

Penyelesaian *Container Stowage Problem* untuk Kontainer Ukuran 20 Feet menggunakan *Whale Optimization Algorithm*

Quinn Nathania P.J.Y¹, Asri Bekti Pratiwi^{2,*}, & . Herry Suprajitno³

Departemen Matematika, Fakultas Sains dan Teknologi, Universitas Airlangga

*Corresponding author: asri.bekti@fst.unair.ac.id

Abstrak. This paper has purpose to solve Container Stowage Problem (CSP) for 20 feet container using Whale Optimization Algorithm (WOA). CSP is a problem discussing about how to stowage a container on the ship where the purpose to minimize the unloading time. Moreover, 20 feet container is one of container types. WOA is a recently developed swarm-based metaheuristic algorithm that is based on the bubble net hunting maneuver technique of humpback whales for solving complex optimization problems. WOA had three procedures, first encircling prey, second bubble-net attacking method or exploitation phase, and third search for prey or exploration phase. WOA application program or resolving solve CSP for 20 feet container was made by using Borland C++ programming language which was implemented in three cases types of CSP data, first, the small data taking about nine containers with the number of bays, rows and tiers, respectively, are 4, 4, 4. The second and third data was medium data and big data with 62 containers and 95 containers each data, and had the number of bays, rows and tiers, respectively, are 14, 4, 5. After executing the program can be concluded the unloading time will be better if the number of whales is larger, while the number of iterations and the number of parameter control for shape of a logarithm spiral (*b*) don't affect the solution.

Keywords: Container Stowage Problem, Unloading Time, Whale Optimization Algorithm

1 Pendahuluan

Pengiriman barang menggunakan kontainer masih menjadi pilihan utama, karena dapat menjangkau jarak yang jauh dan dalam jumlah yang besar. Sebuah kapal pengangkut kontainer dapat mengangkut lebih dari 5000 kontainer dan mengunjungi 10-25 pelabuhan [1]. Ukuran kontainer ada 5 macam yaitu 20 feet dengan panjang 6.1 m, 40 feet dengan panjang 12.2 m, 45 feet dengan panjang 13.7 m, 48 feet dengan panjang 14.6 m, dan 53 feet dengan panjang 16.2 m, dengan ukuran lebar yang sama, yaitu 8 feet. Namun, kontainer yang sering digunakan pada transportasi laut adalah kontainer dengan panjang 20 feet dan 40 feet [2]. Transportasi laut di Indonesia sendiri lebih memilih menggunakan kontainer 20 feet. Kontainer 20 feet biasanya disebut dengan satu TEU (*Twenty-foot Equivalent Units*) kapasitas angkut sampai 20 ton sedangkan kontainer 40 feet yang memiliki ukuran 2 kali kontainer 20 feet hanya memiliki kapasitas angkut maksimal 27 ton. Pembiayaan dan sistem penanganan kontainer 20 feet lebih murah jika dibanding kontainer 40 feet [3].

Container Stowage Problem (CSP) merupakan permasalahan penataan kontainer diatas kapal dengan memperhatikan aturan penataan kontainer dalam kapal. CSP memperhatikan lima aturan (ukuran, tipe, berat dan tujuan kontainer, serta keseimbangan kapal). Aturan-aturan tersebut dapat dijelaskan sebagai letak kontainer harus sesuai dengan tipe dan ukuran kontainer, memperhatikan keseimbangan kapal, menempatkan kontainer yang lebih berat harus ditempatkan dibawah kontainer yang lebih ringan, dan kontainer dengan tujuan terdekat harus ditempatkan di kontainer paling atas. Sehingga CSP dapat dijelaskan sebagai aturan penataan berbagai ukuran kontainer pada sebuah kapal, sedemikian rupa sehingga dapat meminimalkan biaya operasional berdasarkan banyaknya tumpukan kontainer, beban tiap kontainer dan jarak [4]. Berbagai algoritma telah digunakan untuk menyelesaikan permasalahan pengepakan barang tiga dimensi, seperti GRASP (*Greedy Randomized Adaptive Search Procedure*) [5] dan *Particle Swarm Optimization* (PSO) [6].

