

PENERAPAN ALGORITMA DIJKSTRA DALAM MENENTUKAN RUTE TERPENDEK UNTUK JASA PENGIRIMAN BARANG DI PALANGKA RAYA

Edward Christian Rufus, Raydamar Rizkyaka Riyadi, Dicky Nugraha Hasibuan,

Efrans Christian, Viktor Handrianus Pranatawijaya

Teknik Informatika, Universitas Palangka Raya

Tunjung Nyaho Jl. Yos Sudarso Palangka Raya, Indonesia

edwardchristianrufus@mhs.eng.upr.ac.id

ABSTRAK

Dengan pertumbuhan pesat industri e-commerce di Palangka Raya, perusahaan jasa pengiriman paket express menghadapi tuntutan pengiriman yang lebih cepat dan efisien. Salah satu tantangan utama adalah memilih rute pengiriman yang optimal untuk menghindari pemborosan waktu dan biaya akibat jarak tempuh yang jauh, waktu tempuh lama, dan konsumsi bahan bakar berlebih. Hal ini dapat berdampak pada kepuasan pelanggan dan daya saing perusahaan. Salah satu kunci untuk mencapai hal ini adalah dengan memilih rute pengiriman yang optimal. Algoritma Dijkstra adalah algoritma yang terkenal menemukan jalur terpendek antara dua titik. Penelitian ini berfokus mengimplementasikan algoritma Dijkstra untuk mengoptimalkan rute pengiriman paket express berdasarkan peta rute yang dimodelkan ke dalam bentuk graf. Algoritma ini akan digunakan untuk menemukan rute terpendek antara gudang paket express dan alamat tujuan. Hasil dari penelitian menunjukkan bahwa algoritma Dijkstra mampu menciptakan rute terpendek dari peta rute yang diberikan, sehingga menghemat waktu dan biaya pengiriman.

Kata kunci : *Jasa Pengiriman Paket, Algoritma Dijkstra, Shortest Path, Efisiensi Waktu, Efisiensi Biaya*

1. PENDAHULUAN

Permasalahan pada pencarian jalur terpendek dapat ditemukan pada jasa pengiriman paket express. Di Palangka Raya, industri pengiriman paket express mengalami pertumbuhan pesat seiring perkembangan e-commerce. Menurut Deddy, et al[1], Pergeseran menuju belanja online terjadi karena kemudahan, kenyamanan, penghematan biaya, penghematan waktu dan pengiriman cepat dibandingkan dengan belanja konvensional atau tradisional. Penghematan waktu dan pengiriman cepat sebagai ciri khas e-commerce menuntut perusahaan pengiriman paket express meningkatkan efisiensi dan efektivitas layanannya. Rute yang tidak optimal dapat menyebabkan pemborosan waktu dan biaya, seperti jarak tempuh yang jauh, waktu tempuh yang lama, dan konsumsi bahan bakar yang berlebihan. Hal ini dapat berdampak pada kepuasan pelanggan dan daya saing perusahaan. Berdasarkan observasi yang dilakukan oleh Amanda, et al[2], terhadap pencapaian kinerja kurir J&T Express DC Palangka Raya, di tahun 2021 terdapat 2 orang kurir yang diberhentikan dan pada tahun 2022 terdapat 1 orang kurir yang diberhentikan, hal ini dikarenakan selama 3 bulan kurir tersebut tidak dapat memenuhi target yang ditetapkan oleh perusahaan. Permasalahan yang ditemukan muncul dari adanya komplain dari customer dalam hal waktu pengantaran barang yang terbilang cukup lama[2]. Oleh sebab itu kurir express membutuhkan sebuah cara dalam pemilihan rute pengiriman paket. Salah satu faktor kunci untuk mengatasi hal tersebut adalah optimasi rute pengiriman. Hasil dari menentukan rute terpendek dapat menjadi bahan peninjauan dalam pengambilan keputusan untuk menunjukkan lintasan yang akan ditempuh..

