



**PENERAPAN ALGORITMA PENGUINS SEARCH OPTIMIZATION (PeSAO)  
DAN ALGORITMA MIGRATING BIRD OPTIMIZATION (MBO) PADA  
PERMASALAHAN KNAPSACK 0-1**

**SKRIPSI**

Oleh :  
**Rinaldy Ahmad Abdullah**  
**141810101030**

**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS JEMBER  
2019**



**PENERAPAN ALGORITMA PENGUINS SEARCH OPTIMIZATION  
(PeSAO) DAN ALGORITMA MIGRATING BIRD OPTIMIZATION  
(MBO) PADA PERMASALAHAN KNAPSACK 0-1**

**SKRIPSI**

Oleh :  
**Rinaldy Ahmad Abdullah**  
**141810101030**

**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS JEMBER  
2019**

## PERSEMBAHAN

Puji syukur alhamdulillah saya ucapkan kehadirat Allah SWT atas segala rahmat, hidayah, inayah dan karunia-Nya kepada saya sehingga dapat menyelesaikan skripsi ini dengan lancar. Shalawat serta salam selalu tercurahkan kepada junjungan kita Nabi Muhammad SAW yang telah membawa umat dari zaman kegelapan menuju islam. Karya yang sederhana ini saya persembahkan kepada:

1. Kedua orang tua, Ayahanda Suprihono, Ibunda Suswani yang selalu mendoakan, memberi semangat untuk kesuksesanku;
2. Kedua kakakku Ihwal Praja dan Arif Harimukti serta Adikku Luhur Surya Gumilang yang selalu memberi dukungan yang terbaik;
3. Pengasuh PPI.Ash Shiddiqi Putera Jember, K.H.Balya Firjaun Barlaman yang selalu memberikan banyak nasehat dan bimbingan agama selama di jember;
4. Keluarga besar PPI.Ash Shiddiqi Putera Jember yang memberikan warna warni disetiap harinya;
5. Keluarga besar “Extreme 2014” dan “Kuda Perjaka”, terima kasih atas segala bantuan, semangat dan motivasinya;
6. Guru-guruku sejak TK sampai SMA, dosen-dosenku yang telah memberikan ilmu, membimbing dengan kesabaran dan keikhlasan hati serta Almamater Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;

**MOTTO**

“Tidak ada yang tidak bisa, tapi kita mau atau tidak”

(K.H.Balya Firjaun Barlaman)

“Jangan Tinggalkan Sholat 5 Waktu Berjama’ah, Setiap Hari Bacalah Al-qur’an dan Sholawat, Jangan Berbuat Dzolim”

(K.H.Achmad Shiddiq)

**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Rinaldy Ahmad Abdullah

NIM : 141810101030

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul **“Penerapan Algoritma Penguins Search Optimization (PeSOA) dan Algoritma Migrating Birds Optimization (MBO) pada Permasalahan Knapsack 0-1”** adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah disebutkan sumbernya, dan belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenar-benarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar

Jember, Januari 2019

Yang menyatakan,

Rinaldy Ahmad Abdullah

NIM 141810101030

**SKRIPSI**

**PENERAPAN ALGORITMA PENGUINS SEARCH OPTIMIZATION  
(PeSOA) DAN ALGORITMA *MIGRATING BIRDS OPTIMIZATION* (MBO)  
PADA PERMASALAHAN *KNAPSACK 0-1***

Oleh

Rinaldy Ahmad Abdullah

NIM 141810101030

Pembimbing

Dosen Pembimbing I : Abduh Riski, S.Si., M.Si

Dosen Pembimbing II : Ahmad Kamsyakawuni, S.Si., M.Kom

**PENGESAHAN**

Skripsi berjudul “Penerapan Algoritma Penguins Search Optimization (PeSOA) dan Algoritma *Migrating Birds Optimization* (MBO) pada Permasalahan *Knapsack 0-1*” karya Rinaldy Ahmad Abdullah telah diuji dan disahkan pada:

Hari, tanggal :

Tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Jember

Tim Penguji :

Ketua,

Sekretaris,

Abduh Riski, S.Si., M.Si

Ahmad Kamsyakawuni, S.Si., M.Kom

NIP. 19900406 201504 1 001

NIP. 19721129 199802 1 001

Anggota II,

Anggota III,

Kiswara Agung Santoso, S.Si., M.Kom

Ikhsanul Halikin, S.Si., M.Si

NIP. 197209071998031003

NIP. 198610142014041001

Mengesahkan

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Jember

Drs. Sujito, Ph.D

NIP. 196102041987111001



## RINGKASAN

### **Penerapan Algoritma Penguins Search Optimization dan Algoritma Migrating Birds Optimization (MBO) pada Permasalahan Knapsack 0-1;**

Rinaldy Ahmad Abdullah, 141810101030; 2019; 57 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu pengetahuan Alam Universitas Jember

Setiap orang pasti menginginkan keuntungan yang maksimal dengan sumber atau modal seminimal mungkin. Salah satu contoh dalam kehidupan sehari-hari adalah persoalan media penyimpanan yang terbatas tetapi diharuskan mendapat keuntungan yang semaksimal mungkin. Dari permasalahan tersebut munculah istilah yang dikenal dengan masalah *knapsack*. Masalah *knapsack* merupakan suatu permasalahan bagaimana memilih objek dari sekian banyak objek dengan batasan objek tersebut sama atau lebih kecil dari kapasitas media penyimpanan sehingga diperoleh hasil yang optimal. Permasalahan *knapsack* dibagi menjadi tiga jenis diantaranya adalah permasalahan *knapsack* 0-1, permasalahan *bounded knapsack*, permasalahan *unbounded knapsack*. Pembagian tersebut didasarkan atas pola penyimpanan barang dengan nilai dan bobot yang bervariasi.

Penelitian ini akan menyelesaikan permasalahan *knapsack* 0-1 menggunakan algoritma Penguins Search Optimization (PeSOA) dan algoritma *Migrating Birds Optimization* (MBO) dengan tujuan mengetahui hasil optimal dan waktu komputasi dari penerapan algoritma tersebut. Data yang digunakan pada penelitian kali ini adalah data dari toko pertanian UD Alam Subur. Penyelesaian dibantu *software* MATLAB R2015b dan dijalankan pada Laptop Acer Aspire dengan spesifikasi Intel® Core™ i5-3210M CPU @2.50Hz, RAM 8 GB dan 64-bit OS.

Hasil perbandingan antara algoritma *Penguins Search Optimization* (PeSOA) dengan algoritma *Migrating Birds Optimization* (MBO) pada permasalahan *knapsack* 0-1 menunjukkan bahwa kedua algoritma menghasilkan solusi yang



sama-sama optimal karena karena dari 10 kali *running* program disetiap parameter menghasilkan nilai yang sama. Profit dan berat barang yang diangkut menunjukkan hasil yaitu dengan profit sebesar Rp 10.364.000 dengan total berat barang yang diangkut sebanyak 497 kg.

Berdasarkan hasil penelitian, diawal iterasi algoritma PeSOA lebih cepat menuju konvergen namun ketika mendekati solusi optimal algoritma PeSOA akan melambat ketika menuju solusi optimal hal tersebut dikarenakan perubahan nilai yang kecil pada solusi baru dengan solusi lama. Secara umum penggunaan parameter pada masing-masing algoritma memiliki pengaruh terhadap hasil (keuntungan maksimum), iterasi konvergen dan *running time* jika dilihat dari hasil rata-rata 10 kali *running*.

Hasil tersebut menunjukkan bahwa algoritma PeSOA dan algoritma MBO semakin besar jumlah populasi yang digunakan maka hasil yang didapat adalah lebih cepat konvergen tapi pada algoritma PeSOA membutuhkan iterasi yang lebih banyak dibanding dengan algoritma MBO. Hal ini terjadi karena representasi algoritma MBO langsung berupa biner sedangkan algoritma PeSOA berupa riil terlebih dahulu kemudian dikonversikan menjadi biner dan juga dimungkinkan hasil konversi posisi baru sama dengan posisi lama sehingga membutuhkan waktu yang lebih lama untuk mencapai nilai maksimum. Selain itu berdasarkan *running time* algoritma MBO membutuhkan waktu komputasi yang sedikit lebih cepat dibandingkan dengan algoritma PeSOA. Dengan demikian dapat dikatakan algoritma MBO lebih baik daripada algoritma PeSOA untuk menyelesaikan permasalahan *knapsack* 0-1

## PRAKATA

Puji syukur penulis ucapkan kehadiran Allah SWT atas segala limpahan berkah, rahmat serta hidayah-Nya, sehingga penulisan skripsi yang berjudul “Penerapan Algoritma Penguins Search Optimization (PeSOA) dan Algoritma *Migrating Birds Optimization* (MBO) pada Permasalahan *Knapsack 0-1*” dapat terselesaikan. Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan strata satu (S1) di Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Jember.

