Optimasi *Asymmetric City Tour* di Kota Kediri Menggunakan *Ant Colony System*

(*Asymmetric City Tour Optimization in Kediri Using Ant Colony System*)

Abidatul Izzah[1], Benni A. Nugroho[2], Wayan F. Mahmudy[3], Fitra A. Bachtiar[4]

***Abstract*—Kediri City is a stopover/transit city and has many potentials in the fields of tourism, education, and industry. Thus, the City of Kediri became one of the cities that are very likely to develop and be crowded. Therefore, it is required to model city tours in several primary fields of Kediri City. In the literature, determining the optimum route can be approached as a traveling salesman problem. However, traveling salesman problem model cannot be used to determine the city tour path as the distance among the point may vary. In this study, we used the concept of asymmetric traveling salesman problem to solve the city tour path. Furthermore, we used the ant colony system algorithm to solve this problem. The cases resolved in this study are the location of the tourism center, industrial center, and education center in Kediri City. The results show the ant colony system is capable of providing optimum tour route solutions, namely the city tourist route 34.65 km, the industrial route 21.19 km, and the school route 28 km.**

***Intisari*—Kota Kediri merupakan kota persinggahan lalu lintas transportasi dan memiliki sejumlah potensi di bidang pariwisata, pendidikan, dan industri. Dengan demikian, Kota Kediri menjadi salah satu kota yang sangat mungkin untuk berkembang dan ramai dikunjungi. Oleh karena itu, Kota Kediri dipandang perlu memodelkan *city tour* di beberapa bidang yang dapat diunggul-kan. Jika ditilik dari disiplin ilmu komputasi, penentuan rute optimum dapat didekati sebagai permasalahan *traveling salesman problem*. Namun, model *traveling salesman problem* standar tidak dapat diimplementasikan pada kasus penentuan *city tour* karena jarak antar lokasi yang mungkin berbeda. Oleh karena itu, pada makalah ini, *city tour* diselesaikan menggunakan konsep *asymmetric traveling salesman problem*. Selanjutnya, algoritme *ant colony system* diimplementasikan dalam penyelesaian masalah ini. Kasus yang diselesaikan pada makalah ini adalah lokasi pusat pariwisata, pusat industri, dan pusat pendidikan di Kota Kediri. Hasil yang diperoleh adalah *ant colony system* mampu memberikan solusi rute tur yang optimum, yakni rute tempat wisata kota 34,65 km, rute industri 21,19 km, dan rute sekolah 28 km.**

Kata Kunci*—*Asymmetric City Tour, Ant Colony System, Optimasi, Traveling Salesman Problem.

[1,2] *Manajemen Informatika,, PSDKU Politeknik Negeri Malang Kediri, Sukorame, Kec. Mojoroto, Kediri, Jawa Timur 64119 INDONESIA (Tlp: +62-354-683128; e-mail: abidatul.izzah90@gmail.com)*

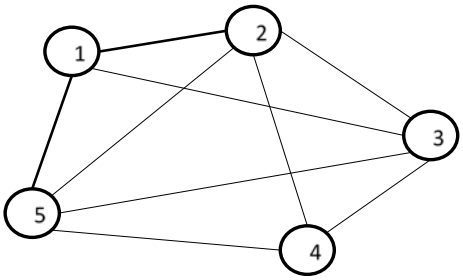
[3,4] *Fakultas Ilmu Komputer, Universitas Brawijaya, Veteran Malang, Ketawanggede, Kec. Lowokwaru, Kota Malang, Jawa Timur 65145 INDONESIA (Tlp: +62-341-577911)*

I. PENDAHULUAN

Kota Kediri merupakan kota persinggahan lalu lintas transportasi dan memiliki sejumlah potensi di bidang pariwisata, pendidikan, dan industri. Kota ini terletak 130 km sebelah barat daya Surabaya dan merupakan kota terbesar ketiga di Jawa Timur setelah Surabaya dan Malang, berdasarkan jumlah penduduk. Kota Kediri memiliki luas wilayah 63,40 km² dan seluruh wilayahnya berbatasan dengan Kabupaten Kediri. Pada tahun 2017, Kota Kediri memperoleh Penghargaan Smart City [1]. Melihat kondisi tersebut, Kota Kediri menjadi salah satu kota yang sangat mungkin untuk berkembang dan ramai dikunjungi. Oleh karena itu, Kota Kediri dipandang perlu untuk memodelkan *city tour* di beberapa bidang yang dapat diunggulkan, antara lain pendidikan, industri, dan pariwisata, untuk membantu upaya pemerintah kota dalam pemberdayaan kota secara menyeluruh.

Jika ditilik dari disiplin ilmu komputasi, penentuan rute *city tour* optimal dapat didekati sebagai permasalahan *Traveling Salesman Problem* (TSP). TSP termasuk kategori *NP-Hard Problem* yang merupakan sebuah permasalahan yang sulit dipecahkan secara analitis, sehingga terdapat banyak variasi metode yang dapat digunakan. Persoalan yang dihadapi ialah cara membangun rute yang optimum dengan mempertimbang-kan aturan pada TSP, yaitu melewati setiap lokasi selain lokasi awal hanya satu kali, dengan tujuan mendapatkan total jarak tempuh minimal, sehingga berdampak pada penghematan biaya transportasi. Secara analitis, permasalahan TSP merupakan permasalahan yang sulit dipecahkan karena terdapat banyak kombinasi alur rute yang mungkin terjadi seiring dengan banyaknya kota yang akan dikunjungi dan juga harus memperhatikan aturan yang berlaku [2]. Namun, model TSP standar tidak dapat diimplementasikan pada penentuan *city tour* karena jarak antar lokasi mungkin berbeda. Kasus seperti ini disebut dengan *asymmetric traveling salesman problem* (ATSP), yakni kasus TSP yang setiap arahnya berlawanan dan direpresentasikan dengan *graph* tidak terarah. Hal ini sesuai dengan fakta di lapangan bahwa lokasi di Kediri dihubungkan oleh jalur yang mungkin tidak ada di kedua arah atau jarak mungkin berbeda dan membentuk *graph* yang tidak berarah [3].

