

PENYELESAIAN VRPSDP MENGGUNAKAN *FIREFLY ALGORITHM* (STUDI KASUS DISTRIBUSI AQUA GALON)

S.K. Salsabila¹, *H. Mayyani², dan P.T. Supriyo³

¹Mahasiswa S1 Program Studi Matematika, Departemen Matematika,
Fakultas Matematika dan Ilmu Pengetahuan Alam,
Institut Pertanian Bogor, Jl. Meranti, Kampus IPB Dramaga Bogor.
syifakhoirunnisa@apps.ipb.ac.id

^{2,3}Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Institut Pertanian Bogor, Jl. Meranti, Kampus IPB Dramaga Bogor.
mayyani_mat15@apps.ipb.ac.id*, praptosu@apps.ipb.ac.id

Abstrak

Penyelesaian masalah distribusi Aqua galon termasuk dalam pengaplikasian kompleks *Vehicle Routing Problem with Simultaneous Delivery and Pickup* (VRPSDP). VRPSDP bisa diselesaikan dengan metode eksak, heuristik atau meta-heuristik. *Firefly Algorithm* merupakan salah satu algoritma meta-heuristik yang terinspirasi oleh perilaku dan cara komunikasi serangga kunang-kunang dengan melalui beberapa tahapan, yaitu intensitas cahaya, tingkat daya tarik, pergerakan, dan mutasi pembalikan. Metode *Firefly Algorithm* dalam penelitian ini digunakan untuk mencari solusi optimal dalam menyelesaikan permasalahan tersebut. Hasil penelitian menunjukkan bahwa *Firefly Algorithm* dapat memberikan solusi yang mana mendekati optimal dalam waktu yang relatif lebih singkat dibandingkan dengan metode eksak.

Kata kunci: Aqua galon, distribusi, *firefly algorithm*, masalah rute kendaraan, VRPSDP

1 Pendahuluan

Distribusi dipahami sebagai kegiatan di mana barang dikirim melalui kendaraan dari produsen ke konsumen sesuai kebutuhan penggunaannya. Permasalahan rute pendistribusian tersebut dikenal dengan *Vehicle Routing Problem* (VRP). VRP pertama kali diperkenalkan oleh Dantziq dan Ramser [2]. Definisi VRP adalah mencari sebuah cara yang efisien dengan menggunakan beberapa *vehicle* yang perlu melakukan perjalanan ke beberapa tempat untuk mengantarkan dan/atau menjemput orang/barang. Penentuan pasangan kendaraan-pelanggan mempertimbangkan kapasitas kendaraan dalam satu kali angkut untuk meminimalkan biaya yang diperlukan. Penetapan biaya minimal tersebut sangat berkaitan dengan jarak minimal. VRP adalah salah satu pemrograman *integer* yang termasuk dalam kategori permasalahan *non polynomial hard* [10]. Penyelesaian VRP menggunakan metode eksak membutuhkan waktu yang relatif lama, sehingga dibutuhkan berbagai pendekatan lain untuk menyelesaikan masalah ini dengan algoritme heuristik [14].

Market leader dalam kompetisi berbagai produk air mineral di Indonesia adalah produk Aqua [9]. Proses distribusi produk Aqua dapat dikategorikan sebagai salah satu contoh masalah pendistribusian. Salah satu variasi VRP yang mempertimbangkan bahwa

pelanggan membutuhkan proses pengiriman dan pengambilan barang secara bersamaan disebut dengan *Vehicle Routing Problem with Simultaneous Delivery and Pickup* (VRPSDP) [1]. Pendistribusian produk Aqua khususnya untuk jenis kemasan ulang alik (*returnable bottle*) yaitu produk Aqua galon yang harus dikembalikan ke pabrik untuk diisi ulang kembali, dapat dimodelkan ke dalam masalah VRPSDP. Beberapa metode yang digunakan peneliti sebelumnya untuk menyelesaikan masalah VRPSDP adalah *Genetic Algorithm* [13], *Hybrid Metaheuristic Algorithm* [17], *Local Search Metaheuristic Algorithm* [18], dan *New Saving-Based Ant Algorithm* [1].

Metode yang digunakan untuk menyelesaikan permasalahan distribusi Aqua galon yaitu *Firefly Algorithm* (FA) atau Algoritma kunang-kunang. FA merupakan algoritma meta-heuristik berdasarkan pola berkedip dan perilaku kunang-kunang. Pertama kali algoritma ini dikembangkan oleh Xin-She pada tahun 2007. Beberapa masalah optimasi yang telah diselesaikan dengan FA yaitu *Unconstrained Optimization* [5], *Mixed Variable Structural Optimization* [3], *Vehicle Routing Problem* [11] dan *Travelling Salesman Problem* [7]. Dalam kasus TSP, FA menghasilkan total jarak yang paling minimal dibandingkan dengan *Genetic Algorithm* (GA), *Simulated Annealing* (SA), dan *Ant Colony Optimization* (ACO) dalam sebagian kasus yang digunakan [7].

Berdasarkan uraian yang dijelaskan sebelumnya, untuk menyelesaikan permasalahan distribusi Aqua galon, akan digunakan *Firefly Algorithm* (FA). Tujuannya untuk menentukan rute yang optimal, dengan memperhatikan kapasitas angkut kendaraan. Rute yang dihasilkan diharapkan memberikan jarak total minimum sehingga akan meminimumkan biaya distribusi.

