

Penentuan Rute Fasilitas Pelayanan Kesehatan Terdekat dengan Algoritma Bellman Ford

Yoel Pratama¹, Esmeralda Lemos², Amelia Lodan³, Shasabila Andryka⁴, Imelda Muda⁵, Putry Rihi⁶, Maiweddy Aridata⁷

^{1,2,3,4,5,6,7}Program Studi Ilmu Komputer, Universitas Nusa Cendana, Jl. Adisucipto, Penfui Kupang

Email: ¹yoelpatty1@gmail.com, ²elsalemos006@gmail.com, ³amelialodan10@gmail.com,
⁴shasabilandryka.09@gmail.com, ⁵imelmuda05@gmail.com, ⁶putryrihi06@gmail.com, ⁷wedvdata168@gmail.com

ABSTRAK

Akses cepat menuju fasilitas pelayanan kesehatan sangat diperlukan terutama pada kondisi darurat, sehingga dibutuhkan sistem yang mampu menentukan rute tercepat secara akurat. Penelitian ini mengimplementasikan algoritma Bellman-Ford pada sistem pemetaan interaktif untuk menentukan rute terpendek menuju fasilitas pelayanan kesehatan terdekat. Jaringan jalan dimodelkan sebagai graf berbobot yang diekstraksi dari data peta, di mana bobot setiap ruas jalan tidak hanya mempertimbangkan jarak geometris tetapi juga kondisi lapangan melalui pemberian penalti, seperti kemacetan dan adanya lampu merah. Algoritma Bellman-Ford digunakan untuk menghitung jarak minimum dari lokasi pengguna menuju fasilitas kesehatan dengan melakukan proses relaksasi secara berulang pada seluruh sisi graf. Hasil perhitungan divisualisasikan menggunakan Leaflet sehingga pengguna dapat melihat rute optimal secara langsung. Sistem yang dibangun mampu memberikan rekomendasi rute yang lebih realistis dan optimal berdasarkan kondisi lalu lintas yang disimulasikan, serta dinilai efektif untuk wilayah dengan jumlah simpul graf yang tidak terlalu besar.

Kata kunci : Rute Terpendek, Bellman-Ford, Graf Berbobot, Penentuan Rute, Fasilitas Pelayanan Kesehatan

ABSTRACT

Fast access to healthcare facilities is crucial, especially in emergency situations that require immediate medical attention. This research implements the Bellman-Ford algorithm in an interactive mapping system to determine the shortest route to the nearest healthcare facility. The road network is modeled as a weighted graph obtained from digital map data, where the weight on each road segment considers not only geometric distance but also real-world conditions through penalty adjustments, such as traffic congestion and the presence of traffic lights. The Bellman-Ford algorithm computes the minimum path from the user's location to healthcare facilities by repeatedly relaxing all graph edges until the optimal solution is achieved. The generated path is visualized using Leaflet, allowing users to observe optimal routes directly on the map interface. The results show that the system can provide more realistic and optimal route recommendations based on simulated traffic conditions and is effective for areas with relatively small to medium-sized graphs.

Keywords: Shortest Path, Bellman-Ford, Weighted Graph, Route Optimization, Healthcare Facilities

1. PENDAHULUAN

Akses yang cepat menuju fasilitas pelayanan kesehatan merupakan faktor penting dalam meningkatkan mutu layanan kesehatan, terutama pada kondisi darurat yang membutuhkan penanganan segera. Sistem navigasi berbasis algoritma jalur terpendek terbukti mampu membantu pengguna menemukan jalur tercepat menuju rumah sakit melalui penerapan algoritma Bellman-Ford pada perangkat seluler [1]. Oleh karena itu, kebutuhan akan sistem penentuan rute yang akurat semakin relevan dalam konteks pelayanan kesehatan modern.

Masalah menentukan rute tercepat pada dasarnya berkaitan dengan menemukan jalur dengan biaya minimum dalam sebuah graf. Metode manual seperti peta statis tidak mampu menangani dinamika jaringan jalan dan tidak menyediakan mekanisme komputasi untuk menghitung jarak minimum. Hal ini dapat diatasi melalui algoritma jalur terpendek yang dirancang khusus untuk menyelesaikan masalah pencarian rute pada graf berbobot, seperti algoritma Dijkstra dan algoritma Bellman-Ford di mana keduanya banyak digunakan dalam berbagai kasus pemetaan rute [2]. Implementasi kedua algoritma ini mampu memproses simpul dan sisi secara lebih efisien daripada pendekatan manual, sehingga memberikan hasil yang lebih optimal [3].