Whale Optimization Algorithm (WOA) pertama kali diperkenalkan oleh S. Mirjalili dan A. Lewis pada tahun 2016. WOA adalah algoritma yang terinspirasi dari cara memburu paus bungkuk dalam mencari makanannya menggunakan jaring gelombang air. Algoritma ini terdiri kemudian diimplementasikan secara matematika menjadi dua hal, yang pertama mengitari mangsa yang terbagi menjadi dua buah gerakan yaitu mengecilkan putaran cincin dan bergerak secara spiral dan yang kedua ialah mencari mangsa [7]. WOA telah terbukti menghasilkan hasil yang lebih baik dibandingkan *Particle Swarm Optimization* (PSO) dan *Gale-Shapely Algorithm* (GSA). PSO dan GSA hanya menggunakan satu buah formula untuk memperbarui posisinya, sehingga resiko solusi hanya optimal lokal sangat tinggi. Sedangkan *Whale Optimization Algorithm* menggunakan salah satu dari dua buah formula pada tiap iterasi untuk memperbarui posisinya sehingga WOA memiliki kemampuan untuk menghindari solusi yang optimal lokal [8].

Berdasarkan uraian di atas, bab selanjutnya membahas mengenai model CSP. Langkah penyelesaian CSP menggunakan WOA pada bab 3, sedangkan bab 4 membahas mengenai hasil dan pembahasan. Terakhir, kesimpulan diberikan pada bab 5.

2 Model Container Stowage Problem

Container Stowage Problem (CSP) merupakan permasalahan penataan kontainer diatas kapal dengan memperhatikan lima aturan utama (ukuran, tipe, berat dan tujuan kontainer, serta keseimbangan kapal) dengan tujuan untuk meminimalkan waktu *unloading* [4]. Pada permasalahan CSP untuk kontainer 20 *feet* akan terdapat kontainer 20 *feet* dengan tujuan dan berat yang berbeda. Kontainer tersebut akan ditata (*stowage*) dengan memperhatikan aturan penataan untuk berat kontainer, tujuan kontainer, dan keseimbangan kapal. Berikut merupakan rumusan masalah dari *Container Stowage Problem* (CSP) untuk kontainer ukuran 20 *feet* [2].

Fungsi tujuan dari model matematis ini meminimalkan total waktu unloading (L) :

$$\text{Min } L = \sum_{l \in N} \sum_{c \in M} t_{lc} x_{lc} \quad (1)$$

dengan variabel keputusannya adalah:

x_{lc} = variable keputusan dari masalah, diasumsikan $l = 1,2,3 \dots, n$ dan $c = 1,2,3, \dots, m$.

$x_{lc} = \begin{cases} 1 & \text{jika kontainer } c \text{ ditempatkan pada lokasi } l \\ 0 & \text{jika kontainer } c \text{ tidak ditempatkan pada lokasi } l \end{cases}$

$x_{lc} = x_{ijkc}$, dengan $i = 1,2,3, \dots$, banyaknya bay, $j = 1,2,3, \dots$, banyaknya row,

$k = 1,2,3, \dots$, banyaknya tier

Batasan diberikan sebagai berikut :

1. Banyak penempatan sama dengan banyaknya kontainer.

$$\sum_{l \in N} \sum_{c \in M} x_{lc} = m \quad (2)$$

2. Setiap kontainer ditempatkan pada satu lokasi.

$$\sum_{l \in M} x_{lc} \leq 1, \forall c = 1,2,3 \dots, m \quad (3)$$

3. Setiap lokasi ditempati oleh satu kontainer.

$$\sum_{c \in M} x_{lc} \leq 1, \forall l = 1,2,3, \dots, n \quad (4)$$

4. Total berat kontainer yang diangkut tidak boleh melebihi kapasitas berat angkut kapal.

$$\sum_{l \in N} \sum_{c \in M} w_c x_{lc} \leq Q \quad (5)$$

5. Kontainer yang lebih berat tidak bisa diletakkan di atas kontainer yang lebih ringan. (Sesuai dengan aturan penataan berdasarkan berat, yaitu kontainer yang lebih berat berada di bawah kontainer yang lebih ringan).