2 Metode

Tahapan penelitian untuk menyelesaikan VRPSDP pada studi kasus pendistribusian Aqua galon menggunakan metode *Firefly Algorithm* adalah sebagai berikut:

1. Mencari dari distribusi Aqua galon pada PT. Tirta Utama Abadi melalui wawancara
2. Menerapkan *Firefly Algorithm* pada data yang telah ditentukan dengan bantuan software Matlab R2021a.
3. Mencatat dan menyimpulkan hasil

Dalam penelitian ini, galon air minum Aqua didistribusikan ke 20 pelanggan menggunakan 3 kendaraan dengan kapasitas sebanyak 144 galon. Penulis mengelompokkan setiap solusi menjadi 3 bagian, di mana setiap bagiannya merepresentasikan rute setiap kendaraan yang disesuaikan dengan hasil solusi eksak. Jarak antara pelanggan didapatkan menggunakan bantuan *Google Maps* dengan memilih jalur terpendek. Metode dieksekusi menggunakan bantuan laptop dengan spesifikasi Lenovo IdeaPad Slim 3i RAM 8GB.

3 Hasil dan Pembahasan

3.1 Vehicle Routing Problem with Simultaneous Delivery and Pickup

Vehicle Routing Problem with Simultaneous Delivery and Pickup (VRPSDP) adalah kasus permintaan gabungan di mana operasi pengiriman dan penjemputan dilakukan secara bersamaan pada satu kali pemberhentian kendaraan [4]. Menurut

Zachariadis *et al.* [17] VRPSDP bertujuan untuk menentukan biaya minimum dari beberapa rute yang memenuhi beberapa kendala, yaitu:

- a. Setiap rute dimulai dan berakhir di depot pusat.
- b. Setiap pelanggan dikunjungi tepat sekali dan pada satu rute.
- c. Setiap rute membawa sejumlah barang pengiriman (*delivery*) dari depot ke pelanggan.
- d. Setiap rute mengangkut sejumlah barang yang diambil (*pick-up*) dari pelanggan kembali ke depot sentral.
- e. Tidak ada titik pada rute mana pun di mana banyak barang yang diangkut melebihi kapasitas kendaraan.

Formulasi matematika dari permasalahan VRPSDP aqua galon dengan tujuan meminimumkan biaya yang dikeluarkan perusahaan dapat ditulis sebagai berikut:

Himpunan

- J = himpunan semua pelanggan yang akan dikunjungi
 J_0 = himpunan semua pelanggan ditambah depot

Indeks

- i, j = indeks untuk menyatakan depot dan pelanggan

Parameter

- Q = kapasitas kendaraan
 K = banyak kendaraan
 N = banyak pelanggan
 c_{ij} = biaya perjalanan dari pelanggan i ke pelanggan j
 d_j = permintaan pengiriman barang pelanggan j
 p_j = permintaan *pickup* barang pelanggan j

Variabel keputusan

- x_{ij} = $\begin{cases} 1, & \text{jika kendaraan berjalan dari pelanggan } i \text{ ke pelanggan } j \\ 0, & \text{selainnya} \end{cases}$
 y_{ij} = akumulasi banyak barang yang sudah diangkut pada busur (i, j)
 z_{ij} = akumulasi banyak barang yang akan dikirim pada busur (i, j)
 U_i = variabel tambahan untuk mengeliminasi *subtour* pada pelanggan i

Fungsi objektif

Fungsi tujuan untuk masalah pendistribusian ini yaitu untuk meminimumkan jarak tempuh.

$$\text{Min } \sum_{i \in J_0} \sum_{j \in J_0} c_{ij} x_{ij} \quad (1)$$

Kendala

1. Setiap pelanggan dikunjungi tepat satu kali dengan menggunakan satu kendaraan

$$\sum_{i \in J_0} x_{ij} = 1, \quad j \in J \quad (2)$$

$$\sum_{j \in J_0} x_{ij} = 1, \quad i \in J \quad (3)$$

2. Batas banyaknya kendaraan yang digunakan

$$\sum_{j \in J_0} x_{ij} \leq K, \quad j \in J_0 \quad (3)$$

3. Banyak barang yang diangkut dari lokasi j harus memenuhi permintaan *pickup*nya

$$\sum_{i=J_0} y_{ji} - \sum_{i=J_0} y_{ij} = p_j \quad j \in J \quad (4)$$

4. Banyak barang yang dikirim ke lokasi j harus memenuhi permintaan *delivery*nya

$$\sum_{i=J_0} z_{ij} - \sum_{i=J_0} z_{ji} = d_j, \quad j \in J \quad (5)$$

5. Beban kendaraan pada busur (i, j) tidak boleh melebihi kapasitas kendaraan

$$y_{ij} + z_{ij} \leq Qx_{ij}, \quad i, j \in J_0 \quad (6)$$

6. Eliminasi subtour

$$U_i - U_j + Nx_{ij} \leq N - 1, \quad i, j \in J, i \neq j \quad (7)$$

7. Kendala biner

$$x_{ij} \in \{0,1\} \quad i, j \in J_0 \quad (8)$$

8. Kendala ketaknegatifan

$$y_{ij} \geq 0, \quad i, j \in J_0 \quad (9)$$

$$z_{ij} \geq 0, \quad i, j \in J_0 \quad (10)$$

3.2 Firefly Algorithm (FA)

Firefly Algorithm dibentuk dan dikembangkan pada akhir tahun 2007 sampai 2008 oleh Dr. Xin She Yang di Cambridge University untuk memecahkan masalah optimasi [12]. *Firefly Algorithm* atau Algoritme kunang-kunang merupakan algoritme meta-heuristik yang didasarkan dari pola berkedipnya kunang-kunang dan perilaku kunang-kunang berkomunikasi melalui cahaya. Kunang-kunang biasanya memancarkan cahaya dalam waktu singkat dan memiliki ritme. Cahaya kunang-kunang tersebut bertujuan