Banyaknya kasus TSP telah diselesaikan baik untuk menyelesaikan kasus secara nyata [3] maupun dalam penelitian pengembangan metode komputasi [4], [5]. Banyaknya penelitian yang telah dilakukan menyebabkan beragam pula metode penyelesaian yang telah digunakan. Salah satu metode yang sering digunakan untuk menyelesaikan masalah *NP-Hard*



Gbr. 1 *Graph* reprentasi *traveling salesman problem.*

adalah *genetic algorithm, Particle Swarm Optimization* (PSO), dan *ant colony system* (ACS) [4]-[10]. Lebih lanjut, ditunjukkan bahwa telah dilakukan penelitian dengan membandingkan beberapa metode untuk kasus pencarian rute ACS yang memberi hasil yang baik dari segi efektivitas waktu dan *cost* penentuan jarak rute dibandingkan dengan metode yang lainnya [6]. Oleh karena itu, pada makalah ini, ACS digunakan sebagai metode untuk menentukan rute terpendek *city tour* di kota Kediri.

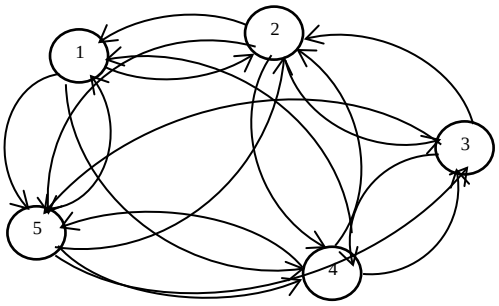
Dengan demikian, makalah ini menyelesaikan permasalahan penentuan rute *city tour* di Kota Kediri untuk lokasi wisata kota, industri dan pendidikan menggunakan metode ACS berbasis model ATSP secara riil. Sistematika penulisan yang digunakan dalam makalah ini diawali dengan studi pustaka tentang ATSP dan ACS. Selanjutnya, dilakukan proses pengambilan data lokasi virtual yang kemudian disimpan dalam matriks *adjecancy*. Selanjutnya, ACS diimplementasikan dan dievaluasi kinerjanya sehingga diperoleh hasil *city tour* yang optimum dan meng-hubungkan lokasi-lokasi tersebut. Kinerja yang dievaluasi antara lain optimasi berdasarkan jarak tempuh dan titik konvergensi algoritme.

II. ASYMMETRIC CITY TOUR

TSP adalah sebuah kasus *combinatory* yang meng-gambarkan seorang *salesman* harus mendatangi seluruh titik kota yang telah ditentukan dan diharuskan tiap titik kota hanya bisa didatangi sekali, dengan ketentuan *salesman* harus memulai dari dan berakhir ke kota asal. Tujuan dari permasalahan TSP ini adalah menentukan rute dengan jarak total atau biaya yang paling optimum [11]. Kasus TSP dapat direpresentasikan dalam *graph* komplet tidak berarah seperti yang ditunjukkan pada Gbr. 1.

Contoh pada Gbr. 1 adalah *graph* yang terdiri atas lima *node* dengan antar *node* memiliki bobot. *Node* pada *graph* tersebut menggambarkan titik-titik lokasi dan bobot antar *node* dapat meggambarkan jarak, waktu, atau *cost* antar lokasi. Lebih lanjut, jika i dan j merepresentasikan lokasi dan d_{ij} merupakan jarak antara i dan j, maka secara matematis, model *graph* tersebut dapat disusun dalam matriks *adjacency* simetris ($d_{ij} = d_{ij}$) sebagai berikut.

$$D = \begin{bmatrix} d_{11} & d_{12} & d_{13} & \dots & d_{1n} \\ d_{21} & d_{22} & d_{23} & \dots & d_{2n} \\ d_{31} & d_{32} & d_{33} & \dots & d_{3n} \\ \dots & \dots & \dots & \ddots & \vdots \\ d_{n1} & d_{n2} & d_{n3} & \dots & d_{nn} \end{bmatrix}$$



Gbr. 2 *Graph* representasi *asymmetric traveling salesman problem.*

Selanjutnya, didefinisikan variabel keputusan x_{ij} yang merepresentasikan ada tidaknya perjalanan dari titik i ke j dalam suatu rute adalah sebagai berikut.

$$x_{ij} = \begin{cases} 1, & \text{jika ada perjalanan kendaraan dari } i \text{ ke } j \\ 0, & \text{jika tidak ada perjalanan kendaraan dari } i \text{ ke } j \end{cases}$$

kemudian jika Z merupakan fungsi tujuan TSP, dan $i, j = 1,2,3, \dots, N$ maka fungsi tujuan Z dirumuskan dengan meminimumkan

$$Z = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \quad (1)$$

dengan batasan kendala

$$\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} = 1 \ , x_{ij} = 0 \text{ atau } 1 \quad (2)$$

Lebih lanjut, kasus TSP tidak dapat diimplementasikan secara langsung pada beberapa kasus, karena pada faktanya jarak antar lokasi tidak selalu sama ($d_{ij} \neq d_{ij}$). Keadaan demikian disebut dengan ATSP. Jika TSP standar direpresentasikan dengan *graph* komplet tidak berarah, sebaliknya, ATSP dapat direpresentasikan menggunakan *graph* berarah seperti ditunjukkan pada Gbr. 2.

