



Implementasi Algoritma Dijkstra dan Bellman-Ford untuk Optimasi Rute Pemadam Kebakaran di Kota Praya

Sunardi ^{1*}, Muhamad Azwar ², Dedy Sofian MZ ¹, Angga Radlisa Samsudin ², dan Fazlul Rahman ¹

¹ Sistem Informasi, Universitas Nahdlatul Ulama Nusa Tenggara Barat, Indonesia

² Ilmu Komputer, Universitas Bumigora, Indonesia

* Korespondensi: soenardhi.75@gmail.com

Sitasi: Sunardi, S.; Azwar, M.; Sofian MZ, D.; Samsudin, A. R.; dan Rahman, F. (2025). Implementasi Algoritma Dijkstra dan Bellman-Ford untuk Optimasi Rute Pemadam Kebakaran di Kota Praya. JTIM: Jurnal Teknologi Informasi Dan Multimedia, 7(2), 398-407. <https://doi.org/10.35746/jtim.v7i2.744>

Diterima: 15-05-2025

Direvisi: 26-05-2025

Disetujui: 28-05-2025



Copyright: © 2025 oleh para penulis. Karya ini dilisensikan di bawah Creative Commons Attribution-ShareAlike 4.0 International License. (<https://creativecommons.org/licenses/by-sa/4.0/>).

Abstract: Forest and land fires are critical emergencies requiring rapid response to minimize casualties and property damage. In urban areas like Praya City, fire department response delays are often caused by inefficient routing, especially with traffic congestion and complex road infrastructure. This study aims to analyze and compare the performance of Dijkstra's and Bellman-Ford's algorithms for optimizing firefighter routes in Praya City. This quantitative research utilized a computational and comparative analysis approach. Road network data from Praya City was obtained from Google Maps and modeled as a static graph consisting of 17 nodes and weighted edges representing actual distances. Dijkstra's and Bellman-Ford's algorithms were implemented in Python to find the shortest routes from a designated starting point (Fire Department office) to all other nodes. Performance was evaluated based on route optimality, completeness, and computation time. Both Dijkstra's and Bellman-Ford's algorithms successfully identified identical optimal shortest routes for all tested origin-destination pairs within the Praya City graph. However, Dijkstra's algorithm demonstrated significantly superior computational efficiency, with an average computation time of 0.5 seconds, compared to Bellman-Ford's 1.5 seconds. For optimizing firefighter routes on the static road network graph of Praya City, Dijkstra's algorithm is recommended due to its combination of optimality and superior speed. This finding provides an empirical basis for developing more efficient emergency response navigation systems. Future research should focus on integrating dynamic parameters like real-time traffic data.

Keywords: Dijkstra's Algorithm; Bellman-Ford Algorithm; Shortest Path; Firefighter Route Optimization; Praya City

Abstrak: Kebakaran hutan dan lahan merupakan keadaan darurat kritis yang memerlukan respons cepat untuk meminimalkan korban jiwa dan kerusakan properti. Di wilayah urban seperti Kota Praya, keterlambatan tim pemadam kebakaran sering disebabkan oleh rute yang tidak efisien, terutama saat menghadapi kepadatan lalu lintas dan infrastruktur jalan yang kompleks. Penelitian ini bertujuan untuk menganalisis dan membandingkan performa algoritma Dijkstra dan Bellman-Ford untuk optimasi rute pemadam kebakaran di Kota Praya. Penelitian kuantitatif ini menggunakan pendekatan komputasi dan analisis komparatif. Data jaringan jalan Kota Praya diperoleh dari Google Maps dan dimodelkan sebagai graf statis yang terdiri dari 17 simpul dan sisi berbobot yang merepresentasikan jarak aktual. Algoritma Dijkstra dan Bellman-Ford diimplementasikan menggunakan Python untuk mencari rute terpendek dari titik awal yang ditentukan (kantor Pemadam Kebakaran) ke semua simpul lainnya. Performa dievaluasi berdasarkan optimalitas rute, kelengkapan, dan waktu komputasi. Algoritma Dijkstra dan Bellman-Ford berhasil mengidentifikasi rute terpendek optimal yang identik untuk semua pasangan asal-tujuan yang diuji pada graf jaringan jalan Kota Praya. Namun, algoritma Dijkstra menunjukkan efisiensi komputasi yang secara

signifikan lebih unggul, dengan rata-rata waktu komputasi 0.5 detik, dibandingkan dengan Bellman-Ford yaitu 1.5 detik. Untuk optimasi rute pemadam kebakaran pada graf jaringan jalan statis Kota Praya, algoritma Dijkstra direkomendasikan karena kombinasi optimalitas dan kecepatan superiornya. Temuan ini memberikan dasar empiris untuk pengembangan sistem navigasi respons darurat yang lebih efisien. Penelitian masa depan sebaiknya berfokus pada integrasi parameter dinamis seperti data lalu lintas real-time.

Kata kunci: Algoritma Dijkstra; Algoritma Bellman-Ford; Rute Terpendek; Optimasi Rute Pemadam Kebakaran; Kota Praya

1. Pendahuluan

Kebakaran merupakan keadaan darurat kritis yang memerlukan respons cepat untuk meminimalkan risiko korban jiwa dan kerusakan properti [1]. Di wilayah urban seperti Kota Praya, keterlambatan tim pemadam kebakaran sering disebabkan oleh ketidakefisienan rute, terutama saat menghadapi kepadatan lalu lintas atau infrastruktur jalan yang kompleks [2]. Data terbaru menunjukkan bahwa 67% kasus kebakaran di Praya mengalami keterlambatan respon lebih dari 8 menit, yang secara signifikan meningkatkan luas area kebakaran dan potensi kerugian [3]. Solusi konvensional berbasis peta statis dan pengalaman petugas dinilai tidak lagi memadai untuk menangani dinamika perkotaan yang terus berkembang [4].