untuk menarik perhatian kunang-kunang lain dan untuk mempertahankan diri dari pemangsa. Fungsi lain dari cahaya kunang-kunang adalah sebagai mekanisme peringatan bahaya. Menurut Yang [16] untuk membentuk *Firefly Algorithm*, digunakan beberapa aturan, yaitu:

1. Semua kunang-kunang bersifat *unisex*, di mana kunang-kunang tertarik antara satu dan yang lainnya tanpa melihat jenis kelamin.
2. Daya tarik kunang-kunang sebanding dengan tingkat kecerahan cahaya yang dipancarkan. Semakin besar jarak antar kunang-kunang maka tingkat kecerahannya akan berkurang. Kunang-kunang dengan cahaya yang redup akan mendekati kunang-kunang yang lebih terang. Jika keduanya tidak ada yang lebih terang maka kunang-kunang akan bergerak secara acak.
3. Tingkat kecerahan kunang-kunang akan ditentukan oleh susunan dari fungsi objektif dari masalah yang ada.

Firefly Algorithm merupakan algoritma yang awalnya dirancang untuk menyelesaikan masalah optimasi kontinu [8]. Namun, sesuai dengan beberapa penelitian sebelumnya FA dibuat menjadi diskret, seperti dalam menyelesaikan permasalahan penjadwalan *flow shop* [12], permasalahan *Travelling Salesman Problem* [6], dan permasalahan *Vehicle Routing Problem* [11]. Oleh karena itu, metode FA akan dibuat menjadi diskret terlebih dahulu untuk menyelesaikan VRPSDP. Berikut ini merupakan beberapa istilah dalam FA dan definisinya:

Intensitas Cahaya

Dalam *Firefly Algorithm*, ada dua hal yang saling berkaitan dan sangat penting, yaitu intensitas cahaya dan fungsi keatraktifan. Diasumsikan keatraktifan sebanding dengan tingkat kecerahan. Intensitas cahaya pada kunang-kunang dipengaruhi oleh fungsi tujuan atau nilai yang menunjukkan seberapa baik solusi yang diperoleh. Tingkat intensitas cahaya untuk kasus optimasi minimum dapat ditentukan oleh:

$$I_i = \frac{1}{F_i} \quad (11)$$

dengan nilai I_i merupakan tingkat intensitas cahaya pada kunang-kunang i yang berbanding terbalik dengan fungsi tujuan F_i .

Dalam penelitian Syauqi [15], kunang-kunang yang akan menerima intensitas cahaya dari kunang-kunang satu ke kunang-kunang lainnya sebanding dengan tingkat daya tarik antar kunang-kunang tersebut. Intensitas cahaya yang diterima kunang-kunang ditentukan oleh:

$$I_{ij} = \beta(r_{ij})I_j \quad (12)$$

dengan I_{ij} adalah intensitas cahaya yang masuk ke kunang-kunang i , yang berasal dari kunang-kunang j dan $\beta(r_{ij})$ adalah keatraktifan.

Jarak antar kunang-kunang

Jarak antar kunang-kunang direpresentasikan dengan jarak Hamming [6]. Jarak Hamming antara dua kunang-kunang adalah jumlah perbedaan urutan *node*/pelanggan. Berikut contoh mencari jarak antara dua kunang-kunang acak yang disusun oleh 5 node:

$$\begin{aligned}x_1 &: \{1, 2, 3, 4, 5\}, \\x_2 &: \{1, 3, 4, 2, 5\}\end{aligned}$$

Pada kedua rute tersebut, dapat dilihat bahwa urutan pelanggan pada kunang-kunang x_1 adalah 1-2-3-4-5, sedangkan urutan pelanggan pada kunang-kunang x_2 adalah 1-3-4-2-5. Terlihat bahwa urutan pertama dan kelima dari kunang-kunang x_1 dan x_2 adalah sama. Sedangkan urutan kedua, ketiga, dan keempat berbeda. Maka jarak hamming antara x_1 dan x_2 (r_{12}) adalah sama dengan jumlah rute yang berbeda, yaitu 3.

