



**OPTIMALISASI METODE PENYELESAIAN  
*ORIENTEERING PROBLEM* MENGGUNAKAN  
ALGORITMA GENETIK  
DAN *ANT COLONY OPTIMIZATION***

**Skripsi**

**diajukan sebagai salah satu persyaratan untuk memperoleh gelar  
Sarjana Pendidikan Program Studi Pendidikan Teknik Informatika dan  
Komputer**

**Oleh**

**Radiva Hera Oktiagi**

**NIM. 5302414066**

**PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI SEMARANG  
2019**

## PENGESAHAN

Skripsi dengan judul “Optimalisasi Metode Penyelesaian *Orienteering Problem* Menggunakan Algoritma Genetik dan *Ant Colony Optimization*” telah dipertahankan di depan sidang Panitia Ujian Skripsi Fakultas Teknik UNNES pada tanggal 23 bulan Mei tahun 2019.

Oleh

Nama : Radiva Hera Oktiagi  
NIM : 5302414066  
Program Studi : Pendidikan Teknik Informatika dan Komputer

Panitia :

Ketua Panitia

Sekretaris

Dr.-Ing. Dhidik Prastiyanto, S.T., M.T.  
NIP.197805312005011002

Ir. Ulfah Mediaty Arief, M.T., IPM.  
NIP. 196605051998022001

Penguji I

Penguji II

Penguji III/Pembimbing

Drs. Slamet Seno Adi, M.Pd., M.T.  
NIP. 195812181985031004

Arief Arfriandi, S.T., M.Eng.  
NIP.198208242014041001

Ir. Ulfah Mediaty Arief, M.T., IPM.  
NIP. 196605051998022001

Mengetahui,

Dekan Fakultas Teknik UNNES



Dr. Nur Qudus, M.T., IPM.  
NIP.196911301994031001

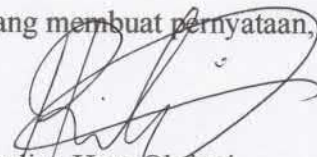
PERNYATAAN KEASLIAN TULISAN

Dengan ini saya menyatakan bahwa:

1. Skripsi ini, adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (sarjana, magister, dan/atau doktor), baik di Universitas Negeri Semarang (UNNES) maupun di perguruan tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan, dan penelitian saya sendiri tanpa bantuan pihak lain, kecuali arahan Pembimbing dan masukan Tim Penguji.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya ini, serta sanksi lainnya sesuai dengan norma yang berlaku di perguruan tinggi ini.

Semarang, 14 Mei 2019

yang membuat pernyataan,



Radiva Hera Oktiagi

NIM. 5302414066

## **MOTO DAN PERSEMBAHAN**

Moto :

- Keberhasilan adalah kemampuan untuk melewati dan mengatasi dari satu kegagalan ke kegagalan berikutnya tanpa kehilangan semangat (Winston Churchill).
- Sebelum menolong orang lain, saya harus dapat menolong diri sendiri. Sebelum menguatkan orang lain, saya harus bisa menguatkan kehidupan diri sendiri dahulu (Petrus Claver).
- Do'a adalah modal yang dimiliki semua orang untuk menjadi apapun dan mendapatkan apapun, tanpa memandang jabatan, status, kekayaan bahkan bentuk fisik (Yusuf Mansur).

Persembahan :

- Ibu, Bapak dan keluarga yang selalu mendukung saya sepenuhnya selama ini.
- Ibu Ulfah sebagai pembimbing yang selalu memberikan semangat, arahan dan dukungan yang maksimal.
- Teman-teman seperjuangan dari PTIK 2014.
- Semua pihak yang senantiasa memberikan doa, dukungan dan semangat.

## INTISARI

Radiva Hera Oktiagi. 2019. “**Optimalisasi Metode Penyelesaian *Orienteering Problem* Menggunakan Algoritma Genetik dan *Ant Colony Optimization*”**”. Skripsi. Jurusan Teknik Elektro. Fakultas Teknik. Universitas Negeri Semarang. Pembimbing : Ir. Ulfah Mediaty Arief, M.T., IPM.****

Dalam menyelesaikan kasus *orienteering problem* dapat digunakan algoritma genetik ataupun *ant colony optimization*. Namun algoritma genetik memiliki kelemahan, dimana hasil yang didapat kurang optimal. Sedangkan algoritma *ant colony optimization* mampu memberikan hasil yang lebih optimal, namun membutuhkan waktu perhitungan yang lebih lama. Dengan menggabungkan kedua algoritma tersebut, diharapkan mampu memberikan hasil yang lebih optimal dengan waktu yang singkat. Penelitian ini bertujuan untuk melihat sejauh mana tingkat optimalisasi diterapkannya penggabungan algoritma genetik dengan *ant colony optimization* dalam menyelesaikan *orienteering problem*.

Metode penelitian yang digunakan adalah eksperimen dimana algoritma gabungan akan di uji coba dan dibandingkan dengan algoritma genetik dan *ant colony optimization*. Hasil yang dibandingkan merupakan waktu komputasi algoritma dan hasil terbaik yang dihasilkan algoritma tersebut.

Berdasarkan hasil uji coba menggunakan 5 skenario yang memiliki batasan jarak 2000, 4000, 6000, 8000, dan 10000, didapatkan hasil bahwa algoritma gabungan mampu meningkatkan waktu komputasi sebesar 3-5 detik dibandingkan *ant colony optimization*, namun masih jauh lebih lambat dibanding dengan algoritma genetik. Sedangkan dalam segi hasil pencarian rute, algoritma gabungan meningkatkan hasil sebesar 20% dari *ant colony optimization* dan 30% - 80% dari algoritma genetik. Saran pada penelitian selanjutnya adalah melakukan pengujian dengan skenario yang lebih banyak, dan melakukan pengujian terhadap parameter tiap algoritma secara lebih mendalam agar hasil yang didapat bisa lebih optimal.

**Kata Kunci :** *Algoritma Gabungan, Algoritma Genetik, Ant Colony, Orienteering Problem*

## PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat dan ridho-Nya sehingga penyusunan skripsi yang berjudul “**Optimalisasi Penyelesaian Orienteering Problem Menggunakan Algoritma Genetik dan Ant Colony Optimization**” dapat diselesaikan dengan baik. Skripsi ini disusun sebagai salah satu persyaratan meraih gelar Sarjana Pendidikan. Dalam penyusunan skripsi ini tidak bisa lepas dari dukungan berbagai pihak. Oleh sebab itu pada kesempatan ini ingin diberikan rasa hormat dan ucapan terima kasih kepada :

1. Ir. Ulfah Mediaty Arief, M.T., IPM. selaku dosen Pembimbing yang penuh kesabaran dalam membimbing, memberi arahan dan motivasi kepada penulis sehingga skripsi ini dapat selesai.
2. Dr. Nur Qudus, M.T., IPM., selaku Dekan Fakultas Teknik Universitas Negeri Semarang.
3. Dr. Ing. Dhidik Prastiyanto, S.T., M.T., selaku Ketua Jurusan Elektro UNNES yang telah membantu kelancaran ujian skripsi.
4. Keluargaku tercinta yang selalu memberi motivasi baik moral maupun material serta do’a restu dalam menyelesaikan skripsi ini.
5. Sahabat-sahabatku, teman PTIK 2014 dan semua pihak yang membantu dalam penyusunan skripsi ini.

