

Perbandingan Algoritma A* Dengan Algoritma Greedy Pada Penentuan Routing Jaringan

Lukman¹, Rahmat Hidayat^{*2}, Muhammad Fachrul Risqi Pribadi³

*Penulis Korespondensi

Fakultas Ilmu Komputer, Universitas AMIKOM Yogyakarta

Email : masman@amikom.ac.id, rahmat.hidayat@students.amikom.ac.id,

Muhammad.pribadi@students.amikom.ac.id³⁾

Abstrak

Meningkatnya Perkembangan Internet saat ini sejalan dengan kompleksitas jalur pada internet. Sehingga Network flow optimization menjadi permasalahan utama terkait dengan pemilihan rute terpendek (routing protocol). Fokus pada penelitian ini yaitu mengetahui dan membandingkan proses dalam pencarian rute terpendek dengan algoritma Greedy dan algoritma A* guna mengurangi beban kerja jaringan. Model perbandingan algoritma yang dilakukan ialah melihat cara kerja dari masing-masing algoritma terhadap penentuan jalur routing dari pengirim ke penerima. Berdasarkan dari implementasi percobaan ditemukan bahwa algoritma a* dengan algoritma greedy menemukan jalan yang sama. namun jelas untuk a* akan lebih efektif jika diterapkan pada jaringan yang luas serta rumit. Karena a* menggunakan perhitungan yang pasti sedangkan greedy hanya memandang apa yang ada didepannya yang paling pendek menuju node selanjutnya itu yang dipilih padahal hasil akhir bisa jadi lebih besar dari pada perhitungan a*

Kata Kunci— Routing Protocol, Algoritma a*, Algoritma Greedy.

Abstract

The increasing development of the Internet today is in line with the complexity of on line on the internet. So the Network flow optimization became the main problems related to the election of the shortest route (routing protocol). Focus on research is to find out and compare the process with the shortest route in the search algorithm is Greedy algorithm and A * in order to reduce the workload of the network. Model comparison algorithm that is done is look at the workings of each algorithm against the determination of the routing path from the sender to the receiver. On the basis of the implementation of the experiment it was found that the algorithm a * greedy algorithm finds its way with the same. but it is clear for the a * would be more effective if applied on a broad network as well as complicated. Because the calculation uses a * definite while greedy simply looked at what is the most short front side towards the next node that was selected when the end result can be greater than the calculation of a *

Keywords—Routing Protocol, A start Algorithm, Greedy Algorithm.

1. PENDAHULUAN

Jaringan komputer merupakan sebuah sistem yang terdiri atas komputer dan perangkat jaringan yang bekerja bersama untuk mencapai suatu tujuan yang sama. Tujuan dari jaringan komputer adalah membagi fungsi sumber daya seperti berbagi pemakaian printer, CPU, RAM, Harddisk. Pada komunikasi contohnya email, instant messaging, chatting. Akses informasi contohnya web browsing, agar dapat mencapai tujuan yang sama setiap bagian dari jaringan komputer meminta dan memberikan layanan (service). Client sebagai yang menerima layanan dan Server sebagai yang memberikan layanan. Arsitektur seperti ini sering disebut dengan

sistem client-server, dan digunakan pada hampir seluruh aplikasi jaringan computer. Beberapa cara untuk membuat jaringan komputer diantaranya dengan menggunakan peralatan seperti Hub, Bridge, Switch dan Router.

Router merupakan suatu alat jaringan komputer yang berfungsi untuk mengirim sebuah paket data menuju jaringan tempat tujuannya melalui suatu proses yang dikenal dengan routing. Routing adalah lapisan kedua atau ketiga, tergantung model jaringan yang digunakan. Tugas routing adalah membagi informasi antara router atau network yang berbeda, supaya antara network bisa berhubungan satu sama lain.

Fungsi utama router untuk memberitahukan jalur-jalur tetangga yang diketahui pada saat terjadi pembaharuan informasi atau status jaringan. Protokol routing harus bisa menjabarkan informasi penting dalam pemilihan jalur, seperti bagaimana pengiriman perubahan, pengetahuan yang diperbaharui, kapan waktu pengiriman pembaharuan, dan bagaimana penyebaran informasi router penerus.[1]

Semakin besar suatu jaringan maka manajemen jaringan akan menjadi kompleks dan rumit, sehingga diperlukan manajemen jaringan dan proses routing yang tepat untuk menentukan jalur tercepat dan terdekat untuk mengirimkan pesan tersebut sampai ke tujuan. Proses pemilihan route dari komputer asal ke komputer tujuan inilah yang disebut routing. Dalam menentukan rute tersebut terdapat protokol routing yang bertujuan untuk mengatur router dalam melakukan proses routing tersebut, baik secara statis maupun dinamis routing harus didesain secara efisien.[2]

Dari proses routing yang akan dilakukan, penentuan jalur routing akan di coba implementasi dalam dua algoritma yang berbeda, hal ini dilakukan untuk mengetahui perbandingan proses penentuan jalur routing dari algoritma yang di pakai. Selain itu juga percobaan ini dilakukan untuk mengetahui jalur mana yang lebih cepat pengiriman data ke tujuan.