Keatraktifan

Keatraktifan atau tingkat daya tarik dilambangkan dengan lambang β . Nilai β relatif karena intensitas cahaya dilihat dan dinilai dari kunang-kunang lainnya. Oleh karena itu, hasil penelitian akan bervariasi tergantung dari jarak antar kunang-kunang yang satu dengan yang lainnya (r_{ij}). Semakin besar jarak antar kunang-kunang maka keatraktifan akan semakin kecil. Begitu pula sebaliknya, sehingga fungsi keatraktifan (β) adalah sebagai berikut [6]:

$$\beta(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2} \quad (13)$$

dengan β_0 adalah daya tarik saat tidak ada jarak antar kunang-kunang ($r = 0$), r_{ij} adalah jarak antara kunang-kunang i dan j , dan γ adalah koefisien penyerapan cahaya. parameter γ juga menentukan besarnya tingkat daya tarik. Parameter γ mencirikan variasi tingkat daya tarik kunang-kunang, dan nilainya sangat penting untuk menentukan kecepatan konvergensi dan bagaimana perilaku FA. Secara teori, $\gamma \in [0, \infty)$, tetapi dalam prakteknya γ ditentukan oleh karakteristik sistem yang akan dioptimalkan. Pada suatu kondisi di mana nilai $\gamma \rightarrow 0$, nilai tingkat daya tarik akan bernilai konstan $\beta = \beta_0$. Kondisi lainnya jika nilai $\gamma \rightarrow \infty$, ini berarti nilai tingkat daya tarik suatu kunang-kunang hampir mendekati nol.

Pergerakan kunang-kunang

Pergerakan kunang-kunang ini bertujuan untuk menghasilkan populasi kunang-kunang baru. Pergerakan kunang-kunang i yang tertarik pada kunang-kunang j dan memiliki intensitas cahaya terbesar dari dirinya ($I_{ij} > I_i$) dapat dihitung menggunakan persamaan berikut [16]:

$$x_{ij} = rand(2, r_{ij}) \quad (14)$$

dengan variabel awal x_{ij} merupakan jumlah pergerakan kunang-kunang i menuju kunang-kunang j dan r_{ij} adalah jarak antara kunang-kunang i dan kunang-kunang j .

Jika tidak ada kunang-kunang yang lebih terang dari dirinya atau intensitas cahaya kunang-kunang I_i lebih besar dari I_{ij} , maka kunang-kunang akan bergerak melangkah secara acak dengan persamaan berikut:

$$x_i = rand(2, nc) \quad (15)$$

dengan x_i merupakan jumlah langkah kunang-kunang i untuk bergerak secara acak. nc merupakan jumlah kota.

Untuk nilai langkah suatu kunang-kunang, akan diacak suatu bilangan bulat dari 2 sampai maksimum jarak antar kunang-kunang, atau sampai jumlah maksimum kota. Angka 2 merupakan pergerakan minimum yang harus dilakukan oleh kunang-kunang, karena masalah VRP membutuhkan minimal dua indeks untuk melakukan pembaruan rute.

Mutasi Pembalikan

Ketika kunang-kunang bergerak, solusi yang sudah terbentuk akan mengalami perubahan. Karena representasi kunang-kunang adalah representasi permutasi, maka digunakan mutasi pembalikan untuk merepresentasikan gerakan tersebut. Mutasi pembalikan merupakan pergerakan yang dilakukan kunang-kunang ke arah yang lebih terang maupun bergerak secara acak jika tidak ada yang lebih terang dari intensitas cahaya dirinya [16]. Tujuan mutasi pembalikan ini untuk melakukan pembaruan rute dan rute yang sudah terbentuk dapat dipertahankan. Dari sebuah rute akan dipilih dua indeks secara acak. Misalkan indeks tersebut adalah T_1 dan T_2 . Indeks T_1 dipilih secara acak, kemudian T_2 diperoleh dengan cara:

$$T_2 = (T_1 + x) \bmod nc \quad (16)$$

dengan x merupakan jumlah langkah kunang-kunang dan nc merupakan jumlah kota.

Sebenarnya, kunang-kunang dalam *Discrete Firefly Algorithm* tidak memiliki arah untuk bergerak. Oleh karena itu, setiap kunang-kunang bergerak sebanyak m kali menggunakan mutasi pembalikan. Dengan kata lain, setiap kunang-kunang akan memiliki m solusi baru. Saat kunang-kunang melalui proses mutasi pembalikan, solusi yang dihasilkan oleh kunang-kunang berubah dan akan ada $(m \times n) + 1$ kunang-kunang di akhir iterasi. Kemudian untuk iterasi berikutnya, akan dipilih n kunang-kunang terbaik berdasarkan nilai intensitas cahaya terbesar. Keadaan ini terus berlanjut hingga jumlah iterasi maksimum tercapai [6].

Dengan memilih m atau n yang besar, maka variasi solusi yang tercipta akan lebih beragam, namun membutuhkan waktu yang lebih banyak dalam proses mendapatkan solusi akhir. Penentuan nilai m dan n tidak ditentukan oleh jumlah kota ataupun jumlah iterasi [16].

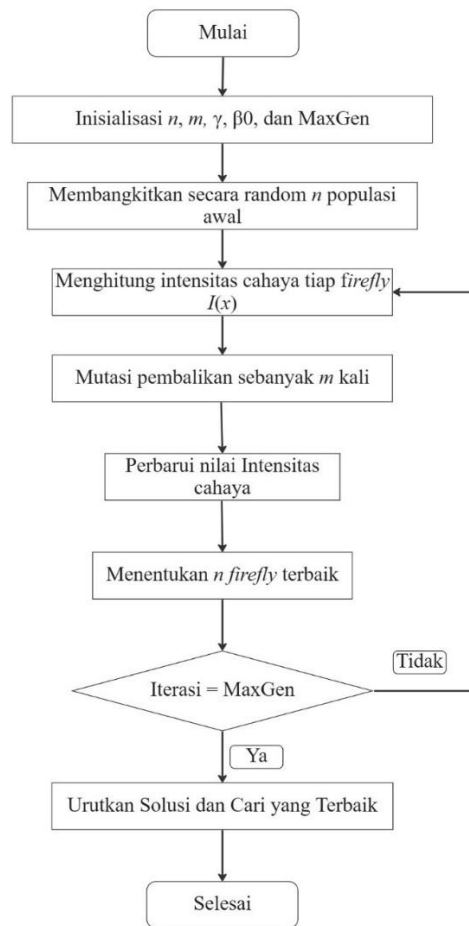
Proses *Firefly Algorithm* (FA)