Algoritma Dijkstra merupakan salah satu algoritma yang paling mudah untuk diimplementasikan karena algoritma ini merupakan bagian dari program dinamis yang menemukan semua jalur terpendek antara setiap kemungkinan pasangan lokasi yang berbeda dan sangat efektif [5] dalam menyelesaikan masalah jalur optimal untuk semua pasangan simpul.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Terdapat beberapa penelitian terkait masalah optimasi menggunakan algoritma ini diantaranya adalah Cantona, et al[4], yang menilai pencarian rute terpendek di Jakarta menggunakan smartphone dapat membantu wisatawan menghemat waktu saat mengunjungi museum. Mereka menemukan bahwa Algoritma Dijkstra menghasilkan rute terpendek yang hanya membutuhkan tujuh poin setara dengan 35% dari total perjalanan. Berdasarkan uji coba yang dilakukan oleh Arthalia, et al[7] menggunakan algoritma Dijkstra, pada sampel uji yang diambil dengan lokasi tujuan yang berbeda, hasil pengujian pada aplikasi dapat menampilkan tujuan jalur dari titik koordinat posisi pengguna. Algoritma Dijkstra memiliki kemampuan yang efektif untuk mencari jalur terpendek, karena dalam algoritma tersebut setiap graf dipilih dengan bobot minimum yang menghubungkan sebuah node terpilih dengan node lain yang tidak terpilih.

2.2. Graf

Graf adalah kumpulan simpul (vertices atau nodes) yang dihubungkan satu sama lain melalui sisi atau busur (edges)[3]. Graf dapat direpresentasikan sebagai pasangan himpunan (V, E) , di mana V adalah

himpunan simpul dan E adalah himpunan sisi yang menghubungkan pasangan simpul dalam graf tersebut[8].

2.2. Graf Berbobot

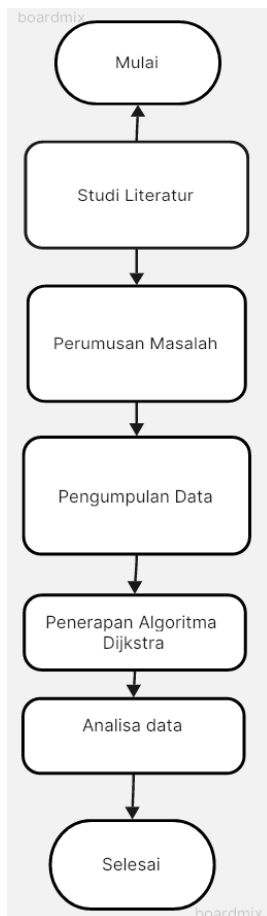
Graf yang masing-masing rusuknya memiliki suatu nilai “bobot” yang non-negatif[9]. Bobot pada tiap sisi dapat berbeda-beda tergantung pada masalah yang dimodelkan dengan graf[10].

2.3. Algoritma Dijkstra

Algoritma Dijkstra merupakan algoritma yang sangat populer untuk menemukan jalur terpendek [5] dari suatu titik ke semua titik lain dalam sebuah graf. Algoritma ini menggunakan strategi Greedy, yang berarti pada setiap langkah, sisi yang berbobot minimum yang menghubungkan sebuah simpul yang sudah terpilih dengan sebuah simpul lain yang belum terpilih akan diambil[3]. Cara kerja algoritma Dijkstra dalam mencari jarak terpendek adalah perhitungan dari titik asal ke titik terdekat, kemudian ke titik kedua, dan seterusnya[7].

3. METODE PENELITIAN

Tahap-tahap yang akan dilakukan dalam penulisan artikel ini dapat dilihat melalui flowchart berikut.



Gambar 1. Flowchart metodologi

Dalam penelitian kuantitatif ini, dikaji suatu model di mana sebuah perusahaan X yang bergerak dalam bidang jasa pengiriman barang mengirimkan paketnya melalui seorang kurir. Penulis mengambil sampel daerah kelurahan Langkai, kecamatan Pahandut, Palangkaraya dari Google Maps. Fokus utama dari sampel map tersebut adalah percabangan jalan yang terhubung pada bangunan JNE di jalan Seth Adji.

