

**Briquette Characteristics of Mixed Charcoal of Taro Peel and Rice Husk**  
Galu Murdikaningrum, Mutiara Putri Utami Susanto, Raden Tarisa Nurhanifah, Mualifah (pp:126–136)

**Analysis of Interest In Using Electrical Bicycles in Palangka Raya**  
Vivien Nopella Valentina, Robby, Sutan Parasian Silitonga (pp: 137–144)

**The Effect Of H<sub>2</sub>O<sub>2</sub> On The Bleaching-Scouring Simultaneous Process Of 100% Cotton Fabric With Pad – Batch System**  
Luciana, Agni Salamah (pp: 145–153)

**Management of Water Quality Parameters In Cultivating Vaname Shrimp (Litopenaeus Vannamei) In Intensive Tambak PT. Aneka Tambak Oseana Nusantara , NTB**  
Pieter Amalo, Riris Yuli Velentine, Catur Pramono Adi, Restye Putri Geofani Mbura (pp: 154–162)

**Analysis of The Ant Number Effects on Ant Colony Optimization for Solving Russia-20-Nodes-SDVRP Instance**  
Ekra Sanggala, Muhammad Ardhya Bisma (pp: 163–174)

**Facilities Re-layout of “X” Health Center**  
Dini Yulianti, Tombak Gapura Bhagya, Didi Kusvendi (pp: 175–186)

**Use of Gold Mine Waste Sand From Penda Pilang Village as Hrs-Base Mixture**  
Deskianto, Supiyan, Devia (pp: 187–199)

**In Vitro Antagonism Test of Endophytic Isolates From The Ciplukan Plant (Physalis Angulata L.) Against Ralstonia Solanacearums**  
Ika Afifah Nugraheni, Inneke Ashri Mawaddah (pp: 200–210)

**Employee Performance Measurement at PT. Cahaya Mekanindo Perkasa Using the Human Resources Scorecard Method**  
Ilyas Habibi, AuliaFashanah Hadining (pp: 211–219)

**The Effect of Temperature Variations in the Pressing Process on Glossing Defects Bigborn 2-Tuck Pants Style 3651 Trousers in the Finishing Department of PT. X**  
Afriani Kusumadewi, Feny Nurherawati, Filly Pravitarsari (pp: 220–227)

**Optimum Splice Thickness Ratio Splicer of a Winding Machine to PE20KT Thread Splicing Quality**  
Hendri Pujiyanto, Bambang Yulianto, Hamdan S Bintang, Dinda Amelia Pramesti (pp: 228–235)

**The Influence of Organizational Culture and Organizational Commitment to Employee Work Discipline at the Bandung City Transportation Service**  
Moch Ruli Chaerudin, Riza Rizkiah (pp: 236–245)

**Feasibility Analysis of Smelter Grade Alumina (SGA) Project Development at PT. X**  
Dio Rianto, Dedy Setyo Oetomo, Rizky Fajar, Ramdhani (pp: 246–258)

**Evaluation of Decision Making on Using Online Media in the D'Amerta Berniaga Bandung Business Group**  
Alam Avrianto, Ira Murwenie, Rahmina Puspa AR, Dwirani Fauzi L, Abdul Fatah H (pp: 259–265)

**5 Year Effectiveness Index From Research Ministry Of Marine And Fisheries**  
Catur Pramono Adi, Pieter Amalo (pp: 266–273)

**Technology Acceptance Model for the Use of Learning Management System in Indonesia**  
Graha Prakarsa, Elly Komala, Tombak Gapura Bhagya, Safira Noor Andinia (pp: 274–284)

## *Analysis of The Ant Number Effects on Ant Colony Optimization for Solving Russia-20-Nodes-SDVRP Instance*

### **Analisis Pengaruh Jumlah Semut Pada *Ant Colony Optimization* Untuk Penyelesaian *Russia-20-Nodes-SDVRP Instance***

Ekra Sanggala<sup>1\*)</sup>, Muhammad Ardhya Bisma<sup>2)</sup>

<sup>1)</sup>Universitas Logistik dan Bisnis Internasional, Jalan Sariasih No. 54, Bandung, 40151  
Email: [ekrasanggala@mail.ru](mailto:ekrasanggala@mail.ru)

<sup>2)</sup> Universitas Logistik dan Bisnis Internasional, Jalan Sariasih No. 54, Bandung, 40151  
Email: [bisma@ulbi.ac.id](mailto:bisma@ulbi.ac.id)

\*) *Corresponding author*

**Abstract:** *SDVRP is a variation of the CVRP that allows different vehicles to visit the same node. ACO performance is influenced by its parameter value. The number of ants parameter has a strong influence on ACO Performance, because it influences directly on solution quality and computation time. It is too hard to find the parameter value that can make ACO performance being the best performance. Because of this, what can be done is to test several parameter values, then compare the ACO performance of each parameter value. Russia-20-Nodes-SDVRP Instance is attractive to be solved because its number of nodes are not too many, only 20 nodes. Based on that number of nodes, the computation time of this instance is relatively fast and it has a lot of solution candidate. This research focus on the ant number effects on ACO for Solving Russia-20-Nodes-SDVRP Instance. Five ant numbers will be tested, they are 5, 10, 15, 20 and 25 ants, each ant number will be tested 10 times, so there will be 50 tests. These tests show that more ants used, greater possibility of getting a better route, but longer calculation time needed. The decrease in average of length of route is 1,1521% while the increase in average of calculation time is 42.5525%.*

