

## Algoritma Dijkstra dan Algoritma Greedy Untuk Optimasi Rute Pengiriman Barang Pada Kantor Pos Gorontalo

Novria Fatmawati Lakutu<sup>1</sup>, Muhammad Rifai Katili<sup>2</sup>, Sri Lestari Mahmud<sup>3\*</sup>, Nisky Imansyah Yahya<sup>4</sup>

<sup>1,3,4</sup>Jurusan Matematika, Universitas Negeri Gorontalo, Bone Bolango 96554, Indonesia

<sup>2</sup>Jurusan Teknik Informatika, Universitas Negeri Gorontalo, Bone Bolango 96554, Indonesia

\*Penulis Korespondensi. Email: [sri.lestari@ung.ac.id](mailto:sri.lestari@ung.ac.id)

---

### Abstrak

Pengiriman barang merupakan salah satu jasa yang digunakan dalam proses jual beli online pada perusahaan Kantor Pos. Perusahaan perlu adanya suatu kinerja dalam pendistribusian barang lebih efisien waktu dan biaya sehingga memerlukan ketetapan dalam pemilihan jalur terpendek. Telah banyak algoritma yang digunakan untuk membantu menemukan jalur terpendek pendistribusian barang di antaranya yaitu Algoritma Dijkstra dan Algoritma Greedy. Pada penelitian ini penerapan kedua algoritma tersebut untuk mencari rute antar kelurahan yang menjadi tujuan pengiriman barang, kemudian dibandingkan untuk melihat algoritma manakah yang lebih optimal. Berdasarkan hasil perhitungan, Algoritma Dijkstra dapat memberikan solusi yang lebih optimal dengan total jarak tempuh yang diperoleh 304,90 Km sedangkan Algoritma Greedy menghasilkan total jarak tempuh 441,60 Km.

**Kata Kunci:** Rute Terpendek; Algoritma Dijkstra; Algoritma Greedy

### Abstract

*Delivery of goods is one of the services used in the online buying and selling process at the Post Office company. Companies need a performance in the distribution of goods that is more time-consuming and cost-efficient, so it requires determination in selecting the shortest path. Many algorithms have been used to help find the shortest path for distributing goods, including the Dijkstra Algorithm and the Greedy Algorithm. In this study, the application of the two algorithms is to find routes between sub-districts that are the destination of goods delivery, then compare to see which algorithm is more optimal. Based on the calculation results, the Dijkstra Algorithm can provide a more optimal solution with a total mileage obtained of 304.90 Km, while the Greedy Algorithm produces a total distance of 441.60 Km.*

**Keywords:** Shortest Route; Dijkstra Algorithm; Greedy Algorithm

---

## 1. Pendahuluan

Matematika merupakan ilmu yang dapat memudahkan penyajian dan pemahaman pada suatu masalah [1]. Salah satu bidang matematika yang dapat menggambarkan representasi masalah adalah Teori Graf. Graf adalah grafik yang memuat informasi tertentu yang dengan cepat di interpretasikan untuk menjelaskan berbagai struktur yang ada, agar objek lebih mudah dipahami [2]. Graf juga dapat menyelesaikan berbagai masalah, salah satunya adalah masalah optimasi. Masalah optimasi yang dapat dipecahkan yaitu pencarian jalur terpendek. Jalur terpendek adalah pencarian lintasan terpendek antar simpul graf dan menghasilkan bobot terkecil [3]. Pencarian jalur terpendek ini telah digunakan di berbagai bidang untuk meminimalkan biaya atau mempersingkat proses [4]. Dalam teori graf, ada beberapa algoritma yang dapat digunakan untuk menyelesaikan masalah rute terpendek, antara lain algoritma Dijkstra, algoritma Bellmand Ford, algoritma Greedy, dan algoritma Floyd Warshall.

Algoritma Dijkstra merupakan salah satu algoritma yang paling mudah untuk di implementasikan karena algoritma ini merupakan bagian dari program dinamis yang menemukan

semua jalur terpendek antara setiap kemungkinan pasangan lokasi yang berbeda dan sangat efektif dalam menyelesaikan masalah jalur optimal untuk semua pasangan simpul. Algoritma ini memecahkan masalah pencarian rute terpendek (rute dengan panjang minimum) pada graf berbobot, bobotnya adalah bilangan positif sehingga simpul negatif tidak dapat melewatinya, tetapi jika ya, maka solusi yang diberikan adalah *infinity* [5]. Sedangkan Algoritma Greedy adalah algoritma yang memperoleh solusi optimal dari masalah optimasi dengan cara menyelesaikannya langkah demi langkah [6]. Algoritma Greedy merupakan algoritma yang dapat memecahkan masalah dengan menemukan alternatif paling baik pada saat tertentu. Walaupun setiap solusi jawaban dari algoritma greedy tidak dipastikan merupakan solusi optimum global, tetapi dapat diandalkan untuk mengoptimasi waktu [7].

Permasalahan yang dapat diterapkan pada pencarian jalur terpendek yaitu pada pengiriman barang. Pengiriman barang merupakan salah satu jasa yang telah banyak digunakan masa kini. Hal tersebut dikarenakan jasa pengiriman barang berperan penting dalam segala kebutuhan, salah satunya dalam proses jual beli online. Di Indonesia, terdapat beberapa perusahaan yang bergerak pada bidang pengiriman, salah satunya yaitu PT. Pos Indonesia atau Kantor Pos, yang merupakan perusahaan besar bergerak di bidang pengiriman barang. Perusahaan ini pertama kali didirikan di Batavia (Jakarta) oleh gubernur Jendral G.W Baron Van Imhoff pada tanggal 26 Agustus 1746 [8]. Proses pengiriman barang yang dilakukan memiliki beberapa tahapan yang diawali dengan proses pencocokan untuk mengetahui barang dalam keadaan utuh dan tidak rusak kemudian dilakukan proses penyerahan barang ke kurir dan selanjutnya melakukan pengiriman barang ke alamat-alamat yang dituju. Adapun permasalahan yang terjadi diantaranya keterlambatan pengiriman barang ke alamat tujuan karena banyaknya barang yang harus dikirim ke beberapa tempat yang berbeda. Hal ini karena kantor pos masih menggunakan panduan manual (perkiraan) untuk menentukan rute pengiriman. Untuk mengatasi masalah tersebut cara yang dapat di ambil oleh perusahaan menentukan rute distribusi yang optimal dengan mempertimbangkan jarak tempuh. Oleh karena itu, Kantor pos memerlukan sebuah kebijakan pemilihan rute pengiriman barang. Hasil dari menentukan rute terpendek dapat menjadi bahan peninjauan dalam pengambilan keputusan untuk menunjukan lintasan yang akan ditempuh.

