

IMPLEMENTASI JALUR TERPENDEK MENGGUNAKAN ANT COLONY OPTIMIZATION

IMPLEMENTATION OF SHORTEST PATH USING ANT COLONY OPTIMIZATION

Arief Kelik Nugroho^{1*}, Ipung Permadi¹

Email: arief.nugroho@unsoed.ac.id¹, ipung.permadi@unsoed.ac.id¹

¹Jurusan Teknik Informatika, Fakultas Teknik, Universitas Jenderal Soedirman, Purwokerto

Abstrak— Dengan mengadopsi perilaku semut dalam mencari makanan dapat diterapkan dalam proses pencarian rute terpendek dengan mempertimbangkan parameter-parameter pada *Ant colony*. Secara alami, semut mampu menemukan jalan terdekat dari ke sumber makanan dengan mekanisme meninggalkan jejak untuk setiap jalur yang dilewatinya. Intensitas *pheromone* digunakan dalam proses pencarian rute dengan mempertimbangkan faktor-faktor lain diantaranya adalah jarak dari titik awal ke titik tujuan. Jejak tersebut digunakan untuk mengukur banyaknya semut yang melewati rute. Proses *update pheromone* dapat dilakukan baik secara lokal ataupun pada *pheromone* global yang mempengaruhi performa semut dalam mencari jalur adalah nilai visibilitas dan intensitas *pheromone*. Hasil pencarian nilai optimal menggunakan algoritma *Ant colony Optimization* (ACO) dengan variabel yang ditetapkan diperoleh nilai terbesar 4.726.

Kata kunci — Koloni, Semut, *Pheromon*, Rute

Abstract— By adopting ant behavior in searching for food can be applied in the process of finding the shortest route by considering the parameters in Ant colony. Naturally, ants are able to find the closest way to the food source by leaving a trail for each path through. Pheromone intensity is used in the process of finding a route by considering other factors including the distance from the starting point to the destination point. The trail is used to measure the number of ants that pass through the route. The process of updating pheromone can be done either locally or on global pheromone that affects the ant's performance in finding the path is the value of pheromone visibility and intensity. The optimal value search results using the *Ant colony Optimization* (ACO) algorithm with the specified variables obtained the largest value of 4.726.

Keywords — Colony, Ant, Pheromone, Trail

I. PENDAHULUAN

Optimasi dapat diterapkan untuk memperoleh nilai tertinggi dan terendah dari suatu permasalahan. Sebagai contoh, pada kasus pencarian *benefit* maka yang akan diambil adalah nilai tertinggi, sedangkan untuk permasalahan risiko kredit dan pencarian rute terdekat maka yang akan diambil adalah nilai terkecil. Pencarian rute terdekat sangat populer dan bisa dijumpai pada *Traveling Salesman Problem* (TSP) [1]. Pada TSP, variabel yang dihitung adalah jarak antara titik atau objek yang akan dikunjungi sebagai bobot dari *graph*. Pada implementasinya pencarian

rute terpendek bisa menggunakan algoritma optimasi seperti algoritma genetika, algoritma *Tabu Search*, algoritma *dijkstra* atau algoritma *Ant Colony Optimization* [1],[2],[3]. Dalam kenyataannya ketika melakukan sebuah perjalanan, jarak bukanlah satu satunya tolak ukur yang dapat dihitung.

Dalam mencari solusi yang optimal dan masalah terdistribusi. Perilaku semut dalam mencari makanan dengan menggunakan zat kimia yang disebut *pheromone*. *Pheromone* digunakan untuk menandai jalur di tanah, misalnya, jalur dari sumber makanan ke sarang. Dengan merasakan jejak *pheromone*, pengumpul dapat mengikuti jalan menuju makanan

yang ditemukan oleh semut lain. Jejak kolektif ini dan perilaku mengikuti jejak di mana semut dipengaruhi oleh jejak kimia yang ditinggalkan semut lain [4], [5], [6].

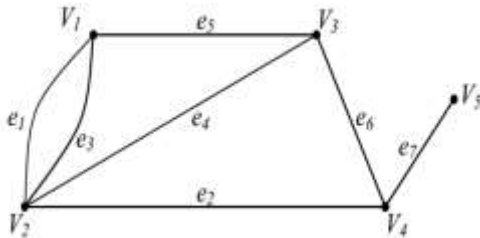
Penerapan algoritma semut berbasis web yang dikembangkan diharapkan memberikan solusi untuk penentuan nilai optimal.

