

ESTIMASI RUTE TERDEKAT DARI UNIVERSITAS NEGERI MEDAN KE SPBU TERDEKAT MENGGUNAKAN ALGORITMA *GREEDY*

Freyro Dobry Sianipar, Muhammad Hidayatul Arifin, Windy Aulia, Putri Harliana

Prodi Ilmu Komputer, Fakultas MIPA, Universitas Negeri Medan

Jl. William Iskandar Ps. V, Medan Estate, Kec. Percut Sei Tuan

Kabupaten Deli Serdang, Sumatera Utara 20222, Sumatera Utara

freYROdobry@gmail.com

ABSTRAK

Penentuan rute terdekat merupakan masalah optimasi yang penting dalam berbagai bidang, terutama dalam transportasi dan navigasi. Penelitian ini bertujuan untuk mengestimasi rute terdekat dari Universitas Negeri Medan (Unimed) menuju beberapa Stasiun Pengisian Bahan Bakar Umum (SPBU) di sekitarnya dengan menggunakan algoritma *Greedy*. Algoritma *Greedy* dipilih karena kesederhanaannya dan kemampuannya untuk memberikan solusi yang cepat dengan memilih jarak terpendek pada setiap langkah. Meskipun demikian, algoritma ini hanya mempertimbangkan solusi lokal terbaik dan tidak menjamin solusi optimal secara global. Penelitian ini dilakukan secara perkiraan tanpa data geografis aktual, namun tetap memberikan pemahaman mendalam tentang cara kerja algoritma *Greedy* dalam kasus rute jarak dekat. Hasil penelitian menunjukkan bahwa algoritma *Greedy* dapat menghasilkan estimasi rute yang efisien dan cepat, meskipun solusi yang dihasilkan mungkin tidak selalu optimal secara keseluruhan. Peta visualisasi yang dihasilkan dari simulasi algoritma ini membantu memperjelas pola pergerakan yang dipilih. Penelitian ini memberikan wawasan tentang potensi penerapan algoritma *Greedy* dalam sistem navigasi sederhana dan bagaimana pendekatan ini dapat diadaptasi untuk skenario yang lebih kompleks di masa depan.

Kata kunci : Algoritma *Greedy*, Rute Terdekat, Universitas Negeri Medan, SPBU, Optimasi Rute, Navigasi.

1. PENDAHULUAN

Masalah penentuan rute terdekat atau *shortest path problem* adalah salah satu masalah optimasi yang sangat umum dalam bidang ilmu komputer, matematika, dan rekayasa transportasi. Dalam konteks kehidupan sehari-hari, masalah ini sering kali dijumpai pada layanan navigasi, pengiriman barang, dan sistem transportasi umum. Optimalisasi rute tidak hanya menghemat waktu tetapi juga biaya, terutama ketika berkaitan dengan jarak tempuh dan penggunaan bahan bakar. Dalam penelitian ini, kami berfokus pada penerapan algoritma *Greedy* untuk mencari rute terdekat dari Universitas Negeri Medan (Unimed) menuju Stasiun Pengisian Bahan Bakar Umum (SPBU) di sekitarnya. Algoritma *Greedy* dipilih karena kesederhanaannya dan kemampuan komputasi yang cepat dalam menentukan solusi lokal terbaik.

Algoritma *Greedy* adalah salah satu pendekatan yang paling sederhana untuk menyelesaikan masalah optimasi. Prinsip dasarnya adalah selalu memilih solusi terbaik pada setiap langkah, dengan harapan bahwa keputusan lokal yang optimal akan menghasilkan solusi global yang optimal pula. Pada masalah penentuan rute terdekat, algoritma *Greedy* akan selalu memilih titik yang memiliki jarak terdekat dari posisi saat ini, dan kemudian melanjutkan ke titik berikutnya yang juga memiliki jarak terdekat dari titik sebelumnya. Pendekatan ini sangat efisien dalam waktu komputasi, namun tidak selalu memberikan solusi optimal secara global.

Algoritma *Greedy* ini mengembangkan solusi secara bertahap, di mana setiap langkahnya

menawarkan berbagai pilihan dan kemungkinan yang dapat dijelajahi. Dengan algoritma ini, keputusan untuk langkah selanjutnya diambil berdasarkan keuntungan maksimal dari keadaan saat ini [1].

Algoritma *Greedy* merupakan salah satu metode untuk menyelesaikan masalah optimasi. Metode ini juga dapat menyelesaikan masalah secara bertahap, dengan memilih opsi terbaik pada setiap langkahnya.

Penentuan rute terdekat merupakan bagian penting dalam perencanaan perjalanan, baik untuk keperluan pribadi maupun bisnis. Dengan semakin banyaknya kendaraan bermotor yang beroperasi di kota-kota besar seperti Medan, menentukan rute terdekat ke SPBU dapat membantu pengemudi menghemat waktu dan bahan bakar. Universitas Negeri Medan sebagai pusat aktivitas pendidikan yang besar juga sering kali menjadi titik awal perjalanan menuju berbagai fasilitas umum di sekitarnya, termasuk SPBU. Oleh karena itu, penelitian ini bertujuan untuk mengidentifikasi rute terdekat dari Unimed ke beberapa SPBU di sekitarnya menggunakan algoritma *Greedy*.

Selain itu, teknologi sistem navigasi modern yang banyak digunakan oleh masyarakat umum, seperti Google Maps dan Waze, juga bergantung pada algoritma optimasi rute untuk memberikan petunjuk arah yang paling efisien [2].