**Keywords:** *SDVRP, ACO, Parameter ACO, The Number of Ant*

**Abstrak:** *SDVRP merupakan pengembangan dari CVRP yang memungkinkan sebuah titik dikunjungi lebih dari sekali untuk pemenuhan kebutuhannya. Performansi dari ACO sangat dipengaruhi oleh nilai parameter-parameternya. Parameter jumlah semut sangat berpengaruh terhadap performansi ACO karena berpengaruh langsung terhadap kualitas solusi yang dihasilkan dan waktu perhitungan. Tidak dapat diketahui nilai-nilai parameter yang dapat menjadikan performansi ACO menjadi performansi yang terbaik. Dikarenakan hal ini, yang dapat dilakukan adalah dengan menguji beberapa nilai-nilai parameter dan kemudian membandingkan performansi ACO untuk setiap nilai-nilai parameter. Russia-20-Nodes-SDVRP Instance menjadi menarik untuk diselesaikan karena jumlah titiknya tidak terlalu banyak, hanya 20 titik saja. Dengan jumlah titik tersebut, menjadikan instance tersebut waktu perhitungannya relatif cepat dan mempunyai kemungkinan solusi yang sangat banyak. Penelitian ini berfokus pada analisis pengaruh jumlah semut terhadap performansi ACO dalam menyelesaikan Russia-20-Nodes-SDVRP Instance. Jumlah semut yang akan diujikan adalah 5, 10, 15, 20 dan 25 semut, pada setiap jumlah semut akan dilakukan 10 kali percobaan, dengan demikian akan ada sebanyak 50 percobaan. Dari percobaan-percobaan tersebut terlihat bahwa semakin banyak semut maka semakin besar peluang mendapatkan rute yang lebih baik, tetapi semakin lama waktu perhitungannya. Penurunan rata-rata panjang rute adalah 1,1521% sedangkan kenaikan rata-rata waktu perhitungan adalah 42,5525%.*

**Kata Kunci:** *SDVRP, ACO, Parameter ACO, Jumlah Semut*

DOI: <http://dx.doi.org/10.37577/sainteks.v%vi%i.608>

Received: 07, 2023. Accepted: 08, 2023.

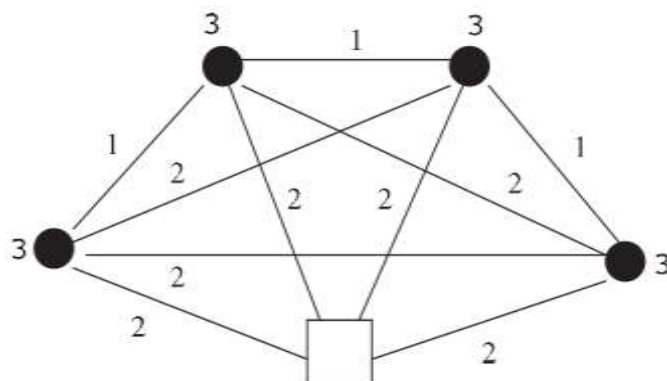
Published: 09, 2023.

## PENDAHULUAN

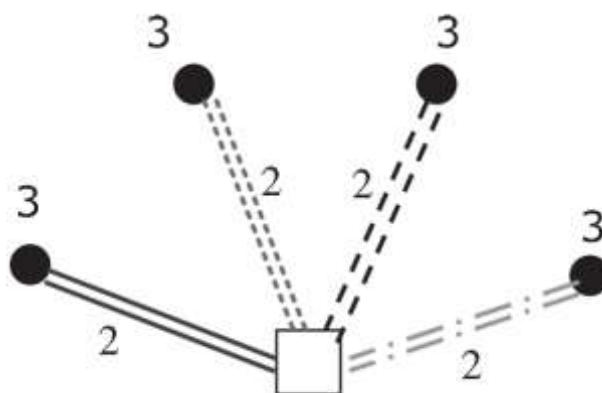
Seiring dengan adanya perkembangan teknologi dan ekonomi, logistik tidak lagi hanya untuk pengiriman saja, tuntutan kebutuhan dan keterbatasan logistik perlu dipertimbangkan. Pada penerapan *Vehicle Routing Problem (VRP)* di dunia nyata, terdapat banyak keterbatasan yang perlu dipertimbangkan, seperti keterbatasan kapasitas kendaraan, waktu perjalanan kendaraan, keterbatasan waktu dan lain-lain. Keterbatasan-keterbatasan ini menyebabkan lahirnya berbagai jenis *VRP*, seperti *Capacity Vehicle Routing Problem (CVRP)*, *Periodic Vehicle Routing Problem (PVRP)*, *Multiple Depot Vehicle Routing Problem (MDVRP)*, *Split Delivery Routing Problem (SDVRP)*, *Vehicle Routing Problem with Time Windows (VRPTW)* dan *Vehicle Routing Problem with Simultaneous Pickups and Deliveries (VRPSPD)* (Chen, 2018).