Akhirnya penulis berharap, semoga penelitian ini bermanfaat bagi pembaca pada khususnya dan perkembangan pendidikan Indonesia pada umumnya.

Semarang, 14 Mei 2019

Penulis

## DAFTAR ISI

HALAMAN JUDUL.....	i
PENGESAHAN.....	ii
PERNYATAAN KEASLIAN TULISAN.....	iii
MOTO DAN PERSEMBAHAN.....	iv
INTISARI.....	v
PRAKATA.....	vi
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	4
1.3 Batasan Masalah.....	5
1.4 Rumusan Masalah.....	5
1.5 Tujuan.....	6
1.6 Manfaat Penelitian.....	6
<b>BAB II KAJIAN PUSTAKA DAN LANDASAN TEORI.....</b>	<b>7</b>
2.1 Kajian Pustaka.....	7
2.2 Landasan Teori.....	10
2.2.1 Orienteering Problem.....	10
2.2.3 Combinatorial Optimization.....	11
2.2.4 Algoritma Genetika.....	13
2.2.5 Ant Colony Optimization.....	20
<b>BAB III METODE PENELITIAN.....</b>	<b>25</b>

3.1 Waktu dan Tempat Pelaksanaan.....	25
3.2 Alat dan Bahan Penelitian.....	25
3.3 Desain Penelitian.....	25
3.3.1 Metode Penelitian.....	25
3.3.2 Langkah Penelitian.....	26
3.3.3 Penggabungan Algoritma.....	28
3.4 Parameter Penelitian.....	37
3.5 Teknik Pengumpulan Data.....	37
3.6 Teknik Analisis Data.....	38
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>	<b>39</b>
4.1 Deskripsi Data.....	39
4.2 Analisis Data.....	40
4.2.1 Algoritma Genetik.....	40
4.2.2 Ant Colony Optimization.....	41
4.2.3 Algoritma Gabungan.....	43
4.3 Pembahasan.....	45
4.3.1. Perbandingan Hasil.....	45
4.3.2. Kinerja Algoritma.....	47
<b>BAB V SIMPULAN DAN SARAN.....</b>	<b>52</b>
5.1 Simpulan.....	52
5.2 Saran.....	52
DAFTAR PUSTAKA.....	54
LAMPIRAN.....	58



## Daftar Gambar

Gambar 2.1: Langkah Algoritma Genetik.....	16
Gambar 2.2: Ilustrasi <i>one point crossover</i> .....	18
Gambar 2.3: Ilustrasi <i>multi point crossover</i> .....	19
Gambar 2.4: Ilustrasi <i>uniform crossover</i> .....	19
Gambar 2.5: Langkah kerja <i>Ant Colony Optimization</i> .....	22
Gambar 3.1: Diagram alir langkah penelitian.....	27
Gambar 3.2: Grafik perbandingan ACO dengan GA.....	29
Gambar 3.3: Flowchart algoritma GACO.....	29
Gambar 3.4: (a) Rute pilihan pertama. (b) Rute pilihan kedua.....	31
Gambar 3.5: Hasil perhitungan fungsi $drop(v_i)$ .....	32
Gambar 3.6: Rute sebelum penambahan jalur.....	33
Gambar 3.7: Perhitungan penempatan jalur.....	33
Gambar 4.1: Sebaran 150 titik.....	39
Gambar 4.2: Hasil jumlah titik algoritma genetik.....	40
Gambar 4.3: Hasil waktu komputasi algoritma genetik.....	41
Gambar 4.4: Hasil jumlah titik ACO.....	42
Gambar 4.5: Hasil waktu komputasi ACO.....	43
Gambar 4.6: Hasil jumlah titik algoritma gabungan.....	44
Gambar 4.7: Hasil waktu komputasi algoritma gabungan.....	45
Gambar 4.8: Perbandingan hasil jumlah titik.....	46
Gambar 4.9: Perbandingan hasil waktu komputasi.....	47
Gambar 4.10: Kinerja Algoritma Genetik.....	48
Gambar 4.11: Kinerja Algoritma ACO.....	48
Gambar 4.12: Kinerja Algoritma Gabungan.....	49
Gambar 4.13: Perbandingan Kinerja Algoritma.....	50

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Pariwisata merupakan salah satu sektor penyumbang devisa negara ke-2 paling besar di Indonesia (Sukmana, 2017). Menurut data dari Unwto(2017) diperkirakan jumlah kunjungan wisata internasional mencapai 1,8 milyar pada tahun 2030 dengan pertumbuhan 4,4% tiap tahunnya. Pertumbuhan rata rata di wilayah Asia Tenggara pada tahun 2016 mencapai 9% dimana indonesia memiliki pertumbuhan sebesar 15%.

Banyaknya destinasi wisata merupakan sebuah keuntungan karena memberikan banyak pilihan untuk wisatawan, namun juga merupakan suatu kelemahan karena waktu wisatawan yang terbatas dalam berkunjung dan juga penentuan preferensi tiap wisatawan yang berbeda (Chaqiqi et al., 2018). Saat ini banyak wisatawan yang merencanakan kunjungan wisata berdasarkan situs web, artikel majalah, atau buku panduan di toko buku maupun perpustakaan (Vansteenwegen & Oudheusden, 2007). Tidak setiap tempat wisata disenangi oleh pengunjung yang datang dikarenakan kesukaan setiap pengunjung berbeda. Wisatawan juga tidak mengetahui secara pasti jarak dan rute antar objek wisata sehingga menyulitkan perencanaan yang efektif untuk berwisata secara mandiri (Priandani & Mahmudi, 2016). Perencanaan rute wisata dengan hasil yang optimal disebut *Tourist Trip Design Problem* (TTDP).

*Tourist Trip Design Problem* (TTDP) adalah suatu masalah perencanaan rute wisata dimana wisatawan ingin mengunjungi beberapa tempat yang sesuai dengan kesukaanya dimana dalam kunjungan tersebut memiliki batasan dimana salah satunya adalah waktu. TTDP merupakan sebuah pengaplikasian dari masalah optimasi yang disebut dengan *Orienteering Problem* (OP) (Ostrowski et al., 2016; Kobeaga et al., 2017; Gavalas et al., 2014).

*Orienteering Problem* (OP) juga dapat disebut *Selective Traveling Salesman Problem* (STSP) merupakan masalah pemilihan rute terpendek dimana setiap titik memiliki nilai tersendiri dan tidak seluruh titik wajib dikunjungi. Tujuan dari OP adalah kita dituntut mengunjungi sebanyak mungkin titik sehingga memiliki total nilai terbanyak namun tidak melebihi jarak yang telah ditentukan sebelumnya (Feillet et al., 2001; Bochar & Radi, 2016; Laporte & Martello, 1990; Piwonska & Seredynski, 2011; Ostrowski et al., 2016).

Berdasarkan tingkat kerumitan permasalahannya, permasalahan yang termasuk *NP-Hard Problem* tidak dapat diselesaikan menggunakan metode perhitungan manual. Dengan berkembangnya ilmu pengetahuan dan teknologi, solusi dari *NP-Hard Problem* dapat dihasilkan dengan mengkombinasikan teknologi komputasi dengan algoritma yang cerdas sehingga dapat mengurangi resiko kesalahan dari metode perhitungan manual oleh manusia dan meningkatkan efisiensi penggunaan sumber daya.