Secara matematis, dapat dituliskan bahwa jika himpunan titik dari *graph* K dinotasikan dengan $V(K)$, bobot arah antara i dan j dinotasikan dengan d_{ij}, P adalah sebuah *path* yang tersusun dari semua titik $(v_1, v_2, \dots, v_n)$, panjang *sikel* (tur) C adalah jumlah jalur di $C(P)$, maka ATSP adalah *graph* komplet berbobot K yang terdiri atas N titik dan terdapat tur Hamiltonian H dari graph K dengan meminimumkan bobot yang ada [3].

III. ANT COLONY SYSTEM (ACS)

Metode ACS merupakan metode optimasi yang diilhami dari mekanisme perilaku semut untuk mencari sumber makanan. Fakta di alam bahwa semut mampu menemukan jalur terpendek dari sumber kaki ke sarang tanpa memberi isyarat visual pada semut lain disebut komunikasi *stigmergic*. Untuk memberikan sebuah isyarat tersebut, semut menyimpan sejumlah zat kimia yang disebut feromon dan akan ditinggalkan di tanah saat berjalan mencari rute menuju sumber makanan. Feromon ini bertindak sebagai pengawetan memori semut untuk menghasilkan jalur terpendek, sehingga berguna untuk meningkatkan kemungkinan semut lain mengikuti jalur yang sama yang telah ditinggali feromon oleh semut sebelumnya.

Semakin banyak semut berjalan di setapak jalan, maka semakin banyak feromon yang disimpan di jalan tersebut. Hal ini menyebabkan semakin banyak pula semut yang akan mengikuti jejak jalan setapak tersebut. Melalui mekanisme inilah semut memiliki kecerdasan untuk menemukan jalur terpendek [8].

Perilaku unik semut yang telah diuraikan di atas disusun menjadi sebuah algoritme ACS. Aspek penting dalam algoritme adalah adanya aturan transisi status (*state transition rule*) yang memberikan keseimbangan antara eksplorasi bobot dan bobot sebelumnya. Selanjutnya adalah adanya proses *update* feromon lokal yang diterapkan saat seekor semut menyusun sebuah solusi. Dan ketiga adalah adanya *update* feromon global yang diterapkan pada solusi tur terbaik [8].

Dari ketiga aspek kunci algoritme tersebut, ACS dimulai dengan menyebarkan semut pada seluruh *node* dan berpindah berdasarkan (3).

$$s = \begin{cases} \arg max_{u \in j_k(r)} \{[\tau(r,u)^{\alpha}], [\eta(r,u)^{\beta}]\}, \text{jika } q < q_0 \\ S \qquad\qquad\qquad\qquad\qquad\quad , \text{yang lain} \end{cases} \quad (3)$$

dengan τ adalah feromon, $\eta = 1/d$ adalah invers dari $d(r,s)$, $j_k(r)$ adalah himpunan *node* yang belum dikunjungi oleh k semut yang ada di sedang berada di *node* r, β adalah parameter yang menentukan perbandingan kepadatan feromon dengan jarak, q adalah bilangan random [0,1], q_0 adalah parameter perbandingan ekploitasi dan eksplorasi, dan *S* adalah *node* yang akan dikunjungi selanjutnya. Untuk mengunjungi *node* selanjutnya, semut berpindah berdasarkan (4).

$$P_k(r,s) = \begin{cases} \frac{[\tau(r,s)]^{\alpha}.[\eta(r,s)]^{\beta}}{\sum_{u \in j_k(r)}[\tau(r,s)]^{\alpha}.[\eta(r,u)]^{\beta}} , \text{jika s} \in j_k(r) \\ 0 \qquad\qquad\qquad\qquad\qquad , \text{yang lain} \end{cases} \quad (4)$$

Selanjutnya, setelah seekor semut selesai membuat tur, algoritme memodifikasi atau menguapkan level feromon lokal menggunakan persamaan *update* feromon sebagai berikut.

$$\tau(r,s) \leftarrow (1-p).\tau(r,s) + p\Delta\tau_0 \quad (5)$$

dengan

$$\Delta\tau_0 = {}^{1}/_{L_{mn}.N}$$

dengan L_{mn} adalah panjang tur dari tetangga terdekat dan *N* adalah jumlah total *node*.

Setelah semua semut selesai membangun tur, dipilih semut terbaik yang menghasilkan tur terpendek dari awal untuk melakukan *update* feromon global menggunakan (6).

$$\tau(r,s) \leftarrow (1-p).\tau(r,s) + p\Delta\tau(r,s) \quad (6)$$

dengan

$$\Delta\tau(r,s) = \begin{cases} \frac{1}{L_{gb}} , & \text{if } (r,s) \in \text{tur terbaik} \\ 0, & \text{untuk yang lain} \end{cases}$$

dengan L_{gb} adalah panjang tur terbaik selama iterasi berjalan. Selanjutnya, secara ringkas algoritme ACS dapat ditulis sebagai berikut.

