

# Penerapan *Dragonfly Optimization Algorithm (DOA)* untuk Menyelesaikan *Fixed Charge Transportation Problem (FCTP)*

Ai Nuri Siti Paojiyah<sup>1)</sup>, Aliffia Putri Az'zahra<sup>1)</sup>, dan Viananda Farisqa Aulia<sup>1)</sup>, dan Elis Ratna Wulan<sup>1\*)</sup>

<sup>1)</sup>UIN Sunan Gunung Djati Bandung, Indonesia

<sup>\*)</sup>email: [elis\\_ratna\\_wulan@uinsgd.ac.id](mailto:elis_ratna_wulan@uinsgd.ac.id)

## Abstrak

Transportasi berperan penting dalam kegiatan sehari-hari terutama di dunia industri modern untuk pendistribusian barang atau jasa. Faktor yang mempengaruhi transportasi adalah biaya transportasi, salah satunya biaya tetap. Artikel ini menyelesaikan masalah transportasi biaya tetap atau Fixed Charge Transportation Problem (FCTP) dengan mengimplementasikan Dragonfly Optimization Algorithm (DOA) atau algoritma capung. Metode algoritma capung terinspirasi oleh kecerdasan atau perilaku hidup capung dalam mengetahui jalur optimal untuk mencapai tujuan yang diinginkan. Tujuan dari artikel ini adalah untuk menganalisis tahapan-tahapan pengerjaan algoritma capung berdasarkan masalah transportasi biaya tetap agar dapat meminimalkan biaya transportasi dalam pendistribusian barang. Penelitian dengan menggunakan systematic literature review menghasilkan langkah-langkah algoritma capung untuk FCTP adalah inialisasi, evaluasi kebugaran populasi awal, menentukan jumlah iterasi maksimum, distribusi solusi ke lingkungan sekitar, menentukan sumber makanan dan musuhnya, solusi terbaik di setiap lingkungan, menjalankan operasi algoritma capung, ulangi hingga jumlah iterasi maksimum terpenuhi, dan kembali ke rencana transportasi optimal.

*Kata kunci: Algoritma Capung, Biaya Tetap, Fixed Charge Transportation Problem (FCTP).*

*MSC2020: 90B06*

## Abstract

*Transportation plays a crucial role in daily activities, especially in the modern industrial world for the distribution of goods or services. Factors influencing transportation include transportation costs, one of which is fixed costs. This article addresses the Fixed Charge Transportation Problem (FCTP) by implementing the Dragonfly Optimization Algorithm (DOA), inspired by the intelligence or behavior of dragonflies in finding optimal paths to desired destinations. The goal of this article is to analyze the steps of implementing the dragonfly algorithm based on the fixed charge transportation problem to minimize transportation costs in goods distribution. Research using a systematic literature review produces the steps of the dragonfly algorithm for FCTP include initialization, evaluating the fitness of the initial population, determining the maximum number of iterations, distributing solutions to the surrounding environment, determining food sources and enemies, finding the best solution in each environment, running the dragonfly algorithm operation, repeating until the maximum number of iterations is met, and returning to the optimal transportation plan.*

*\*) Corresponding Author*

*Diterima: 20-06-2024, Disetujui: 28-10-2024, Diterbitkan: 10-11-2024*

**Keywords:** *Dragonfly Algorithm, Fixed Charge, Fixed Charge Transportation Problem (FCTP).*  
**MSC2020:** 90B06

Format sitasi: A.N.S. Paojiyah, dkk, "Penerapan *Dragonfly Optimization Algorithm* (DOA) untuk Menyelesaikan *Fixed Charge Transportation Problem* (FCTP)", *KUBIK J. Publ. Ilm. Mat.*, Vol. 9, No. 2, pp. 187-197, 2024.

## Pendahuluan

Dalam perkembangan sektor industri, transportasi menjadi salah satu aspek krusial bagi kegiatan sehari-hari perusahaan, seperti dalam pendistribusian barang maupun jasa. Di era modern saat ini, banyak ragam alternatif transportasi untuk pergerakan produk dan penyimpanan produk sementara saat pendistribusian. Pendistribusian dapat dilakukan dengan berbagai moda, seperti penerbangan, kereta api, jalan raya, dan laut[1],[2].