Banyak algoritma yang telah digunakan untuk menyelesaikan kasus ini beberapa diantaranya menggunakan algoritma *ant colony optimization* (Luo et al., 2016; Baranowski et al., 2014; Chaparro & Valdez, 2013).

*Ant Colony Optimization* (ACO) merupakan algoritma heuristik yang menggunakan perilaku semut dalam mencari makanan sebagai dasarnya. Terdapat kelemahan pada algoritma ini dimana pada tahap awal memiliki performa yang rendah akibat penentuan feromon awal yang buruk karena ketiadaan informasi mengenai jalur awal (Kong et al., 2007; Zhao et al., 2016). Untuk mengatasi masalah tersebut maka digunakanlah pendekatan metaheuristik dengan *Genetic Algorithm*/Algoritma Genetik (GA) dimana memiliki tingkat kinerja awal yang tinggi (Zhu et al., 2014). Algoritma genetik telah banyak digunakan untuk menyelesaikan kasus optimasi dalam beberapa kasus OP (Bochar & Radi, 2016; Piwonska & Seredynski, 2011; Golden et al., 2008; Koszelew & Ostrowski, 2013; Karbowska-chili, 2012). Namun, GA memiliki kelemahan dimana tidak memiliki mekanisme umpan balik yang praktis sehingga menyebabkan kinerjanya semakin rendah pada iterasi akhir (Zhao et al., 2016).

Algoritma gabungan yang dibentuk dengan menggabungkan GA dengan ACO ditujukan untuk mengurangi kelemahan dalam hal kinerja pencarian rute sehingga dapat menghasilkan performa pencarian yang lebih maksimal. Penggabungan 2 algoritma ini telah digunakan untuk mengoptimalkan hasil yang didapat (Zhao et al., 2016; Zhu et al., 2014; Lee et al., 2008; Zukhri et al., 2013). Salah satunya dengan menggabungkan tingkat konvergensi antara GA dan ACO.

Namun algoritma ini belum dicobakan menggunakan kasus yang lebih kompleks dan dengan data yang lebih besar (Zhao et al., 2016).

Berdasar uraian diatas algoritma genetik dan *ant colony optimization* dapat digunakan untuk menyelesaikan kasus *orienteering problem* dan penggabungan kedua algoritma ini diharapkan mampu menghasilkan hasil optimal dengan waktu lebih cepat. Untuk itu penelitian ini membahas Optimalisasi Metode Penyelesaian *Orienteering Problem* Menggunakan Algoritma Genetik dan *Ant Colony Optimization*.

## 1.2. Identifikasi Masalah

Berdasarkan latar belakang diatas, ditemukan beberapa permasalahan yaitu:

1. Wisatawan masih perlu mengumpulkan informasi wisata secara mandiri
2. Keterbatasan waktu dan kesenangan wisatawan yang berbeda satu sama lain
3. *Tourist Trip Design Problem* (TTDP) memerlukan banyak parameter agar dapat berguna mencari rute wisatawan yang sesuai
4. *Orienteering Problem* (OP) merupakan sebuah permasalahan pencarian rute dimana membutuhkan waktu komputasi yang lama untuk mendapatkan hasil terbaik.
5. Algoritma Genetik dengan waktu pencarian yang cepat memiliki kelemahan dimana hasil yang diberikan kurang optimal.

6. *Ant Colony Optimization* mampu memberikan hasil pencarian yang lebih baik, namun membutuhkan waktu yang lebih lama.

### 1.3. Batasan Masalah

Agar penelitian yang dilakukan lebih terfokus, maka penulis memberikan batasan masalah sebagai berikut :

- a. Fokus penelitian ini adalah hasil penggabungan algoritma genetika dengan *ant colony optimization* secara berurutan
- b. Hasil penggabungan algoritma secara berurutan tersebut akan dibandingkan dengan algoritma genetika dan *ant colony optimization* untuk melihat perbedaan hasilnya
- c. Data yang digunakan meliputi:
  - Sejumlah titik tujuan
  - Lokasi tiap-tiap titik tujuan
  - Lokasi awal
  - Jarak maksimal rute yang diperbolehkan

### 1.4. Rumusan Masalah

Berdasarkan latar belakang yang telah diutarakan sebelumnya, dapat dirumuskan masalah, sejauh mana tingkat optimalisasi diterapkannya penggabungan algoritma genetik dengan *ant colony optimization* secara berurutan dalam menyelesaikan *orienteering problem*.

### **1.5. Tujuan**

Tujuan dari penelitian ini adalah untuk mengetahui sejauh mana tingkat optimalisasi diterapkannya penggabungan algoritma genetik dengan *ant colony optimization* secara berurutan dalam menyelesaikan *orienteering problem*.

### **1.6. Manfaat Penelitian**

Penelitian ini dapat memberikan manfaat sebagai berikut :

- a. Mengembangkan metode penggabungan algoritma genetik dengan *ant colony optimization* dalam menyelesaikan *orienteering problem*.
- b. Menambah pengetahuan mengenai tingkat optimalisasi metode gabungan algoritma genetik dan *ant colony optimization* dalam menyelesaikan beberapa skenario *orienteering problem*.

## BAB II

### KAJIAN PUSTAKA DAN LANDASAN TEORI

#### 2.1. Kajian Pustaka

Penggabungan algoritma genetik dengan *ant colony optimization* sebelumnya dilakukan oleh FuTao Zhao, Zhong Yao, Jing Luan, Xing Song (2019) dalam penelitiannya yang berjudul *A novel method to solve supplier selection problem: Hybrid algorithm of genetic algorithm and ant colony optimization*. Dalam penelitian ini, kedua algoritma tersebut dikombinasikan dengan cara memanfaatkan tingkat konvergensi yang berbeda. Algoritma genetika/*Genetic Algorithm* (GA) dijalankan pada iterasi awal yang kemudian hasil sementara dari algoritma genetika dijadikan sebagai data awal oleh *ant colony optimization* (ACO) yang selanjutnya dilakukan perhitungan hingga mencapai hasil akhir paling optimal. Uji coba dilakukan dengan menyelesaikan sebuah kasus *Supplier Selection Problem* dengan data 20 jenis barang dan 130 pemasok. Analisis dari hasil yang diperoleh dari kegiatan eksperimen yang dilakukan, menunjukkan bahwa algoritma *hybrid* algoritma genetik dan *ant colony optimization* memiliki waktu pengerjaan dan hasil yang lebih baik dibandingkan dengan algoritma genetik maupun *ant colony optimization*.

Orhan Engin dan Abdullah Güçlü (2018) menggabungkan GA dengan ACO dalam kasus *No-Wait Flow Shop Scheduling Problem* dalam penelitian berjudul *A New Hybrid Ant Colony Optimization Algorithm For Solving The No*



*-Wait Flow Shop Scheduling Problems*. Engin dan Güçlü menggunakan proses *crossover* dan mutasi yang terdapat pada GA untuk dimasukkan ke dalam proses dari algoritma ACO sehingga menghasilkan algoritma *Hybrid ACO (HACO)*. Algoritma tersebut kemudian diuji menggunakan 192 jenis data dan dibandingkan dengan algoritma *Adaptive Learning Approach (ALA)* dan *Genetic Heuristic (GEN-2)*. Hasil penelitian menunjukkan bahwa HACO mampu memberikan hasil lebih baik dibandingkan dengan ALA dan GEN-2, dimana HACO mampu memberikan hasil 0.052% mendekati hasil optimal sedangkan ALA dan GH masing-masing memperoleh hasil 0.2558% dan 0.4881% dan HACO mampu mendapatkan nilai optimal dari 180 kasus sedangkan ALA dan GH mendapatkan nilai optimal dari 142 dan 70 kasus.

