

Penerapan Algoritma Greedy

Penentuan Jarak terdekat dari Kost menuju Institut Teknologi Sumatera

Fadhilah Fauza Hamda 120140153

Program Studi Teknik Informatika

Jurusan Teknologi Produksi dan Industri

Institut Teknologi Sumatera, Jl. Terusan Ryacudu, Way Huwi, Kec. Jati Agung,

Kabupaten Lampung Selatan, Lampung 35365

E-mail : fadhilah.120140153@student.itera.ac.id

Abstract - Dalam kehidupan sehari-hari kita sering melakukan perjalanan suatu tempat ke tempat lain yg akan dituju. Oleh karena itu sangat diperlukan penentuan rute atau jarak terpendek antar satu tempat ke tempat lain yg akan menjadi tujuan. Rumusan masalah pada penelitian ini adalah bagaimana mencari rute terpendek dari Kost ke Kampus dari jarak yg akan ditempuh menggunakan algoritma Greedy. Terdapat beberapa alternatif rute yg bisa dilalui dari Kost ke Kampus Untuk menentukan rute yg paling optimal yaitu rute yg mempunyai jarak terpendek menggunakan pendekatan algoritma Greedy.

Keywords: Jarak, Dekat, Solusi, Greedy, Algoritma

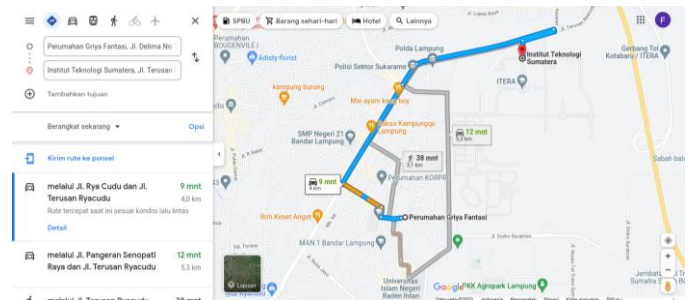
I. PENDAHULUAN

Latar belakang Masalah dengan algoritma greedy yaitu mencari jarak terpendek dari sebuah peta jalur perjalanan dapat direpresentasikan dengan menggunakan graph, spesifiknya *Directed Graph* (graph berarah). Maka dari itu, untuk menyelesaikan permasalahan jarak terpendek ini kita akan menggunakan struktur data graph untuk merepresentasikan peta berikut.

Untuk mencari rute tercepat atau lintasan terpendek dalam Algoritma Greedy dilakukan dengan rumus berikut:

1. Pertama lakukan periksa sisi yg berhubungan dengan sisi pertama (lokasi pertama). Pilih sisi yg memiliki bobot (jarak) terkecil dengan sisi berikutnya. Jarak tersebut merupakan menjadi rute terpendek pertama, dapat dimisalkan sebagai (L_1).
2. Selanjutnya menentukan rute terpendek berikutnya dapat diminalkan (D) dengan cara sebagai berikut :
 - a) Untuk mencari rute berikutnya dengan rumus $D(i) = L_1 + \text{bobot sisi berikutnya}$. Dijelaskan jarak pertama dijumlahkan dengan bobot berikutnya. Jika terdapat rute lainnya, maka lakukan penjumlahan jarak rute tersebut dengan jarak sebelumnya.
 - b) Pilih $D(i)$ yg memiliki jarak terkecil, kemudian jika terdapat rute lainnya lakukan perbandingan. Jika rute lainnya tersebut memiliki jarak lebih kecil daripada $D(i)$, maka rute tersebut merupakan rute yg akan diperhitungkan selanjutnya.

3. Untuk mencari lintasan terpendek berikutnya dapat dilakukan dengan menggunakan cara kedua di atas.. Algoritma greedy membentuk solusi langkah per langkah. Pendekatan yg digunakan di dalam algoritma greedy adalah membuat pilihan yg memberikan perolehan terbaik yaitu dengan membuat pilihan optimum pada setiap langkah dengan harapan bahwa sisanya mengarah ke solusi optimum secara keseluruhan.