Meskipun algoritma yang digunakan pada aplikasi tersebut lebih kompleks, seperti algoritma Dijkstra atau A*, studi ini menunjukkan bagaimana algoritma sederhana seperti *Greedy* dapat digunakan dalam kasus yang lebih terbatas atau dalam konteks

yang lebih sederhana. *Greedy* dapat menjadi solusi alternatif yang cepat dan mudah diterapkan dalam situasi di mana solusi optimal secara global tidak selalu diperlukan.

Dalam konteks teoritis, algoritma *Greedy* dikenal memiliki beberapa keterbatasan, terutama dalam masalah optimasi yang lebih kompleks seperti *Traveling Salesman Problem* (TSP). Pada TSP, seorang penjual harus mengunjungi sejumlah kota dan kembali ke kota asal dengan total jarak terpendek. Algoritma *Greedy* cenderung memberikan solusi sub-optimal karena hanya mempertimbangkan jarak terpendek pada setiap langkah tanpa memperhitungkan konsekuensi jangka panjang dari pilihan tersebut. Meskipun demikian, dalam skenario yang lebih sederhana seperti pencarian rute terdekat dari satu titik ke beberapa titik tujuan (dalam hal ini Unimed ke SPBU), algoritma *Greedy* masih dapat menghasilkan solusi yang memadai.

Penelitian-penelitian sebelumnya juga menunjukkan bahwa algoritma *Greedy* sangat cocok untuk kasus-kasus sederhana di mana jumlah titik yang harus dikunjungi relatif sedikit dan waktu komputasi menjadi faktor penting. Misalnya, dalam penelitian terkait sistem distribusi dan logistik, algoritma *Greedy* sering kali diterapkan untuk menentukan rute pengiriman barang dalam skala kecil atau menengah. Hasil yang diperoleh dari algoritma *Greedy* mungkin tidak selalu optimal, namun waktu komputasi yang cepat menjadikannya pilihan yang baik dalam konteks di mana efisiensi waktu lebih diutamakan daripada solusi optimal.

Di sisi lain, penelitian yang melibatkan optimasi rute dengan data geografis sebenarnya sering kali menggunakan algoritma yang lebih kompleks dan mempertimbangkan berbagai faktor selain jarak. Dalam dunia nyata, faktor seperti kondisi jalan, lalu lintas, dan bahkan cuaca bisa sangat mempengaruhi pemilihan rute terdekat. Meskipun algoritma *Greedy* hanya mempertimbangkan jarak secara langsung, hasil dari penelitian ini dapat dijadikan dasar untuk pengembangan algoritma yang lebih canggih di masa depan, yang mempertimbangkan faktor-faktor lain yang memengaruhi perjalanan.

Algoritma *Greedy* tidak hanya relevan dalam bidang transportasi tetapi juga digunakan dalam berbagai aplikasi lain. Dalam bidang komputasi, algoritma ini sering digunakan dalam pemrograman heuristik, seperti pada pemecahan masalah pencarian lokal, alokasi sumber daya, dan pemilihan rute dalam jaringan komputer. Kekuatan algoritma ini terletak pada kesederhanaannya, yang memungkinkan pengguna untuk memecahkan masalah kompleks dengan pendekatan yang mudah diimplementasikan dan dengan waktu komputasi yang cepat. Dalam penelitian ini, kesederhanaan dan kecepatan *Greedy* diandalkan untuk menyelesaikan masalah penentuan rute terdekat dari Unimed ke beberapa SPBU.

Dengan demikian, penelitian ini tidak hanya bertujuan untuk menerapkan algoritma *Greedy* dalam

mencari rute terdekat dari Universitas Negeri Medan ke SPBU terdekat, tetapi juga untuk menunjukkan bagaimana algoritma ini dapat menjadi solusi yang efisien untuk masalah optimasi sederhana dalam kehidupan sehari-hari. Meskipun memiliki keterbatasan, algoritma *Greedy* tetap dapat memberikan solusi yang efektif dan cepat, yang sangat berguna dalam aplikasi praktis yang membutuhkan keputusan cepat. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan lebih lanjut sistem navigasi dan perencanaan rute yang lebih efisien.

2. TINJAUAN PUSTAKA

Dalam penelitian sebelumnya, Ginting dkk. membahas penerapan algoritma *Greedy* untuk mencari lokasi sekolah terdekat di Kota Medan. Algoritma ini berhasil menemukan rute terpendek dengan akurasi yang mencapai lebih dari 95%, sehingga memberikan solusi efisien bagi pengguna dalam memilih sekolah berdasarkan jarak [2].

Sejalan dengan itu, Hadibrata dan Maudin juga menerapkan Algoritma *Greedy* dalam konteks yang berbeda, yakni untuk menentukan rute tercepat menuju objek wisata di Kota Cirebon. Penelitian ini menunjukkan bahwa algoritma tersebut efektif dalam membantu wisatawan menghindari kemacetan dengan memberikan rute alternatif yang lebih cepat, mempermudah akses ke lokasi wisata [3].

Kedua studi ini memberikan bukti yang signifikan mengenai keefektifan Algoritma *Greedy* dalam mengoptimalkan rute di berbagai bidang, mulai dari pendidikan hingga pariwisata. Selain itu, penelitian ini menunjukkan potensi penerapan yang lebih luas dalam pengembangan sistem yang berbasis lokasi

2.1. Rute

Rute merupakan jarak atau arah yang harus ditempuh dan dilewati secara berurutan. Rute tercepat adalah jarak dari titik awal menuju tujuan yang ditempuh secara berurutan dalam jangka waktu tertentu. Secara lebih spesifik, rute tercepat adalah lintasan dengan jarak minimum yang membutuhkan waktu tertentu untuk mencapai lokasi tujuan dari satu tempat ke tempat [3].