Pendekatan berbasis algoritma graf telah terbukti efektif dalam optimasi rute darurat di berbagai konteks [5]. Algoritma Dijkstra, sebagai salah satu metode pencarian rute terpendek yang paling umum digunakan, telah berhasil diimplementasikan dalam sistem navigasi ambulans dengan reduksi waktu respon hingga 40% [6]. Namun demikian, implementasi spesifik untuk pemadam kebakaran di kota menengah seperti Praya masih menghadapi beberapa tantangan unik, termasuk keterbatasan data lalu lintas real-time dan karakteristik jaringan jalan yang spesifik [7]. Selain itu, kompleksitas komputasi menjadi pertimbangan penting dalam pemilihan algoritma untuk sistem yang memerlukan respons cepat [8].

Berdasarkan tinjauan literatur, terdapat beberapa kesenjangan penelitian yang perlu diatasi. Mayoritas studi sebelumnya berfokus pada wilayah metropolitan dengan infrastruktur data yang lengkap [9], sementara kebutuhan kota menengah seperti Praya masih kurang mendapatkan perhatian. Penelitian oleh [10] dan [11] menunjukkan bahwa pertimbangan parameter dinamis seperti kepadatan lalu lintas sering diabaikan dalam optimasi rute pemadam kebakaran. Selain itu, belum ada konsensus mengenai algoritma yang paling optimal untuk kondisi dengan data terbatas dan kompleksitas graf sedang [12].

Penelitian ini bertujuan untuk mengatasi kesenjangan tersebut melalui tiga kontribusi utama. Pertama, mengembangkan model simulasi berbasis graf jalan Praya yang mengintegrasikan parameter statis (jarak) dan dinamis (estimasi kepadatan lalu lintas)[13]. Kedua, melakukan analisis komparatif menyeluruh terhadap performa algoritma Dijkstra dan Bellman-Ford dalam berbagai skenario kebakaran[14]. Ketiga, merancang framework seleksi algoritma yang adaptif berdasarkan karakteristik insiden dan ketersediaan data[15]. Kesimpulan utama yang diharapkan dari penelitian ini adalah identifikasi algoritma (Dijkstra atau Bellman-Ford) atau kombinasi kondisi penerapan yang paling efektif dan efisien untuk optimasi rute pemadam kebakaran di Kota Praya, yang mampu mencapai target akurasi RMSE <1.5 menit dengan waktu komputasi kurang dari 2 detik per kalkulasi rute.

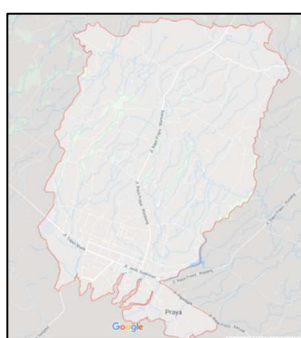
Hasil penelitian ini diharapkan dapat memberikan dasar empiris yang kuat untuk pengembangan sistem navigasi dinamis yang mampu meningkatkan efektivitas layanan pemadam kebakaran. Selain kontribusi akademis berupa publikasi dataset graf jalan Praya, penelitian ini juga menghasilkan protokol implementasi praktis yang dapat diadopsi oleh dinas terkait. Temuan penelitian memiliki potensi aplikasi yang luas tidak hanya untuk Kota Praya, tetapi juga wilayah-wilayah lain dengan karakteristik serupa [16].

2. Bahan dan Metode

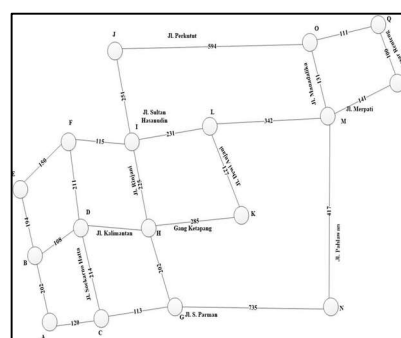
Penelitian ini menerapkan pendekatan kuantitatif dengan metode komputasi dan analisis komparatif untuk mengevaluasi performa algoritma Dijkstra dan Bellman-Ford dalam optimasi rute pada jaringan jalan di Kecamatan Praya. Tahapan penelitian meliputi pengumpulan data spasial, pemodelan graf jaringan jalan, implementasi dan eksekusi algoritma, serta analisis perbandingan kinerja algoritma.