Faktor yang mempengaruhi transportasi adalah biaya transportasi, salah satunya biaya tetap. Biaya tetap adalah biaya yang tidak berubah dalam jangka pendek dan harus dilakukan sekalipun ketika sebuah perusahaan tidak beroperasi, seperti saat liburan. Manajemen biaya yang baik sangat diperlukan, yaitu dengan meminimalkan biaya transportasi terutama untuk biaya tetap. Karena biaya tetap merupakan masalah transportasi yang sulit diselesaikan[3],[4],[5].

Masalah transportasi biaya tetap atau *Fixed Charge Transportation Problem* (FCTP) melibatkan distribusi dari  $m$  pemasok ke sejumlah  $n$  pelanggan sehingga permintaan di setiap tujuan terpenuhi untuk mencegah adanya pasokan dari *supplier* lain. Tujuannya untuk memenuhi permintaan semua pelanggan dan meminimalkan total biaya transportasi[3]. Penelitian sebelumnya yang membahas tentang FCTP diantaranya menggunakan algoritma Siregar [6], algoritma Cartly [7], dan dengan pemrograman tak tentu [8].

Untuk mengatasi FCTP terdapat beberapa metode heuristik (penemuan baru atau solusi untuk memecahkan suatu masalah) dan metaheuristik (gabungan beberapa metode supaya tidak terjebak dalam daerah terbatas pencarian), seperti penerapan metode *Dragonfly Optimization Algorithm* (DOA) atau algoritma capung. Metode ini terinspirasi dari kehidupan capung yang unik dan unggul, seperti navigasi. Algoritma capung memperoleh hasil yang sangat komparatif untuk mengatasi FCTP dibandingkan dengan metode yang terkenal di literatur-literatur lainnya[1]. Penelitian sebelumnya yang menggunakan metode DOA adalah penelitian pada permasalahan Multiple Constraints Bounded knapsack [10], Oleh karena itu, tulisan ini membahas mengenai "Penerapan *Dragonfly Optimization Algorithm* (DOA) untuk Menyelesaikan *Fixed Charge Transportation Problem* (FCTP)".

## Metode

### **Fixed Charge Transportation Problem (FCTP)**

Masalah biaya tetap biasanya muncul dalam sistem produksi dan transportasi yang dimodelkan dengan variabel biner, yaitu 0 dan 1. Masalah ini melibatkan distribusi dari  $m$  pemasok ke sejumlah  $n$  pelanggan sehingga permintaan di setiap tujuan terpenuhi untuk mencegah adanya pasokan dari *supplier* lain. Dalam biaya tetap seimbang, jumlah total produk yang tersedia di semua pemasok harus sama dengan jumlah total produk yang dibutuhkan di semua pelanggan[9]. Tujuannya untuk menentukan jenis rute yang akan dilalui dan mengetahui ukuran pengiriman pada rute-rute tersebut, sehingga dapat meminimalkan total biaya dari pemenuhan permintaan berdasarkan kendala persediaan. Biaya tetap merupakan biaya yang tidak berubah-ubah nilainya atau tidak terpengaruh

oleh variabel lain[1],[10]. Biaya ini tidak terpengaruh oleh jarak dan waktu, sehingga jarak dekat atau jauh akan tetap sama biayanya[1]. Biaya tetap juga besarnya tidak tergantung pada jumlah produk yang dikirimkan. Secara matematis, masalah transportasi biaya tetap dapat dimodelkan sebagai berikut[2], [11],[12]:

$$P : \text{Min } Z = \sum_{i=1}^m \sum_{j=1}^n (c_{ij} x_{ij} + f_{ij} y_{ij}) \quad (1)$$

Dengan kendala

$$\sum_{j=1}^n x_{ij} = a_i, \text{ untuk } i = 1, 2, \dots, n$$

$$\sum_{i=1}^m x_{ij} = b_j, \text{ untuk } j = 1, 2, \dots, n$$

$$x_{ij} \geq 0, \text{ untuk } (i, j)$$

$$y_{ij} = \begin{cases} 0, & \text{jika } x_{ij} = 0 \\ 1, & \text{jika } x_{ij} > 0 \end{cases}$$

dengan :