2.2. Algoritma Greedy

Algoritma *Greedy* adalah algoritma yang memperoleh solusi optimal dari masalah optimasi dengan cara menyelesaikan langkah demi langkah. Algoritma *Greedy* merupakan algoritma yang dapat memecahkan masalah dengan menemukan alternatif paling baik pada saat tertentu. Walaupun setiap solusi jawaban dari algoritma *greedy* tidak dipastikan merupakan solusi optimum global, tetapi dapat diandalkan untuk mengoptimasi waktu [4].

Algoritma ini merupakan algoritma yang sederhana dan fleksibel sehingga dapat digunakan pada berbagai kasus persoalan dengan hasil yang

cukup memuaskan [5].

Algoritma Greedy merupakan metode optimasi dengan graf berbobot yang memiliki verteks sebagai titik lokasi dan edge sebagai jarak tempuh[6].

Sebuah metode yang mampu memecahkan masalah optimasi secara bertahap melalui proses yang terus berkembang hingga masalah terselesaikan. Algoritma Greedy dapat menentukan rute dengan mengambil langkah awal yang memberikan solusi lokal terbaik, yang kemudian menghasilkan solusi optimal secara keseluruhan untuk menemukan rute tercepat dari Madiun menuju Telaga Sarangan. Selain itu, Algoritma Greedy membuat beberapa pilihan terbaik secara langsung pada setiap langkah. Beberapa konsep penting dalam optimasi menggunakan Algoritma Greedy meliputi [7]:

- Himpunan Kandidat (C): Berisi elemen-elemen yang membentuk solusi. Setiap langkah, kandidat dipilih dari himpunan ini.
- Himpunan Solusi (S): Berisi kandidat-kandidat yang terpilih untuk menjadi solusi dari permasalahan.
- Fungsi Seleksi: Memilih kandidat terbaik di setiap langkah untuk mencapai solusi optimal. Fungsi ini berdasarkan pada fungsi objektif yang digunakan.
- Fungsi Kelayakan (Feasible): Memeriksa kandidat untuk memastikan solusi layak dan tidak melanggar batasan yang ada.
- Fungsi Objektif: Digunakan untuk memaksimalkan atau meminimalkan nilai solusi.

Untuk mencari rute tercepat atau lintasan terpendek dengan Algoritma Greedy, langkah-langkahnya adalah sebagai berikut:

- Periksa sisi yang terhubung dengan lokasi pertama, lalu pilih sisi dengan bobot (jarak) terkecil sebagai rute terpendek pertama, yang bisa diasumsikan sebagai (L1).
- Untuk menentukan rute terpendek berikutnya (D), gunakan langkah berikut: a) Cari rute berikutnya dengan rumus $D(i) = L1 + \text{bobot sisi berikutnya}$, di mana jarak pertama dijumlahkan dengan bobot sisi berikutnya. Jika ada rute lain, tambahkan jarak rute tersebut ke jarak sebelumnya. b) Pilih $D(i)$ dengan jarak terkecil, dan jika ada rute lain yang lebih pendek, pertimbangkan rute tersebut. c) Gunakan metode yang sama untuk mencari lintasan terpendek selanjutnya.

2.3. Teori Graf

Teori graf yaitu ilmu yang mempelajari tentang sifat-sifat grafik. Graf merupakan dari himpunan beberapa benda dan disebut dengan simpul. Simpul biasanya dihubungkan dengan garis panah dan saling terhubung. Banyak struktur graf yang dapat dipresentasikan, misalnya dalam pertemanan pada aplikasi Twitter. Pengguna Twitter merupakan simpul-simpulnya dan sisinya merupakan pengguna yang saling mengikuti satu sama lain. Selain itu graf juga dapat dipresentasikan pada jaringan di jalan raya dengan kota dimodelkan sebagai simpulnya (node).

Jalan yang saling menghubungkan antar kota atau daerah disebut sebagai sisi (edge). Dan panjang dari jalan tersebut merupakan bobot atau jaraknya yaitu (weight)[8].

Masalah mencari lintasan terpendek dalam graf adalah salah satu jenis persoalan optimasi. Graf yang digunakan untuk pencarian lintasan terpendek adalah graf berbobot (weighted graph), di mana setiap sisi diberi nilai atau bobot tertentu. Bobot pada sisi graf dapat merepresentasikan berbagai hal, seperti jarak antar kota, waktu pengiriman pesan, biaya pembangunan, dan lain-lain [9].

Dalam elemen graf G , didefinisikan sebagai pasangan himpunan (V, E) dan ditulis sebagai $G = (V, E)$. Di mana V adalah himpunan simpul (node) yang tidak kosong, dan E adalah himpunan sisi (edge) yang menghubungkan pasangan simpul. Node dalam graf dapat dilambangkan dengan huruf abjad, seperti $a, b, c, d, \dots, v, w, \dots$ atau dengan bilangan asli seperti $1, 2, 3, \dots$, atau kombinasi keduanya. Untuk sisi yang menghubungkan simpul u dengan v , notasinya ditulis sebagai $e = (u, v)$.

Pencari lintasan terpendek di dalam graf merupakan persoalan untuk mencari lintasan antara dua atau lebih simpul pada. Graf yang digunakan adalah graf berbobot (weighted graph), yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Bobot pada sisi graf dapat menyatakan jarak antar kota. Dalam hal ini bobot harus bernilai positif, walau dalam hal lain dapat bernilai negatif. Lintasan terpendek dengan verteks awal dan verteks tujuan didefinisikan sebagai lintasan terpendek dengan bobot minimum dan berupa lintasan sederhana (simple path). Kata "terpendek" pada persoalan lintasan terpendek berarti minimalisasi dari bobot pada suatu lintasan di dalam graf[10].