2.1. Pengumpulan dan Persiapkan Data

Gambar 1. dan Gambar 2. Data jaringan jalan Kecamatan Praya diperoleh dari peta digital layanan Google, yang menjadi dasar identifikasi sebelas (11) ruas jalan utama beserta persimpangannya. Dari peta ini, sebanyak 17 simpul (*nodes*/titik) kunci—merepresentasikan persimpangan jalan utama dan titik strategis lainnya, seperti Titik A (Kantor BPBD) sebagai titik awal (V1) dan Titik Q sebagai salah satu titik tujuan utama (V17)—diidentifikasi secara visual. Jaringan jalan ini kemudian direpresentasikan sebagai graf $G = (V, E)$, di mana V adalah himpunan ke-17 simpul dan E adalah himpunan sisi (*edges*) yang merepresentasikan ruas jalan penghubung, dengan setiap sisi memiliki atribut bobot (*weight*) berupa jarak aktual (diasumsikan dalam meter) yang ditentukan melalui pengukuran pada Google Maps atau data sekunder terverifikasi. Hasil dari pemodelan ini adalah sebuah dataset graf statis, yang mencakup daftar simpul, daftar sisi, beserta atribut bobot (jarak) untuk setiap sisi, dan dataset inilah yang menjadi input utama untuk algoritma Dijkstra dan Bellman-Ford.



Gambar 1. Peta Google Maps Kecamatan Praya



Gambar 2. Graf Rute Jalan Peta Kecamatan Praya

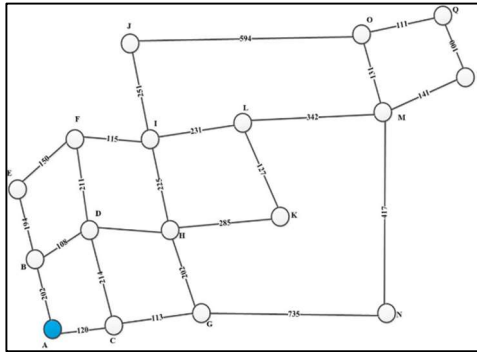
Keterangan:

Rute A – C – G – N melewati Jl. S. Parman, sedangkan rute K – L melalui Jl. Dewi Anjani. N – M terhubung oleh Jl. Pahlawan, sementara H – K dapat dilalui melalui Gg. Ketapang. Untuk menuju dari M – O, digunakan Jl. Mandalika, dan B – D – H dilayani oleh Jl. Kalimantan. M – P dihubungkan oleh Jl. Merpati, dan C – D – F dapat ditempuh melalui Jl. Sukarno Hatta. Rute P – Q melewati Jl. Pasar Renteng, sedangkan E – F – I – L – M dilayani oleh Jl. Sultan Hasanudin. Terakhir, J – O terhubung melalui Jl. Perkutut.

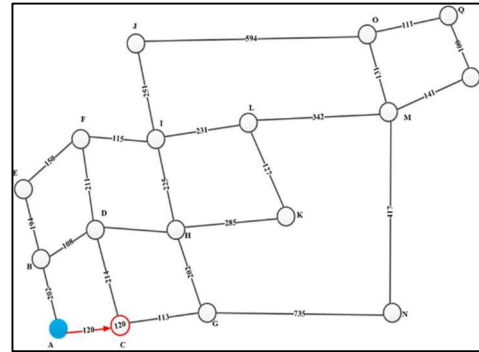
2.2. Implementasi dan Eksekusi Algoritma

2.2.1. Algoritma Dijkstra

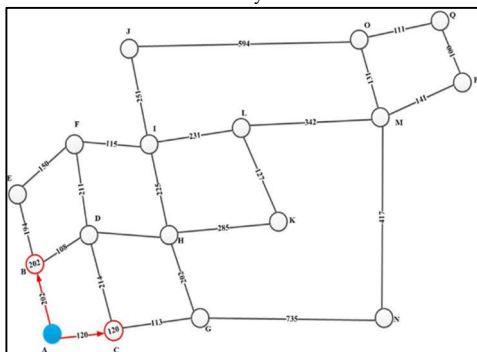
Gambar 3. sampai Gambar 19. menggambarkan Algoritma Dijkstra diimplementasikan untuk menemukan rute terpendek dari titik awal (Titik A) ke semua titik lain dalam graf, termasuk titik tujuan spesifik (Titik Q). Proses pencarian jalur dilakukan mengikuti prinsip iteratif pemilihan node dengan jarak terakumulasi terkecil.



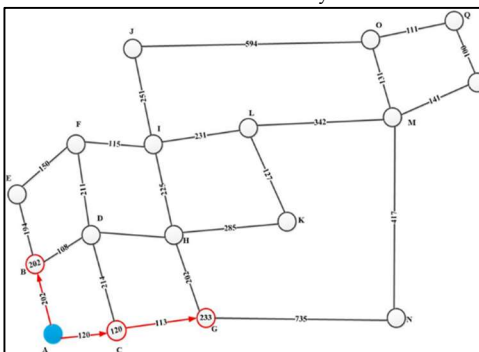
Gambar 3. Inisialisasi Node Awal dan Node Berikutnya



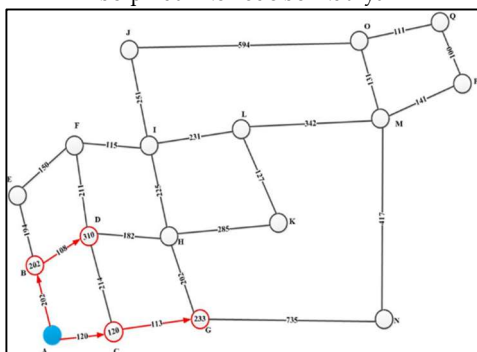
Gambar 4. Menentukan node terkecil untuk berpindah ke node berikutnya