Penulis menyadari bahwa dalam proses penulisan skripsi ini banyak menerima bantuan, bimbingan, serta motivasi dari berbagai pihak. Oleh karena itu penulis menyampaikan ucapan terimakasih yang tiada terhingga kepada:

1. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, Drs. Sujito, Ph.D dan Ketua Jurusan Matematika, Kusbudiono, S.Si., M.Si yang telah membimbing dengan kesabaran dan keikhlasan hati;
2. Dosen Pembimbing Utama, Abduh Riski, S.Si., M.Si dan Dosen Pembimbing Ahmad Kamsyakawuni, S.Si., M.Kom yang telah memberikan ilmu, membimbing dengan kesabaran dan keikhlasan hati;
3. Dosen Penguji Utama, Kiswara Agung Santoso, S.Si., M.Kom. dan Dosen Penguji Anggota, Ikhsanul Halikin, S.Si., M.Si. yang telah memberikan ilmu, membimbing dengan kesabaran dan keikhlasan hati;

Penulis menyadari bahwa penulisan ini masih jauh dari sempurna. Oleh karena itu, penulis mengharap kritik dan saran demi kesempurnaan penelitian selanjutnya. Semoga bantuan, bimbingan serta motivasi beliau dicatat sebagai amal baik oleh Allah SWT. Harapan terakhir, semoga skripsi ini bermanfaat bagi kemajuan ilmu pengetahuan khususnya di bidang matematika. Aamiin.

Jember, Januari 2019

Penulis

DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL</b> .....	i
<b>HALAMAN PERSEMBAHAN</b> .....	ii
<b>HALAMAN PERNYATAAN</b> .....	iv
<b>HALAMAN PEMBIMBING</b> .....	v
<b>HALAMAN PENGESAHAN</b> .....	vi
<b>RINGKASAN</b> .....	vii
<b>PRAKATA</b> .....	ix
<b>DAFTAR ISI</b> .....	x
<b>DAFTAR TABEL</b> .....	xii
<b>DAFTAR GAMBAR</b> .....	xiii
<b>BAB 1. PENDAHULUAN</b> .....	1
<b>1.1 Latar Belakang</b> .....	1
<b>1.2 Rumusan Masalah</b> .....	2
<b>1.3 Batasan Masalah</b> .....	3
<b>1.4 Tujuan</b> .....	3
<b>1.5 Manfaat</b> .....	3
<b>BAB 2. TINJAUAN PUSTAKA</b> .....	4
<b>2.1 Optimasi</b> .....	4
<b>2.2 Permasalahan <i>Knapsack 0-1</i></b> .....	4
<b>2.3 Algoritma <i>Penguins Search Optimization (PeSOA)</i></b> .....	6
<b>2.4 implementasi algoritma <i>Penguins Search Optimization (PeSOA)</i></b> .....	7
2.4.1 Inisialisasi Populasi Awal.....	7
2.4.2 Perhitungan Setiap Penguins .....	7
2.4.3 Update Solusi dan Kriteria Pemberhentian .....	8
<b>2.5 Algoritma <i>Migrating Bird Optimization (MBO)</i></b> .....	8
<b>2.6 Implementasi algoritma <i>Migrating Bird Optimization</i></b> .....	9

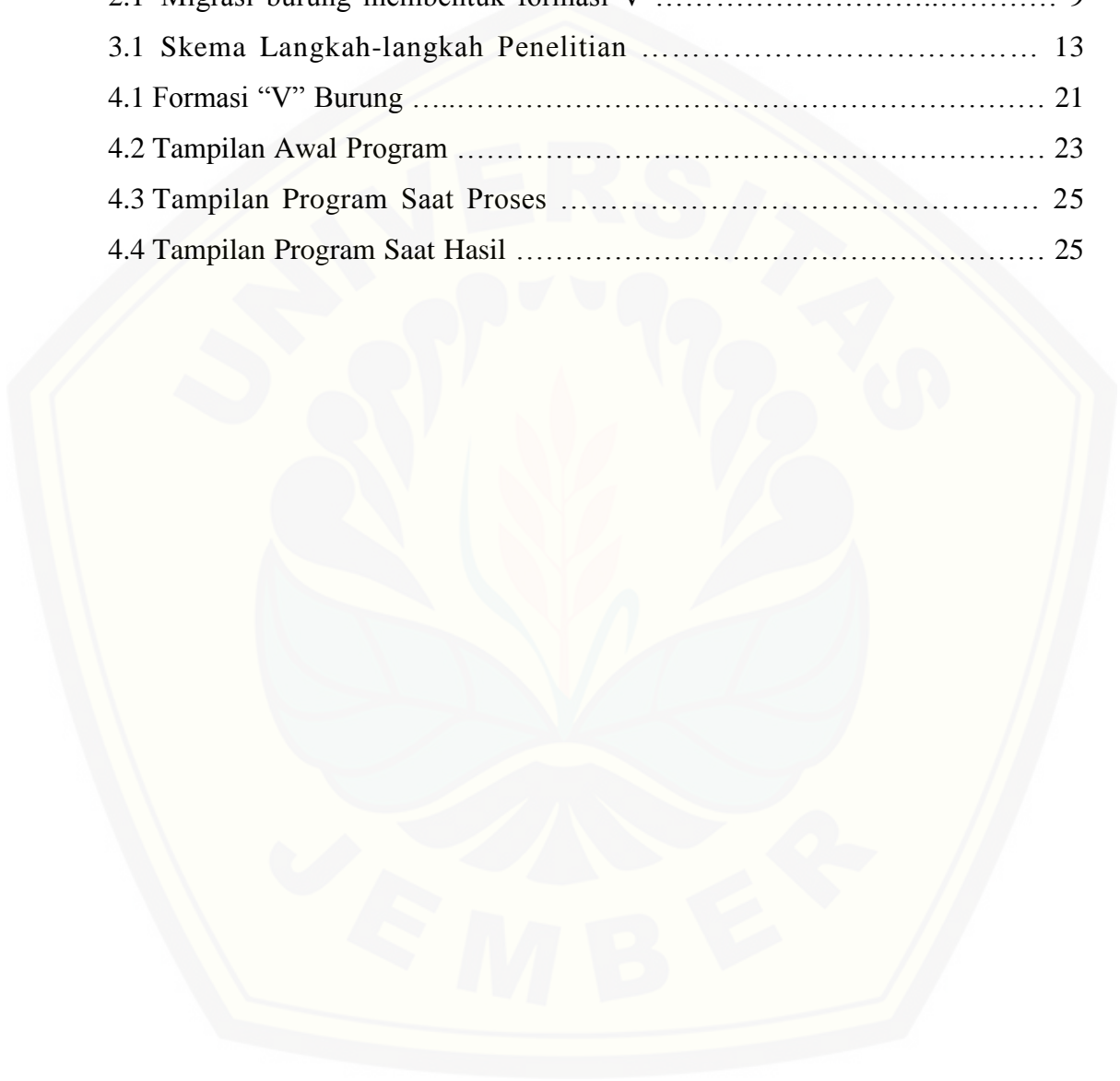
2.6.1 Inisialisasi Populasi Awal .....	9
2.6.2 Lingkungan dan Pembagian Solusi .....	10
2.6.3 Kriteria Pemberhentian .....	11
<b>BAB 3 METODE PENELITIAN .....</b>	<b>12</b>
<b>3.1 Langkah-langkah Penelitian .....</b>	<b>12</b>
<b>BAB. 4 HASIL DAN PEMBAHASAN.....</b>	<b>14</b>
<b>4.1 Hasil.....</b>	<b>14</b>
4.1.1 Perhitungan manual.....	14
4.1.2 Program.....	23
4.1.3 Simulasi Program.....	26
<b>4.2 Pembahasan.....</b>	<b>27</b>
<b>BAB 5. PENUTUP.....</b>	<b>30</b>
<b>5.1 Kesimpulan.....</b>	<b>30</b>
<b>5.2 Saran .....</b>	<b>30</b>
<b>DAFTAR PUSTAKA .....</b>	<b>31</b>
<b>LAMPIRAN.....</b>	<b>32</b>