TABEL I
DAFTAR TAMAN KOTA DI KEDIRI

Nama	Lokasi	
	Latitude	***Longitude***
Taman Sekartaji	-7,81039	112,004
Taman Ngronggo	-7,83844	112,011
Taman Harmoni Kediri	-7,81232	112,006
Taman Kota	-7,81745	112,027
Taman Brantas	-7,81228	112,008
Taman Wisata Tirtoyoso	-7,81623	112,029
Taman Bahagia	-7,82224	112,022
Taman Kresek	-7,84017	112,064
Taman Wisata Pagora	-7,81608	112,031
Taman Bunga Matahari	-7,80943	111,975
Taman Apartemen Rakyat	-7,80558	112,018
Surya Taman Wisata	-7,83632	112,056
Taman Unit Halim PT Gudang Garam Tbk	-7,79758	112,027
SLG Green Garden	-7,81231	112,060

```
Inisialisasi: Set feromon semut awal untuk seluruh
node dan matriks jarak antar node
Loop /*perulangan untuk iterasi*/
   Untuk setiap semut di node awal
   Loop /*perulangan untuk step*/
        Setiap semut melakukan state transition
        rule untuk mengusun solusi
        Update feromon local
   Untill seluruh semut selesai membuat solusi
   Update feromon global
Untill kondisi selesai
```

IV. METODOLOGI

A. Pengumpulan Data

Data yang digunakan dalam makalah ini adalah data lokasi pusat pariwisata, pendidikan, dan industri di Kota Kediri. Data yang diambil berupa titik koordinat *latitude* dan *longitude* serta jarak antar *node* lokasi tersebut. Data diambil secara virtual menggunakan layanan peta Google (Google Map). Berikut ini contoh *url* yang digunakan untuk mengambil data lokasi wisata. Untuk mencari lokasi yang lain, maka *query* dapat disesuaikan dengan pencarian yang diinginkan.

```
https://maps.googleapis.com/maps/api/pla
ce/textsearch/json?key=API_KEY&radius=25
000&query=tempat+wisata
```

Dalam pengumpulan sampel data pusat pariwisata, Google menunjukkan taman kota sebagai tempat wisata kota. Dari pencarian ini, ditemukan sampel lokasi sebanyak empat belas titik, seperti yang ditunjukkan pada Tabel I. Kemudian, dalam pengumpulan data lokasi pusat industri, ditemukan sampel sebesar empat belas titik seperti yang ditunjukkan pada Tabel II. Pengumpulan data dilanjutkan untuk kategori pusat pendidikan. Pusat pendidikan dapat berupa sekolah, kampus, lembaga kursus, dan lain-lain. Dari proses pencarian ini, ditemukan sampel lokasi sebanyak empat belas titik sekolah menengah atas seperti yang ditunjukkan pada Tabel III.

Selanjutnya, untuk menentukan *city tour* diperlukan jarak antar titik. Data jarak yang terkumpul kemudian disusun dalam

TABEL II
DAFTAR INDUSTRI DI KEDIRI

Nama	Lokasi	
	Latitude	***Longitude***
PT Mandala Finance	-7,80442	112,00857
PT Gudang Garam Unit 1	-7,80481	112,00975
PT Bhineka Life Indonesia	-7,81148	112,01678
PT Keong Nusantara Abadi	-7,81420	112,01130
PT Adira Dinamika Multi Finance	-7,81574	112,02246
PT Sumber Sari Petung	-7,81764	112,02828
PT Oto Kredit Mobil	-7,80774	112,03293
PT Coca Cola Distributor	-7,80482	112,03448
PT Mekar Karya Nugraha	-7,79721	112,02554
PT Alam Tirta Plast	-7,79732	112,02338
PT Indomarco Adi Prima	-7,79558	112,02197
PT Wonojati Wijoyo	-7,79489	112,01693
PT Aglies Jaya Record	-7,80454	112,01077
PT Triples Group	-7,76658	111,98000

TABEL III
DAFTAR SEKOLAH MENENGAH DI KEDIRI

Nama	Lokasi	
	Latitude	***Longitude***
SMAN 1 Kediri	-7,81122	112,00288
SMAN 2 Kediri	-7,81134	111,99920
SMAN 3 Kediri	-7,83196	112,04413
SMAN 4 Kediri	-7,85435	112,00699
SMAN 5 Kediri	-7,80993	111,97995
SMAN 6 Kediri	-7,85318	112,02202
SMAN 7 Kediri	-7,81367	112,00037
SMAN 8 Kediri	-7,81655	112,02803
SMKN 1 Kediri	-7,81171	111,99742
SMKN 2 Kediri	-7,81125	112,00031
SMKN 3 Kediri	-7,81223	112,01790
SMK PGRI 1 Kediri	-7,81338	111,99352
SMK PGRI 2 Kediri	-7,81979	111,99728
SMK PGRI 4 Kediri	-7,80378	112,00550

matriks *adjacency* asimetris untuk ditentukan rute *city tour*. Oleh karena itu, pengumpulan data dilanjutkan dengan mengambil data jarak menggunakan url seperti berikut ini.

```
${url}origin=${list.get(i).lat},${list.get(
i).lng}&mode=driving&destination=place_id:$
{list.get(j).placeId}
```

B. Penerapan Ant Colony System

Setelah data terkumpul, tahapan selanjutnya adalah menerapkan algoritme ACS terhadap data tersebut. Berdasarkan *pseudocode* yang telah dituliskan sebelumnya, berikut adalah tahapan ACS dalam menyelesaikan kasus ATSP [7].