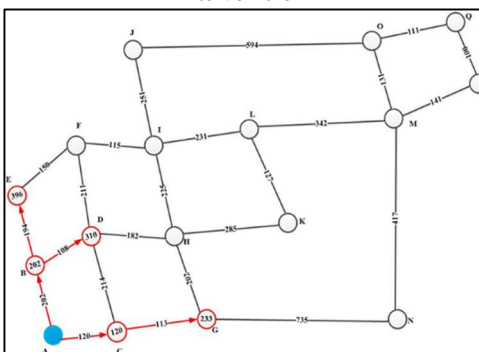
Gambar 5. Menentukan node selanjutnya untuk bisa berpindah ke node berikutnya



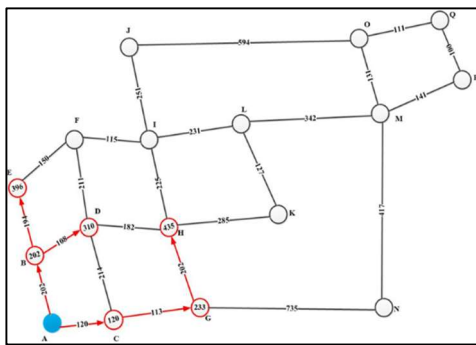
Gambar 6. Menentukan node dengan bobot terkecil untuk di lalui



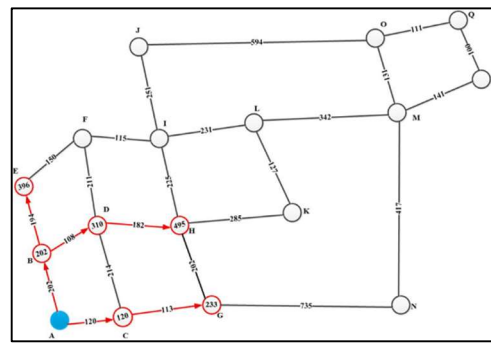
Gambar 7. Menentukan Node Ketiga untuk pindah ke node berikutnya



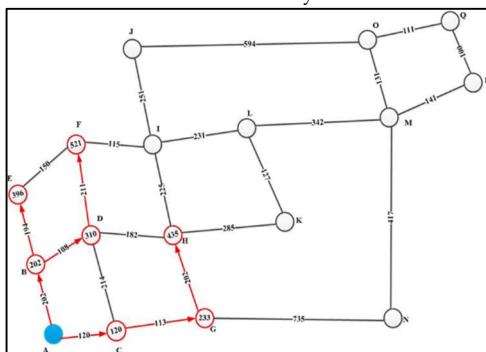
Gambar 8. Menentukan Node Keempat untuk berpindah ke node beerikutnya



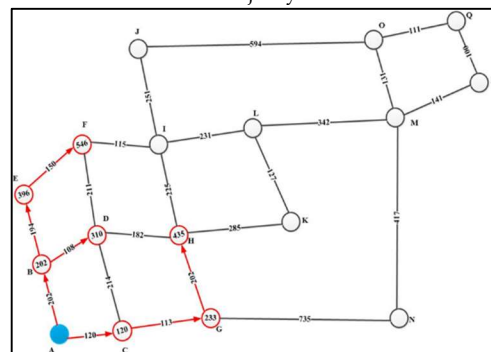
Gambar 9. Menentukan Node Kelima untuk menuju node berikutnya



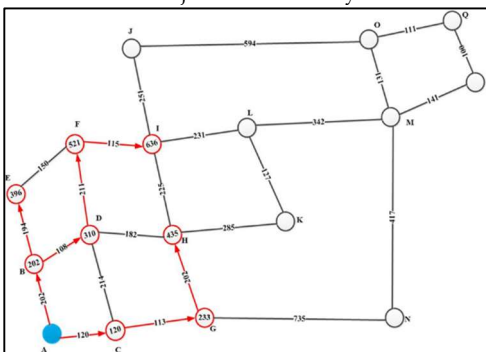
Gambar 10. Mencari bobot terkecil untuk menuju node selanjutnya



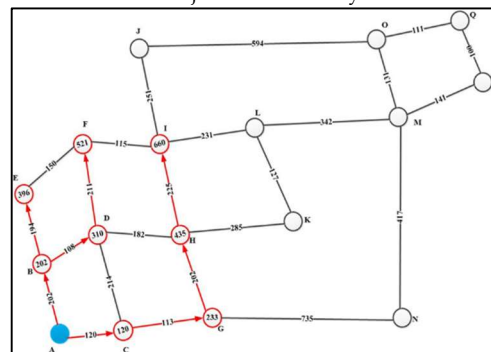
Gambar 11. Menentukan Node Keenam untuk menuju Node Berikutnya



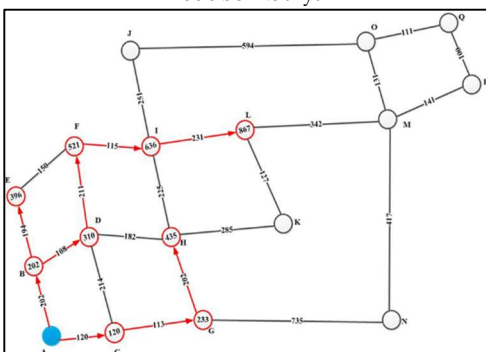
Gambar 12. Tentukan node dengan bobot terkecil untuk menuju node berikutnya