Penerapan algoritma jalur terpendek juga telah digunakan dalam sistem layanan publik yang memerlukan waktu perjalanan minimal. Dalam menentukan rute optimal untuk kendaraan operasional, algoritma Bellman-Ford terbukti mampu mengoptimalkan rute dengan mempertimbangkan bobot jarak yang bervariasi [4]. Selain itu, dalam skenario darurat seperti menentukan rute tercepat untuk unit pemadam kebakaran, algoritma jalur terpendek dapat memberikan rekomendasi rute yang lebih efektif melalui pemodelan graf yang terstruktur [5]. Hal ini menunjukkan bahwa penggunaan algoritma jalur terpendek relevan dalam membantu proses penentuan rute menuju fasilitas pelayanan kesehatan. Secara komputasi, algoritma Bellman-Ford memiliki keunggulan dalam menangani grafik dengan bobot yang beragam, termasuk graf yang mungkin memiliki bobot negatif, dan mampu melakukan perhitungan dengan struktur data yang fleksibel seperti representasi JavaScript Object Notation (JSON). Evaluasi kinerja algoritma jalur terpendek menunjukkan bahwa Bellman-Ford tetap kompetitif dibandingkan dengan algoritma Dijkstra dan A* dalam berbagai skenario pencarian rute [6]. Hal ini menjadikannya pilihan yang tepat untuk diterapkan dalam sistem penentuan rute fasilitas pelayanan kesehatan.

Kajian terhadap penelitian terdahulu menunjukkan bahwa implementasi algoritma Bellman-Ford dalam penentuan rute fasilitas kesehatan masih memiliki beberapa keterbatasan. Penelitian yang dilakukan oleh Setiawan, mengenai pencarian rute terpendek menuju rumah sakit di Kota Yogyakarta berbasis Android hanya memanfaatkan bobot jarak geometris tanpa mempertimbangkan faktor kondisi jalan seperti kemacetan atau lampu merah, sehingga rute yang dihasilkan bersifat statis dan kurang mencerminkan situasi nyata di lapangan. Selain itu, penelitian tersebut belum mengintegrasikan visualisasi berbasis peta interaktif dan tidak menyediakan dukungan lokasi dinamis melalui GPS, sehingga akurasi penentuan titik awal masih bergantung pada input manual [1].

Sementara itu, penelitian lain yang menerapkan algoritma Bellman-Ford pada perhitungan rute juga masih berfokus pada aspek komputasional graf tanpa menambahkan penalti kondisi nyata dan tanpa dukungan peta interaktif berbasis web. Penelitian-penelitian tersebut lebih menekankan perbandingan performa algoritma atau perhitungan rute sederhana tanpa mempertimbangkan dinamika jaringan jalan [7]. Menurut penelitian sebelumnya, termasuk pada studi di Yogyakarta teridentifikasi bahwa belum ada penelitian yang menggabungkan algoritma Bellman-Ford dengan pemodelan bobot dinamis, visualisasi peta interaktif, dan lokasi real-time untuk menghasilkan rute fasilitas kesehatan yang lebih realistis dan akurat dalam kondisi lapangan. Sehingga penelitian kami akan menggunakan penalti kondisi lapangan, menerapkan snap-to-road, dan menggunakan GPS real time.

Berdasarkan latar belakang tersebut, penelitian ini merumuskan tiga rumusan masalah, yaitu bagaimana menentukan rute terdekat antar lokasi, bagaimana melakukan representasi data lokasi menggunakan JavaScript Object Notation (JSON), dan bagaimana menerapkan algoritma Bellman-Ford untuk menentukan rute terpendek menuju fasilitas pelayanan kesehatan. Penelitian ini bertujuan untuk mengimplementasikan algoritma Bellman-Ford, membangun graf lokasi fasilitas kesehatan, serta menghasilkan rute terdekat menggunakan menggunakan HyperText Markup Language (HTML) dan JavaScript Object Notation (JSON). Penelitian ini diharapkan dapat memberikan manfaat bagi beberapa pihak. Bagi pengguna, diharapkan dapat membantu pengguna menemukan fasilitas pelayanan kesehatan terdekat dengan rute tercepat secara efisien. Bagi pengembang sistem, penelitian ini dapat menjadi referensi dalam pengembangan sistem navigasi berbasis algoritma jalur terpendek melalui penerapan algoritma Bellman-Ford. Bagi akademisi, penelitian ini dapat memberikan kontribusi dalam pengembangan algoritma jalur terpendek dan pemodelan graf dalam sistem layanan publik.

2. MATERI DAN METODE

Konsep Graf

Graf merupakan struktur yang digunakan untuk merepresentasikan objek-objek diskrit beserta hubungan di antara objek tersebut. Banyak permasalahan dalam kehidupan sehari-hari sebenarnya dapat digambarkan melalui graf. Contohnya adalah peta, yang dapat dilihat sebagai bentuk visual dari graf. Melalui representasi tersebut, berbagai analisis dapat dilakukan, seperti menentukan rute terpendek antara

dua lokasi, memberi warna berbeda pada dua kota yang berdekatan, merancang jalur transportasi, mengatur jaringan komunikasi maupun jaringan internet, dan berbagai aplikasi lainnya.