Beberapa penelitian terkait masalah optimasi telah banyak dilakukan diantaranya adalah Usman, *et al.* [9] meneliti tentang optimasi rute distribusi dalam menyalurkan produk obat dan barang dengan menerapkan Algoritma Greedy tujuannya untuk meminimalkan biaya pengiriman dan meminimalkan total waktu perjalanan distribusi. Hasil perhitungan menunjukkan rute dalam melakukan distribusi pada PT. Z lebih optimal dibandingkan dengan rute sebelum menggunakan Algoritma Greedy. Cantona, *et al.* [10] membahas tentang optimasi dalam mencari rute terpendek yang diterapkan pada smartphone, pengguna dapat menghemat waktu mencari museum di Jakarta. Hasil penelitian yang diperoleh bahwa Algoritma Dijkstra mendapatkan rute terpendek dengan memerlukan 7 bobot yang mana hanya perlu 35% untuk mencapai museum di Jakarta.

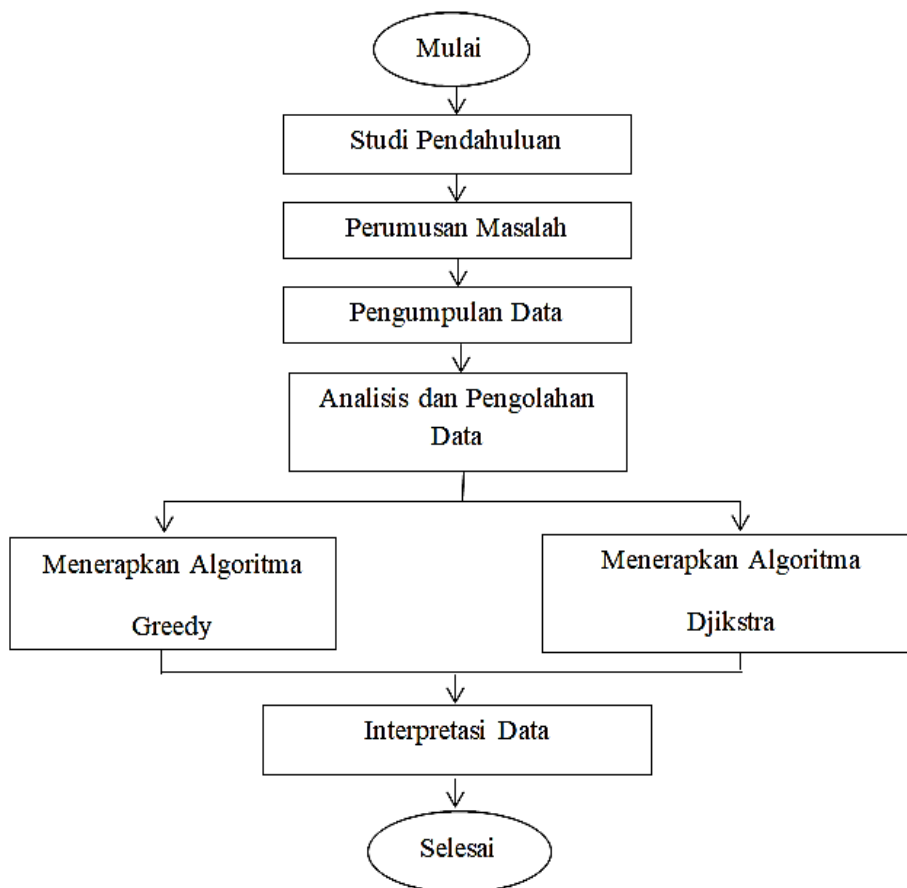
Pada penelitian sebelumnya yang dilakukan [11] meneliti tentang optimasi jalur bus kampus menuju Kampus UNG Bone Bolango dengan menerapkan Algoritma Dijkstra. Hasil di dapat bahwa algoritma Dijkstra menghasilkan solusi yang optimal dengan lintasan terpendek dari V1 ke V16 atau dengan keterangan (Kampus 1 UNG ke Kampus 4 UNG Bone Bolango) dengan bobot jarak tempuh 99,5 Km. Di samping penelitian yang dilakukan Arviantino, *et al.* [12] penelitian tersebut membahas tentang penerapan Algoritma Greedy dalam mencari titik nilai terkecil dengan menghitung jalur yang dilewati. Hasil dari penelitian tersebut menunjukkan bahwa rute yang dikunjungi untuk safari dakwah pada semua masjid dimulai dari titik *ADCEFGB* dengan durasi waktu yang optimal. Berbeda dari penelitian-penelitian sebelumnya, pada penelitian ini dilakukan optimasi pencarian rute terpendek pada Kantor Pos Gorontalo dengan menerapkan Algoritma Dijkstra dan Algoritma Greedy untuk mengoptimalkan rute pengiriman barang dengan menghitung jarak terpendek di masing-masing alamat yang dituju.

## 2. Metode Penelitian

Penelitian ini merupakan penelitian kuantitatif dengan studi kasus distribusi barang di Kantor Pos Gorontalo. Data yang digunakan dalam penelitian ini adalah data primer yang diperoleh hasil melakukan wawancara dan observasi kepada pihak penanggung jawab di Kantor Pos Gorontalo, dengan langkah-langkah sebagai berikut:

- 1) Studi pendahuluan dimana pada Penelitian ini menelaah jurnal, artikel dan referensi yang akan digunakan dalam penelitian.
- 2) Merumuskan masalah dengan melakukan wawancara dan observasi.
- 3) Mengumpulkan data yang dibutuhkan berupa data lokasi.
- 4) Mengolah data dengan menerapkan Algoritma Dijkstra dan Algoritma Greedy
- 5) Menginterpretasikan data dengan menggunakan kedua Algoritma untuk mendapatkan solusi rute terpendek kemudian dibuat kesimpulan dari hasil penelitian yang dilakukan.

Tahapan-tahapan yang di gunakan dalam menerapkan Algoritma Dijkstra dan Algoritma Greedy dapat dilihat pada Gambar 1.