Terdapat beberapa jenis persoalan lintasan terpendek, antara lain:

- Lintasan terpendek antara dua buah simpul tertentu.
- Lintasan terpendek antara semua pasangan simpul.
- Lintasan terpendek dari simpul tertentu ke semua simpul yang lain.
- Lintasan terpendek antara dua buah simpul yang melalui

3. METODE PENELITIAN

Penelitian ini dilakukan dengan pendekatan simulasi berbasis estimasi, di mana algoritma Greedy diterapkan untuk mencari rute terdekat dari Universitas Negeri Medan (Unimed) menuju beberapa Stasiun Pengisian Bahan Bakar Umum (SPBU) di sekitarnya. Metode penelitian ini mencakup beberapa tahap utama, yaitu pengumpulan data estimasi lokasi, penerapan algoritma Greedy, serta analisis hasil yang didapatkan dari simulasi. Berikut adalah rincian dari metode yang digunakan dalam penelitian ini:

3.1. Desain Penelitian

Penelitian ini menggunakan desain kualitatif berbasis simulasi untuk menerapkan algoritma *Greedy* dalam menentukan rute terdekat. Penelitian difokuskan pada simulasi berbasis perkiraan tanpa penggunaan data geografis aktual. Simulasi ini bertujuan untuk mengestimasi rute terdekat berdasarkan asumsi jarak antara titik-titik lokasi yang direpresentasikan secara dua dimensi.

3.2. Pengumpulan Data

Pengumpulan data dilakukan secara perkiraan dengan asumsi beberapa titik lokasi, yaitu Universitas Negeri Medan sebagai titik awal, dan beberapa SPBU di sekitar kampus sebagai titik tujuan. Data yang dikumpulkan dalam penelitian ini tidak menggunakan data geografis (koordinat GPS) aktual, namun dibuat berdasarkan posisi relatif dalam ruang dua dimensi. Koordinat lokasi diperkirakan dan dimasukkan ke dalam sistem simulasi untuk memungkinkan algoritma *Greedy* menghitung jarak antara masing-masing titik.

3.3. Penerapan Algoritma *Greedy*

Tahap selanjutnya adalah penerapan algoritma *Greedy* untuk menentukan rute terdekat dari Unimed menuju beberapa SPBU yang tersedia. Langkah-langkah penerapan algoritma *Greedy* dalam simulasi adalah sebagai berikut:

- Langkah Awal: Dimulai dari titik awal, yaitu Unimed.
- Pemilihan Titik Terdekat: Algoritma menghitung jarak dari lokasi saat ini ke semua titik yang belum dikunjungi (dalam hal ini SPBU). Algoritma memilih titik dengan jarak terpendek.
- Pindah ke Titik Baru: Setelah memilih titik terdekat, algoritma pindah ke titik tersebut dan kembali menghitung jarak dari lokasi saat ini ke titik-titik yang tersisa.
- Pengulangan: Proses ini diulang hingga semua titik tujuan (SPBU) telah dikunjungi.

Penerapan algoritma ini dilakukan menggunakan bahasa pemrograman Python untuk mensimulasikan perhitungan dan visualisasi rute.

3.4. Simulasi dan Visualisasi

Hasil dari algoritma *Greedy* divisualisasikan dalam bentuk peta perkiraan. Peta ini memberikan representasi grafis tentang bagaimana algoritma *Greedy* memilih rute terdekat berdasarkan posisi relatif dari Unimed dan SPBU-SPBU yang tersedia. Visualisasi ini bertujuan untuk memperjelas pola pergerakan yang dihasilkan oleh algoritma. Selain itu, simulasi ini juga menunjukkan bagaimana algoritma *Greedy* dapat memberikan solusi cepat, meskipun mungkin tidak optimal secara global.

3.5. Analisis Data

Setelah simulasi dilakukan, data rute yang dihasilkan dianalisis untuk mengevaluasi keefektifan

algoritma *Greedy* dalam menyelesaikan masalah optimasi rute terdekat. Analisis ini meliputi evaluasi terhadap jarak total yang ditempuh dan waktu komputasi yang dibutuhkan untuk menghasilkan solusi. Selain itu, kelemahan dari algoritma *Greedy*, seperti kecenderungannya untuk tidak memberikan solusi global yang optimal, juga dianalisis dan dibahas dalam konteks hasil simulasi.

3.6. Evaluasi dan Kesimpulan

Tahap akhir dari metode penelitian ini adalah evaluasi keseluruhan dari hasil simulasi. Hasil yang diperoleh dari penerapan algoritma *Greedy* dibandingkan dengan teori dan referensi terkait masalah optimasi rute. Evaluasi dilakukan untuk menentukan sejauh mana algoritma ini mampu memberikan solusi yang efisien dalam konteks simulasi, serta bagaimana metode ini dapat diterapkan atau dikembangkan lebih lanjut dalam konteks nyata.

3.7. Instrumen Penelitian

Dalam penelitian ini, alat utama yang digunakan untuk simulasi adalah:

- Python: Digunakan untuk penerapan algoritma *Greedy*, simulasi, dan visualisasi.

Penelitian ini dilakukan secara perkiraan tanpa data geografis aktual, sehingga hasil yang diperoleh tidak merepresentasikan kondisi lapangan secara nyata. Selain itu, algoritma *Greedy* yang digunakan dalam penelitian ini hanya mempertimbangkan jarak terdekat secara lokal, sehingga hasil optimasi mungkin tidak selalu optimal secara global. Meski demikian, penelitian ini tetap memberikan kontribusi teoritis dan praktis dalam memahami penerapan algoritma *Greedy* dalam optimasi rute sederhana.