II. TINJAUAN PUSTAKA

A. Pemodelan dengan Graf

Model merupakan gambaran sederhana dalam dimensi yang lebih kecil dibanding dengan objek aslinya. Contoh : model rumah, model pesawat dan lainnya.

Graf atau $G(V,E)$ terdiri dari dua (2) buah himpunan vertek atau simpul dan himpunan *edge* atau sisi. Himpunan V bukan himpunan kosong dan unsur-unsurnya disebut simpul. Sedangkan unsur-unsur himpunan E adalah sisi. Sedemikian sehingga tiap sisi menghubungkan dua buah simpul, seperti tampak pada Gambar-1.



Gambar-1. Graf.

B. Algoritma Semut dan Jarak minimum

Algoritma semut merupakan algoritma metaheuristik yang dapat diterapkan dalam model pencarian. Metaheuristik merupakan konsep algoritmik yang dapat diterapkan untuk mendefinisikan metode heuristik yang berlaku untuk serangkaian masalah yang berbeda [1].

Proses adopsinya dengan menggunakan salah satu perilaku semut dalam kegiatan pencarian makanan adalah menemukan jalur terpendek. Pola perilaku ini yang mendasari dalam bidang komputer untuk mengembangkan suatu algoritma untuk menangani masalah pemilihan rute terpendek [8], [9]. Bidang algoritma semut mempelajari model yang diturunkan dari hasil observasi perilaku semut.

C. Penguapan Pheromone

Penguapan jejak *pheromone* dapat dilihat sebagai mekanisme eksplorasi yang menghindari konvergensi cepat semua semut menuju jalur

suboptimal. Penurunan intensitas *pheromone* mendukung eksplorasi jalur yang berbeda selama seluruh proses pencarian [10].

Intensitas *pheromone* semut dari titik ke titik berikutnya dimungkinkan ada perubahan dalam masa penguapan dan perbedaan jumlah semut yang melintasi. Semakin banyak intensitas *pheromone* yang terdapat pada lintasan tersebut, maka semakin banyak pula semut yang melalui lintasan [1].

1) Aturan transisi

Probabilitas kunjungan semut dari titik i ke titik j digunakan persamaan 2:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} & \text{jika } j \in N_i^k \\ 0 & \text{lainnya} \end{cases} \quad (1)$$

dimana,

τ_{ij} = jumlah *pheromone*

i = titik awal

j = titik akhir

α = parameter pengendali bobot *pheromone*

β = parameter pengendali jarak

N_i^k = titik yang belum dikunjungi semut

2) Mengupdate intensitas Pheromone

Proses perubahan nilai intensitas *pheromone* terjadi jika telah terbentuk suatu lintasan. Perubahan intensitas *pheromone* dilakukan dengan menggunakan persamaan

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \Delta\tau_{ij} \quad (2)$$

$$\Delta\tau_{ij} = \sum_k^m \Delta\tau_{ij}^k \quad (3)$$

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{jika } (i,j) \in \text{tur pada tabu}_k \\ 0 & \end{cases} \quad (4)$$

dimana,

ρ = intensitas *pheromone*

m = jumlah semut

L_k = panjang tur semut

Q = jumlah *pheromone* yang tersimpan

III. HASIL DAN PEMBAHASAN

Proses penerapan *Ant Colony Optimization* (ACO) berbasis web terdapat beberapa tahapan. Tahap pertama fase perencanaan yang berisi hasil dari aktivitas mengidentifikasi masalah, mengumpulkan data, menganalisis data dan mengidentifikasi kebutuhan penelitian.

Pada tahap ini dihasilkan rincian dari setiap aktivitas yang dilakukan, mulai dari identifikasi masalah yaitu bagaimana membuat aplikasi pemilihan objek wisata berdasarkan biaya transportasi terendah, pengumpulan data yang didapatkan dari Dinas Perhubungan Purbalingga, menganalisis data biaya angkutan umum dan objek wisata Kabupaten Purbalingga serta menerapkan data tersebut pada perhitungan algoritma ACO, melakukan perhitungan pemilihan objek wisata dengan biaya terendah menggunakan algoritma ACO dan mendefinisikan kebutuhan penelitian yaitu laptop.