$$\sum_{\substack{c, e \in M: \\ w_c \neq w_e}} (w_c x_{ijkc} - w_e x_{ijk1e}) \geq 0 \forall i, j, k = 1,2,3 \dots, \text{banyak tier} - 1 \quad (6)$$

6. Kontainer dengan tujuan awal diletakkan di atas kontainer dengan tujuan akhir. (Sesuai dengan aturan penataan berdasarkan tujuan, yaitu kontainer tujuan akhir diletakkan di bawah kontainer yang tujuan awal).

$$\sum_{\substack{c,e \in M: \\ d_c \neq d_e}} (d_c x_{ijkc} - d_e x_{ijk1e}) \geq 0 \quad \forall i, j, k = 1, 2, 3 \dots, \text{banyak tier} - 1 \quad (7)$$

7. Selisih total berat kontainer pada sisi *left* dengan sisi *right* tidak boleh melebihi toleransi keseimbangan *left-right* (sesuai dengan aturan penataan kontainer berdasarkan keseimbangan kapal, yaitu perbedaan berat total kontainer antara bagian sumbu kapal ke kiri dan bagian sumbu kapal ke kanan tidak melebihi batas yang ditentukan).

$$-Q_1 \leq \sum_{i,j \in L} \sum_{c,k} w_c x_{ijkc} - \sum_{i,j \in R} \sum_{c,j,k} w_c x_{ijkc} \leq Q_1 \quad (8)$$

8. Selisih total berat kontainer pada bagian *anterior* dengan bagian *posterior* tidak boleh melebihi toleransi keseimbangan *anterior-posterior* (sesuai dengan aturan penataan kontainer berdasarkan keseimbangan kapal, yaitu perbedaan berat total kontainer antara bagian sumbu kapal ke depan dan bagian sumbu kapal ke belakang tidak melebihi batas yang ditentukan).

$$-Q_2 \leq \sum_{i \in A} \sum_{c,j,k} w_c x_{ijkc} - \sum_{i \in P} \sum_{c,j,k} w_c x_{ijkc} \leq Q_2 \quad (9)$$

9. Variabel keputusan merupakan bilangan biner (0,1)

$$x_{lc} \in \{0,1\} \quad \forall l = 1, 2, 3 \dots, n \quad \forall c = 1, 2, 3, \dots, m \quad (10)$$

Tabel 1. Keterangan simbol pada model CSP

Simbol	Keterangan
t_{lc}	Waktu yang diperlukan untuk <i>unloading</i> kontainer c di lokasi l
N	Himpunan lokasi : $N = \{1, 2, 3, \dots, n\}$ dengan n adalah banyaknya lokasi
M	Himpunan kontainer : $M = \{1, 2, 3, \dots, m\}$ dengan m adalah banyaknya kontainer
Q	Kapasitas berat angkut kapal
w_c	Berat kontainer ke $-c$
w_e	Berat kontainer ke $-e$
d_c	Tujuan kontainer ke $-c$
d_e	Tujuan kontainer ke $-e$
Q_1	Toleransi keseimbangan <i>left-right</i>
Q_2	Toleransi keseimbangan <i>anterior-posterior</i>
A	Himpunan <i>bay</i> bagian depan : $A = \{1, 2, 3, \dots, \text{banyaknya bay bagian depan}\}$
P	Himpunan <i>bay</i> bagian belakang : $P = \{1, 2, 3, \dots, \text{banyaknya bay bagian belakang}\}$
L	Himpunan row bagian kiri :

	$L = \{1,2,3, \dots, \text{banyaknya row bagian kiri}\}$
R	Himpunan row bagian kanan : $R = \{1,2,3, \dots, \text{banyaknya row bagian kanan}\}$

3 Whale Optimization Algorithm (WOA)

Whale Optimization Algorithm (WOA) adalah algoritma *metaheuristik* yang meniru perilaku sosial dari ikan paus bungkuk. Algoritma ini terinspirasi dari metode berburu mangsa ikan paus tersebut. Cara berburu paus ini disebut *bubble-net foraging method* yang kemudian diimplementasikan secara matematika menjadi dua hal, yang pertama mengitari mangsa yang terbagi menjadi dua buah gerakan yaitu mengecilkan putaran cincin dan bergerak secara spiral dan yang kedua ialah mencari mangsa [7].

