

Identifikasi Parameter yang Berpengaruh pada Ant Colony Optimization yang Dimodifikasi pada Penyelesaian Travelling Salesman Problem

Andalani Diri Astami, Chastine Fatichah dan Victor Hariadi

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: chastine@if.its.ac.id

Abstrak — *Travelling Salesman Problem (TSP)* merupakan permasalahan dalam mencari jarak minimal sebuah perjalanan pada sejumlah kota. Dimana setiap kota hanya dikunjungi sekali dan kota awal merupakan kota tujuan. Tujuan utama dari TSP adalah untuk meminimalkan total jarak yang ditempuh.

Permasalahan TSP diselesaikan menggunakan metode *Ant Colony Optimization (ACO)* yang sudah dimodifikasi. Ada dua modifikasi yang akan dilakukan yaitu optimalisasi routing dan individual variation dengan menggunakan metode *Routing Optimization and Individual Variation (ROIVA)*. Optimalisasi routing dapat mengurangi frekuensi routing dan kompleksitas waktu. Individual variation dapat meningkatkan konvergensi dari algoritma ACO.

Dari hasil uji coba pada beberapa dataset dapat disimpulkan bahwa Algoritma ACO yang sudah dimodifikasi mampu mengurangi kompleksitas waktu dan meningkatkan konvergensi pada algoritma ACO konvensional.

Kata kunci: *ant colony optimization (ACO), travelling salesman problem (TSP), routing optimization and individual variation (ROIVA)*

I. PENDAHULUAN

Travelling Salesman Problem (TSP) dinyatakan sebagai permasalahan dalam mencari jarak minimal sebuah perjalanan pada sejumlah n kota. Setiap kota hanya dikunjungi sekali dimana kota awal juga merupakan kota akhir. Persoalan TSP sudah banyak diaplikasikan di beberapa perusahaan antara lain untuk rute pengambilan surat dari kotak pos, rute patroli polisi, rute penjemputan siswa sekolah, dsb. Oleh karena itu, solusi optimal dari permasalahan TSP ini akan sangat membantu perusahaan untuk mengefisienkan proses yang terjadi baik dari segi waktu maupun biaya.

Ant Colony Optimization (ACO) diadopsi dari perilaku koloni semut yang dikenal sebagai sistem semut [1]. Dalam dunia nyata, semut dapat menemukan jalur terpendek dari sarang ke sumber makanan tanpa menggunakan isyarat visual. Mereka juga dapat beradaptasi dengan perubahan lingkungan. Misalnya mencari jalur terpendek baru setelah jalur lama bukan lagi menjadi jalur terpendek saat ini. Hal ini dikarenakan semut bergerak mengikuti jejak *pheromone*.

Pheromone adalah zat kimia yang berasal dari kelenjar endokrin dan digunakan oleh makhluk hidup lain untuk mengenali sesama jenis, individu lain, dan kelompok. Saat semut berjalan, mereka meninggalkan sejumlah *pheromone* pada jalur yang dilalui. Masing-masing semut lebih memilih jalur yang memiliki jumlah *pheromone* lebih banyak.

Dengan prinsip algoritma yang didasarkan pada perilaku koloni semut dalam menemukan rute terpendek, maka ACO dapat diterapkan pada permasalahan TSP. Berdasarkan hasil penelitian yang dilakukan oleh M. Dorigo dan L. M. Gambardella (1997) dalam penyelesaian permasalahan TSP, terbukti bahwa algoritma *Ant Colony Optimization (ACO)* mampu mendapatkan hasil perjalanan terbaik dibandingkan dengan *Genetic Algorithm (GA)*, *Evolutionary Programming (EP)*, *Simulated Annealing (SA)*, dan *Annealing-Genetic Algorithm (AG)* [1].

II. URAIAN PENELITIAN

A. *Ant Colony Optimization*

Ant Colony Optimization (ACO) diadopsi dari perilaku koloni semut yang dikenal sebagai sistem semut [1]. Dalam dunia nyata, semut dapat menemukan jalur terpendek dari sarang ke sumber makanan tanpa menggunakan isyarat visual. Mereka juga dapat beradaptasi dengan perubahan lingkungan. Semut bergerak secara acak dalam mencari makanan. Saat menemukan makanan dan kembali ke koloni, mereka meninggalkan jejak *pheromone*.