2. METODE PENELITIAN

Adapun penelitian ini dilakukan dengan tahapan berikut:

2.1. Perancangan dan Analisis

Pembuatan topologi routing dilakukan dengan mensimulasikan node yang sekiranya sesuai dengan jarak antar router. Ditahapan analisa ini dilakukan analisis perbandingan yang dilakukan oleh masing masing algoritma terhadap jalur routing yang akan dilalui.

2.2. Landasan Teori

2.2.1. Routing

Routing merupakan proses sebuah router dapat meneruskan paket data ke jaringan yang akan dikirim. Routing menentukan kemana datagram akan dikirim agar mencapai tujuan yang diinginkan.[3]

2.2.2. Algoritma Greedy

Algoritma greedy merupakan metode yang paling populer untuk memecahkan persoalan optimasi. Greedy sendiri diambil dari bahasa inggris yang artinya rakus, tamak atau serakah. Prinsip algoritma greedy adalah: "take what you can get now!"[5][6]

2.2.3. Algoritma A*

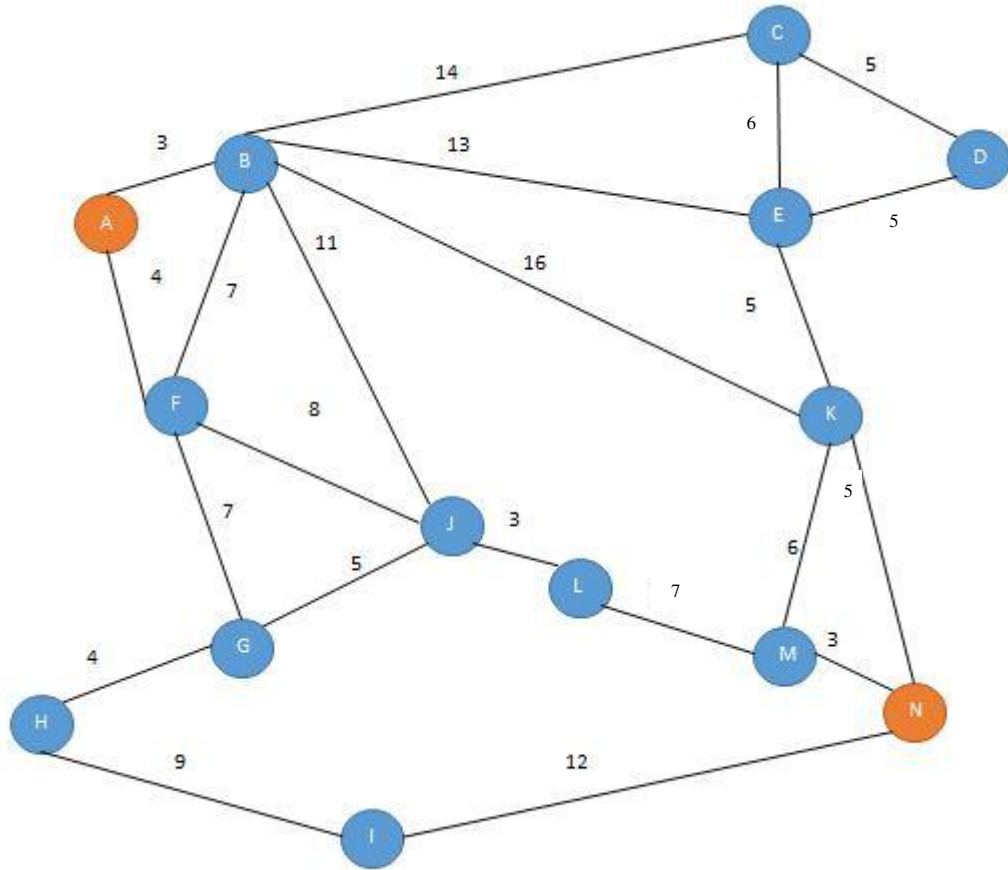
Algoritma A* (A Star) pertama kali diperkenalkan pada tahun 1968 oleh Peter Hart, Nils Nilsson, dan Bertran Raphael. Algoritma ini merupakan algoritma pencarian *graph* yang digunakan untuk menemukan jalur dengan biaya pengeluaran paling kecil dari node awal hingga node tujuan yang diharapkan. Algoritma ini merupakan *best first search* dengan estimasi solusi biayaterkecil dengan jarak dan nilai heuristik (kriteria untuk menentukan alternatif) sebagai dasar pertimbangan [4].