4 Langkah-langkah untuk Menyelesaikan CSP untuk Kontainer Ukuran 20Feet Menggunakan WOA

Pada bab ini, akan dijelaskan mengenai langkah-langkah untuk menyelesaikan CSP untuk kontainer ukuran 20 *feet* menggunakan WOA. Berikut merupakan langkah-langkahnya [8]:

- i) **Input data dan inisialisasi parameter**
Menginputkan data yang didalamnya sudah termasuk jumlah kontainer, berat kontainer, tujuan kontainer, waktu unloading, dan koordinat lokasi acak. Pada penelitian ini terdapat 3 data, yaitu data kecil, data sedang dan data besar. Inisialisasi parameter yang digunakan untuk menyelesaikan permasalahan ini seperti, banyaknya calon solusi (jumlah paus), nilai parameter b , dan maksimum iterasi.
- ii) **Pembangkitan Solusi Awal**
Membangkitkan solusi awal dengan jumlah sesuai banyaknya calon lokasi. Solusi awal yang dibangkitkan merupakan bilangan real acak dari 0 sampai 1
- iii) **Penentuan Lokasi dan Penetapan Lokasi**
Setelah dibangkitkan bilangan real acak akan diurutkan dari yang terkecil ke yang terbesar, dan hasil pengurutan akan menjadi nomor kontainer sedangkan urutan pembangkitan bilangan real acak akan menjadi lokasi bagi kontainer tersebut. Untuk nomor kontainer yang lebih besar dari jumlah kontainer, nilainya akan berubah menjadi nol (0). Selanjutnya tiap kontainer akan diletakkan pada lokasi sesuai data koordinat lokasi, jika pada saat diletakkan lokasi kontainer tidak sesuai dengan syarat penataan kontainer maka akan dilakukan pergeseran lokasi.

iv) **Evaluasi Fungsi Tujuan**

Evaluasi fungsi tujuan dilakukan dengan mentotal waktu *unloading* dan nilai penalti.

a) Menghitung Waktu *Unloading*

Waktu *unloading* untuk tiap kontainer akan ditotal sesuai dengan lokasi *tier* dan *row* tiap kontainer yang disesuaikan dengan data waktu *unloading*.

b) Menghitung Penalti

Setelah menghitung toral waktu *unloading* selanjutnya adalah memeriksa syarat penataan kontainer dan memberikan penalti seandainya terdapat kontainer yang diletakan tidak sesuai dengan aturan penataan kontainer. Terdapat empat kategori penalti, penalti 1000000 diberikan jika berat total kontainer melebihi kapasitas kapal (Q), penalti 10000 diberikan jika terdapat kontainer yang lebih berat berada diatas kontainer yang lebih ringan, penalti 100 diberikan jika selisih total berat kontainer pada bagian kiri dan kanan kapal tidak lebih dari sama dengan negatif toleransi keseimbangan *left-right* (Q_1) dan tidak kurang dari sama dengan toleransi keseimbangan *left-right* (Q_1) atau jika selisih total berat kontainer pada bagian depan dan belakang kapal tidak lebih dari sama dengan negatif toleransi keseimbangan *anterior-posterior* (Q_2) dan tidak kurang dari sama dengan toleransi keseimbangan *anterior-posterior* (Q_2), dan penalti 1 diberikan jika terdapat kontainer dengan tujuan dekat diletakan dibawah kontainer dengan tujuan jauh.

v) **Generate Parameter p**

Akan dibangkitkan satu parameter p untuk tiap calon paus, yang merupakan bilangan real acak [0,1]. Nilai parameter p tersebut yang akan menjadi nilai keputusan langkah yang diambil setelahnya, bergerak spiral atau pengecekan nilai mutlak parameter A . Jika nilai parameter p kurang dari 0.5 maka akan dilakukan pengecekan parameter A , tapi jika nilai parameter p lebih dari sama dengan 0.5 maka akan dilakukan langkah pergerakan spiral.