Secara matematis, graf didefinisikan sebagai berikut:

sebuah graf G adalah pasangan himpunan (V, E) , dengan:

V = himpunan tidak kosong berisi simpul (vertex atau node): $\{v_1, v_2, \dots, v_n\}$

E = himpunan sisi (edge atau arc) yang menghubungkan pasangan simpul: $\{e_1, e_2, \dots, e_n\}$

Dengan demikian, graf dapat dinotasikan sebagai $G = (V, E)$.

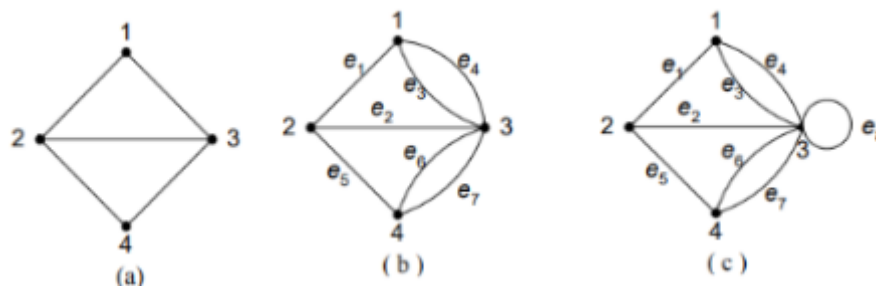
Jenis-Jenis Graf

Jika ditinjau dari keberadaan gelang (loop) dan sisi ganda, graf secara umum dapat dibedakan menjadi dua kategori:

1. Graf sederhana (simple graph), yaitu graf yang tidak mengandung gelang maupun sisi ganda.
2. Graf tidak sederhana (unsimple graph), yaitu graf yang memiliki gelang atau sisi ganda. Graf tidak sederhana terbagi menjadi dua bentuk:
 - a. Multigraph, yaitu graf yang memiliki lebih dari satu sisi yang menghubungkan simpul yang sama.
 - b. Pseudograph, yaitu graf yang mengandung gelang, yakni sisi yang menghubungkan simpul dengan dirinya sendiri.

Jika dilihat berdasarkan jumlah simpul yang dimiliki, graf juga dapat dikelompokkan menjadi dua:

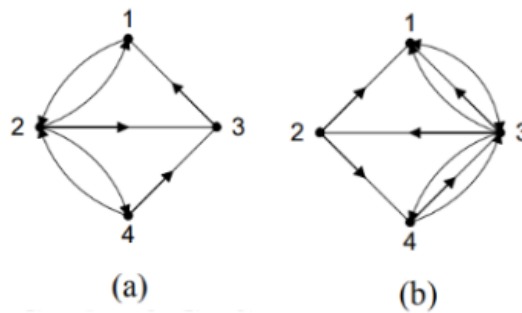
1. Graf berhingga (limited graph), yaitu graf dengan jumlah simpul yang terbatas.
2. Graf tak berhingga (unlimited graph), yaitu graf yang jumlah simpulnya tidak terbatas.



Gambar 1. Beberapa Graf a) graf sederhana b) graf ganda c) graf semu

Berdasarkan ada atau tidaknya orientasi pada setiap sisi, graf dapat dibedakan menjadi dua jenis:

1. Graf berarah (directed graph), yaitu graf yang setiap sisinya memiliki arah tertentu. Pada graf jenis ini, pasangan simpul (v_j, v_k) dan (v_k, v_j) dianggap sebagai dua busur yang berbeda, sehingga $(v_j, v_k) \neq (v_k, v_j)$. Pada busur (v_j, v_k) , simpul v_j disebut simpul asal (*initial vertex*), sedangkan v_k disebut simpul terminal (*terminal vertex*).
2. Graf takberarah (undirected graph), yaitu graf yang sisinya tidak memiliki arah. Pada graf ini, urutan simpul yang dihubungkan tidak berpengaruh, sehingga (v_j, v_k) sama dengan (v_k, v_j) .



Gambar 2. Graf berarah (directed graph)

Teori rute terpendek

Lintasan terpendek merupakan salah satu konsep penting dalam teori graf. Pada graf berbobot, permasalahan jarak terpendek berkaitan dengan upaya menemukan jalur yang memiliki total bobot paling kecil di antara semua jalur yang mungkin. Masalah ini termasuk dalam kategori optimasi, di mana tujuannya adalah memilih solusi paling efisien dari berbagai alternatif jalur yang tersedia pada suatu graf.[8] Algoritma Bellman-Ford digunakan untuk menghitung jarak terpendek dari satu simpul sumber pada graf berbobot. Algoritma ini menentukan nilai jarak minimum dari simpul asal menuju seluruh simpul lainnya dalam graf.