3. HASIL DAN PEMBAHASAN

3.1. Kasus Pertama

3.1.1. Algoritma greedy

Pada topologi ini, menggambarkan node A akan menuju node N dengan jarak antar node sudah ditentukan terlebih dahulu.



Gambar 1. Rancangan Topologi

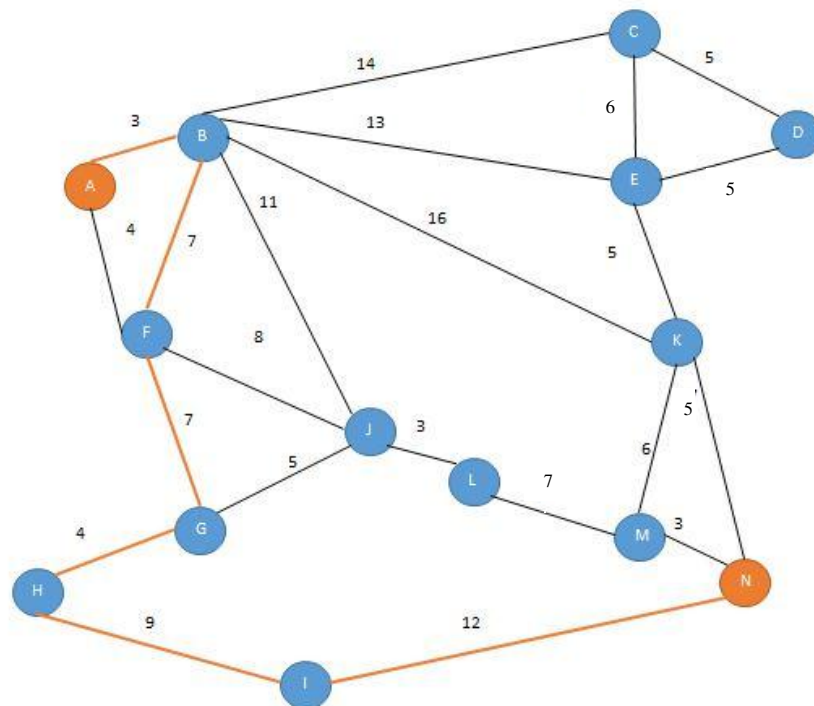
Pada kasus ini kita ambil node A yang akan menuju node N. Dari awal node A bercabang 2, yakni menuju node B dan node F. Maka dilihat jarak terkecil dan dipilihlah jalur menuju node B karena berjarak 3 sedangkan yang menuju node F berjarak 4. Begitu seterusnya hingga bertemu node yang dituju.

Tabel 1. Tabel Komputasi Algoritma Greedy

Proses komputasi	Node		Jarak	hasil	urutan
	asal	tujuan			
1	A	B	3	3	1
		F	4	4	
2	B	C	14	17	
		E	13	16	
		K	16	19	
		J	11	14	
		F	7	10	2

3	F	J	8	18	
		G	7	17	3
4	G	H	4	21	4
		J	5	22	
5	H	I	9	30	5
6	I	N	12	42	6

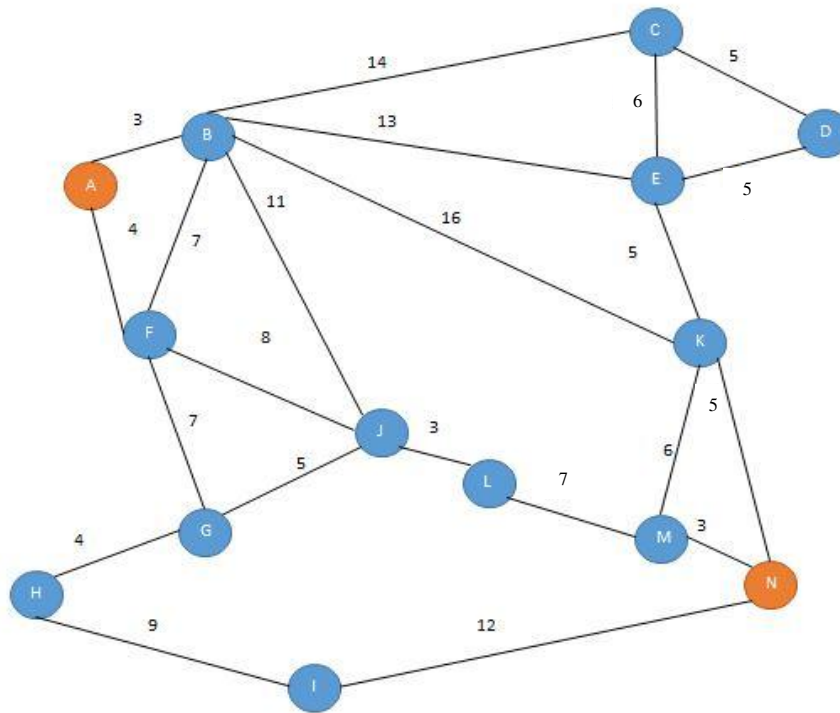
Berdasarkan perhitungan Algoritma GREEDY greedy akan memproses jalur mana yang ditemui paling kecil jaraknya, didapatkan total jarak dari node A menuju node N sebesar 42 dengan total proses komputasi sebanyak 6 proses untuk menemukan node tujuannya. Node-node yang dilewati dari node A agar mencapai N diantaranya adalah node A - B - F - G - H - I. Jalur ini didapatkan dari proses 1 node A menuju B, proses 2 node B menuju F, proses 3 node F menuju G, proses 4 node G menuju H, proses 5 node H menuju I. Node-node tersebut diambil dengan memperhatikan keseluruhan kemungkinan jarak terkecil yang belum pernah dilalui. Pada proses pertama, didapatkan bahwa node B merupakan jarak terkecil dari node A berarti yang diproses selanjutnya adalah node B. Kemudian perhitungan dilanjutkan dari node B dan ditemukan jarak menuju node F adalah yang terkecil dan ditambahkan dari jarak antara A - B dan B - F. Terus seperti itu hingga Greedy menemukan node yang dituju.