*Split Delivery Vehicle Routing Problem (SDVRP)* merupakan pengembangan dari *Capacitated Vehicle Routing Problem (CVRP)* yang memungkinkan sebuah titik dikunjungi lebih dari sekali untuk pemenuhan kebutuhannya. Pada *SDVRP*, kebutuhan dari sebuah titik mungkin lebih banyak dari pada kapasitas kendaraan. Dengan demikian dimungkinkan ada titik yang dikunjungi oleh beberapa kendaraan dan pemenuhan kebutuhan titik tersebut dibagi pada kendaraan-kendaraan tersebut, bahkan hal ini dapat dilakukan pada titik yang kebutuhannya tidak melebihi kapasitas kendaraan. (Shi, 2018).

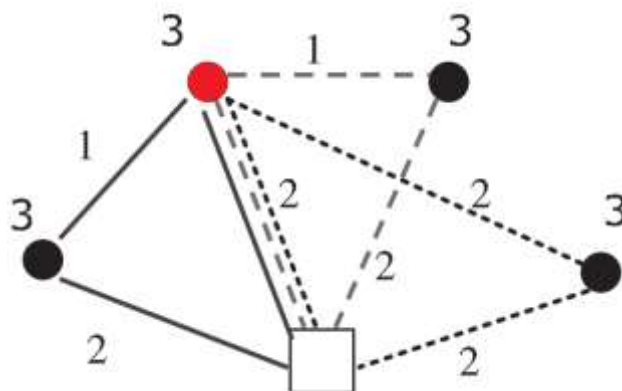
Ilustrasi perbedaan antara *VRP* dengan *SDVRP* dapat dilihat pada gambar 1. Diketahui bahwa ada 1 depot dengan 4 titik yang kebutuhannya perlu dipenuhi. Titik-titik tersebut mempunyai kebutuhan sebanyak 3 produk, dan diketahui kapasitas kendaraan yang digunakan paling banyak mampu membawa 4 produk (Archetti, 2012).



(a) Data jarak antar titik & kebutuhan setiap titik.



(b) Penyelesaian *VRP* dengan 4 rute dan total jarak 16.

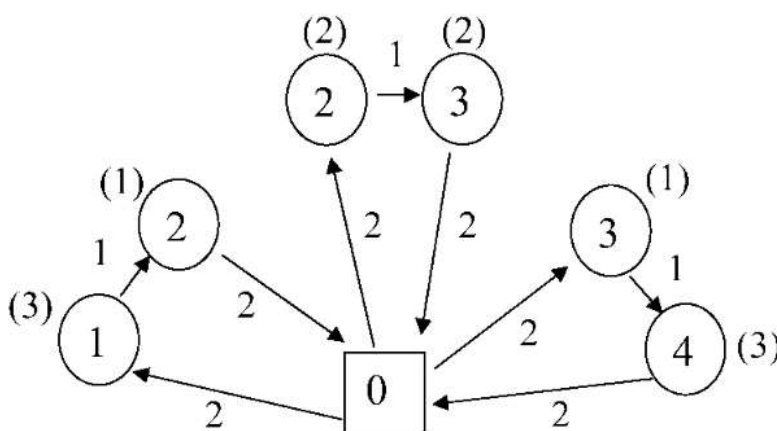


(c) Penyelesaian *SDVRP* dengan 3 rute dan total jarak 16.

**Gambar 1: Ilustrasi Perbedaan Antara *VRP* dengan *SDVRP* (Archetti, 2012).**

Pada gambar 1(b) terlihat bahwa untuk memenuhi kebutuhan setiap titik dilakukan dengan sekali kunjungan, sedangkan pada gambar 1(c) terlihat bahwa ada sebuah titik yang pemenuhan kebutuhannya dilakukan dengan tiga kali kunjungan, titik tersebut ditandai dengan warna merah.

Dengan adanya kemungkinan pemenuhan kebutuhan sebuah titik dilakukan oleh lebih dari satu kali kunjungan, mengakibatkan munculnya potensi untuk mengurangi jumlah rute kunjungan dan mengurangi total jaraknya (Chen, 2007). Pada gambar 2, terlihat penyelesaian *SDVRP* yang dapat melakukan pengurangan jumlah rute kunjungan dan pengurangan total jaraknya dalam menyelesaikan permasalahan yang terdapat pada gambar 1.