Algoritma Bellman-Ford

Algoritma Bellman-Ford merupakan salah satu metode yang digunakan untuk menemukan lintasan terpendek pada sebuah graf. Sama seperti algoritma Dijkstra, algoritma ini menghitung jarak terpendek dari satu simpul sumber menuju seluruh simpul lainnya dalam graf berbobot. Yang dimaksud dengan *satu sumber* adalah bahwa perhitungan jarak dilakukan dengan titik awal yang sama untuk seluruh tujuan.[9] Keunggulan utama Bellman-Ford dibandingkan algoritma pencarian jarak terpendek lainnya adalah kemampuannya menangani graf yang memiliki bobot sisi bernilai negatif.

Berikut adalah pseudocode Algoritma Bellman-Ford yang diadaptasi dari buku *Introduction to Algorithms* [10]:

```

BELLMAN-FORD( $G, w, s$ )
  INITIALIZE-SINGLE-SOURCE( $G, s$ )
  for  $i = 1$  to  $|G.V| - 1$ 
    for each edge  $(u, v) \in G.E$ 
      RELAX( $u, v, w$ )
  for each edge  $(u, v) \in G.E$ 
    if  $v.d > u.d + w(u, v)$ 
      return FALSE
  return TRUE

```

Adapun fungsi *Initialize-Single-Source* dan *Relax* dituliskan sebagai berikut:

```

INITIALIZE-SINGLE-SOURCE( $G, s$ )

```

```

for each vertex  $v \in G.V$ 

     $v.d = \infty$ 

     $v.\pi = \text{NIL}$ 

 $s.d = 0$ 

RELAX( $u, v, w$ )

    if  $v.d > u.d + w(u, v)$ 

         $v.d = u.d + w(u, v)$ 

         $v.\pi = u$ 

```

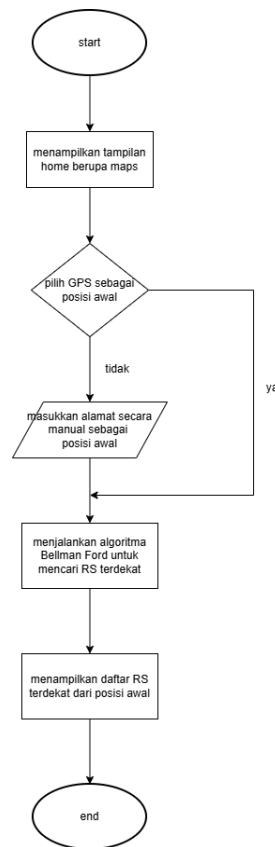
Penjelasan Pseudocode

Melakukan inisialisasi graf G serta bobot setiap sisi w , dengan menentukan simpul asal s .

1. Semua simpul diberikan nilai awal tak hingga (*infinity*) sebagai jarak sementara.
2. Setiap simpul diatur agar tidak memiliki *predecessor* ($\pi = \text{NIL}$).
3. Nilai jarak simpul awal (s) diatur menjadi 0.
4. Algoritma menjalankan proses relaksasi sebanyak $|V| - 1$ kali (sesuai jumlah simpul dikurangi satu).
5. Pada setiap iterasi, dilakukan pemeriksaan terhadap seluruh sisi (u, v) dalam graf.
6. Proses berpindah ke fungsi *Relax* untuk mengecek apakah nilai jarak menuju v ($v.d$) lebih besar dibanding jarak menuju u ($u.d$) ditambah bobot sisi antara keduanya $w(u, v)$.
7. Jika kondisi tersebut terpenuhi, maka jarak menuju v diperbarui dengan nilai baru tersebut dan u ditetapkan sebagai *predecessor* dari v ($v.\pi$). Hal ini digunakan untuk merekam jalur yang membentuk lintasan terpendek.
8. *Predecessor* inilah yang nantinya mempermudah pelacakan kembali jalur yang menjadi solusi akhir.
9. Langkah nomor 5–7 pada pseudocode utama berfungsi mendeteksi keberadaan *negative-weight cycle* dalam graf, yaitu siklus dengan total bobot negatif.
10. Pemeriksaan dilakukan dengan cara memastikan tidak ada sisi (u, v) yang masih dapat direlaksasi setelah seluruh iterasi selesai.
11. Jika nilai $v.d$ masih dapat dikurangi, maka dapat dipastikan graf memiliki siklus berbobot negatif, sehingga algoritma mengembalikan nilai false.