A. Proses Analisis

Di dalam tahapan analisis ini meliputi analisis teori algoritma ACO, yaitu berupa data biaya transportasi objek pariwisata di Purbalingga dan hasil perhitungan ACO terhadap variabel biaya pada rute pariwisata di Purbalingga. Serta analisis sistem untuk kebutuhan pembangunan aplikasi.

Pada tahapan ini akan menjelaskan tentang menentukan suatu *tour* terbaik berdasarkan biaya transportasi terendah pada objek wisata di Purbalingga, yang akan dipecahkan dengan menggunakan algoritma ACO. Perhitungan algoritma ACO dapat dibagi dalam beberapa tahapan seperti berikut:

1) Identifikasi d_{ij}

d_{ij} adalah jarak dari titik awal i ke titik tujuan j . pada penelitian ini, variabel d_{ij} diganti dengan c_{ij} yaitu biaya dari titik i ke titik j . berikut data c_{ij} dalam bentuk matriks dapat dilihat pada Tabel-1.

Tabel-1. Matriks Biaya antar *Node* Objek Wisata

i/j	1	2	3	4
1	0	2.318	2.83	3.847
2	2.318	0	1.381	2.052
3	2.83	1.381	0	1.027
4	3.847	2.052	1.027	0

Keterangan *node* objek :

1. x_1
2. x_2
3. x_3
4. x_4

2) Inisialisasi nilai

Proses perhitungan dengan menggunakan algoritma semut adalah dengan mengatur parameter-parameter sebagai berikut :

```

$alpha      =$_post["alpha"];
$beta       =$_post["beta"];
$ro         =$_post["ro"];
$tij_awala  =$_post["beta"];
$phe        =$_post["phe"];
$byk_semut  =$_post["Jumlah Semut"];
    
```

Visibilitas antar titik i ke titik j dengan menggunakan $(\eta_{ij}) = 1/d_{ij}$ dapat dilihat pada Tabel-2.

Tabel-2. Visibilitas antar Titik

i/j	1	2	3	4
1	0	0.01	0.01	0.01
2	0.01	0	0.01	0.01
3	0.01	0.01	0	0.01
4	0.01	0.01	0.01	0

Sehingga diperoleh hasil perhitungan pada Tabel-3

Tabel-3. Hasil perhitungan

i/j	1	2	3	4
1	0	0.431406	0.353357	0.259943
2	0.431406	0	0.724113	0.487329
3	0.353357	0.724113	0	0.97371
4	0.259943	0.487329	0.97371	0

Nilai parameter visibilitas (η) dan intensitas *pheromone* (τ) dalam hal ini merupakan variabel yang dapat mempengaruhi kinerja semut dalam pemilihan ke titik berikutnya.

3) Menentukan Jumlah Semut

Pada tahap ini adalah penentuan jumlah semut yang akan diletakkan pada tiap *node* untuk melakukan *tour*. Jumlah semut disimbolkan dengan (m) dan diberi nilai 4 sesuai dengan jumlah *node* objek wisata.

4) Membuat *Tabulist*

Semut yang telah diletakkan pada masing-masing *node* mulai melakukan *tour* dan menghasilkan sebuah *path* atau rute. Jumlah *tabulist* akan sebanding dengan jumlah semut, maka *tabulist* akan berjumlah 4.

A. Proses Perhitungan

Proses berikutnya adalah dengan menghitung nilai probabilitas jumlah semut ke titik i ke titik j , kemudian menghitung probabilitas dari titik 1 ke titik berikutnya dengan menggunakan persamaan 1

$$\begin{aligned} & \sum [\tau_{(r,u)}]^\alpha \cdot [\eta_{(r,u)}]^\beta \\ & = (0.01 * 1) + (0.01 * 0.431406) + (0.01 * 0.353357) \\ & \quad + (0.01 * 0.259943) \\ & = 0.01045 \end{aligned}$$

Dari hasil perhitungan probabilitas titik 1 menuju ke titik berikutnya diperoleh hasil :

$$\begin{aligned} \text{Titik 1} &= 0.00; \\ \text{Titik 2} &= 1.00 \cdot (0,431406)1.00 / 0,01045 = \\ & \quad 0.412829; \\ \text{Titik 3} &= (0.01)1.00 \cdot (0,353357)1.00 / 0,01045 = \\ & \quad 0.338141 \\ \text{Titik 4} &= (0.01)1.00 \cdot (0,259943)1.00 / 0,01045 = \\ & \quad 0.248749 \end{aligned}$$

Probabilitas kumulatif dari ke empat titik adalah 0,000;0,413; 0.751; 1,000.