1. Pertama, menentukan parameter-parameter awal algoritme, yaitu $\alpha, \beta, \rho, \tau_0 = \tau_{ij}$, dan Q.
2. Kedua, menghitung jarak antar *node* dan menghitung visibilitas antar *node*.
3. Ketiga, menyebarkan semut pada tiap *node* secara acak. Posisi awal semut pada *node* pertama ini merupakan posisi pertama *tabu list* tiap semut.

TABEL IV
NILAI DARI PARAMETER *ANT COLONY SYSTEM*

Parameter	Nilai
α	1
β	1
ρ	0,05
Q	1
Jumlah semut	10 - 50
Jumlah iterasi	100 - 300

4. Keempat, mengunjungi *node* lain yang belum ada pada *tabu list* berdasarkan aturan *random proportional rule* pada (4). *Node* dengan probabilitas terbesar merupakan *node* yang akan dikunjungi. Kemudian kunjungan tersebut diisikan pada *tabu list*.
5. Kelima, setelah semua semut mengunjungi seluruh *node* dan masing-masing *tabu list* semut penuh, panjang tur dari *node* pada *tabu list* masing-masing semut dihitung.
6. Keenam, melakukan proses *update* feromon lokal (5) dan mengosongkan *tabu list* masing-masing semut.
7. Ketujuh, mengulang proses ketiga sampai proses keenam dengan menggunakan feromon hasil *update* sebagai feromon awal. Proses ini juga dapat berhenti jika *NCmax* sudah terpenuhi.

V. HASIL DAN PEMBAHASAN

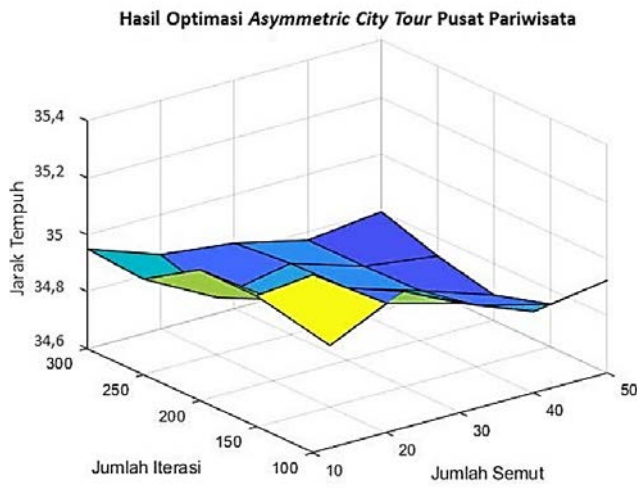
Pengujian dilakukan, untuk menentukan rute *city tour* optimum untuk tiga kategori lokasi tersebut, menggunakan algoritme ACS dengan variasi parameter berbeda. Parameter yang digunakan diperlihatkan pada Tabel IV.

Nilai yang dievaluasi untuk pengujian ini adalah nilai jarak tempuh yang optimum dan meninjau hasil kinerja ASC pada titik konvergensi dan waktu komputasi yang dibutuhkan.

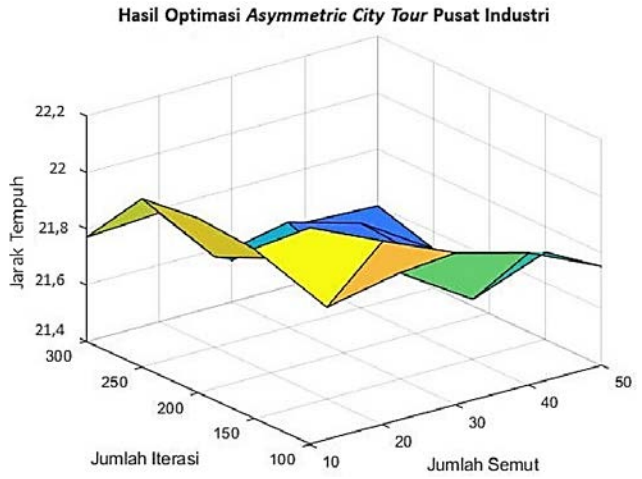
A. Hasil Pengujian Jarak Tempuh

Uji coba dilakukan dengan mulai menerapkan algoritme ACS untuk menentukan rute *city tour* pusat pariwisata dengan empat belas titik lokasi. Pengujian dilakukan dengan memodifikasi jumlah semut dan iterasi. Perhitungan telah dilakukan sebanyak sepuluh kali untuk masing-masing parameter yang sama. Pengujian tersebut kemudian dirata-rata dan dibandingkan dengan parameter lain. Hasil dari pengujian ini ditunjukkan pada Gbr. 3. Dari hasil pengujian ini, nilai optimum sebesar 34,65 km sudah diperoleh bahkan ketika jumlah semut = 30 dan maksimum iterasi = 100. Nilai ini kemudian ditunjukkan kembali dan mendominasi solusi ketika jumlah semut = 50 dan maksimum iterasi = 300. Selanjutnya, dilakukan uji varians hasil rute dan menghasilkan bahwa varians yang ditunjukkan oleh masing-masing parameter tidak memiliki perbedaan yang signifikan. Hal ini yang menyebabkan hasil implementasi dalam grafik 3D yang ditunjukkan pada Gbr. 3 terlihat menurun untuk pertambahan jumlah semut dan iterasi, tetapi penurunan ini tidak bersifat curam.