Metode

Diagram Alur Penelitian (Flowchart)



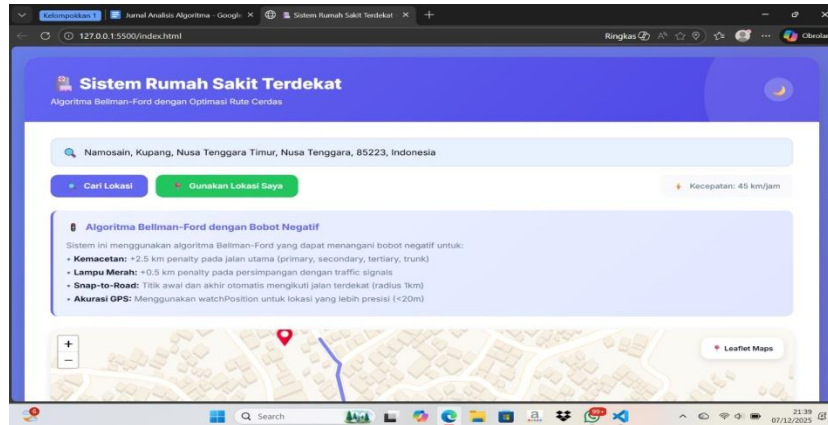
Flowchart ini menggambarkan proses sistem dalam menentukan rumah sakit terdekat berdasarkan lokasi awal pengguna. Proses dimulai dengan menampilkan halaman utama berupa peta yang berfungsi sebagai antarmuka awal bagi pengguna. Pada tahap ini, pengguna diminta untuk memilih sumber lokasi awal yang akan digunakan dalam proses pencarian rute. Sistem kemudian memberikan opsi keputusan, yaitu apakah pengguna ingin menggunakan GPS sebagai posisi awal atau tidak. Jika pengguna memilih menggunakan GPS, sistem secara otomatis mengambil koordinat lokasi pengguna dan langsung menjalankan algoritma Bellman-Ford untuk menghitung jarak menuju seluruh rumah sakit yang tersedia. Sebaliknya, jika pengguna tidak memilih GPS, maka pengguna harus menentukan lokasi awal secara manual, misalnya dengan memasukkan alamat atau memilih titik pada peta. Setelah lokasi awal ditetapkan, sistem kembali menjalankan algoritma Bellman-Ford dengan menggunakan lokasi tersebut sebagai titik sumber. Tahap selanjutnya adalah proses perhitungan untuk menentukan rumah sakit yang memiliki jarak paling dekat dari posisi awal. Setelah jarak terpendek berhasil dihitung, sistem menampilkan informasi rumah sakit terdekat kepada pengguna. Proses ini kemudian diakhiri setelah data rumah sakit terdekat berhasil ditampilkan.

3. HASIL DAN PEMBAHASAN

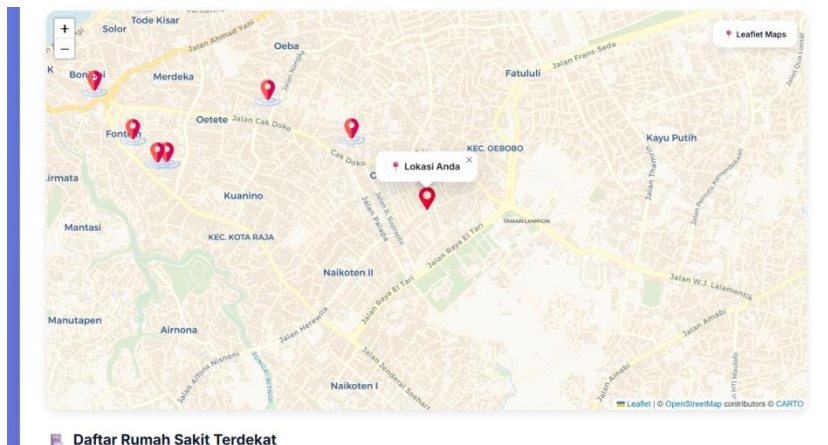
Tampilansistem (UI + screenshot)

Ketika aplikasi dijalankan, akan muncul halaman utama yang memuat peta interaktif (Leaflet) yang memuat marker untuk fasilitas pelayanan kesehatan (rumah sakit, puskesmas, klinik) dan marker lokasi pengguna. Antarmukamenyediakan:

1. Kotak pencarianalamat (GeoSearch) dan tombol “Gunakan Lokasi Saya” (watchPosition).
2. Panel informasi yang menampilkan: nama fasilitas terpilih, jarak terhitung, waktu komputasi algoritma, dan opsi penalti (kemacetan, lampu merah).
3. Kontrol *snap-to-road* untuk menyesuaikan titik input ke jaringan jalan terdekat dalam radius tertentu.