**DAFTAR TABEL**

	Halaman
1.1 Sampel Data .....	15
1.2 Populasi Partikel Awal .....	16
1.3 Populasi Awal .....	16
1.4 Perhitungan Nilai Berat .....	17
1.5 Konversi Biner Ke Riil .....	17
1.6 Partikel proses <i>Penalty</i> .....	17
1.7 Hasil <i>Penalty</i> .....	18
1.8 Partikel Posisi Baru .....	19
1.9 Posisi Baru .....	23
1.10 Populasi Awal Algoritma MBO .....	24
1.11 Proses Lingkungan dan .....	24
1.12 Hasil.....	24
1.13 Perbandingan Parameter .....	26
1.14 Hasil Rata-rata perbandingan Populasi = 21 .....	27
1.15 Hasil Rata-rata perbandingan Populasi = 31 .....	27
1.16 Hasil Rata-rata perbandingan Populasi = 41 .....	27

**DAFTAR GAMBAR**

	Halaman
2.1 Migrasi burung membentuk formasi V .....	9
3.1 Skema Langkah-langkah Penelitian .....	13
4.1 Formasi “V” Burung .....	21
4.2 Tampilan Awal Program .....	23
4.3 Tampilan Program Saat Proses .....	25
4.4 Tampilan Program Saat Hasil .....	25





## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Setiap orang pasti menginginkan keuntungan yang maksimal dengan sumber atau modal seminimal mungkin. Salah satu contoh dalam kehidupan sehari-hari adalah persoalan media penyimpanan yang terbatas tetapi diharuskan mendapat keuntungan yang semaksimal mungkin. Dari permasalahan tersebut munculah istilah yang dikenal dengan masalah *knapsack*. *Knapsack* sendiri dapat dimisalkan sebagai suatu kantong atau media penyimpanan. Masalah *knapsack* merupakan suatu permasalahan bagaimana memilih objek dari sekian banyak objek dengan batasan objek tersebut sama atau lebih kecil dari kapasitas media penyimpanan sehingga diperoleh hasil yang optimal.

Suyanto (2010) menjelaskan bahwa permasalahan *knapsack* dibagi menjadi tiga jenis diantaranya adalah permasalahan *knapsack* 0-1, permasalahan *bounded knapsack*, permasalahan *unbounded knapsack*. Pembagian tersebut didasarkan atas pola penyimpanan barang dengan nilai dan bobot yang bervariasi. Pada penelitian ini penulis ingin membahas tentang masalah *knapsack* 0-1, dimana pada permasalahan *knapsack* 0-1 adalah permasalahan penyimpanan barang dimana barang akan dimasukkan secara utuh atau tidak sama sekali.

Permasalahan *knapsack* dapat diselesaikan dengan menggunakan algoritma metaheuristik. Metaheuristik merupakan metode untuk mencari solusi yang memadukan interaksi antar prosedur pencarian lokal dan strategi yang lebih tinggi untuk menciptakan proses yang mampu keluar dari titik-titik lokal optimal dan melakukan pencarian di ruang solusi untuk menemukan solusi global. Seiring dengan perkembangan jaman, perkembangan ilmu pengetahuan, dan teknologi, algoritma metaheuristik terus berkembang dan semakin efisien dalam menyelesaikan permasalahan optimasi. Efisiensi dari penerapan algoritma metaheuristik tidak lepas dari bantuan komputer, karena keterbatasan perhitungan dan penyelesaian manual yang dilakukan oleh manusia.



Banyak algoritma metaheuristik yang dapat digunakan untuk menyelesaikan masalah *knapsack* 0-1 diantaranya adalah algoritma *Penguins Search Optimization* (PeSOA) dan algoritma *Migrating Bird Optimization* (MBO). Gherabia dan Maoussaoui (2013) melakukan penelitian yang mengkaji tentang perilaku sekumpulan penguin dengan teknik pencarian makanan, dikembangkan menjadi sebuah algoritma metaheuristik *Penguins Search Optimization Algorithm*. Pada penelitiannya yang membandingkan algoritma *Penguins Search Optimization* dengan *Particle Swarm Optimization* (PSO) dan *Genetika Algorithm* (GA), menunjukkan bahwa algoritma *Penguins Search Optimization* lebih kuat dan efisien dibandingkan dengan algoritma *Particle Swarm Optimization* dan Algoritma *Genetika* karena strategi pencarian makanan tidak hanya mengandalkan perubahan posisi yang paling optimum tapi juga menemukan jalan yang paling optimum. Algoritma ini dapat mendeteksi semua minimum lokal dan global di ruang pencarian yang sangat besar.

Menurut Pradana (2018) yang melakukan penelitian tentang permasalahan *knapsack* 0-1 menggunakan algoritma *Migrating Bird Optimization* dan algoritma *Genetika*, dimana algoritma *Migrating Bird Optimization* dan algoritma *Genetika* (GA) memiliki hasil yang sama namun dalam penelitiannya Pradana menyatakan bahwa algoritma *Migrating Bird Optimization* lebih baik dibandingkan dengan algoritma *Genetika* karena memiliki iterasi yang lebih sedikit.

Berdasarkan dari penelitian tersebut penulis tertarik menyelesaikan masalah *Knapsack* 0-1 dengan menggunakan algoritma *Penguins Search Optimization Algorithm* dan algoritma *Migrating Bird Optimization*. Selain itu penulis ingin mengetahui kinerja dari algoritma *Penguins Search Optimization Algorithm* dan algoritma *Migrating Bird Optimization* dan membandingkan keduanya

## 1.2 Rumusan Masalah

Berdasar latar belakang yang telah diuraikan, maka permasalahan yang akan dibahas adalah

- a. Bagaimana perbandingan solusi optimal yang diperoleh dari penerapan algoritma *Penguins Search Optimization* (PeSOA) dan *Migrating Bird Optimization* (MBO) dalam permasalahan *knapsack* 0-1 berdasarkan keuntungan maksimum?
- b. Bagaimana perbandingan solusi optimal yang diperoleh dari penerapan algoritma *Penguins Search Optimization* (PeSOA) dan *Migrating Bird Optimization* (MBO) berdasarkan konvergensi, dan *running time* ?

### 1.3 Batasan Masalah

Adapun batasan permasalahan dalam penelitian ini adalah :

Tingkat penjualan atau permintaan konsumen untuk masing-masing barang yang diasumsikan sama.

### 1.4 Tujuan

Adapun tujuan dari penelitian ini adalah

- a. Mengetahui penyelesaian permasalahan *knapsack* 0-1 menggunakan algoritma *Penguins Search Optimization* (PeSOA) dan algoritma *Migrating Bird Optimization* (MBO).
- b. Membandingkan uji kovergensi algoritma *Penguins Search Optimization* (PeSOA) dengan algoritma *Migrating Bird Optimization* (MBO).

### 1.5 Manfaat

Adapun manfaat dari penelitian ini adalah

- a. Mendapatkan solusi yang optimal yaitu keuntungan yang maksimum dari pemilihan barang.
- b. Dapat dijadikan referensi dalam penelitian selanjutnya mengenai algoritma *Penguins Search Optimization* (PeSOA) dan algoritma *Migrating Bird Optimization* (MBO)

## BAB 2 TINJAUAN PUSTAKA

### 2.1 Optimasi

Optimasi berarti pencarian nilai terbaik dari beberapa fungsi yang diberikan pada suatu konteks. Secara umum optimasi adalah suatu proses untuk mencapai hasil yang optimal pada masalah yang berhubungan dengan keputusan terbaik. Untuk dapat mencapai nilai optimasi baik minimal ataupun maksimal tersebut, secara sistematis dilakukan pemilihan variabel bilangan bulat atau bilangan riil yang akan memberikan solusi optimal. Berikut ini beberapa persoalan yang memerlukan optimasi yang sering muncul dalam bentuk program linier (Munir, 2005):

- a. Penentuan pemilihan barang pada masalah *knapsack*.
- b. Menentukan lintasan terpendek dari suatu tempat ketempat yang lain.
- c. Menentukan jumlah pekerja seminimal mungkin untuk melakukan suatu proses produksi agar pengeluaran biaya pekerja dapat diminimalkan dan hasil produksi tetap maksimal.
- d. Mengatur jalur kendaraan umum agar semua lokasi dapat dijangkau.

### 2.2 Masalah *Knapsack* 0-1

Masalah *knapsack* 0-1 merupakan masalah pemilihan barang yang akan disimpan pada suatu media penyimpanan dari sekian banyak barang dimana setiap barang memiliki berat dan keuntungan. Penyelesaian masalah *Knapsack* 0-1 bertujuan untuk memilih barang-barang yang mempunyai keuntungan maksimum dan total berat tidak melebihi kapasitas berat media penyimpanan (Gherboudj, et al., 2012).

