

Penerapan Metode Dijkstra Dalam Menentukan Rute Terdekat Bengkel Di Kabupaten Nganjuk

¹Ahmad Rafi' Izzuddin, ²Ahmad Bagus Setiawan, ³Patmi Kasih

¹⁻³ Teknik Informatika, Universitas Nusantara PGRI Kediri

¹ahmrafi29@gmail.com, ²ahmadbagus@unpkediri.ac.id, ³fatkasih@gmail.com

Penulis Korespondens : Ahmad Rafi' Izzuddin

Abstrak—Penelitian ini bertujuan untuk merancang dan menerapkan Metode Algoritma Dijkstra dalam mencari rute jarak terdekat bengkel di Kabupaten Nganjuk. Manfaat penelitian ini mencakup penghematan waktu bagi pengguna saat mencari bengkel terdekat, menambah pengetahuan bagi peneliti mengenai penerapan metode ini, serta sebagai referensi bagi peneliti lain dan instansi terkait. Metode penelitian yang digunakan adalah kuantitatif, dengan fokus pada pengukuran jarak dan perhitungan rute terpendek.

Kata Kunci— Algoritma Dijkstra, Bengkel Terdekat, Kabupaten Nganjuk, Metode Kuantitatif, Rute Terpendek

Abstract— *This study aims to design and implement the Dijkstra algorithm to determine the shortest route to workshops in the Nganjuk Regency area. The application of this algorithm is expected to help users save time and travel distance when searching for the nearest workshop. A quantitative research method was employed, focusing on distance measurement between location points and the calculation of the most efficient routes using a road network graph. The results of this study are beneficial not only for users but also contribute to the researcher's understanding of Dijkstra algorithm implementation. Additionally, it may serve as a reference for future research and for relevant institutions in developing location-based information systems*

Keywords— *Dijkstra Algorithm, Nearest Workshop, Nganjuk Regenc , Quantitative Method, Shortest Route*

This is an open access article under the CC BY-SA License.



I. PENDAHULUAN

Kebutuhan akan transportasi yang efisien dan cepat menjadi salah satu tuntutan utama di era modern, termasuk di wilayah Kabupaten Nganjuk yang memiliki mobilitas tinggi. Kendaraan bermotor, baik sepeda motor maupun mobil, merupakan sarana transportasi vital bagi masyarakat dalam menunjang aktivitas sehari-hari. Namun, permasalahan umum yang sering dihadapi oleh para pengguna kendaraan adalah terjadinya kerusakan atau gangguan teknis di tengah perjalanan. Mencari lokasi bengkel terdekat secara manual seringkali menjadi kendala yang menyulitkan pengemudi, terutama ketika berada di daerah yang asing atau dalam kondisi darurat. Situasi ini dapat menimbulkan rasa panik, membuang-buang waktu, energi, dan bahkan berpotensi membahayakan keselamatan, khususnya saat malam hari atau di lokasi yang minim penerangan [1]. Selain itu, kondisi jalan yang tidak selalu mulus, seperti kemacetan atau rute yang berliku, dapat memperparah situasi dan menghambat pengemudi untuk segera mendapatkan bantuan perbaikan kendaraan.

Isu-isu terkait yang muncul dari permasalahan ini adalah kurangnya ketersediaan informasi yang akurat dan *real-time* mengenai lokasi bengkel terdekat yang dapat diakses dengan mudah oleh pengemudi. Pengemudi seringkali hanya bergantung pada petunjuk arah dari orang lain atau menggunakan aplikasi peta konvensional yang tidak secara spesifik menampilkan daftar bengkel beserta rute tercepat dan optimal menuju lokasi tersebut. Hal ini dapat berujung pada kerugian waktu, pemborosan bahan bakar, dan potensi risiko keamanan di jalan. Oleh karena itu, diperlukan sebuah sistem informasi geografis berbasis *mobile* yang dapat membantu pengemudi menemukan bengkel terdekat dengan rute yang paling efisien, sehingga dapat meminimalkan dampak negatif dari insiden kerusakan kendaraan di perjalanan.