4. HASIL DAN PEMBAHASAN

4.1. Pemaparan Algoritma

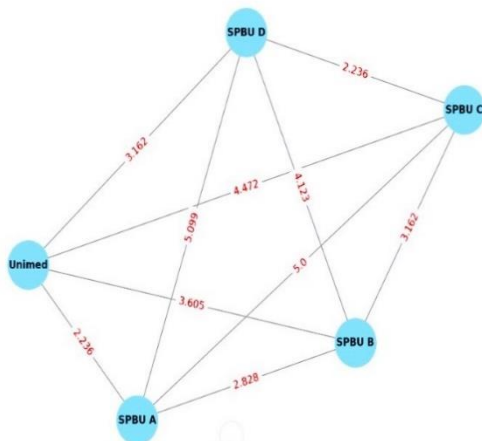
Algoritma *Greedy* diterapkan untuk menentukan rute terdekat dari Universitas Negeri Medan menuju SPBU di sekitarnya. Hasil menunjukkan bahwa algoritma berhasil menentukan rute terdekat dengan langkah-langkah yang efisien. Rute yang dihasilkan oleh algoritma *Greedy* memprioritaskan SPBU yang memiliki jarak paling dekat dari lokasi awal.

Algoritma *Greedy* memiliki kecepatan komputasi yang tinggi karena pada setiap iterasi, hanya jarak terdekat yang dihitung dari satu titik ke titik berikutnya [11].

Namun, kelemahan dari pendekatan ini adalah kemungkinan algoritma *Greedy* tidak selalu menghasilkan solusi global yang paling optimal karena sifatnya yang hanya mempertimbangkan solusi lokal terbaik di setiap langkah.

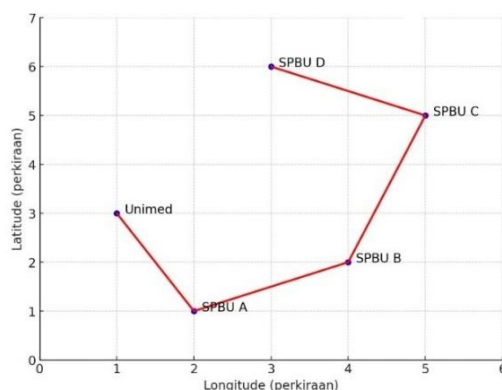
Berikut adalah gambar visualisasi algoritma *Greedy* untuk mencari rute terdekat dari Universitas Negeri Medan (Unimed) ke beberapa SPBU di sekitarnya. Setiap node mewakili lokasi (Unimed dan SPBU), sedangkan angka di sepanjang tepi

menggambarkan jarak antara dua lokasi. Algoritma *Greedy* akan memilih jalur dengan jarak terdekat pada setiap langkah.



Gambar 1. Visualisasi Algoritma Greedy Estimasi Pencarian Rute Terdekat dari Unimed ke SPBU

Berikut adalah peta perkiraan yang menggambarkan rute terdekat dari Universitas Negeri Medan (Unimed) ke beberapa SPBU menggunakan algoritma *Greedy*. Garis merah menunjukkan jalur yang diambil oleh algoritma *Greedy*, dimulai dari Unimed menuju SPBU A, SPBU B, dan seterusnya, berdasarkan jarak terdekat di setiap langkah.



Gambar 2. Perkiraan Peta Rute Terdekat dari Unimed ke SPBU Terdekan (Algoritma *Greedy*)

4.2. Implementasi dan Pembahasan

Pada hasil visualisasi peta di atas, kita menggunakan skala perkiraan dengan rentang sumbu X (longitude) dari 0 hingga 6 dan sumbu Y (latitude) dari 0 hingga 7. Meskipun peta ini tidak didasarkan pada data geografis sebenarnya seperti latitude dan longitude, skala ini tetap berfungsi untuk memberikan gambaran abstrak mengenai posisi relatif antara Universitas Negeri Medan (Unimed) dan SPBU di sekitarnya. Peta ini bertujuan untuk menggambarkan prinsip dasar algoritma *Greedy* dalam memilih jalur terdekat antara lokasi-lokasi yang tersedia.

Dimensi peta ini berfungsi untuk merepresentasikan wilayah sekeliling Unimed secara konseptual. Sumbu X menggambarkan longitude

(arah barat-timur), sedangkan sumbu Y menggambarkan latitude (arah utara-selatan). Dalam konteks geografis sebenarnya, posisi longitude dan latitude akan menggambarkan jarak dalam satuan derajat atau meter, yang bisa digunakan untuk menghitung jarak sebenarnya antar lokasi. Pada peta abstrak ini, dimensi dalam satuan yang lebih kecil (0-6 pada sumbu X dan 0-7 pada sumbu Y) disesuaikan untuk visualisasi yang lebih mudah dan dipahami oleh pembaca.

Selain itu, penting untuk diperhatikan bahwa meskipun dalam skala visual peta terlihat sederhana, setiap titik pada peta mewakili lokasi yang cukup jauh jika diterapkan pada konteks geografis asli. Dalam dunia nyata, jarak antara Unimed dan SPBU terdekat mungkin bisa mencapai beberapa kilometer. Oleh karena itu, skala peta ini harus dilihat sebagai representasi relatif, dan bukan sebagai pengukuran akurat dari jarak sebenarnya. Penggunaan skala perkiraan seperti ini umum diterapkan dalam penelitian yang bertujuan untuk menjelaskan algoritma, tanpa menitikberatkan pada ketepatan geografis.

Perbedaan jarak yang terlukis di peta abstrak ini juga memperlihatkan bagaimana algoritma *Greedy* memilih jalur berdasarkan jarak lokal terdekat, yang ditampilkan dalam bentuk garis merah yang menghubungkan titik-titik. Sumbu X dan Y menunjukkan perbedaan jarak antar lokasi, di mana algoritma *Greedy* melakukan pemilihan SPBU yang jaraknya paling pendek dari lokasi awal atau lokasi sebelumnya. Jika algoritma lain, seperti Dijkstra atau A*, digunakan dengan mempertimbangkan variabel tambahan, mungkin kita akan melihat rute yang berbeda atau bahkan lebih pendek secara keseluruhan.

