

PENERAPAN ALGORITMA *ANT COLONY OPTIMIZATION* UNTUK PENCARIAN RUTE TERPENDEK LOKASI WISATA (STUDI KASUS WISATA DI KOTA PALEMBANG)

Daniel Udjulawa¹, Serly Oktarina²

¹Program Studi Informatika Universitas MDP, Palembang

²Program Studi Ilmu Komputer Universitas Sumatera Selatan, Palembang

E-mail: daniel@mdp.ac.id

Abstrak

Lokasi wisata di kota Palembang secara geografis memang tidak terlalu jauh antar satu tempat dengan tempat lainnya. Namun bagi wisatawan yang berkunjung pertama kali di kota Palembang, biasanya akan mencari informasi sejauh mana lokasi wisata satu ke tempat wisata yang lainnya dan berapa jaraknya kalau mereka berada di suatu titik lokasi sebagai pusat nya dan akan menuju ke lokasi yang lain. Penyajian informasi rute terdekat dengan sistem informasi berbasis website akan sangat membantu bagi wisatawan agar mudah dalam mencari lokasi wisata tertentu. Penelitian ini dititik beratkan pada penerapan algoritma menggunakan PHP yang berupa website dan digunakan dalam menentukan rute terdekat lokasi wisata. Algoritma yang digunakan adalah algoritma Ant Colony Optimization atau AntCO. Algoritma ini merupakan algoritma heuristic yang telah terbukti diterapkan ke sejumlah masalah Travelling Salesman Problem (TSP). Tujuannya untuk mengetahui nilai terpendek rute dari satu lokasi tertentu ke lokasi lainnya dengan menerapkan algoritma tersebut menggunakan PHP. Hasil penerapan algoritma tersebut di dapatkan bahwa menggunakan algoritma Ant Colony Optimization menghasilkan nilai rute terpendek 205,12025621393. Perhitungan tersebut berdasarkan penggunaan satu titik awal, dimana titik awal tersebut merupakan pusat dari penentuan rute ke berbagai lokasi wisata

Kata kunci: *wisata,PHP,AntCO, heuristic, TSP*

1. Pendahuluan

Kota Palembang merupakan pusat kota provinsi di Sumatera Selatan. Disamping itu kota ini merupakan kota tertua di Sumatera Selatan, karena didirikan pada tanggal 17 Juni 683 M. Selain menjadi kota tertua, yang menarik dari kota Palembang adalah adanya ikon dari kota ini yakni jembatan Ampera. Banyak destinasi wisata yang dapat dikunjungi oleh wisatawan. Tak hanya jembatan Ampera saja yang menjadi destinasi andalan saat berkunjung ke kota Palembang, namun masih banyak lokasi wisata lainnya yang dapat dikunjungi wisatawan ketika

berkunjung ke kota ini. Wisata sejarah maupun budaya dapat ditemukan di sini, wisatawan akan dibuat tertarik untuk datang memperoleh pengetahuan baru. Kondisi saat penelitian ini dibuat, pariwisata di kota Palembang mulai perlahan bangkit dengan kunjungan wisatawan sebesar 80 %.

Kedatangan wisatawan memang selalu diharapkan oleh pihak dinas pariwisata, namun yang menjadi kendala adalah kurangnya informasi akan peta lokasi, dan informasi jarak dari lokasi wisata satu ke tempat lokasi wisata lainnya bagi wisatawan yang berkunjung ke kota Palembang. Informasi yang diberikan akan membuat

wisatawan menjadi lebih mudah menentukan lokasi mana yang dituju untuk berwisata.

Teknologi tidak akan bisa berhenti berkembang di era seperti sekarang ini. Telepon genggam saja sudah sangat beragam macamnya. *Gadget* telah menjadi alat komunikasi yang sangat dibutuhkan oleh manusia. Pengetahuan terkait destinasi wisata dapat ditemukan dengan *handphone*, destinasi wisata sendiri merupakan lokasi dimana tempat tersebut menjadi tujuan untuk melakukan kegiatan berwisata saat memiliki waktu luang.