Semut yang melewati lintasan yang pendek akan meninggalkan aroma *pheromone* yang lebih tajam daripada yang menempuh lintasan yang lebih panjang. Hal ini dikarenakan *pheromone* dapat menguap sehingga kekuatan daya tariknya akan berkurang. Semakin lama waktu yang dibutuhkan seekor semut dalam mencari makanan dan kembali ke koloni, maka semakin banyak juga *pheromone* tersebut menguap. Penguapan *pheromone* ini bertujuan untuk menghindari terjadinya solusi optimum lokal. Jika sama sekali tidak terjadi penguapan *pheromone*, maka rute yang dipilih oleh semut pertama cenderung akan dipilih oleh semut lainnya. Hal ini akan membatasi ruang penjelajahannya.

Proses dalam ACO bisa dijelaskan sebagai berikut, misalkan ada m semut dalam satu koloni. Semut-semut akan memulai dari sarang mereka menuju tujuan akhir melalui beberapa simpul dan berakhir pada simpul tujuan di akhir setiap siklus atau iterasi. Jika semua semut sudah

menyelesaikan lintasannya, jumlah *pheromone* pada lintasan terbaik secara global akan diperbarui. Lintasan global terbaik artinya terbaik diantara semua semut. Pada awal proses yaitu pada iterasi pertama, semua ruas dari simpul awal akan diberi jumlah *pheromone* yang sama. Sehingga pada iterasi pertama, semua semut akan mulai dari simpul awal dan berakhir pada simpul tujuan. Proses berakhir jika jumlah iterasi maksimum sudah tercapai atau tidak ada lagi solusi yang lebih baik yang bisa didapat dalam beberapa iterasi.

B. ACO untuk TSP

Pertama kali ACO diperkenalkan, ACO sudah diterapkan untuk menyelesaikan persoalan *Traveling Salesman Problem* (TSP) [2]. Dalam TSP, semut memulai dan mengakhiri rutenya di kota yang sama. Pada setiap tahap, semut memilih untuk berpindah dari satu kota ke kota lain menurut beberapa aturan:

1. Semut harus mengelilingi setiap kota tepat satu kali.
2. Sebuah kota yang jauh memiliki kesempatan yang lebih kecil untuk dipilih.
3. Semakin kuat jejak *pheromone* yang ada di jalur antara dua kota, semakin besar kemungkinan jalur tersebut akan dipilih.
4. Setelah setiap iterasi selesai, jejak *pheromone* akan menguap.

Setiap semut memiliki sebuah memori yang disebut dengan *tabulist* atau *tabu_k*. *Tabulist* berisi semua kota yang telah dikunjungi semut. *Tabulist* mencegah semut untuk mengunjungi kota-kota yang sebelumnya telah dikunjungi sehingga tidak akan ada satu kota yang dikunjungi lebih dari sekali.

Pemilihan kota-kota yang akan dilaluinya berdasarkan suatu fungsi probabilitas dengan mempertimbangkan *visibility* (*invers* dari jarak) kota tersebut dan jumlah *pheromone* yang terdapat pada ruas antar kota. Semut-semut akan memilih jalur yang lebih pendek dan atau memiliki tingkat *pheromone* yang tinggi. Untuk menentukan kota tujuan digunakan (2.1).

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, & \text{jika } j \in \mathcal{N}_i^k \\ 0, & \text{lainnya} \end{cases} \quad (2.1)$$

Dimana P_{ij}^k adalah probabilitas semut *k* dari titik *i* ke titik *j*. τ_{ij} adalah jumlah *pheromone* yang terdapat pada jalur antara titik *i* dan titik *j*. $\eta_{ij} = \frac{1}{d_{ij}}$ adalah *invers* jarak titik *i* ke titik *j* (d_{ij}). α adalah sebuah parameter pengendali *pheromone* dan β adalah parameter pengendali jarak ($\alpha > 0$ dan $\beta > 0$). \mathcal{N}_i^k adalah himpunan yang berisi titik-titik yang akan dikunjungi oleh semut *k*. *l* adalah titik yang berada dalam \mathcal{N}_i^k .

Koloni semut akan meninggalkan jejak *pheromone* pada lintasan antar kota yang dilalui. Adanya penguapan dan perbedaan jumlah semut yang lewat menyebabkan kemungkinan terjadinya perubahan jumlah intensitas jejak *pheromone* semut antar kota. Perubahan jumlah intensitas ini dihitung berdasarkan (2.2).