$x_{ij}$  : jumlah produk yang dikirim dari  $i$  sampai  $j$

$c_{ij}$  : kapasitas persediaan yang dimiliki oleh sumber  $i$  ke tujuan  $j$

$a_i$  : kapasitas persediaan yang dimiliki oleh sumber  $i$

$b_j$  : kapasitas permintaan yang dimiliki oleh tujuan  $j$

$y_{ij}$  : *binary integer programming* untuk biaya tetap tanpa mengurangi bentuk umumnya, diasumsikan bahwa[3]

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j, \text{ dengan } a_i, b_j, c_{ij}, f_{ij} \geq 0 \quad (2)$$

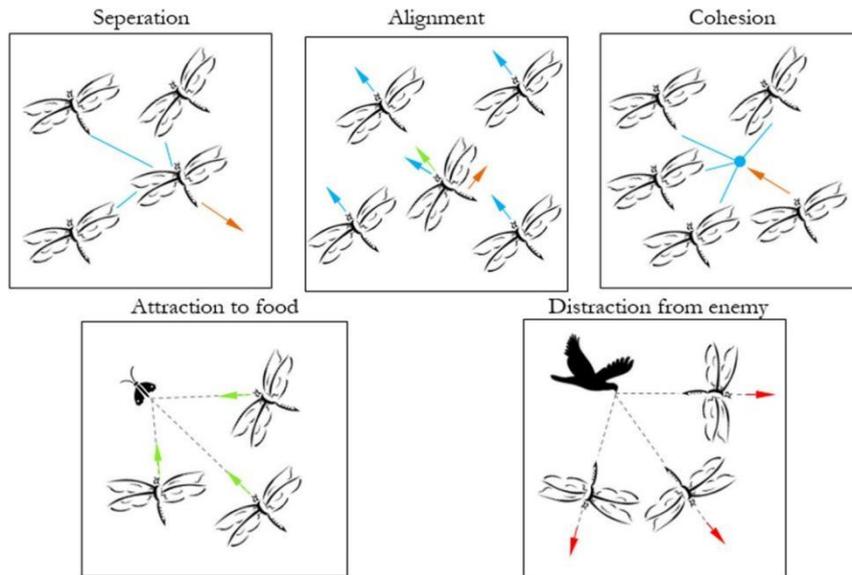
Biaya tetap yang digunakan dapat berupa biaya pendaratan di bandara, biaya jalan tol, biaya setup untuk mesin di lingkungan manufaktur, biaya pemasangan dalam sistem produksi atau biaya pembangunan jalan seperti biaya sewa kendaraan jika dibutuhkan dalam pemenuhan maupun perputaran logistik[1],[3],[13].

### **Dragonfly Optimization Algorithm (DOA)**

Metode *Dragonfly Optimization Algorithm* (DOA) atau dalam bahasa Indonesia adalah algoritma capung merupakan suatu metode penyelesaian masalah sebagai optimasi untuk pengambilan keputusan yang ditiru dari pengamatan perilaku kehidupan capung[14]. Capung terkenal memiliki kecerdasan mengetahui jalur optimal untuk mencapai tujuan yang diinginkan, karena itu kecerdasan atau perilaku hidup capung ini dapat dijadikan salah satu alasan untuk dipakainya metode algoritma capung ini.

Capung memiliki perilaku hidup statis dan dinamis yang mirip dengan dua fase utama optimasi menggunakan metaheuristik, yaitu fase eksplorasi dan fase eksploitasi. Dalam perilaku dinamis capung akan terbang di wilayah yang berbeda, sedangkan ketika statis capung akan terbang dalam kawanan yang lebih besar dengan arah yang sama[15],[16],[17].