Berkaitan dengan Teknologi dan tempat wisata tersebut, maka diperlukan sebuah aplikasi yang dapat membantu wisatawan dalam mendapatkan informasi lokasi dan rute terpendek dari tempatnya berada. Dalam aplikasi tersebut perlu menerapkan algoritma, yang menghitung rute terpendek dari satu lokasi. Untuk hal tersebut penelitian ini menggunakan algoritma yang menghitung rute terpendek tersebut yakni algoritma *Ant Colony Optimization* atau yang sering disingkat AntCO.

AntCO merupakan algoritma *heuristic* yang telah terbukti diterapkan ke sejumlah masalah *Travelling Salesman Problem* (TSP) dan mampu menemukan jalur terpendek dengan baik. Terbukti dengan penelitian-penelitian sebelumnya. Penelitian yang dilakukan [1], yaitu menentukan jalur evaluasi terpendek pada industri plastik dengan menggunakan AntCO. Pada penelitian ini digunakan *pheromone* awal sebesar 0,0098. Hasil penentuan parameter yang akan digunakan, meliputi *number of iterations* yaitu 500, *number of ant* sebesar 100, nilai Alpha sebesar 1 serta Beta sebesar 5 dan *evaporation rate* sebesar 0.5. Dari hasil *optimasi cost* yang paling rendah adalah *cost 2* sehingga didapatkan rute evakuasi untuk ruang *Circullar Loom* menuju Koridor 3 dilanjutkan keluar melalui pintu *exit 4* sehingga menuju Titik kumpul 1. Semakin tinggi *cost* yang dihasilkan maka akan mengakibatkan rute yang dilewati menjadi jauh sehingga meningkatkan waktu

evakuasi.

Untuk penerapan Algoritma AntCO ini dibuatkan aplikasi yang berbasis web. Hal ini dilakukan agar algoritma AntCO dapat menjadi algoritma utama dalam aplikasi tersebut. Penyajian informasi rute lokasi wisata yang terdekat dan yang dibangun ini bertujuan agar memudahkan wisatawan dalam mencari lokasi wisata tertentu. Dengan alasan tersebut, maka dibuatlah aplikasi ini dengan menerapkan algoritma AntCO, dalam pencarian rute terpendek lokasi wisata.

2. Kajian Pustaka

2.1. Wisata

Wisata menurut Kamus Besar Bahasa Indonesia adalah bepergian bersama-sama (untuk memperluas pengetahuan, bersenang – senang, dst), bertamasya (Kamus Besar Bahasa Indonesia, n.d.). Selain itu, Wisata adalah kegiatan yang bertujuan untuk melakukan perjalanan atau sebagian kegiatan tersebut dilakukan secara sukarela dan bersifat sementara untuk menikmati objek atau daya tarik wisata. Sementara itu, jenis-jenis wisata yaitu wisata sejarah, wisata alam, wisata budaya, wisata buatan hingga beragam wisata minat khusus (Pleanggra, 2012). Di Kota Palembang, banyak sekali jenis obyek wisata yang terkenal yaitu Wisata Kuliner, Wisata Belanja, Wisata Budaya dan Sejarah, Wisata Alam, Wisata Rekreasi dan Hiburan.