Uji coba selanjutnya dilakukan dengan menerapkan algoritme ACS untuk menentukan rute *city tour* pusat industri dengan empat belas titik lokasi. Pengujian juga dilakukan dengan memodifikasi jumlah semut dan iterasi. Perhitungan telah dilakukan sebanyak sepuluh kali untuk masing-masing



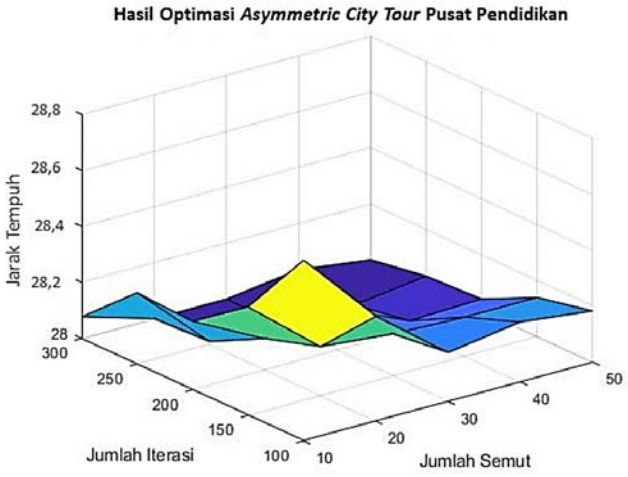
Gbr. 3 Hasil optimasi *asymmetric city tour* pusat pariwisata di Kediri.



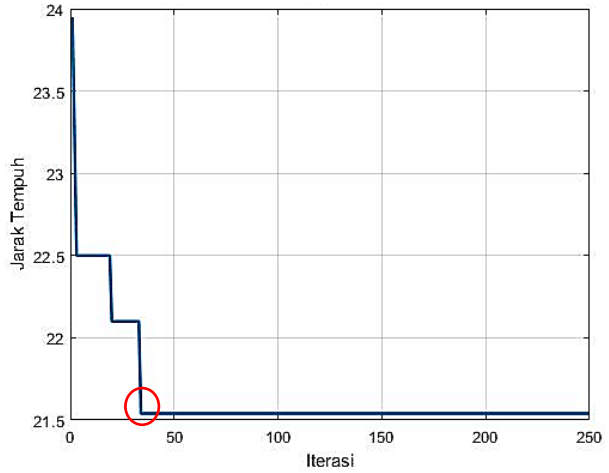
Gbr. 4 Hasil optimasi *asymmetric city tour* pusat industri di Kediri.

parameter yang sama. Pengujian tersebut kemudian dirata-rata dan dibandingkan dengan parameter lain. Hasil dari pengujian ini ditunjukkan pada Gbr. 4. Dari hasil pengujian ini, diperoleh nilai optimum sebesar 21,19 km. Nilai ini kemudian ditunjukkan kembali dan mendominasi seiring bertambahnya jumlah semut dan maksimum iterasi. Selanjutnya, dilakukan uji varians hasil rute dan menghasilkan bahwa varians yang ditunjukkan dari masing-masing parameter memiliki selisih yang lebih besar daripada percobaan pada kasus sebelumnya. Hal ini yang menyebabkan hasil implementasi dalam grafik 3D yang ditunjukkan pada Gbr. 4 lebih terlihat menurun dan curam untuk pertambahan jumlah semut dan iterasi.

Uji coba terakhir dilakukan dengan mulai menerapkan algoritme ACS untuk menentukan rute *city tour* pusat pendidikan dengan empat belas titik lokasi. Skenario pengujian juga seperti yang dilakukan sebelumnya, yakni dengan memodifikasi jumlah semut dan iterasi serta dilakukan sebanyak sepuluh kali untuk masing-masing parameter yang sama. Hasil dari pengujian ini ditunjukkan pada Gbr. 5. Dari hasil pengujian ini, nilai optimum sebesar 28 km sudah diperoleh bahkan ketika jumlah semut = 10 dan maksimum iterasi = 100. Sama seperti pada pengujian pertama, nilai ini



Gbr. 5 Hasil optimasi *asymmetric city tour* pusat pendidikan di Kediri.



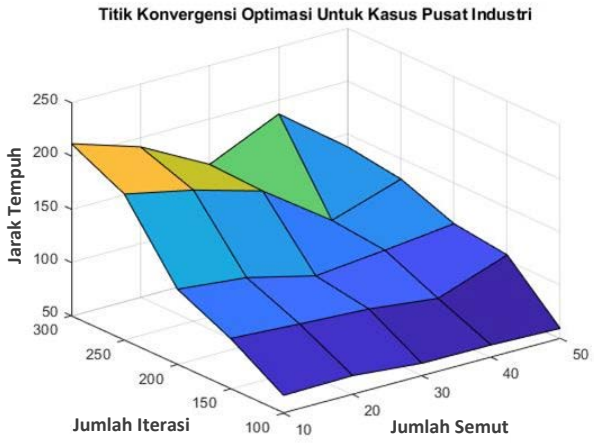
Gbr. 6 Kondisi konvergen.

kemudian ditunjukkan kembali dan mendominasi solusi ketika jumlah semut = 50 dan maksimum iterasi = 300. Setelah dilakukan uji varians untuk masing-masing parameter, hasil yang ditunjukkan pun sama dengan pengujian pertama, yakni varians dari hasil pengujian antar parameter memiliki selisih yang kecil. Hal ini yang menyebabkan hasil implementasi pada Gbr. 5 terjadi penurunan untuk pertambahan jumlah semut dan iterasi, tetapi tidak bersifat curam seperti pada Gbr. 3.

