

Optimasi Metaheuristik Koloni Semut untuk Solusi Masalah Jalur Terpendek pada Jaringan Jalan Riil

Ant Colony Metaheuristic Optimization for Shortest Path Problem Solution in Real Road Network

EDWIN TENDA^{1*}, IMAS SUKAESIH SITANGGANG¹, BABA BARUS²

Abstrak

Salah satu permasalahan utama analisis jaringan pada sistem informasi geografis (SIG) adalah menentukan jalur terpendek antara dua lokasi dalam suatu jaringan. Meski terdapat beberapa metode untuk menyelesaikan permasalahan ini tetapi pencarian metode alternatif masih penting dilakukan. Penelitian ini menggunakan pendekatan metode optimasi metaheuristik koloni semut yang terinspirasi dari perilaku alamiah semut, untuk mencari jalur terpendek antara dua titik, pada data jaringan jalan riil (JJR). Pengujian terhadap metode *ant colony optimization (ACO)* dilakukan dalam dua tahap. Pertama, pengujian menggunakan data buatan untuk mendapatkan gambaran pengaturan terbaik dari parameter-parameter metode koloni semut. Kedua, pengujian menggunakan data JJR untuk melihat kinerja metode optimasi koloni semut terhadap data JJR. Hasil pengujian optimasi koloni semut pada data JJR kemudian dibandingkan dengan hasil pengujian algoritme jalur terpendek Dijkstra pada JJR dalam hal panjang jalur optimal dan waktu eksekusi. Hasil pengujian menunjukkan bahwa secara umum metode Dijkstra menghasilkan solusi yang lebih baik dari metode ACO namun pada pengaturan parameter tertentu ACO lebih cepat dari Dijkstra.

Kata Kunci: algoritme Dijkstra, analisis jaringan, jaringan jalan riil, optimasi koloni semut, permasalahan jalur terpendek

Abstract

One of the main problems in geographic information systems (GIS) network analysis is to determine the shortest path between two locations in a network. Although several methods existed to solve the problem but the search for alternative methods is still important. This study uses the approach of ant colony metaheuristic optimization method, that inspired by the natural behavior of ants, to find the shortest path between two points on the real road network (RRN) data. Tests on ant colony optimization (ACO) method is done in two stages. First, the test uses artificial data to get the idea of the best settings of the ant colony method parameters. Second, the test uses the RNN to see the performance of ACO on RRN data. The ant colony method test result on RRN data then compared with the results of the Dijkstra shortest path algorithm test on RRN data in terms of the optimal path length and execution time. The test results showed that generally the Dijkstra method generates better solution than the ant colony optimization method, but on certain parameters setting the ant colony method is faster than the Dijkstra method.

Keywords: *Ant colony optimization, Dijkstra algorithm, network analysis, real road network, shortest path problem*

PENDAHULUAN

Analisis jaringan merupakan salah satu analisis kunci dalam sistem informasi geografis yang diaplikasikan dalam berbagai kebutuhan seperti jaringan telekomunikasi, transportasi, penjadwalan, manajemen proyek, navigasi, perencanaan, pengiriman barang, dan lain-lain.

¹Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor, Bogor 16680;

²Departemen Ilmu Tanah dan Sumber Daya Lahan, FAPERTA IPB, Bogor 16680;

* Penulis Korespondensi: Tel: 085256736925; Surel: tendaedwin@gmail.com

Jaringan adalah suatu sistem dari fitur linear dimana terdapat atribut-atribut yang digunakan untuk menyatakan aliran suatu objek berbasis topologi yang terdiri atas garis dan pertemuan antargaris yaitu simpul, dimana garis tersebut memiliki arah (Curtin 2007; Chang 2008). Analisis jaringan bertumpu pada subdisiplin matematika yakni teori graf. Jaringan didefinisikan sebagai sebuah graf berarah, $G = (V, E)$ yang terdiri atas sebuah himpunan simpul dan sebuah himpunan sisi di mana jumlah simpul $n = |V|$ dan jumlah sisi $m = |E|$, di mana setiap sisi memiliki bobot yang menyatakan ukuran berat, panjang, atau satuan lain sesuai dengan aplikasinya.