Pada algoritma capung terdapat lima faktor penting yang dapat dimodelkan menggunakan beberapa operasi berikut ini[18]:



Gambar 1. Operasi pada Algoritma Capung

1. *Separation* atau pemisahan, operasi ini mengacu pada kehidupan capung untuk menghindari tabrakan antarcapung ketika sedang terbang[18]. Pengaplikasian operasi ini dalam masalah transportasi digunakan ketika pengiriman barang dari titik pusat ke titik tujuan dengan satu pengiriman agar tidak terjadi tabrakan. Contohnya melakukan pengiriman barang ke kota A menggunakan satu jalur, ke kota B satu jalur, dan seterusnya sehingga tidak terjadi tabrakan. Rumus untuk operasi pemisahan sebagai berikut[1]:

$$S(p, r) = - \sum_{q=1}^N X(p, r) - X(q, r) \tag{3}$$

Keterangan:

$X(p, r)$  : Posisi capung ke- $p$  pada iterasi ke- $r$

$X(q, r)$  : Posisi capung ke- $q$  pada iterasi ke- $r$

$N$  : Jumlah capung (solusi)

$S(p, r)$  : Pemisah antara capung dengan capung lainnya

2. *Alignment* atau keselarasan, operasi ini untuk menjaga kecepatan capung agar tetap sama. Pengaplikasian operasi ini dalam masalah transportasi digunakan ketika melakukan pengiriman barang antara pengiriman sebelumnya dengan selanjutnya memakan kecepatan waktu yang sama. Contohnya ketika pengiriman sebelumnya memakan kecepatan waktu dua hari, maka pengiriman selanjutnya memakan kecepatan waktu yang sama, yaitu dua hari juga. Rumus untuk operasi keselarasan sebagai berikut[1]:

$$A(p, r) = \frac{\sum_{q=1}^N V(p, r)}{N} \tag{4}$$

Keterangan:

$A(p, r)$  : Keselarasan capung ke- $p$  pada iterasi ke- $r$

$V(q, r)$  : Kecepatan capung ke- $q$  pada iterasi ke- $r$

3. Kohesi, operasi ini digunakan untuk menjaga kekompakan capung dalam menuju satu pusat yang sama agar tidak terpecah-belah. Pengaplikasian operasi ini dalam masalah transportasi

digunakan ketika melakukan pengiriman barang 1, barang 2, barang 3, dan barang lainnya dilakukan dalam satu kesatuan pengiriman yang sama. Contohnya dalam satu pengiriman melakukan pengiriman barang 1, barang 2, barang 3, dan barang lainnya lalu sampai ke titik tujuan secara bersamaan atau tidak terpisah. Rumus untuk operasi kohesi sebagai berikut[1]:

$$C(p, r) = \frac{\sum_{q=1}^N X(q, r)}{N} - X(p, r) \quad (5)$$

Keterangan:

$C(p, r)$ : Kohesi capung ke- $p$  pada iterasi ke- $r$

4. Atraksi makanan, operasi ini capung bergerak menuju sumber makanan[1]. Pengaplikasian operasi ini dalam masalah transportasi digunakan ketika melakukan pengiriman barang dari titik pusat menuju titik tujuan yang tepat. Rumus untuk operasi atraksi makanan sebagai berikut[1]:

$$F(p, r) = X^+ - X(p, r) \quad (6)$$

Keterangan:

$X^+$  : Posisi sumber makanan pada iterasi ke- $r$

$F(p, r)$  : Makanan capung ke- $p$  pada iterasi ke- $r$

5. Mengalihkan perhatian musuh, operasi ini capung mengalihkan perhatian musuh dengan cara membuat jejak palsu agar tidak dikejar musuh. Pengaplikasian operasi ini dalam masalah transportasi digunakan ketika melakukan pengiriman barang dengan membuat jalur utama dan jalur cadangan sebagai jejak palsu supaya tidak diganggu oleh saingan perusahaan. Contohnya ketika akan melakukan pengiriman barang digunakan dua jalur yang mana satu jalur utama dan jalur lainnya sebagai pengalihan jejak supaya tidak terdeteksi dari saingan. Rumus untuk operasi atraksi makanan sebagai berikut[1]:

$$E(p, r) = X^- - X(p, r) \quad (7)$$

Keterangan:

$X^-$  : Lokasi musuh pada iterasi ke- $r$

$E(p, r)$  : Gangguan musuh untuk solusi ke- $p$  pada iterasi ke- $r$

Dalam menyelesaikan masalah optimasi algoritma capung menggunakan dua vektor, yaitu vektor langkah dan vektor posisi. Vektor langkah didefinisikan sebagai berikut[1]:

$$\Delta X(p, r + 1) = sS(p, r) + aA(p, r) + cC(p, r) + fF(p, r) + eE(p, r) + w\Delta X(p, r) \quad (8)$$

Dimana  $s, a, c, f, e, w$  adalah bobot masing-masing untuk pemisahan, penyesuaian, kohesi, makanan, gangguan musuh, dan kelembaban.