B. Hasil Kinerja Ant Colony System untuk Menyelesaikan Asymmetric City Tour

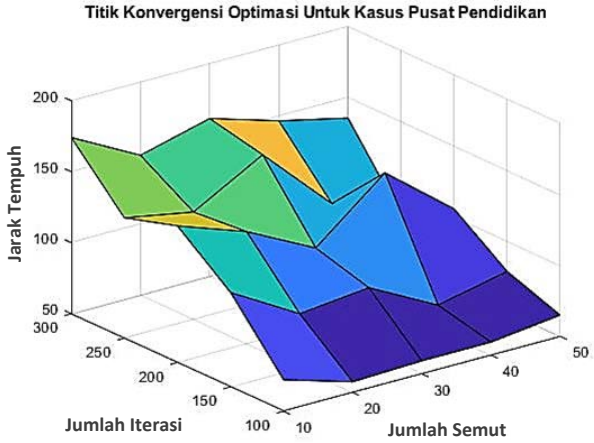
Selanjutnya, uji coba ditinjau dari keadaan berhentinya proses iterasi di suatu titik atau keadaan solusi tersebut tidak berubah sama sekali. Keadaan ini kemudian disebut dengan titik konvergensi (*convergent point)*. Titik konvergensi ini terjadi ketika dalam satu *swarm* seluruh solusi memiliki nilai yang sama, dengan demikian lintasan semut tersebut tidak akan dapat berubah untuk setiap isi dari *tabu list*. Kondisi konvergensi seperti ini dapat dilihat seperti pada Gbr. 6.

Titik konvergensi seperti yang ditunjukkan pada Gbr. 6 dapat membuat kondisi komputasi ini menemukan solusi lebih cepat, tetapi dapat pula membuat kondisi ini tidak menemukan solusi global. Uji coba untuk mengetahui kondisi titik



(a)

(b)



(c)

Gbr. 7 Hasil pengujian titik konvergensi.

konvergensi telah dilakukan dengan skenario yang sama untuk penentuan jarak tempuh *city tour* sebelumnya. Hasil perekaman titik konvergensi ditunjukkan pada Gbr. 7.

Pada Gbr. 7 terdapat kecenderungan yang sama, yaitu semakin bertambahnya jumlah iterasi, semakin bertambah pula

TABEL V
HASIL OPTIMASI KASUS *ASYMMETRIC CITY TOUR* DI KEDIRI

Jenis Tur	Hasil Implementasi ACS		
	Rute	Jarak Tempuh	Waktu Komputasi
Rute tempat wisata kota	6 → 9 → 4 → 7 → 2 → 3 → 5 → 1→ 10 → 13 → 11 → 14 → 12 → 8	34,65 km	0,543 s
Rute industri	13 → 3 → 7 → 1 → 5 → 10 → 14 → 2 → 9 → 6 → 8 → 4 →12 → 11	21,19 km	0,468 s
Rute sekolah	8 → 3 → 6 → 4 → 13 → 7→ 10 → 2 → 9 →12 → 5 → 1→ 14 → 11	28 km	0,566 s

TABEL VI
PERBANDINGAN ACS DAN PSO DALAM PENYELESAIAN KASUS

Kinerja Uji	ACS	PSO		
		Standar	*Hybrid* Mutasi	Paralel
Titik konvergensi	63,4	19,83	16,96	20,63
Waktu komputasi	0,268 s	0,046 s	0,109 s	0,093 s

nilai titik konvergensi. Hal ini menunjukkan bahwa ACS tidak terjebak pada konvergensi dini. Hal ini menyebabkan ACS dapat menemukan solusi global saat penyelesaian kasusnya. Pengujian kinerja ACS kemudian ditinjau dari waktu komputasi yang dibutuhkan. Hasil penyelesaian optimasi rute *asymmetric city tour* menggunakan ACS, berupa jarak tempuh dan waktu komputasi, untuk tiga kasus ini diperlihatkan pada Tabel V.

Tabel V menunjukkan waktu yang dibutuhkan algoritme untuk mendapatkan solusi yang optimum. Namun, jika dilihat dari seluruh hasil pengujian yang dilakukan, waktu yang dibutuhkan algoritme untuk mendapatkan solusi adalah $0{,}3739513 < t < 2{,}679453$. Hal ini dirasa cukup baik karena waktu komputasi ini relatif cepat dan memberikan hasil yang optimum.

C. Hasil Perbandingan Metode ACS dalam Penyelesaian ATSP

Selanjutnya, untuk mengetahui keandalan ACS dalam penyelesaian penentuan rute *city tour*, perlu dilakukan perbandingan dengan metode yang lain. Penelitian tentang optimasi rute *city tour* untuk studi kasus taman kota di Kediri juga pernah dilakukan [12]. Dalam penelitian tersebut, optimasi dilakukan menggunakan metode PSO dengan mengombinasikan operator mutasi dan membagi *swarm* menjadi dua *subswarm*. Hasil perbandingan dari optimasi tersebut dengan metode ACS ditunjukkan pada Tabel VI.