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (2.2)$$

Dimana $\Delta\tau_{ij}^k$ adalah jumlah penambahan jejak *pheromone* antar kota setiap semut dapat dihitung berdasarkan (2.3).

$$\Delta\tau_{ij}^k = \begin{cases} \frac{1}{L_k}, & \text{untuk } (i,j) \in \text{tabu}_k \\ 0, & \text{lainnya} \end{cases} \quad (2.3)$$

Dimana L_k merupakan panjang lintasan yang dilalui semut *k*. *N* Setelah satu siklus terselesaikan atau semua semut sudah selesai melakukan perjalanan, terjadi pembaruan *pheromone* pada setiap kota. Pembaruan *pheromone* dihitung berdasarkan (2.4).

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \Delta\tau_{ij} \quad (2.4)$$

Dimana ρ tingkat evaporasi *pheromone*. Setelah satu siklus terselesaikan, *tabulist* perlu dikosongkan untuk diisi lagi dengan urutan kota yang baru pada siklus selanjutnya jika jumlah siklus maksimum belum tercapai atau belum terjadi konvergensi.

C. Routing Optimization and Individual Variation (ROIVA)

a. Routing Optimization Strategy

Routing pada algoritma *Ant Colony Optimization* (ACO) adalah kondisi semut mencari kota yang akan disinggahi selanjutnya. *Routing* merupakan operasi yang paling sering dilakukan saat menjalankan ACO. Kompleksitas waktu dari ACO juga bergantung pada proses *routing*. *Routing Optimization Strategy* (ROS) dapat mengoptimalkan algoritma ACO dengan mengurangi frekuensi dari *routing*.

Jumlah *routing* dalam proses ACO dapat dihitung menggunakan $m * n * (n - 1)/2$ dalam satu iterasi [4]. Dimana *m* adalah jumlah semut dan *n* adalah jumlah kota. Untuk mengurangi frekuensi dari *routing*, metode ini mempunyai suatu variable *Lmin* yang bertugas untuk menyimpan solusi minimum saat ini. *Lmin* awalnya diinisialisasikan dengan ∞ . Pada saat semut selesai melakukan *routing*, akan dibandingkan panjang jalur saat ini dengan *Lmin*, jika *Lmin* lebih kecil nilainya maka semut yang melakukan perjalanan langsung diberhentikan perjalanannya dari siklus saat ini, jika tidak maka dicari semut minimum lagi.

b. Individual Variation

Dalam ACO, parameter α dan β adalah statik dan semua menggunakan nilai yang sama di setiap prosesnya. Dorigo menyarankan untuk nilai $\alpha = 1$ dan $\beta = 5$ adalah tepat untuk dibanyak situasi [3]. Dalam ACO, awalnya jarak memiliki dampak yang lebih kuat pada proses *routing*. Setelah algoritma berjalan, dampak dari *pheromone* harus diperkuat karena informasi selanjutnya tentang jalur yang lebih baik disimpan dalam *pheromone*. Dari pengamatan ini, ACO dengan *Individual Variation* menggunakan nilai parameter α dan β yang berbeda dan dapat berubah.

Di awal siklus ditentukan nilai α dan β sebagai nilai awal parameter. Pada akhir setiap siklus, semut pemenang atau semut yang memiliki panjang jalur minimum diperbolehkan untuk memodifikasi parameter dengan aturan sebagai berikut:

- α meningkat untuk meningkatkan pengaruh *pheromone*.
- β menurun untuk mengurangi pengaruh jarak.

Strategi ini dinamakan *encouragement* [4]. $P_{ij}^k(\alpha, \beta)$ menunjukkan probabilitas semut *k* dalam memilih kota mana yang akan dikunjungi dari kota *i* ke *j* berdasarkan parameter α

dan β . Dengan x adalah sebuah angka *real*, $0 < x < \tau_{max}$, yang membuat persamaan berikut menjadi (2.5).

$$\frac{x^\alpha}{\sum_{l \in N_i^k} [\tau_{i,l}(t)]^\alpha [\eta_{i,l}]^\beta} = \frac{x^{\alpha+1}}{\sum_{l \in N_i^k} [\tau_{i,l}(t)]^{\alpha+1} [\eta_{i,l}]^\beta} \quad (2.5)$$

