

PENCARIAN RUTE TERPENDEK MENGGUNAKAN ALGORITMA GREEDY

Enty Nur Hayati¹, Antoni Yohanes²

^{1,2}Program Studi Teknik Industri, Universitas Stikubank Semarang

Jl. Trilomba Juang No 1, Semarang

Email:enty_nur@yahoo.co.id, antoniyohanes@yahoo.co.id

Abstrak

Dalam kehidupan sehari-hari sering dilakukan perjalanan suatu tempat ke tempat lain yang akan dituju. Oleh karena itu sangat diperlukan penentuan rute terpendek antar satu tempat ke tempat lain yang akan menjadi tujuan. Rumusan masalah pada penelitian ini adalah bagaimana mencari rute terpendek dari node A (kecamatan Ngaliyan) ke node L (kecamatan Sampangan) dari jarak yang akan ditempuh menggunakan algoritma Greedy. Terdapat beberapa alternatif rute yang bisa dilalui dari node A menuju node L. Untuk menentukan rute yang paling optimal yaitu rute yang mempunyai jarak terpendek menggunakan pendekatan algoritma Greedy. Algoritma Greedy adalah algoritma yang memecahkan masalah langkah demi langkah dan merupakan salah satu metode dalam masalah optimasi. Pendekatan yang digunakan di dalam algoritma Greedy adalah membuat pilihan yang terlihat memberikan perolehan terbaik yaitu dengan membuat pilihan optimum local pada setiap langkah dan diharapkan akan mendapatkan solution optimum global. Rute dari node A ke node L yang paling optimal dengan jarak terpendek adalah rute $A \rightarrow B \rightarrow D \rightarrow G \rightarrow I \rightarrow K \rightarrow J \rightarrow L$ dengan jarak 12 km.

Kata kunci : *algoritma greedy, optimasi, rute terpendek*

1. PENDAHULUAN

Pada kota besar, transportasi adalah persoalan penting bagi masyarakat kota yang dinamis. Luasnya sebuah kota serta banyaknya jalan raya seringkali menyulitkan seseorang untuk mencari rute optimum, baik dari segi jarak maupun biaya yang dikeluarkan untuk berpergian dari satu kota ke kota lain. Pada akhir-akhir ini pencarian rute optimum menjadi masalah yang semakin penting dipicu oleh kenaikan harga bahan bakar yang hampir naik dua kali lipat, sehingga orang berusaha menempuh perjalanan secepat mungkin untuk dapat sampai ke kota tujuan sehingga tidak ada biaya yang terbuang untuk masalah transportasi dari satu kota ke kota lainnya. Untuk dapat memilih rute yang optimum, maka orang harus mengetahui jarak antar kota dan juga keadaan alam dari rute itu. Kemudian dipilihlah jalur terpendek dari kota awal ke kota tujuan. Tetapi hal ini seringkali tidak membantu karena banyaknya jalan yang ada sehingga menyebabkan banyaknya pilihan jalur yang dapat ditempuh. Tujuan rute optimum adalah mendapatkan jarak yang optimal maupun biaya yang optimal untuk menempuh perjalanan dari kota asal ke kota tujuan. Penulis mengamati bahwa metode yang digunakan orang-orang pada umumnya terutama perusahaan bidang ekspedisi untuk mencari rute optimum tersebut belum cukup memuaskan.

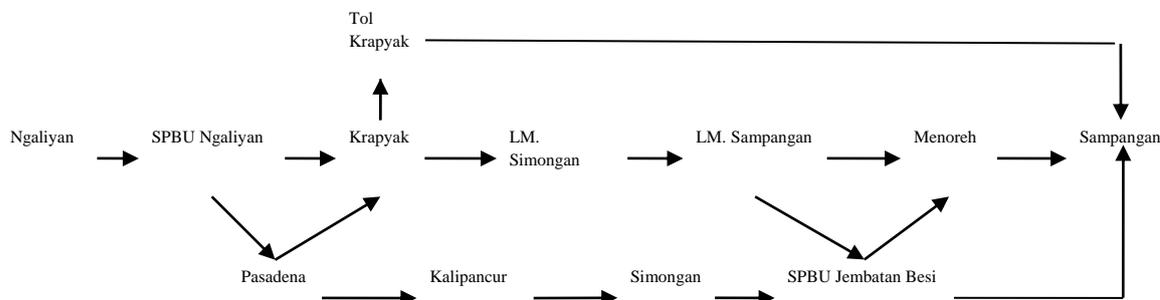
Makalah ini memaparkan solusi yang lebih optimal untuk mencari rute yang terpendek dari Kecamatan Ngaliyan ke Kecamatan Sampangan di Kabupaten Semarang dengan menggunakan metode Graf dan algoritma *Greedy*.