Gambar 2. Hasil simulasi greedy

3.1.2. Algoritma Astar

Pada topologi ini, menggambarkan node A akan menuju node N dengan jarak antar node sudah ditentukan terlebih dahulu.



Gambar 3. Rancangan topologi

Dengan menggunakan persamaan *Ecludian Distance* ,Proses komputasi *algoritma A** . Dilihat dari gambar di atas node A terhubung dengan B dan F,di hitung perkiraan biaya dari node A ke node B dan F sebagai contoh Node A ke B

$$h(B) = \sqrt{(2 - 4)^3 - (7 - 8)^2} = 2,23 \quad (1)$$

Setelah diperoleh *ecludian distance* dari node A ke node B, hasilnya ditambahkan dengan jarak sebenarnya untuk mendapatkan perkiraan jalur terpendek sebagai berikut:

$$f(B) = 2,23 + 3 = 5,23 \quad (2)$$

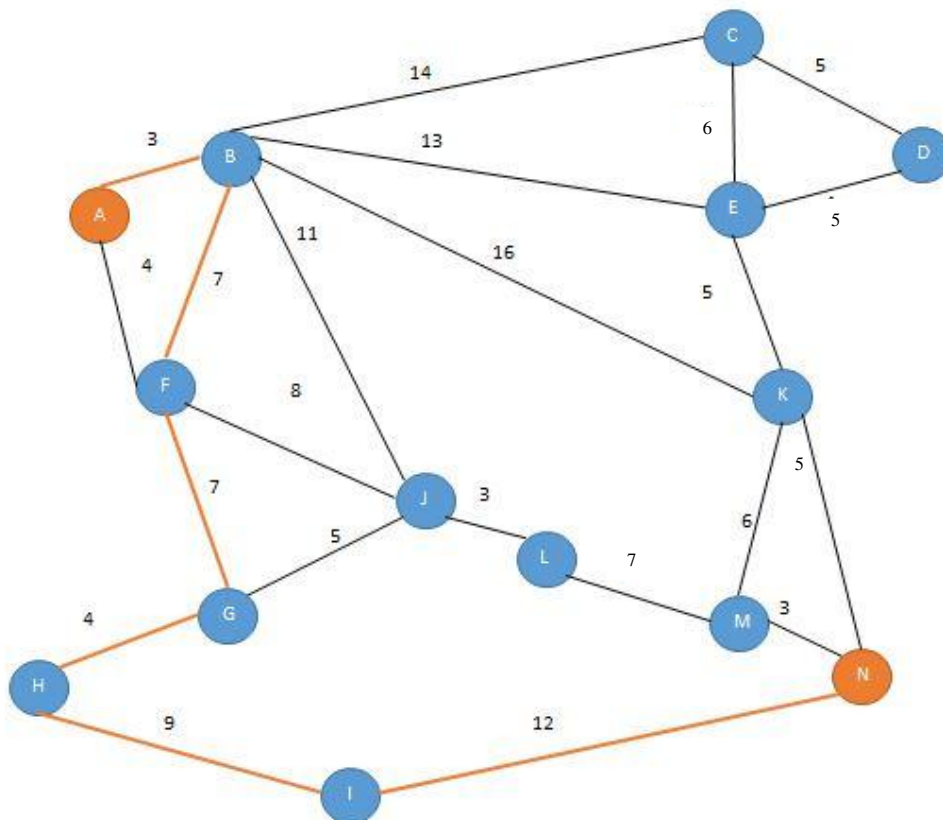
Setelah nilai $f(B)$, $f(F)$,didapat, kemudian dipilih nilai $f(n)$ yang terkecil dari nilai $f(n)$ yang belum pernah dipilih yaitu $f(B) = 5,23$, $f(F) = 5,41$ sehingga nilai $f(n)$ terkecil yang digunakan sebagai node selanjutnya adalah node B. Kemudian ditetapkan node B sebagai node awal dan node C, E, K, J dan F sebagai node tujuan selanjutnya. Tahap perhitungan dengan persamaan *ecludian distance* dan perkiraan jalur terpendek terus diulang hingga node tujuannya adalah tujuan dari kasus ini yaitu N.