Jenis masalah *knapsack* yang dikenal antara lain masalah *knapsack* 0-1, masalah *bounded knapsack* dan masalah *unbounded knapsack*. Masalah *knapsack* 0-1 merupakan masalah *knapsack* dimana barang yang akan dimasukkan ke dalam media penyimpanan harus semua (1) atau tidak sama sekali (0). *Bounded knapsack* yaitu masalah *knapsack* dimana setiap barang tersedia sebanyak  $n$  unit

dan jumlah barang yang di maksukan ke dalam media penyimpanan terbatas. Masalah *unbounded knapsack* adalah masalah *knapsack* dimana setiap barang tersedia lebih dari satu unit dan jumlah barang yang dimasukkan kedalam media penyimpanan tidak terbatas (Pisinger,1995).

Permasalahan *knapsack* yang akan dibahas pada penelitian ini adalah permasalahan *knapsack* 0-1. Permasalahan *knapsack* 0-1 merupakan permasalahan dimana terdapat  $n$  buah objek dan sebuah media penyimpanan yang memiliki daya tampung maksimal senilai  $M$ . Benda-benda pada permasalahan ini memiliki bobot ( $x_i$ ) dengan nilai keuntungan profit ( $p_i$ ). Permasalahannya adalah bagaimana memilih objek-objek yang dimasukkan ke dalam media penyimpanan sehingga tidak melebihi kapasitas dari media penyimpanan namun memaksimalkan total keuntungan yang diperoleh. Permasalahan *knapsack* 0-1 memiliki sistem kerja dimana barang yang diangkut dimensinya harus diangkut seluruhnya atau tidak sama sekali (Dimiyati *et al.*, 2004).

Permasalahan *knapsack* 0-1 memiliki solusi penyelesaian yang dinyatakan sebagai himpunan:

$$X = \{x_1, x_2, x_3, \dots, x_n\}$$

dimana  $x_i = 1$  jika benda ke- $i$  dimasukkan ke dalam media penyimpanan, dan  $x_i = 0$  jika benda ke- $i$  tidak dimasukkan ke dalam media penyimpanan. Misalkan solusi dari suatu permasalahan *knapsack* adalah  $X = \{0,1,0,1\}$  itu berarti benda ke-1 dan ke-3 tidak dimasukkan ke dalam media penyimpanan sedangkan benda ke-2 dan ke-4 dimasukkan ke dalam media penyimpanan. Secara matematis, permasalahan *knapsack* 0-1 menurut Dimiyati *et al* (2004), sebagai berikut.

$$\text{Fungsi tujuan Maks} \quad : Z = \sum_{i=1}^n p_i x_i, \quad x_i \in \{0,1\} \quad (2.1)$$

$$\text{Kendala} \quad : z = \sum_{i=1}^n w_i x_i \leq M, x_i \in \{0,1\} \quad (2.2)$$

Keterangan:

$Z$ : nilai optimum dari fungsi tujuan

$z$  : kendala fungsi tujuan

$n$ : banyak barang

$p_i$ : keuntungan barang ke- $i$ , dimana  $i = 1, 2, 3, \dots, n$

$w_i$ : berat barang ke- $i$ , dimana  $i = 1, 2, 3, \dots, n$

$x_i$ : Nama barang ke- $i$ , dengan variabel keputusan (1 jika dipilih, 0 jika tidak dipilih)

$M$ : kapasitas media penyimpanan (*knapsack*)

### 2.3 Algoritma *Penguins Search Optimization* (PeSOA)

Gherabia dan Maoussaoui (2013), melakukan penelitian yang terinspirasi berdasarkan strategi kerjasama penguin saat mencari makanan, untuk mulai penencarian makanan dimulai dengan setiap masing masing individu penguin yang bergerak secara acak, dimana penguin akan memberikan informasi kepada kelompok mengenai posisi dan jumlah ikan yang ditemukan, tujuan kerjasama ini adalah untuk menyatukan informasi sehingga penguin mengetahui posisi dengan jumlah ikan terbanyak.

Mori (2002), mengatakan bahwa strategi penguin dalam pencarian makanan memiliki kesamaan dengan prinsip ekonomi, ketika energi yang didapat lebih besar daripada energi yang dikeluarkan, merupakan suatu keuntungan untuk penguin dalam mencari makanan. Penguin menggunakan strategi untuk mengolah informasi mengenai waktu dan pengeluaran saat mencari makanan dan kebutuhan energi untuk mencari makanan. Selain itu pilihan untuk mencari makanan atau tidak pada suatu daerah tergantung pada jumlah makanan dan jarak antara daerah tempat makanan.

Dalam algoritma *Penguins Search Optimization*, setiap penguin merepresentasikan posisi dan jumlah banyak ikan yang bisa dimakan. Pergerakan penguin didasarkan keberadaan ikan pada setiap posisi. Penguin memulai perburuan dalam posisi secara acak. Setelah menemukan kedalaman yang tepat penguin kembali ke permukaan dan mulai bertukar informasi tentang posisi dan jumlah makanan yang ditemukan. Oleh karena itu dalam setiap posisi baru dapat disesuaikan dengan persamaan berikut

$$D_{new} = D_{id} + Stepsize \cdot rand \cdot (|X_{Best} - X_{id}| + 1) \quad (2.3)$$

$D_{new}$  : solusi baru setelah menyelam



$D_{id}$	: solusi sebelum menyelam
Stepzise	: ukuran langkah
$rand()$	: bilangan acak
$X_{Best}$	: solusi terbaik dalam grup
$X_{id}$	: solusi awal di permukaan

## 2.4 Implementasi algoritma Penguins Search Optimization (PeSOA) pada *knapsack 0-1*

Berikut ini adalah implementasi algoritma PeSOA pada *knapsack 0-1*

### 2.4.1 Inisialisasi populasi

populasi awal pada algoritma PeSOA merepresentasikan ruang penyimpanan yang dimisalkan sebagai penguin P yang didalamnya terdapat sekumpulan barang dan dibentuk secara acak dalam bentuk bilangan riil kemudian dikonversikan menjadi bilangan biner tanpa melebihi kapasitas berat ( $M$ ) yang ditentukan, apabila melebihi kapasitas maka akan diberlakukan *penalty*. Proses *penalty* dilakukan dengan merubah kembali bentuk populasi dari bilangan biner menjadi bilangan riil awal yang ada didalam media penyimpanan. Pada populasi awal akan dipilih salah satu barang dari populasi secara acak, barang yang bernilai  $\geq 0,5$  maka akan dikurang 0,5 dan hasil pengurangan tersebut akan menggantikan nilai barang sebelumnya.

### 2.4.2 Perhitungan setiap penguin

Untuk memperoleh posisi baru dari setiap penguin, pada setiap iterasi akan dipilih penguin dengan nilai optimasi terbesar di dalam grup ( $X_{best}$ ). sebelum penyelaman setiap penguin memiliki solusi ( $X_{id}$ ). Dengan menggunakan perhitungan persamaan 2.3 akan didapat solusi baru ( $D_{new}$ ) yang nantinya solusi baru akan digunakan dalam perhitungan selanjutnya sebagai ( $D_{id}$ ). Terdapat parameter CO (Cadangan Oksigen) yang merepresantasikan berapa kali penguin akan melakukan penyelaman di setiap iterasi sampai cadangan oksigen habis dan kembali ke kelompok untuk berbagi informasi solusi optimal yang didapat.

### 2.4.3 Update solusi terbaik dan kriteria pemberhentian

Setelah semua penguin memiliki beberapa solusi optimal, maka akan dipilih solusi optimal terbaik dari setiap penguin yang menggantikan solusi optimal sebelumnya. Ketika solusi terbaik sudah diperbarui akan dipilih solusi terbaik di dalam kelompok penguin. Algoritma PeSOA akan berhenti ketika iterasi yang di gunakan telah mencapai maksimal.

## 2.5 Algoritma Migrating Bird Optimization (MBO)

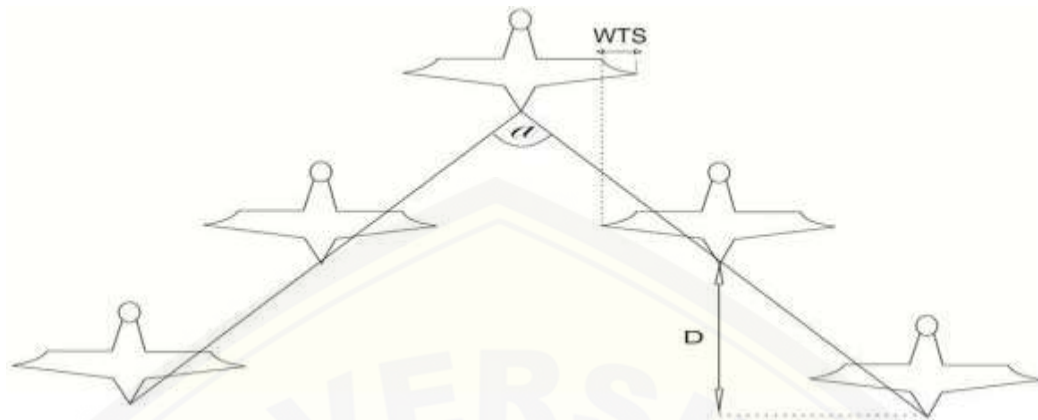
Duman et al. 2012, Dalam penelitiannya mengusulkan algoritma *Migrating Bird Optimization* yang terinspirasi dari oleh formasi huruf “V” yang dibentuk oleh kawanan burung ketika bermigrasi, formasi”V” bermanfaat sebagai efisiensi energi saat burung terbang dinangkasa. semakin efisien energi yang dikeluarkan maka kawanan burung dapat bermigrasi dengan jarak semakin jauh. Selain itu, formasi “V” juga dapat berguna untuk menghindari tabrakan antara sesama burung yang terdapat pada satu kawanan.