**Gambar 1.** Diagram Alir Penelitian

## 3. Hasil dan Pembahasan

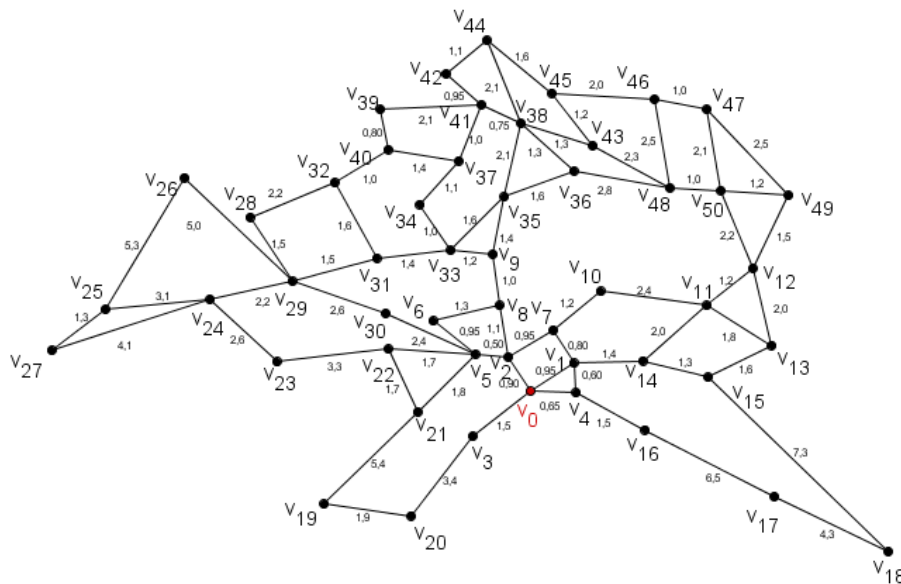
PT. Pos Indonesia (Persero) merupakan salah satu Badan Usaha Milik Negara (BUMN) yang bergerak dalam bidang jasa pengiriman surat maupun barang. Cabang PT. Pos Indonesia salah satunya terletak di Kota Gorontalo. Berdasarkan hasil observasi yang dilakukan, penyaluran barang di Kantor Pos ini terbagi menjadi beberapa wilayah. Pada penelitian ini di batasi ke 9 Kecamatan yang ada di Kota Gorontalo untuk digunakan sebagai penentuan rute terpendek dari setiap kelurahan yang ada di suatu Kecamatan dengan menerapkan dua algoritma yaitu Algoritma Dijkstra dan Algoritma Greedy. Dalam proses penyaluran barang pada Kantor Pos Gorontalo sebagai simpul awal

meju ke lokasi penyaluran atau sebagai simpul tujuan dengan total 50 simpul yang disajikan pada Tabel 1.

**Tabel 1.** Keterangan Simpul pada Graf

Simpul	Keterangan	Simpul	Keterangan
$V_0$	Kantor Pos Gorontalo	$V_{26}$	Kelurahan Dembe 1
$V_1$	Kelurahan Ipilo	$V_{27}$	Kelurahan Pilolodaa
$V_2$	Kelurahan Biawao	$V_{28}$	Kelurahan Tuladenggi
$V_3$	Kelurahan Tenda	$V_{29}$	Kelurahan Buladu
$V_4$	Kelurahan Bugis	$V_{30}$	Kelurahan Molosipat W
$V_5$	Kelurahan Biawu	$V_{31}$	Kelurahan Libuo
$V_6$	Kelurahan Limba B	$V_{32}$	Kelurahan Huangobotu
$V_7$	Kelurahan Heledulaa Selatan	$V_{33}$	Kelurahan Wumialo
$V_8$	Kelurahan Limba U1	$V_{34}$	Kelurahan Dulalowo
$V_9$	Kelurahan Limba U2	$V_{35}$	Kelurahan Dulalowo Timur
$V_{10}$	Kelurahan Heledulaa	$V_{36}$	Kelurahan Liluwo
$V_{11}$	Kelurahan Moodu	$V_{37}$	Kelurahan Pulubala
$V_{12}$	Kelurahan Dembe 2	$V_{38}$	Kelurahan Paguyaman
$V_{13}$	Kelurahan Tamalate	$V_{39}$	Kelurahan Tomulabutao
$V_{14}$	Kelurahan Padebuolo	$V_{40}$	Kelurahan Tomulabutao Selatan
$V_{15}$	Kelurahan Botu	$V_{41}$	Kelurahan Tapa
$V_{16}$	Kelurahan Talumolo	$V_{42}$	Kelurahan Molosipat U
$V_{17}$	Kelurahan Leato Utara	$V_{43}$	Kelurahan Tanggikiki
$V_{18}$	Kelurahan Leato Selatan	$V_{44}$	Kelurahan Bulotadaa
$V_{19}$	Kelurahan Tanjung Kramat	$V_{45}$	Kelurahan Bulotadaa Timur
$V_{20}$	Kelurahan Pohe	$V_{46}$	Kelurahan Dulomo
$V_{21}$	Kelurahan Siendeng	$V_{47}$	Kelurahan Dulomo Selatan
$V_{22}$	Kelurahan Donggala	$V_{48}$	Kelurahan Wongkaditi Barat
$V_{23}$	Kelurahan Tenilo	$V_{49}$	Kelurahan Wongkaditi Timur
$V_{24}$	Kelurahan Buliide	$V_{50}$	Kelurahan Dembe Jaya
$V_{25}$	Kelurahan Lekobalo		

Tabel 1. merupakan keterangan dari letak 50 simpul pada graf berbobot yang di interpretasikan dalam bentuk Graf yang disajikan pada Gambar 1.



**Gambar 1.** Graf Berbobot Titik Penyaluran

### 3.1 Menentukan Rute Terpendek Menggunakan Algoritma Dijkstra

Algoritma Dijkstra adalah sebuah variasi dari algoritma populer untuk memecahkan masalah optimisasi dan bersifat sederhana [13]. Langkah Algoritma Dijkstra menggunakan pendekatan dalam dalam penentuan lintasan terpendek dari suatu titik tertentu ke setiap titik lain pada suatu graf dengan pengulangan sebanyak  $|V(G)| - 1$ , sehingga membentuk pohon pembangun Dijkstra setelah pengulangan terakhir dari Graf [14]. Berikut perubahan bobot dari setiap iterasi yang disajikan pada Tabel 2.