Penelitian oleh Zainudin Zukhri dan Irving Vitra Papatungan (2013) berjudul *A Hybrid Optimization Algorithm Based on Genetic Algorithm and Ant Colony Optimization* menggabungkan GA dengan ACO dalam menyelesaikan masalah TSP. Parameter dan langkah langkah dalam kedua algoritma yang sama digabungkan untuk membuat algoritma baru yang disebut GACO. Pengujian dilakukan menggunakan data acak dan juga data standar dari TSPLib. Hasil perhitungan menunjukkan bahwa penggabungan kedua algoritma ini tidak terlalu berpengaruh terhadap data dengan jumlah titik kecil namun memiliki hasil yang jauh lebih baik terhadap data dengan jumlah titik besar.

Jiaxu Ning, Qin Zhang, Changsheng Zhang, Bin Zhang (2018) melakukan pengembangan algoritma ACO dalam penelitiannya yang berjudul *A best-path-*

*updating information-guided ant colony optimization algorithm*. Penelitian ini berfokus untuk meningkatkan waktu konvergensi tanpa memengaruhi kualitas dari hasil yang didapat. ACO dikembangkan dengan memberikan peningkatan kemampuan pembaruan feromon pada tiap jalur (*edge*) dan juga penambahan mekanisme perataan feromon saat terjadi stagnansi dalam proses pencarian rute. Pengujian dilakukan dengan membandingkan algoritma ACOH dengan 3 algoritma lain dalam kasus *Constraint Satisfaction Problem* (CSP) dan 5 algoritma lain dalam kasus *Traveling Salesman Problem* (TSP). Hasil pengujian menunjukkan ACOH memberikan hasil memuaskan saat menyelesaikan kasus dengan tingkat kesulitan tinggi, namun memberikan hasil yang kurang memuaskan pada kasus dengan tingkat kesulitan rendah.

Gorka Kobeaga, Maria Merino, Jose A. Lozano (2017) dalam penelitian yang berjudul *An Efficient Evolutionary Algorithm for the Orienteering Problem* menggunakan Algoritma Evolusi/*Evolutionary Algorithm* (EA) yang berdasar kepada Algoritma Genetika untuk menyelesaikan *Orienteering Problem*. Menambahkan operasi penambahan dan pengurangan vertex untuk meningkatkan nilai *profit* dan menjaga agar jarak tempuh tidak melebihi jarak maksimal ( $t_{max}$ ). Perhitungan dilakukan menggunakan 344 data dari TSPLib yang dibagi menjadi 4 generasi dan tiap generasi dibagi menjadi 2 ukuran yaitu sedang (hingga 400 titik) dan luas (hingga 7397 titik). Hasil perhitungan dibandingkan dengan 3 algoritma lain yaitu *Branch & Cut* (B&C), GRASP-PR, dan 2-P IA dengan hasil menunjukkan dalam ukuran medium EA mendapat peringkat ke-2 baik dari segi

nilai *profit* dan juga waktu komputasi, dan dalam ukuran luas EA mendapatkan posisi terbaik baik dari segi nilai *profit* dan juga waktu komputasi.

## 2.2. Landasan Teori

### 2.2.1. Orienteering Problem

*Orienteering Problems* (OP) diperkenalkan oleh Tsiligirides (1984) dinamai sesuai dengan olahraga bernama *orienteering*. Nama lain yang biasa digunakan untuk OP adalah *Selective Traveling Salesman Problem* (STSP) (Laporte & Martello, 1990). OP dapat diformulasikan sebagai berikut: diketahui  $G = (V, E)$  adalah graf sudut berbobot dengan *profits* (poin atau nilai) di simpulnya. Kemudian memiliki simpul awal  $\alpha$ , sebuah simpul tujuan  $t$ , dan batas waktu (*budget*) bernilai positif  $B$ , OP bertujuan untuk mencari jalur dari  $\alpha$  menuju ke  $t$  (atau tur  $\alpha = t$ ) dengan panjang perjalanan tidak melebihi  $B$  dengan nilai *profit* dari seluruh kunjungan termaksimalkan.

OP dapat diformulasikan sebagai *integer programming problem* (Vansteenwegen et al., 2011) : Diketahui  $N$  adalah nomor dari simpul bernilai  $1, 2, \dots, N$  dimana  $s = 1$  dan  $t = N$ ,  $p_i$  merupakan nilai *profit* tiap mengunjungi simpul  $i$  dan  $c_{ij}$  adalah nilai jarak perjalanan dari  $i$  ke  $j$ . Untuk setiap jalur dari  $1$  ke  $N$ , jika simpul  $i$  diikuti oleh simpul  $j$  maka variabel  $x_{ij}$  sama dengan  $1$  atau sama dengan  $0$  untuk sebaliknya. Terakhir,  $u_i$  melambangkan posisi dari simpul  $i$  dalam jalur. Dari notasi tersebut dapat diformulasikan relasi sebagai berikut:

$$\max \sum_{i=1}^{N-1} \sum_{j=2}^{N-1} s_i x_{ij} \quad (2.1)$$

$$\sum_{j=2}^N x_{ij} = \sum_{i=1}^{N-1} x_{iN} = 1 \quad (2.2)$$

$$\sum_{i=1}^{N-1} x_{ir} = \sum_{j=1}^N x_{ij} \leq 1, r \in \{2, \dots, N-1\} \quad (2.3)$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^N c_{ij} n_{ij} \leq B \quad (2.4)$$

$$2 \leq u_i \leq N, i \in \{1, 2, \dots, N\} \quad (2.5)$$

$$u_i - u_j + 1 \leq (N-1)(1 - x_{ij}), i, j \in \{2, \dots, N\} \quad (2.6)$$

$$x_{ij} \in \{0, 1\}, i, j = \{1, \dots, N\} \quad (2.7)$$

Fungsi objektif (2.1) adalah untuk memaksimalkan total nilai *profit* dari simpul yang dikunjungi. Batasan (2.2) memastikan agar jalur dimulai dari simpul 1 dan berakhir di simpul N. Batasan (2.3) memastikan agar jalur yang dimulai dari simpul 1 dan berakhir di simpul N terhubung dan tiap simpul dikunjungi paling banyak 1 kali. Batasan (2.4) memastikan bahwa jalur yang dibuat tidak melebihi batasan waktu yang ditentukan. Terakhir, batasan (2.5) dan (2.6) memastikan tidak ada sub-tur. (Gavalas et al., 2014). OP merupakan permasalahan yang termasuk kasus NP-hard (Golden et al., 1987; Laporte & Martello, 1990).

### 2.2.2. Combinatorial Optimization

*Combinatorial Optimization* (CO) adalah salah satu cabang permasalahan yang dibahas dalam lingkungan *computer science and operation research*. CO

sendiri dapat meliputi beberapa aplikasi dalam permasalahan seperti *crew scheduling, jobs to machines allocation, vehicle routing and scheduling, circuit design and wiring*, dan banyak lainnya (Hosny, 2010). Namun dalam penelitian ini berfokus pada permasalahan OP yang juga merupakan cabang dari implementasi CO pada topik *traveling salesman problem*.