Oleh karena itu (2.5) menjadi (2.6).

$$x = \frac{\sum_{l \in N_i^k} [\tau_{i,l}(t)]^{\alpha+1} [\eta_{i,l}]^\beta}{\sum_{l \in N_i^k} [\tau_{i,l}(t)]^\alpha [\eta_{i,l}]^\beta} \quad (2.6)$$

Untuk kota K , jika $\tau_{i,K}(t) \geq x$ maka persamaan menjadi (2.7).

$$\tau_{i,K}(t) \geq \frac{\sum_{l \in N_i^k} [\tau_{i,l}(t)]^{\alpha+1} [\eta_{i,l}]^\beta}{\sum_{l \in N_i^k} [\tau_{i,l}(t)]^\alpha [\eta_{i,l}]^\beta} \quad (2.7)$$

Dengan demikian (2.7) menjadi (2.8).

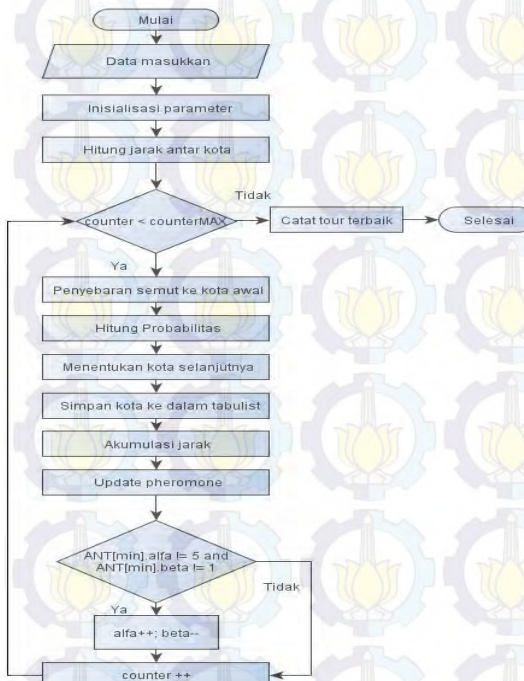
$$\frac{[\tau_{i,K}(t)]^{\alpha+1} [\eta_{i,K}]^\beta}{\sum_{l \in N_i^k} [\tau_{i,l}(t)]^{\alpha+1} [\eta_{i,l}]^\beta} \geq \frac{[\tau_{i,K}(t)]^\alpha [\eta_{i,K}]^\beta}{\sum_{l \in N_i^k} [\tau_{i,l}(t)]^\alpha [\eta_{i,l}]^\beta} \quad (2.8)$$

Berdasarkan (2.1), maka menjadi (2.9) dan (2.10).

$$P_{i,K}^t(\alpha + 1, \beta) \geq P_{ij}^t(\alpha, \beta) \quad (2.9)$$

$$P_{i,K}^t(\alpha, \beta - 1) \geq P_{ij}^t(\alpha, \beta) \quad (2.10)$$

Probabilitas dari kota K untuk dipilih oleh semut k meningkat jika $\tau_{i,K}(t)$ lebih besar daripada sebuah nilai, ketika α dinaikkan. Begitu juga untuk β dengan analisis yang sama. Diagram alir proses penyelesaian TSP menggunakan algoritma ACO dengan metode ROIVA dapat dilihat pada Gambar 1.



Gambar 1. Diagram Alir Proses Penyelesaian TSP Menggunakan Algoritma ACO dengan Metode ROIVA

III. HASIL UJI COBA

A. Dataset

Percobaan uji coba dilakukan pada lima *dataset*. *Dataset* didapatkan dari TSPLIB [5]. *Dataset* yang digunakan adalah Ulysses16 (16 *nodes*), Eil76 (76 *nodes*), Gr96 (96 *nodes*), Gr137 (137 *nodes*), dan Ch150 (150 *nodes*).

B. Uji Coba Ulysses16

Pada permasalahan ulysses16 terdapat 16 kota yang akan dilalui seluruhnya. Hasil percobaan penyelesaian TSP dapat dilihat pada Tabel 1.