Vektor posisi didefinisikan sebagai berikut[1]:

$$X(p, r + 1) = X(p, r) + \Delta X(p, r) \quad (9)$$

## Hasil dan Diskusi

Tujuan dari masalah transportasi biaya tetap adalah untuk meminimalkan biaya transportasi pendistribusian barang. Salah satu solusi dari permasalahan transportasi biaya tetap adalah menggunakan algoritma capung. Pada bagian ini membahas algoritma capung berdasarkan masalah transportasi biaya tetap. Dimulai dari lima faktor yang ada di dalam algoritma capung dapat digabungkan menjadi suatu model algoritma capung berdasarkan masalah transportasi biaya tetap.

Berikut tahapan-tahapan dalam pengerjaan algoritma capung berdasarkan masalah transportasi biaya tetap secara rinci dan sistematis, yang meliputi:

1. Inisialisasi

Langkah pertama adalah menginisialisasi populasi capung yang dapat dilakukan dengan cara acak. Populasi capung ini dibuat dari kumpulan solusi  $p$ , kemudian dibuat permutasi acak elemen-elemen yang tersedia mulai dari 1 hingga  $l = m + n$  untuk setiap solusi, dengan  $m$  sebagai pemasok dan  $n$  sebagai pelanggan[1].

2. Evaluasi kebugaran populasi awal

Pada langkah ini, setiap solusi dalam populasi awal akan dievaluasi.

3. Menentukan jumlah iterasi maksimum

Jumlah maksimum iterasi dapat berbeda-beda untuk menentukan nilai yang sesuai dengan masalah tertentu. Idealnya pada algoritma capung berdasarkan masalah transportasi biaya tetap menggunakan 300 dan 500 iterasi.

4. Distribusi solusi ke lingkungan sekitar

Untuk mendistribusikan solusi ke dalam lingkungan yang berbeda maka tentukan kesamaan diantara semua solusi yang ada. Apabila node pada indeks  $i$  dari S1 sama dengan node pada indeks  $i$  dari S2 maka bernilai 1, dan apabila tidak sama maka bernilai 0. Representasi distribusi solusi ke lingkungan sekitar sebagai berikut[1]

S1	1	6	8	2	4	3	9	7	5
S2	9	6	2	8	4	3	1	7	5
Matriks kesamaan	0	1	0	0	1	1	0	1	1

Gambar 2. Representasi Solusi

5. Tentukan sumber makanan dan musuhnya

Berdasarkan kesamaan nilai fungsi tujuan yang ada dalam populasi, maka tentukan solusi untuk sumber makanan atau solusi terbaik yang dilambangkan dengan  $X^+$  dan solusi untuk musuh atau solusi terburuk yang dilambangkan dengan  $X^-$ .

6. Tentukan solusi terbaik di setiap lingkungan

Solusi terbaik atau solusi sumber makanan dari algoritma capung adalah solusi yang nilainya paling optimal di setiap lingkungan.

7. Jalankan operasi DOA

*Operasi Pemisahan*

Ketika di lingkungan yang sama terdapat kesamaan antara dua solusi, maka pilihlah *node* secara acak kemudian tukar posisi kedua *node* pada elemen yang sama. Representasi operasi pemisahan sebagai berikut[1]



Gambar 3. Operasi Pemisahan

#### *Operasi Keselarasan*

Dari solusi terbaik, pilih dua *node* berturut-turut secara acak. Apabila kedua *node* tidak berada pada posisi yang sama maka lakukan penyeselarasan dengan menukar salah satu *node* agar bersebelahan. Representasi operasi keselarasan sebagai berikut[1]



Gambar 4. Operasi Keselarasan

#### *Operasi Kohesi*

Pada operasi kohesi mempertahankan massa lingkungan dengan cara membulatkan semua solusi yang berada dalam lingkungan yang sama.