Tabel 2. Proses komputasi algoritma Astar

Proses komputasi	Node		hasil	urutan
	asal	tujuan		
1	A	B	5,23	1
		F	5,41	
2	B	C	18,12	2
		E	17,12	
		K	21,09	
		J	15,47	
		F	9,23	

3	F	J	11,60	
		G	10,16	3
4	G	H	7,16	4
		J	7,23	
5	H	I	13,12	5
6	I	N	17,09	6

Berdasarkan perhitungan Algoritma A*, didapatkan total jarak dari node A menuju node N sebesar 51,83 dengan total proses komputasi sebanyak 6 proses untuk menemukan node tujuannya seperti yang dijelaskan pada Tabel 2. Node-node yang dilewati diantaranya adalah node A - B - F - G - H-I. Node-node ini didapatkan dari proses 1 node A menuju B, proses 2 node B menuju F, proses 3 node F menuju G, proses 4 node H menuju node I, dan proses 5 node I menuju node N. Jalur yang didapat dari hasil komputasi dengan algoritma A* digambarkan pada Gambar 4



Gambar 4. Jalur Terpendek dengan algoritma A*

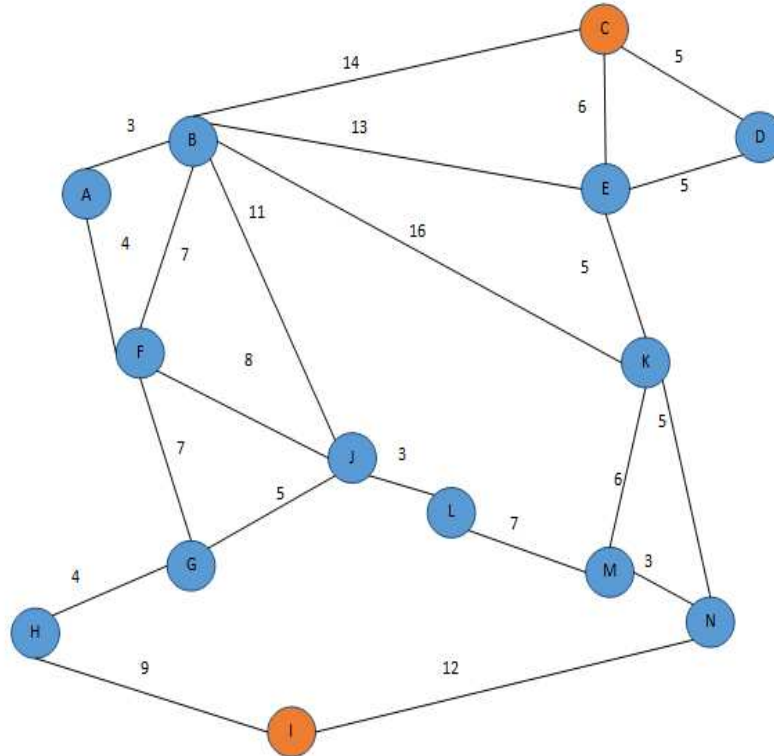
Algoritma A* memperhitungkan dekat atau tidaknya posisi node satu dengan node lainnya bukan hanya melalui jalur yang tersedia diantaranya, melainkan dari posisi sesungguhnya antar node tersebut. Dari penjelasan berikut, dapat dikatakan bahwa algoritma A* cocok untuk topologi jaringan yang berfokus untuk mengirimkan satu paket tertentu atau dapat

dikatakan yang berfokus pada satu tujuan saja. Sehingga untuk paket dengan tujuan yang berbeda pada penerapan algoritma A*, router harus melakukan routing ulang.

3.2. Kasus Kedua

3.2.1. Algoritma greedy

Pada topologi ini, menggambarkan node C akan menuju node I dengan jarak antar node sudah ditentukan terlebih dahulu.