Dapat dilihat pada Tabel 1 bahwa solusi terbaik terdapat pada uji coba dengan parameter $\rho = 0,9$ dan dilakukan sebanyak 20 iterasi dengan nilai 74. Pada solusi terbaik, jumlah *routing* yang dilakukan sebanyak 137 kali dan memiliki kompleksitas waktu sebesar 1,57 detik.

C. Uji Coba Eil76

Pada permasalahan eil76 terdapat 76 kota yang akan dilalui seluruhnya. Hasil percobaan penyelesaian TSP dapat dilihat pada Tabel 2.

Dapat dilihat pada Tabel 2 bahwa solusi terbaik terdapat pada uji coba dengan parameter $\rho = 0,9$ dan dilakukan sebanyak 20 iterasi dengan nilai 570,06. Pada solusi terbaik, jumlah *routing* yang dilakukan sebanyak 2934 kali dan memiliki kompleksitas waktu sebesar 29,91 detik.

D. Uji Coba Gr96

Pada permasalahan gr96 terdapat 96 kota yang akan dilalui seluruhnya. Hasil percobaan penyelesaian TSP dapat dilihat pada Tabel 3.

Dapat dilihat pada Tabel 3 bahwa solusi terbaik terdapat pada uji coba dengan parameter $\rho = 0,6$ dan dilakukan sebanyak 20 iterasi dengan nilai 550. Pada solusi terbaik, jumlah *routing* yang dilakukan sebanyak 4713 kali dan memiliki kompleksitas waktu sebesar 29,94 detik.

E. Uji Coba Gr137

Pada permasalahan gr137 ini terdapat 137 kota yang akan dilalui seluruhnya. Hasil percobaan penyelesaian TSP dapat dilihat pada Tabel 4.

Dapat dilihat pada Tabel 4 bahwa solusi terbaik terdapat pada uji coba dengan parameter $\rho = 0,6$ dan dilakukan sebanyak 30 iterasi dengan nilai 816. Pada solusi terbaik, jumlah *routing* yang dilakukan sebanyak 9475 kali dan memiliki kompleksitas waktu sebesar 100,06 detik.

Tabel 1. Hasil Uji Coba untuk Ulysses16

		Solusi Terbaik	Jumlah Routing	Waktu (d)
Iterasi 10	$\rho = 0,1$	75,37	139	2,19
	$\rho = 0,6$	74,87	138	2,33
	$\rho = 0,9$	74,95	139	2,06
Iterasi 20	$\rho = 0,1$	74,84	138	4,46
	$\rho = 0,6$	74,69	138	2,81
	$\rho = 0,9$	74,61	137	1,57
Iterasi 30	$\rho = 0,1$	75,60	138	6,53
	$\rho = 0,6$	75,18	139	7,56
	$\rho = 0,9$	75,30	137	3,81

Tabel 2. Hasil Uji Coba untuk Eil76

		Solusi Terbaik	Jumlah Routing	Waktu (d)
Iterasi 10	$\rho = 0,1$	661,95	2931	14,63
	$\rho = 0,6$	585,95	2932	15,16
	$\rho = 0,9$	572,45	2960	15,24
Iterasi 20	$\rho = 0,1$	646,70	2939	29,57
	$\rho = 0,6$	576,58	2945	29,73
	$\rho = 0,9$	570,06	2934	29,91
Iterasi 30	$\rho = 0,1$	626,65	2939	46,44
	$\rho = 0,6$	577,63	2954	22,49
	$\rho = 0,9$	579,54	2948	14,94

Tabel 3. Hasil Uji Coba untuk Gr96

		Solusi Terbaik	Jumlah Routing	Waktu (d)
Iterasi 10	$\rho = 0,1$	625,63	4685	20,71
	$\rho = 0,6$	564,98	4711	23,28
	$\rho = 0,9$	558,41	4668	17,13
Iterasi 20	$\rho = 0,1$	618,00	4695	45,37
	$\rho = 0,6$	550,78	4713	29,94
	$\rho = 0,9$	558,86	4683	19,42
Iterasi 30	$\rho = 0,1$	609,55	4689	70,95
	$\rho = 0,6$	558,71	4719	27,77
	$\rho = 0,9$	554,77	4671	17,23