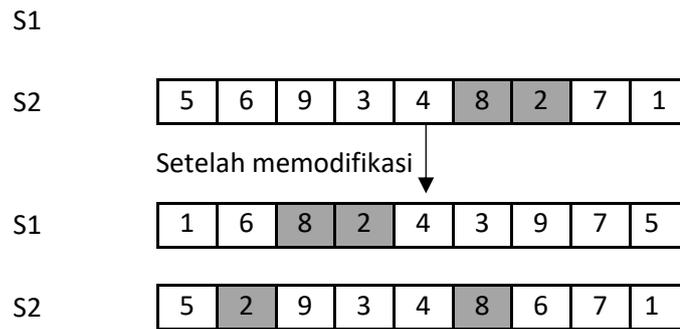
#### *Operasi Atraksi Makanan*

Pada operasi atraksi makanan dilakukan dengan mencari solusi terbaik untuk mempertahankan populasi.

#### *Operasi Mengalihkan Perhatian Musuh*

Pada operasi ini, temukan dua *node* berurutan yang ada dalam posisi yang sama. Lalu tukar *node* kedua dengan *node* lain secara acak. Representasi operasi mengalihkan perhatian musuh sebagai berikut[1]



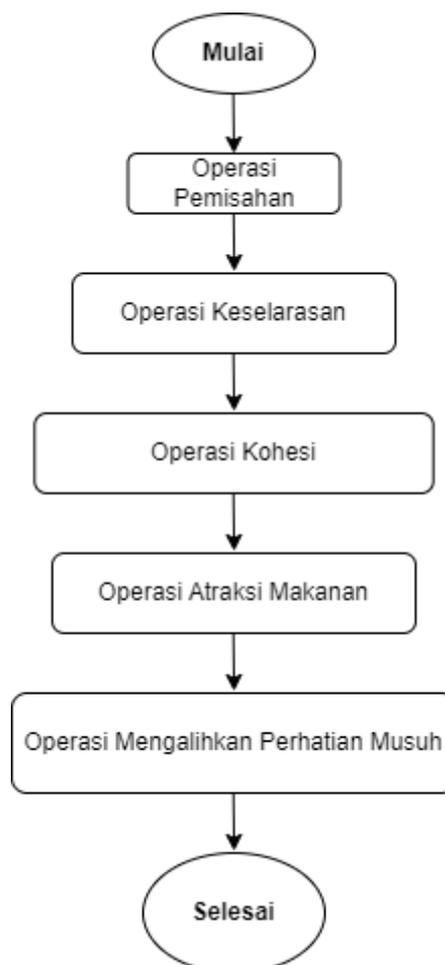


Gambar 5. Operasi Mengalihkkan Perhatian Musuh

8. Ulangi tahapan 4 sampai dengan tahapan 8, hingga menemukan jumlah iterasi maksimum

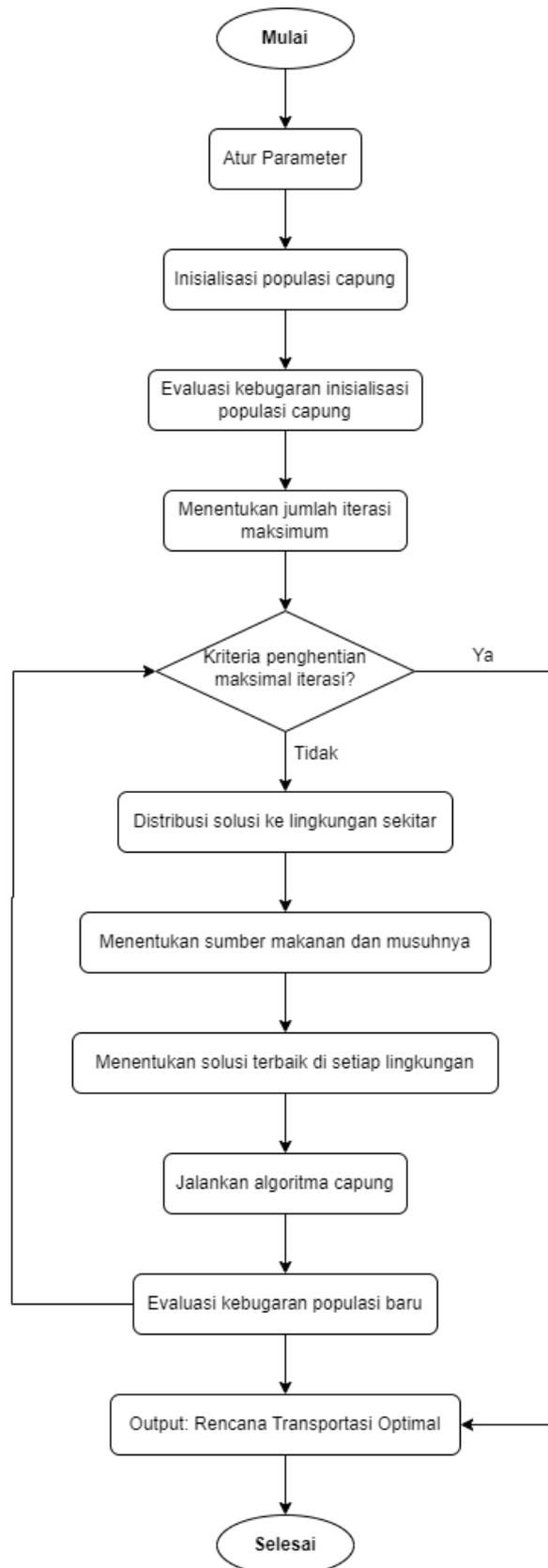
9. Kembali Rencana Transportasi Optimal

Operasi-operasi pada algoritma capung disajikan dalam bentuk diagram alir (*flowchart*) seperti gambar di bawah ini.



Gambar 6. Operasi Capung

Tahapan-tahapan dalam pengerjaan algoritma capung berdasarkan masalah transportasi biaya tetap disajikan dalam bentuk diagram alir (*flowchart*) seperti gambar di bawah ini.



Gambar 7. Algoritma Capung untuk Masalah Transportasi Biaya Tetap

## Kesimpulan

Algoritma capung menjadi salah satu solusi paling optimal dalam menyelesaikan masalah transportasi biaya tetap yang bertujuan untuk meminimalkan biaya transportasi pendistribusian barang atau jasa. Langkah-langkah algoritma capung yaitu operasi *separation* atau pemisahan, operasi *alignment* atau keselarasan, operasi kohesi, operasi atraksi makanan, dan operasi mengalihkan perhatian musuh. Sedangkan langkah-langkah algoritma capung untuk FCTP adalah inisialisasi, evaluasi kebugaran populasi awal, menentukan jumlah iterasi maksimum, distribusi solusi ke lingkungan sekitar, menentukan sumber makanan dan musuhnya, solusi terbaik di setiap lingkungan, menjalankan operasi algoritma capung, ulangi hingga jumlah iterasi maksimum terpenuhi, dan kembali ke rencana transportasi optimal.