Menurut Lissaman dan Scholllenbarger (1970), mekanisme kerja dari kawanan burung yang bermigrasi membentuk formasi “V” sebagai berikut :

- a) Setiap kelompok dipimpin oleh satu pemimpin yang berada di bagian paling depan dari formasi. Pemimpin ini merupakan burung dengan kemampuan terbang yang terbaik. Tenaga yang dihabiskan oleh pemimpin formasi lebih besar dari pada anggota lain yang berada di belakang.
- b) Ketika burung pemimpin formasi mengalami kelelahan, maka akan digantikan dengan burung yang berada disebelah kirinya. Burung pemimpin yang mengalami kelelahan akan bergeser kesebelah kanan paling belakang dari formasi kawanan.
- c) Setiap anggota dari kawanan burung akan menjadi pemimpin formasi setidaknya sebanyak satu kali untuk menjadi pemimpin.

Dengan mekanisme yang digunakan dalam algoritma *migrating bird optimization* untuk memecahkan masalah optimisasi sehingga didapat solusi optimal. Formasi burung bermigrasi membentuk “V” ini bisa dilihat pada Gambar 2.1.





Gambar 2.1 Migrasi burung membentuk formasi V

Formasi “V” akan mendapatkan hasil yang maksimal apabila nilai  $WTS_{opt} = -0,005b$  dan  $D_{opt} = 2w$  (Lissaman dan Schollenberger,1970).

WTS = panjang sayap

$b$  = rentang sayap

$D$  = jarak antara ekor

$w$  = lebar maksimum sayap

## 2.6 Implementasi algoritma *Migrating Bird Optimization* pada masalah *Knapsack 0-1*

Menurut Ulker et al (2017), berikut ini merupakan implemtasi algoritma *migrating bird optimization* pada permasalahan *knapsack 0-1* sebagai berikut :

### 2.6.1 Inisialisasi Populasi Awal

populasi awal pada algoritma MBO merepresentasikan ruang penyimpanan yang dimisalkan sebagai burung B yang dibentuk secara acak dalam bentuk bilangan biner dengan syarat tidak melebihi kapasitas berat ( $M$ ) yang ditentukan, apabila melebihi kapsitas maka akan diberlakukan penalty. Proses penalty dilakukan dengan memilih secara acak barang yang diangkut dan meninggalkan barang yang terpilih. Setiap burung didalam populasi mewakili semua kemungkinan solusi yang dicari.

### 2.6.2 Lingkungan dan Pembagian Solusi

Burung pemimpin sedikitnya membentuk 3 solusi tetangga ( $k$ ) dengan syarat banyak solusi tetangga harus bernilai ganjil 3,5,7 dan seterusnya, solusi tetangga terbaik yang dibangkitkan digunakan untuk meningkatkan solusi saat ini. Sisa solusi yang lain dibagikan kepada burung anggota di sisi kiri dan kanan. Banyaknya solusi tetangga yang dibangkitkan dan banyaknya solusi yang dibagikan maka dibentuk parameter seperti berikut.

$$k \in N^+; k = \{3,5,7, \dots\} \quad (2.4)$$

$$x \in N^+; x = \{1,2, \dots, \frac{(k-1)}{2}\} \quad (2.5)$$

$$n = k - x \quad (2.6)$$

Keterangan

$k$  = jumlah solusi tetangga yang harus di pertimbangkan.

$x$  = jumlah solusi tetangga untuk dibagikan dengan solusi berikutnya

$n$  = jumlah solusi tetangga kecuali *sharing*.

Solusi yang diperoleh ketika menerapkan algoritma *Migrating Bird Optimization* (MBO) dipengaruhi faktor lingkungan. Faktor lingkungan disini merupakan solusi tetangga dari solusi utama yang merepresentasikan oleh kawanan burung disebelah kanan atau kiri dari burung pemimpin. Penentuan hasil dari solusi tetangga dilakukan hingga iterasi menghasilkan nilai yang optimal. Solusi tetangga yang paling optimal digunakan untuk memperbaiki solusi terbaik yang sudah didapatkan sebelumnya. Sisa solusi ( $x$ ) burung pemimpin dibagi  $\frac{x}{2}$  ke tetangga sebelah kiri kawanan dan  $\frac{x}{2}$  kanan sisi kawanan. Solusi tetangga terbaik pada burung selain burung pemimpin akan dipakai dan solusi yang tersisa akan dibagikan ke burung berikutnya. Burung berikutnya yang dimaksud adalah burung lain yang berada dibelakang barisan.

### 2.5.3 Kriteria Pemberhentian

Menurut Ulker *et al* (2017), proses iterasi akan terus berlangsung hingga kriteria pemberhentian terpenuhi. Kriteria pemberhentian pada algoritma *Migrating Birds Optimization* (MBO) akan terpenuhi apabila iterasi yang dilakukan mencapai maksimal atau hasil yang diperoleh konvergen. Iterasi maksimal tercapai apabila iterasi yang dilakukan sesuai dengan jumlah iterasi yang telah ditentukan pada awal perhitungan.



### BAB 3. METODE PENELITIAN

Penelitian penerapan algoritma *Penguins Search Optimization* (PeSOA) dan algoritma *Migrating Bird Optimization* (MBO) pada permasalahan *knapsack* 0-1 akan dilakukan dengan langkah-langkah sebagai berikut :

a. Studi literatur

Studi literatur dilakukan dengan mengumpulkan dan mempelajari berbagai literature atau teori tentang algoritma *Penguin Search Optimizations*, algoritma *Migrating Bird Optimization* serta teori lain yang berhubungan dengan permasalahan *knapsack* 0-1

b. Pengumpulan dan Identifikasi data

Peneliti akan menggunakan sampel data dari UD.ALAM SUBUR untuk menentukan perbandingan dari algoritma *Penguins Search Optimizations* dengan algoritma *Migrating Bird Optimization*.

c. Penerapan algoritma

Pada tahap ini peneliti akan menyesuaikan langkah-langkah algoritma *Penguins Search Optimization* dan algoritma *Migrating Bird Optimization* dengan masalah *knapsack* 0-1 berdasarkan di bab 2.

d. Pembuatan Program

Membuat program sesuai dengan algoritma yang telah di tentukan pada langkah (c) menggunakan *software* Matlab R2015b lalu di simulasikan menggunakan menu *guide* pada Matlab R2015b.

e. Analsis Hasil

Membandingkan kedua algoritma menggunakan program yang telah dibuat berdasarkan hasil keuntungan maksimal, kekonvergenan dan *running time*.

f. Kesimpulan

Membuat kesimpulan berdasarkan hasil simulasi menggunakan program yang nantinya akan memberikan jawaban yang mengacu dari tujuan peneliti. Saran-saran yang bersifat membangun dan dirasa perlu untuk perkembangan penelitian selanjutnya juga dapat dipaparkan pada bagian ini.

Skema langkah-langkah penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Skema Langkah-langkah Penelitian

## BAB 5. PENUTUP

### 5.1 Kesimpulan

Berdasarkan hasil dan pembahasan diambil kesimpulan sebagai berikut.

- a. Perbandingan algoritma *Penguins Search Optimization* (PeSOA) dengan algoritma *Migrating bird Optimization* (MBO) dalam menyelesaikan permasalahan *Knapsack* 0-1 memiliki solusi optimal yang sama sebesar Rp 10.364.000 dengan berat maksimal yang terangkut adalah 497 kg.
- b. Algoritma *Migrating Birds Optimization* (MBO) lebih baik daripada algoritma *Penguins Search Optimization* (PeSOA). Karena algoritma *Migrating Birds Optimization* (MBO) memiliki iterasi yang lebih sedikit dan *running time* yang lebih cepat konvergen dari algoritma *Penguins Search Optimization* (PeSOA) untuk memperoleh solusi yang optimal

### 5.2 Saran

Masalah *knapsack* 0-1 merupakan permasalahan optimasi yang dapat dikerjakan dengan berbagai macam metode dan algoritma. Masih terbuka bagi peneliti berikutnya untuk menerapkan algoritma metaheuristik yang lain dan membandingkan hasil dari masing-masing algoritma tersebut, sehingga didapatkan metode penyelesaian yang terbaik untuk menyelesaikan permasalahan yang *knapsack* 0-1.