Analisis jaringan merupakan salah satu area riset yang signifikan dan terus dilakukan dalam ilmu pengetahuan geografis (Curtin 2007). Di antara berbagai analisis jaringan terdapat suatu permasalahan kunci yang disebut permasalahan jalur terpendek yaitu mencari jalur kumulatif minimum pada jaringan (Zhan dan Noon 1998; Zeng dan Church 2008; Leng dan Zeng 2009). Jalur dapat didefinisikan sebagai hubungan antara dua simpul yaitu titik awal dan titik tujuan ataupun memiliki titik perhentian di antara kedua simpul tersebut (Chang 2008). Permasalahan jalur terpendek pada pemanfaatannya digunakan untuk berbagai aplikasi seperti perencanaan lokasi fasilitas (Horner dan Grubestic 2001), analisis konsumen potensial (Farhan dan Murray 2005), serta perencanaan rute pengiriman barang dan layanan darurat yang menghubungkan fasilitas-fasilitas penting (Chang 2008).

Permasalahan jalur terpendek dapat dikategorikan dalam tiga jenis yaitu (Resende dan Pardalos 2006): menemukan jalur terpendek antara dua simpul, menemukan jalur terpendek dari satu simpul tunggal ke semua simpul lain, menemukan jalur terpendek dari beberapa simpul sumber ke sejumlah simpul tujuan.

Meskipun permasalahan jalur terpendek menjadi topik penelitian selama bertahun-tahun dan banyak algoritme yang dikembangkan serta studi empirik yang sudah dilakukan, tetapi diketahui bahwa tidak ada algoritme tunggal yang terbaik ketika menghadapi semua data uji ataupun semua model representasi jaringan. Belum terdapat jawaban yang jelas untuk algoritme atau set algoritme yang terbaik (Zhan dan Noon 1998; Leng dan Zeng 2009). Beberapa algoritme penyelesaian permasalahan jalur terpendek antara lain algoritme Dijkstra, algoritme A*, algoritme Bellman-Ford, dan algoritme Floyd-Warshall.

Kajian terhadap masalah jalur terpendek masih penting dilakukan untuk mencari dan menerapkan metode-metode alternatif yang dapat digunakan untuk memecahkan permasalahan jalur terpendek pada berbagai macam model representasi jaringan seperti jaringan jalan riil (JJR).

Metode koloni Semut atau *Ant Colony Optimization* (ACO) adalah metode yang terinspirasi dari perilaku alamiah semut dalam mencari makanan di mana semut akan mencari jalur terdekat dari sarang menuju ke sumber makanan dengan mengandalkan kerjasama kelompok dalam bentuk komunikasi. Pada penerapannya ACO telah digunakan pada banyak kasus antara lain, penjadwalan proyek (Merkle *et al.* 2002), *maximum clique* (Fenet dan Solnon 2003), Pewarnaan graf (Jabber *et al.* 2006), dan *travelling salesman problem* (Brezina dan Cickova 2011).

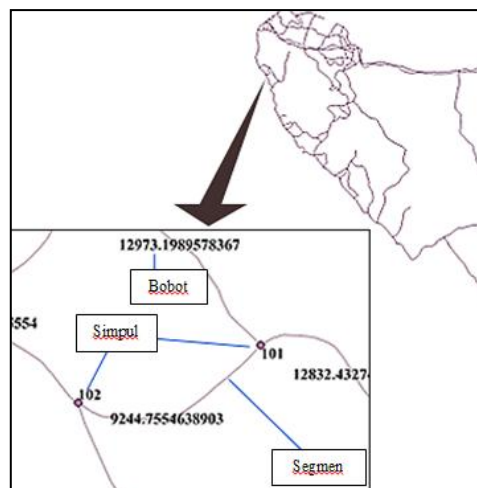
Pada penelitian ini diimplementasikan metode optimasi metaheuristik ACO untuk memecahkan permasalahan jalur terpendek pada model data JJR yang diperoleh dari praproses peta dasar dan untuk mengetahui kinerja dari metode ACO sebagai metode alternatif dalam pemecahan masalah jalur terpendek antara dua titik yang dibandingkan dengan algoritme penyelesaian jalur terpendek Dijkstra (Dijkstra 1959) dalam hal solusi optimal dan waktu eksekusi yang diaplikasikan pada data JJR.