**Tabel 2.** Daftar Bobot Antar Jarak

No	Sisi	Bobot	No	Sisi	Bobot	No	Sisi	Bobot
1	$V_0 - V_1$	0,95	27	$V_{13} - V_{12}$	2,0	53	$V_{35} - V_{36}$	1,6
2	$V_0 - V_2$	0,90	28	$V_{14} - V_{15}$	2,0	54	$V_{35} - V_{33}$	1,6
3	$V_0 - V_3$	1,5	29	$V_{14} - V_{11}$	1,6	55	$V_{36} - V_{38}$	1,3
4	$V_0 - V_4$	0,65	30	$V_{15} - V_{13}$	7,3	56	$V_{37} - V_{41}$	1,0
5	$V_1 - V_4$	0,60	31	$V_{15} - V_{18}$	6,5	57	$V_{37} - V_{40}$	1,4
6	$V_1 - V_7$	0,80	32	$V_{16} - V_{17}$	4,3	58	$V_{38} - V_{41}$	0,75
7	$V_1 - V_{14}$	1,4	33	$V_{17} - V_{18}$	1,9	59	$V_{38} - V_{35}$	2,1
8	$V_2 - V_7$	0,95	34	$V_{20} - V_{19}$	5,4	60	$V_{38} - V_{44}$	2,1
9	$V_2 - V_8$	1,1	35	$V_{21} - V_{19}$	1,7	61	$V_{39} - V_{40}$	0,80
10	$V_2 - V_5$	0,50	36	$V_{21} - V_{22}$	3,3	62	$V_{40} - V_{32}$	1,0
11	$V_3 - V_{20}$	3,4	37	$V_{22} - V_{23}$	2,6	63	$V_{41} - V_{39}$	2,1
12	$V_4 - V_{16}$	1,5	38	$V_{23} - V_{24}$	4,1	64	$V_{41} - V_{42}$	0,95
13	$V_5 - V_{21}$	1,8	39	$V_{24} - V_{27}$	3,1	65	$V_{43} - V_{45}$	1,2
14	$V_5 - V_{30}$	2,4	40	$V_{24} - V_{25}$	1,3	66	$V_{43} - V_{38}$	1,3
15	$V_5 - V_{22}$	1,7	41	$V_{25} - V_{27}$	5,3	67	$V_{44} - V_{42}$	1,1
16	$V_6 - V_5$	0,95	42	$V_{25} - V_{26}$	1,5	68	$V_{45} - V_{44}$	1,6
17	$V_7 - V_{10}$	1,2	43	$V_{28} - V_{29}$	5,0	69	$V_{46} - V_{45}$	2,0
18	$V_8 - V_6$	1,3	44	$V_{29} - V_{26}$	2,2	70	$V_{47} - V_{46}$	1,0
19	$V_8 - V_9$	1,0	45	$V_{29} - V_{24}$	1,5	71	$V_{48} - V_{36}$	2,8
20	$V_9 - V_{35}$	1,4	46	$V_{29} - V_{31}$	2,6	72	$V_{48} - V_{46}$	2,5
21	$V_9 - V_{33}$	1,2	47	$V_{30} - V_{29}$	1,6	73	$V_{48} - V_{43}$	2,3
22	$V_{10} - V_{11}$	2,4	48	$V_{31} - V_{32}$	2,2	74	$V_{49} - V_{47}$	2,5
23	$V_{11} - V_{12}$	1,2	49	$V_{32} - V_{28}$	1,4	75	$V_{49} - V_{50}$	1,2
24	$V_{12} - V_{50}$	2,2	50	$V_{33} - V_{31}$	1,0	76	$V_{50} - V_{48}$	1,0
25	$V_{12} - V_{49}$	1,5	51	$V_{33} - V_{34}$	1,1	77	$V_{50} - V_{47}$	2,1
26	$V_{13} - V_{11}$	1,8	52	$V_{34} - V_{37}$				

Selanjutnya, penyelesaian perhitungan iterasi yang dapat dilihat pada Tabel 2.

**Tabel 2.** Perhitungan Iterasi Berdasarkan Algoritma Dijkstra

Vertex	Bobot	Vertex	Bobot	Vertex	Bobot	Vertex	Bobot	Vertex	Bobot
1	0	11	$\infty$	21	$\infty$	31	$\infty$	41	$\infty$
2	0,95	12	$\infty$	22	$\infty$	32	$\infty$	42	$\infty$
3	0,9	13	$\infty$	23	$\infty$	33	$\infty$	43	$\infty$
4	1,5	14	$\infty$	24	$\infty$	34	$\infty$	44	$\infty$
5	0,65	15	$\infty$	25	$\infty$	45	$\infty$	45	$\infty$
6	$\infty$	16	2,15	26	$\infty$	36	$\infty$	46	$\infty$
7	$\infty$	17	$\infty$	27	$\infty$	37	$\infty$	47	$\infty$
8	$\infty$	18	$\infty$	28	$\infty$	38	$\infty$	48	$\infty$
9	$\infty$	19	$\infty$	29	$\infty$	39	$\infty$	49	$\infty$
10	$\infty$	20	$\infty$	30	$\infty$	40	$\infty$	50	$\infty$

Iterasi 1 pada Tabel 2. dapat dijelaskan bahwa simpul  $V_0$  terhubung dengan simpul  $V_1, V_2, V_3$  dan  $V_4$  dengan bobot  $V_0 - V_1 = 0,95$ ,  $V_0 - V_2 = 0,90$ ,  $V_0 - V_3 = 1,5$  dan  $V_0 - V_4 = 0,65$ . Karena simpul  $V_4$  bobot terkecil dari semua titik yang berhubungan dengan  $V_0$  maka simpul  $V_4$  memiliki label tetap atau permanen yang artinya simpul tersebut memiliki nilai yang lebih kecil. Selanjutnya turunkan simpul  $V_4$ . Sehingga diperoleh iterasi selanjutnya yang dapat dilihat pada Tabel 3.

**Tabel 3.** Lanjutan Perhitungan Iterasi berdasarkan Algoritma Dijkstra

Vertex	Bobot	Vertex	Bobot	Vertex	Bobot	Vertex	Bobot	Vertex	Bobot
1	0	11	$\infty$	21	$\infty$	31	$\infty$	41	$\infty$
2	0,95	12	$\infty$	22	$\infty$	32	$\infty$	42	$\infty$
3	0,9	13	$\infty$	23	$\infty$	33	$\infty$	43	$\infty$
4	1,5	14	$\infty$	24	$\infty$	34	$\infty$	44	$\infty$
5	0,65	15	$\infty$	25	$\infty$	45	$\infty$	45	$\infty$
6	$\infty$	16	2,15	26	$\infty$	36	$\infty$	46	$\infty$
7		17	$\infty$	27	$\infty$	37	$\infty$	47	$\infty$
8	$\infty$	18	$\infty$	28	$\infty$	38	$\infty$	48	$\infty$
9	$\infty$	19	$\infty$	29	$\infty$	39	$\infty$	49	$\infty$
10	$\infty$	20	$\infty$	30	$\infty$	40	$\infty$	50	$\infty$

Iterasi 2 pada Tabel 3. dapat dijelaskan bahwa simpul  $V_4$  terhubung dengan simpul  $V_{16}$  dengan bobot  $V_4 - V_{16} = 1,5$  kemudian tambahkan dengan bobot yang sudah diwarnai ( $0,65 + 1,5 = 2,15$ ). Karena simpul  $V_2$  bobot terkecil maka simpul  $V_2$  memiliki label tetap atau permanen yang artinya simpul tersebut memiliki nilai yang lebih kecil, selanjutnya turunkan simpul  $V_2$ . Sehingga diperoleh iterasi selanjutnya.