Setelah diperoleh hasil probabilitas membangkitkan bilangan acak dengan nilai 0,68 dan memeriksa $q_k - 1 < r \leq q_k \rightarrow$ titik terpilih berdasar bilangan acak adalah titik 3.

Daftar kunjungan = 1; 3, setelah dipeoleh data kemudian melakukan pembaruan lokal *pheromone* menggunakan persamaan (3):

$$\begin{aligned} \tau(t,v) &\leftarrow (1 - \rho) \cdot \tau(t,v) + \Delta\tau(t,v) \\ \Delta\tau(t,v) &= \frac{1}{L_{nn}C} \end{aligned}$$

Maka diperoleh pembaharuan *pheromone* lokal dipeoleh hasil sebagai berikut :

$$\begin{aligned} \Delta\tau(1,3) &= \frac{1}{2,830.4} = 0.088 \\ \tau(1,3) &\leftarrow (1 - 0.50) \cdot 0,01 + 0.088 \\ \tau(1,3) &\leftarrow 0.93 \end{aligned}$$

Dengan nilai 0.93. Setelah dilakukan perubahan intensitas *pheromone* lokal, kemudian mencari nilai probabilitas dari titik 3 ke titik selanjutnya dengan :

$$\begin{aligned} & \sum [\tau_{(r,u)}]^\alpha \cdot [\eta_{(r,u)}]^\beta \\ & = (0.01 * 0.353357) + (0.01 * 0.724113) + (0.01 * 0.97371) \\ & = 0.0205 \end{aligned}$$

```

$jumlah = 0;
$total = array();
$city = array();
$prob_tiap_kota = 0;

for ($m=1, $m <=4; $m++){
  if(!in_array($m, $tabulist))
    $prob_tiap_kota +=
      $local_feromon[$sterpilih][$m]*$nij*[$sterpilih]
        [$m]
}

$prob_tiap_kota = round($prob_tiap_kota, 5);
echo "<p><b>Probabilitas Kota ". $sterpilih. " ke
  setiap kota = </b>". $prob_tiap_kota "</p>";

for ($i=1, $i <=4; $i++){
  if(!in_array($m, $tabulist))
    $kota =
      round($local_feromon[$sterpilih][$i]*$nij[$sterpilih]
        [$i]/$prob_tiap_kota, 6);

  if(in_array($i, $tabulist)){
    $kota = 0;
  }

  $jumlah = $kota + $jumlah
  array_push($city, $kota);
  array_push($total, $jumlah);
}

```

Gambar-2. Potongan kode program.

Proses perhitungan nilai probabilitas dilakukan dengan mencari hasil probabilitas terbesar dan *pheromone* terkecil pada jalur untuk dijadikan rute terbaik dengan potongan kode program seperti tampak pada Gambar-2. Berikut tahapannya:

- Membangkitkan bilangan acak yaitu = 0.48
- 1) Periksa $q_k - 1 < r \leq q_k \rightarrow$ Titik yang dipilih adalah dari hasil membangkitkan bilangan adalah 2
- 2) Daftar kunjungan = 1;3;2
- 3) Melakukan pembaruan lokal *pheromone*

$$\Delta\tau(3,2) = \frac{1}{1,381.4} = 0.181$$

$$\tau(3,2) \leftarrow (1 - 0.50) \cdot 0.01 + 0.181$$

$$\tau(3,2) \leftarrow 0.187$$
- 4) Titik tujuan selanjutnya hanya ke titik 4
Daftar kunjungan semut = 1;3;2;4

Setelah melakukan kunjungan kemudian dilakukan pembaruan lokal *pheromone* menggunakan persamaan (3):

$$\begin{aligned} \tau(2,4) &\leftarrow (1 - 0,50) \cdot 0,01 + 0,1218 \\ \tau(2,4) &\leftarrow 0,1268 \end{aligned}$$

Iterasi dilakukan dengan sampai diperoleh titik i ke titik j . Cetak hasil Hasil Rute Terbaik diambil berdasarkan probabilitas terbesar dan *pheromone* terkecil yaitu pada tabulist 2.