Gambar : Lokasi Kost (Perumahan Griya Fantasi) ke ITERA

Untuk mencari jarak terpendek dari Kost ke Kampus ITERA, sebuah algoritma greedy akan menjalankan langkah-langkah seperti berikut:

1. Kunjungi satu titik pada graph, dan ambil seluruh titik yg dapat dikunjungi dari titik sekarang.
2. Cari lokal maksimum ke titik selanjutnya.
3. Tandai graph sekarang sebagai graph yg telah dikunjungi, dan pindah ke lokal maksimum yg telah ditentukan.
4. Kembali ke langkah 1 sampai titik tujuan didapatkan.

Tujuan penelitian ini adalah sebagai berikut:

1. Mengetahui proses algoritma Greedy dalam menentukan rute terpendek dari kost penulis ke Kampus ITERA

2. Mengetahui rute terpendek dari kost penulis ke Kampus ITERA

Metode penelitian yg digunakan oleh peneliti dalam penelitian ini yaitu menggunakan pendekatan kualitatif dengan metode deskriptif. Pada penelitian ini penulis menggunakan metode sesuai metode yg ada pada algoritma greedy dan alur seperti berikut :

1. Penulis melakukan reset terhadap metode yg akan digunakan pada penelitian, yaitu algoritma greedy
2. Melakukan pemecahan masalah dengan metode yg digunakan
3. Mempelajari identifikasi masalah yg diambil dalam penelitian
4. Penentuan rute awal untuk menghasilkan data dengan algoritma greedy
5. Melakukan analisis kembali dan menyimpulkan hasil dari keseluruhan penelitian yg dilakukan.

Dalam sistem pencarian tempat wisata di Palabuhanratu penulis mengimplementasikan algoritma greedy untuk pencarian jarak terdekat. Dan dilakukan analisa data untuk menyimpulkan hasil dari penelitian tersebut. Dimana data yg telah dipastikan valid dan akurat, selanjutnya akan dikumpulkan dan dikelompokkan untuk dianalisa menggunakan tahap analisa data dengan algoritma greedy.

Berikut beberapa tahapannya :

1. Pengelompokan data : Dalam tahapan ini data yg yg telah terkumpul akan dikelompokkan untuk dilakukan tahapan terhadap langkah-langkah perhitungan algoritma greedy untuk menghasilkan rute alternatif tercepat.
2. Penerapan data : Dalam proses ini data yg siap akan diaplikasikan untuk diperhitungkan dalam algoritma greedy.

II. DASAR TEORI

A. Penentuan Jalur Terpendek

penyelesaian masalah dengan algoritma greedy yaitu mencari jarak terpendek dari sebuah peta jalur perjalanan dapat direpresentasikan dengan menggunakan graph, spesifiknya Directed Graph (graph berarah). Maka dari itu, untuk menyelesaikan permasalahan jarak terpendek ini kita akan menggunakan struktur data graph untuk merepresentasikan peta.

B. Pengertian Algoritma

Algoritma adalah metode efektif yg diekspresikan sebagai rangkaian terbatas. Algoritma juga merupakan kumpulan perintah untuk menyelesaikan suatu masalah. Perintah-perintah ini dapat diterjemahkan secara bertahap dari awal hingga akhir. Masalah tersebut dapat berupa apa saja, dengan syarat untuk setiap permasalahan

memiliki kriteria kondisi awal yg harus dipenuhi sebelum menjalankan sebuah algoritma. Algoritma juga memiliki pengulangan proses (iterasi), dan juga memiliki keputusan hingga keputusan selesai.