1.1. Latar Belakang

Kemacetan yang sering terjadi selama perjalanan, sering mengganggu kegiatan kita sehari-hari. Setiap manusia ingin sampai ke tujuan dengan tepat waktu. Tetapi, sering kali kemacetan menyebabkan keinginan manusia terganggu. Oleh karena itu, dibutuhkan suatu cara untuk menaggulangi gangguan tersebut. Untuk mencapai suatu tempat dengan waktu yang lebih cepat, kita akan mencari lintasan terpendek dari tempat asal ke tempat tujuan. Lintasan terpendek ini juga memperhitungkan waktu-waktu dimana kemacetan sering terjadi. Demikian juga yang terjadi pada lintasan yang dilewati dari Kecamatan Ngaliyan ke Kecamatan Sampangan, terutama pada waktu-waktu “sibuk”.

1.2. Rumusan Masalah

Rumusan masalah pada makalah ini adalah bagaimana menentukan rute optimum dari Kecamatan Ngaliyan ke Kecamatan Sampangan dengan menggunakan Algoritma *Greedy* sehingga mempunyai jarak terpendek.



Gambar 1. Rute yang dilewati dari kecamatan Ngaliyan ke kecamatan Kendeng

1.3. Tujuan Penulisan Makalah

Tujuan penulisan makalah ini adalah untuk melakukan eksplorasi terhadap metode dan masalah yang dipelajari dalam pengembangan serta menyebarkan aplikasi yang mendukung materi optimasi, sebagai media untuk berbagi informasi hasil-hasil pemikiran dan penelitian dan sebagai referensi untuk penyelesaian masalah rute (lintasan) terpendek supaya bisa diperoleh hasil yang optimal.

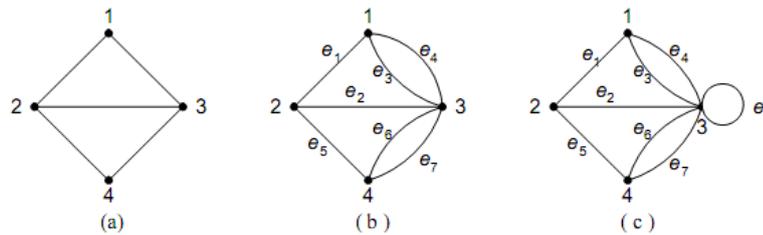
2. METODOLOGI

Dalam menentukan rute yang paling optimal ini, penulis melakukan langkah-langkah yaitu menggambarkan rute-rute (lintasan) yang dilalui dengan diagram jaringan menggunakan teori graf kemudian mencari rute (lintasan) yang paling optimal dengan menggunakan algoritma *greedy*.

2.1. Teori Graf

Graf merupakan suatu cabang ilmu yang memiliki banyak terapan. Banyak sekali struktur yang bisa direpresentasikan dengan graf, dan banyak masalah yang bisa diselesaikan dengan bantuan graf. Seringkali graf digunakan untuk merepresentasikan suatu jaringan. Misalkan jaringan jalan raya dimodelkan graf dengan kota sebagai simpul (*vertex/node*) dan jalan yang menghubungkan setiap kotanya sebagai sisi (*edge*) yang bobotnya (*weight*) adalah panjang dari jalan tersebut. Dalam beberapa model persoalan dimungkinkan bahwa bobot dari suatu sisi bernilai negatif. Misalkan simpul merepresentasikan kota, sisi merepresentasikan perjalanan yang memungkinkan, dan bobot dari setiap sisi adalah biaya yang dikeluarkan dalam perjalanan yang memungkinkan, dan bobot dari setiap sisi adalah jarak yang ditempuh dalam perjalanan tersebut. Graf G didefinisikan sebagai pasangan himpunan (V, E) . ditulis dengan notasi $G = (V, E)$, yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (*vertices* atau *node*) dan E adalah himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul. V tidak boleh kosong, sedangkan E boleh kosong. (Ristono dan Puryani, 2011).