Iterasi Ke-1

Pada semut 1 titik awal adalah 1 dengan probabilitas masing-masing titik = 0.010 dengan rincian pencarian dari titik i ke j

Tabel-4. Probabilitas ij

i	j			
	1	2	3	4
1	0	0.412829	0.75097	0.999719

Nilai *Roulette* (acak) = 0.954

Dari proses perhitungan nilai *roulette* maka titik yang terpilih berada pada nilai rentang titik ke 4, sehingga titik yang terpilih adalah titik ke 4, jalur terpilih adalah 1;4

Tabel dari Titik 4, dengan rincian pencarian dari titik i ke j

Tabel-5. Probabilitas ij

i	j			
	1	2	3	4
1	0	0.333559	0.666468	0

Dengan membangkitkan nilai acak yang diset pada roulette 0.271 terpilih titik 2 (Tabel-6), sehingga jalur 1;4;2

Tabel-6. Dari Titik 2

i	j			
	1	2	3	4
1	0	0	1.000	1.000

Dengan membangkitkan nilai acak yang diset pada roulette 0.271 Nilai *Roulette* (Acak) = 0.73, terpilih 2. Jalur terpilih untuk semut 1 adalah 1;4;3;2.

Tabel-7. Update Pheromone Global

i/j	1	2	3	4
1	0.0025	0.16222	0.04666	0.03499
2	0.0025	0.0025	0.19881	0.0025
3	0.0025	0.0025	0.0025	0.23001
4	0.0025	0.06341	0.0025	0.0025

Iterasi terus berlanjut sampai dengan iterasi maksimum dan dihasilkan tabulist. Hasil perhitungan algoritma ACO pada objek wisata pada Tabel-8.

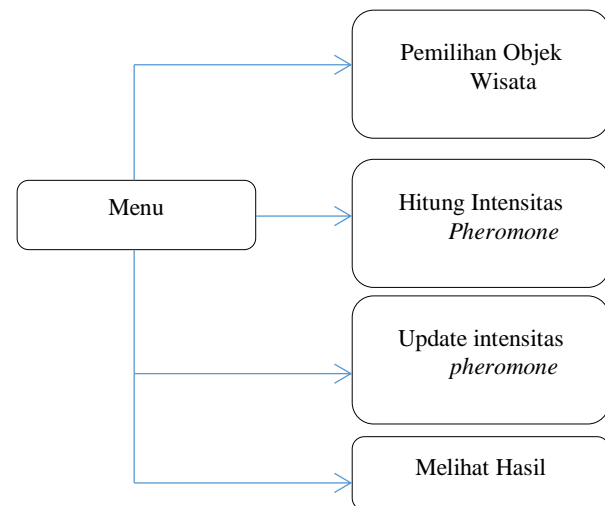
Tabel-8. Hasil Perhitungan

Tabulist	Jalur	Nilai
1	1-4-2-3	7,280
2	1-2-3-4	4,726
3	1-4-3-2	6,255
4	1-3-4-2	5,909

Jarak terpendek dari keseluruhan iterasi adalah 4.726 km dengan rute jalur 1; 2; 3; 4

B. Analisis Sistem

Pada tahapan Design Phase menggunakan *Object Oriented Design* (OOD) berupa desain proses dan desain antarmuka aplikasi, seperti tampak pada Gambar-3. Dimana desain proses menggambarkan proses aplikasi secara keseluruhan [12], dimulai dari tampilan splash saat aplikasi pertama dibuka, pemilihan menu hingga keluar dari aplikasi.



Gambar-3. Desain Proses Aplikasi Pemilihan Objek Wisata.