2.2.PHP

PHP adalah bahasa pemrograman yang berjalan di server, karena itu dibutuhkan sebuah web server. Web server di windows bisa dibuat dengan berbagai macam cara. Ada yang menggunakan PHP secara manual dengan web server bawaan PHP, ada yang menggunakan WSL lalu menginstal LAMP Stack, ada yang menggunakan XAMPP, ada yang menggunakan Docker, dan lain sebagainya. Contoh *script* PHP :

```
<? // object.php
class Test
{ var $str = "Variabel Class";
function set_var($str)
```

```
{ $this->str = $str; }
}
$class = new Test;
echo$class->str;
$class->set_var ("Variabel Object");
echo("<br>$class->str"); ?>
```

2.3. Heuristic

Metode *Heuristic* merupakan metode alternatif yang berbasis kecerdasan. Metode ini merupakan sub bidang dari kecerdasan buatan yang digunakan untuk melakukan pencarian dan penentuan rute terpendek. Metode Heuristik ini merupakan metode yang menggunakan system pendekatan dalam melakukan pencarian dalam optimasi. Ada beberapa algoritma pada metode heuristik yang biasa digunakan dalam permasalahan optimasi, diantaranya Algoritma Genetika, *Ant Colony Optimization*, logika *Fuzzy*, jaringan syaraf tiruan, *Tabu Search*, *Simulated Annealing*, dan lain-lain (Mutakhirroh, I., Saptono, F., Hasanah, N., dan Wiryadinata, R., 2007).

2.4. Ant Colony Optimization

Algoritma semut diperkenalkan oleh *Moyson* dan *Manderick* dan secara meluas dikembangkan oleh *Marco Dorigo*. Algoritma semut adalah bioinspired metaheuristic, mempunyai sekelompok khusus yang berusaha menyamai karakteristik kelakuan dari serangga sosial, yaitu koloni semut. Kelakuan dari tiap pelaku dalam meniru kelakuan dari semut hidup dan bagaimana mereka berinteraksi satu dengan lainnya agar dapat menemukan sumber makanan dan membawanya ke koloni mereka dengan efisien. Selama berjalan tiap semut mengeluarkan *feromon*, dimana semut lainnya sensitif dengan *feromon* tersebut sehingga memberikan harapan untuk mengikuti jejaknya. Lebih atau kurang intensitasnya tergantung pada konsentrasi dari *feromon*. Setelah beberapa waktu, jalur terpendek akan lebih sering diikuti dan *feromon*nya menjadi jenuh.

Parameter-parameter yang digunakan dalam Algoritma Koloni Semut antara lain:

1. Intensitas jejak semut ($ij \tau$) dan

perubahannya ($ij \Delta \tau$). $ij \tau$ harus diinisialisasi sebelum memulai siklus. $ij \tau$ digunakan dalam persamaan probabilitas node yang akan dikunjungi. $ij \Delta \tau$ diinisialisasi setelah selesai satu siklus. $ij \Delta \tau$ digunakan untuk menentukan $ij \tau$ untuk siklus selanjutnya.

2. Tetapan siklus semut (Q), Q merupakan konstanta yang digunakan dalam persamaan untuk menentukan $ij \Delta \tau$. Nilai Q ditentukan oleh pengguna.

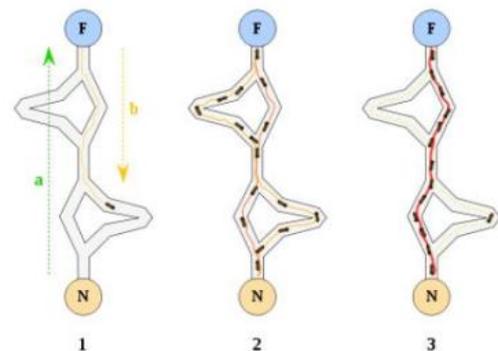
3. Tetapan pengendali intensitas jejak semut (α), α digunakan dalam persamaan probabilitas node yang akan dikunjungi yang berfungsi sebagai pengendali intensitas jejak semut. Nilai α ditentukan oleh pengguna.

4. Tetapan pengendali visibilitas (β) β digunakan dalam persamaan probabilitas node yang akan dikunjungi yang berfungsi sebagai pengendali visibilitas. Nilai β ditentukan oleh pengguna

5. Visibilitas antar kota (η_{ij}) η_{ij} digunakan dalam persamaan probabilitas node yang akan dikunjungi. Nilai η_{ij} merupakan hasil dari $1/d_{ij}$, .