Untuk mengatasi permasalahan penentuan rute terpendek, berbagai penelitian telah dilakukan dengan mengimplementasikan algoritma pencarian jalur. Algoritma Dijkstra adalah salah satu metode yang populer dan efektif untuk menemukan rute terpendek dari satu titik ke titik lainnya dalam sebuah graf berbobot non-negatif [7]. Algoritma ini bekerja secara iteratif untuk menemukan jarak minimum dari titik awal ke semua titik lain yang terhubung, menjamin ditemukannya rute terpendek jika semua bobot sisi bernilai positif [8]. Berikut adalah ulasan penelitian terdahulu yang relevan dengan topik ini, yang disajikan untuk menunjukkan "keadaan seni" dari subjek penelitian

Dalam upaya mengatasi permasalahan penentuan rute terpendek, berbagai penelitian telah dilakukan dengan mengimplementasikan algoritma pencarian jalur, di mana Algoritma Dijkstra menjadi salah satu metode yang paling populer dan efektif untuk menemukan rute terpendek dari satu titik ke titik lainnya dalam sebuah graf berbobot non-negatif [7, 8]. Berdasarkan tinjauan pustaka, beberapa peneliti telah mengaplikasikan algoritma ini dalam berbagai konteks yang relevan. Sebagai contoh, Saputra dan Dores (2021) [2] berhasil mengembangkan aplikasi pencarian bengkel terdekat berbasis Android, membuktikan efektivitas algoritma Dijkstra dalam menghemat waktu dan bahan bakar pengguna. Efektivitas ini juga diperkuat oleh penelitian Sudibyo *et al.* (2020) [3] yang mengimplementasikan algoritma serupa untuk pencarian rute terpendek objek wisata. Dalam konteks yang lebih spesifik, Seo (2024) [4] menerapkan algoritma Dijkstra pada aplikasi pencarian bengkel tambal ban di Kota Medan, menunjukkan relevansi penerapannya untuk wilayah geografis tertentu. Adaptabilitas algoritma ini juga terlihat dalam penelitian Sunardi *et al.* (2019) [5] yang menggunakannya untuk optimasi rute pengiriman barang. Sementara itu, penelitian dari Septiani *et al.* (2024) [6] memberikan landasan teknis terkait pengembangan sistem informasi berbasis web dengan PHP dan MySQL yang dapat diadaptasi. Penerapan algoritma ini pada data spasial juga ditunjukkan oleh Al Fatta *et al.* (2023) [9] untuk pencarian lokasi objek wisata dan oleh Ramadhan dan Rahadi (2023) [10] juga secara konsisten mengimplementasikan algoritma Dijkstra pada sistem informasi geografis untuk menentukan rute terpendek ke berbagai tujuan, seperti pusat perbelanjaan dan rute pengiriman, menunjukkan konsensus ilmiah yang kuat mengenai efektivitas algoritma ini.

Berdasarkan tinjauan pustaka di atas, terlihat bahwa algoritma Dijkstra telah banyak diterapkan untuk berbagai kasus pencarian rute terpendek. Namun, penelitian-penelitian sebelumnya belum secara spesifik berfokus pada kondisi geografis dan sebaran bengkel di Kabupaten Nganjuk. Sebagian besar penelitian yang ada umumnya berfokus pada kota-kota besar atau konteks yang berbeda. Oleh karena itu, terdapat kesenjangan (*research gap*) dalam literatur yang membahas implementasi algoritma Dijkstra secara spesifik untuk memetakan dan

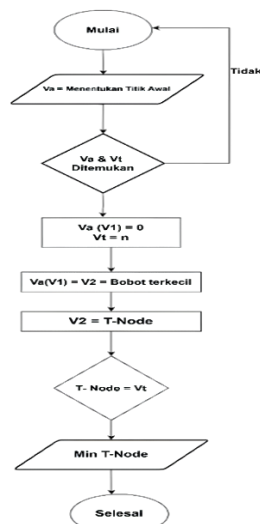
menentukan rute terpendek ke bengkel-bengkel di wilayah Kabupaten Nganjuk. Penelitian ini bertujuan untuk mengisi kesenjangan tersebut dengan menyediakan solusi yang relevan dan kontekstual bagi masyarakat Nganjuk.

II. METODE

Penelitian ini menggunakan pendekatan kuantitatif, dengan fokus utama pada pengukuran jarak dan perhitungan jalur terpendek antara titik lokasi pengguna dan bengkel terdekat. Pendekatan kuantitatif dipilih karena mampu menyediakan data yang objektif, terukur, dan dapat dianalisis secara statistik, sehingga hasil yang diperoleh lebih akurat dan dapat dipertanggungjawabkan. Melalui metode ini, penelitian memanfaatkan data numerik yang diperoleh dari peta digital, seperti koordinat geografis (*latitude dan longitude*), jarak antar titik, serta bobot atau nilai yang mewakili kondisi jalur tertentu. Data tersebut kemudian diolah menggunakan algoritma pencarian jalur terpendek, yang memungkinkan pemodelan dan perhitungan rute optimal secara sistematis dan terstruktur. Dengan demikian, pendekatan ini dianggap sesuai untuk mendukung pengambilan keputusan dalam penentuan rute tercepat menuju bengkel, sekaligus memberikan gambaran yang komprehensif mengenai efisiensi dan efektivitas rute yang dihasilkan.