Kemudian, penulis mengimplementasikan graf yang dibuat dari sampel peta daerah ke dalam program yang memanfaatkan algoritma Dijkstra berdasarkan pseudocode ini[6].

```

    INITIALIZE-SINGLE-SOURCE (G, s)
    S = ∅
    Q = V
    while Q ≠ ∅ ;
    DIJKSTRA (G, w, s)
    u = EXTRACT-MIN(Q)
    S = S ∪ {u}
    for each vertex v ∈ G.Adj[u]
    RELAX(u, v, w)
  
```

Algoritma Dijkstra adalah algoritma greedy yang digunakan untuk menemukan jalur terpendek antara dua simpul dalam graf berbobot [7] . Berikut adalah tahapan-tahapan yang dilalui dalam algoritma Dijkstra:

- a. Inisialisasi
 - Tetapkan jarak dari simpul awal ke simpul awal = 0
 - Tetapkan jarak dari simpul awal ke simpul lainnya = tak terhingga
 - Buat kumpulan simpul yang telah dikunjungi (S) = kosong
 - Buat kumpulan simpul yang belum dikunjungi (Q) = semua simpul
- b. Pemilihan Simpul
 - Dari kumpulan simpul yang belum dikunjungi (Q), pilih simpul dengan jarak terkecil dari simpul awal. Misalkan simpul ini sebagai u.
- c. Relaksasi
 - Untuk setiap tetangga v dari simpul u yang belum dikunjungi:
 - Hitung jarak alternatif dari simpul awal ke v melalui u
 - Jika jarak alternatif lebih kecil dari jarak sebelumnya, perbarui jarak dari simpul awal ke v
- d. Pembaruan Kumpulan
 - Pindahkan simpul u dari kumpulan Q ke kumpulan S
- e. Pengulangan
 - Jika kumpulan Q masih tidak kosong, ulangi langkah 2-4. Jika kosong, algoritma selesai.

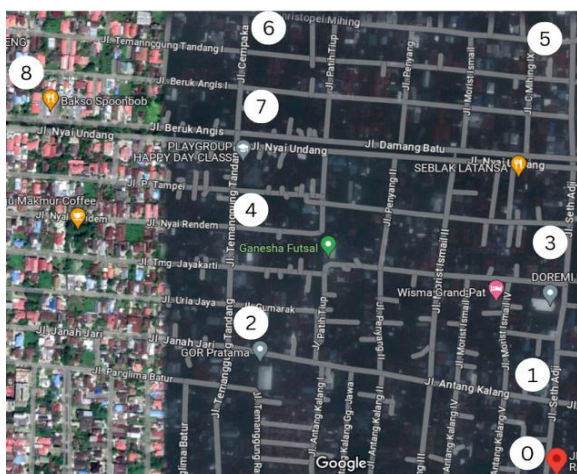
Pada setiap iterasi, algoritma memastikan bahwa jarak terpendek dari simpul awal ke simpul yang telah dikunjungi (di S) adalah terpendek. Ketika algoritma selesai, jarak terpendek dari simpul awal ke semua simpul lainnya telah ditemukan.

Algoritma Dijkstra menggunakan pendekatan greedy dengan memilih simpul terdekat dari simpul awal pada setiap iterasi. Ini memastikan bahwa setiap simpul hanya dikunjungi sekali dan jalur yang dihasilkan adalah jalur terpendek.

Pseudocode yang diberikan dalam artikel menunjukkan implementasi algoritma Dijkstra secara umum, dengan inisialisasi awal, ekstraksi simpul minimum, relaksasi, dan pembaruan kumpulan simpul yang telah dikunjungi.