**Gambar 2: Penyelesaian *SDVRP* dengan 3 rute dan total jarak 15 (Chen, 2007).**

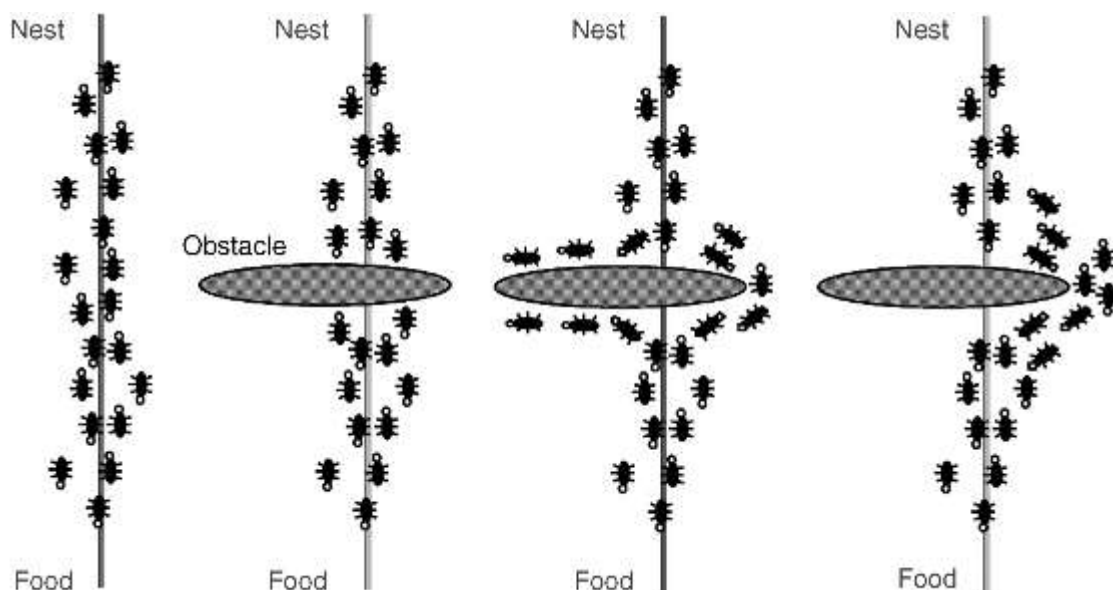
Penyelesaian *SDVRP* pada gambar 1(c) mempunyai total jarak yang sama dengan penyelesaian *VRP* pada gambar 1(b), dengan demikian penyelesaian *SDVRP* ini belum mampu mengurangi total jarak, sedangkan penyelesaian *SDVRP* pada gambar 2 mempunyai total jarak 15, dengan demikian penyelesaian *SDVRP* ini mampu mengurangi total jarak.

Pada *SDVRP* jumlah produk yang dibawa oleh kendaraan untuk titik yang dikunjungi lebih dari satu kali, menjadi sebuah variabel. Hal ini menyebabkan meningkatnya kompleksitas *VRP*. Kompleksitas ini menjadikan *SDVRP* sebuah *NP-Hard Problem (Nondeterministic polynomial time)* (Ozbaygin, 2018). Dalam menyelesaikan *NP-Hard Problem* diperlukan waktu yang sangat lama, yang tidak wajar untuk ditunggu (Applegate, 2006). Meskipun demikian *NP-Hard Problem* dapat diselesaikan dalam jangka waktu yang wajar dengan menggunakan algoritma *heuristic* serta *metaheuristic* meskipun solusi yang dihasilkan belum tentu yang terbaik (Labadie, 2016).

*Heuristic* merupakan sebuah kata dari bahasa Yunani yang berarti “menemukan” atau “mengeksplorasi”. *Heuristics* disebut sebagai *approximate techniques*. Tujuan utama dari *heuristics* adalah untuk membangun sebuah model optimasi yang mudah dipahami dan mampu memberikan solusi yang baik dalam waktu perhitungan yang wajar (Kumar, 2020). Sufiks *meta*, juga berasal dari bahasa Yunani yang berarti “upper level methodology”. Maka *metaheuristic* dapat didefinisikan sebagai sebuah *template* yang dapat dikembangkan menjadi *heuristic* untuk menyelesaikan sebuah masalah optimasi (Talbi, 2009).

*Ant Colony Optimization (ACO)* merupakan sebuah *metaheuristic* dimana sekumpulan semut tiruan bekerja sama untuk menemukan solusi yang baik dalam menyelesaikan masalah optimasi (Dorigo, 2004). Konsep *ACO* ini terinspirasi dari perilaku semut. Semut adalah serangga yang hidup secara berkoloni dan berperilaku untuk kepentingan koloni. Hal menarik dari perilaku semut adalah kemampuannya untuk menemukan rute terpendek antara sarangnya dengan sumber makanan (Santosa, 2017).

Perilaku semut dalam mencari makanan dapat dideskripsikan seperti berikut: Sekelompok semut yang berada di depan harus memilih jalur tertentu untuk dilewati. Pada saat semut pertama berjalan, semut tersebut meninggalkan pheromone yang dapat dicium oleh semut-semut dibelakangnya, sehingga semut-semut tersebut dapat mengetahui apakah jalur tersebut sudah terlewati atau belum. Mereka pun selama berjalan meninggalkan pheromone pada jalur yang dilewatinya. Semakin banyak semut yang melewati suatu jalur, maka semakin pekat pheromone yang ditinggalkan pada jalur tersebut. Kepekatan pheromone ini pun dipengaruhi oleh adanya penguapan pada pheromone. Semakin pekat pheromone pada suatu jalur, maka semakin besar peluang jalur tersebut akan dipilih untuk dilewati. Pada gambar 3 dapat dilihat gambaran sekelompok semut dalam memilih jalur untuk menemukan makanan.