METODE PENELITIAN

Secara garis besar tahapan penelitian ini dibagi dalam empat bagian yaitu, 1) praproses data JJR, 2) implementasi algoritme ACO, 3) pengujian terhadap data buatan dan data jaringan jalan riil, dan 4) perbandingan kinerja dengan algoritme Dijkstra sebagai salah satu algoritme terbaik pada permasalahan rute terpendek antara dua titik (Zeng dan Church 2008).

Praproses Data Jaringan Jalan Riil

Peta dasar untuk pembuatan data jaringan jalan riil diambil dari layanan peta Google Map. Dilakukan tiga tahapan praproses peta. Tahapan pertama, pengecekan peta untuk memastikan apakah unsur-unsur dalam peta sudah lengkap. Pada tahapan ini peta dasar yang dipilih dipastikan memiliki semua unsur jalan yang diperlukan untuk analisis jaringan. Tahapan kedua adalah proses georeferensi untuk memastikan bahwa posisi peta pada representasi muka bumi sudah tepat secara geografis. Tahapan ini dilakukan dengan cara menentukan setidaknya empat titik kontrol pada peta dasar dimana koordinat dari setiap titik menjadi *input* georeferensi. Selain itu ditentukan jenis proyeksi yang digunakan yaitu proyeksi WGS1984. Proses ketiga yaitu digitasi, dilakukan konversi atau perekaman fitur-fitur peta, baik titik yang merepresentasikan titik awal, titik akhir atau persimpangan jalan dan garis yang merepresentasikan segmen jalan yang memiliki bobot panjang segmen jalan (Gambar 1). Struktur atribut dibuat mengikuti model topologi *arc-node* yang menyatakan informasi bagaimana suatu segmen jalan diapit oleh simpul awal dan simpul akhir.



Gambar 1 Contoh fitur-fitur digitasi peta untuk JJR

Implementasi Metode ACO

Pada semut terdapat sebuah mekanisme komunikasi tak langsung yang disebut *stigmergy*. Selama pencarian semut meletakkan sejumlah feromon di sepanjang jalur yang dilewati sebagai penanda. Semut lain akan memilih rute berdasarkan intensitas feromon dari tiap kemungkinan jalur. Walaupun terjadi penguapan feromon, rute yang dilewati semut secara berulang-ulang menyebabkan jalur yang akan dipilih lama kelamaan akan konvergen pada satu jalur terpendek

Metode metaheuristik ACO pada penelitian ini menggunakan algoritme yang diusulkan oleh Dorigo dan Di Caro (1999) yaitu *simple ant colony optimization* (SACO). Pada SACO terdapat beberapa parameter. α yaitu koefisien pengaruh feromon bernilai positif, ρ merupakan konstanta penguapan feromon, t jumlah iterasi, dan n_k jumlah semut dimana penentuan kombinasi parameter akan sangat menentukan kinerja dari algoritme SACO.

Dalam menyusun solusi, semut yang berada pada simpul i akan menentukan simpul berikutnya yaitu $j \in N_i^k$ secara stokastik, menggunakan persamaan probabilitas transisi

(Persamaan 1) dimana N_i^k adalah himpunan dari simpul-simpul tetangga yang terhubung dengan node i yang dapat dilalui oleh semut k (Dorigo dan Di Caro 1999).

$$p_{ij}^k(t) = \begin{cases} \frac{r_{ij}^\alpha(t)}{\sum_{j \in N_i^k} r_{ij}^\alpha(t)} & , j \in N_i^k \\ 0 & , j \notin N_i^k \end{cases} \quad (1)$$

Intensitas feromon pada tiap jalur akan mengalami penguapan, di mana untuk setiap sisi (i, j) feromon menguap sejumlah Persamaan 2, dengan nilai $\rho \in [1, 0]$ sebagai konstanta tingkat penguapan feromon. Konstanta ρ mengendalikan pengaruh ingatan pencarian. Untuk nilai ρ yang besar maka feromon akan menguap dengan cepat. Untuk $\rho = 1$ berarti pencarian sepenuhnya dilakukan secara acak. Dalam Dorigo dan Di Caro (1999), formulasi penguapan feromon adalah:

$$\Gamma_{ij}(t) \leftarrow (1-\rho) \Gamma_{ij}(t) \quad (2)$$

Disaat semut buatan telah berhasil membangun solusi jalur dari simpul sumber ke simpul tujuan maka semut buatan akan kembali ke simpul sumber mengikuti jalur yang dibuat sebelumnya sambil menaruh sejumlah feromon di tiap sisi jalur tersebut. Dalam Dorigo dan Di Caro (1999), jumlah feromon yang ditambahkan pada tiap sisi sepanjang solusi pada iterasi ke- t adalah sebesar Persamaan 3 dimana $L^k(t)$ adalah panjang dari jalur solusi yang dibangun oleh semut k pada langkah ke- t :

$$\Delta \Gamma_{ij}^k(t) = \frac{1}{L^k(t)} \quad (3)$$

Jumlah feromon yang diletakan di sepanjang jalur solusi adalah (Dorigo dan Di Caro 1999):

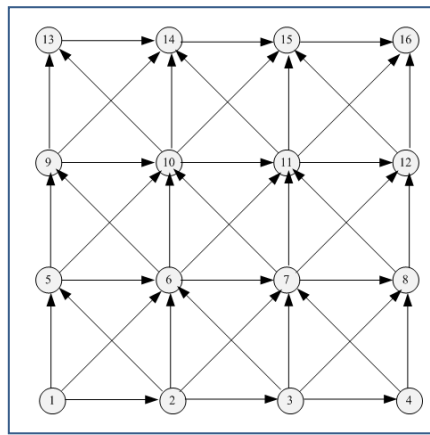
$$\Gamma_{ij}(t+1) = \Gamma_{ij}(t) + \sum_{k=1}^{n_k} \Delta \Gamma_{ij}^k(t) \quad (4)$$

Kualitas solusi ACO dihitung berdasarkan total bobot dari jalur yang ditempuh. Jika $x^k(t)$ menyatakan sebuah solusi pada iterasi ke- t maka $f(x^k(t))$ menyatakan kualitas dari solusi di mana solusi terbaik dengan mencari nilai f paling minimum. Terminasi dari proses dapat dilakukan berdasarkan beberapa kondisi yaitu jika nilai iterasi n_t sudah mencapai nilai yang ditentukan, jika solusi yang dapat diterima sudah ditemukan dengan $f(x^k(t)) \leq \epsilon$ (batas nilai tertentu) atau jika semua semut atau sebagian besar semut sudah melewati jalur yang sama.

Pengujian terhadap Data Buatan dan Data Jaringan Jalan Riil (JJR)

Untuk memperoleh gambaran pengaruh parameter-parameter metode Koloni Semut terhadap kinerja algoritme, dilakukan percobaan pada graf buatan, yaitu graf berarah semi lengkap berpola persegi di mana jumlah simpul (v) dan sisi (e) pada tepian berjumlah sama (Gambar 2). Jumlah sisi dari tiap tepian adalah $v-1$ dengan bobot masing-masing bernilai 1. Pengujian dilakukan untuk melihat pengaruh kombinasi parameter dalam algoritme ACO yaitu α , ρ , n_k , dan t terhadap hasil optimal dan waktu eksekusi. Pada percobaan ini dipilih simpul pertama sebagai simpul sumber dan simpul terakhir sebagai simpul tujuan pada graf buatan.

Dari hasil pengujian tersebut kemudian dipilih kombinasi α , ρ , n_k , dan t yang memberikan hasil terbaik pada data buatan. Kombinasi parameter tersebut kemudian digunakan untuk percobaan pada data JJR. Pada data JJR, dijalankan metode koloni semut dan metode Dijkstra. Pada tiap percobaan ditentukan dua titik, yaitu simpul sumber dan simpul tujuan. Untuk setiap percobaan dilakukan pengulangan sebanyak lima kali dan dihitung nilai rata-ratanya.



Gambar 2 Contoh Graf berarah semi lengkap dengan pola persegi, jumlah node tepian, $v = 4$.