vi) **Cek Nilai Parameter A**

Pengecekan nilai parameter A dilakukan untuk menentukan langkah selanjutnya yaitu mengecilkan putaran cincin atau pencarian mangsa baru. Nilai parameter A didapatkan dengan menggunakan persamaan (11) pada rumus sebagai berikut :

$$A_i^t = 2 \cdot a_i^t \cdot r_i^t - a_i^t \quad (11)$$

$$a_i^t = 2 - (t * (2 \div \text{Maks iterasi})) \quad (12)$$

Dengan :

- a : Parameter yang bernilai 2 dan nilainya akan turun seiring berjalannya iterasi, sesuai persamaan (12) [9].
 r : Parameter bilangan real yang dibangkitkan secara acak pada interval 0 dan 1
 i : $i = \{1,2, \dots, \text{banyaknya paus yang dibangkitkan}\}$
 t : $t = \{1,2, \dots, \text{maksimum iterasi}\}$

vii) Pengecilan Putaran Cicin

Jika tidak terdapat nilai mutlak parameter A yang bernilai kurang dari sama dengan satu maka akan dilakukan langkah pengecilan putaran cincin sesuai persamaan (13) dan persamaan (14) pada rumus sebagai berikut :

$$D_i^t = |C_i^t \cdot X_i^{*t} - X_i^t| \quad (13)$$

$$X_i^{t+1} = X_i^{*t} - A_i^t \cdot D_i^t \quad (14)$$

$$C_i^t = 2 \cdot r_i^t \quad (15)$$

Dengan :

- D : Parameter yang menyatakan jarak antara mangsa dengan seekor paus yang dipengaruhi oleh parameter C .
 C : Sebuah parameter yan didapatkan melalui perhitungan sesuai persamaan (15)
 X^* : Paus terbaik
 X : Paus
 i : $i = \{1,2, \dots, \text{banyaknya paus yang dibangkitkan}\}$
 t : $t = \{1,2, \dots, \text{maksimum iterasi}\}$
 r : Parameter bilangan real yang dibangkitkan secara acak pada interval 0 dan 1

viii) Bergerak Spiral

Akan dibangkitkan parameter l , yang merupakan bilang real [-1,1]. Selanjutnya adalah menentukan nilai parameter D , yang menyatakan jarak antara mangsa dengan seekor paus, menggunakan persamaan (16) pada rumus sebagai berikut :

$$D_i^t = |X_i^{*t} - X_i^t| \quad (16)$$

Kemudian dilakukan perhitungan untuk paus iterasi selanjutnya (X_i^{t+1}), menggunakan persamaan (17) pada rumus sebagai berikut :

$$X_i^{t+1} = D_i^t \cdot \exp^{bl} \cdot \cos(2\pi l) + X_i^{*t} \quad (17)$$

Dengan :

- D : Parameter yang menyatakan jarak antara mangsa dengan seekor paus.
 X^* : Paus terbaik
 X : Paus
 i : $i = \{1,2, \dots, \text{banyaknya paus yang dibangkitkan}\}$
 t : $t = \{1,2, \dots, \text{maksimum iterasi}\}$
 b : Sebuah konstanta yang menentukan bentuk dari logaritma spiral yang merupakan bilangan rasional non negatif.
 l : Sebuah bilangan real yang dibangkitkan secara acak pada interval -1 dan 1.

ix) Pencarian Mangsa Baru

Jika terdapat nilai mutlak parameter A yang bernilai kurang dari sama dengan satu maka akan dilakukan langkah pencarian mangsa baru sesuai persamaan (18) dan persamaan (19) pada rumus sebagai berikut :

$$D_i^t = |C_i^t \cdot X_{rand_i}^t - X_i^{t+1}| \quad (18)$$

$$X_i^{t+1} = X_{rand_i}^t - A_i^t \cdot D_i^t \quad (19)$$

Dengan :

