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 12, pp. 1779–1781, 2017.
- [3] Siswanto, *Algoritma dan Struktur Data Non Linear dengan Java*. Yogyakarta: Graha Ilmu, 2011.
- [4] A. Pitri, "Penerapan Metode Djikstra Pencarian Rute Terpendek Sekolah Luar

- Biasa (SLB) di Kota Medan,” *J. Ris. Komput.*, vol. 5, no. 6, pp. 638–643, 2018.
- [5] P. Sembiring, A. S. Harahap, and K. S. Zalukhu, “Implementation of Dijkstra’s algorithm to find an effective route to avoid traffic jam on a busy hour,” *J. Phys. Conf. Ser.*, vol. 1116, no. 2, 2018.
- [6] S. Maimunah, Ni’matuzahroh, S. Prasetyaningrum, and B. I. Suwandayani, “Implementasi Model Pendidikan Inklusi di Sekolah Dasar Kota Batu,” *J. Pendidik. Surya Edukasi*, vol. 4, no. 2, 2018.
- [7] T. Jimeng, S. Quanxin, and C. Zhijie, “A New Implementation of Dijkstra ’ s Algorithm on Urban Rail Transit Network,” 2016, no. Iccte, pp. 507–513.
- [8] I. P. W. Gautama and K. Hermanto, “Penentuan Rute Terpendek dengan Menggunakan Algoritma Dijkstra pada Jalur Bus Sekolah,” *J. Mat.*, vol. 10, no. 2, p. 116, 2020.
- [9] M. Kubale, *Graph Colorings*. America: AMS Bookstore.
- [10] R. Diestel, *Graph Theory*, Electronic. New York: Springer-Verleg, 2000.
- [11] R. Munir, *Matematika diskrit*. Bandung: Informatika, 2012.
- [12] A. W. Bustan and M. R. Salim, “Implementation of Graph Colouring Using Welch-Powell Algorithm to Determine Student Mentoring Schedule,” *THEOREMS (The Orig. Res. Math.)*, vol. 4, no. 1, pp. 79–86, 2019.
- [13] U. M. Rifanti, “Pemilihan Rute Terbaik Menggunakan Algoritma Dijkstra Untuk Mengurangi Kemacetan Lalu Lintas di Purwokerto,” *JMPM J. Mat. dan Pendidik. Mat.*, vol. 2, no. 2, p. 90, 2017.
- [14] M. K. Harahap and N. Khairina, “Pencarian Jalur Terpendek dengan Algoritma Dijkstra,” *Sinkron*, vol. 2, no. 2, p. 18, 2017.
- [15] D. Ardana and R. Saputra, “Penerapan Algoritma Dijkstra pada Aplikasi Pencarian Rute Bus Trans Semarang,” in *Seminar Nasional Ilmu Komputer (SNIK 2016)*, 2016, no. 978-602-1034-40–8, pp. 299–306.
- [16] I. Budayasa, *Teori Graph dan Aplikasinya*. Surabaya: Unesa University Press, 2007.
- [17] E. A. Sarwoko, “Perancangan Arsitektur Pamaralelan Untuk Mencari Shortest Path Dengan Algoritma Dijkstra,” *Matematika*, vol. 6, no. 3, pp. 137–143, 2003.
- [18] Y. Rusdiana and A. Maulani, “Algoritma Welch-Powell Untuk Pewarnaan Graf pada Penjadwalan Perkuliahan,” *Sci. Phys. Educ. J.*, vol. 3, no. 1, pp. 37–47, 2019.
- [19] G. Chartrand, L. Lesniak, and P. Zhang, *Graphs and Digraphs*. California: Pacific Grove California, 2016.
- [20] R. Balakrishnan and K. Ranganathan, *A Textbook of Graph Theory*, Second Edi. Springer Science+Business Media New York, 2021.
- [21] A. M. Soimah and N. S. M. Mussafi, “Pewarnaan Simpul Dengan Algoritma Welch-Powell Pada Traffic Light Di Yogyakarta,” *J. Fourier*, vol. 2, no. 2, p. 73, 2013.