Selanjutnya, dimensi peta abstrak ini juga memudahkan kita untuk memvisualisasikan bagaimana jalur rute dioptimalkan secara lokal. Peta sederhana ini menunjukkan bagaimana *Greedy* hanya fokus pada jarak terdekat pada setiap langkahnya. Penggunaan skala visual seperti ini membantu pembaca memahami pola pemilihan jalur yang dilakukan oleh algoritma, yang berdasarkan perbandingan jarak relatif antara satu titik dengan titik lainnya. Dimensi yang dipilih di sini juga menggambarkan bahwa solusi yang diperoleh oleh *Greedy* tidak memerlukan pemetaan atau penghitungan jarak yang kompleks, sehingga lebih cocok untuk penggunaan pada situasi di mana waktu komputasi menjadi prioritas.

Dalam konteks yang lebih luas, skala ini juga menggambarkan batasan dari algoritma *Greedy* dalam menyelesaikan masalah pencarian rute terdekat. Pada peta ini, kita dapat melihat bahwa jalur yang dipilih selalu didasarkan pada jarak langsung terdekat, tanpa memperhitungkan jalur yang mungkin lebih efisien secara keseluruhan. Jika kita memperluas dimensi peta ini atau memasukkan lebih banyak titik SPBU yang lebih jauh, kita mungkin

akan melihat bahwa *Greedy* masih akan cenderung memilih jalur yang terdekat secara lokal meskipun jalur tersebut mungkin tidak optimal jika dilihat dari keseluruhan rute. Dimensi yang lebih luas ini juga akan menunjukkan bagaimana *Greedy* akan tetap bekerja dalam waktu komputasi yang cepat meskipun pada kasus yang lebih kompleks.

4.3. Pola Pergerakan dan Rute yang Dipilih

Pada peta hasil visualisasi, algoritma *Greedy* menunjukkan pola pergerakan yang sederhana namun efektif dalam mencari rute terdekat. Dimulai dari titik awal, yaitu Universitas Negeri Medan (Unimed), algoritma *Greedy* secara bertahap memilih jalur menuju SPBU yang terdekat dari posisi saat ini, tanpa mempertimbangkan jalur selanjutnya. Pada setiap langkah, algoritma hanya fokus pada solusi lokal (jarak terdekat dari posisi terakhir) tanpa memikirkan keseluruhan rute hingga semua SPBU tercapai. Pola ini konsisten dengan cara kerja algoritma *Greedy* yang selalu memilih solusi terbaik pada langkah tertentu tanpa memperhatikan langkah selanjutnya secara global.

4.4. Langkah Pertama: Dari Unimed ke SPBU A

Algoritma *Greedy* dimulai dari titik Universitas Negeri Medan sebagai lokasi awal. Dari beberapa SPBU yang tersedia, SPBU A dipilih sebagai tujuan pertama karena memiliki jarak terdekat dari Unimed. Ini adalah langkah yang tepat secara lokal, karena algoritma *Greedy* secara cepat menghitung jarak antara Unimed dan semua SPBU lainnya, kemudian memilih yang paling dekat dalam jarak Euclidean. Jalur ini ditandai dengan garis merah dari Unimed ke SPBU A.

4.5. Langkah Kedua: Dari SPBU A ke SPBU B

Setelah mencapai SPBU A, algoritma kembali menghitung jarak ke SPBU lainnya yang belum dikunjungi, yakni SPBU B, SPBU C, dan SPBU D. SPBU B dipilih sebagai tujuan berikutnya karena jaraknya lebih pendek dibandingkan SPBU lainnya dari SPBU A. Pada tahap ini, algoritma *Greedy* menunjukkan bagaimana ia memilih jalur terbaik pada setiap langkah, tetapi belum mempertimbangkan apakah pilihan tersebut optimal secara keseluruhan. Jalur dari SPBU A ke SPBU B adalah pilihan yang masuk akal pada langkah ini, karena meminimalkan jarak yang harus ditempuh dari titik sebelumnya.

4.6. Langkah Ketiga: Dari SPBU B ke SPBU C

Pada langkah ketiga, setelah mencapai SPBU B, algoritma *Greedy* kembali mengevaluasi jarak ke SPBU C dan SPBU D. SPBU C dipilih sebagai lokasi selanjutnya karena jaraknya lebih pendek dari SPBU B dibandingkan SPBU D. Ini merupakan contoh pola dasar algoritma *Greedy* dalam menentukan solusi optimal lokal. Algoritma ini hanya memperhatikan jarak terdekat dari SPBU B ke SPBU C dan tidak

mempertimbangkan bahwa SPBU D mungkin lebih efisien jika dilihat dari perspektif rute keseluruhan. Namun, untuk konteks masalah lokal seperti ini, keputusan *Greedy* tetap terlihat efektif.

4.7. Langkah Keempat: Dari SPBU C ke SPBU D

Pada tahap terakhir, algoritma *Greedy* hanya memiliki satu pilihan yang tersisa, yaitu SPBU D. Karena SPBU D adalah satu-satunya SPBU yang belum dikunjungi, maka algoritma secara otomatis memilih rute dari SPBU C ke SPBU D. Meskipun *Greedy* telah membuat keputusan terbaik secara lokal di setiap langkah, rute ini mungkin tidak selalu optimal dalam konteks yang lebih luas. Namun, untuk masalah seperti mencari rute terdekat dengan hanya beberapa titik, pendekatan *Greedy* masih memberikan solusi yang dapat diterima.