Gambar 5. Rancangan Topologi

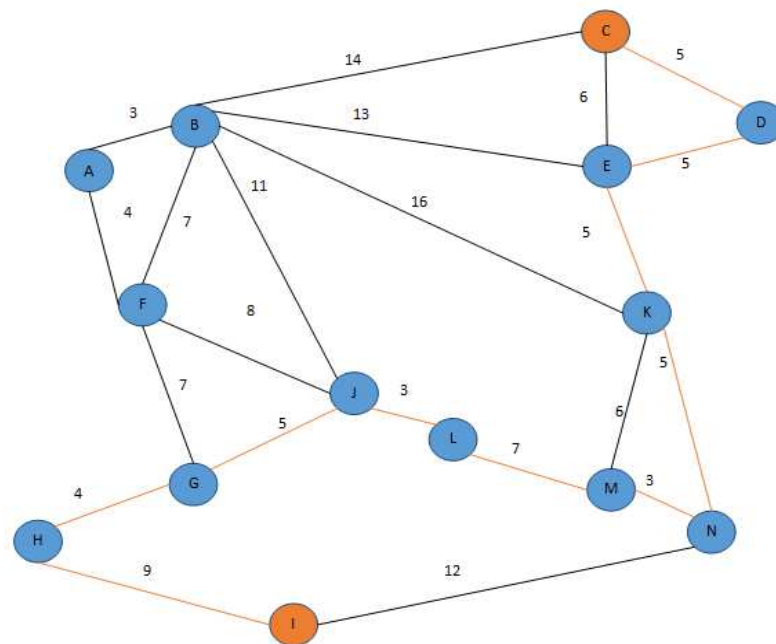
Pada kasus ke-2 ini kita ambil node C yang akan menuju node I. Dari awal node C bercabang 3, yakni menuju node B, node D, dan node E. Maka dilihat jarak terkecil dan dipilihlah jalur menuju node D karena berjarak 5 sedangkan yang menuju node E berjarak 6 dan menuju node B berjarak 14. Begitu seterusnya hingga bertemu node yang dituju.

Tabel 3. Proses Komputasi

Proses komputasi	Node		jarak	hasil	urutan
	asal	tujuan			
1	C	D	5	5	1
		B	14	14	
		E	6	6	
2	D	E	5	10	2

3	E	B	13	23	
		K	3	13	3
4	K	B	16	29	
		M	6	19	
		N	5	18	4
5	N	M	3	21	5
		I	12	30	
6	M	L	7	28	6
7	L	J	3	31	7
8	J	B	11	42	
		F	8	39	
		G	5	36	8
9	G	F	7	43	
		H	4	40	9
10	H	I	9	49	10

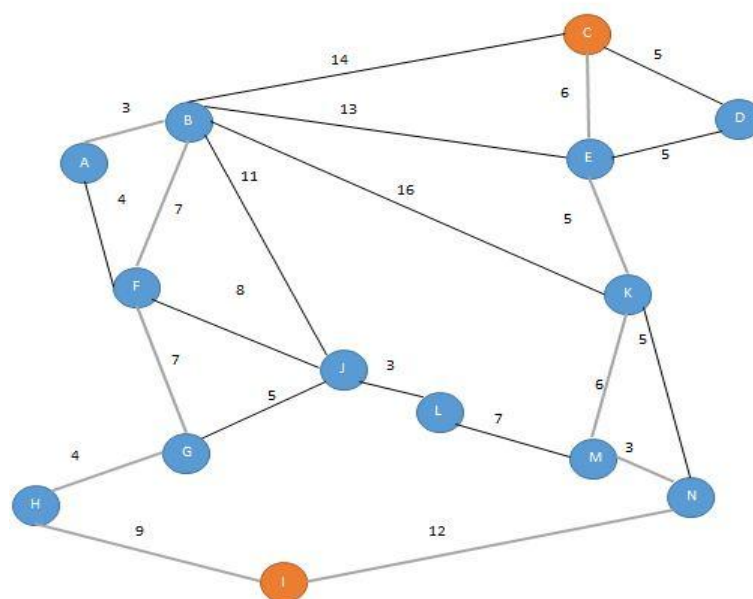
Pada kali ini greedy memproses jalur mana yang ditemui paling kecil, didapatkan total jarak dari node C menuju node I sebesar 49 dengan total proses komputasi sebanyak 10 proses untuk menemukan nodetujuannya. Node-node yang dilewati dari node C agar mencapai node I diantaranya adalah node C - D - E - K - N-M - L - J - G - H - I. Jalur ini didapatkan dari proses 1 node C menuju D, proses 2 node D menuju E, proses 3 node E menuju K, proses 4 node K menuju node N, proses 5 node N menuju node M, proses 6 node M menuju node L, proses 7 node L menuju node J, proses 8 node J menuju node G, proses 9 node G menuju node H, proses 10 node H menuju node I. Node-node tersebut diambil dengan memperhatikan keseluruhan kemungkinan jarakterkecil yang belum pernah dilalui. Pada proses pertama, didapatkan bahwa node D merupakan jarak terkecil dari node E berarti yang diproses selanjutnya adalah node D. Kemudian perhitungan dilanjutkan dari node D dan ditemukan jarak menuju node E adalah yang terkecil dan ditambahkan dari jarak antara C - D dan D - E. Terus seperti itu hingga Greedy menemukan node yang dituju.