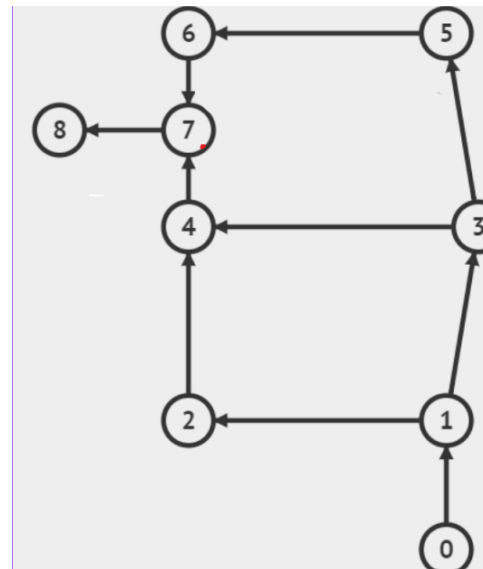
Bagian penting dari penelitian ini adalah aplikasi praktis dari algoritma Dijkstra dalam konteks optimalisasi rute pengiriman. Maka dari itu, penulis akan mengimplementasikan program yang memanfaatkan algoritma Dijkstra dalam bahasa javascript. Program ini bertujuan untuk memvisualisasikan graf berbobot dan melakukan perhitungan menggunakan program tersebut. Terdapat 2 file yang telah dibuat untuk menciptakan program ini, yaitu SSSPWidget.js dan sssp.html. File pertama bertugas melakukan kalkulasi dan pengimplementasian algoritma Dijkstra serta menggambar graf yang akan ditampilkan program. File ini ibarat otak program yang memproses data dan menghasilkan visualisasi graf. File kedua yaitu sssp.html. File ini berperan sebagai file utama program dan bertanggung jawab untuk menjalankan GUI (antarmuka pengguna grafis). GUI adalah bagian yang dilihat dan diinteraksikan oleh pengguna untuk menjalankan program. Terakhir, penulis mengambil hasil dari program yang telah dijalankan dan memasukkan data berupa jarak antar simpulnya ke dalam tabel hasil.

4. HASIL DAN PEMBAHASAN



Gambar 2. Peta jalur

Peta jalur perjalanan di atas dapat direpresentasikan dengan menggunakan graf berarah yang menggambarkan rute pengiriman barang yang harus dilakukan kurir perusahaan X.



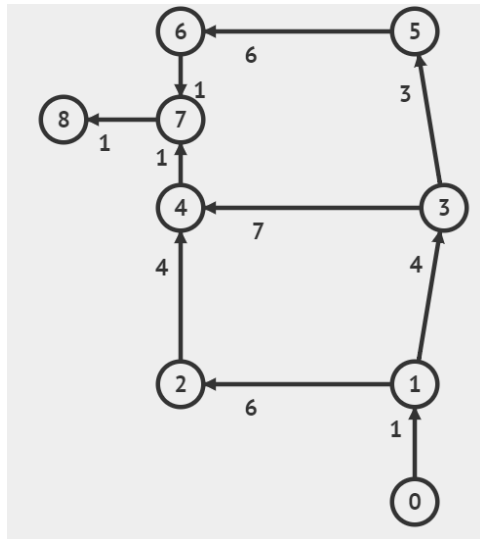
Gambar 3. Rute pengiriman barang

Langkah pertama dalam menentukan rute pengiriman barang adalah dengan memodelkan peta rute pengiriman paket ke dalam struktur yang memungkinkan evaluasi yang lebih rinci. Struktur tersebut dapat direpresentasikan sebagai graf berbobot. Jenis graf ini memungkinkan penulis untuk mengaplikasikan Algoritma Dijkstra. Di sini, setiap simpul ditandai sebagai setiap persimpangan, terkecuali untuk simpul “0” dan “8” yang digunakan sebagai titik awal, dan titik akhir. Sedangkan setiap sisi merepresentasikan estimasi jarak tempuh dalam skala 1 : 10 meter.

Tabel 1. Keterangan simpul graf

Simpul	Keterangan
s0	Kantor Pusat
s1	Rute 1
s2	Rute 2
s3	Rute 3
s4	Rute 4
s5	Rute 5
s6	Rute 6
v7	Rute 7
v8	Tempat Tujuan

Tabel 1 merupakan keterangan dari letak delapan simpul yang ada pada graf yang diinterpretasikan dalam bentuk graf yang disajikan pada gambar 3.