Sifat –sifat Algoritma => sifat utama dari algoritma adalah sebagai berikut:

1. Input: Algoritma memiliki input atau kondisi awal sebelum implementasinya, yg dapat berupa nilai variabel yg diambil dari set khusus.
2. Output : suatu algoritma akan menghasilkan output setelah implementasinya, atau algoritma tersebut akan mengubah kondisi awal menjadi kondisi akhir, di mana nilai output diperoleh dari nilai input yg telah diproses melalui algoritma.
3. Definiteness: langkah-langkah yg dituliskan dalam algoritma terdefinisi dengan jelas jadi mudah dilakukan oleh pengguna algoritma.
4. Finiteness : Setiap algoritma harus memberikan hasil akhir atau keluaran setelah sejumlah langkah yg terbatas dilakukan terhadap setiap kondisi awal atau masukan yg diberikan.
5. Effectiveness: Setiap langkah dalam algoritma dapat dilakukan dalam suatu selang waktu tertentu sehingga pada akhirnya diperoleh solusi yg sesuai yg diharapkan.
6. Generality : langkah-langkah algoritma berlaku untuk setiap set input yg sesuai dengan masalah yg diberikan, bukan hanya untuk set tertentu

C. Algoritma greedy

Strategi algoritma greedy adalah strategi algoritma yg digunakan untuk menyelesaikan persoalan optimasi, yaitu persoalan yg bertujuan memaksimalkan atau meminimalkan suatu satuan. Algoritma greedy membentuk solusi secara langkah per langkah. Pada setiap langkah terdapat banyak variabel dan pilihan yg harus dipertimbangkan dan akan diambil pilihan terbaik yg didapatkan tanpa mempertimbangkan konsekuensinya pada langkah-langkah berikutnya. Prinsip utama strategi algoritma greedy adalah dengan dipilihnya hasil optimum pada setiap langkah, disebut optimum lokal, akan dihasilkan hasil optimal atau optimum global. Setiap algoritma greedy memiliki elemen-elemen sebagai berikut

1. Himpunan Kandidat (C), berisi kandidat yg akan dipilih pada setiap langkah.
2. Himpunan Solusi (S), berisi kandidat terbaik dari setiap langkah yg dipilih sebagai solusi.
3. Fungsi solusi, menentukan apakah himpunan kandidat yg dipilih sudah memberikan solusi.
4. Fungsi seleksi, menentukan kandidat yg akan dipilih dengan suatu strategi greedy tertentu.
5. Fungsi kelayakan, menentukan apakah kandidat yg dipilih dapat dimasukkan ke dalam himpunan solusi.
6. Fungsi objektif, memaksimalkan atau meminimumkan.

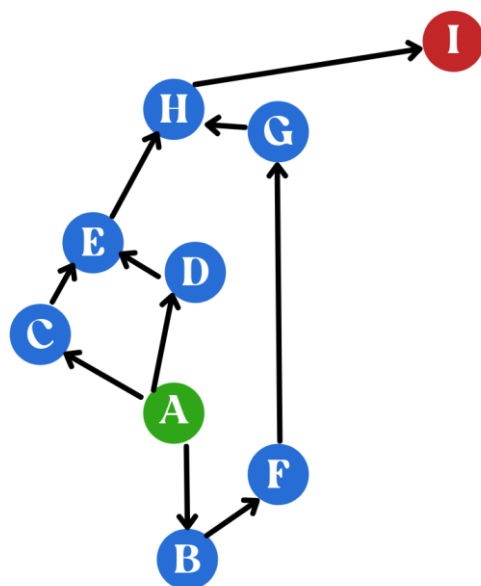
Algoritma greedy juga memecahkan masalah langkah demi langkah, pada setiap langkah:

1. mengambil pilihan yg terbaik yg dapat diperoleh saat itu
2. berharap bahwa dengan memilih optimum lokal pada setiap langkah akan mencapai optimum global. Algoritma greedy mengasumsikan bahwa optimum lokal merupakan bagian dari optimum global.