Simpul pada graf dapat dinomori dengan huruf, seperti $a, b, c, \dots, v, w, \dots$, dengan bilangan asli $1, 2, 3, \dots$, atau gabungan keduanya. Sedangkan sisi yang menghubungkan simpul u dengan simpul v dinyatakan dengan pasangan (u, v) atau dinyatakan dengan lambing e_1, e_2, \dots . Dengan kata lain, jika e adalah sisi yang menghubungkan simpul u dengan simpul v , maka e dapat ditulis sebagai $e = (u, v)$. Secara geometri graf digambarkan sebagai sekumpulan noktah (simpul) di dalam bidang dwimitra yang dihubungkan dengan sekumpulan garis (sisi).



Gambar 2. Tiga buah graf (a) graf sederhana, (b) graf ganda, dan (c) graf semu

Graf sederhana adalah graf yang tidak mengandung gelang maupun sisi-ganda. Graf ganda adalah graf yang mengandung sisi-ganda. Graf semu adalah graf yang mengandung gelang (*loop*), graf semu lebih umum daripada graf ganda, karena sisi pada graf semu dapat terhubung ke dirinya sendiri. (Munir, 2005).

2.2. Lintasan Terpendek

Persoalan mencari lintasan terpendek di dalam graf merupakan salah satu persoalan optimasi. Graf yang digunakan dalam pencarian lintasan terpendek adalah graf berbobot (*weighted graph*), yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Bobot pada sisi graf dapat menyatakan jarak antar kota, waktu pengiriman pesan, ongkos pembangunan, dan sebagainya. Asumsi yang digunakan di sini adalah bahwa semua bobot bernilai positif. Lintasan terpendek adalah jalur yang dilalui dari suatu *node* ke *node* lain dengan besar atau nilai pada sisi yang jumlah akhirnya dari *node* awal ke *node* akhir paling kecil. Lintasan terpendek adalah lintasan minimum yang diperlukan untuk mencapai suatu tempat dari tempat lain. Lintasan minimum yang dimaksud dapat dicari dengan menggunakan graf. Graf yang digunakan adalah graf yang berbobot yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot.

2.3. Algoritma Greedy

Algoritma greedy adalah algoritma yang memecahkan masalah langkah demi langkah, pada setiap langkah :

- Mengambil pilihan yang terbaik yang dapat diperoleh saat itu
- Berharap bahwa dengan memilih optimum lokal pada setiap langkah akan mencapai optimum global. Algoritma greedy mengasumsikan bahwa optimum lokal merupakan bagian dari optimum global.

Persoalan optimasi dalam konteks algoritma greedy disusun oleh elemen-elemen sebagai berikut:

- Himpunan kandidat, C .
Himpunan ini berisi elemen-elemen pembentuk solusi. Pada setiap langkah, satu buah kandidat diambil dari himpunannya.
- Himpunan solusi, S .
Merupakan himpunan dari kandidat-kandidat yang terpilih sebagai solusi persoalan. Himpunan solusi adalah himpunan bagian dari himpunan kandidat.
- Fungsi seleksi – dinyatakan sebagai predikat SELEKSI – merupakan fungsi yang pada setiap langkah memilih kandidat yang paling mungkin untuk mendapatkan solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.
- Fungsi kelayakan (*feasible*) – dinyatakan dengan predikat LAYAK – merupakan fungsi yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala yang ada.
- Fungsi obyektif, merupakan fungsi yang memaksimumkan atau meminimumkan nilai solusi.

Kita berharap optimum global merupakan solusi optimum dari persoalan. Namun, adakalanya 2 optimum global belum tentu merupakan solusi optimum (terbaik), tetapi dapat

merupakan solusi sub-optimum atau pseudo-optimum. Hal ini dapat dijelaskan dari dua faktor berikut:

- algoritma *greedy* tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada.
- pemilihan fungsi SELEKSI: fungsi SELEKSI biasanya didasarkan pada fungsi obyektif (fungsi SELEKSI bisa saja identik dengan fungsi obyektif).