6. Banyak semut (m) m merupakan banyak semut yang akan melakukan siklus dalam algoritma semut. Nilai m ditentukan oleh pengguna. Disini banyak semut diibaratkan sama dengan banyak lokasi wisata budaya yang dilalui.

7. Tetapan penguapan jejak semut (ρ) ρ digunakan untuk menentukan τ_{ij} untuk siklus selanjutnya. Nilai ρ ditentukan oleh pengguna.



Gambar 1. Ilustrasi rute yang dibentuk oleh koloni semut

Adapun langkah-langkah penyelesaian

menggunakan algoritma koloni semut adalah

1. Dari sarang, semut berkeliling secara acak mencari makanan kemudian dicatat jarak antar node yang semut lalui.
2. Ketika sampai ke makanan, total jarak dari tiap node yang telah ditempuh oleh semut, dijumlahkan untuk mendapatkan jarak dari sarang ke makanan.
3. Ketika kembali ke sarang, sejumlah pheromone ditambahkan pada jalur yang telah ditempuh berdasarkan total jarak jalur tersebut. Semakin kecil total jarak, maka semakin banyak kadar pheromone yang ditambahkan pada masing-masing busur pada jalur tersebut.
4. Untuk memilih busur mana yang harus dilalui berikutnya, dihitung menggunakan rumus

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta} \text{ untuk } j \in \mathcal{N}_i^k$$

dengan

$$\eta_{ij} = \frac{1}{d_{ij}}$$

5. Pada iterasi selanjutnya, busur-busur yang mengandung pheromone lebih tinggi akan dipilih sebagai busur yang harus ditempuh berikutnya berdasarkan probabilitas yang ada di langkah ke-4. Akhirnya diperoleh jalur optimal, yaitu jalur yang dibentuk oleh busur-busur dengan kadar pheromone tinggi.

2.5. Traveling Salesman Problem (TSP)

Traveling Salesman Problem adalah pencarian rute terpendek atau jarak minimum oleh seorang salesman dari suatu kota ke n-kota tepat satu kali dan kembali ke kota awal keberangkatan. TSP dapat diterapkan pada graph komplit berbobot yang memiliki total bobot sisi minimum,

dimana bobot pada sisi adalah jarak. Rute TSP ini memuat semua titik pada graph tersebut tepat satu kali. Proses optimalisasi ini dilakukan dengan memperhitungkan fungsi *heuristic* yang akan mempersempit ruang pencarian.

3. Metode Penelitian

3.1. Analisis Kebutuhan

Dalam tahap ini akan dilakukan analisa terhadap kebutuhan sistem dan pengumpulan data dengan melakukan studi literature dan wawancara. Tahapan ini akan menghasilkan dokumen user *requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan aplikasi. Dokumen inilah yang akan menjadi acuan dalam pembuatan aplikasi *website*.

3.2 Perancangan

Tahap ini merupakan proses untuk merancang sistem dengan menerapkan algoritma *Ant Colony Optimization* pada pencarian lokasi wisata yang ada di Palembang. Perancangan dilakukan dengan menentukan beberapa titik wisata yang ada di kota Palembang dan menentukan titik koordinat yang akan dijadikan titik awal. Masing-masing titik diberikan bobot dan hambatan yang sesuai dengan jarak dan kemacetan. Algoritma tersebut mencari jalur terpendek dari titik awal menuju titik akhir dengan memperhatikan bobot terkecil.