Contoh Algoritma Greedy

Algoritma greedy dapat digunakan untuk memecahkan masalah penukaran uang. Masalah penukaran uang adalah salah satu contoh soal klasik dari algoritma greedy dengan persoalan sebagai berikut, diberikan sekumpulan pecahan uang misal 1, 5, 10, dan 25 maka jika diberikan sebuah jumlah 35 maka bagaimana cara menukarkan uang berjumlah 35 sedemikian sehingga jumlah koin/lembar uang yg digunakan seminimal mungkin. Pendekatan yg dapat digunakan adalah mengurutkan pecahan dengan urutan paling besar ke paling kecil, kemudian melakukan iterasi untuk memilih apakah pecahan merupakan bagian dari himpunan solusi, berikut pemecahan masalah penukaran uang dengan strategi greedy

1. Langkah 1:
 - Pecahan = [25, 10, 5, 1]
 - Sisa uang = 35
 - Pilih 25, karena sisa uang dikurangi 25 masih lebih besar dari 0
2. Langkah 2:
 - Pecahan = [25, 10, 5, 1] \circ Sisa uang = $35 - 25 = 10$
 - 25 tidak bisa dipilih lagi karena akan menghasilkan nilai negatif, maka pilih 10
3. Selesai, total koin yg digunakan adalah 2



Gambar : Contoh Graph Berarah dari Titik A ke I

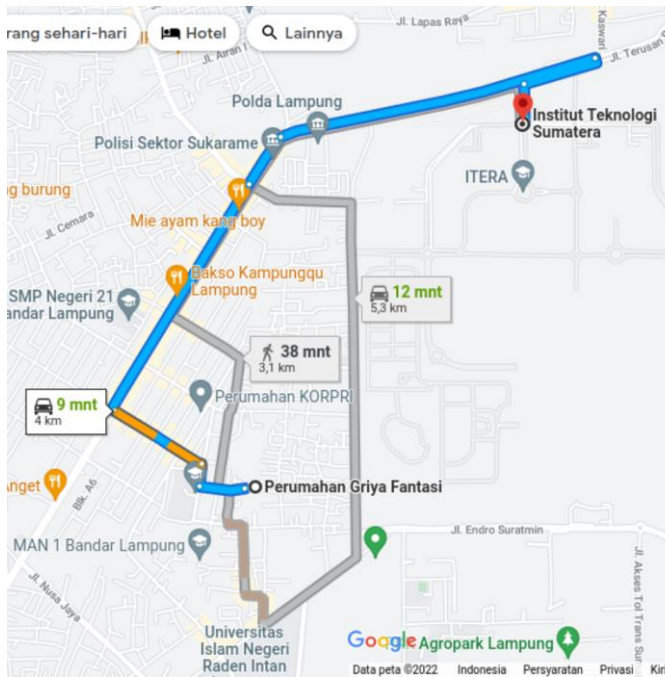
Untuk mempelajari Graf secara lebih, perlu diketahui beberapa terminologi sebagai berikut:

1. Bertetangga (Adjacent) Dua buah simpul pada grafik tidak berarah jika keduanya terhubung disebut bertetangga.
2. Bersisian (Incident) Untuk setiap sisi $e = (v_j, v_k)$, sisi e seharusnya hidup berdampingan dengan simpul v_j dan v_k .
3. Simpul terkecil (Isolated Vertex) Node yg tidak memiliki sisi berdampingan dalam arah node jarak jauh.
4. Graph Kosong (Null Graph) Grafik yg himpunan sisinya merupakan himpunan kosong.
5. Derajat (Degree) Tingkat simpul adalah jumlah sisi yg memotongnya.
6. Lintasan (Path) Jalur yg panjangnya n dari peralihan awal menuju pergantian v_k dengan melewati peralihan sisi dan pergantian penuh..
7. Siklus/Sirkuit (Cycle/Circuit) Jalur yg dimulai dan berakhir pada simpul yg sama.
8. Terhubung (Connected) Grafik berorientasi adalah grafik terhubung jika, untuk setiap simpul grafik, ada jalur ke simpul ini.
9. Upagraf (Subgraph) Upagraf merupakan bagian dari suatu grafik (subset suatu grafik).
10. Upagraf Merentang (Spanning Subgraph) Sebuah upagraf yg berisi semua simpul dari grafik utama.
11. Cut-Set/Cut-Set grafik adalah jika anggota tertentu dari set tepi dihapus, yg mencegah grafik agar tidak terhubung..
12. Graph Berbobot (Weighted Graph) Graf Berbobot adalah grafik yg setiap sisinya diberikan nilai / bobot (biaya).