Gambar 6. Hasil simulasi greedy

Hasil dari algoritma Greedy sudah dipastikan bahwa dia memilih jarak yang paling terkecil dari sekian cabang dari suatu node. Tabel menunjukkan bahwa algoritma Greedy tidak membutuhkan perulangan yang banyak untuk memeriksa semua jarak dari satu *node* ke *node* yang lainnya. Ini sama dengan menghemat atau memperingan beban komputasi. Dalam konsep routing, dapat diartikan *router* hanya butuh jarak (hop) yang paling dekat dari *router* pengirim menuju *router* penerima. Sehingga Greedy cocok untuk penggunaan jaringan yang tidak terlalu rumit karena hanya akan menghitung menurutnya yang paling dekat menuju *router* selanjutnya

3.2.2. Algoritma Astar



Gambar 7. Rancangan topologi

Pada kasus kedua Dengan menggunakan persamaan *Ecluidian Distance* ,Proses komputasi *algoritma A** . Dilihat dari gambar di atas node C terhubung dengan B , E dan D,di hitung perkiraan biaya dari node C ke node B , E dan D sebagai contoh Node C ke E

$$h(E) = \sqrt{(8 - 9)^3 - (8 - 7)^2} = 1,41 \quad (3)$$

Setelah diperoleh *ecluidian distance* dari node C ke node E, hasilnya ditambahkan dengan jarak sebenarnya untuk mendapatkan perkiraan jalur terpendek sebagai berikut:

$$f(E) = 1,41 + 6 = 7,41 \quad (4)$$

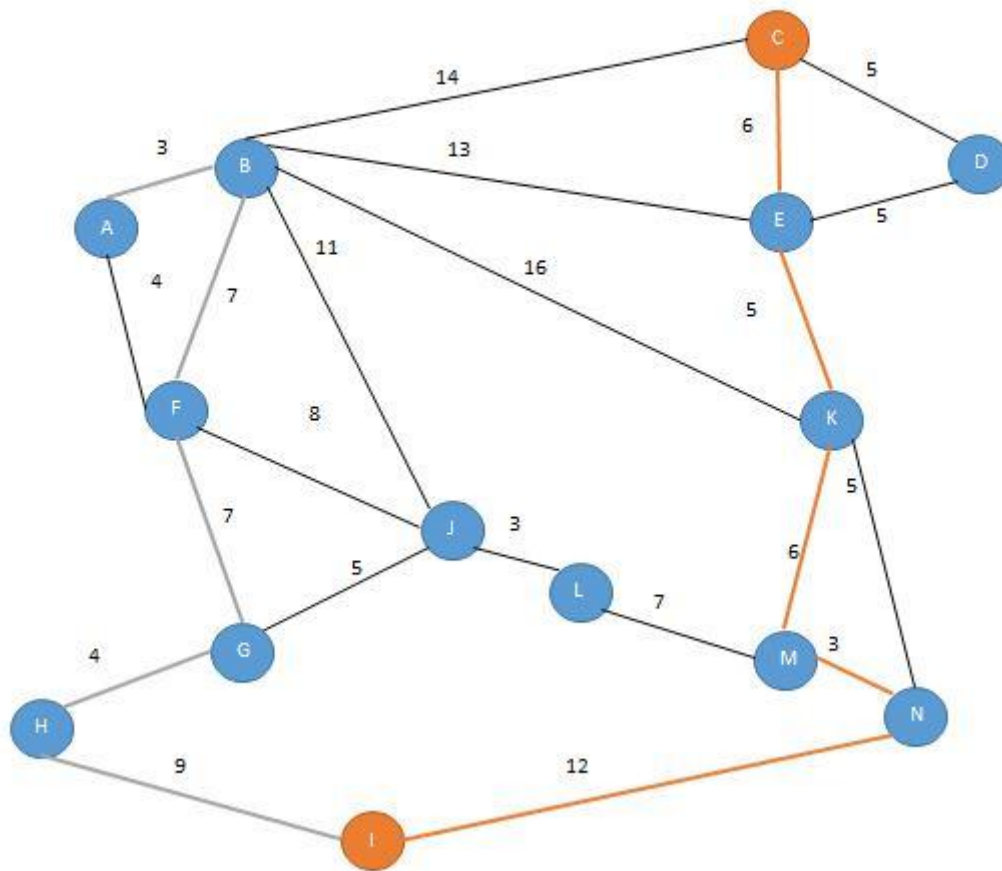
Setelah nilai f(B), f(E),f(D),didapat, kemudian dipilih nilai f(n) yang terkecil dari nilai f(n) yang belum pernah dipilih yaitu f(B) = 18,12 f(E) = 7,14 dan f(D)= 9,16 sehingga nilai f(n) terkecil yang digunakan sebagai node selanjutnya adalah node E

Kemudian ditetapkan node E sebagai node awal dan node B, K dan D sebagai node tujuan selanjutnya. Tahap perhitungan dengan persamaan *ecluidian distance* dan perkiraan jalur terpendek terus diulang hingga node tujuannya adalah tujuan dari kasus ini yaitu I.