**Gambar 3: Gambaran Semut Dalam Memilih Jalur Untuk Menemukan Makanan (Talbi, 2009).**

Peluang suatu jalur untuk dilewati oleh semut dapat dituliskan pada persamaan berikut ini (Gao, 2020):

$$P_{ij}^k(t) = \frac{\tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta}{\sum_{j \in N_i^k} \tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta}, \text{ if } j \in N_i^k$$

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

- $P_{ij}^k(t)$  : peluang semut  $k$  memilih jalur dari titik  $i$  ke titik  $j$   
 $t$  : iterasi  
 $k$  : semut  $k$   
 $i$  : titik  $i$   
 $j$  : titik  $j$   
 $\alpha$  : derajat kepentingan pheromone  
 $\beta$  : derajat kepentingan kebalikan jarak antar titik  
 $N_i^k$  : titik-titik yang dapat dipilih oleh semut  $k$  pada saat berada di titik  $i$   
 $\tau_{ij}$  : kepekatan pheromone jalur dari titik  $i$  ke titik  $j$   
 $\eta_{ij}$  : kebalikan jarak jalur dari titik  $i$  ke titik  $j$   
 $d_{ij}$  : jarak antara titik  $i$  dengan titik  $j$

Kepekatan pheromone pada suatu jalur setelah terjadinya penguapan pheromone dapat dituliskan pada persamaan berikut ini (Gao, 2020):

$$\tau_{ij}(t + 1) = (1 - \rho)\tau_{ij}(t)$$

- $\tau_{ij}$  : kepekatan pheromone jalur titik  $i$  ke titik  $j$   
 $\rho$  : derajat penguapan pheromone  
 $t$  : iterasi

Banyaknya pheromone yang ditinggalkan seekor semut pada suatu jalur dapat dituliskan pada persamaan berikut ini (Gao, 2020):

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k}(t), & \text{if ant } k \text{ travels on edge } (i, j) \\ 0, & \text{otherwise} \end{cases}$$

- $\Delta\tau_{ij}^k$  : banyaknya pheromone yang ditinggalkan semut  $k$  pada jalur titik  $i$  ke titik  $j$   
 $Q$  : konstanta  
 $L^k$  : panjang rute yang dilalui oleh semut  $k$   
 $t$  : iterasi

Kepekatan pheromone pada suatu jalur setelah adanya pheromone yang ditinggalkan semut yang melaluinya dapat dituliskan pada persamaan berikut ini (Gao, 2020):

$$\tau_{ij}(t + 1) = \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t)$$

- $\tau_{ij}$  : kepekatan pheromone jalur titik  $i$  ke titik  $j$   
 $\Delta\tau_{ij}^k$  : banyaknya pheromone yang ditinggalkan semut  $k$  pada jalur titik  $i$  ke titik  $j$   
 $m$  : banyaknya semut  
 $t$  : iterasi

Proses pada persamaan-persamaan diatas akan terus berulang hingga mencapai jumlah iterasi  $R$  (Gao, 2020).

Berdasarkan persamaan-persamaan diatas dapat diketahui bahwa parameter-parameter untuk  $ACO$  adalah sebagai berikut ini (Wang, 2021):

1.  $\alpha$  : derajat kepentingan pheromone
2.  $\beta$  : derajat kepentingan kebalikan jarak antar titik
3.  $\rho$  : derajat penguapan pheromone ( $0 \leq \rho \leq 1$ )
4.  $\tau_{ij}(1)$  : kepekatan pheromone pada iterasi 1
5.  $m$  : jumlah semut
6.  $R$  : maksimal iterasi

Nilai-nilai parameter ini dapat mempengaruhi performansi dari  $ACO$ . Biasanya pengukuran performansi tersebut adalah berdasarkan solusi yang dihasilkan dan waktu perhitungan yang diperlukan. Parameter jumlah semut merupakan parameter yang sangat berpengaruh terhadap performansi dari  $ACO$  karena berpengaruh langsung terhadap solusi yang dihasilkan dan waktu perhitungan. Semakin banyak jumlah semut, maka semakin besar peluang untuk mendapatkan solusi yang lebih baik, tetapi mengakibatkan waktu perhitungan semakin lama. Hal ini terjadi karena semakin banyak semut maka semakin banyak rute yang dicoba, tetapi hal ini mengakibatkan semut memerlukan waktu lebih lama untuk mencoba banyak rute. Berbeda dengan parameter-parameter lainnya yang hanya berpengaruh langsung terhadap waktu perhitungan karena berapa pun nilainya, jumlah rute yang dicoba tetap sama.

Tidak dapat diketahui nilai-nilai parameter yang dapat menjadikan performansi  $ACO$  menjadi performansi yang terbaik. Penentuan nilai-nilai parameter ini tidak ada ketentuannya, sehingga nilai-nilai ini ditentukan langsung oleh pengguna  $ACO$ . Dikarenakan hal ini, yang dapat dilakukan adalah dengan menguji beberapa nilai-nilai parameter dan kemudian membandingkan performansi  $ACO$  untuk setiap nilai-nilai parameter.

Sebuah  $SDVRP Instance$  yang dibuat oleh penulis adalah ***Russia-20-Nodes-SDVRP Instance***. *Instance* ini dibuat berdasarkan 20 kota berpenduduk terbanyak di *Russia* (Sanggala, 2023). *Instance* ini menjadi menarik untuk diselesaikan karena jumlah titiknya tidak terlalu banyak, hanya 20 titik saja. Dengan jumlah titik sebanyak 20 titik, menjadikan *instance* tersebut sebagai permasalahan yang menarik untuk dilakukannya pengujian nilai-nilai parameter  $ACO$ , karena waktu perhitungannya yang relatif cepat dan mempunyai kemungkinan solusi yang sangat banyak.