## Referensi

- [1] I. E. Amrani, A. L. El Idrissi, A. Bahri, and A. El Allaoui, "A dragonfly algorithm for solving the Fixed Charge Transportation Problem FCTP," *Data Metadata*, vol. 3, 2024, doi: 10.56294/dm2024218.
- [2] F. Muhtarulloh, E. W. Mardiah, A. F. Huda, and D. Zulkarnaen, "Penyelesaian Masalah Transportasi dengan Degenerasi dan Siklus Berulang Menggunakan Minimum Demand Method dan Maximum Difference Extreme Difference Method," *KUBIK J. Publ. Ilm. Mat.*, vol. 8, no. 1, pp. 35–43, 2023.
- [3] T. Wahyuni, *Penerapan Metode Branching Dalam Masalah Transportasi Untuk Meminimalkan Biaya Agar Persediaan Optimal (Studi Kasus Pt. Xyz)*. 2018.
- [4] E. Safitri, S. Basriati, and H. Najmi, "Penerapan Metode Branch and Bound dalam Optimalisasi Produk Mebel (Studi kasus: Toko Mebel di Jalan Marsan, Panam)," *KUBIK J. Publ. Ilm. Mat.*, vol. 5, no. 1, pp. 43–53, 2020.
- [5] F. Hanesti, W. Syafmen, and S. Rozi, "The Optimization Problem of Batik Cloth Production with Fuzzy Multi-Objective Linear Programming and Application of Branch and Bound Method," *KUBIK J. Publ. Ilm. Mat.*, vol. 7, no. 1, pp. 19–30, 2022.
- [6] M. Q. Ramadhansyah and T. T. Dimiyati, "Penerapan Algoritma Siregar Dalam Penyelesaian Relaxed Fixed Charge Transportation Problem (RFCTP)," 2019, *Universitas Pasundan*.
- [7] N. Kartli, E. Bostanci, and M. S. Guzel, "A new algorithm for optimal solution of fixed charge transportation problem," *Kybernetika*, vol. 59, no. 1, pp. 45–63, 2023.
- [8] Y. Sheng and K. Yao, "Fixed charge transportation problem and its uncertain programming model," *Ind. Eng. Manag. Syst.*, vol. 11, no. 2, pp. 183–187, 2012.
- [9] U. Habiba, A. Quddoos, and Masihuddin, "On Solving Fixed Charge Transportation Problems Having Interval Valued Parameters," pp. 1–9, 2023.
- [10] D. L. S. Kamais, "Analisis Biaya Dan Pendapatan Usahatani Jagung Di Desa Bebuak Kecamatan Kopang Kabupaten Lombok Tengah," vol. 1, no. 1, pp. 48–57, 2023.
- [11] D. Rani, "Solving non-linear fixed-charge transportation problems using nature inspired non-linear particle swarm optimization algorithm," *Appl. Soft Comput.*, vol. 146, p. 110699, 2023.
- [12] F. Xie and R. Jia, "A heuristic algorithm for solving fixed-charge transportation problem," in *2009 WRI World Congress on Computer Science and Information Engineering*, IEEE, 2009, pp. 574–580.
- [13] S. Santoso and R. M. Heryanto, "Analisis Performansi Metode Heuristik untuk Masalah Distribusi Rantai Pasokan Dua Tahap dengan Biaya Tetap," *Performa Media Ilm. Tek. Ind.*, vol. 21, no. 2, p. 200, 2022, doi: 10.20961/performa.21.2.58274.
- [14] B. S. WIBOWO, S. HANDOKO, and H. HERMAWAN, "Optimasi Ekonomi dan Emisi Pembangkit Listrik di Kalimantan menggunakan Dragonfly Algorithm," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 9, no. 3, p. 619, 2021, doi: 10.26760/elkomika.v9i3.619.
- [15] L. Febriana Nilasari, K. Agung Santoso, and A. Riski Jurusan Matematika, "Penerapan Dragonfly

- Optimization Algorithm (DOA) Pada Permasalahan Multiple Constraints Bounded Knapsack (Studi Kasus: Kerajinan Bambu Hitam Desa Pugerbaru Kecamatan Maesan Kabupaten Bondowoso) (Application of Dragonfly Optimization Algorithm (DOA) in Multiple Constraints Bounded Knapsack Problems (Case Study: Black Bamboo Crafts Pugerbaru Village Maesan District Bondowoso Regency)),” *Maj. Ilm. Mat. dan Stat.*, vol. 19, no. 1, pp. 39–52, 2019.
- [16] B. S. Wibowo, “Penjadwalan Ekonomis dan Emisi Pembangkit Listrik Hidro dan Termal dalam Sistem 150 KV di Kalimantan Selatan dan Kalimantan Tengah menggunakan Algoritma Dragonfly,” *Electrician*, vol. 15, no. 2, pp. 100–106, 2021, doi: 10.23960/elc.v15n2.2172.
- [17] D. Hendaridi *et al.*, “Dragonfly algorithm in 2020: Review,” *Commun. Math. Biol. Neurosci.*, vol. 2021, pp. 1–21, 2021, doi: 10.28919/cmbn/5767.
- [18] L. F. Nilasari, K. A. Santoso, and A. Riski, *Penerapan Dragonfly Optimization Algorithm (Doa) Pada Permasalahan Multiple Constraints Bounded Knapsack*, vol. 19, no. 1. 2019. doi: 10.19184/mims.v19i1.17264.