Tabel 4. Hasil Uji Coba untuk Gr137

		Solusi Terbaik	Jumlah Routing	Waktu (d)
Iterasi 10	$\rho = 0,1$	930,72	9474	35,80
	$\rho = 0,6$	844,56	9496	39,25
	$\rho = 0,9$	825,35	9476	37,56
Iterasi 20	$\rho = 0,1$	920,49	9454	77,87
	$\rho = 0,6$	818,21	9472	74,58
	$\rho = 0,9$	838,05	9480	40,61
Iterasi 30	$\rho = 0,1$	932,30	9461	116,47
	$\rho = 0,6$	816,32	9475	100,06
	$\rho = 0,9$	844,40	9470	98,10

F. Uji Coba Ch150

Pada permasalahan ch150 ini terdapat 150 kota yang akan dilalui seluruhnya. Hasil percobaan penyelesaian TSP dapat dilihat pada Tabel 5.

Dapat dilihat pada Tabel 5 bahwa solusi terbaik terdapat pada uji coba dengan parameter $\rho = 0,6$ dan dilakukan sebanyak 20 iterasi dengan nilai 6762,64. Pada solusi terbaik, jumlah routing yang dilakukan sebanyak 11368 kali dan memiliki kompleksitas waktu sebesar 91,16 detik.

G. Perbandingan Solusi Terbaik

Berdasarkan hasil uji coba yang sudah dilakukan pada subbab sebelumnya, hasil dikatakan lebih baik jika memiliki nilai lebih kecil. Secara keseluruhan akurasi solusi terbaik ditunjukkan pada

Dapat dilihat pada Tabel 6 bahwa akurasi jarak optimal yang dihasilkan dari uji coba menggunakan ACO ROIVA jika dibandingkan dengan akurasi yang dihasilkan menggunakan ACO konvensional, hasil dari ACO ROIVA memiliki akurasi

Tabel 5. Hasil Uji Coba untuk Ch150

		Solusi Terbaik	Jumlah Routing	Waktu (d)
Iterasi 10	$\rho = 0,1$	8278,76	11371	44,65
	$\rho = 0,6$	6977,43	11378	45,62
	$\rho = 0,9$	6841,00	11382	40,67
Iterasi 20	$\rho = 0,1$	8257,83	11370	89,61
	$\rho = 0,6$	6762,64	11368	91,16
	$\rho = 0,9$	6817,48	11378	86,97
Iterasi 30	$\rho = 0,1$	8298,31	11355	133,38
	$\rho = 0,6$	6859,73	11368	118,17
	$\rho = 0,9$	6773,49	11349	69,41

Tabel 6. Perbandingan Solusi Terbaik

Dataset	Solusi Terbaik		
	TSPLIB	ACO	ACO ROIVA
Ulysses16	68,59	74,62	74,61
Eil76	538,00	563,51	570,06
Gr96	552,09	544,31	550,78
Gr137	698,53	815,15	816,32
Ch150	6528,00	6675,82	6762,64

Tabel 7. Evaluasi Kompleksitas Waktu

Dataset	ACO (d)	ACO ROIVA (d)	Penurunan (%)
Ulysses16	5,04	1,57	68,85
Eil76	30,12	20,91	30,58
Gr96	41,57	29,94	27,98
Gr137	115,05	100,06	13,03
Ch150	96,21	91,16	5,25
Rata-rata			29,14

Tabel 8. Evaluasi Jumlah Routing

Dataset	ACO	ACO ROIVA	Penurunan (%)
Ulysses16	147	137	6,80
Eil76	2975	2948	0,91
Gr96	4751	4671	1,68
Gr137	9538	9470	0,71
Ch150	11430	11349	0,71
Rata-rata			2,16

yang hampir sama dengan akurasi ACO konvensional dalam menyelesaikan permasalahan TSP.

H. Evaluasi Kompleksitas Waktu

Berdasarkan hasil uji coba yang sudah dilakukan pada subbab sebelumnya, hasil dikatakan lebih baik jika memiliki nilai lebih kecil. Evaluasi kompleksitas waktu dilakukan berdasarkan pada solusi terbaik. Secara keseluruhan hasil kompleksitas waktu ditunjukkan pada Tabel 7.

Dapat dilihat pada Tabel 7 bahwa semakin banyak nodes, maka semakin besar waktu yang dibutuhkan untuk menyelesaikan permasalahan tersebut. Kompleksitas waktu dari ACO ROIVA dalam menyelesaikan permasalahan TSP jika dibandingkan dengan kompleksitas waktu dari ACO konvensional memiliki waktu yang lebih sedikit dalam menyelesaikan permasalahan TSP. Dan memiliki rata-rata penurunan sebesar 29,14%.