Firefly Algorithm bekerja dengan cara sebagai berikut:

- a. Inisialisasi parameter FA yang meliputi n , m , γ , β_0 , dan $MaxGen$.
- b. Membangkitkan populasi awal sebanyak n *firefly* secara random.
- c. Menghitung intensitas cahaya setiap *firefly* $I(x)$ berdasarkan nilai fungsi tujuan serta membandingkan intensitas cahaya setiap *firefly* satu dan lainnya.
- d. Jika terdapat intensitas cahaya *firefly* yang terbesar, maka akan diperbarui pergerakan menuju ke salah satu *firefly* lain yang lebih terang, sedangkan *firefly* yang paling terang akan melakukan pergerakan secara random (mutasi pembalikan).

- e. Perbarui nilai intensitas cahaya *firefly* asli maupun hasil mutasi. Pada iterasi pertama, menentukan n *firefly* terbaik dengan intensitas cahaya terbesar.
- f. Melakukan proses sebelumnya sampai batas iterasi telah dipenuhi.
- g. *Firefly* dengan intensitas cahaya terbaik akan dipilih sebagai solusi akhir.

Langkah-langkah dalam *Firefly Algorithm* (FA) dapat diilustrasikan pada *flowchart* di bawah ini:



Gambar 1 *Flowchart Firefly Algorithm*

3.3 Penyelesaian VRPSDP dengan *Firefly Algorithm*

Penyelesaian menggunakan metode FA dapat dilakukan dengan memberikan input parameter yang berbeda-beda. Parameter yang digunakan adalah maxGen yang merupakan generasi maksimum dari evolusi kunang-kunang atau jumlah iterasi maksimum, n adalah jumlah populasi, m adalah jumlah mutasi, γ koefisien penyerapan cahaya, dan β_0 adalah tingkat keatraktifan saat tidak ada jarak antar kunang-kunang [20]. Untuk beberapa kondisi dilakukan percobaan dengan variasi parameter sebagai berikut:

1. Parameter maxGen berbeda-beda dan parameter n , m , γ , β_0 bernilai tetap.
2. Parameter n berbeda-beda dan parameter maxGen, m , γ , β_0 bernilai tetap.
3. Parameter m berbeda-beda dan parameter maxGen, n , γ , β_0 bernilai tetap.

4. Parameter γ berbeda-beda dan parameter maxGen, n , m , β_0 bernilai tetap.
5. Parameter β_0 berbeda-beda dan parameter maxGen, n , m , γ bernilai tetap.

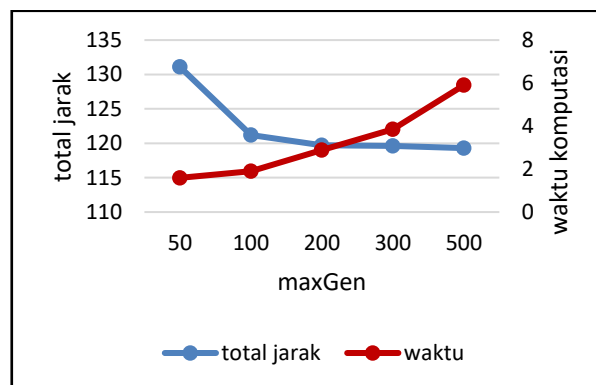
Kondisi I

Hasil dari parameter *maxGen* atau iterasi dengan berbagai nilai tercantum dalam Tabel 1.

Tabel 1 Kondisi I

maxGen	n	M	γ	β_0	total jarak	waktu
50	50	1	0.01	1	131.1	1.592713
100	50	1	0.01	1	121.2	1.904872
200	50	1	0.01	1	119.7	2.879795
300	50	1	0.01	1	119.6	3.844893
500	50	1	0.01	1	119.3	5.900967

Berdasarkan Tabel 1, semakin bertambah nilai maxGen atau iterasi, total jarak yang ditempuh semakin kecil. Grafik pergerakan total jarak dan waktu komputasi dapat dilihat pada Gambar 2.



Gambar 2 Pengaruh *maxGen* terhadap total jarak tempuh dan waktu komputasi

Berdasarkan Gambar 2, semakin besar maxGen, total jarak tempuh semakin kecil namun waktu komputasi semakin besar. Total jarak tempuh yang paling kecil bernilai 119.3 km yang didapat saat banyak iterasi 500, populasi 50, mutasi sebanyak 1, nilai gamma 0.01 dan beta nol bernilai 1.

Kondisi II

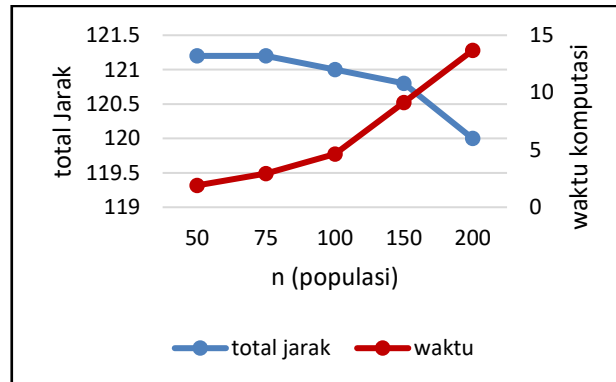
Hasil dari parameter n atau jumlah populasi kunang-kunang dengan berbagai nilai tercantum dalam Tabel 2.