1) Implementasi

Dilakukan dengan menkodekan ke dalam bahasa pemrograman berbasis *website*. Proses dilakukan dengan :

a. Instalasi Sistem

Instalasi sistem terdiri dari instalasi perangkat lunak dan perangkat keras. Perangkat lunak yang dibutuhkan agar dapat menjalankan aplikasi ini adalah xampp versi 7.1.7. Agar aplikasi dapat berjalan dengan baik dibutuhkan perangkat keras laptop dengan minimal spesifikasi sebagai berikut. Prosesor i3-7100 CPU 3.90 GHz dengan RAM 4 GB

b. Perhitungan jarak

Proses perhitungan jarak pada sistem dapat diatur sesuai dengan kebutuhan *user* sesuai pada Gambar-5

	Kota 1	Kota 2	Kota 3	Kota 4
Kota 1	1	2.76	2.01	3.47
Kota 2	2.76	0	1.81	2.82
Kota 3	2.01	1.81	0	1.07
Kota 4	3.47	2.82	1.07	0

Gambar-5. Jarak Antar Titik

alpha: 1
 beta: 1
 rho: 0.5
 feromon awal: 0.01
 Jumlah Semut: 3
 Maksimal Iterasi: 10

Cari Rule

Gambar-6. Update variabel penentu hasil perhitungan.

Intensitas Feromon

Iterasi Ke-1

Semut Ke-1

Probabilitas Kota 1 ke setiap kota = 0.01045
 Probabilitas Kumulatif = 0 0.412629 0.75097 0.999719
 Bilangan Random = 0.096
 Kota terpilih = 4
 Probabilitas Kota 4 ke setiap kota = 0.01461
 Probabilitas Kumulatif = 0 0.333559 1.000027 1.000027
 Bilangan Random = 0.517
 Kota terpilih = 3
 Probabilitas Kota 3 ke setiap kota = 0.00724
 Probabilitas Kumulatif = 0 1.000156 1.000156 1.000156
 Bilangan Random = 0.326
 Kota terpilih = 2
 Jarak ke kota terpilih = 6.255
 Jarak terbaik = 6.255

Semut Ke-2

Probabilitas Kota 1 ke setiap kota = 0.01045
 Probabilitas Kumulatif = 0 0.412629 0.75097 0.999719
 Bilangan Random = 0.932
 Kota terpilih = 4
 Probabilitas Kota 4 ke setiap kota = 0.01461
 Probabilitas Kumulatif = 0 0.333559 1.000027 1.000027
 Bilangan Random = 0.327
 Kota terpilih = 2
 Probabilitas Kota 2 ke setiap kota = 0.00724
 Probabilitas Kumulatif = 0 0 1.000156 1.000156
 Bilangan Random = 0.524
 Kota terpilih = 3
 Jarak ke kota terpilih = 7.26
 Jarak terbaik = 6.255

Gambar-7. Perhitungan Algoritma ACO

Iterasi Ke-10

Hasil Akhir Pencarian

Jalur Terbaik : 1 - 2 - 3 - 4

Jarak Jalur Terbaik: 4.726

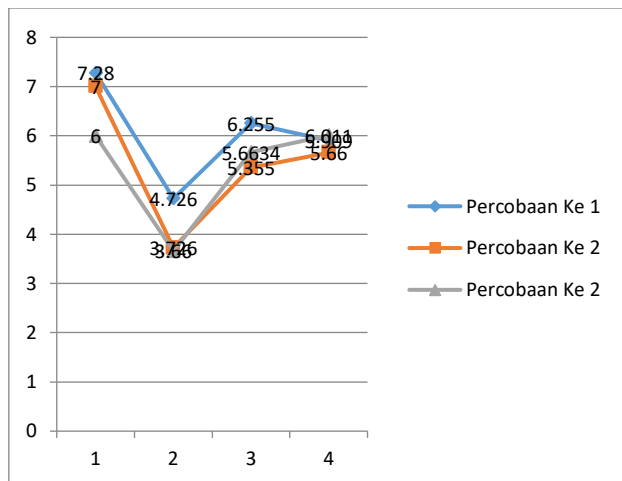
Gambar-8. Hasil Akhir Iterasi.