Tabel 4. Proses komputasi algoritma Astar

Proses komputasi	Node		Hasil	urutan
	asal	tujuan		
1	C	B	18,12	1
		E	7,14	
		D	9,16	
2	E	B	17,12	2
		K	7,20	
		D	8,16	
3	K	B	20,4	3
		M	11,38	
		N	13,18	
4	M	L	13,4	4
		N	12,38	
5	N	I	20,8	5

Berdasarkan perhitungan Algoritma A*, didapatkan total jarak dari node C menuju node I sebesar 58,9 dengan total proses komputasi sebanyak 5 proses untuk menemukan node tujuannya seperti yang dijelaskan pada Tabel 2. Node-node yang dilewati diantaranya adalah node C-E-K-M-N. Node-node ini didapatkan dari proses 1 node C menuju E, proses 2 node E menuju K, proses 3 node K menuju M, proses 4 node M menuju node N, dan proses 5 node N menuju node I. Jalur yang didapat dari hasil komputasi dengan algoritma A* digambarkan pada Gambar 5.1



Gambar 8. Jalur Terpendek dengan algoritma A*

Algoritma A* memperhitungkan dekat atau tidaknya posisi node satu dengan node lainnya bukan hanya melalui jalur yang tersedia diantaranya, melainkan dari posisi sesungguhnya antar node tersebut. Dari penjelasan berikut, dapat dikatakan bahwa algoritma A* cocok untuk topologi jaringan yang berfokus untuk mengirimkan satu paket tertentu atau dapat dikatakan yang berfokus pada satu tujuan saja. Sehingga untuk paket dengan tujuan yang berbeda pada penerapan algoritma A*, router harus melakukan routing ulang.

4. KESIMPULAN

Dari implementasi kedua percobaan diatas dengan masalah yang sama telah ditemukan bahwa Kasus Pertama kedua Algoritma menemukan solusi yang sama dan pada Kasus Kedua kedua Algoritma menemukan solusi yang berbeda. Namun kami simpulkan Algoritma A* lebih efisien dibandingkan Algoritma Greedy. Sebabnya ada dua, yang pertama : Dapat dilihat pada nilai hasil akhir pada kedua kasus

Tabel 5. Nilai Hasil Akhir

	A*	Greedy
Kasus Pertama		
	17,09	42
Kasus Kedua		
	20,8	49

Dapat dilihat secara perhitungan nilai akhir A* lebih sedikit dari pada greedy. Lalu sebab yang kedua :

Dapat dilihat pada jalur solusi antara algoritma A* dan greedy bahwa A* memilih jalan yang terefektif dari pada greedy. Karena greedy akan memilih jarak terkecil sehingga meskipun tujuan hanya tinggal satu node saja, namun jarak tersebut lebih besar dari pada jarak yang lain maka greedy akan memilih jalur dengan jarak terkecil meskipun itu akan membuat hasil akhir lebih banyak dan jalur yang berputar-putar.

5. SARAN

Sebaiknya setelah ini ada dilakukan kembali penelitian membuktikan Algoritma lain selain A* dan Greedy. Agar ditemukan Algoritma mana yang lebih efektif dari sekian Algoritma yang ada.

DAFTAR PUSTAKA

- [1] Ady Satria, (2018). Faktor Load Balancing Menggunakan Algoritma Greedy dalam Protokol Routing untuk Peningkatan Akses Internet. Universitas Sumatera Utara, Medan
- [2] Suhardi, L. S. (2011). Pengaruh Model Jaringan Terhadap Optimasi Routing Open Shortest Path First (Ospf). Teknologi Vol. 1, 2.
- [3] Sofana, I. (2008). Membangun Jaringan Komputer, Membangun Jaringan Komputer, Membuat jaringan Komputer (Wire & Wireless) Untuk Pengguna Windows dan Linux. Bandung: Informatika
- [4] Y. Yuliani and F. Agus, "WEBGIS PENCARIAN RUTE TERPENDEK MENGGUNAKAN ALGORITMA STAR (A*) (Studi Kasus: Kota Bontang) 1) Yuliani," J. Inform. Mulawarman JIM, vol. 8, no. 2, pp. 50–55, Jun. 2016.
- [5] Nur Hayati, Antoni Yohanes "PENCARIAN RUTE TERPENDEK MENGGUNAKAN ALGORITMA GREEDY" Seminar Nasional IENACO-2014 ISSN: 2337-4349
- [6] Wiradeva Arief K., "Perbandingan Kompleksitas Penerapan Algoritma Greedy untuk Beberapa Masalah", Institut Teknologi Bandung, 2007.