Penelitian ini berfokus pada analisis pengaruh jumlah semut terhadap performansi  $ACO$  dalam menyelesaikan ***Russia-20-Nodes-SDVRP Instance***. Performansi  $ACO$  yang akan diukur adalah solusi yang dihasilkan dan waktu perhitungan. Dengan demikian diharapkan penelitian ini dapat menjadi acuan dalam penentuan jumlah semut dalam  $ACO$ .

## **METODOLOGI**

Secara umum tahapan-tahapan yang dilakukan adalah sebagai berikut ini:

1. Penentuan *Instance*

*Instance* merupakan sekumpulan data awal yang akan menjadi acuan pengerjaan pada penelitian ini. *Instance* yang akan digunakan adalah ***Russia-20-Nodes-SDVRP Instance***. Pada tabel 1 dapat dilihat data yang terdapat pada *instance* tersebut.

**Tabel 1: Russia-20-Node-SDVRP Instance**

NAME	: Russia 20 Nodes
TYPE	: SDVRP
DIMENSION	: 20
NODE_NO OF DEPOT	: 1
CAPACITY	: 100
SOURCE	: GeoName (geonames.org)
EDGE WEIGHT TYPE	: Haversine Formula, rounded to smallest integer not less than x
CREATOR	: Sanggala
CREATED DATE	: 1st June 2022
COMMENT	:

Node_No	Name	Latitude	Longitude	Demand
1	Moscow	55,75222	37,61556	0
2	Saint Petersburg	59,93863	30,31413	26
3	Novosibirsk	55,04150	82,93460	32
4	Yekaterinburg	56,85190	60,61220	14
5	Nizhniy Novgorod	56,32867	44,00205	35
6	Samara	53,20007	50,15000	10
7	Omsk	54,99244	73,36859	20
8	Kazan	55,78874	49,12214	14
9	Rostov-na-Donu	47,23135	39,72328	15
10	Chelyabinsk	55,15402	61,42915	33
11	Ufa	54,74306	55,96779	19
12	Volgograd	48,71939	44,50183	23
13	Perm	58,01046	56,25017	18
14	Krasnoyarsk	56,01839	92,86717	26
15	Saratov	51,54056	46,00861	32
16	Voronezh	51,67204	39,18430	17
17	Tol'yatti	53,53030	49,34610	14
18	Krasnodar	45,04484	38,97603	11
19	Ulyanovsk	54,32824	48,38657	22
20	Izhevsk	56,84976	53,20448	19

2. Penentuan Nilai-Nilai Parameter Jumlah Semut

Karena yang akan dianalisis adalah pengaruh nilai parameter jumlah semut terhadap performansi *ACO*, maka nilai-nilai parameter ini akan berbeda-beda. Nilai-nilai parameter tersebut adalah 5 semut, 10 semut, 15 semut, 20 semut dan 25 semut.

3. Penentuan Nilai-Nilai Parameter Selain Parameter Jumlah Semut

Berikut ini adalah nilai-nilai parameter selain parameter jumlah semut:

$$\begin{aligned} \alpha &= 2 \\ \beta &= 2 \\ \rho &= 0,01 \\ \tau_{ij}(1) &= 10 \\ R &= 20 \end{aligned}$$

Nilai-nilai parameter tersebut akan selalu tetap pada setiap percobaan.

4. Pengujian *ACO* Terhadap *Instance*

Dengan menggunakan nilai-nilai parameter yang sudah ditentukan, maka dilakukan pengujian *ACO* terhadap *Russia-20-Nodes-SDVRP Instance*. Pada setiap percobaan

dilakukan pencatatan solusi yang dihasilkan dan pencatatan waktu perhitungan yang diperlukan. Untuk setiap nilai parameter jumlah semut akan dilakukan 10 kali percobaan. Dengan demikian pada pengujian ini akan terdapat 50 kali percobaan.

5. Analisis Pengaruh Jumlah Semut Terhadap Performansi *ACO*

Berdasarkan pencatatan yang dilakukan pada tahap ke-4 dilakukan analisis pengaruh jumlah semut terhadap performansi *ACO*. Analisis tersebut dilakukan terhadap nilai-nilai berikut ini:

- a) Panjang rute terpendek
- b) Panjang rute terjauh
- c) Rata-rata panjang rute
- d) Penurunan rata-rata panjang rute
- e) Waktu perhitungan tercepat
- f) Waktu perhitungan terlama
- g) Rata-rata waktu perhitungan
- h) Kenaikan rata-rata waktu perhitungan

Untuk membantu menyelesaikan pengujian, digunakan perangkat lunak *GNU Octave*. *GNU Octave* merupakan alat multifungsi untuk melakukan analisis numerik. Berbagai alat bantu yang tersedia di dalam *GNU Octave* adalah sebagai berikut ini (Quarterno, 2006):

1. Sekumpulan *function* untuk menyelesaikan berbagai masalah.
2. Bahasa pemrograman yang dapat digunakan untuk mengembangkan kemampuan *GNU Octave*.
3. Berbagai fasilitas untuk melakukan *plotting*.

Nama *GNU Octave* diambil dari seorang ahli kimia yang bernama Octave Levenspiel. *GNU Octave* ini merupakan proyek resmi dari *GNU* dan lisensi *source code* berada di bawah *GNU General Public License (GPL)*, sehingga siapa pun boleh menggunakannya untuk berbagai tujuan (Hansen, 2011).