Lalu jika menggunakan fitur lokasi saya atau cari lokasi, saat menekan leafletmaps akan muncul simulasi-simpul fasilitas pelayanan kesehatan terdekat sesuai lokasi yang kita masukan. Leaflet seperti mesin peta interaktif yang menampilkan lokasi pengguna, lokasi fasilitas kesehatan, dan rute hasil perhitungan Algoritma Bellman-Ford. Ketika program dibuka, Leaflet pertama-tama memuat peta dasar dari *OpenStreetMap* dan menampilkannya di layar. Setelah itu, program menempatkan marker-marker di atas peta untuk menunjukkan posisi rumah sakit dan titik pengguna. Saat pengguna mengaktifkan lokasi GPS atau memilih titik tertentu, Leaflet memperbarui posisi marker tersebut di peta. Ketika algoritma Bellman-Ford selesai menghitung rute terpendek, program mengirim daftar titik koordinat ke Leaflet.

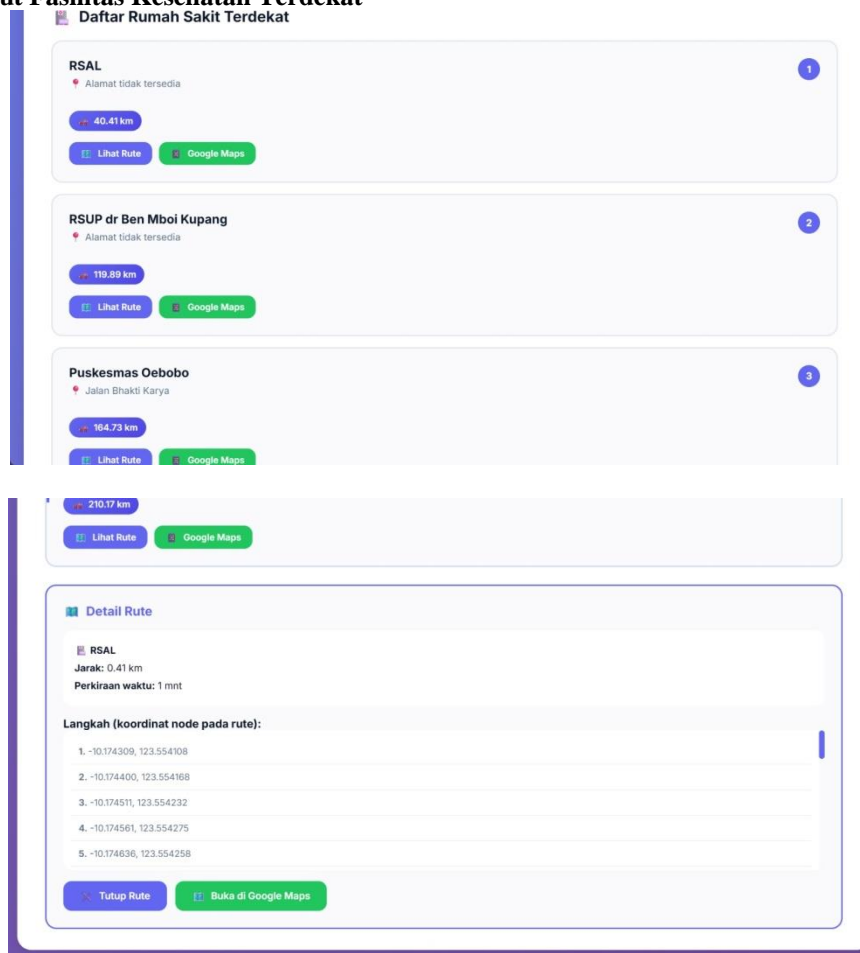


Cara kerja Bellman-Ford di program

Aplikasi ini menggunakan algoritma Bellman Ford untuk mencari rute terbaik dari lokasi pengguna menuju ke fasilitas pelayanan kesehatan terdekat dimana jarak tiap ruas jalan dimodifikasi dengan penalti sesuai kondisi nyata di lapangan seperti yang tertulis pada layar: kemacetan di jalan utama diberi tambahan +2.5 km, dan lampu merah di persimpangan diberi tambahan +0.5 km. Bobot ini membuat jalur yang tampak dekat di peta tetapi macet atau banyak trafficlight menjadi “lebih jauh” secara perhitungan, sehingga Bellman-Ford akan memilih rute yang paling cepat dan lancar. Sistem juga otomatis menyesuaikan titik awal/akhir ke jalan terdekat (Snap-to-Road) dan menggunakan GPS (*watchPosition*) untuk menentukan posisi pengguna secara presisi. Setelah perhitungan selesai, aplikasi menampilkan rumah sakit yang benar-benar paling optimal untuk dituju, bukan hanya yang terdekat secara jarak peta.



Hasil Output Fasilitas Kesehatan Terdekat



Kelebihan dan kekurangan sistem

Kelebihan

1. Bisa Menggunakan Bobot Negatif (Cocok untuk Penalti). Sistem memberikan penalti +2.5 km untuk jalan utama (kemacetan) dan +0.5 km untuk lampu merah. Penalti ini dapat direpresentasikan sebagai bobot positif atau negatif, dan Bellman-Ford bisa memprosesnya tanpa error.