Dengan melakukan langkah yang sama pada setiap iterasi sampai mencapai batas maksimal iterasi  $K = 50$  maka diperoleh hasil perhitungan rute terpendek menggunakan Algoritma Dijkstra yaitu  $V_0 - V_4 - V_{16} - V_{17} - V_{18} - V_{15} - V_{13} - V_{14} - V_{11} - V_{12} - V_{49} - V_{50} - V_{48} - V_{47} - V_{46} - V_{43} - V_{38} - V_{36} - V_{35} - V_{45} - V_{44} - V_{41} - V_{39} - V_{40} - V_{37} - V_{34} - V_{33} - V_{31} - V_{32} - V_{28} - V_{26} - V_{25} - V_{27} - V_{24} - V_{29} - V_{23} - V_{22} - V_{21} - V_{19} - V_{20} - V_3 - V_5 - V_{30} - V_6 - V_8 - V_9 - V_2 - V_7 - V_{10} - V_1$ , dengan total jarak keseluruhan 304,9 Km.

### 3.2 Menentukan Rute Terpendek Menggunakan Algoritma Greedy

Selanjutnya akan dilakukan pencarian lintasan terpendek dari graf berbobot yang telah terbentuk dengan menerapkan Algoritma Greedy. Algoritma greedy yaitu algoritma yang memecahkan masalah langkah demi langkah dan merupakan salah satu metode untuk masalah optimasi. Beberapa permasalahan mungkin tidak memiliki solusi yang efektif, tetapi algoritma Greedy dapat memberikan solusi yang hampir optimal dan efektif. Solusi yang dapat dibentuk dengan algoritma Greedy setiap langkah yaitu dalam setiap langkah, pilihan harus merupakan pilihan terbaik, pilihan yang dibuat dalam satu langkah tidak dapat diubah pada langkah berikutnya, kemudian Metode yang digunakan oleh algoritma greedy adalah dengan menentukan local optimum pada setiap langkah. Masalah optimasi algoritma greedy ditentukan oleh bagian-bagian berikut [14]. Komponen-komponen dari Algoritma Greedy adalah:

- 1) Himpunan kandidat. Set berisi komponen yang membuat solusi. Untuk masalah lintasan terpendek graf, himpunan kandidat ini adalah himpunan simpul dari graf.
- 2) Himpunan solusi. Set solusi berisi kandidat yang dipilih sebagai solusi untuk masalah tersebut. Himpunan ini merupakan bagian dari himpunan kandidat atau kandidat tetapi tidak semua atau dengan kata lain himpunan penyelesaian ini merupakan bagian dari himpunan kandidat.
- 3) Fungsi seleksi. Fungsi seleksi diwakili oleh predikat seleksi. Pemilihan predikat adalah fungsi yang ada pada setiap langkah, memilih kandidat yang paling mungkin memenuhi solusi

optimal. Kandidat yang terpilih pada satu tahap tidak akan dipertimbangkan lagi pada tahap selanjutnya.

- 4) Fungsi kelayakan. Fitur yang menunjukkan apakah kandidat terpilih dapat memberikan solusi yang layak.
- 5) Fungsi tujuan. Tindakan yang memaksimalkan dan meminimalkan nilai solusi. Tujuannya adalah untuk memilih solusi terbaik dari setiap komponen himpunan solusi.

Perhitungan akan dilakukan dari simpul awal yaitu Kantor Pos Gorontalo menuju ke alamat untuk penyaluran barang dari setiap simpul sebagai berikut [15]:

- 1) Periksa semua sisi yang berbatasan langsung dengan simpul; a) Pilih tepi dengan bobot terendah. Tepi ini pertama-tama menjadi jalur terpendek, sebut saja  $L(1)$ .
- 2) Hitung jalur terpendek kedua sebagai berikut:
- 3) Menghitung:  $d(i) = \text{panjang } L(1) + \text{bobot sisi dari simpul terakhir } L(1) \text{ ke simpul kedua } i$ .
- 4) Pilih  $d(i)$  terkecil. Bandingkan  $d(i)$  dengan bobot sisi ( $a; i$ ). Jika bobot sisi ( $a; i$ ) lebih kecil dari  $d(i)$ , maka  $L(2) = L(1) \cup$  (tepi dari simpul terakhir di  $L(i)$  adalah simpul ( $i$ )).
- 5) Demikian pula, ulangi langkah 2 untuk menentukan rute terpendek berikutnya.

Berdasarkan langkah-langkah pada Algoritma Greedy maka diperoleh proses iterasi sebagai berikut:

- 1) Proses perhitungan Iterasi 1

Himpunan Kandidat: Kantor Pos Gorontalo ke Kelurahan Ipilo.

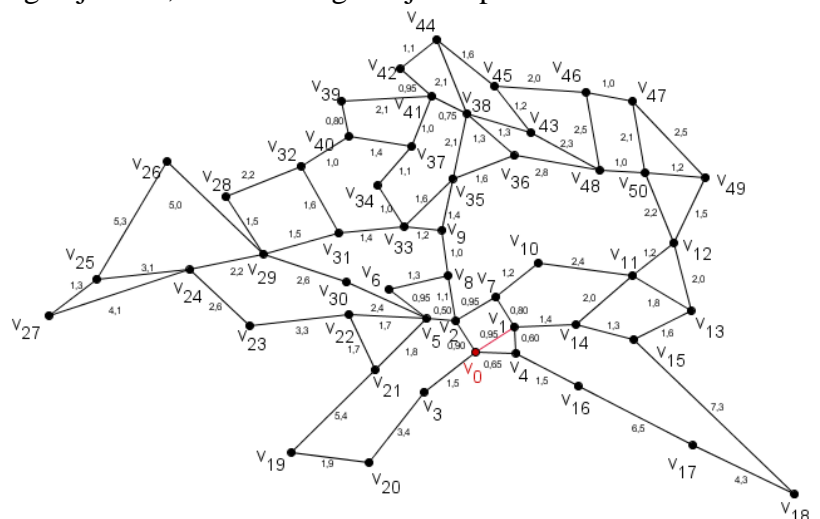
Himpunan Solusi: simpul  $V_0 - V_1$

Fungsi Seleksi :  $V_0 - V_1$

Fungsi Kelayakan:  $V_0 - V_1$

Fungsi Objektif:  $V_0 - V_1$ .

Karena simpul  $V_0$  terhubung dengan simpul  $V_1$  untuk menuju ke simpul  $V_1$  dengan bobot lebih optimal maka hasil yang diperoleh untuk menentukan jalur terpendek menggunakan algoritma greedy simpul  $V_0 - V_1$ , dengan jarak 0,95 Km. Yang disajikan pada Gambar 2.