3.3 Implementasi

Tahap ini merupakan tahap dalam merancang tampilan data yang dibangun lalu diimplementasikan ke dalam pemograman PHP, sehingga didapat tampilan hasil data akhir yang diperoleh dalam bentuk website. Dilakukan juga proses coding yang merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer. Hasil dari penelitian ini merupakan aplikasi berbentuk *Website* akan disesuaikan dengan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai

maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

```

$awal = "Punti Kayu";
$akhir = "TPKS";
$nomor = 1;
$query1 = mysqli_query($koneksi, "SELECT jalur, a.nama as nama_awal, b.nama as nama_akhir, a.lat as lat_awal, b.lat as lat_akhir, a.long as long_awal, b.long as long_akhir from jalur where a.id = $awal and b.id = $akhir");
while ($data1 = mysqli_fetch_array($query1)) {
    $lat_1 = $data1['lat_awal'];
    $long_1 = $data1['long_awal'];
}

$query2 = mysqli_query($koneksi, "SELECT jalur, a.nama as nama_awal, b.nama as nama_akhir, a.lat as lat_awal, b.lat as lat_akhir, a.long as long_awal, b.long as long_akhir from jalur where a.id = $awal and b.id = $akhir");
while ($data2 = mysqli_fetch_array($query2)) {
    $lat_2 = $data2['lat_awal'];
    $long_2 = $data2['long_awal'];
}

$query3 = mysqli_query($koneksi, "SELECT jalur, a.nama as nama_awal, b.nama as nama_akhir, a.lat as lat_awal, b.lat as lat_akhir, a.long as long_awal, b.long as long_akhir from jalur where a.id = $awal and b.id = $akhir");
while ($data3 = mysqli_fetch_array($query3)) {
    $lat_3 = $data3['lat_awal'];
    $long_3 = $data3['long_awal'];
}

$query4 = mysqli_query($koneksi, "SELECT jalur, a.nama as nama_awal, b.nama as nama_akhir, a.lat as lat_awal, b.lat as lat_akhir, a.long as long_awal, b.long as long_akhir from jalur where a.id = $awal and b.id = $akhir");
while ($data4 = mysqli_fetch_array($query4)) {
    $lat_4 = $data4['lat_awal'];
    $long_4 = $data4['long_awal'];
}

$jarak = sqrt(pow($lat_1 - $lat_2, 2) + pow($long_1 - $long_2, 2));
$jarak2 = sqrt(pow($lat_2 - $lat_3, 2) + pow($long_2 - $long_3, 2));
$jarak3 = sqrt(pow($lat_3 - $lat_4, 2) + pow($long_3 - $long_4, 2));
$jarak4 = sqrt(pow($lat_4 - $lat_1, 2) + pow($long_4 - $long_1, 2));

$jarak_akhir = $jarak + $jarak2 + $jarak3 + $jarak4;

echo $jarak_akhir;
}
    
```

Gambar 2. Coding Implementasi AntACO (1)

```

<?php echo $jarak; ?>
<br>
<?php
echo $fn_1+$fn_2+$fn_3+$fn_4;
$pheromone = 1/($fn_1+$fn_2+$fn_3+$fn_4);
echo " " . $pheromone;
if ($nomor == 1){
    $max = $pheromone;
    $jarak_max = $jarak;
} else {
    if ($pheromone > $max){
        $max = $pheromone;
        $jarak_max = $jarak;
    }
}
$nomor++;
}
<?php echo $jarak_max; ?> <br> <?php echo $max; ?>
</div>
</div>
</div>
    
```

Gambar 3. Coding Implementasi AntCO (2)

3.4 Pengujian

Tahapan ini merupakan tahap pengujian dalam pembuatan aplikasi berbasis web menggunakan bahasa pemrograman PHP, dan menerapkannya algoritma *Ant Colony Optimization*. Tahapan ini dimulai dengan memasukan lokasi dan bobot dan menghubungkan melalui jalur tiap lokasi. Setiap jalur memiliki bobot masing-masing digunakan untuk menentukan hasil dari pengujian. Selanjutnya menentukan Titik awal dari jalur yang akan di cari jarak terpendeknya menggunakan algoritma *Ant Colony Optimization*