Gambar 4. Graf berbobot titik pengiriman

Untuk mengaplikasikan algoritma Dijkstra, maka graf rute harus merupakan graf berbobot. Dalam menganalisis bobot pada sisi graf, penulis mempertimbangkan beberapa faktor yang mempengaruhi efisiensi pengiriman paket, dimulai dari jarak tempuh serta biaya bahan bakar yang digunakan dan faktor lainnya yang relevan, namun dari beberapa faktor yang sudah dievaluasi, telah ditetapkan bahwa jarak tempuh adalah faktor paling utama dalam menilai efisiensi pengiriman. Penulis dapat mengkonversi estimasi jarak tempuh menjadi estimasi penggunaan bahan bakar, walaupun banyak faktor yang dapat mempengaruhi penggunaan bensin dalam pengiriman paket seperti kondisi cuaca, dan kondisi jalan yang sedang dilalui. Dalam penelitian ini penulis membatasi variabel demi memastikan konsistensi data.

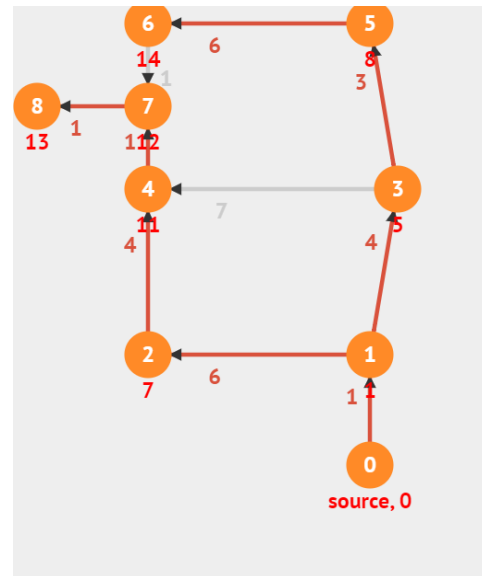
Kemudian, penulis akan mencari jalur atau sisi terpendek pada graf berbobot di atas dengan bantuan aplikasi yang telah dibuat untuk algoritma dijkstra.

Kode dimulai dengan mendefinisikan fungsi dijkstra yang menerima dua parameter: sourceVertex (vertex awal) dan versiontype (versi implementasi Dijkstra yang akan digunakan). Yang terakhir, SSSP.html berperan sebagai aplikasi antarmuka pengguna (GUI) untuk menerima input simpul awal dan simpul akhir, menghitung jalur terpendek menggunakan algoritma Dijkstra, dan menampilkan hasilnya baik dalam bentuk teks maupun visualisasi graf. Penulis memanfaatkan bahasa pemrograman html dan css untuk membuat GUI program.

Setelah memasukan graf sebelumnya ke dalam program, penulis dapat memulai program untuk melakukan visualisasi dan perhitungan algoritma dijkstra. Dengan menjalankan program javascript SSSPWidget.js, diperoleh hasil perhitungan rute terpendek menggunakan Algoritma Dijkstra sebagai berikut.

Program melakukan langkah yang sama untuk mencari sisi terpendek antar simpul dalam graf. Maka hasil perhitungan didapatkan bahwa rute terpendek

dari graf di atas adalah 0-1-2-4-7-8 dengan total bobot 13, dan rute kedua yaitu 0-1-3-5-6-7-8 dengan bobot 16. Berikut adalah tabel hasil jarak antar simpul graf tersebut.