### HASIL PENELITIAN DAN PEMBAHASAN

Sebelum dilakukan pengujian dilakukan perhitungan untuk menghitung jarak antar titik. Jarak antar titik ini dihitung berdasarkan *Haversine Formula*. *Haversine Formula* merupakan persamaan yang digunakan untuk menghitung jarak antara dua titik di Bumi, bentuk persamaannya adalah seperti berikut ini (Idrizi, 2020):

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_1 \cdot \cos\phi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$c = 2 \cdot \text{atan2}\left[\sqrt{a}, \sqrt{(1-a)}\right]$$

$$d = R \cdot c$$

- $\phi_1$  : *Latitude* titik pertama  
 $\phi_2$  : *Latitude* titik kedua  
 $\lambda_1$  : *Longitude* titik pertama  
 $\lambda_2$  : *Longitude* titik kedua  
 $\Delta\phi$  :  $\phi_1 - \phi_2$   
 $\Delta\lambda$  :  $\lambda_1 - \lambda_2$   
R : Radius Bumi (6371 Km)

Berikut ini merupakan contoh penggunaan *Haversine Formula* dalam menghitung jarak antara dua titik di Bumi:

Moscow (Lat: 55,75222; Long: 37,61556)

St.Petersburg (Lat:59,93863; Long: 30,31413)

$$a = \sin^2 \left( \frac{\Delta\phi}{2} \right) + \cos \phi_1 \cdot \cos \phi_2 \cdot \sin^2 \left( \frac{\Delta\lambda}{2} \right)$$

$$a = 0,002477$$

$$c = 2 \cdot \text{atan2} \left[ \sqrt{0,002477}, \sqrt{0,998761} \right]$$

$$c = 0,099581$$

$$d = 6371 \cdot 0,099581$$

$$d = 634,4309 \approx 635 \text{ km}$$

Dengan menggunakan bantuan *GNU Octave* dilakukan pengujian *ACO* terhadap *instance* sehingga diperoleh hasil seperti yang terlihat pada tabel 2. Bagi para pembaca yang membutuhkan *file* dari *listing program* yang digunakan, dapat langsung menghubungi penulis melalui *e-mail*.

**Tabel 2: Hasil Pengujian *ACO* Pada *Russia-20-Node-SDVRP Instance***

Percobaan	Jumlah Semut	Panjang Rute (Km)	Waktu Perhitungan (Detik)
1	5	19.348	117,2192
2	5	19.844	120,4796
3	5	19.635	118,8260
4	5	19.696	124,6760
5	5	20.246	135,7053
6	5	19.285	131,2748
7	5	21.251	120,8228
8	5	18.867	117,3596
9	5	21.174	125,8928
10	5	19.907	118,6544
11	10	18.127	245,8420
12	10	18.889	237,5271
13	10	20.249	232,5975
14	10	19.271	233,2371
15	10	19.568	262,3937
16	10	18.795	245,0152
17	10	19.264	244,0948
18	10	18.727	245,9356
19	10	20.948	251,3020
20	10	19.817	251,3800

**Tabel 2: Hasil Pengujian ACO Pada Russia-20-Node-SDVRP Instance (Lanjutan)**

<b>Percobaan</b>	<b>Jumlah Semut</b>	<b>Panjang Rute (Km)</b>	<b>Waktu Perhitungan (Detik)</b>
21	15	20.090	393,9649
22	15	18.717	395,4625
23	15	18.643	394,8541
24	15	19.803	424,0419
25	15	18.685	407,6306
26	15	19.332	399,6746
27	15	19.349	420,3915
28	15	19.450	401,0006
29	15	18.993	416,9595
30	15	20.429	406,4606
31	20	19.010	482,6827
32	20	19.931	362,5619
33	20	19.258	431,7172
34	20	19.981	386,8045
35	20	19.296	368,3340
36	20	18.790	368,0688
37	20	19.067	361,1111
38	20	18.757	348,0538
39	20	18.085	350,8618
40	20	20.421	419,9859
41	25	18.563	613,0995
42	25	19.044	610,7751
43	25	19.972	546,8771
44	25	19.090	367,3668
45	25	18.976	320,6757
46	25	20.026	364,0439
47	25	18.654	325,1217
48	25	18.515	405,1970
49	25	18.091	361,5323
50	25	19.254	349,1146

Pada tabel 3 dapat dilihat panjang rute terpendek, panjang rute terjauh, rata-rata panjang rute dan penurunan rata-rata panjang rute

Tabel 3: Panjang Rute Terpendek, Panjang Rute Terjauh, Rata-Rata Panjang Rute dan Penurunan Rata-Rata Panjang Rute

Jumlah Semut	Panjang Rute Terpendek (Km)	Panjang Rute Terjauh (Km)	Rata-Rata Panjang Rute (Km)	Penurunan Rata-Rata Panjang Rute
5	18.867	21.251	19.925,3	0,0000%
10	18.127	20.948	19.365,5	2,8095%
15	18.643	20.429	19.349,1	0,0847%
20	18.085	20.421	19.259,6	0,4626%
25	18.091	20.026	19.018,5	1,2518%
			Rata-Rata	1,1521%

Berdasarkan tabel 3 terlihat bahwa semakin banyak jumlah semut yang digunakan, maka panjang rute terpendek, panjang rute terjauh dan rata-rata panjang rute cenderung mengalami penurunan. Tetapi berdasarkan penurunan rata-rata panjang rute, terlihat juga penurunan ini tidak signifikan, dengan rata-rata penurunan 1,1521% saja.