**Gambar 2.** Perhitungan Iterasi berdasarkan Algoritma Greedy

- 2) Proses perhitungan Iterasi 2

Himpunan Kandidat: Kantor Pos Gorontalo ke Kelurahan Biawao.

Himpunan Solusi: simpul  $V_0 - V_2$

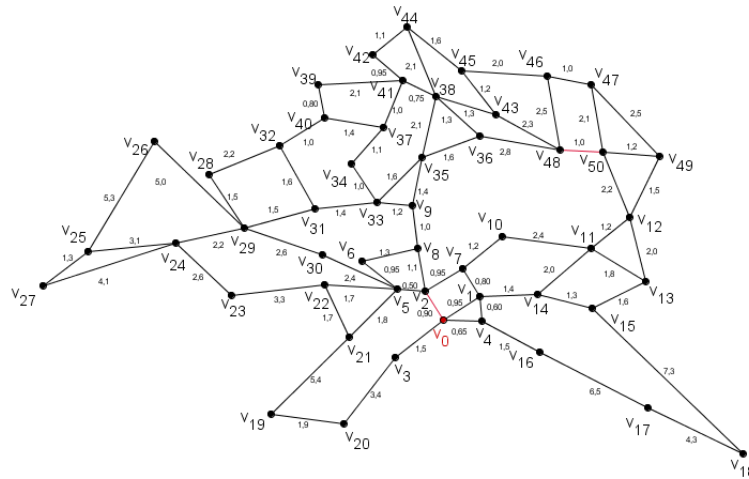
Fungsi Seleksi :  $V_0 - V_2$

Fungsi Kelayakan:  $V_0 - V_2$

Fungsi Objektif:  $V_0 - V_2$ .



Karena simpul  $V_0$  terhubung dengan simpul  $V_2$  untuk menuju ke simpul  $V_2$  dengan bobot lebih optimal maka hasil yang diperoleh menentukan jalur terpendek menggunakan algoritma greedy melalui simpul  $V_0 - V_2$ , dengan jarak 0,90 Km. Yang disajikan pada Gambar 3.



**Gambar 3.** Lanjutan perhitungan Iterasi berdasarkan Algoritma Greedy

Dengan melakukan langkah yang sama untuk melakukan perhitungan antar kelurahan maka hasil perhitungan di dapatkan sampai iterasi ke 50 dengan total jarak keseluruhan 441,6 Km.

### 3.3 Perbandingan Algoritma Dijkstra dan Algoritma Greedy

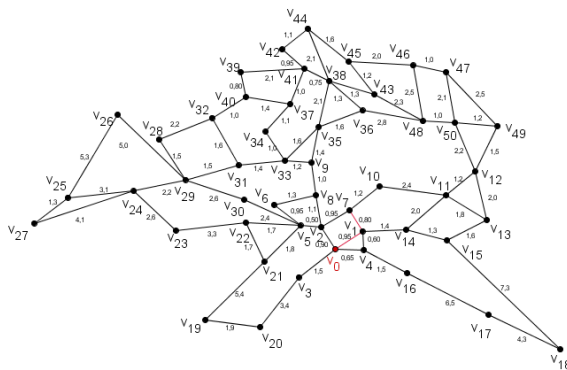
Untuk melihat perbandingan rute pada kelurahan yang di tampilkan dengan menerapkan kedua algoritma disajikan pada Tabel 4.

**Tabel 4.** Perbandingan Rute

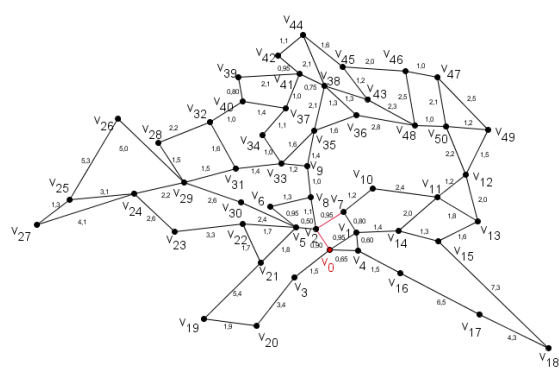
Kelurahan	Algoritma Dijkstra	Algoritma Greedy
Heledulaa Selatan	$V_0 - V_1 - V_7$	$V_0 - V_2 - V_7$
Libuo	$V_0 - V_2 - V_8 - V_9 - V_{33} - V_{31}$	$V_0 - V_2 - V_5 - V_{30} - V_{29} - V_{31}$
Bulotadaa	$V_0 - V_2 - V_8 - V_9 - V_{35} - V_{36} - V_{38} - V_{44}$	$V_0 - V_2 - V_7 - V_{10} - V_{11} - V_{12} - V_{49} - V_{50} - V_{48} - V_{43} - V_{45} - V_{44}$
Pilolodaa	$V_0 - V_2 - V_5 - V_{30} - V_{29} - V_{24} - V_{27}$	$V_0 - V_2 - V_5 - V_{22} - V_{23} - V_{24} - V_{25} - V_{27}$
Tomulabutao Selatan	$V_0 - V_2 - V_8 - V_9 - V_{33} - V_{34} - V_{37} - V_{40}$	$V_0 - V_2 - V_7 - V_{10} - V_{11} - V_{12} - V_{49} - V_{50} - V_{48} - V_{43} - V_{38} - V_{41} - V_{39} - V_{40}$

Selanjutnya akan disajikan ke dalam bentuk Gambar Graf berikut:

#### 1. Rute Kelurahan Heledulaa Selatan



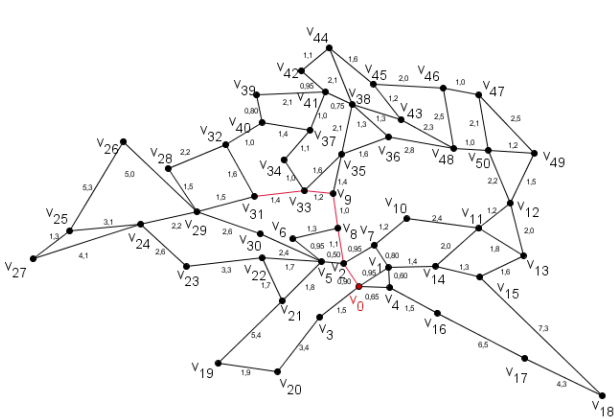
**Gambar 4.** Rute dengan Algoritma Dijkstra