A. Alur Algoritma Dijkstra

menunjukkan alur kerja algoritma Dijkstra dalam mencari rute terpendek. Proses dimulai dengan menetapkan titik awal (V_a) dan titik tujuan (V_t). Jika titik belum ditemukan, sistem akan mencarinya kembali. Titik awal diberi nilai 0, sedangkan titik lainnya, termasuk V_t , diberi nilai tak hingga. Algoritma kemudian memilih simpul dengan bobot terkecil, menandainya sebagai T-node, lalu memeriksa apakah simpul tersebut adalah tujuan. Jika ya, proses selesai. Jika belum, sistem memperbarui bobot simpul tetangga dan mengulangi proses hingga rute terpendek ditemukan.



Gambar 1. Alur Algoritma Dijkstra

B. Proses Algoritma Dijkstra

Algoritma Dijkstra mencari jalur terpendek dengan mengunjungi satu per satu node dan memperbarui bobot jarak secara bertahap. Dimulai dari node awal dengan bobot 0, algoritma mengevaluasi jarak ke node tetangga, memperbarui jarak jika lebih kecil, lalu menandai node yang sudah dilewati agar tidak dicek ulang. Proses ini diulang hingga mencapai node tujuan. Pengujian dilakukan dengan membandingkan jarak hasil algoritma dengan jarak pengukuran menggunakan persentase selisih untuk mengukur akurasi

$$S_j = \left[\frac{\sum_{i=1}^N (P_1 - P_2)}{N} \right] \times 100\%$$

C. Data dan Sumber

Sistem ini diimplementasikan menggunakan bahasa pemrograman Python dengan memanfaatkan pustaka seperti Pandas untuk pengolahan data, NumPy untuk perhitungan numerik, dan *NetworkX* untuk pemodelan graf dan perhitungan jalur terpendek menggunakan algoritma Dijkstra. Untuk mempermudah interaksi pengguna, antarmuka berbasis web dikembangkan menggunakan framework Flask, sehingga pengguna dapat memasukkan data lokasi dan melihat hasil perhitungan rute secara langsung melalui browser. Data lokasi bengkel dan jaringan jalan diperoleh melalui *Google Maps* API yang menyediakan informasi koordinat dan jarak dengan tingkat akurasi tinggi.

D. Evaluasi Sistem

Evaluasi sistem dilakukan dengan mengukur akurasi hasil perhitungan jalur terpendek yang dihasilkan oleh algoritma Dijkstra dibandingkan dengan data jarak sebenarnya di lapangan. Pengujian dilakukan dengan menggunakan beberapa rute uji dari titik awal ke bengkel tujuan, kemudian membandingkan jarak yang dihitung oleh sistem dengan jarak pengukuran langsung atau data referensi. Selain itu, sistem juga dievaluasi dari segi waktu respon dan kemudahan penggunaan antarmuka. Hasil evaluasi menunjukkan bahwa sistem mampu memberikan rute terpendek dengan tingkat kesesuaian yang tinggi serta memberikan kemudahan akses informasi bagi pengguna.

III. HASIL DAN PEMBAHASAN

Implementasi sistem ini dilakukan dengan mengembangkan antarmuka berbasis web yang dirancang agar ramah pengguna (user friendly), sehingga memungkinkan pengguna untuk dengan mudah memasukkan titik awal keberangkatan dan menentukan tujuan akhir berupa lokasi bengkel yang diinginkan. Melalui antarmuka ini, pengguna dapat secara real-time memperoleh rekomendasi rute tercepat yang dihitung menggunakan algoritma Dijkstra. Sistem secara otomatis memproses input koordinat geografis dan memvisualisasikan jalur terpendek pada peta digital, dilengkapi informasi jarak tempuh total, estimasi durasi perjalanan, serta daftar urutan simpul yang dilalui. Selain itu, sistem ini juga menyediakan fitur pemilihan rute alternatif apabila terdapat beberapa jalur yang memiliki jarak hampir sama. Hasil perhitungan diverifikasi menggunakan