Jika fungsi SELEKSI tidak identik dengan fungsi obyektif, kita harus memilih fungsi yang tepat untuk menghasilkan nilai yang optimum. Karena itu, pada sebagian masalah algoritma *greedy* tidak selalu berhasil memberikan solusi yang benarbenar optimum. Tetapi, algoritma *greedy* pasti memberikan solusi yang mendekati (*approximation*) nilai optimum. (Defindal dkk.,)

Algoritma *greedy* untuk mencari lintasan terpendek dapat dirumuskan sebagai berikut:

- Periksa semua sisi yang langsung bersisian dengan simpul a . Pilih sisi yang bobotnya terkecil. Sisi ini menjadi lintasan terpendek pertama, sebut saja $L(1)$.
- Tentukan lintasan terpendek kedua dengan cara berikut:
 - hitung: $d(i) = \text{panjang } L(1) + \text{bobot sisi dari simpul akhir } L(1) \text{ ke simpul } i \text{ yang lain}$
 - pilih $d(i)$ yang terkecil
 Bandingkan $d(i)$ dengan bobot sisi (a, i) .
 Jika bobot sisi (a, i) lebih kecil daripada $d(i)$, maka $L(2) = L(1) \cup (\text{sisi dari simpul akhir } L(1) \text{ ke simpul } i)$
- Dengan cara yang sama, ulangi langkah 2 untuk menentukan lintasan terpendek berikutnya.

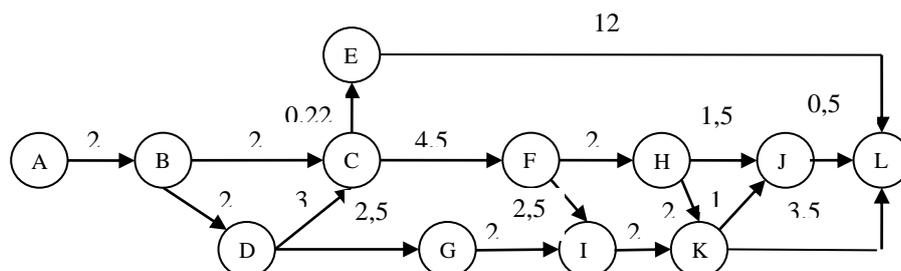
3. HASIL DAN PEMBAHASAN

Nama daerah dari kecamatan Ngaliyan sampai kecamatan Sampangan yang akan dilalui disimbolkan dengan abjad A sampai dengan L, sedangkan daerah disimbolkan dengan lingkaran (*node*), seperti pad tabel 1.

Tabel 1. Keterangan nama kota

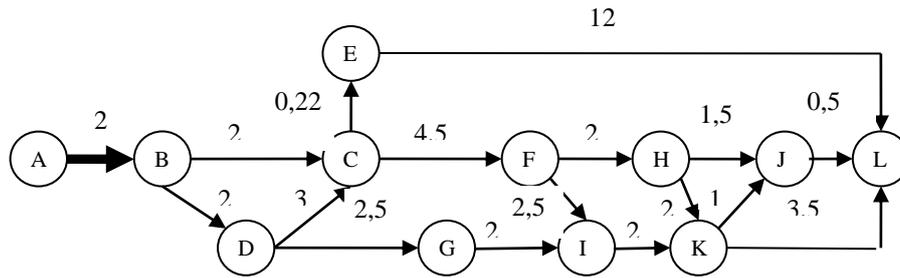
<i>Node</i>	Nama Kota
A	Ngaliyan
B	SPBU Ngaliyan
C	Krapyak
D	Pasadena
E	Tol Krapyak
F	LM. Simongan (Sam Pho Khong)
G	Kalipancur
H	LM.Sampangan
I	Simongan
J	Menoreh
K	SPBU Jembatan Besi
L	Sampangan

Graf dari *node* A sampai dengan *node* L dan bobot tiap-tiap sisi (bobot menyatakan jarak dari setiap *node* satu ke *node* lain) seperti ditunjukkan pada gambar 3 berikut ini :



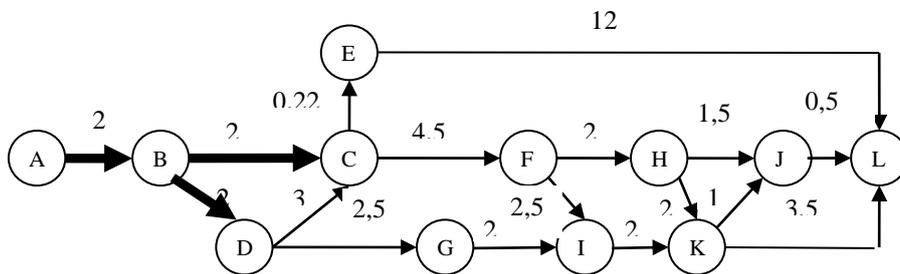
Gambar 3. Graf rute dari *node* A ke *node*

Lintasan pertama yang harus dipilih dari *node* A ke *node* B (sisi AB) dengan bobot 2, ditunjukkan pada gambar 4.