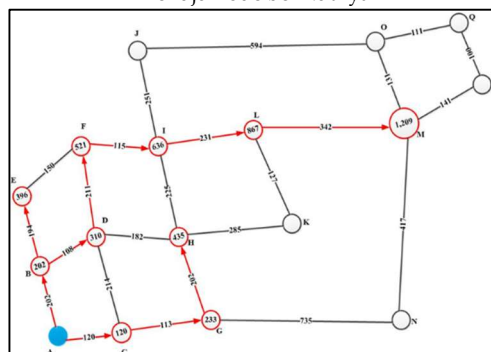
Gambar 13. Menentukan node ketujuh untuk menuju node berikutnya



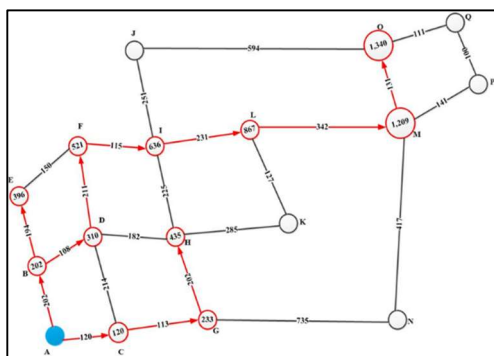
Gambar 14. Tentukan node dengan bobot terkecil untuk menuju node berikutnya



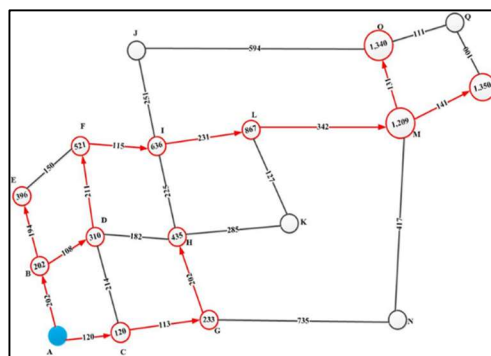
Gambar 15. Menentukan node dengan bobot paling kecil untuk menuju node berikutnya.



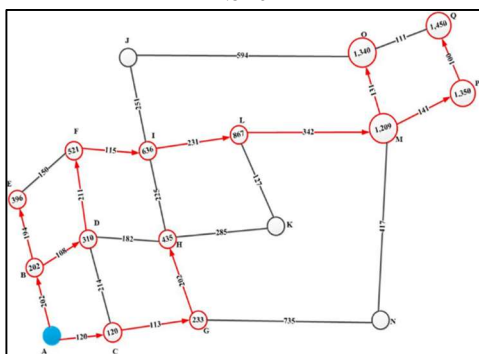
Gambar 16. Menentukan node M sebagai node dengan bobot terkecil menuju node tujuan



Gambar 17. Menentukan node O sebagai node yang ke-10



Gambar 18. Menjadikan Node P sebagai node ke sebelas menuju node tujuan



Gambar 19. Menjadikan node Q sebagai node tujuan

2.2.2. Algoritma Bellman-Ford

Algoritma Bellman-Ford juga diimplementasikan untuk tujuan yang sama, yaitu menemukan rute terpendek dari titik awal ke semua titik lain pada graf yang sama untuk keperluan perbandingan.

2.2.3. Lingkungan dan Proses Komputasi

Implementasi algoritma dan proses perhitungan dilakukan menggunakan bahasa pemrograman Python. Untuk setiap algoritma, perhitungan dilakukan untuk menentukan rute terpendek dari Titik A ke berbagai titik tujuan, dengan hasil jarak kumulatif dicatat. Waktu komputasi yang dibutuhkan oleh masing-masing algoritma untuk menyelesaikan setiap proses pencarian rute sesuai dengan baris pada Tabel 1 diukur menggunakan fungsi `time.time()` pada Python, yang mencatat selisih waktu sebelum dan sesudah eksekusi fungsi algoritma utama. Hasil pengukuran waktu komputasi ini disajikan pada Tabel 2.

Tabel 1. Hasil Perhitungan Algoritma Dijkstra dan Bellman-Ford

Vertex asal	Tujuan	Jarak Dengan Vertex Tujuan		
A	A → B	0	202	202
A	A → C	0	120	120
B	B → D	202	108	310
...
M	M → P	1.209	141	1.350
O	O → Q	1.340	111	1.451
P	P → Q	1.350	100	1.450

Tabel 2. Perbandingan Waktu yang dibutuhkan dari kedua algoritma

No	Dijkstra (s)	Bellman-Ford (s)
1	0,5	1,6
2	0,5	1,5
3	0,5	1,4

No	Dijkstra (s)	Bellman-Ford (s)

22	0.5	1,5
23	0.5	1,4
24	0.5	1,4
Jumlah	12	35.5
Rata-Rata	0.5	1.5

2.3. Evaluasi Performa dan Analisis Komparatif

Performa kedua algoritma (Dijkstra dan Bellman-Ford) dievaluasi dan dibandingkan berdasarkan lima kriteria utama. Pertama, kelengkapan (completeness), yaitu kemampuan algoritma untuk menemukan solusi rute jika ada, dari titik awal ke semua simpul yang terhubung dalam graf. Kedua, optimalitas (optimality), yang menganalisis apakah rute yang dihasilkan benar-benar memiliki total bobot (jarak) minimum. Ketiga, kompleksitas waktu (time complexity), mencakup tinjauan teoritis (Big-O) dan perbandingan waktu eksekusi aktual (dalam detik) untuk setiap kasus pencarian rute, dengan detail asal, tujuan, dan jarak (Tabel 1) serta data waktu komputasi (Tabel 2). Keempat, kompleksitas ruang (space complexity), yaitu analisis kebutuhan memori kedua algoritma. Kelima, waktu komputasi dan hasil rute, termasuk total jarak rute terpendek (Tabel 1) dan rata-rata waktu komputasi (Tabel 2).