Perbandingan Hasil Optimisasi Koloni Semut dan Dijkstra

Hasil dari metode koloni semut kemudian dibandingkan dengan hasil dari metode Dijkstra. Pada metode Dijkstra (Algoritme 1) untuk mendapatkan jalur terpendek algoritma akan mengunjungi semua simpul di dalam jaringan, sedangkan pada ACO konstruksi jalur terbaik dilakukan secara probabilistik, tanpa mengunjungi semua simpul dalam jaringan. Perbandingan kinerja dilakukan untuk melihat kualitas dari solusi jalur terpendek dan waktu eksekusi dari metode ACO.

Algoritme 1 Algoritme Dijkstra jalur terpendek (Cormen *et al.* 2009)

```

DIJKSTRA (G, l, s)
Initialize-Single-Source (G, s)
S =  $\emptyset$ 
Q = G.V
while Q  $\neq$   $\emptyset$ 
    u = Extract – Min (Q)
    S = S  $\cup$  {u}
    for each vertex v  $\in$  G. Adj[u]
        //relaxation process
        if v.d > u.d + l(u, v)
            v.d = u.d + l(u, v)
            v. $\pi$  = u

```

HASIL DAN PEMBAHASAN

Pengujian pada Data buatan

Pengujian menggunakan dua buah graf berarah semi lengkap, dengan jumlah simpul sisi, $v = 4$ dan $v = 5$. Untuk tiap graf dilakukan pengujian dengan kombinasi nilai parameter α dan ρ yaitu $\alpha = 0.1, 0.5, 1.5, 2.0$ dan $\rho = 0.1, 0.5, 0.9$ dan untuk masing-masing kombinasi α dan ρ dengan dua perlakuan nilai parameter jumlah iterasi $t = 10$ dan 20 . Untuk parameter jumlah semut, $n_k = v$. Untuk tiap pasangan percobaan kombinasi α dan ρ percobaan dilakukan sebanyak lima kali kemudian diambil nilai rata-rata dari solusi optimal serta waktu eksekusi. Dari data hasil pengujian yang disajikan pada Tabel 1, 2, 3 dan 4 dapat dilihat bahwa kombinasi parameter α dan ρ yang menghasilkan nilai paling optimal dalam tiap percobaan yaitu pada kombinasi parameter dengan nilai $\alpha = 0.01$ dan $\rho = 0.1$.

Tabel 1 Hasil percobaan pada graf dengan jumlah sisi, $v = 4$ dan jumlah iterasi, $t = 10$

α	Waktu Eksekusi(detik)			Panjang(Satuan)		
	ρ			ρ		
	0.1	0.5	0.9	0.1	0.5	0.9
0.1	0.0117224	0.0112883	0.0117548	3	3.6	3.4
0.5	0.0107037	0.0116803	0.0121161	4	5	5.6
1.5	0.0100845	0.0120601	0.0125367	4.6	6	6
2	0.0111961	0.0113361	0.0248681	5.4	5.6	6.8

Tabel 2 Hasil percobaan pada graf dengan jumlah sisi, $v = 4$ dan jumlah iterasi, $t = 20$

α	Waktu Eksekusi(detik)			Panjang(Satuan)		
	ρ			P		
	0.1	0.5	0.9	0.1	0.5	0.9
0.1	0.023509	0.0238319	0.0231544	3	3.6	3.2
0.5	0.0214536	0.0233152	0.0192221	3.8	4.6	4.2
1.5	0.0173837	0.0221698	0.0208868	3.6	5.2	5
2	0.0217033	0.0204788	0.0223415	4.8	4.6	5.4

Tabel 3 Hasil percobaan pada graf dengan jumlah sisi, $v = 5$ dan jumlah iterasi, $t = 10$

α	Waktu Eksekusi(detik)			Panjang(Satuan)		
	P			P		
	0.1	0.5	0.9	0.1	0.5	0.9
0.1	0.024612652	0.025014911	0.024547880	4.4	4.6	5.2
0.5	0.023728416	0.025922887	0.020357798	5.2	7.6	6.4
1.5	0.023400748	0.025321331	0.023244680	6.2	7.4	7
2	0.021809152	0.025042314	0.024236478	6	7.4	7.4

Tabel 4 Hasil percobaan pada graf dengan jumlah sisi, $v = 5$ dan jumlah iterasi, $t = 20$