**DAFTAR PUSTAKA**

- Dimiyati, T. Tjutju, dan A. Dimiyati. 2004. *Operations Research Model-model Pengambilan Keputusan*. Bandung: Sinar Baru Algensindo.
- Duman, E., M. Uysal, dan A.F. Alkaya. 2012. Migrating birds optimization: a new metaheuristic approach and its performance on quadratic assignment problem. *Information Science*, 217:65–77.
- Gherbaudj,A,A Layeb dan S.Chiki. 2012. Solving 0-1Knapsack Problem by a Discret Binari Version of Cuckoo Search Algorithm. *International Journal Bio-Inspired Comparation*,4(4): 229-236.
- Lissaman, P. B.,dan, C.A. Schollenberger. 1970. Formation flight of birds. *Science* 168: 1003–1005.
- Mori, Y .2002. Optimal diving behavior foraging in relation to body size. *The American Naturalist*,Vol.15,269-276.
- Munir,R. 2005. *Matematika Diskrit*. Revisi 5. Bandung: Informatika.
- Pisinger, D. 1995. A minimal Algorithm for the Multiple-Choice Knapsack Problem. *Europan Journal of Operational Research*, 83(2): 349-410.
- Pradana,R. 2018. Penerapan Algoritma Genetika dan Algoritma Migrating Bird Optimization (MBO) Pada Permasalahan *Knapsack* 0-1. *Skripsi*. Jember:Jurusan Matematika FMIPA Universitas Jember.
- Simpson,G. 1976. penguins : Past and Present,Here and There. Yale university press
- Suyanto. 2010. algoritma Optimasi : Deterministik atau Probabilistik, Yogyakarta: Graha Ilmu.
- Ulker, E., dan V. Tongur. 2017. Migrating birds optimization (MBO) algorithm to solve knapsack problem. *Procedia Computer Science*, 111: 71-76.
- Y,Gheraibia dan A, Moussaoui, 2013. Penguins Search Optimization Algorithm (PeSOA).*In Recent Trends in Applied Artificial Intelence*. Springer Berlin Heidelberg.



## LAMPIRAN

## Lampiran A. Data Penelitian

No.	Nama Barang	Berat Satuan (ml/gr)	Volume/isi	Harga Beli (Rp)	Harga Jual (Rp)
1.	AMNESTI 40 EC 1	100	50	13.500	20.000
2.	AMNESTI 40 EC 2	200	40	24.500	30.000
3.	AMNESTI 40 EC 3	400	20	43.500	55.000
4.	BARRIER GOLD 6 GR	1	10	23.000	30.000
5.	BI-LANCER BM 1	225	50	29.000	35.000
6.	BI-LANCER BM 2	450	20	55.000	65.000
7.	BI-LANCER UMBI1	225	50	29.000	35.000
8.	BI-LANCER UMBI2	450	20	55.000	65.000
9.	BI-LANCER PADI	225	50	29.000	35.000
10.	BIOTAN 45	1	12	66.000	80.000
11.	CROWEN 113 EC1	80	50	13.000	20.000
12.	CROWEN 113 EC2	200	40	23.500	30.000
13.	CROWEN 113 EC3	400	20	40.000	49.500
14.	EXTRA BOR1	250	50	29.000	30.000
15.	EXTRA BOR2	500	20	52.000	55.000
16.	GA JERUK	500	20	25.000	30.000
17.	GA PADI	500	20	38.000	41.000
18.	GA SAYUR DAUN	500	20	38.000	40.000
19.	GA BAWANG MERAH	500	20	38.000	40.000
20.	GA SEMANGKA	500	20	38.000	40.000
21.	GA RECOVERY	500	20	60.000	63.000
22.	GRAN BORER 1	100	50	18.000	20.000
23.	GRAN BORER 2	225	50	38.000	40.000
24.	GRAN BORER 3	450	20	60.000	70.000
25.	GRAN BORER 4	1	10	120.000	135.000
26.	HEXSAGON 1	100	50	21.000	25.000
27.	HEXSAGON 2	200	40	38.500	45.000
28.	HEXSAGON 3	400	20	70.000	75.000

No.	Nama Barang	Berat Satuan (ml/gr)	Volume/isi	Harga Beli (Rp)	Harga Jual (Rp)
29.	LUDES 50 EC 1	100	50	66.000	70.000
30.	LUDES 50 EC 2	250	30	156.000	165.000
31.	LEILA	50	50	30.500	40.000
32.	MUN UP 503 SL1	1	10	52.500	70.000
33.	MUN UP 503 SL2	5	4	226.000	250.000
34.	MUN UP 503 SL3	20	1	876.000	950.000
35.	MANXYL68 WP	500	20	75.000	90.000
36.	MAKMUR	1	10	55.000	90.000
37.	OPLOSAN 610 EC	80	50	16.500	20.000
38.	PLASTIK MULSA 9	9	5	305.000	350.000
39.	PLASTIK MULSA 15	15	4	457.000	500.000
40.	PLASTIKMULSA 18	18	2	730.000	790.000
41.	PAJERO 320 EC1	100	50	17.500	25.000
42.	PAJERO 320 EC2	200	40	24.000	30.000
43.	PAJERO 320 EC3	400	20	55.000	60.000
44.	TEKU 100 EC 1	100	50	21.000	30.000
45.	TEKU 100 EC 2	200	40	38.000	45.000
46.	TEKU 100 EC 3	400	20	72.500	80.000
47.	FOYER 70 WP 1	500	40	50.000	70.000
48.	FOYER 70 WP 2	1	20	92.500	110.000
49.	AMEGRASS 1	5	10	315.000	350.000
50.	AMEGRASS 2	20	2	350.000	600.000

#### Lampiran B. Hasil Perhitungan Selisih Harga Jual dengan Harga Beli

No.	Nama Barang	VOL/ISI	Berat Barang ( $w_i$ ) (kg)	Profit ( $p_i$ ) (Rp)
1.	AMNESTI 40 EC 1	50	5	325.000
2.	AMNESTI 40 EC 2	40	8	220.000
3.	AMNESTI 40 EC 3	20	8	230.000
4.	BARRIER GOLD 6 GR	10	10	70.000
5.	BI-LANCER BM 1	50	11,25	300.000
6.	BI-LANCER BM 2	20	9	200.000

No.	Nama Barang	VOL/ISI	Berat Barang ( $w_i$ ) (kg)	Profit ( $p_i$ ) (Rp)
7.	BI-LANCER UMBI1	50	11,25	300.000
8.	BI-LANCER UMBI2	20	9	200.000
9.	BI-LANCER PADI	50	11,25	300.000
10.	BIOTAN 45	12	12	168.000
11.	CROWEN 113 EC1	50	4	350.000
12.	CROWEN 113 EC2	40	8	260.000
13.	CROWEN 113 EC3	20	8	190.000
14.	EXTRA BOR1	50	12,5	50.000
15.	EXTRA BOR2	20	10	60.000
16.	GA JERUK	20	10	100.000
17.	GA PADI	20	10	60.000
18.	GA SAYUR DAUN	20	10	40.000
19.	GA BAWANG MERAH	20	10	40.000
20.	GA SEMANGKA	20	10	40.000
21.	GA RECOVERY	20	10	60.000
22.	GRAN BORER 1	50	5	100.000
23.	GRAN BORER 2	50	11,25	100.000
24.	GRAN BORER 3	20	9	200.000
25.	GRAN BORER 4	10	10	150.000
26.	HEXSAGON 1	50	5	200.000
27.	HEXSAGON 2	40	8	260.000
28.	HEXSAGON 3	20	8	100.000
29.	LUDES 50 EC 1	50	5	200.000
30.	LUDES 50 EC 2	30	7,5	270.000
31.	LEILA	50	2,5	475.000
32.	MUN UP 503 SL1	10	10	175.000
33.	MUN UP 503 SL2	4	20	96.000
34.	MUN UP 503 SL3	1	20	74.000
35.	MANXYL68 WP	20	10	300.000
36.	MAKMUR	10	10	350.000
37.	OPLOSAN 610 EC	50	4	175.000
38.	PLASTIK MULSA 9	5	45	225.000