Tabel 2 Kondisi II

maxGen	n	M	γ	β_0	total jarak	Waktu
100	50	1	0.01	1	121.2	1.904872

100	75	1	0.01	1	121.2	2.943905
100	100	1	0.01	1	121	4.632492
100	150	1	0.01	1	120.8	9.145625
100	200	1	0.01	1	120	13.682083

Berdasarkan Tabel 2, semakin bertambah jumlah n atau populasi awal kunang-kunang, total jarak yang ditempuh semakin kecil. Grafik pergerakan total jarak dan waktu komputasi dapat dilihat pada Gambar 3.



Gambar 3 Pengaruh jumlah populasi terhadap total jarak tempuh dan waktu komputasi

Berdasarkan Gambar 3, semakin besar jumlah populasi kunang-kunang, maka total jarak tempuh semakin kecil dan waktu komputasi semakin lambat. Total jarak tempuh yang paling kecil bernilai 120 km yang didapat saat banyak iterasi 100, populasi 200, mutasi sebanyak 1, nilai gamma 0.01 dan beta nol bernilai 1.

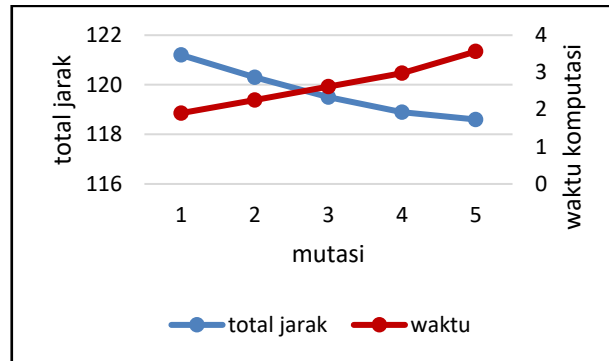
Kondisi III

Hasil dari parameter m atau jumlah mutasi kunang-kunang dengan berbagai nilai tercantum dalam Tabel 3.

Tabel 3 Kondisi III

maxGen	n	m	γ	β_0	total jarak	waktu
100	50	1	0.01	1	121.2	1.904872
100	50	2	0.01	1	120.3	2.252618
100	50	3	0.01	1	119.5	2.620186
100	50	4	0.01	1	118.9	2.977813
100	50	5	0.01	1	118.6	3.560428

Berdasarkan Tabel 3, semakin bertambah jumlah m atau mutasi dari setiap kunang-kunang, total jarak yang ditempuh semakin kecil. Grafik pergerakan total jarak dan waktu komputasi dapat dilihat pada Gambar 4.



Gambar 4 Pengaruh jumlah mutasi terhadap total jarak tempuh dan waktu komputasi

Berdasarkan Gambar 4, semakin besar jumlah mutasi, maka total jarak tempuh semakin kecil, namun waktu komputasi semakin besar. Total jarak tempuh yang paling kecil bernilai 118.6 km yang didapat saat banyak iterasi 100, populasi 50, mutasi sebanyak 5, nilai gamma 0.01 dan beta nol bernilai 1.

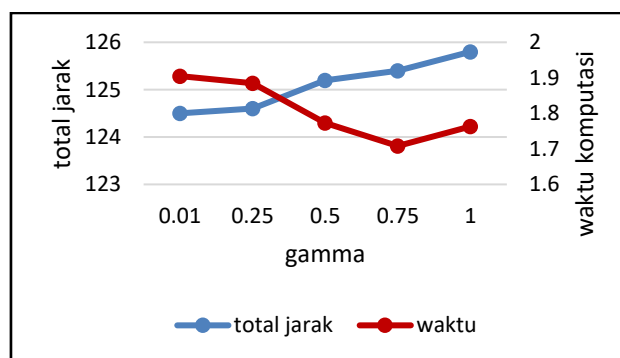
Kondisi IV

Hasil dari parameter γ atau koefisien penyerapan cahaya dengan berbagai nilai tercantum dalam Tabel 4.

Tabel 4 Kondisi IV

maxGen	n	M	γ	β_0	total jarak	waktu
100	50	1	0.01	1	124.5	1.904872
100	50	1	0.25	1	124.6	1.884337
100	50	1	0.5	1	125.2	1.773139
100	50	1	0.75	1	125.4	1.708206
100	50	1	1	1	125.8	1.762856

Berdasarkan Tabel 4, semakin besar nilai gamma, total jarak yang ditempuh semakin besar. Grafik pergerakan total jarak dan waktu komputasi dapat dilihat pada Gambar 5.



Gambar 5 Pengaruh nilai parameter γ terhadap total jarak tempuh dan waktu komputasi

Berdasarkan Gambar 5, semakin besar parameter γ , maka total jarak tempuh semakin besar dan waktu komputasi relatif menurun, namun saat gamma bernilai 1 waktu komputasi melambat. Total jarak tempuh yang paling kecil bernilai 124.5 km yang didapat saat banyak iterasi 100, populasi 50, mutasi sebanyak 1, nilai gamma 1 dan beta nol bernilai 1.