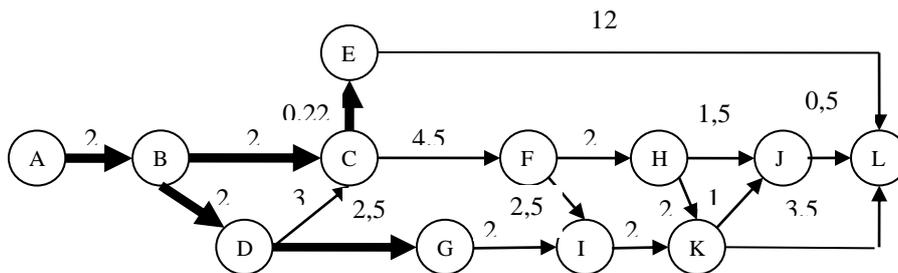
Gambar 4. Graf rute yang dipilih dari *node* A ke ke *node* B

Selanjutnya memilih sisi yang berdekatan dengan sisi AB, yaitu sisi BC dengan bobot 2 dan sisi BD dengan bobot 2. Dengan memilih sisi AB maka bobot totalnya 4 demikian juga memilih sisi BD maka bobot totalnya 4. Sehingga memilih sisi BC dan sisi BD, seperti pada gambar 5.



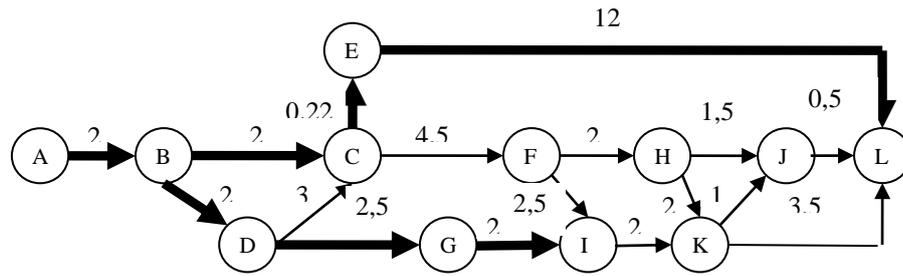
Gambar 5. Graf rute yang dipilih dari *node* B ke ke *node* C atau *node* D

Selanjutnya memilih sisi yang berdekatan dengan sisi BC, yaitu sisi CE dengan bobot 0,22 sehingga bobot totalnya 4,22. Sisi yang berdekatan dengan sisi BD adalah sisi DC dengan bobot 3, sehingga bobot totalnya 7, sisi DG juga berdekatan dengan sisi BD dengan bobot 2,5 sehingga bobot totalnya 6,5. Jadi memilih sisi DG karena mempunyai bobot lebih kecil, ditunjukkan pada gambar 6.



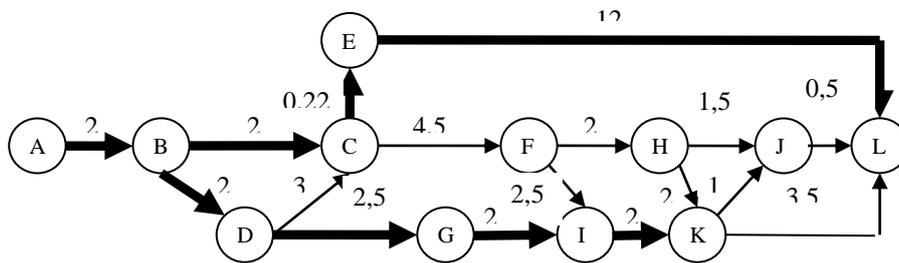
Gambar 6. Graf rute yang dipilih dari *node* C ke ke *node* E dan *node* D ke ke *node* G

Selanjutnya memilih sisi yang berdekatan dengan sisi CE, yaitu sisi EL dengan bobot 12 jadi bobot totalnya 16,22. Sisi GI dengan bobot 2 berdekatan dengan sisi DG, jadi bobot totalnya 9, seperti pada gambar 7.