- D : Parameter yang menyatakan jarak antara mangsa dengan seekor paus yang dipengaruhi oleh parameter C .
 C : Sebuah parameter yan didapatkan melalui perhitungan sesuai persamaan (15)
 X_{rand} : Sebuah mangsa baru yang dibangkitkan secara acak, berupa bilangan real $[0,1]$.
 X : Paus
 i : $i = \{1,2, \dots, \text{banyaknya paus yang dibangkitkan}\}$
 t : $t = \{1,2, \dots, \text{maksimum iterasi}\}$

x) Update Solusi

Akan dibandingkan solusi terbaik pada pada tiap iterasi, dan nilai fungsi tujuan yang paling minimum yang digunakan sebagai paus terbaik untuk iterasi selanjutnya.

xi) Mengecek maksimum iterasi

Pada langkah ini, dilakukan pengecekan maksimum iterasi. Apabila maksimum iterasi sudah tercapai, maka proses berhenti dan didapatkan nilai solusi terbaik (waktu *unloading*) dan lokasi terbaik untuk tiap kontainer. Jika

tidak maka proses akan terus berjalan menggunakan nilai solusi terbaik sebagai paus terbaik.

5 Hasil dan Pembahasan

Program untuk menyelesaikan *Container Stowage Problem* (CSP) untuk kontainer ukuran 20 feet menggunakan *Whale Optimization Algorithm* (WOA) dilakukan pada implementasi dalam beberapa data. Program yang digunakan adalah bahasa pemrograman C++ dengan menggunakan *software* Borland C++ 5.02. Data yang digunakan diperoleh dari [11]. Ada 3 kategori data pada penelitian ini, untuk kriteria data dapat dilihat pada **Tabel 2**.

Tabel 2. Kriteria Data

Kasus Data	Jumlah Kontainer	Berat			Tujuan			Ukuran Kapal		
		Light (10-15)	Medium (20-25)	Heavy (30-35)	1	2	3	Bay	Row	Tier
Kecil	9	9	0	0	5	4	0	4	4	4
Sedang	62	26	19	17	32	30	0	14	4	5
Besar	95	39	33	23	31	27	37	14	4	5

Dengan batasan parameter (Q) yang digunakan untuk data kecil, sedang, dan besar secara berturut-turut adalah 250, 2050, dan 2050 satuan berat, kemudian untuk parameter keseimbangan kapal (Q_1) yang digunakan untuk data kecil, sedang, dan besar secara berturut-turut adalah 20, 350, dan 450 satuan berat, dan untuk parameter keseimbangan kapal (Q_2) yang digunakan untuk data kecil, sedang, dan besar secara berturut-turut adalah 40, 500, dan 850 satuan berat. Hasil selengkapnya disajikan pada **Tabel 3**, **Tabel 4**, dan **Tabel 5**.

Tabel 3. Hasil Eksekusi Program Data Kecil

Parameter	Jumlah Paus	Jumlah Iterasi				
		10	50	100	500	
$b = 0.1$	10	1273	1273	1273	1273	
	50	1272	1272	1272	1272	
	100	1249	1249	1249	1249	
	$b = 1$	10	1273	1273	1273	1273
		50	1272	1272	1272	1272
		100	1249	1249	1249	1249
P	10	1273	1273	1273	1273	

	$b = 10$	50	1272	1272	1272	1272
		100	1249	1249	1249	1249

Tabel 4. Hasil Eksekusi Program Data Sedang

		Jumlah Paus	Jumlah Iterasi			
			10	50	100	500
Parameter	$b = 0.1$	10	19143	19143	19143	19143
		50	19041	19041	19041	19041
		100	9323	9323	9323	9323
	$b = 1$	10	19155	19155	19155	19155
		50	19047	18957	18945	18945
		100	9323	9323	9323	9323
	$b = 10$	10	19179	19179	19179	19101
		50	19143	19113	19107	19089
		100	9323	9323	9323	9323

Tabel 5. Hasil Eksekusi Program Data Besar

		Jumlah Paus	Jumlah Iterasi			
			10	50	100	500
Parameter	$b = 0.1$	10	23937	23867	23855	23081
		500	14087	14087	14087	14087
		1000	13991	13991	13991	13991
	$b = 1$	10	23897	23795	23795	23795
		500	14075	14075	14075	14075
		1000	13991	13991	13991	13991
	$b = 10$	10	23937	23795	23783	23777
		500	14087	14087	14087	14087
		1000	13991	13991	13991	13991