No.	Nama Barang	VOL/ISI	Berat Barang ( $w_i$ ) (kg)	Profit ( $p_i$ ) (Rp)
39.	PLASTIK MULSA 15	4	60	172.000
40.	PLASTIKMULSA 18	2	36	120.000
41.	PAJERO 320 EC1	50	5	375.000
42.	PAJERO 320 EC2	40	8	240.000
43.	PAJERO 320 EC3	20	8	100.000
44.	TEKU 100 EC 1	50	5	450.000
45.	TEKU 100 EC 2	40	8	280.000
46.	TEKU 100 EC 3	20	8	150.000
47.	FOYER 70 WP 1	40	20	800.000
48.	FOYER 70 WP 2	20	20	350.000
49.	AMEGRASS 1	10	50	350.000
50.	AMEGRASS 2	2	40	500.000

### Lampiran C. Perhitungan manual PeSOA

#### C1. Perhitungan Berat Total Setiap Penguins Pada Populasi Awal

$$\text{Kendala : } z = \sum_{i=1}^n w_i x_i \leq 130$$

$$\begin{aligned} \text{Penguin1} &= w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 \\ &= (60x_1) + (20x_0) + (12x_0) + (20x_0) + (50x_0) \\ &= 60 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{Penguin2} &= w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 \\ &= (60x_1) + (20x_1) + (12x_1) + (20x_0) + (50x_0) \\ &= 92 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{Penguin3} &= w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 \\ &= (60x_1) + (20x_0) + (12x_1) + (20x_0) + (50x_0) \\ &= 72 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{Penguin4} &= w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 \\ &= (60x_0) + (20x_0) + (12x_0) + (20x_1) + (50x_1) \\ &= 70 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{Penguin5} &= w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 \\ &= (60x_1) + (20x_0) + (12x_1) + (20x_0) + (50x_1) \\ &= 142 \text{ kg} \end{aligned}$$

**C2. Perhitungan Profit Setiap Penguins Pada Populasi Awal**

$$\text{Fungsi tujuan : } Z = \sum_{i=1}^n p_i x_i$$

$$\begin{aligned} \text{Penguin1} &= p_1 x_1 + p_2 x_2 + p_3 x_3 + p_4 x_4 + p_5 x_5 \\ &= (172.000x1) + (96.000x0) + (168.000x0) + (800.000x0) + \\ &\quad (350.000x0) \\ &= \text{Rp.172.000} \end{aligned}$$

$$\begin{aligned} \text{Penguin2} &= p_1 x_1 + p_2 x_2 + p_3 x_3 + p_4 x_4 + p_5 x_5 \\ &= (172.000x1) + (96.000x1) + (168.000x1) + (800.000x0) + \\ &\quad (350.000x0) \\ &= \text{Rp.436.000} \end{aligned}$$

$$\begin{aligned} \text{Penguin3} &= p_1 x_1 + p_2 x_2 + p_3 x_3 + p_4 x_4 + p_5 x_5 \\ &= (172.000x1) + (96.000x0) + (168.000x1) + (800.000x0) + \\ &\quad (350.000x0) \\ &= \text{Rp.340.000} \end{aligned}$$

$$\begin{aligned} \text{Penguin4} &= p_1 x_1 + p_2 x_2 + p_3 x_3 + p_4 x_4 + p_5 x_5 \\ &= (172.000x0) + (96.000x0) + (168.000x0) + (800.000x1) + \\ &\quad (350.000x1) \\ &= \text{Rp.1150.000} \end{aligned}$$

$$\begin{aligned} \text{Penguin5} &= p_1 x_1 + p_2 x_2 + p_3 x_3 + p_4 x_4 + p_5 x_5 \\ &= (172.000x1) + (96.000x0) + (168.000x1) + (800.000x0) + \\ &\quad (350.000x1) \\ &= \text{Rp.690.000} \end{aligned}$$

**C3. Perhitungan Berat dan Profit Setiap Penguins Pada Populasi Baru Pada Iterasi 1**

Penguins	Partikel				
P1	0,6787	0,1712	0,0971	0,0344	0,1869
P2	0,7577	0,7060	0,8235	0,4387	0,4898
P3	0,7431	0,0318	0,6948	0,3816	0,4456
P4	0,3922	0,2769	0,3171	0,7655	0,6463
P5	0,6555	0,0462	0,9502	0,2952	0,7094



Diketahui P4 adalah  $X_{best}$  karena memiliki profit paling besar didalam grup sebelum penguin melakukan penyelaman.  $X_{id}$  adalah solusi awal pada penguins. Rand () adalah nilai random yang dibangkitkan dalam bentuk bilangan rill

a. Penguin 1,  $CO = 2$ ,  $Stepzise = 1$ ,  $iterasi = 1$

➤ Posisi baru penyelaman pertama ( $CO = 1$ ). Untuk langkah awal perhitungan

$$X_{id} = D_{id}$$

$X_{best}$	= 0,3922	0,2769	0,3171	0,7655	0,6463
$X_{id}$	= 0,6787	0,1712	0,0971	0,0344	0,1869
rand ()	= 0,5155	-0,2155	0,6576	-0,9363	0,9077
$D_{id}$	= 0,6787	0,1712	0,0971	0,0344	0,1869

$$D_{new} = D_{id} + Stepsize \cdot rand \cdot (|X_{Best} - X_{id}| + 1)$$

$$D_{new} = 1,3419 \quad -0,0671 \quad -0,7052 \quad -1,5864 \quad -1,1378$$

Karena solusi baru terdapat nilai  $> 1$  maka akan dilakukan tranformasi  $\frac{x}{x_{max}}$ .

Untuk nilai  $< 0$  dilakukan tranformasi  $\frac{x-x_{min}}{x_{max}-x_{min}}$

$$D_{new} = 1 \quad 0,5188 \quad 0,3009 \quad 0 \quad 0,1532$$

Kemudian di konversi menjadi biner

$$D_{new} = 1 \quad 1 \quad 0 \quad 0 \quad 0$$

$$w = 60Kg + 20Kg + 0 + 0 + 0 = 80Kg$$

$$Z = Rp.172.000 + Rp.96.000 + 0 + 0 + 0 = Rp.268.000$$

Setelah melakukan penyelaman pertama, hasil dari  $D_{new}$  akan dibandingkan dengan solusi awal yang merujuk pada Tabel 4.7, diketahui bahwa solusi awal P1 memiliki profit sebesar Rp.172.000, sehingga solusi baru ( $D_{new}$ ) akan menggantikan solusi awal dari P1 karena memiliki nilai profit yang lebih baik Rp.268.000.

➤ Posisi baru penyelaman kedua ( $CO = 2$ ) untuk penyelaman selanjutnya  $D_{id}$  diperoleh dari  $D_{new}$  penyelaman sebelumnya.

$$X_{best} = 0,3922 \quad 0,2769 \quad 0,3171 \quad 0,7655 \quad 0,6463$$



$$\begin{array}{r}
 X_{id} = 0,6787 \quad 0,1712 \quad 0,0971 \quad 0,0344 \quad 0,1869 \\
 \text{rand} () = 0,4863 \quad 0,3110 \quad 0,4121 \quad -0,4462 \quad -0,8057 \\
 D_{id} = 1 \quad 0,5188 \quad 0,3009 \quad 0 \quad 0,1532
 \end{array}$$

$$D_{new} = D_{id} + \text{Stepsize} \cdot \text{rand} \cdot (|X_{Best} - X_{id}| + 1)$$

$$D_{new} = 1,6256 \quad 0,8627 \quad 0,8037 \quad -0,7724 \quad -1,1758$$

Karena solusi baru terdapat nilai  $> 1$  maka akan dilakukan tranformasi  $\frac{x}{x_{max}}$ .

Untuk nilai  $< 0$  dilakukan tranformasi  $\frac{x-x_{min}}{x_{max}-x_{min}}$

$$D_{new} = 1 \quad 0,7119 \quad 0,6896 \quad 0,0945 \quad 0$$

Kemudian di konversi menjadi biner

$$D_{new} = 1 \quad 1 \quad 1 \quad 0 \quad 0$$

$$w = 60Kg + 20Kg + 12Kg + 0 + 0 = 92Kg$$

$$Z = Rp. 172.000 + Rp. 96.000 + Rp. 168.000 + 0 + 0 = Rp. 436.000$$

Setelah melakukan penyelaman kedua, hasil dari  $D_{new}$  akan dibandingkan dengan solusi pada penyelaman pertama, diketahui bahwa solusi penyelaman pertama memiliki profit sebesar Rp.268.000, sehingga solusi baru pada penyelaman kedua ( $D_{new}$ ) akan menggantikan solusi penyelaman pertama karena memiliki profit yang lebih besar Rp.436.000.

b. Penguin 2  $p = 5, CO = 2, \text{Stepsize} = 1, \text{iterasi} = 1$

➤ Posisi baru penyelaman pertama ( $CO = 1$ ). untuk penyelaman selanjutnya  $D_{id}$  diperoleh dari  $D_{new}$  penyelaman sebelumnya.

$$\begin{array}{r}
 X_{best} = 0,3922 \quad 0,2769 \quad 0,3171 \quad 0,7655 \quad 0,6463 \\
 X_{id} = 0,7577 \quad 0,7060 \quad 0,8235 \quad 0,4387 \quad 0,4898 \\
 \text{rand} () = 0,6469 \quad -0,3658 \quad 0,9311 \quad -0,2369 \quad 0,6828 \\
 D_{id} = 1 \quad 0,7119 \quad 0,6896 \quad 0,0945 \quad 0
 \end{array}$$

$$D_{new} = D_{id} + \text{Stepsize} \cdot \text{rand} \cdot (|X_{Best} - X_{id}| + 1)$$

$$D_{new} = 1,8833 \quad 0,1891 \quad -0,7130 \quad -0,2198 \quad 0,6828$$

Karena solusi baru terdapat nilai  $> 1$  maka akan dilakukan tranformasi  $\frac{x}{x_{max}}$ .