CO secara umum dapat diartikan sebagai sebuah permasalahan yang membutuhkan pencarian terhadap solusi terbaik di antara banyak kandidat solusi yang mungkin ditemukan dengan batasan-batasan (*constraint*) tertentu. Dalam tesisnya, Hosny (2010) menuliskan CO adalah usaha pencarian skema alokasi sejumlah *resource* yang terbatas untuk mencapai suatu nilai fungsi tertentu (*objective*).

Metode penyelesaian permasalahan CO berkaitan dengan kompleksitas permasalahan masing-masing, beberapa di antara permasalahan CO seperti *traveling salesman problem, vehicle routing problem, dan orienteering problem* seperti yang dibahas pada penelitian ini termasuk dalam kategori *NP-Hard Problem* (Gajpal & Abad, 2009; Dethloff, 2001) yang mana permasalahan dengan kategori ini memiliki tingkat kompleksitas yang tinggi dengan tingkat perkembangan *solution space* yang meningkat secara eksponensial dan disebut *intracable* (Kovacs, 2008). Dengan tingkat kompleksitas *NP-Hard Problem*, hanya sedikit peneliti yang menggunakan metode eksak dan lebih banyak metode *heuristic* dan *meta-heuristic* yang memiliki kemampuan menyelesaikan permasalahan dengan skala yang besar yang lebih menjanjikan (Thangiah, 1993).

### **2.2.3. Algoritma Genetika**

#### **2.2.3.1. Pengertian Algoritma Genetika**

Algoritma Genetik/*Genetic Algorithm* (GA) merupakan sebuah algoritma yang berdasar pada proses seleksi alam dimana individu terkuat yang akan bertahan di akhir. GA adalah metodologi pencarian optimasi adaptif umum berdasarkan pada analogi langsung ke seleksi alam Darwinian dan genetika dalam sistem biologis, merupakan alternatif yang menjanjikan untuk metode heuristik konvensional. GA bekerja dengan satu set solusi kandidat yang disebut populasi. GA mendapatkan solusi optimal setelah serangkaian perhitungan berulang. GA menghasilkan populasi solusi pengganti berurutan yang diwakili oleh kromosom, yaitu solusi untuk masalah, hingga hasil yang dapat diterima diperoleh. Terkait dengan karakteristik pencarian eksploitasi dan eksplorasi, GA dapat berurusan dengan ruang pencarian besar secara efisien, dan karenanya memiliki sedikit kesempatan untuk mendapatkan solusi optimal lokal daripada algoritma lainnya.

Algoritma Genetika terinspirasi dari prinsip-prinsip genetika dan seleksi alam (teori evolusi Darwin) yang pertama kali ditemukan di Universitas Michigan, Amerika Serikat, oleh John Holland pada tahun 1975 melalui sebuah penelitian dan kemudian dipopulerkan oleh salah satu muridnya, David Golberg. Konsep dasar dari GA sebenarnya dirancang untuk mensimulasikan proses-proses dalam sistem alam yang diperlukan untuk evolusi, khususnya teori evolusi alam yang dicetuskan oleh Charles Darwin, yaitu *Survival of the Fittest*. Menurut teori ini, dalam suatu populasi, makhluk yang kuat akan mendominasi makhluk yang

lebih lemah dalam persaingan mereka untuk memperebutkan sumber daya alam yang langka (Sutojo et al., 2011:403).

Algoritma Genetika adalah teknik pencarian heuristik yang didasarkan pada gagasan evolusi seleksi alam dan genetika. Algoritma ini memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi. Dalam proses evolusi, individu secara terus menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya. Hanya individu yang kuat yang mampu bertahan. Proses seleksi alamiah ini melibatkan perubahan gen yang terjadi pada individu melalui proses perkembangbiakan. Proses perkembangbiakan ini didasarkan pada analogi struktur genetika dan perilaku kromosom dalam populasi individu dengan menggunakan dasar sebagai berikut:

1. Individu dalam populasi bersaing untuk sumber daya alam dan pasangannya.
2. Mereka yang paling sukses di setiap kompetisi akan menghasilkan keturunan yang lebih baik daripada individu-individu yang berkinerja buruk.
3. Gen dari individu yang baik akan menyebar ke seluruh populasi sehingga dua orang tua (*parent*) yang baik kadang-kadang akan menghasilkan keturunan yang lebih baik dari orangtuanya.
4. Setiap ada pergantian generasi terbaru ini biasanya lebih cocok dengan lingkungan mereka. Dengan kata lain, generasi baru ini bisa menyesuaikan dengan keadaan lingkungannya.

### 2.2.3.2. Istilah

Berikut beberapa definisi atau istilah penting yang sering dijumpai pada algoritma genetika:

1. *Genotype* (Gen), sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu. Dalam Algoritma Genetika, gen ini bisa berupa biner, *float*, *integer*, maupun karakter, atau kombinatorial.
2. *Allele*, nilai atau isi dari gen.
3. Kromosom, gabungan dari gen gen yang membentuk nilai tertentu.
4. Individu, suatu solusi (nilai atau keadaan) yang mungkin dari suatu permasalahan. Merupakan gabungan dari satu atau lebih kromosom.
5. Populasi, sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.
6. Generasi, menyatakan satu siklus proses evolusi atau satu iterasi di dalam Algoritma Genetika.

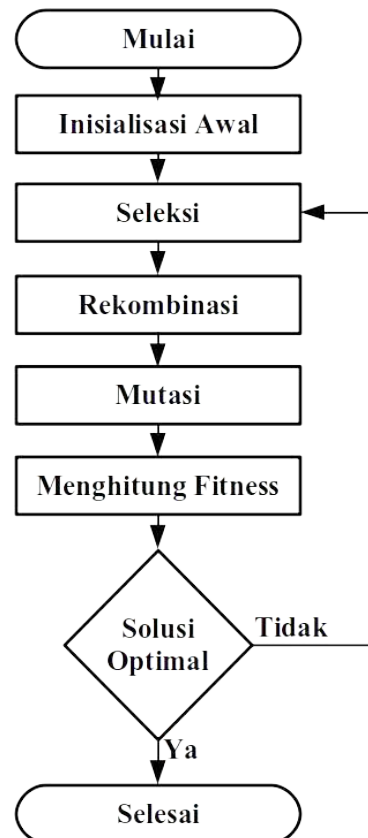
### 2.2.3.3. Langkah Kerja

Diagram alir proses dalam algoritma genetik digambarkan pada Gambar

2.1. Proses algoritma genetik terdiri dari beberapa tahap, dua diantaranya adalah seleksi dan reproduksi yang terdiri dari rekombinasi (*crossover*) dan mutasi.

Secara singkat, proses dalam algoritma genetik adalah sebagai berikut :





Gambar 2.1: Langkah Algoritma Genetik

### 1. Inialisasi Awal

Inialisasi awal dilakukan untuk memasukan nilai parameter-parameter yang digunakan dalam algoritma genetik seperti nilai probabilitas seleksi dan mutasi dan juga digunakan untuk melakukan pembentukan populasi awal.