Berdasarkan implementasi program pada ketiga data tersebut untuk menyelesaikan *Container Stowage Problem* (CSP) untuk kontainer ukuran 20 feet menggunakan *Whale Optimization Algorithm* (WOA), dapat dilihat jika jumlah paus sangat berpengaruh untuk mendapat nilai solusi yang baik, sedangkan jumlah maksimum iterasi tidak memberikan pengaruh banyak pada hasil solusi yang lebih baik, dapat dilihat juga nilai solusi yang paling baik jika menggunakan nilai parameter b sebesar satu (1), tetapi nilai parameter b juga sangat berpengaruh pada nilai solusi saat jumlah iterasi semakin banyak. Oleh karena itu dapat disimpulkan jika jumlah paus yang lebih banyak berpengaruh pada nilai solusi yang semakin baik, sedangkan maksimum iterasi dan nilai parameter b tidak memberikan pengaruh.

6 Kesimpulan

Container Stowage Problem (CSP) untuk kontainer ukuran 20 feet adalah masalah penataan kontainer diatas kapal dengan memerhatikan aturan penataan berat kontainer, tujuan kontainer, dan keseimbangan kapal khususnya untuk kontainer ukuran 20 feet dengan tujuan untuk mendapatkan waktu *unloading* minimal dan lokasi penempatan kontainer terbaik. *Whale Optimization Algorithm* (WOA) merupakan algoritma yang digunakan untuk menyelesaikan *Container Stowage Problem* (CSP) untuk kontainer ukuran 20 feet. Berdasarkan hasil yang diperoleh menggunakan program C++ menunjukkan bahwa WOA memberikan solusi yang lebih baik ketika jumlah paus lebih banyak. Sedangkan, maksimum iterasi dan nilai parameter b tidak berpengaruh dalam menghasilkan fungsi tujuan yang lebih baik.

Referensi

- [1] Dubrovsky dkk. 2002. A Genetic Algorithm with a Compact Solution Encoding for the Container Ship Stowage Problem. *Journal of Heuristics*, November. **Volume VIII**. 585–599.
- [2] Ambrosino dkk. 2004. Stowing A Containership: The Master Bay Plan Problem. *Transportation Research Part A*. **38**. 81-99.
- [3] Wang dkk. 2017. Optimal Container Routing in Liner Shipping Networks Considering Repacking 20 ft Containers into 40 ft Containers. *Journal of Advanced Transportation*. **Volume 2017**. 1-9
- [4] Martins dkk. 2009. Container Stowage Problem Solution for Short Sea Shipping, *IO 2009 – 14^o Congresso da APDIO*. **7 a 9 de Setembro de 2009**. 1-8
- [5] Parreño-Torres dkk. 2019. Solution Strategies for a Multiport Container Ship Stowage Problem. *Mathematical Problems in Engineering*. **Volume 2019**. 1-12.
- [6] Matsaini dan Santosa, B. 2017. Solving the Container Stowage Problem (CSP) using Particle Swarm Optimization (PSO). *International Conference on Industrial and System Engineering (IConISE)*. **337**. 1-6.

- [7] H Mohammed dkk. 2019. A Systematic and Meta-Analysis Survey of Whale Optimization Algorithm. *Computational Intelligence and Neuroscience*. **Volume 2019**. 1-25.
- [8] Mirjalili, S. dan Lewis, A. 2016. The Whale Optimization Algorithm. *Advances in Engineering Software*. **Volume 95** . 51-67.
- [9] Jjang dkk. 2020. Improving the Performance of Whale Optimization Algorithm through OpenCL-Based FPGA Accelerator. **Volume 2020. Article ID 8810759**. 3
- [11] Matsaini dan Santosa, B. 2018. Solving the Container Stawage Problem (CSP) using Particle Swarm Optimization (PSO). *IOP Conf. Ser : Mater. Sci. Eng.* **337** 012002.