Gambar 5. Rute terpendek

Tabel 2. Tabel jarak antar simpul dan total bobotnya

Simpul Tujuan	Simpul-simpul yang Dilalui	Total Bobot (1:10m) ke Simpul Tujuan
s0 – s1	s0 – s1	1
s0 – s2	s0 – s1 – s2	7
s0 – s3	s0 – s1 – s3	5
s0 – s4	s0 – s1 – s2 – s4	11
s0 – s5	s0 – s1 – s3 – s5	8
s0 – s6	s0 – s1 – s3 – s5 – s6	14
s0 – s7	s0 – s1 – s2 – s4 – s7	12
s0 – s8	s0 – s1 – s2 – s4 – s7 – s8	13

Untuk menguji kinerja algoritma Dijkstra dalam mengoptimalkan rute pengiriman barang, observasi dilakukan terhadap tiap iterasi dari algoritma Dijkstra dalam program yang digunakan. Pengujian ini bertujuan untuk mengukur efisiensi yang dapat dicapai dalam hal jarak tempuh. Hasil dari observasi yang telah dilakukan adalah sebagai berikut.

Langkah 1: Simpul awal adalah s0. Simpul terdekat dan yang hanya terhubung dengan s0 adalah s1 dengan jarak sebesar 1. Simpul yang dipilih adalah s1.

Langkah 2: Langkah dimulai pada simpul s1. Simpul yang terhubung dengan s1 adalah s2 dan s3. Program memprioritaskan s3 sebagai simpul terdekat dari s1 untuk diproses (total jarak s0 ke s3 adalah 5). Simpul lain yang terhubung dengan s1 adalah s2 dengan jarak sebesar 6 (total jarak s0 ke s2 adalah 7).

Langkah 3: Langkah dimulai pada simpul s3. Simpul yang terhubung ke s3 adalah s4 dengan total jarak dari simpul awal sebesar 12, dan s5 dengan total jarak

dari simpul awal sebesar 8. s5 dipilih sebagai simpul terdekat dari s3. Kemudian, program beralih ke s2 yang juga terhubung ke s4 dengan total jarak dari simpul awal sebesar 11. Rute ini lebih pendek daripada jalur s0-s1-s3-s4. Sehingga melalui s2, total jarak s0 ke s4 diperbarui.

Langkah 4: Langkah dimulai pada simpul s5. Simpul terdekat dan satu-satunya yang terhubung ke s5 adalah s6 dengan total jarak dari simpul awal yaitu 14. Kemudian program beralih ke s4, di mana simpul yang terhubung ke simpul ini adalah s7 dengan total jarak dari simpul awal yaitu 12.

Langkah 5: Langkah dimulai pada s7. s7 terhubung ke s8 dengan jarak total dari simpul awal sebesar 13. Kemudian program beralih ke s6. Jarak total dari simpul awal ke s7 melalui s6 adalah 15, yang mana nilai ini lebih besar daripada rute s0-s1-s2-s4-s7-s8 dengan total jarak 13. Sehingga, rute yang dipilih sebagai rute terpendek dari simpul awal hingga simpul akhir adalah s1-s2-s4-s7-s8. Proses algoritma selesai.

Menggunakan algoritma Dijkstra, program melakukan iterasi sebanyak 9 kali untuk menghitung jarak antar tiap simpul sekaligus menjumlahkan total jarak dari simpul awal.

5. KESIMPULAN DAN SARAN

Penelitian ini menunjukkan bahwa algoritma Dijkstra mampu mengoptimalkan rute pengiriman barang untuk jasa kurir express di Palangka Raya dengan mencari rute terpendek dari peta jalur yang dimodelkan ke dalam bentuk graf. Meskipun demikian, algoritma Dijkstra memiliki keterbatasan yaitu hanya mempertimbangkan jarak sebagai faktor utama dan membutuhkan waktu komputasi yang lebih lama untuk graf dengan ukuran yang lebih besar karena jumlah iterasinya juga akan semakin banyak. Saran untuk penelitian selanjutnya adalah untuk mengembangkan algoritma yang dapat mempertimbangkan faktor-faktor selain jarak seperti kondisi jalan dan kemacetan lalu lintas, serta meningkatkan efisiensi waktu komputasi. Penerapan algoritma optimasi rute seperti ini dapat membantu perusahaan jasa pengiriman meningkatkan kepuasan pelanggan dan daya saing dengan menyediakan layanan yang lebih cepat dan efisien.