Dalam pembentukan populasi awal perlu memperhatikan 2 hal yaitu:

- Tingkat keragaman individu dalam populasi yang terlalu kecil dapat mengakibatkan hasil yang didapat tidak optimal
- Ukuran atau jumlah individu dalam populasi tidak boleh terlalu besar karena mampu memperlambat kinerja dari GA. Namun, tidak boleh terlalu kecil karena dapat memunculkan hasil yang kurang optimal.

Pembentukan setiap individu untuk populasi tahap awal menggunakan metode acak (*random*) sehingga mampu memberikan tingkat keragaman yang tinggi dalam populasi yang dibuat dan juga dilakukan penghitungan nilai *fitness* untuk digunakan dalam langkah selanjutnya, sedangkan dalam penentuan ukuran populasi digunakan metode *trial and error*.

## 2. Seleksi

Seleksi digunakan untuk memilih individu mana saja yang akan dipilih untuk proses rekombinasi (*crossover*) dan mutasi. Beberapa metode digunakan dalam proses seleksi diantaranya:

- *Roulette Wheel Selection*

Pada metode ini, pemilihan individu didasarkan pada persentase nilai *fitness* individual terhadap keseluruhan nilai *fitness* individual dalam populasi. Semakin tinggi nilai persentasenya, maka kemungkinan individu tersebut terpilih juga semakin besar.

- *Tournament Selection*

Dalam metode ini, sejumlah  $k$  individu dipilih secara acak yang kemudian individu dengan nilai *fitness* tertinggi akan terpilih.

- *Rank Selection*

Pemilihan individu pada metode ini mirip dengan *Roulette Wheel Selection* dengan perbedaan penentuan nilai persentasenya tidak berasal dari nilai *fitness* secara langsung, namun setiap individu memiliki peringkat berdasarkan nilai *fitness*-nya. Semakin tinggi nilai *fitness*, maka

semakin tinggi pula peringkatnya sehingga kemungkinan individu tersebut terpilih juga semakin besar.

- *Random Selection*

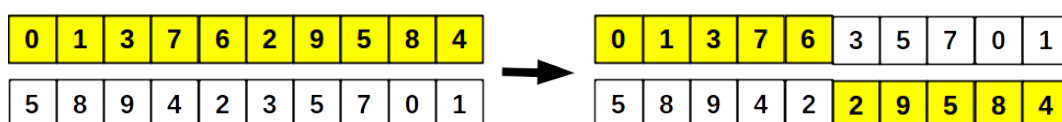
Pemilihan individu pada metode ini dilakukan secara acak tanpa memperhatikan variabel maupun nilai yang terkandung dalam setiap individu yang tersedia.

### 3. Rekombinasi

Rekombinasi atau *crossover* adalah operator dari Algoritma Genetika yang melibatkan dua individu yang terpilih sebelumnya untuk membentuk kromosom atau individu baru yang disebut *offspring* (Huang and Wang 2006). Rekombinasi hanya berlaku pada beberapa individu yang memenuhi syarat. Individu terpilih yang tidak terkena *crossover* akan langsung diturunkan menjadi *offspring*. Prinsip dari *crossover* adalah menggabungkan 2 individu yang telah terpilih untuk mendapatkan individu baru. Beberapa operasi *crossover* yang biasa digunakan diantaranya:

- *One Point Crossover*

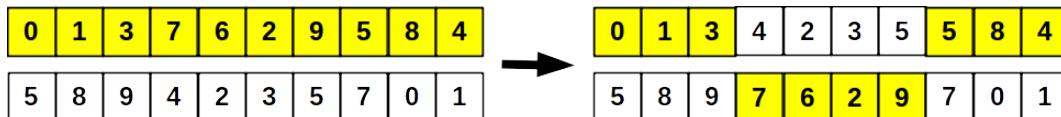
Dalam operasi ini, setiap individu akan dipotong menjadi 2 buah bagian yang kemudian potongan bagian belakang akan ditukar dan disambungkan dengan bagian depan individu lain sehingga menghasilkan individu baru.



Gambar 2.2: Ilustrasi *one point crossover*

- *Multi Point Crossover*

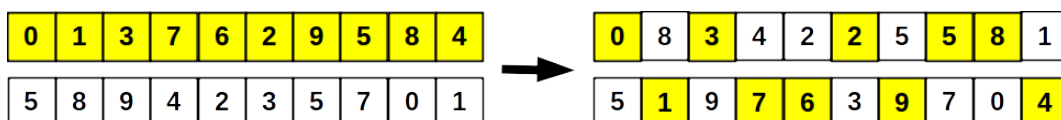
Operasi ini memotong setiap individu terpilih menjadi 3 buah bagian dimana bagian yang ditukar merupakan bagian tengah potongan.



Gambar 2.3: Ilustrasi *multi point crossover*

- *Uniform Crossover*

Operasi ini tidak membagi individu menjadi bagian-bagian, namun setiap gen pada 2 individu tersebut ditukar dengan probabilitas tiap kromosomnya sebesar 50%.



Gambar 2.4: Ilustrasi *uniform crossover*

#### 4. Mutasi

Setelah mengalami proses rekombinasi, pada *offspring* dapat dilakukan mutasi. Variabel *offspring* dimutasi dengan menambahkan nilai random yang sangat kecil (ukuran langkah mutasi), dengan probabilitas yang rendah. Mutasi berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi.

#### 5. Menghitung Nilai *Fitness*

Evaluasi digunakan untuk menghitung nilai kebugaran (*fitness*) setiap kromosom. Semakin besar *fitness* maka semakin baik kromosom tersebut untuk

dijadikan calon solusi. Cara untuk mendapatkan nilai *fitness* berbeda tergantung dalam penyelesaian permasalahan suatu algoritma genetika. Misalnya : persentase kecocokan, eror, total jarak dan sebagainya.

## 6. Penentuan Solusi Optimal

Dalam algoritma genetik, penentuan solusi optimal yang juga merupakan syarat berhentinya keseluruhan proses memiliki beberapa pendekatan diantaranya:

- Sebagian populasi memiliki nilai *fitness* yang sama
- Nilai *fitness* terbaik tidak berubah setelah beberapa iterasi
- Sudah mencapai generasi tertentu
- Nilai *fitness* sudah sama atau melampau nilai yang telah ditentukan sebelumnya

Apabila solusi optimal telah didapat, maka proses algoritma genetik dihentikan.

Namun, apabila solusi yang didapat belum optimal, maka proses algoritma genetik akan berjalan kembali ke proses seleksi.

### ***2.2.4. Ant Colony Optimization***

#### **2.2.4.1. Pengertian**

*Ant Colony Optimization* (ACO) diperkenalkan oleh Moyson dan Manderick dan secara meluas dikembangkan oleh Marco Dorigo. ACO diadopsi dari perilaku koloni semut yang dikenal sebagai sistem semut (Dorigo & Caro,

1999). ACO merupakan algoritma yang dimunculkan sebagai suatu pendekatan multi-agen (*Multi-agent approach*) terhadap optimasi berbagai permasalahan yang berkaitan dengan graf.

Algoritma *Ant Colony Optimization* (ACO) terinspirasi oleh pengamatan terhadap suatu koloni semut. Semut merupakan hewan yang hidup sebagai suatu kesatuan dalam koloninya dibandingkan jika dipandang sebagai individu yang hidup sendiri-sendiri dan tidak bergantung terhadap koloninya. Suatu perilaku penting dan menarik untuk ditinjau dari suatu koloni semut adalah perilaku mereka pada saat mencari makan, terutama bagaimana mereka mampu menentukan rute untuk menghubungkan antara sumber makanan dengan sarang mereka.