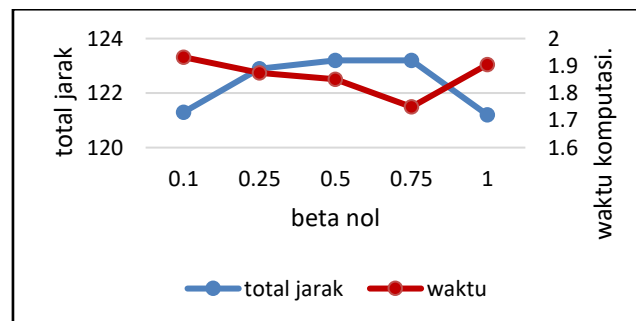
Kondisi V

Hasil dari parameter β_0 atau tingkat daya tarik saat tidak ada jarak antar kunang-kunang dengan berbagai nilai tercantum dalam Tabel 5.

Tabel 5 Kondisi V

maxgen	n	M	γ	β_0	total jarak	waktu
100	50	1	0.01	0.1	121.3	1.931730
100	50	1	0.01	0.25	122.9	1.873954
100	50	1	0.01	0.5	123.2	1.851092
100	50	1	0.01	0.75	123.2	1.749251
100	50	1	0.01	1	121.2	1.904872

Berdasarkan Tabel 5, semakin bertambahnya nilai β_0 , total jarak yang ditempuh awalnya mengalami kenaikan, lalu mengalami penurunan saat β_0 bernilai 1. Grafik pergerakan total jarak dan waktu komputasi dapat dilihat pada Gambar 6.



Gambar 62 Pengaruh nilai parameter β_0 terhadap total jarak tempuh dan waktu komputasi.

Berdasarkan Gambar 6, total jarak tempuh mengalami kenaikan, lalu mengalami penurunan dan waktu komputasi cenderung mengalami penurunan. Total jarak tempuh yang paling kecil bernilai 121.2 km yang didapat saat banyak iterasi 100, populasi 50, mutasi sebanyak 1, nilai gamma 0.01 dan beta nol bernilai 1.

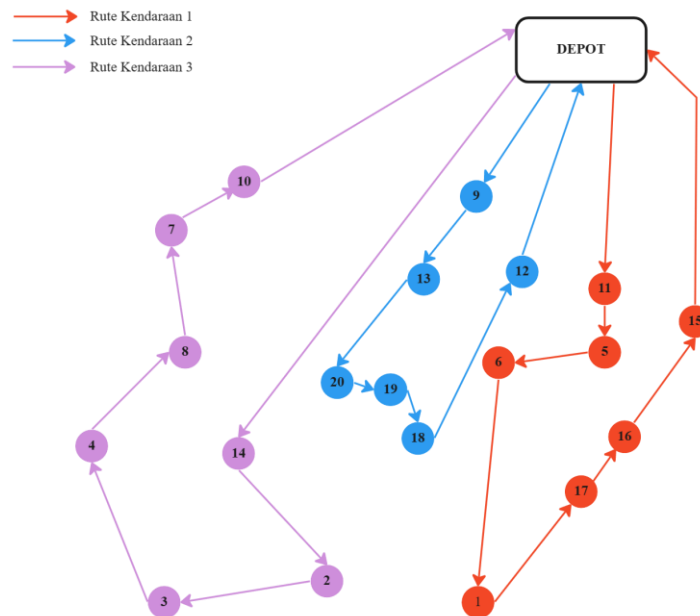
Rute terbaik

Solusi yang dihasilkan menggunakan *firefly algorithm* dalam menyelesaikan permasalahan mencari jarak tempuh terkecil yaitu 118.6 km, yang didapat saat banyak iterasi 100, populasi 50, mutasi sebanyak 5, nilai gamma 0.01 dan beta nol bernilai 1. Rute yang dihasilkan ditampilkan pada Tabel 6 dan ilustrasinya pada Gambar 7.

Tabel 6 Solusi menggunakan *Firefly Algorithm*

kendaraan ke-	Rute	permintaan	jarak tempuh
1	0→11→5→6→1→17→16→15→0	140	38.9
2	0→9→13→20→19→18→12→0	108	34.6
3	0→14→2→3→4→8→7→10→0	140	45.1
Total			118.6

Berdasarkan Tabel 6, pendistribusian Aqua Galon ke 20 *node* pelanggan dapat menggunakan 3 kendaraan. Kendaraan pertama dapat melayani 7 *node* dengan membawa 140 galon, kendaraan kedua melayani 6 *node* dengan membawa 108 galon dan kendaraan ketiga melayani 7 *node* dengan membawa 140 galon.



Gambar 7 Ilustrasi rute dengan total jarak tempuh terkecil.

Berdasarkan Gambar 7, untuk setiap *node* atau pelanggan digambarkan dengan lingkaran dan rute kendaraan digambarkan dengan panah. Dimana panah berwarna merah merupakan rute kendaraan 1, panah berwarna biru merupakan rute kendaraan 2, dan panah berwarna ungu merupakan rute kendaraan 3.