Pada tabel 4 dapat dilihat waktu perhitungan tercepat, waktu perhitungan terlama, rata-rata waktu perhitungan dan kenaikan rata-rata waktu perhitungan

Tabel 4: Waktu Perhitungan Tercepat, Waktu Perhitungan Terlama, Rata-Rata Waktu Perhitungan dan Kenaikan Rata-Rata Waktu Perhitungan

Jumlah Semut	Waktu Perhitungan Tercepat (Detik)	Waktu Perhitungan Terlama (Detik)	Rata-rata Waktu Perhitungan (Detik)	Kenaikan Rata-Rata Waktu Perhitungan
5	117,2192	135,7053	123,0911	0,0000%
10	232,5975	262,3937	244,9325	98,9848%
15	393,9649	424,0419	406,0441	65,7780%
20	348,0538	482,6827	388,0182	-4,4394%
25	320,6757	613,0995	426,3804	9,8867%
			Rata-Rata	42,5525%

Berdasarkan tabel 4 terlihat bahwa semakin banyak jumlah semut yang digunakan, maka waktu perhitungan tercepat, waktu perhitungan terlama dan rata-rata waktu perhitungan cenderung mengalami kenaikan. Berdasarkan kenaikan rata-rata waktu perhitungan, terlihat kenaikan ini cukup signifikan, dengan rata-rata kenaikan 42,5525%.

Berdasarkan uraian-uraian diatas, diharapkan penelitian ini dapat menjadi acuan para pengguna *ACO* dalam menentukan nilai parameter jumlah semut, jika mementingkan penurunan panjang rute maka dapat digunakan jumlah semut yang banyak dengan resiko kenaikan rata-rata waktu perhitungan yang terus meningkat secara signifikan.

## SIMPULAN

Pendekatan *metaheuristic* seperti *ACO* tidak selalu menampilkan performansi yang terbaik. Performansi yang diukur biasanya adalah solusi yang dihasilkan dan waktu perhitungan. Nilai-nilai parameter sangat berpengaruh terhadap performansi tersebut. Parameter jumlah semut sangat berpengaruh terhadap performansi tersebut, karena berdampak langsung terhadap solusi yang dihasilkan dan waktu perhitungan. Penelitian ini memperlihatkan bahwa penambahan jumlah semut yang digunakan tidak berpengaruh signifikan terhadap peningkatan kualitas solusi, tetapi berpengaruh cukup signifikan terhadap peningkatan waktu perhitungan. Meskipun begitu, pengujian dengan jumlah semut yang lebih banyak dapat dilakukan pada penelitian selanjutnya agar semakin terlihat pengaruh jumlah semut terhadap kualitas solusi yang dihasilkan. Begitu pula, pengujian terhadap nilai parameter-parameter lainnya perlu dilakukan, sehingga semakin banyak acuan dalam menentukan nilai-nilai parameter *ACO*.

## DAFTAR PUSTAKA

- Archetti, A. & Speranza, M. G. (2012). Vehicle Routing Problems With Split Deliveries, *International Transactions In Operational Research*
- Chen, R., & Guo, J. (2018). Optimal Routing for Split Delivery Vehicle Routing Problems, *7<sup>th</sup> International Congress on Advanced Applied Informatics*
- Chen, S., & Golden, B. (2007). The Split Delivery Vehicle Routing Problem: Applications, Algorithms, Test Problems, and Computational Results, *Networks An International Journal*
- Dorigo, M. & Stutzle, T. (2004). Ant Colony Optimization, *The MIT Press Cambridge, Massachusetts*
- Gao, W. (2020). New Ant Colony Optimization Algorithm for The Traveling Salesman Problem, *International Journal of Computational Intelligence Systems*
- Hansen, J.S. (2011). GNU Octave Beginner's Guide, *PACKT – Birmingham*
- Idrizi, B. (2020). Necessity for Geometric Corrections of Distances in Web and Mobile Maps, *International Conference on Cartography and GIS*
- Labadie, N., Prins, C., & Prodhon, C. (2016). Metaheuristics for Vehicle Routing Problem, *ISTE – London*
- Ozbayhin, G., Karasan, O. & Yaman, H. (2018). New Exact Solution Approaches for the Split Delivery Vehicle Routing Problem, *Euro Journal on Computational Optimization*
- Quarteroni, A. & Saleri, F. (2006). Scientific Computing with MATLAB and Octave Second Edition, *Springer*
- Sanggala, E. & Bisma, M. A. (2023). Perbandingan *Savings Algorithm* dengan *Nearest Neighbour* dalam Menyelesaikan *Russian TSP Instances*, *Jurnal Media Teknik & Sistem Industri*
- Santosa, B., & Ai, T. J. (2017). Pengantar Metaheuristik Implementasi dengan Matlab, *ITS Tekno Sains*
- Shi, J. (2018). Particle Swarm Optimization for Split Delivery Vehicle Routing Problem, *Asia-Pacific Journal of Operational Research*
- Talbi, E. (2009). Metaheuristics from Design to Implementation, *John Wiley & Sons – USA*
- Wang, Y., & Han, Z. (2021). Ant Colony Optimization for Traveling Salesman Problem Based On Parameters Optimization, *Applied Soft Computing*