Ketika berjalan menuju sumber makanan dan sebaliknya, semut meninggalkan jejak berupa suatu zat yang disebut *Pheromone*. Semut-semut dapat mencium *Pheromone*, dan ketika memilih rute yang akan dilalui, semut akan memiliki kecenderungan untuk memilih rute yang memiliki tingkat konsentrasi *Pheromone* yang tinggi. Jejak *Pheromone* tersebut memungkinkan semut untuk menemukan jalan kembali ke sumber makanan atau sarangnya.

#### **2.2.4.2. Istilah**

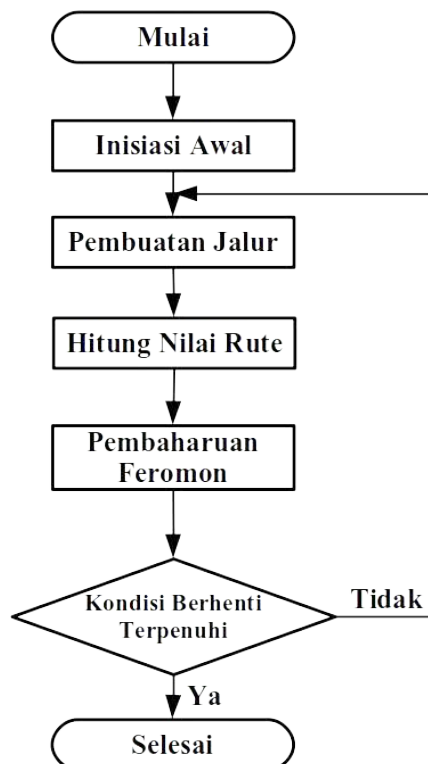
Berikut beberapa definisi atau istilah yang sering digunakan pada algoritma *ant colony optimization*:

1. *Ant* atau semut merupakan agen yang digunakan untuk melakukan perjalanan
2. *Tabu list* merupakan daftar simpul/titik yang dikunjungi oleh semut

3. Feromon/*Pheromone* merupakan jejak yang ditinggalkan oleh semut di setiap jalur yang dilewatinya yang difungsikan sebagai bobot tiap jalur
4. *Evaporation rate* merupakan kecepatan feromon untuk semakin berkurang

#### 2.2.4.3. Langkah Kerja

Proses algoritma genetik terdiri dari beberapa tahap, dua diantaranya adalah seleksi dan reproduksi yang terdiri dari crossover dan mutasi. Secara singkat, proses dalam algoritma genetik adalah sebagai berikut :



Gambar 2.5: Langkah kerja *Ant Colony Optimization*

##### 1. Inisialisasi Awal

Inisialisasi dilakukan untuk menentukan parameter awal yang akan digunakan dalam perhitungan yaitu populasi semut ( $m$ ), nilai feromon awal ( $\tau_0$ ), tingkat

evaporasi/penguapan feromon ( $\rho$ ), konstanta  $\alpha$  sebagai bobot dari feromon dan  $\beta$  sebagai bobot dari fungsi objektif.

## 2. Pembuatan Jalur

Pada fase ini semut akan bergerak ke titik selanjutnya dengan mempertimbangkan nilai probabilitas tiap jalur dengan persamaan 2.20. Nantinya titik tujuan akan disimpan dalam daftar tabel. Proses ini berlangsung terus hingga semut mencapai titik akhir.

$$P_{ij} = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{j \in US} (\tau_{ij})^\alpha (\eta_{ij})^\beta} & j \in US \\ 0 & \text{Lainnya} \end{cases} \quad (2.20)$$

## 3. Hitung Nilai Rute

Perhitungan nilai tiap jalur dilakukan setelah seluruh semut menyelesaikan pembuatan rute. Nilai setiap rute didapatkan dengan menjumlahkan seluruh skor dari titik yang dilewati rute tersebut. Rute terbaik merupakan rute dengan skor tertinggi.

## 4. Pembaharuan Feromon

Setelah seluruh semut membuat rute untuk mencapai tujuan, selanjutnya dilakukan perhitungan nilai secara individu berdasarkan data pada tabel hasil tiap semut. Hasil tersebut kemudian ditentukan nilai terbaik. Selanjutnya



dilakukan perhitungan pembaharuan feromon pada setiap jalur untuk iterasi selanjutnya menggunakan persamaan 2.21.

$$\tau_{ij}^{new} = \rho \tau_{ij}^{old} + \sum_{k=1}^m \Delta \tau_{ij}^k + e \Delta \tau_{ij}^{bs} \quad (2.21)$$

Persamaan 2.2 menunjukkan perhitungan pembaharuan nilai feromon dimana

$\Delta \tau_{ij}^k$  merupakan total nilai feromon pada jalur pada semut k, sedangkan

$\Delta \tau_{ij}^{bs}$  merupakan nilai feromon pada jalur terbaik sementara pada keseluruhan iterasi. Nilai  $e$  merupakan nilai konstan yang menyatakan bobot dari jalur terbaik.

##### 5. Kondisi Berhenti

Proses evaluasi melakukan pengecekan apakah hasil yang diterima merupakan hasil terbaik dari beberapa iterasi. Apabila hasil sudah optimal, maka algoritma ini akan berhenti dan menampilkan hasilnya. Jika tidak, maka daftar tabel tiap semut akan dikosongkan dan memulai lagi fase perjalanan dari titik awal.

## BAB V

### SIMPULAN DAN SARAN

#### 5.1. Simpulan

Berdasarkan hasil penelitian dan pembahasan, dapat ditarik kesimpulan, Dengan diterapkannya penggabungan algoritma secara berurutan, maka:

- Hasil jumlah titik yang didapat mampu ditingkatkan hingga sebesar 20% (6 titik) dibandingkan dengan algoritma *any colony optimization*, dan 30% - 80% (4 – 14 titik) dibandingkan dengan algoritma genetik.
- Waktu perhitungan mampu dipercepat hingga sebesar 3-5 detik dibandingkan dengan algoritma *ant colony optimization*. Namun jika dibandingkan dengan algoritma genetik, waktu perhitungan algoritma gabungan masih terpaut sangat jauh, dikarenakan algoritma genetik hanya memerlukan waktu kurang dari 1 detik pada sebagian besar skenario. Sedangkan algoritma gabungan memerlukan waktu perhitungan lebih dari 6 detik.

#### 5.2. Saran

Berdasarkan keterbatasan mengenai penelitian yang telah dilaksanakan, ada beberapa saran yang dapat dipertimbangkan pada penelitian selanjutnya yaitu :

1. Melakukan modifikasi skenario yang lebih banyak untuk menguji tingkat optimalisasi algoritma gabungan secara lebih mendetail

2. Melakukan pengujian parameter yang digunakan dalam algoritma genetik dan *ant colony optimization* secara lebih mendalam
3. Untuk penelitian selanjutnya dapat dilakukan pengujian dengan penggabungan algoritma lainnya untuk mengembangkan algoritma yang digunakan meningkatkan hasil yang didapat.