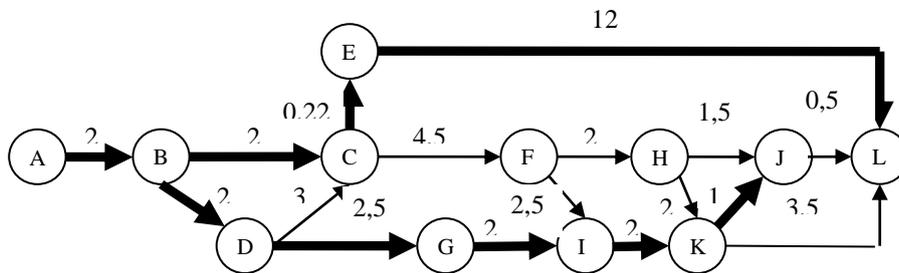
Gambar 7. Graf rute yang dipilih dari node E ke ke node L dan node G ke node I

Selanjutnya memilih sisi yang berdekatan dengan sisi GI, yaitu sisi IK dengan bobot 2 jadi bobot totalnya 10,5, seperti pada gambar 8.



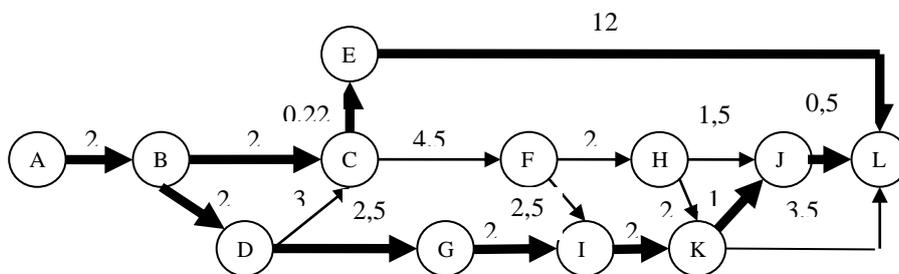
Gambar 8. Graf rute yang dipilih dari node I ke node K

Selanjutnya memilih sisi yang berdekatan dengan sisi IK, yaitu sisi KJ dengan bobot 1 sehingga bobot totalnya 11,5 dan sisi KL dengan bobot 3,5 jadi bobot totalnya 14. Sehingga memilih sisi KJ karena mempunyai bobot lebih kecil, ditunjukkan pada gambar 9.



Gambar 9. Graf rute yang dipilih dari node K ke node J

Selanjutnya memilih sisi yang berdekatan dengan sisi KJ, yaitu sisi JL dengan bobot 0,5 sehingga bobot totalnya 12, ditunjukkan pada gambar 10.



Gambar 10. Graf rute yang dipilih dari node J ke node L

4. KESIMPULAN

Rute yang optimal dengan jarak yang paling pendek adalah $A \rightarrow B \rightarrow D \rightarrow G \rightarrow I \rightarrow K \rightarrow J \rightarrow L$ dengan jarak 12, yaitu Ngaliyan \rightarrow SPBU Ngaliyan \rightarrow Pasadena \rightarrow Kalipancur \rightarrow Simongan \rightarrow SPBU Jembatan Besi \rightarrow Menoreh \rightarrow Sampangan.

Pencarian lintasan terpendek sangat berguna untuk menentukan jalan tersingkat untuk menuju suatu tempat. Sehingga, kita dapat sampai tepat waktu menuju tempat tujuan.

Algoritma *Greedy* dapat diimplementasikan pada kasus pencarian rute terpendek yang memiliki jarak antar node-nodenya pendek. Sebagai contoh untuk kasus TSP pada tempat yang berada di dalam kota

DAFTAR PUSTAKA

- Defindal dkk.,, *Algoritma Greedy untuk Menentukan Lintasan Terpendek*, <http://www.scribd.com/doc/38875940/Metode-Pencarian-Lintasan-Terpendek-Dalam-Graf>, diakses tanggal 7 Maret 2014.
- Dimiyati T. dan Dimiyati A., 1994, *Operation Reserach Model-Model Pengambilan Keputusan*. Edisi Kedua, Sinar Baru Algensindo, Bandung.
- Munir R., 2005, *Matematika Diskrit*. Revisi Kelima, Informatika, Bandung.
- Ristono A. dan Puryani, 2011, *Penelitian Operasional lanjut*. Edisi Pertama, Graha Ilmu, Yogyakarta