4 Simpulan dan Saran

Penyelesaian VRPSDP menggunakan *Firefly Algorithm* dipengaruhi oleh beberapa faktor, seperti ukuran populasi, jumlah iterasi, jumlah mutasi, koefisien daya tarik, dan koefisien daya serap cahaya. Oleh karena itu, perlu dilakukan penyesuaian parameter-parameter tersebut untuk mengoptimalkan performa algoritma FA dalam menyelesaikan

VRPSDP. Perbedaan input parameter yang dilakukan penulis mengakibatkan perbedaan pada solusi yang diperoleh. Semakin tinggi maxGen (iterasi), ukuran populasi, jumlah mutasi, dan koefisien daya serap cahaya, maka total jarak tempuh yang dihasilkan semakin kecil. Sedangkan pengaruh koefisien daya tarik menghasilkan solusi yang beragam.

Firefly Algorithm cukup baik dalam menyelesaikan masalah VRPSDP untuk mendapatkan solusi yang mendekati optimal dengan waktu yang relatif lebih singkat. Namun dalam beberapa data dengan jumlah pelanggan yang kecil, FA berhasil mencapai solusi optimal.

Penelitian ini dapat dikembangkan dengan menggunakan asumsi yang lebih spesifik seperti memperhitungkan kondisi jalan dan faktor kemacetan serta dengan menambahkan beberapa kendala lain seperti kendala biaya pengiriman dan biaya penalti.

Saran untuk pengembangan FA lebih lanjut adalah dengan mencoba menerapkan algoritma lain seperti *nearest neighbour* dan algoritma Greedy dalam mencari populasi awal untuk membantu FA dalam mencari solusi yang lebih optimal serta menggunakan jumlah node yang lebih besar.

Daftar Pustaka

- [1] Catay B. 2010. A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery. *Expert systems with applications*. 37(10): 6809- 6817.
- [2] Dantzig GB, Ramser JH. 1959. The truck dispatching problem. *Management Science*. 6(1):80-91. <https://doi.org/10.1287/mnsc.6.1.80>.
- [3] Gandomi AH, Yang X, Alavi AH. 2011. Mixed Variable Structural Optimization Using Firefly Algorithm. *Computer & Structures*. 89:2325-2336.
- [4] Golden BL, Assad AA. 1988. *Vehicle Routing: Methods and Studies*, Elsevier. 309-318.
- [5] Hasmi A, Goel N, Goel S, Grupta D. 2013. Firefly Algorithm for Unconstrained Optimization. *IOSR Journal of Computer Engineering*. 11:75-78.
- [6] Jati GK, Suyanto. 2011. Evolutionary Discrete Firefly Algorithm for Travelling Salesman Problem. In Proceedings of the 2nd International Conference on Adaptive and Intelligence System (ICAIS). 6943:393-403.
- [7] Kumbharana SN, Pandey GM. 2013. Solving Travelling Salesman Problem using Firefly Algorithm. *International Journal for Research in Science & Advanced Technologies*. 2: 053-057.
- [8] Lukasik S, Zak S. 2010. Algoritma Firefly untuk tugas optimasi yang dibatasi terus menerus. *International Journal for Research in Science*.
- [9] Nabilla P. 2014. Profil Umum PT. Tirta Investama Danaone AQUA. Universitas Negeri Jakarta.
- [10] Nugroho DP. 2015. Optimasi Solusi Permasalahan Rute Kendaraan dengan Pemerataan Beban Menggunakan Genetic Algorithm. Universitas Gadjah Mada.
- [11] Osaba E, Yang X, Diaz F, Onieva E, Masegosa AD, Perallos A. 2016. A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy. *Methodologies and application*.
- [12] Rizaldi E, Alkaff A, Sahal M. 2016. Penerapan *firefly algorithm* pada proses penentuan rute dan pemberangkatan kendaraan di PT PERTAMINA TBBM Surabaya Group. *Jurnal Teknik Pomits*. 1(1):1-6.
- [13] Sayadi MK, Ramezani R, Ghaffari-Nasab N. 2010. Meta-Heuristik kunang-kunang diskrit dengan pencarian lokal untuk minimisasi makepan dalam masalah penjadwalan toko aliran permutasi. *International Journal for Computer*. 110.
- [14] Serdar AT, Gen M. 2012. A genetic algorithm based approach to vehicle routing problem with simultaneous pickup and deliveries. *Computers and Industrial Engineering*.
- [15] Syakina L. 2022. Penerapan Genetic Algorithm dalam Vehicle Routing Problem dan Implementasinya pada Distribusi Rastra [skripsi]. Bogor: Institut Pertanian Bogor. Febrizal R. 2017.

- [16] Syauqi A. 2021. Optimasi Rute Travelling Salesman Problem Menggunakan Firefly Algorithm [skripsi]. Bogor: Institut Pertanian Bogor.
- [17] Yang XS. 2010. Nature-Inspire Metaheuristic Algorithm. Edisi 2. United Kingdom: Luniver Press.
- [18] Zachariadis E, Tarantilis CD, Kiranoudis CT. 2009. A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pickup service. *Expert System with Application*. 36:1070-1081.
- [19] Zachariadis E, Kiranoudis CT. 2011. A local metaheuristic algorithm for the vehicle routing problem with simultaneous pickup and deliveries. *Expert System with Application*. 38:2717-2726.
- [20] Zhang W, Jiao C, Zhou Q, Liu Y, Xu T. 2021. Gender-based deep learning firefly optimization method for test data generation. *Computational Intelligence and Neuroscience*. 2021:1-11. doi: <https://doi.org/10.1155/2021/8056225>.