## DAFTAR PUSTAKA

- Baranowski, K., Koszałka, L., Poźniak-Koszałka, I. & Kasprzak, A. 2014. Ant Colony Optimization Algorithm for Solving the Provider - Modified Traveling Salesman Problem. Springer, Cham, hal.493–502. Tersedia di [http://link.springer.com/10.1007/978-3-319-05476-6\\_50](http://link.springer.com/10.1007/978-3-319-05476-6_50) [Accessed 5 Maret 2018].
- Bochar, L. & Radi, B. 2016. A new approach to treat the selective travelling salesman problem. *International Mathematical Forum*, 11(January 2018): 757–768. Tersedia di <http://www.m-hikari.com/imf/imf-2016/13-16-2016/6665.html>.
- Chaparro, I. & Valdez, F. 2013. Variants of Ant Colony Optimization: A Metaheuristic for Solving the Traveling Salesman Problem. Springer, Berlin, Heidelberg, hal.323–331. Tersedia di [http://link.springer.com/10.1007/978-3-642-33021-6\\_26](http://link.springer.com/10.1007/978-3-642-33021-6_26) [Accessed 5 Maret 2018].
- Chaqiqi, C., Widodo, A.W. & Sari, Y.A. 2018. Penerapan Evolution Strategies untuk Optimasi Travelling Salesman Problem With Time Windows pada Sistem Rekomendasi Wisata Malang Raya. 2(7).
- Engin, O. & Güçlü, A. 2018. A NEW HYBRID ANT COLONY OPTIMIZATION ALGORITHM FOR SOLVING THE NO-WAIT FLOW SHOP SCHEDULING PROBLEMS. *Applied Soft Computing Journal*. Tersedia di <https://doi.org/10.1016/j.asoc.2018.08.002>.
- Feillet, D., Dejax, P. & Gendreau, M. 2001. Traveling salesman problems with profits: an overview. 1–14.
- Gavalas, D., Konstantopoulus, C., Mastakas, K. & Pantziou, G. 2014. A survey on algorithmic approaches for solving tourist trip design problems. *Journal of Heuristics*, 20(3): 281–328.
- Golden, B., Raghavan, S. & Wasil, E. 2008. Using a Genetic Algorithm to Solve the Generalized Orienteering Problem. *Operations Research/ Computer Science Interfaces Series*, 43: 263–274.
- Karbowska-chili, J. 2012. Genetic Algorithm Solving Orienteering Problem in Large Networks. 28–38.

- Kobeaga, G., Lozano, J.A. & Lozano, J.A. 2017. An Efficient Evolutionary Algorithm For The Orienteering Problem. *Computer and Operations Research*.
- Kong, H., Dong, Y., Gu, J., Li, N., Hou, X. & Yan, W. 2007. Combination of genetic algorithm and ant colony algorithm for distribution network planning. (August): 19–22.
- Koszelew, J. & Ostrowski, K. 2013. A genetic algorithm with multiple mutation which solves orienteering problem in large networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8083 LNAI: 356–366.
- Laporte, G. & Martello, S. 1990. The selective travelling salesman problem. *Discrete Applied Mathematics*, 26(2–3): 193–207.
- Lee, Z.J., Su, S.F., Chuang, C.C. & Liu, K.H. 2008. Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment. *Applied Soft Computing Journal*, 8(1): 55–78.
- Luan, J., Yao, Z., Zhao, F. & Song, X. 2019. ScienceDirect A novel method to solve supplier selection problem : Hybrid algorithm of genetic algorithm and ant colony optimization. *Mathematics and Computers in Simulation*, 156: 294–309. Tersedia di <https://doi.org/10.1016/j.matcom.2018.08.011>.
- Luo, W., Lin, D. & Feng, X. 2016. An Improved Ant Colony Optimization and Its Application on TSP Problem. *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 136–141. Tersedia di <http://ieeexplore.ieee.org/document/7917076/>.
- Ning, J., Zhang, Q., Zhang, C. & Zhang, B. 2018. A best-path-updating information-guided ant colony optimization algorithm. *Information Sciences*, 433–434: 142–162. Tersedia di <https://doi.org/10.1016/j.ins.2017.12.047>.
- Ostrowski, K., Karbowska-chilinska, J., Koszelew, J. & Zabielski, P. 2016. Evolution-inspired local improvement algorithm solving orienteering problem. *Annals of Operations Research*.
- Piwońska, A. & Koszelew, J. 2011. A Memetic Algorithm for a Tour Planning in the Selective Travelling Salesman Problem on a Road Network. Springer,

- Berlin, Heidelberg, hal.684–694. Tersedia di [http://link.springer.com/10.1007/978-3-642-21916-0\\_72](http://link.springer.com/10.1007/978-3-642-21916-0_72) [Accessed 4 Maret 2018].
- Piwonska, A. & Sereczynski, F. 2011. A Genetic Algorithm with a Penalty Function in the Selective Travelling Salesman Problem on a Road Network. *2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum*, 381–387. Tersedia di <http://ieeexplore.ieee.org/document/6008855/>.
- Priandani, N.D. & Mahmudi, W.F. 2016. Rekomendasi jadwal dan paket rute wisata di Pulau Bali menggunakan algoritma genetika. (January 2015).
- Sorensen, K. 2015. Metaheuristics — the metaphor exposed. *International Transactions in Operational Research*, 22: 3–18.
- Sukmana, Y. 2017. 3 Tahun Jokowi-JK, Pariwisata Sumbang Devisa Terbesar Kedua - Kompas.com. Tersedia di <https://ekonomi.kompas.com/read/2017/10/17/220236426/3-tahun-jokowi-jk-pariwisata-sumbang-devisa-terbesar-kedua> [Accessed 4 Maret 2018].
- Tsai, Y.L., Yang, Y.J. & Lin, C.H. 2010. A dynamic decision approach for supplier selection using ant colony system. *Expert Systems with Applications*, 37(12): 8313–8321. Tersedia di <http://dx.doi.org/10.1016/j.eswa.2010.05.053>.
- Unwto 2017. Tourism highlights. *E-unwto*, (August): 16. Tersedia di [www.e-unwto.org/content/hq/4538/fulltext.pdf](http://www.e-unwto.org/content/hq/4538/fulltext.pdf).
- Vansteenwegen, P. & Oudheusden, D. Van 2007. The Mobile Tourist Guide: An OR Opportunity. 20(3).
- Vansteenwegen, P., Souffriau, W., Berghe, G. Vanden & Oudheusden, D. Van 2011. The City Trip Planner: An expert system for tourists. *Expert Systems With Applications*, 38(6): 6540–6546. Tersedia di <http://dx.doi.org/10.1016/j.eswa.2010.11.085>.
- Vazirani, V. V. 2003. *Approximation Algorithms*. Springer.
- Wei, Z. 2012. The Research of Genetic Ant Colony Algorithm and Its Application. *Procedia Engineering*, 37: 101–106.

- Zhao, F., Yao, Z., Luan, J. & Song, X. 2016. A Novel Fused Optimization Algorithm of Genetic Algorithm and Ant Colony Optimization. *Mathematical Problems in Engineering*, 2016.
- Zhu, S., Dong, W. & Liu, W. 2014. Logistics distribution route optimization based on genetic ant colony algorithm. *Journal of Chemical and Pharmaceutical Research*, 6(6): 2264–2267.
- Zukhri, Z., Islam, U. & Zukhri, Z. 2013. A Hybrid Optimization Algorithm based on Genetic Algorithm and Ant Colony Optimization. *International Journal of Artificial Intelligence and Application*, 4(September 2013): 63–75.