4.8. Analisis Efisiensi dan Pengaruh Pola Pergerakan

Pola pergerakan yang ditampilkan dalam gambar ini menunjukkan efisiensi yang tinggi untuk masalah dengan skala kecil. Algoritma *Greedy* mengambil jalur paling sederhana dengan waktu komputasi yang minimal, karena pada setiap langkah hanya perlu mengevaluasi jarak terdekat dari lokasi saat ini ke beberapa titik yang tersisa. Meskipun tidak mempertimbangkan semua kemungkinan rute dalam satu kali perhitungan, algoritma *Greedy* tetap memberikan solusi cepat dan efisien dalam kondisi di mana solusi terbaik pada setiap langkah lokal lebih diutamakan daripada mencari solusi global optimal.

Algoritma *Greedy* juga mudah diterapkan karena kesederhanaannya. Sebagai contoh, dalam kasus pencarian rute terdekat di perkotaan seperti antara universitas dan SPBU, pola ini cocok diterapkan untuk pengambilan keputusan cepat dalam sistem navigasi sederhana. Sistem ini juga memiliki keuntungan dalam situasi di mana rute harus ditentukan dengan segera, tanpa memerlukan analisis mendalam atau komputasi berat untuk mencari solusi optimal secara keseluruhan.

Namun, penting untuk dicatat bahwa pola pergerakan algoritma *Greedy* ini juga memperlihatkan salah satu kelemahan utamanya: ia tidak selalu menghasilkan solusi terbaik secara global. Misalnya, meskipun jalur dari SPBU A ke SPBU B dipilih karena jaraknya paling pendek dalam konteks langkah tersebut, ada kemungkinan bahwa rute dari Unimed ke SPBU C lalu ke SPBU D akan lebih optimal secara keseluruhan. Dengan kata lain, pola ini menunjukkan bagaimana *Greedy* dapat melewatkan solusi terbaik jika terlalu fokus pada solusi lokal terbaik di setiap langkah.

Secara keseluruhan, pola pergerakan yang dipilih oleh algoritma *Greedy* pada visualisasi ini cukup efektif untuk situasi di mana jumlah titik yang harus dihitung relatif sedikit dan keputusan harus diambil dengan cepat. Namun, untuk masalah yang lebih kompleks, penggunaan algoritma lain yang

mempertimbangkan keseluruhan jalur, seperti algoritma Dijkstra atau A*, dapat memberikan solusi yang lebih baik dari segi optimasi global.

PYTHON :

```
import networkx as nx
import matplotlib.pyplot as plt

# Membuat graf dengan node sebagai lokasi dan edge sebagai jarak antar lokasi
G = nx.Graph()

# Node merepresentasikan lokasi (Unimed dan SPBU-SPBU)
locations = {
    'Unimed': (0, 0),
    'SPBU A': (1, 5),
    'SPBU B': (4, 3),
    'SPBU C': (6, 1),
    'SPBU D': (3, 6)
}

# Menambahkan node ke graf
for location, pos in locations.items():
    G.add_node(location, pos=pos)

# Menambahkan edge (jarak antar lokasi)
edges = [
    ('Unimed', 'SPBU A', 5.1),
    ('Unimed', 'SPBU B', 5.8),
    ('Unimed', 'SPBU C', 6.4),
    ('Unimed', 'SPBU D', 5.5),
    ('SPBU A', 'SPBU B', 3.0),
    ('SPBU A', 'SPBU C', 4.7),
    ('SPBU A', 'SPBU D', 2.5),
    ('SPBU B', 'SPBU C', 2.8),
    ('SPBU B', 'SPBU D', 4.0),
    ('SPBU C', 'SPBU D', 3.2)
]

for edge in edges:
    G.add_edge(edge[0], edge[1], weight=edge[2])

# Fungsi untuk algoritma Greedy
def greedy_shortest_path(G, start):
    visited = [start]
    total_distance = 0

    current_node = start
    while len(visited) < len(G.nodes):
        neighbors = [(neighbor, G[current_node][neighbor]['weight']) for neighbor in G.neighbors(current_node) if neighbor not in visited]
```

```
        next_node = min(neighbors, key=lambda x: x[1])
        visited.append(next_node[0])
        total_distance += next_node[1]
        current_node = next_node[0]

    return visited, total_distance

# Menjalankan algoritma Greedy dari Unimed
path, distance = greedy_shortest_path(G, 'Unimed')

print("Rute yang dipilih oleh algoritma Greedy:", path)
print("Jarak total:", distance)

# Visualisasi graf dan rute
pos = nx.get_node_attributes(G, 'pos')
labels = nx.get_edge_attributes(G, 'weight')

nx.draw(G, pos, with_labels=True, node_size=2000, node_color='lightblue', font_size=10, font_weight='bold')
nx.draw_networkx_edge_labels(G, pos, edge_labels=labels)

# Menyoroti rute yang dipilih
path_edges = [(path[i], path[i + 1]) for i in range(len(path) - 1)]
nx.draw_networkx_edges(G, pos, edgelist=path_edges, edge_color='r', width=2)

plt.show()
```

OUTPUT:

Rute yang dipilih oleh algoritma Greedy:
['Unimed', 'SPBU A', 'SPBU D', 'SPBU C', 'SPBU B']

Jarak total: 13.6

5. KESIMPULAN DAN SARAN

Penelitian ini menunjukkan bahwa algoritma Greedy dapat digunakan untuk menentukan rute terdekat dari Universitas Negeri Medan ke SPBU terdekat dengan hasil yang cepat dan efisien. Algoritma ini memilih jarak terpendek pada setiap langkah, tetapi hanya mempertimbangkan solusi lokal terbaik sehingga tidak selalu memberikan solusi optimal secara keseluruhan. Meskipun simulasi dilakukan tanpa data geografis aktual, algoritma ini tetap efektif dalam skenario navigasi sederhana.