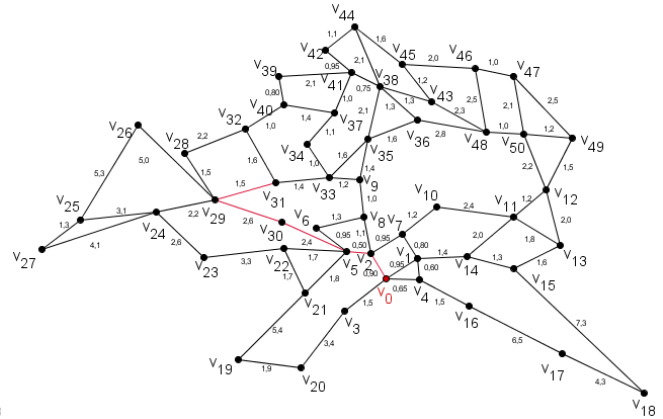
**Gambar 5.** Rute dengan Algoritma Greedy



2. Rute Kelurahan Libuo

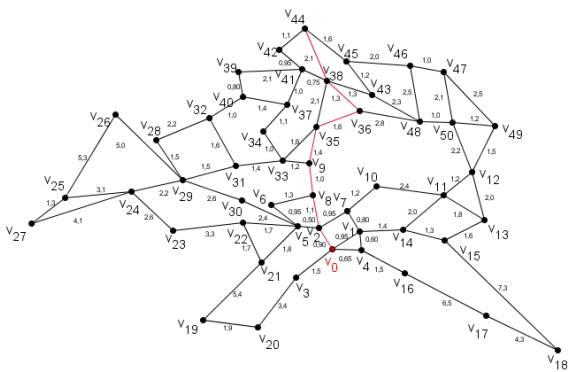


Gambar 6. Rute dengan Algoritma Dijkstra

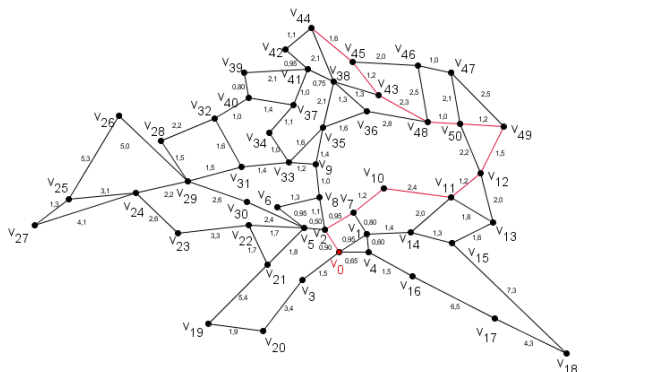


Gambar 7. Rute dengan Algoritma Greedy

3. Rute Kelurahan Bulotadaa

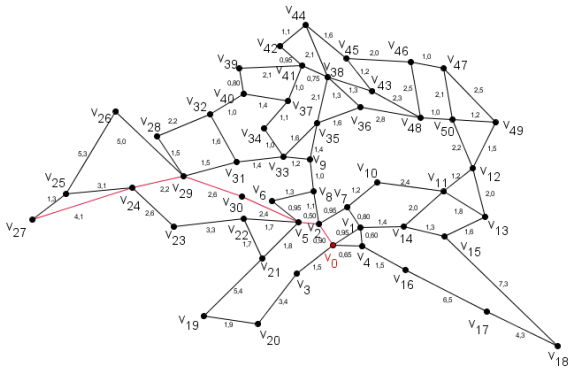


Gambar 8. Rute dengan Algoritma Dijkstra

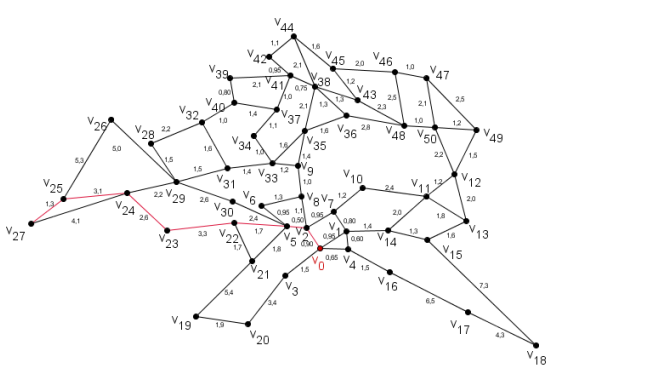


Gambar 9. Rute dengan Algoritma Greedy

4. Rute Kelurahan Pilolodaa

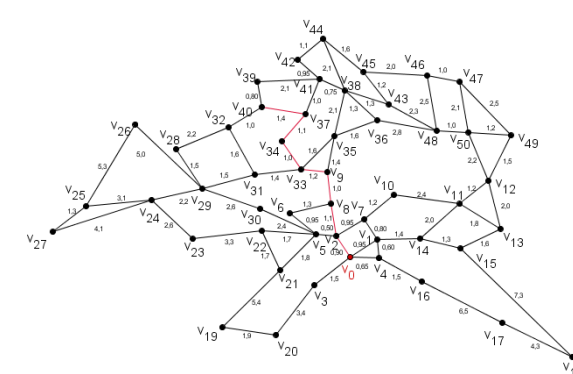


Gambar 10. Rute dengan Algoritma Dijkstra

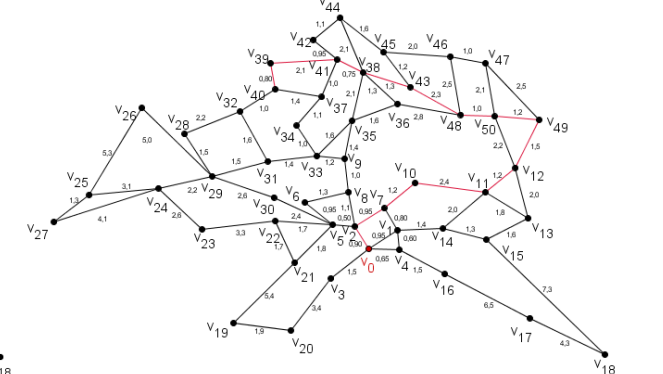


Gambar 11. Rute dengan Algoritma Greedy

5. Rute Kelurahan Tomulabutao Selatan



Gambar 12. Rute dengan Algoritma Dijkstra



Gambar 13. Rute dengan Algoritma Greedy

Berdasarkan hasil perbandingan, rute optimal yang diperoleh dari perhitungan masing-masing algoritma memiliki hasil yang berbeda, dimana Algoritma Dijkstra memiliki jarak tempuh yang lebih kecil jika dibandingkan dengan jarak tempuh pada Algoritma Greedy. Dalam proses pencarian iterasi dari kedua algoritma, Algoritma Dijkstra bekerja secara menyeluruh terhadap alternatif fungsi yang ada, hingga rute terpendek tidak hanya dari simpul awal ke simpul tujuan, tetapi jalur terpendek dapat diperoleh dari semua simpul. Dimana masing-masing simpul memiliki bobot jarak yang telah ditentukan. Hal ini karena algoritma Dijkstra melibatkan penempatan label ke setiap simpul. Misalkan  $L_v$  menyatakan label dari simpul  $v$ . Dari setiap penjelasan, beberapa simpul memiliki label sementara dan yang lain memiliki label tetap. Dalam mendeskripsikan algoritma, simpul dengan pengidentifikasi tetap dilingkari. Setiap iterasi dari algoritma tersebut mengubah status label dari sementara menjadi tetap, akibatnya algoritma dapat berakhir ketika simpul yang nilai lebih kecil menerima sebuah label tetap. Sedangkan algoritma Greedy tidak sepenuhnya bekerja pada semua alternatif fungsi yang ada, sehingga diperoleh rute terpendek hanya dari simpul awal ke simpul tujuan. Algoritma Greedy kurang efisien saat digunakan untuk mencari rute terpendek dengan banyak simpul yang bercabang karena Algoritma Greedy tergantung pada bobot setiap simpulnya, sehingga rute yang dihasilkan kurang optimal. Algoritma greedy hanya dapat di implementasikan pada kasus pencarian rute terpendek yang memiliki jarak antar simpul tidak banyak atau pendek. Oleh karena itu, algoritma Dijkstra merupakan algoritma yang optimal dalam menentukan rute pengiriman barang pada Kantor Pos Gorontalo dengan memperhatikan rute terpendek.

#### 4. Kesimpulan

Berdasarkan hasil analisis dapat disimpulkan bahwa dalam menentukan rute optimal dengan menerapkan Algoritma Dijkstra memiliki total jarak tempuh yaitu 304,90 Km. Hasil rute dengan menerapkan Algoritma Greedy memiliki total jarak tempuh yaitu 441,60 Km. Hal ini menunjukkan bahwa Algoritma Dijkstra merupakan algoritma yang dapat memberikan hasil optimal dalam menentukan rute pengiriman barang pada Kantor Pos Gorontalo dengan memperhatikan jarak terpendek.