Gambar-6 menunjukkan proses pengaturan parameter semut untuk mendapatkan hasil optimasi pada proses pencarian jalur. Pada Gambar-7, masing-

masing semut akan mencari rute dengan jarak terdekat dari titik awal semut ke titik tujuan semut. Data hasil perhitungan disimpan di tabu dan akan

dibandingkan dengan semua semut sehingga diperoleh hasil rute terpendek yang ditunjukkan pada Gambar-8.

Performa semut dilakukan dengan merubah parameter semut diperoleh perbedaan nilai pada masing-masing percobaan. Tujuan mengubah parameter adalah untuk mendapatkan kriteria terbaik pada masalah pencarian. Performa semut dengan masing-masing nilai ditunjukkan pada Gambar-9.



Gambar-9. Hasil Akhir Iterasi dengan bebrapa percobaan.

IV. PENUTUP

A. Kesimpulan

Dari hasil pencarian yang dilakukan maka dapat disimpulkan dalam menentukan nilai optimal menggunakan algoritma ACO dengan variabel dapat dilakukan dengan menentukan probabilitas terbesar dan nilai *pheromone* terkecil berdasarkan 8 tahapan perhitungan algoritma ACO, yakni identifikasi d_{ij} , inisialisasi parameter awal, menentukan jumlah semut, membuat *tabulist*, menghitung probabilitas dan *pheromone*, lakukan iterasi, kemudian lakukan *update pheromone* dan cetak hasil rute terbaik.

B. Saran

Pembentukan rute dilakukan dengan mempertimbangkan parameter lain selain jarak, sehingga dapat memaksimalkan algoritma *Ant Colony Optimizaton*. Dengan menggunakan probabilitas pada intensitas *pheromone* pada masing-masing semut untuk dijadikan sebagai penunjuk untuk semut yang lain.

DAFTAR PUSTAKA

- [1]. Dorigo M, Stutzle T. *Ant Colony Optimizaton*. 2004:Cambridge: MIT Press.
- [2]. Arief KN, Ipung P. Ant Colony Optimization Untuk Menyeleksi Fitur Dan Klasifikasi Artikel. 2019; *Jurnal Simetris.*, Vol 10, No 1.Hal.223-232
- [3]. Singh R G. Metha, Sonigoswami, Katiyar S. Implementation of travelling salesman problem using ant colony optimization. 2014.*Journal of Engineering Research and Applications*, vol. 4, no. 6, p. 63-67.
- [4]. Idris H, Ezugwu A E, Junaidi SB, Adewumi A O.An improved ant colony optimization algorithm with fault tolerance for job scheduling in grid computing systems. 2017; *PLoS ONE*, vol. 12, no. 5.
- [5]. Sobri A , Mita A S, Corina K. Ant Colony Optimization With Double Selections For Solving Integrated Scheduling Problem In Manufacturer.2019; *Journal Of Engineering And Management In Industrial System*.Vol. 7 No. 1.
- [6]. Arief KN, Dadang I. Algoritma Iterative Dichotomizer 3 (ID3) Pengambilan Keputusan.2015;*Dinarek Unsoed*,Vol. 11,No 2.
- [7]. Gao W. *Improved ant colony clustering algorithm and its performance study*. 2016;Computational Intelligence and Neuroscience.
- [8]. Asma S, Azzam S. Analysis of Ant Colony Optimization Algorithm solutions for Travelling Salesman Problem. 2018.*International Journal of Scientific & Engineering Research Volume 9, Issue 2. ISSN 2229-5518*.
- [9]. Ahmad SR, Abu Bakar A, Yaakub MR. Ant colony optimization for text feature selection in sentiment analysis. *Intelligent Data Analysis*. 2019; Jan 1;23(1):133-158.
- [10]. Nwamae, Believe B. , Kabari, Ledisi G. Solving Travelling Salesman Problem(TSP) Using Ant Colony Optimization(ACO).2018. *International Journal Of Engineering Research & Technology (IJERT) Volume 07*.
- [11]. Riadi ICJ. *Cognitive Ant Colony Optimization: A New Framework in Swarm Intelligence*. 2014. :University of Salford:Manchester.
- [12]. Nugroho, AK..2018. *Image Quantization in Psoriasis Using K-Mean Clustering*.Prosiding Seminar Nasional Teknologi Informasi dan Kedirgantaraan :SENATIK 2018.Vol. IV.ISBN 978-602-52742-0-6.