Tabel VI menunjukkan perbandingan kinerja ACS dan PSO dari segi titik konvergensi dan waktu komputasi. Hasil penentuan rute tidak ditampilkan karena menghasilkan nilai yang sama. Sedangkan dari sisi konvergensi *swarm*, ACS memiliki keunggulan dibandingkan dengan PSO standar maupun yang telah dimodifikasi karena dari hasil tersebut

terlihat bahwa ACS tidak terjebak pada *premature convergence* seperti halnya PSO. Akan tetapi, karena ACS tidak terjebak pada *premature convergence*, maka algoritme ini membutuhkan waktu komputasi yang lebih besar dibandingkan PSO. Lebih lanjut, pengujian lebih dalam perlu dilakukan, karena nilai ini hanya diambil dari sebuah studi kasus dengan dimensi kecil. Oleh karena itu, studi lanjut dalam pengambilan kesimpulan ini sangat diperlukan.

VI. KESIMPULAN

Model TSP standar tidak dapat diimplementasikan pada penentuan *city tour* karena jarak antar lokasi mungkin berbeda. Oleh karena itu, pada makalah ini *city tour* diselesaikan menggunakan konsep ATSP. Selanjutnya, algoritme ACS diimplementasikan dalam penyelesaian masalah ini. Kasus yang diselesaikan adalah lokasi pusat pariwisata, pusat industri, dan pusat pendidikan di Kota Kediri. Hasil yang diperoleh adalah ACS mampu memberikan solusi rute tur yang optimum dan lebih efektif, yakni rute tempat wisata kota 34,65 km, rute industri 21,19 km, dan rute sekolah 28 km. Pengujian untuk kinerja yang baik oleh ACS juga ditunjukkan pada titik konvergensi yang tidak terjebak pada lokal optimum dan waktu komputasi yang cukup singkat.

Selanjutnya, penelitian ini belum diimplementasikan pada sebuah peranti. Mengingat studi kasus yang diambil sangatlah cocok untuk implementasi pada peranti lunak bergerak (aplikasi *mobile*), maka diharapkan penelitian ini dapat dikembangkan lebih lanjut pada peranti lunak bergerak berbasis Android maupun IOS.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada Kementrian Riset, Teknologi, dan Pendidikan Tinggi atas dukungan yang diberikan hingga terselesaikannya Penelitian Kerjasama Antar Perguruan Tinggi 2019 ini.

REFERENSI

[1] A. Chusna (2017) ANTARA News Jawa Timur. [Online], https://jatim.antaranews.com/berita/245508/kediri-dapat-penghargaan-kota-cerdas-2017-dari-wapres-jusuf-kalla, tanggal akses: 28-Nov-2019.

[2] M.D.A.C. Hasibuan dan Lusiana, "Pencarian Rute Terbaik pada Travelling Salesman Problem (TSP) Menggunakan Algoritma Genetika pada Dinas Kebersihan dan Pertamanan Kota Pekanbaru," *Jurnal SATIN - Sains dan Teknologi Informasi*, Vol. 1, No. 1, hal. 35-46, Jun. 2015.

[3] F. Glover, G. Gutin, A. Yeo, dan A. Zverovich, "Construction Heuristics for The Asymmetric TSP," *European Journal of Operational Research*, Vol. 129, No. 3, hal. 555–568, Mar. 2001.

[4] I.K. Gupta, S. Shakil, dan S. Shakil, "A Hybrid GA-PSO Algorithm to Solve Traveling Salesman Problem," *Computational Intelligence: Theories, Applications and Future Directions*, Vol. 798, hal. 453–462, Agustus 2018.

[5] X.H. Zhi, X.L. Xing, Q.X. Wang, L.H. Zhang, X.W. Yang, C.G. Zhou, dan Y.C. Liang, "A Discrete PSO Method for Generalized TSP Problem," *Proc. International Conference on Machine Learning and Cybernetics*, 2004, hal. 2378-2383.

[6] N. Chandekar dan M.J. Pillai, "In a Comparative Study of GA and ACO for Solving Travelling Salesman Problem," *International Journal of Mechanical and Production Engineering*, Vol. 5, No. 11, hal. 34-37, Nov. 2017.

[7] A.A. Ismail dan S. Herdjunanto, "Penerapan Algoritma Ant System dalam Menemukan Jalur Optimal pada Traveling Salesman Problem (TSP) dengan Kekangan Kondisi Jalan," *JNTETI*, Vol. 1, No. 3, hal. 43-48, Nov. 2012.

[8] A. Bajpai dan R. Yadav, "Ant Colony Optimization (ACO) for the Traveling Salesman Problem (TSP) Using Partitioning," *International Journal of Scientific & Technology Research*, Vol. 4, No. 9, hal. 376-381, Sep. 2015.

[9] B. Freisleben dan P. Merz, "Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems," *Proc. IEEE Conference on Evolutionary Computation*, 1996, hal. 616-621.

[10] A. Maria, E.Y. Sinaga, dan M.H. Iwo, "Penyelesaian Masalah Travelling Salesman Problem Menggunakan Ant Colony Optimization (ACO)", *Jurnal Media Informatika*, Vol. 6, hal. 1-5, 2008.

[11] X. Chen, Y. Zhou, Z. Tang, dan Q. Luo, "A Hybrid Algorithm Combining Glowworm Swarm Optimization and Complete 2-Opt Algorithm for Spherical Travelling Salesman Problems," *Applied Soft Computing Journal*, Vol. 58, hal. 104–114, September 2017.

[12] A. Izzah, B.A. Nugroho, W.F. Mahmudy, F.A. Bachtiar, T.A. Cinderatama, dan Y.A. Sari, "Convergence Analysis in Swarm Intelligence for City Tour Optimization," *Proc. International Seminar on Research of Information Technology and Intelligent System*, 2019, hal. 109.