2.4. Ketersediaan Bahan, dan Data

Penelitian ini mengadopsi prinsip terbuka dengan menyediakan dataset graf jalan Kecamatan Praya (berisi simpul, sisi, dan bobot) serta langkah-langkah pemodelan berbasis Google Maps secara detail agar dapat direplikasi.

2.5. Pertimbangan Etis

Karena hanya menggunakan data peta publik (Google Maps) tanpa melibatkan subjek manusia atau hewan, penelitian ini tidak memerlukan persetujuan etik khusus dari komite etik.

3. Hasil

Bagian ini menyajikan hasil eksperimen yang diperoleh dari implementasi dan pengujian algoritma Dijkstra dan Bellman-Ford pada model graf jaringan jalan Kecamatan Praya.

3.1. Pemodelan Graf Jaringan Jalan dan Representasi Data

Pemodelan jaringan jalan Kecamatan Praya menghasilkan sebuah graf statis yang terdiri dari 17 simpul (*nodes*) yang merepresentasikan persimpangan atau titik penting, dan sejumlah sisi (*edges*) yang merepresentasikan ruas jalan penghubung. Titik A (Kantor BPBD) ditetapkan sebagai simpul awal (V1) untuk semua skenario pencarian rute. Setiap sisi memiliki bobot yang merepresentasikan jarak aktual antar simpul, yang diperoleh melalui analisis peta Google Maps (sebagaimana diilustrasikan pada Gambar 1 dan detail graf pada Gambar 2 hingga 4 dataset graf ini menjadi input standar untuk kedua algoritma yang diuji.

3.2. Hasil Perhitungan Rute Terpendek

Implementasi kedua algoritma pada graf jaringan jalan Kecamatan Praya menghasilkan data jarak terpendek dari simpul awal (Titik A) ke semua simpul lainnya. Tabel 1 (Hasil Perhitungan Algoritma Dijkstra dan Bellman-Ford) merangkum hasil perhitungan jarak ini untuk Algoritma Dijkstra dan Bellman-Ford. Sebagai contoh spesifik, penentuan rute terpendek dari Titik A ke Titik Q menggunakan Algoritma Dijkstra menghasilkan rute $A \rightarrow C \rightarrow D \rightarrow F \rightarrow I \rightarrow L \rightarrow M \rightarrow P \rightarrow Q$ dengan total jarak 1.450 satuan (diasumsikan meter).

Observasi utama dari Tabel 1 adalah bahwa kedua algoritma, Dijkstra dan Bellman-Ford, menghasilkan nilai jarak terpendek yang identik untuk setiap pasangan simpul awal dan tujuan yang diuji pada graf jaringan jalan Kota Praya. Hal ini mengindikasikan bahwa untuk graf yang diuji, kedua algoritma mencapai solusi optimal yang sama.

3.3. Hasil Perbandingan Kinerja Algoritma

Kinerja kedua algoritma dibandingkan berdasarkan beberapa kriteria yang relevan, yaitu waktu komputasi, kelengkapan, dan optimalitas.

3.4. Waktu Komputasi

Waktu komputasi yang dibutuhkan oleh masing-masing algoritma untuk menyelesaikan seluruh proses pencarian rute dari titik awal ke semua titik lainnya Rata-rata waktu komputasi untuk Algoritma Dijkstra adalah 0.5 detik, sedangkan untuk Algoritma Bellman-Ford adalah 1.5 detik. Hasil ini secara jelas menunjukkan bahwa Algoritma Dijkstra secara signifikan lebih cepat dalam menyelesaikan perhitungan rute terpendek dibandingkan dengan Algoritma Bellman-Ford pada dataset dan lingkungan komputasi yang digunakan dalam penelitian ini.

3.5. Kelengkapan dan Optimalitas

Kedua algoritma berhasil menemukan rute terpendek untuk semua simpul tujuan yang dapat dijangkau dari simpul awal, menunjukkan kelengkapan (completeness) dari kedua metode. Lebih lanjut, kesamaan hasil jarak terpendek yang dihasilkan oleh kedua algoritma mengonfirmasi optimalitas (optimality) solusi yang ditemukan pada graf yang diuji, yang tidak memiliki bobot sisi negatif.

4. Pembahasan

Pada bagian ini, hasil penelitian yang telah dipaparkan akan dibahas dan diinterpretasikan lebih lanjut, dikaitkan dengan penelitian sebelumnya, hipotesis kerja (jika ada yang eksplisit), serta implikasi temuan untuk konteks optimasi rute pemadam kebakaran di Kota Praya.

4.1. Interpretasi Hasil dan Kesesuaian dengan Teori Algoritma

Temuan bahwa Algoritma Dijkstra dan Bellman-Ford menghasilkan jarak terpendek yang identik (Tabel 1) konsisten dengan teori algoritma. Algoritma Dijkstra dirancang untuk graf dengan bobot sisi non-negatif, kondisi yang terpenuhi oleh graf jarak fisik jaringan jalan. Algoritma Bellman-Ford, meskipun mampu menangani bobot sisi negatif (yang tidak relevan dalam konteks ini), juga akan menghasilkan rute terpendek yang benar pada graf tanpa siklus negatif. Karena graf jaringan jalan Kota Praya hanya menggunakan jarak fisik sebagai bobot, yang selalu positif, kedua algoritma diharapkan memberikan hasil optimal yang sama. Hal ini sejalan dengan penelitian [5], [6], dan [7] yang juga menunjukkan keberhasilan Dijkstra dalam menemukan rute terpendek pada jaringan jalan.