4. Hasil dan Pembahasan

4.1. Proses penerapan

Proses yang dilakukan dalam menerapkan algoritma AntCO ini di tuliskan dalam *Coding PHP* dan tampak pada Gambar 2 dan Gambar 3. Proses tersebut merupakan implementasi algoritma *Ant Colony Optimization (ACO)* ke PHP yang bertujuan untuk mencari jarak terdekat lokasi wisata di kota Palembang. Cara kerja dari algoritma ini adalah menentukan jarak terdekat dengan mengambil lokasi wisata awal dan akhir. Tampak pada proses tersebut Bahasa yang digunakan adalah Bahasa *Scripting PHP*.

4.2. Hasil

Hasil pengujian menggunakan Algoritma AntCO dapat dilihat pada gambar 4. Pada gambar tersebut menunjukkan hasil dari penerapan algoritma menggunakan PHP, sesuai *coding* pada gambar 2 dan 3 diatas.

```

Punti Kayu&Bukit Siguntang -> Bukit Siguntang&Kambang Iwak -> Kambang Iwak&Jembatan Ampera -> Jembatan Ampera&TPKS
250.12979846754 0.003997924302209
Punti Kayu&Bukit Siguntang -> Bukit Siguntang&Jembatan Ampera -> Jembatan Ampera&Kambang Iwak -> Kambang Iwak&TPKS
265.13621158484 0.0037716462568279
Punti Kayu&Kambang Iwak -> Kambang Iwak&Bukit Siguntang -> Bukit Siguntang&Jembatan Ampera -> Jembatan Ampera&TPKS
275.14956096822 0.0036343070456398
Punti Kayu&Kambang Iwak -> Kambang Iwak&Jembatan Ampera -> Jembatan Ampera&Bukit Siguntang -> Bukit Siguntang&TPKS
235.1264763792 0.0042530301793289
Punti Kayu&Jembatan Ampera -> Jembatan Ampera&Bukit Siguntang -> Bukit Siguntang&Kambang Iwak -> Kambang Iwak&TPKS
260.14975392125 0.0038439982665042
Punti Kayu&Jembatan Ampera -> Jembatan Ampera&Kambang Iwak -> Kambang Iwak&Bukit Siguntang -> Bukit Siguntang&TPKS
205.12025621393 0.0048751889182365

Punti Kayu&Jembatan Ampera -> Jembatan Ampera&Kambang Iwak -> Kambang Iwak&Bukit Siguntang -> Bukit Siguntang&TPKS
0.0048751889182365
    
```

Gambar 4. Hasil Pengujian AntCO

Lokasi awal dan akhir digunakan sebagai parameter pada metode AntCO, hasil Output dari ACO mendapatkan jalur terpendek dengan lokasi awalnya Pundi Kayu, lalu jalur selanjutnya ke Jembatan Ampera, Kambang Iwak, Bukit Siguntang dan berakhir di TPKS dengan nilai sebesar 205.12025621393. Hasil tersebut bisa dikatakan sebagai Output terpendek dari sejumlah iterasi. Adapun hasil lainnya yakni Awal Pundi Kayu, Bukit

Siguntang selanjutnya Kambang Iwak, jembatan Ampera dan berakhir di TPKS mempunyai nilai sebesar 250.12979846754 Hasil lainnya yakni dari Punt kayu, Bukit Siguntang, Jembatan Ampera, Kambang Iwak dan berakhir di TPKS mempunyai nilai 265.13621158484 dan yang lainnya lagi untuk titik awal Punt Kayu, Kambang Iwak, Bukit Siguntang, Jembatan Ampera dan berakhir di TPKS mempunyai nilai yang paling besar yakni 275.14956096922. Dari sekian iterasi maka yang paling pendek dapat dilihat pada tabel 1. Sehingga hasil rute terpendek pada pengujian tersebut dapat digambarkan dalam bentuk tabel seperti berikut ini :