data jarak aktual yang diperoleh dari Google Maps API dan dikonfirmasi melalui survei lapangan guna memastikan akurasi rute yang dihasilkan. Dengan adanya implementasi ini, pengguna dapat menghemat waktu perjalanan dan meminimalkan kebingungan saat mencari bengkel terdekat. Namun demikian, perlu dicatat bahwa sistem ini pada tahap pengembangan awal belum mengakomodasi kondisi lalu lintas dinamis secara otomatis, sehingga faktor-faktor seperti kemacetan atau penutupan jalan yang dapat mempengaruhi waktu tempuh belum sepenuhnya diperhitungkan dalam perhitungan jalur optimal.

A. Permintaan Izin Lokasi

Gambar ini menunjukkan tampilan awal halaman fitur Pencarian Rute pada situs web dijkstra.kingperseus.online yang dibuka melalui browser Safari di perangkat iPhone. Di bagian tengah layar muncul pop-up permintaan izin lokasi dari browser dengan pesan “dijkstra.kingperseus.online” ingin menggunakan lokasi Anda saat ini. Situs web ini akan menggunakan lokasi tepat Anda karena saat ini “Chrome” memiliki akses ke lokasi tepat Anda. Permintaan ini muncul karena sistem memerlukan akses lokasi pengguna untuk menentukan titik awal pencarian rute secara akurat. Di latar belakang terlihat peta wilayah Kabupaten Nganjuk, Jawa Timur, dengan beberapa penanda lokasi berupa ikon pin biru yang kemungkinan mewakili titik-titik penting seperti bengkel atau tempat tujuan lainnya. Pengguna disarankan untuk mengklik tombol “Izinkan” agar sistem dapat mendeteksi lokasi secara otomatis dan menghitung rute terbaik dari posisi pengguna menuju lokasi tujuan terdekat menggunakan algoritma Dijkstra.

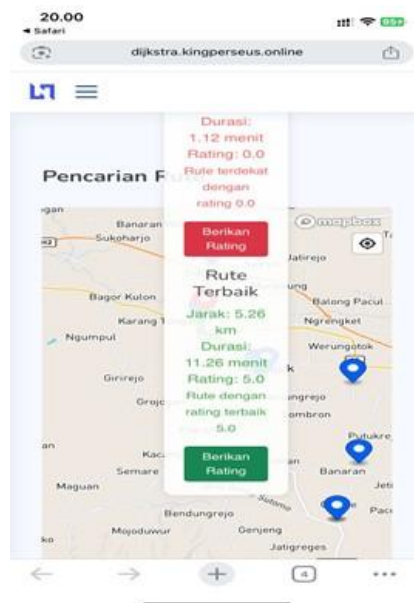


Gambar 2. Permintaan Izin Lokasi

B. Hasil Pencarian Rute

Gambar ini menampilkan hasil pencarian rute setelah pengguna memberikan izin akses lokasi. Dalam tampilan tersebut, sistem secara otomatis menghitung dan menyajikan dua alternatif rute berdasarkan perhitungan algoritma, yaitu *Rute Terdekat* dan *Rute Terbaik*. Rute Terdekat ditampilkan dalam warna merah dengan durasi tempuh hanya 1,12 menit dan memiliki rating 0.0, yang berarti rute ini belum pernah dinilai oleh pengguna sebelumnya. Rute ini merupakan opsi tercepat dari lokasi pengguna saat ini menuju tujuan terdekat, namun belum terverifikasi kualitasnya oleh pengguna lain. Sementara itu, Rute Terbaik ditampilkan dalam warna hijau dengan jarak tempuh sejauh 5,26 km dan durasi 11,26 menit. Rute ini memiliki rating sempurna, yaitu 5.0, yang menunjukkan bahwa rute tersebut sangat direkomendasikan berdasarkan penilaian dari pengguna sebelumnya, meskipun memerlukan waktu tempuh yang lebih lama dibandingkan rute terdekat.

Kedua rute tersebut dilengkapi dengan tombol "Berikan Rating" yang memungkinkan pengguna memberikan penilaian terhadap kualitas rute setelah mencobanya. Peta pada tampilan ini menggunakan layanan Mapbox dan menunjukkan posisi awal pengguna serta beberapa titik lokasi tujuan yang tersebar di wilayah Nganjuk dan sekitarnya. Dengan tersedianya dua opsi rute tersebut, pengguna memiliki kebebasan untuk memilih antara rute tercepat atau rute dengan rating tertinggi, sesuai dengan preferensi dan kebutuhan perjalanan masing-masing.