2. Mampu Deteksi *Negative Cycle* jika Ada Kesalahan Data Peta. Jika data graph dari Anda mengandung kesalahan (misal dua node saling terhubung dengan bobot negatif yang membentuk loop), Bellman-Ford dapat mendeteksi itu. Mengurangkan siklus "berputar-putar tak berakhir"
3. Cocok untuk peta yang dibentuk manual (graph non-standar). Sistem membentuk graph dari Snap-to-road (radius 1 km), titik start & end otomatis, representasi JSON graph hasil perhitungan. Graph hasil pengolahan manual seperti ini lebih aman diproses oleh Bellman-Ford dibanding Dijkstra, karena lebih toleran terhadap anomali bobot.
4. Implementasi lebih mudah untuk rute dengan banyak penalti. Kode Bellman-Ford untuk menghitung rute \rightarrow cukup loop $(V-1) \times \text{edges}$. Meskipun lambat, lebih mudah dimodifikasi, misalnya tambah penalti waktu, tambah penalti tingkat kemacetan dan tambah kelas jalan primary,
5. Aman Digunakan untuk Area Kota Kecil (Graf Tidak Terlalu Besar). Wilayah Kupang atau kecamatan seperti Namosain punya jumlah edge relatif sedikit dibanding kota besar. Waktu komputasi Bellman-Ford tetap aman karena graph ukuran kecil.

Kekurangan

1. Lebih Lambat (Boros Performa) karena kompleksitas Bellman-Ford = $O(V \times E)$, memakai peta satu kota penuh, menambah ribuan node dari openstreetmap dan menghitung rute berulang kali. Maka rute akan terasa lebih lambat untuk dihitung.
2. Tidak efisien untuk rute panjang / kota besar. Jika sistem diperluas untuk seluruh NTT atau Indonesia, Bellman-Ford menjadi tidak ideal. Sistem ini bisa mengalami loading lama, UI freeze saat menghitung dan worker web lambat.
3. Tidak menggunakan informasi heuristik. Berbeda dengan A*, Bellman-Ford tidak mempertimbangkan arah menuju tujuan. Kekurangannya adalah perhitungan rute tidak dipandu, semua edge dihitung meski jauh dari jalur logis boros proses jika peta besar.
4. Penalti yang Banyak Membuat Bobot Bisa Tidak Stabil seperti lampu merah, kemacetan, dan snap-to-road. Jika salah set, bobot negatif bisa membuat rute tidak realistis, misalnya memilih jalur memutar dan menghindari jalan utama padahal lebih cepat. Bellman-Ford akan memprosesnya tetapi tidak menjamin logikanya benar.

Perbandingan dengan algoritma lain

Aspek	Bellman–Ford	Dijkstra	Floyd–Warshall	Prim's	Kruskal
Jenis masalah	Shortest Path (single-source)	Shortest Path (single-source)	Shortest Path (all-pairs)	MST	MST
Bobot negatif	Bisa	Tidak	Bisa	Bisa	Bisa
Negative cycle detection	Bisa	Tidak	Bisa	Tidak	Tidak
Kompleksitas	$O(V \times E)$	$O(E \log V)$	$O(V^3)$	$O(E \log V)$	$O(E \log E)$
Cocok untuk graf besar	Tidak	Bisa	Tidak	Bisa	Bisa
Output	Jalur terpendek	Jalur terpendek	Matriks jarak antar semua node	MST	MST
Kelemahan	Lambat	Tidak dukung negatif	Berat untuk graf besar	Tidak untuk shortest path	Tidak untuk shortest path

4. KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem penentuan rute fasilitas pelayanan kesehatan terdekat menggunakan algoritma Bellman-Ford, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Algoritma Bellman-Ford berhasil diimplementasikan secara efektif dalam sistem penentuan rute fasilitas pelayanan kesehatan terdekat dengan memodelkan jaringan jalan sebagai graf berbobot. Algoritma ini mampu menentukan lintasan terpendek dari lokasi pengguna menuju fasilitas kesehatan dengan mempertimbangkan variasi bobot ruas jalan.
2. Penerapan bobot dan penalti berbasis kondisi lapangan seperti kemacetan, lampu merah, dan penyesuaian snap-to-road dapat meningkatkan realisme hasil rute yang dihasilkan. Sistem tidak hanya menentukan jarak terdekat secara geometris, tetapi juga rute yang lebih optimal berdasarkan kondisi lalu lintas yang disimulasikan.
3. Integrasi algoritma Bellman-Ford dengan antarmuka peta interaktif (Leaflet) mampu menampilkan hasil rute secara visual dan informatif, sehingga memudahkan pengguna dalam memahami jalur yang direkomendasikan. Sistem ini dinilai cukup andal untuk diterapkan pada wilayah dengan cakupan graf kecil hingga menengah.
4. Keterbatasan utama sistem terletak pada kompleksitas komputasi algoritma Bellman-Ford, yang cenderung lebih lambat ketika digunakan pada graf berukuran besar atau pada perhitungan rute yang dilakukan secara berulang. Oleh karena itu, penerapan sistem ini saat ini lebih sesuai untuk cakupan wilayah lokal atau kota dengan jumlah simpul yang terbatas.