III. PENGUJIAN DAN ANALISIS

Pengujian masalah dengan algoritma greedy yaitu mencari jarak terpendek dari kost Penulis di Perumahan Griya Fantasi, Jl. Delima No.19, Harapan Jaya, Kec. Sukarame menuju Kampus Institut Teknologi Sumatera, Jl. Terusan Ryacudu, Way Huwi, Kec. Jati Agung.

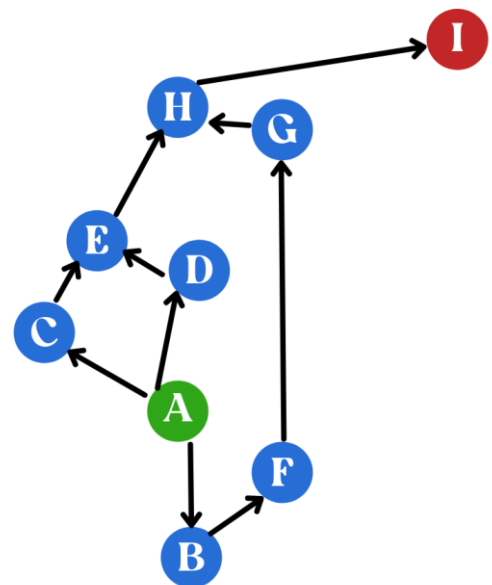
- Misalkan kita ingin bergerak dari titik A ke titik I, dan kita telah menemukan beberapa jalur dari peta:



Dari peta yg ditampilkan di atas, dapat dilihat bahwa terdapat beberapa jalur dari titik A ke titik I. Sistem peta pada gambar secara otomatis telah memilih jalur terpendek (berwarna biru). Kita akan mencoba mencari jalur terpendek juga, dengan menggunakan algoritma greedy.

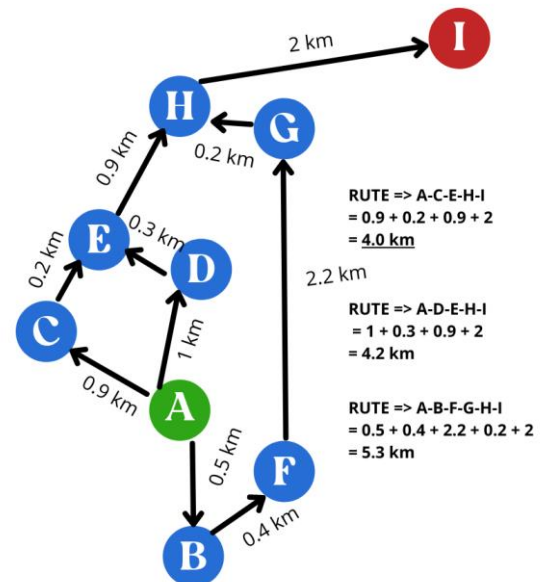
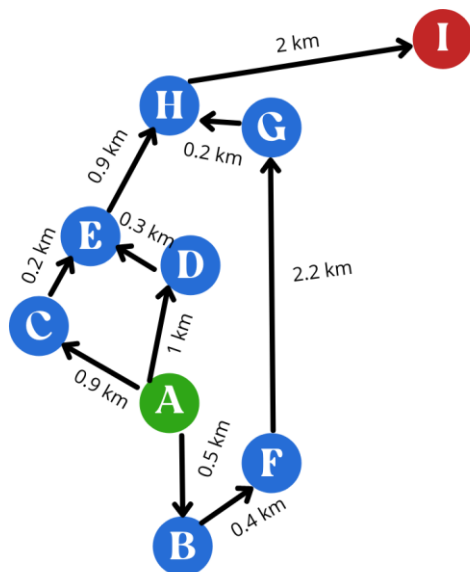
Dari gambar di atas, kita dapat melihat bagaimana sebuah peta jalur perjalanan dapat direpresentasikan dengan menggunakan graph, spesifiknya Directed Graph (graph berarah). Maka dari itu, untuk menyelesaikan permasalahan jarak terpendek ini kita akan menggunakan struktur data graph untuk merepresentasikan peta. Berikut adalah graph yg akan digunakan:

- Graph Berarah dari Titik A ke I :



- Graph Sederhana dari Titik A ke I :

- Graph Berarah Beserta Jarak Masing-Masing Titik dari Titik A ke I :



- Untuk mencari jarak terpendek dari A ke I, sebuah algoritma greedy akan menjalankan langkah-langkah seperti berikut:
 - Kunjungi satu titik pada graph, dan ambil seluruh titik yg dapat dikunjungi dari titik sekarang.
 - Cari local maximum ke titik selanjutnya.
 - Tandai graph sekarang sebagai graph yg telah dikunjungi, dan pindah ke local maximum yg telah ditentukan.
 - Kembali ke langkah 1 sampai titik tujuan didapatkan.

IV. KESIMPULAN

Algoritma greedy adalah algoritma yg berguna untuk menentukan hasil optimal pada setiap langkah, optimum lokal dengan harapan akan menghasilkan optimum global atau hasil paling optimal, namun pada prakteknya tidak semua strategi greedy selalu menghasilkan optimum global. Hal Tersebut diakibatkan karena fungsi seleksi yg memiliki kekurangan untuk kasus tertentu. Dengan menggunakan algoritma greedy pada graph di atas, hasil akhir yg akan didapatkan sebagai jarak terpendek adalah **A-C-E-H-I**. Algoritma greedy memang tidak selamanya memberikan solusi yg optimal, dikarenakan pencarian local maximum pada setiap langkahnya, tanpa memperhatikan solusi secara keseluruhan.

UCAPAN TERIMA KASIH

Rasa syukur saya panjatkan kepada Tuhan yg Maha Esa karena rahmat-Nya sehingga penulis dapat menyelesaikan makalah ini dengan baik dan lengkap. Selain itu, penulis mengucapkan terima kasih kepada bapak Imam Ekowicaksono S.Si., M.Si selaku dosen pengampu kelas IF2211 Strategi Algoritma yg sudah dengan sabar mengajar penulis selama satu semester serta orang tua penulis yg tidak henti-hentinya memberi dukungan penuh terhadap studi penulis. Dengan makalah ini, penulis berharap agar dapat memberikan manfaat dan wawasan baru bagi para pembaca.

REFERENSI

- [1] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf) - diakses pada tanggal 28 Maret 2022
- [2] <https://www.scribd.com/doc/38875940/Metode-Pencarian-Lintasan-Terpendek-Dalam-Graf> - diakses pada tanggal 28 Maret 2022
- [3] <https://jurnal.habi.ac.id/index.php/Info/article/view/67> - diakses pada tanggal 28 Maret 2022
- [4] <https://docplayer.info/44961542-Penggunaan-algoritma-greedy-dalam-penentuan-rute-wisata.html> - diakses pada tanggal 28 Maret 2022
- [5] <https://wahidari.github.io/Greedy-Shortest-Path/> - diakses pada tanggal 28 Maret 2022
- [6] <http://repository.uinsu.ac.id/9724/1/PDF%20-%20SKRIPSI%20-%20ILMU%20KOMPUTER%20-%20TAING%20SAPITRI%20NURDIANTI.pdf> - diakses pada tanggal 29 Maret 2022

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yg saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Lampung Selatan, 28 Maret 2022



Fadhilah Fauza Hamda - 120140153