Gambar 3. Hasil Pencarian Rute

IV. KESIMPULAN

Isi kesimpulan merupakan jawaban dari tujuan penelitian bukan rangkuman hasil penelitian. Kesimpulan dan saran dibuat secara singkat, jelas, dan padat didasarkan pada hasil penelitian. Kesimpulan dan Saran berupa paragraf tanpa numbering. Kesimpulannya harus dikaitkan dengan judul dan tujuan penelitian. Jangan membuat pernyataan yang tidak didukung secara memadai oleh temuan Anda. Tuliskan peningkatan yang dilakukan pada bidang teknik industri atau sains secara umum. Jangan membuat diskusi lebih lanjut, ulangi abstrak, atau hanya daftar hasil dari hasil penelitian. Jangan gunakan poin berpoin, gunakan kalimat paragraf sebagai gantinya.

DAFTAR PUSTAKA

- [1] Aprilianingsih E. P., Primananda, R., & Suharsono, A. (2017). Analisis fail path pada arsitektur software defined network menggunakan Dijkstra algorithm. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK)* Universitas Brawijaya, 1(3), 174–183.
- [2] Moh Hilal Fauzi, “Perancangan Strategi Pemeliharaan Dengan Pendekatan Metode Cmms Dan Ommp Dengan Fmeca Sebagai Perspektif: (Studi Kasus: Instrument Air Compressor Pt. Xyz),” *Noe*, vol. 8, no. 01, hlm. 1–11, Apr. 2025, doi: 10.29407/noe.v8i01.24674.
- [3] A. Z. Abdullah dan E. B. Setiawan, “Sentiment Analysis Accuracy for 2024 Indonesian Election Tweets Using CNN-LSTM With Genetic Algorithm Optimization,” *INTENSIF: J. Ilm. Penelit. dan Penerap. Tek. Sist. Inf.*, vol. 9, no. 1, hlm. 1–14, Feb. 2025, doi: 10.29407/intensif.v9i1.22999.
- [4] J. P. Randalongi, J. D. I. Manongko, dan Z. Mansjur, “Pengaruh Variasi Fraksi Volume dan Lama Perendaman NaOH Terhadap Kekuatan Tarik dan Struktur Mikro Komposit Serat Eceng Gondok (*Eichornia Crassipes*),” *JMN*, vol. 7, no. 2, hlm. 166–176, Dec. 2024, doi: 10.29407/jmn.v7i2.22240.
- [5] Y. Saputra dan A. Does, “Pengembangan aplikasi pencarian bengkel terdekat menggunakan algoritma Dijkstra berbasis Android,” *Jurnal Sistem Informasi, Teknologi Informasi dan Komputer (JUST IT)*, vol. 11, no. 3, hlm. 15–20, 2021.
- [6] N. A. Sudibyo, P. E. K. A. Setyawan, Y. P. Surya, dan R. Hidayat, “Implementasi Algoritma Dijkstra Dalam Pencarian Implementation Dijkstra Algorithm in Searching the Shortest Path Tourism Route,” *Riemann Research of Mathematics and Mathematics Education*, vol. 2, no. 1, hlm. 1–9, 2020.
- [7] A. J. Seo, “IMPLEMENTASI ALGORITMA DIJKSTRA PADA APLIKASI PENCARIAN BENGKEL TAMBAL BAN TERDEKAT DI KOTA MEDAN BERBASIS ANDROID,” 2024.
- [8] Sunardi, A. Yudhana, dan A. A. Kadim, “Implementasi... TERPENDEK JASA KIRIMAN BARANG BERBASIS MOBILE DENGAN METODE ALGORITMA DIJKSTRA,” 2019, hlm. 15–19.
- [9] D. R. N. Septiani, Hariyanto, dan B. T. Nugroho, “Sistem informasi administrasi cluster BPK RI Residence (SI MISTER PRESIDEN) pada Perumahan Samesta Parayasa berbasis web dengan menggunakan PHP dan MySQL,” *Jurnal SIKOMTEK*, vol. 14, no. 1, hlm. 73–81, 2024.
- [10] R. S. Pressman, *Rekayasa Perangkat Lunak: Pendekatan Praktisi (Edisi 7, Buku 1)*. Yogyakarta: Andi, 2010.