Tabel 1. Hasil Penerapan AntCO

No	Lokasi	Ant Colony Optimazition
1	Punti Kayu – Jembatan Ampera	205.12025621393
2	Jembatan Ampera – Kambang Iwak	
3	Kambang Iwak – Bukit Siguntang	
4	Bukit Siguntang - TPKS	

4.3. Pembahasan

Bahasan yang dituangkan dalam bentuk tahapan yang dilakukan dalam mengimplementasikan algoritma AntCO yakni:

Langkah 1

Inisialisasi harga parameter.

- a) (τ_{ij}) adalah intensitas jejak pheromone semut
- b) (n) adalah banyak tempat termasuk koordinat (x,y) atau jarak antar tempat (d_{ij})

Rumus untuk menentukan jarak adalah

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

- c) Q adalah tetapan siklus semut.
- d) α adalah tetapan pengendali intensitas jejak semut, nilai $\alpha > 0$
- e) β adalah tetapan pengendali visibilitas,

nilai $\beta > 0$

f) Visibilitas antar lokasi $(\eta_{ij}) = 1/d_{ij}$

g) m adalah jumlah semut.

h) ρ adalah tetapan penguapan jejak semut, dimana $0 \leq \rho \leq 1$ i) NC_{max} adalah jumlah siklus maksimum.

Langkah 2

Pengisian node pertama ke dalam *tabu list*. *Tabu list* digunakan untuk menyimpan daftar urutan node-node yang sudah di kunjungi setiap semut. Setiap kali semut berkunjung ke suatu lokasi wisata maka elemen *tabu list* akan bertambah satu, seterusnya sampai *tabu list* penuh

Langkah 3

Hitung persamaan probabilitas untuk menentukan lokasi tujuan.

$$P_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{u \in J^k} [\tau_{ij}]^\alpha [\eta_{ij}]^\beta} \quad (1)$$

Langkah 4

Tahap ini dilakukan pembaruan pheromone (t) lokal.

$$\tau(t, v) = (1 - \rho) \cdot \tau(t, v) + \rho \cdot \Delta\tau(t, v) \quad (2)$$

$$\Delta\tau(t, v) = \frac{1}{L_{mn} \cdot C} \quad (3)$$

Dimana

L_{mn} = Panjang tur yang diperoleh

C = Jumlah h lokasi

ρ = Parameter dengan nilai 0 sampai 1

$\Delta\tau$ = Perubahan *pheromone*

Langkah 5

Tahap ini merupakan pembaharuan *Pheromone* global

$$\tau(t, v) \leftarrow (1 - \alpha) \cdot \tau(t, v) + \alpha \cdot \Delta\tau(t, v) \quad (4)$$

$$\Delta\tau(t, v) = \begin{cases} L_{gb}^{-1} & \text{jika } (t, v) \text{ etur_terbaik} \\ 0 & \text{lainnya} \end{cases} \quad (5)$$

Dimana

$\tau(t, v)$ = nilai *pheromone* setelah mengalami pembaruan lokal

L_{gb} = Panjang tur terpende pada akhir siklus

α = parameter dengan nilai antara 0 sampai 1

$\Delta\tau$ = Perubahan *pheromone*

Menurut (Fenwa et al., 2014 dan Chawda & Sureja, 2012) tentang algoritma koloni semut *pseudocode* nya secara global adalah :

Begin

Initialize the pheromone trails and parameter;

Generate of m solutions

(ants);

For each individual ant k to m;

Calculate probabilitas;

For each and determine its best position;

Determine the best global

ant;

Update the pheromone trail;

Check if distance true;

End

5. Kesimpulan

Berdasarkan dari penelitian yang telah dilakukan didapatkan kesimpulan berupa :