Perbedaan signifikan dalam waktu komputasi (Tabel 2), di mana Dijkstra (rata-rata 0.5 detik) jauh lebih unggul daripada Bellman-Ford (rata-rata 1.5 detik), juga sesuai dengan kompleksitas waktu teoritis masing-masing algoritma. Algoritma Dijkstra dengan implementasi menggunakan antrian prioritas (seperti binary heap) umumnya memiliki kompleksitas $O((E + V) \log V)$ atau $O(E \log V)$, sedangkan Bellman-Ford memiliki kompleksitas $O(V \cdot E)$. Untuk graf yang relatif kecil seperti yang digunakan (17 simpul), perbedaan absolut mungkin tidak drastis, namun persentase perbedaannya tetap signifikan dan akan semakin terasa pada graf yang lebih besar. Hasil ini mendukung pentingnya pemilihan algoritma berdasarkan efisiensi komputasi, terutama untuk aplikasi yang memerlukan respons cepat seperti layanan darurat [8].

4.2. Implikasi untuk Optimasi Rute Pemadam Kebakaran di Kota Praya

Berdasarkan hasil penelitian ini, untuk kondisi graf statis jaringan jalan Kota Praya yang hanya mempertimbangkan jarak fisik, Algoritma Dijkstra merupakan pilihan yang lebih superior dibandingkan Bellman-Ford karena menawarkan efisiensi waktu komputasi yang jauh lebih baik dengan tetap menjamin optimalitas rute. Dalam skenario darurat kebakaran, di mana setiap detik berharga [1, 3], kemampuan untuk menghasilkan rute optimal dalam waktu sesingkat mungkin adalah krusial. Penggunaan Dijkstra dapat berkontribusi pada pengurangan waktu perencanaan rute, yang secara tidak langsung dapat mempercepat waktu respons tim pemadam kebakaran.

Namun, penting untuk dicatat bahwa penelitian ini memiliki beberapa keterbatasan. Pertama, model graf yang digunakan bersifat statis dan belum mengintegrasikan parameter dinamis seperti kepadatan lalu lintas real-time atau kondisi jalan temporer (misalnya, penutupan jalan). Pendahuluan awal penelitian ini menyoroti pentingnya parameter dinamis [10, 11], namun implementasi saat ini belum mencapainya. Dalam kondisi lalu lintas yang dinamis, bobot sisi graf akan berubah, dan algoritma yang lebih adaptif atau modifikasi pada bobot graf mungkin diperlukan. Kedua, penelitian ini terbatas pada jaringan jalan yang dimodelkan untuk Kecamatan Praya dan belum tentu dapat digeneralisasi ke kota lain dengan karakteristik jaringan yang sangat berbeda tanpa penyesuaian. Ketiga, akurasi model graf sangat bergantung pada kualitas data peta awal (Google Maps) dan proses digitalisasinya.

4.3. Arah Penelitian Masa Depan

Berdasarkan temuan dan keterbatasan penelitian ini, beberapa arah penelitian masa depan dapat diidentifikasi: (1) Integrasi Parameter Dinamis: Mengembangkan model yang mampu mengintegrasikan data lalu lintas real-time atau estimasi kepadatan lalu lintas berdasarkan waktu untuk menyesuaikan bobot sisi graf secara dinamis. Ini akan meningkatkan relevansi rute yang dihasilkan dengan kondisi aktual di lapangan. Algoritma seperti A* atau modifikasi Dijkstra mungkin lebih sesuai. (2) Pengembangan Sistem Interaktif: Merancang dan mengimplementasikan prototipe sistem informasi geografis (SIG) interaktif yang dapat digunakan oleh petugas pemadam kebakaran, menampilkan rute optimal secara visual pada peta. (3) Validasi Lapangan: Melakukan validasi hasil rute yang diusulkan dengan kondisi nyata di lapangan, misalnya dengan membandingkan estimasi waktu tempuh dengan waktu tempuh aktual oleh unit pemadam kebakaran. (4) Skalabilitas Algoritma: Menguji performa algoritma pada jaringan graf yang lebih besar dan kompleks untuk mengevaluasi skalabilitasnya. (5) Studi Komparatif dengan Algoritma Heuristik: Membandingkan performa Dijkstra dan Bellman-Ford dengan algoritma heuristik lain yang mungkin menawarkan kompromi antara optimalitas dan kecepatan komputasi untuk graf yang sangat besar atau dinamis.

5. Kesimpulan

Penelitian ini berhasil melakukan analisis komparatif terhadap performa Algoritma Dijkstra dan Bellman-Ford untuk optimasi rute pemadam kebakaran pada model graf statis jaringan jalan Kota Praya. Hasil menunjukkan bahwa kedua algoritma mampu menghasilkan rute terpendek yang optimal. Namun, Algoritma Dijkstra menunjukkan efisiensi waktu komputasi yang secara signifikan lebih baik (rata-rata 0.5 detik) dibandingkan Algoritma Bellman-Ford (rata-rata 1.5 detik). Berdasarkan temuan ini, untuk aplikasi pada graf statis Kota Praya, Algoritma Dijkstra direkomendasikan karena kombinasi optimalitas dan kecepatannya. Penelitian lebih lanjut diperlukan untuk mengintegrasikan parameter dinamis guna meningkatkan akurasi dan relevansi sistem dalam kondisi nyata.