DAFTAR PUSTAKA

- [1] Deddy Rakhmad Hidayat and Peridawaty Peridawaty, "ANALISIS FAKTOR-FAKTOR PERILAKU PEMBELIAN ONLINE (E-COMMERCE) KONSUMEN MAHASISWA FAKULTAS EKONOMI DAN BISNIS UNIVERSITAS PALANGKA RAYA," *Jurnal Ekonomi Pembangunan, Manajemen & Bisnis, Akuntansi*, vol. 1, no. 2, pp. 64–74, Jun. 2021, doi: <https://doi.org/10.52300/jemba.v1i2.2985>.
- [2] A. Dhea and None Aprilita, "Analysis of Workload, Courier Performance, and Work Motivation at the J&T Express DC Expedition Service Company in Palangka Raya," *Jurnal Manajemen Sains dan Organisasi/Jurnal Manajemen Sains Dan Organisasi*, vol. 4, no. 2, Aug. 2023, doi: <https://doi.org/10.52300/jmso.v4i2.10350>.
- [3] I. Djafar and Faizal, "SINGLE-SOURCE SHORTEST PATH PADA GRAF BERBOBOT MENGGUNAKAN ALGORITMA DIJKSTRA DAN BELLMAN-FORD," *SISITI : Seminar Ilmiah Sistem Informasi dan Teknologi Informasi*, vol. 4, no. 2, 2015, Accessed: Apr. 17, 2024. [Online]. Available: <https://ejournal.diponegara.ac.id/index.php/sisiti/article/view/144-152>
- [4] Cantona, A., Fauziah, F., and Winarsih, W., "Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta" *Jurnal Teknologi dan Manajemen Informatika*, vol. 6, no. 1, 2020, doi <https://doi.org/10.26905/jtmi.v6i1.3837>
- [5] H. Pratiwi, "Application Of The Dijkstra Algorithm To Determine The Shortest Route From City Center Surabaya To Historical Places," *Jurnal Teknologi Dan Sistem Informasi Bisnis/Jurnal Teknologi dan Sistem Informasi Bisnis*, vol. 4, no. 1, pp. 213–223, Jan. 2022, doi: <https://doi.org/10.47233/jteksis.v4i1.407>.
- [6] T. Cormen, C. Leiserson, R. Rivest, C. Stein, *INTRODUCTION TO ALGORITHMS, THIRD EDITION*. Massachusetts: The MIT Press, 2009. pp. 658.
- [7] I. Wulandari and P. Sukmasetyan, "Implementasi Algoritma Dijkstra untuk Menentukan Rute Terpendek Menuju Pelayanan Kesehatan," *Jurnal Ilmiah Sistem Informasi*, vol. 1, no. 1, pp. 30–37, Apr. 2022, doi: <https://doi.org/10.24127/jisi.v1i1.1953>.
- [8] N. Awalloedin, Windu Gata, and Nurul Qomariyah, "ALGORITMA DIJKSTRA DALAM PENETUAN RUTE TERPENDEK PADA JALAN RAYA ANTAR KOTA JAKARTA - TANGERANG," *Just IT : Jurnal Sistem Informasi, Teknologi Informasi dan Komputer*, vol. 13, no. 1, 2022, doi: <https://doi.org/10.24853/justit.13.1.%p>.
- [9] M. Yasin and B. Afandi, "SIMULASI MINIMUM SPANNING TREE GRAF BERBOBOT MENGGUNAKAN ALGORITMA PRIM DAN ALGORITMA KRUSKAL," vol. 2, no. 2, Nov. 2014, doi: <https://doi.org/10.0034/edu.v2i2.133>.
- [10] R. Aditya, "ALGORITMA 'HANCURKAN SEMUA SIKEL' UNTUK MENENTUKAN POHON PERENTANG MINIMUM DARI SUATU GRAF BERBOBOT," *Jurnal Ilmiah Matrik/Jurnal ilmiah matrik*, vol. 21, no. 2, pp. 91–98, Sep. 2019, doi: <https://doi.org/10.33557/jurnalmatrik.v21i2.571>.