α	Waktu Eksekusi(detik)			Panjang(Satuan)		
	P			P		
	0.1	0.5	0.9	0.1	0.5	0.9
0.1	0.047855867	0.049693800	0.048335207	4.2	4.4	5.2
0.5	0.046673564	0.049877271	0.046112454	6.2	6.8	6.2
1.5	0.050596940	0.049877271	0.043833868	6.4	7.4	6.4
2	0.047911553	0.045340029	0.048661410	6.2	6.8	7.2

Dari hasil percobaan tampak bahwa pengaruh parameter ρ adalah yang paling signifikan terhadap hasil optimal. Nilai ρ yang kecil mengakibatkan pengaruh ingatan semut terhadap sejarah pencarian sangat kuat sehingga kecenderungan untuk memilih jalur terbaik sebelumnya lebih besar. Makin kecil nilai α maka jalur lebih pendek cenderung akan lebih dipilih. Selain itu juga dapat dilihat bahwa jumlah iterasi mempengaruhi panjang bobot minimum jalur solusi, dimana pada $t = 20$ panjang jalur optimal lebih baik dari $t = 10$ namun dengan waktu eksekusi dua kali lebih lama.

Pengujian Pada Data Jaringan Jalan Riil

Untuk pengujian pada data riil digunakan data jaringan jalan provinsi di Aceh dan jaringan jalan kecamatan Bogor Tengah (Gambar 3). Karakteristik dari dataset yang digunakan dinyatakan oleh rasio sisi terhadap simpul yang menunjukkan kompleksitas dari jaringan jalan. Makin besar nilai rasio sisi terhadap simpul maka makin banyak persimpangan yang terbentuk, makin banyak alternatif jalur yang terbentuk dalam jaringan jalan atau dengan kata lain makin kompleks jaringan jalan tersebut. Karakteristik data JJR dapat dilihat pada Tabel 5. Parameter algoritme ACO menggunakan konfigurasi parameter terbaik dari hasil pengujian pada graf buatan yaitu $\alpha = 0.01$, $\rho = 0.1$, dengan jumlah iterasi $t = 10$, dan jumlah semut digunakan $n_k = 5$. Hasil pengujian disajikan pada Tabel 6, 7, 8, dan 9.



Gambar 3 (a) Jaringan jalan kecamatan Bogor Tengah dan (b) Jaringan Jalan Provinsi Aceh

Pada Tabel 6 dan 8, panjang rata-rata adalah panjang jalur rata-rata solusi optimal sedangkan waktu eksekusi adalah waktu eksekusi rata-rata solusi optimal dari tiap percobaan.

Tabel 5 Karakteristik dataset JJR Aceh dan JJR Bogor Tengah

Data JJR	Jumlah sisi	Jumlah simpul	Rasio Sisi terhadap Simpul
Data JJR Aceh	224	161	1.391
Data JJR Bogor Tengah	534	372	1.43

Dari hasil percobaan pada data jaringan jalan Provinsi Aceh dan jaringan jalan Bogor Tengah, untuk kualitas solusi optimal dari algoritme ACO didapati bahwa meskipun algoritme ACO dapat memberikan kualitas solusi hasil jalur optimal yang sama dengan algoritme Dijkstra (Tabel 6 percobaan ke-2, dan Tabel 8 percobaan ke-1, dan ke-5) namun secara umum algoritme Dijkstra menghasilkan kualitas solusi yang lebih baik dibandingkan dengan algoritme ACO.

Tabel 6 Hasil percobaan pada data jaringan jalan riil Aceh

Per-cobaan	Simpul awal	Simpul akhir	ACO		Dijkstra		Selisih panjang optimal	Selisih waktu eksekusi
			Panjang (meter)	Waktu eksekusi (detik)	Panjang (meter)	Waktu eksekusi (detik)		
1	121	64	309203.6	0.01021582	309007	0.0110551	196.6	-0.00084
2	121	68	243879	0.00551971	243879	0.0089504	0	-0.00343
3	162	22	479771.11	0.01645317	443074	0.0166712	36697.1	-0.00022
4	162	139	559497.8	0.02150217	557987	0.0176953	1510.8	0.00380
5	162	142	746079.4	0.02216618	744507	0.0184161	1572.4	0.00375