Referensi

- [1] N. Challands, "The relationships between fire service response time and fire outcomes," *Fire Technol*, vol. 46, no. 3, pp. 665–676, Jul. 2010, <https://doi.org/10.1007/s10694-009-0111-y>.
- [2] J. Monteiro, N. Sousa, J. Coutinho-Rodrigues, and E. Natividade-Jesus, "Challenges Ahead for Sustainable Cities: An Urban Form and Transport System Review," Jan. 01, 2024, Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/en17020409>.
- [3] Dataset - Satu Data NTB." Accessed: May 26, 2025. <https://data.ntbprov.go.id/dataset/9cc4c25d-fb17-44fc-be7e-b6932897007a/show>
- [4] D. S. Septifany, A. L. Nugraha, and M. Awaluddin, "ANALISIS OPTIMALISASI RUTE PEMADAM KEBAKARAN BERDASARKAN AREA CAKUPAN PIPA HIDRAN DI KOTA SEMARANG," *Jurnal Geodesi Undip*, vol. 6, no. 3, pp. 28-36, Sep. 2017, <https://ejournal3.undip.ac.id/index.php/geodesi/article/view/17183>.
- [5] B. Junanda, D. Kurniadi and Y. Huda, "PENCARIAN RUTE TERPENDEK MENGGUNAKAN ALGORITMA DIJKSTRA PADA SISTEM INFORMASI GEOGRAFIS PEMETAAN STASIUN PENGISIAN BAHAN BAKAR UMUM," *Jurnal Voteteknika*, vol. 4, no. 1, pp. 107-115, 2016, <http://dx.doi.org/10.24036/voteteknika.v4i1.6014>.
- [6] L. Y. . Sipayung, C. R. . Sinaga, and A. C. . Sagala, "Application of Dijkstra's Algorithm to Determine the Shortest Route from City Center to Medan City Tourist Attractions", *CNAHPC*, vol. 5, no. 2, pp. 648–655, Aug. 2023, <https://doi.org/10.47709/cnahpc.v5i2.2699>.
- [7] I. Putu, W. Gautama, and K. Hermanto, "Penentuan Rute Terpendek dengan Menggunakan Algoritma Dijkstra pada Jalur Bus Sekolah," *Jurnal Matematika*, vol. 10, no. 2, pp. 116–123, Dec. 2020, <https://doi.org/10.24843/JMAT.2020.V10.I02.P128>.
- [8] T. J. Arsita, N. T. Lapatta, Y. Y. Joefri, D. S. Angreni, and S. A. Pratama, "Optimization of Urban Waste Collection Routes Using the Held-Karp Algorithm in a Web and Mobile-Based System", *JAIC*, vol. 9, no. 1, pp. 202–210, Jan. 2025, <https://doi.org/10.30871/jaic.v9i1.8832>.
- [9] A. Nabila and R. Aprilia, "Pengoptimalan Jaringan Pipa Primer PUDAM Tirtabina Menggunakan Algoritma Greedy," *Jurnal Pendidikan Matematika : Judika Education*, vol. 7, no. 2, pp. 187-194, 2024, <https://journal.ipm2kpe.or.id/index.php/IJUDIKA/article/view/12391>.
- [10] 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, <https://doi.org/10.37676/jmi.v15i2.867>.
- [11] S. Andayani and E. W. Perwitasari, "Penentuan Rute Terpendek Pengambilan Sampah di Kota Merauke Menggunakan Algoritma Dijkstra," *Semantik*, vol. 4, no. 1, Nov. 2014, Accessed: May 26, 2025. <https://publikasi.dinus.ac.id/index.php/semantik/article/view/859>
- [12] L. Fitriani and R. N. Labani, "Rancang Bangun Sistem Informasi Geografis Monitoring Data Kebakaran di Dinas Pemadam Kebakaran," *Jurnal Algoritma*, vol. 16, no. 1, p. 27–33, 2019, <https://jurnal.itg.ac.id/index.php/algoritma/article/view/455>.
- [13] N. Yousefzadeh, R. Sengupta, and S. Ranka, "Dynamic Graph Attention Networks for Travel Time Distribution Prediction in Urban Arterial Roads," Dec. 2024, <https://doi.org/10.48550/arxiv.2412.11095>.
- [14] M. Apriyaningsih, A. Muid, and N. Nurhasanah, "Prototipe Sistem Pemadam Kebakaran Otomatis Berbasis Mikrokontroler Atmega328p," *PRISMA FISIKA*, vol. 5, no. 3, pp. 106–110, Oct. 2017, <https://jurnal.untan.ac.id/index.php/jpfu/article/view/22429>.
- [15] S. W. G. AbuSalim, R. Ibrahim, M. Z. Saringat, S. Jamel, and J. A. Wahab, "Comparative Analysis between Dijkstra and Bellman-Ford Algorithms in Shortest Path Optimization," vol. 917, no. 1, p. 012077, Sep. 2020, <https://doi.org/10.1088/1757-899X/917/1/012077>.
- [16] H. Yu, D. Zhang, and L. Rauchwerger, "An Adaptive Algorithm Selection Framework," *International Conference on Parallel Architectures and Compilation Techniques*, pp. 278–289, Sep. 2004, <https://doi.org/10.1109/PACT.2004.1342561>.