Saran

Berdasarkan hasil penelitian dan keterbatasan yang ditemukan, beberapa saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Mengembangkan sistem dengan algoritma alternatif atau hybrid, seperti A* atau Dijkstra, terutama untuk wilayah dengan graf yang lebih besar, guna meningkatkan efisiensi waktu komputasi dan respons sistem.
2. Mengoptimalkan model bobot ruas jalan berbasis waktu tempuh, bukan hanya jarak, dengan memasukkan parameter kecepatan kendaraan, kondisi lalu lintas secara dinamis, atau data historis kemacetan agar rekomendasi rute menjadi lebih akurat.
3. Memperluas cakupan data dan wilayah pemetaan, misalnya hingga tingkat kota atau provinsi, dengan tetap memperhatikan performa sistem melalui optimasi struktur data dan pemrosesan algoritma di sisi server atau menggunakan web worker.
4. Meningkatkan aspek usability dan evaluasi pengguna, seperti melakukan pengujian langsung kepada pengguna atau tenaga medis, sehingga sistem dapat dikembangkan lebih sesuai dengan kebutuhan nyata di lapangan, khususnya dalam kondisi darurat.

DAFTAR PUSTAKA

- [1] R. Setiawan, R. G. Santosa, and J. K. Tampubolon, "Implementasi Algoritma Bellman-Ford untuk Pencarian Jalur Terpendek Menuju Rumah Sakit di Kota Yogyakarta Berbasis Android," *J. Terap. Teknol. Inf.*, vol. 3, no. 2, pp. 95–104, 2021, doi: 10.21460/jutei.2019.32.184.
- [2] I. Djafar and A. Dijkstra, "Single-Source Shortest Path Pada Graf Berbobot Menggunakan Algoritma Dijkstra Dan Bellman-," *SISITI Semin. Ilm. Sist. Inf. dan Teknol. ...*, 2017.
- [3] Dhea Ayu Devi Mayang Sari, Granita, and Dinda Handayani, "Penerapan Algoritma Dijkstra dan Algoritma Bellman-Ford untuk Menentukan Rute Terpendek," *JMT J. Mat. dan Terap.*, vol. 6, no. 1, pp. 35–41, 2024, doi: 10.21009/jmt.6.1.5.

-
- [4] R. E. Putri Harahap and I. Husein, "Bellman-ford and greedy algorithms to optimize the shortest route of PT. TIKI Jalur NugrahaEkakurir (JNE)," *Desimal J. Mat.*, vol. 7, no. 2, pp. 453–462, Aug. 2024, doi: 10.24042/djm.v7i2.23792.
- [5] P. Kebakaran, "JTIM: JurnalTeknologiInformasi dan Multimedia ImplementasiAlgoritma Dijkstra dan Bellman-Ford untuk Op-," vol. 7, no. 2, pp. 398–407, 2025.
- [6] A. B. A. A. B. R and A. S. A. Ahmed, "Designing and Implementing Shortest and Fastest Paths; A Comparison of Bellman-Ford algorithm, A*, and Dijkstra's algorithms," *Int. J. Comput. Trends Technol.*, vol. 69, no. 5, pp. 6–12, 2021, doi: 10.14445/22312803/ijctt-v69i5p102.
- [7] R. Munir, *Studi dan Implementasi Persoalan Lintasan Terpendek Suatu Graf dengan Algoritma Dijkstra dan Algoritma Bellman-Ford*, Institut Teknologi Bandung, 2006.
- [8] A. A. Faqih, A. F. Arifianto, and A. Y. Wijaya, "Optimasi Jalur Tercepat dengan Menggunakan Modifikasi Algoritma Bellman-Ford (Studi Kasus Lintasan antar Kecamatan Kota Malang)," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 3, pp. 321–328, 2018. \
- [9] R. Setiawan, R. G. Santosa, and J. K. Tampubolon, "Implementasi Algoritma Bellman-Ford untuk Pencarian Jalur Terpendek Menuju Rumah Sakit di Kota Yogya Berbasis Android," Program Studi Informatika, Universitas Kristen Duta Wacana, Yogyakarta, 2019.
- [10] Cormen, T. H. (2009). Introduction to Algorithms (Vol. III). Cambridge: The MIT Press.