Tabel 7 Jalur terbaik untuk percobaan pada data jaringan jalan riil Aceh

Per-cobaan	Jalur terbaik	
	ACO	Dijkstra
1	121-76-77-84-74-71-70-69-164-65-63-62-68-163-56-55-53-54-57-64	121-76-77-84-74-71-70-69-164-65-63-62-68-163-56-55-53-54-57-64
2	121-76-77-84-74-71-70-69-164-65-63-62-68	121-76-77-84-74-71-70-69-164-65-63-62-68
3	162-126-124-121-76-77-84-74-71-70-69-65-63-62-68-56-55-53-54-38-34-13-29-32-33-20-12-2-5-4-3-6-8-9-10-22	162-126-124-121-76-77-84-74-71-70-69-164-65-63-62-68-163-56-55-53-54-38-34-13-29-32-33-20-12-2-5-4-3-6-8-9-10-22
4	162-126-124-121-76-77-84-74-71-70-69-65-63-62-67-68-163-56-55-53-54-57-64-66-85-91-107-134-135-140-139	162-126-124-121-76-77-84-74-71-70-69-164-65-63-62-68-163-56-55-53-54-57-64-66-85-91-107-134-135-140-139
5	162-126-124-121-76-77-84-74-71-70-69-65-62-68-56-58-54-57-64-66-85-91-107-134-135-140-139-136-138-142	162-126-124-121-76-77-84-74-71-70-69-164-65-63-62-68-163-56-55-53-54-57-64-66-85-91-107-134-135-140-139-136-138-142

Tabel 8 Hasil percobaan pada data jaringan jalan riil Bogor Tengah

Per-cobaan	Simpul awal	Simpul akhir	ACO		Dijkstra		Selisih panjang optimal	Selisih waktu eksekusi
			Panjang (meter)	Waktu eksekusi (detik)	Panjang (meter)	Waktu eksekusi (detik)		
1	0	95	1822.7129	0.014211	1822.71	0.016185	0	-0.00197
2	0	330	6212.6392	0.051140	5932.81	0.056654	279.83	-0.00551
3	0	384	8166.1269	0.052579	8033.45	0.058516	132.68	-0.00593
4	0	381	3793.7962	0.032925	3364.15	0.038182	429.64	-0.00526
5	236	144	2723.1988	0.022598	2723.19	0.041496	0	-0.01890

Tabel 9 Jalur terbaik untuk percobaan pada data jaringan jalan riil Bogor Tengah

Per-cobaan	Jalur Terbaik Pada Percobaan	
	ACO	Dijkstra
1	0-2-4-5-6-13-20-24-54-99-95	0-2-4-5-6-13-20-24-54-99-95
2	0-2-4-5-6-13-20-24-53-52-95-112-201-120-340-118-218-219-241-222-324-239-183-182-176-166-154-155-158-146-87-78-70-68-63-49-36-30-31-355-329-330	0-2-4-5-6-13-20-24-54-99-334-112-201-120-340-118-218-219-241-222-324-239-183-182-176-166-154-155-158-146-87-76-75-58-38-40-35-28-329-330
3	0-2-4-5-6-13-20-24-54-99-334-112-201-120-340-341-219-241-222-324-239-183-183-182-182-176-166-154-155-158-254-171-165-151-153-136-137-141-139-138-140-142-144-221-230-266-261-281-384	0-2-4-5-6-13-20-24-54-99-334-112-201-120-340-118-218-219-241-222-324-239-183-182-176-166-154-155-158-254-171-165-151-153-136-137-141-139-138-140-142-144-221-230-266-261-281-384
4	0-2-4-5-6-13-20-24-54-99-334-112-201-202-337-338-169-160-152-134-125-110-100-86-84-82-85-381	0-2-4-5-6-13-20-24-54-99-334-336-335-168-169-160-152-134-125-113-84-82-85-381
5	236-199-167-166-154-155-158-254-171-165-151-153-136-137-141-139-138-140-142-144	236-199-167-166-154-155-158-254-171-165-151-153-136-137-141-139-138-140-142-144

Untuk waktu eksekusi, dari hampir semua percobaan algoritme ACO memiliki waktu eksekusi yang lebih cepat daripada algoritme Dijkstra kecuali pada percobaan ke-4 dan ke-5 dalam Tabel 6. Perbedaan tersebut diakibatkan oleh mekanisme mundur yang diberikan kepada semut buatan dimana saat semut berada pada simpul yang tidak memiliki kemungkinan simpul selanjutnya(simpul buntu), semut akan kembali ke simpul sebelumnya, dan melanjutkan pencarian tanpa memperhitungkan lagi simpul yang sudah dikunjungi tersebut. Mekanisme balik tersebut dapat mengakibatkan algoritme ACO menggunakan waktu yang lebih lama.