Namun, algoritma Greedy memiliki keterbatasan dalam menyelesaikan masalah yang lebih kompleks karena tidak mempertimbangkan solusi global. Oleh karena itu, dalam kasus-kasus dengan lebih banyak variabel atau titik tujuan, algoritma yang lebih canggih seperti Dijkstra atau A* mungkin lebih tepat digunakan. Penelitian ini memberikan wawasan penting tentang penerapan algoritma Greedy dalam sistem navigasi sederhana serta potensi pengembangannya di masa depan.

Algoritma Greedy cocok digunakan untuk kasus sederhana dengan jumlah titik yang terbatas. Untuk kasus yang lebih kompleks, disarankan menggunakan algoritma yang lebih canggih seperti Dijkstra atau A* yang mempertimbangkan solusi global. Penggunaan data geografis yang aktual dapat meningkatkan akurasi hasil, sehingga disarankan untuk mengintegrasikan algoritma dengan data spasial nyata pada penelitian berikutnya. Kombinasi algoritma Greedy dengan algoritma lain dapat dipertimbangkan untuk menghasilkan solusi yang lebih optimal dan efisien dalam konteks rute yang lebih kompleks.

DAFTAR PUSTAKA

- [1] D. Yulia dan Rozali 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, hlm. 57, 2019.
- [2] R. F. Ginting, A. Situmorang, dan S. Manurung, "Sistem Informasi Gografis Pencarian Lokasi Sekolah Terdekat Pada Kota Medan Dengan Algoritma Greedy Berbasis Web," *Methodika : Jurnal Ilmiah Teknik Informatika*, vol. 2, no. 2, hlm. 85–91, 2022, [Daring]. Tersedia pada: <http://ojs.fikom-methodist.net/index.php/methodika>
- [3] B. Hadibrata dan S. Maudin, "PENCARIAN RUTE TERPENDEK MENUJU TEMPAT WISATA MENGGUNAKAN METODE ALGORITMA GREEDY PADA DINAS PEMUDA OLAHRAGA KEBUDAYAAN DAN PARIWISATA KOTA CIREBON," *Jurnal Ilmiah Indonesia*, vol. 5, no. 5, hlm. 26–35, 2020.
- [4] N. F. Lakutu, M. R. Katili, S. L. Mahmud, dan N. I. Yahya, "Algoritma Dijkstra dan Algoritma Greedy Untuk Optimasi Rute Pengiriman Barang Pada Kantor Pos Gorontalo," *Euler : Jurnal Ilmiah Matematika, Sains dan Teknologi*, vol. 11, no. 1, hlm. 55–65, Jun 2023, doi: 10.34312/euler.v11i1.18244.
- [5] A. Roihan, K. Nasution, dan Mhd. Z. Siambaton, "Implementasi Algoritma Greedy Kombinasi dengan Perulangan pada Aplikasi Penjadwalan Praktikum," *sudo Jurnal Teknik Informatika*, vol. 1, no. 2, hlm. 42–50, Jun 2022, doi: 10.56211/sudo.v1i2.8.
- [6] E. R. Ningrum, A. Sanwidi, R. Akbarita, dan M. N. H. Qomaruddin, "Optimasi Rute Pendistribusian Gas Elpiji Menggunakan Algoritma Floyd Warshall Dan Algoritma Greedy," *JURNAL ILMIAH MATEMATIKA DAN TERAPAN*, vol. 20, no. 1, hlm. 1–14, Jun 2023, doi: 10.22487/2540766x.2023.v20.i1.15568.
- [7] I. H. Santi dan D. Budianti, "PENERAPAN ALGORITMA GREEDY DALAM Mencari RUTE TERDEKAT LOKASI SPBU BERBASIS WEB," *Jurnal Penelitian Multidisiplin Ilmu*, vol. 2, no. 1, hlm. 1225–1234, 2023, [Daring]. Tersedia pada: <http://melatijournal.com/index.php/Metta>
- [8] H. P. Kekal, W. Gata, S. Nurdiani, A. Jati, A. J. S. Rini, dan D. S. Wita, "ANALISA PENCARIAN RUTE TERCEPAT MENUJU TEMPAT WISATA PULAU KUMALA KOTA TENGGARONG MENGGUNAKAN ALGORITMA GREEDY," *Jurnal Ilmiah Ilmu Komputer*, vol. 7, no. 1, hlm. 9–15, 2021.
- [9] D. Mahendra, Nuryanto, dan A. Burhanuddin, "SISTEM PENENTUAN JARAK TERDEKAT DALAM PENGIRIMAN DARAH DI PMI KOTA SEMARANG DENGAN METODE ALGORITMA GREEDY," *Jurnal Komtika-Komputasi dan Informatika*, vol. 136, no. 2, hlm. 136–142, 2019.
- [10] M. C. Bunaen, H. Pratiwi, dan Y. F. Riti, "PENERAPAN ALGORITMA DIJKSTRA UNTUK MENENTUKAN RUTE TERPENDEK DARI PUSAT KOTA SURABAYA KE TEMPAT BERSEJARAH," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 4, no. 1, hlm. 213–223, Jan 2022, doi: 10.47233/jteksis.v4i1.407.
- [11] A. D. Hartanto, A. S. Mandala, D. R. P.L., S. Aminudin, dan A. Yudirianto, "Implementasi Algoritma Dijkstra Pada Game Pacman," *CCIT Journal*, vol. 12, no. 2, hlm. 170–176, 2019, doi: 10.33050/ccit.v12i2.687