1. Algoritma *Ant Colony Optimization* yang selama ini digunakan dapat digunakan untuk mencari rute terpendek dalam pencarian lokasi wisata.
2. Pengujian pada titik lokasi yang sama untuk pencarian jalur terpendek terbukti algoritma ACO mampu bekerja dengan baik.
3. Penerapan Algoritma ini menggunakan PHP.
4. Hasil penerapan AntCO menggunakan PHP mendapatkan nilai 205.12025621393

6. Saran

Penelitian ini menerapkan algoritma AntCO menggunakan bahasa pemrograman PHP, dimana bentuk output yang diinginkan dalam bentuk *Website*. Harapannya ada pengembangan penelitian lebih lanjut dengan menggunakan proses perbandingan dengan banyak algoritma menggunakan bahasa program lainnya, sehingga dapat membedakan manakah algoritma yang lebih baik dalam menentukan rute terpendek.

7. Referensi

- [1] Y. Fernando, M. A. Mustaqov, and D. A. Megawaty, "Penerapan Algoritma a-Star Pada Aplikasi Pencarian Lokasi Fotografi Di Bandar Lampung Berbasis Android," *J. Teknoinfo*, vol. 14, no. 1, p. 27, 2020, doi: 10.33365/jti.v14i1.509.
- [2] R. Astri and Sularno, "Implementation of A-Star Algorithm for Searching Routes Near the Tsunami Evacuation Shelter Point," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 2, pp. 254–259, 2020, doi: 10.29207/resti.v4i2.1602.
- [3] Alhanjouri, M. & Alfarra, B., 2012. Ant Colony Versus Genetic Algorithm Based on Travelling Salesman Problem. *International Journal Comp.Tech.Appl.*, 2 (3), pp.570-78.
- [4] Tyas, Y.S. & Prijodiprojo, W., 2013. Aplikasi Pencarian Rute Jurnal Pendidikan Teknologi Informasi (JUKANTI) Volume (1) No (1) Maret 2018 32 Terbaik dengan Metode Ant Colony Optimazation (ACO). *IJCCS*, 7, pp.55-64.
- [5] S. Purnama, D. A. Megawaty, and Y. Fernando, "Penerapan Algoritma A Star Untuk Penentuan Jarak Terdekat Wisata Kuliner di Kota Bandarlampung," *J. Teknoinfo*, vol. 12, no. 1, p. 28, 2018, doi: 10.33365/jti.v12i1.37.
- [6] M. D. Khairansyah, M. Luqman Ashari, and I. Mufidah, "Penentuan Jalur Evakuasi Terpendek Pada Industri Plastik Menggunakan Ant Colony

- Optimization,” *J. Keselam. Transp. Jalan (Indonesian J. Road Safety)*, vol. 8, no. 1, pp. 53–61, 2021, doi: 10.46447/ktj.v8i1.312.
- [7] Murakhiroh, I., Saptono, F., Hasanah, N. & Wiryadinata, R., (2007). Pemanfaatan Metode Heuristic dalam pencarian Jalur terpendek dengan algoritma semut dan algoritma genetika. Seminar nasional aplikasi teknologi informasi.
- [8] Dorigo, M., 1996. *The Ant Colony Optimization Metaheuristics: Algorithms, Applications and Advances*. Universitas Libre de Bruxelles, IRIDIA.
- [9] Bullnheimer, B., Hartl, R. F., dan Strauss, C. (1999). *An improved ant system algorithm for the vehicle routing problem*. Technical report, Institute of Management Science, University of Vienna, Austria.
- [10] Karl F.Doerner, Daniel Merkle, and Thomas Stzle, “Special Issue on Ant Colony Optimization”, *Swarm Intell* (2009) 3: 1-2, DOI 10.1007/s11721-008-0025-1
- [11] Mindaputra, E. (2009). Algoritma Ant Colony Sistem (ACS) Untuk Menyelesaikan Travelling Salesman Problem (TSP). Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Diponegoro, Semarang.