SIMPULAN

Metode koloni semut dapat diimplementasikan pada permasalahan jalur terpendek antara dua titik. Berdasarkan percobaan yang dilakukan pada data jaringan jalan riil, sebagian besar percobaan yang dilakukan menunjukkan algoritme Dijkstra menghasilkan kualitas solusi yang lebih baik dari algoritme ACO walaupun pada beberapa kasus algoritme ACO menghasilkan kualitas solusi yang setara dengan algoritme Dijkstra. Dengan pengaturan parameter tertentu algoritme ACO memiliki waktu eksekusi yang lebih cepat daripada algoritme Dijkstra walaupun terdapat kasus-kasus tertentu dimana waktu eksekusi dari algoritme ACO lebih lama dibandingkan algoritme Dijkstra, yaitu jika semut bertemu dengan simpul buntu.

SARAN

Pada penelitian selanjutnya dapat memanfaatkan fitur dalam Sistem Informasi Geografis seperti fitur *Buffer* untuk mengendalikan ukuran ruang pencarian. Selain itu dapat dicoba varian lainnya dari algoritme ACO lainnya untuk mencari solusi optimal dalam masalah jalur terpendek pada jaringan jalan riil.

DAFTAR PUSTAKA

- Brezina I, Cickova Z. 2010. Solving The Travelling Salesman Problem Using the Ant Colony Optimization. *Management Information System*. 6(4): 10-14.
- Chang KT. 2008. *Introduction to GIS Ed ke,4*. New York (US). McGraw Hill.
- Curtin KM. 2007. Network analysis in geographic information science: review, assessment and projection. *Cartography and Geographic Information Science*. 34(2): 103-111.
- Dijkstra EW. 1959. A Note on two problems in connection with graphs. *Numerishce Mathematik*. 1: 269-271.
- Dorigo M, DiCaro G. 1999. The ant colony optimization meta, heuristic. Di dalam: Corne D, Dorigo M, Glover F, editor. *New ideas in optimization*; Maidenhead (UK). McGraw Hill. hlm 11-32.
- Farhan B, Murray AT. 2005. A SIG based approach for delineating market areas for park and ride facilities. *Transaction in GIS*. 9: 91-108.
- Fenet S, Solnon C. 2003. Searching for maximum cliques with ant colony optimization. Di dalam: Raidl GR, Meyer JA, Middendorf M, Cagnoni S, Cardalda JJR, Corne D, Gottlieb J, Guillot A, Hart E, Johnson C, Marchiori E, editor. *Application of evolutionary computing-Lecture notes in computer science*. Berlin (DE). Springer-Verlag. 2611: 236-245.
- Horner M, Grubestic T. 2001. A SIG based planning approach to locating urban rail terminals. *Transportation*. 28: 55-77.
- Jabber FM, Ali KH, Hassan KR. 2006. Ant Colony Optimization for Coloring Problem. *Basrah Journal of Science*. 24(2): 38-47.
- Leng J, Zeng W. 2009. An improved shortest path algorithm for computing one-to-one shortest paths on road networks. *The 1st International conference on information science and engineering (ICSE)*; 2009 Des 26-28; Nanjing, Tiongkok. Nanjing (CH): IEEE. 1979-1982.
- Merkle D, Middendorf M, Schmeck H. 2002. Ant Colony Optimization for resource constrained project scheduling. *IEEE Transaction on Evolution Computation*. 6(4): 333-346.
- Resende MGC, Pardalos PM. 2006. *Handbook of optimization in telecommunications*. New York (US). Springer.

- Zeng W, Church RL. 2008. Finding shortest path on real road networks: the case for A*. *International journal of geographical information science*. 23: 531-543.
- Zhan FB, Noon CE. 1998. Shortest path algorithms: an evaluation using real road networks. *Transportation Science*. 32(1):65-73.