Tabel 9. Hasil Uji T

	ACO	ACO ROIVA
Mean	5768.20	5715.00
Hypotesis Mean		0
Alpha		0.05
P-value		0.03

I. Evaluasi Jumlah Routing

Berdasarkan hasil uji coba yang sudah dilakukan pada subbab sebelumnya, hasil dikatakan lebih baik jika memiliki nilai lebih kecil. Jumlah *routing* dilakukan berdasarkan pada solusi terbaik Secara keseluruhan jumlah *routing* ditunjukkan pada Tabel 8.

Dari data diatas diketahui bahwa semakin banyak jumlah *nodes* maka akan semakin banyak *routing* yang dilakukan. Jumlah *routing* yang dihasilkan dari uji coba menggunakan ACO ROIVA jika dibandingkan dengan jumlah *routing* ACO memiliki jumlah yang lebih sedikit. Karena persentase penurunan memiliki persentase yang cukup kecil, maka dilakukan pengujian signifikansi menggunakan uji T untuk mengetahui apakah memiliki perbedaan yang signifikan atau tidak. Hasil dari uji T dapat dilihat pada Tabel 9.

Berdasarkan Tabel 9, rata-rata jumlah *routing* dari ACO konvensional sebesar 5768,20 dan rata-rata jumlah *routing* dari ACO ROIVA sebesar 5715,00. Nilai *p-value* hasil uji T memiliki nilai kurang dari nilai *alpha* maka H_0 ditolak sehingga jumlah *routing* dikatakan memiliki perbedaan yang signifikan.

IV. SIMPULAN DAN SARAN

Dari uji coba yang telah dilakukan dan berdasarkan analisis hasil pengujian terhadap sistem ini, dapat diambil kesimpulan dan saran antara lain:

- Algoritma *Ant Colony Optimization* (ACO) dengan metode ROIVA memiliki nilai akurasi solusi terbaik yang hampir sama jika dibandingkan dengan ACO konvensional, tetapi ACO dengan metode ROIVA menurunkan rata-rata kompleksitas waktu sebesar 29,14% dan jumlah *routing* sebesar 2,16% jika dibandingkan dengan ACO konvensional.
- Setiap *dataset* memiliki nilai parameter ρ (tingkat penguapan *pheromone*) yang berbeda-beda untuk mendapatkan hasil yang optimal.
- Iterasi mempengaruhi hasil dari solusi terbaik. Berdasarkan hasil uji coba yang sudah dilakukan, solusi optimum didapatkan pada uji coba sebanyak 20 iterasi dan 30 iterasi.
- Algoritma dapat dioptimalkan kembali sehingga hasil dari solusi terbaik bias lebih baik dari algoritma sebelumnya.

UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kepada Allah SWT yang melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penelitian dengan lancar. Penulis juga mengucapkan terima kasih kepada Ibu Chastine Faticah dan Bapak Victor Hariadi yang telah banyak membantu penulis dalam menyelesaikan penelitian ini. Penulis juga

mengucapkan terima kasih kepada pihak-pihak lain yang turut membantu terselesaikannya penelitian ini.

DAFTAR PUSTAKA

- [1] M. Dorigo and L.M Gambardella, "Ant Colonies for The Travelling Salesman Problem," *IEEE Transaction on Evolutionary Computation*, vol. 1, no. 1, pp. 53-66, 1997.
- [2] J.M Kan and Y. Zhang, "Application of an Improved Ant Colony Optimization on Generalized Travelling Salesman Problem," *Energy Procedia*, vol. 17, pp. 319-325, 2012.
- [3] Vittorio Maniezzo, Alberto Colomi Dorigo M, "The Ant System: optimization by a colony of cooperating agents," *IEEE Transaction on System, Man, and Cybernetics*, vol. Part B, no. 26(1), p. 1 13, 1996.
- [4] Z. L.Pei, J. Yang, and Y. Liang Y. Zhang, "An Improved Ant Colony Optimization Algorithm Based on Route Optimization and Its Application in Travelling Salesman Problem," *BIBE*, pp. 693-698, 2007.
- [5] Heidelberg. <http://www.iwr.uni-heidelberg.de> [Online] <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/STSP.html>.