#### Referensi

- [1] L. Maro, and L. K. S. Purab, "Penerapan Konsep Pewarnaan Graf dalam Penyusunan Jadwal Perkuliahan Menggunakan Metode Algoritma Welch-Powell pada Program Studi Teknik Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Tribuana Kalabahi," *Jurnal Ilmiah Wahana Pendidikan (JIWP)*, vol. 7, no. 6, pp. 193–197, 2021. doi: <https://doi.org/10.5281/zenodo.5574383>
- [2] J. Siang, *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*. Yogyakarta: Andi, 2009.
- [3] N. Nurdianti, *Penerapan Algoritma Greedy Untuk Menentukan Rute Terpendek Antar Klinik Gigi*. Skripsi: Universitas Islam Negeri Sumatra Utara, 2020.
- [4] M. K. Harahap, and N. Khairina, "Pencarian Jalur Terpendek dengan Algoritma Dijkstra. *Sinkron*, vol. 2, no. 2, p. 18, 2017. doi: <https://doi.org/10.33395/sinkron.v2i2.61>
- [5] I. K. Budayasa, *Teori Graph dan Aplikasinya*. Surabaya: Unesa University Press, 2007.
- [6] Y. D. Mahendra, N. Nuryanto, and A. Burhanuddin, "Sistem Penentuan Jarak Terdekat Dalam Pengiriman Darah Di Pmi Kota Semarang Dengan Metode Algoritma Greedy," *Jurnal Komtika*, vol. 2, no. 2, pp. 136–142, 2019. doi: <https://doi.org/10.31603/komtika.v2i2.2601>
- [7] Y. Darnita, and R. Toyib, "Penerapan Algoritma Greedy Dalam Pencarian Jalur Terpendek Pada Instansi-Instansi Penting Di Kota Argamakmur Kabupaten Bengkulu Utara," *Jurnal Media Infotama*, vol. 15, no. 2, pp. 57–64, 2019. doi: <https://doi.org/10.37676/jmi.v15i2.867>
- [8] Pos Indonesia, <http://www.posindonesia.co.id/index.php/profil-perusahaan/sejarah-pos>, 2017.

- [9] M. Z. Usman, and T. Oktiarso, "Implementasi Algoritma Greedy Untuk Menyelesaikan Travelling Salesman Problem di Distributor PT. Z," *Journal of Integrated System*, vol. 1, no. 2, pp. 216–229, 2019. doi: <https://doi.org/10.28932/jis.v1i2.1049>
- [10] A. Cantona, F. Fauziah, and W. Winarsih, "Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta," *Jurnal Teknologi Dan Manajemen Informatika*, vol. 6, no. 1, pp. 27–34, 2020. doi: <https://doi.org/10.26905/jtmi.v6i1.3837>
- [11] N. 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." *Jurnal Matematika MANTIK*, vol. 7, no. 1, pp. 31–40, 2021. doi: <https://doi.org/10.15642/mantik.2021.7.1.31-40>
- [12] F. N. Arviantino, W. Gata, L. Kurniawati, Y. A. Setiawan, and D. Priansyah, "Penerapan Algoritma Greedy Dalam Pencarian Jalur Terpendek Pada Masjid–Masjid Di Kota Samarinda," *Metik Jurnal*, vol. 5, no. 1, pp. 8–11, 2021. doi: <https://doi.org/10.47002/metik.v5i1.188>
- [13] J. S. Iskandar, and Y. F. Riti, "Perbandingan Algoritma Greedy dan Algoritma Dijkstra dalam Pencarian Rute Terpendek dari Kabupaten Tuban ke Kota Surabaya," *PETIK: Jurnal Pendidikan Teknologi Informasi dan Komunikasi*, vol. 8, no. 2, pp. 96–106, 2022. doi: <https://doi.org/10.31980/jpetik.v8i2.1756>
- [14] F. Hartanto, and Y. Safitri, "Rancang bangun aplikasi pencarian lokasi wisata Kota Bogor Menggunakan algoritma Greedy Berbasis Android," *Jurnal Techno Nusa Mandiri*, vol. 11, no. 2, pp. 169–175, 2014
- [15] S. Oktaviana, and A. Naufal, "Algoritma Greedy untuk Optimalisasi Ruang dalam Penyusunan Jadwal Perkuliahan," *Multinetics*, vol. 3, no. 1, pp. 54–59, 2017. doi: <https://doi.org/10.32722/vol3.no1.2017.pp54-59>