Untuk nilai  $< 0$  dilakukan tranformasi  $\frac{x-x_{min}}{x_{max}-x_{min}}$

$$D_{new} = \begin{matrix} 1 & 0,3475 & 0 & 0,1900 & 0,5376 \end{matrix}$$

Kemudian di konversi menjadi biner

$$D_{new} = 1 \quad 1 \quad 0 \quad 0 \quad 1$$

$$w = 60 \text{ kg} + 0 + 0 + 0 + 50 \text{ kg} = 110 \text{ kg}$$

$$Z = \text{Rp.} 172.000 + 0 + 0 + 0 + \text{Rp.} 350.000 = \text{Rp.} 522.000$$

Setelah melakukan penyelaman pertama, hasil dari  $D_{new}$  akan dibandingkan dengan solusi awal yang merujuk pada Tabel 4.7, diketahui bahwa solusi awal P2 memiliki profit sebesar Rp.436.000, sehingga solusi baru ( $D_{new}$ ) akan menggantikan solusi awal dari P2 karena memiliki nilai profit yang lebih baik Rp.522.000.

➤ Posisi baru penyelaman kedua ( $CO = 2$ ) untuk penyelaman selanjutnya  $D_{id}$  diperoleh dari  $D_{new}$  penyelaman sebelumnya.

$$X_{best} = 0,3922 \quad 0,2769 \quad 0,3171 \quad 0,7655 \quad 0,6463$$

$$X_{id} = 0,7577 \quad 0,7060 \quad 0,8235 \quad 0,4387 \quad 0,4898$$

$$\text{rand} () = 0,3897 \quad 0,9004 \quad -0,1227 \quad 0,5310 \quad -0,6263$$

$$D_{id} = \begin{matrix} 1 & 0,3475 & 0 & 0,1900 & 0,5376 \end{matrix}$$

$$D_{new} = D_{id} + \text{Stepsize} \cdot \text{rand} \cdot (|X_{Best} - X_{id}| + 1)$$

$$D_{new} = 1,5321 \quad 1,6343 \quad -0,1848 \quad 0,8945 \quad -0,1867$$

Karena solusi baru terdapat nilai  $> 1$  maka akan dilakukan tranformasi  $\frac{x}{x_{max}}$ .

Untuk nilai  $< 0$  dilakukan tranformasi  $\frac{x-x_{min}}{x_{max}-x_{min}}$

$$D_{new} = 0,9439 \quad 1 \quad 0 \quad 0,5937 \quad 0,0010$$

Kemudian di konversi menjadi biner

$$D_{new} = 1 \quad 1 \quad 0 \quad 1 \quad 0$$

$$w = 60 \text{ kg} + 20 \text{ kg} + 0 + 20 \text{ kg} + 0 = 100 \text{ kg}$$

$$Z = \text{Rp.} 172.000 + \text{Rp.} 96.000 + 0 + \text{Rp.} 800.000 + 0 = \text{Rp.} 1.068.000$$

Setelah melakukan penyelaman kedua, hasil dari  $D_{new}$  akan dibandingkan dengan solusi pada penyelaman pertama, diketahui bahwa solusi penyelaman pertama

memiliki profit sebesar Rp.522.000, sehingga solusi baru pada penyelaman kedua ( $D_{new}$ ) akan menggantikan solusi penyelaman pertama karena memiliki profit yang lebih besar Rp.1.068.000.

c. Penguin 3  $p = 5, CO = 2, Stepzise = 1, iterasi = 1$

➤ Posisi baru penyelaman pertama ( $CO = 1$ ). untuk penyelaman selanjutnya  $D_{id}$  diperoleh dari  $D_{new}$  penyelaman sebelumnya.

$X_{best}$	= 0,3922	0,2769	0,3171	0,7655	0,6463
$X_{id}$	= 0,7431	0,0318	0,6948	0,3816	0,4456
rand ()	= 0,6469	-0,3658	0,9311	-0,2369	0,6828
$D_{id}$	= 0,9439	1	0	0,5937	0,0010

$$D_{new} = D_{id} + Stepzise \cdot rand \cdot (|X_{Best} - X_{id}| + 1)$$

$$D_{new} = 2,0403 \quad 2,0294 \quad -1,1089 \quad 0,7235 \quad 1,1174$$

Karena solusi baru terdapat nilai  $> 1$  maka akan dilakukan tranformasi  $\frac{x}{x_{max}}$ .

Untuk nilai  $< 0$  dilakukan tranformasi  $\frac{x-x_{min}}{x_{max}-x_{min}}$

$$D_{new} = \quad 1 \quad 0,9965 \quad 0 \quad 0,0819 \quad 0,7069$$

Kemudian di konversi menjadi biner

$$D_{new} = 1 \ 1 \ 0 \ 0 \ 1$$

$$w = 60Kg + 20Kg + 0 + 0 + 50 kg = 130Kg$$

$$Z = Rp.172.000 + Rp.96.000 + 0 + 0 + Rp.350.000 = Rp.618.000$$

Setelah melakukan penyelaman pertama, hasil dari  $D_{new}$  akan dibandingkan dengan solusi awal yang merujuk pada Tabel 4.7, diketahui bahwa solusi awal P3 memiliki profit sebesar Rp.340.000, sehingga solusi baru ( $D_{new}$ ) akan menggantikan solusi awal dari P2 karena memiliki nilai profit yang lebih baik Rp.618.000.

Posisi baru penyelaman kedua ( $CO = 2$ )

$X_{best}$	= 0,3922	0,2769	0,3171	0,7655	0,6463
$X_{id}$	= 0,7431	0,0318	0,6948	0,3816	0,4456

$$\begin{array}{l} \text{rand} () = 0,6294 \quad -0,7460 \quad 0,2647 \quad -0,4430 \quad 0,9150 \\ D_{id} = 0,9439 \quad 1 \quad 0 \quad 0,5937 \quad 0,0010 \end{array}$$

$$D_{new} = D_{id} + \text{Stepsize} \cdot \text{rand} \cdot (|X_{Best} - X_{id}| + 1)$$

$$D_{new} = 1,8503 \quad 0,0677 \quad 0,3647 \quad -0,5312 \quad 1,8055$$

Karena solusi baru terdapat nilai  $> 1$  maka akan dilakukan tranformasi  $\frac{x}{x_{max}}$ .

Untuk nilai  $< 0$  dilakukan tranformasi  $\frac{x-x_{min}}{x_{max}-x_{min}}$

$$D_{new} = 1 \quad 0,2515 \quad 0,3762 \quad 0 \quad 0,9812$$

Kemudian di konversi menjadi biner

$$D_{new} = 1 \ 0 \ 0 \ 0 \ 1$$

$$w = 60Kg + 0 + 0 + 0 + 50Kg = 110Kg$$

$$Z = Rp.172.000 + 0 + 0 + 0 + Rp.350.000 = Rp.522.000$$

Setelah melakukan penyelaman kedua, hasil dari  $D_{new}$  akan dibandingkan dengan solusi pada penyelaman pertama, diketahui bahwa solusi penyelaman pertama memiliki profit sebesar Rp.618.000, sehingga solusi baru pada penyelaman kedua ( $D_{new}$ ) tetap menggunakan solusi penyelaman pertama karena memiliki profit yang lebih rendah Rp.522.000.

d. Penguin 4 ,  $CO = 2$ ,  $Stepsize = 1$ ,  $iterasi = 1$

➤ Posisi baru penyelaman pertama ( $CO = 1$ ). untuk penyelaman selanjutnya  $D_{id}$  diperoleh dari  $D_{new}$  penyelaman sebelumnya.

$$\begin{array}{l} X_{best} = 0,3922 \quad 0,2769 \quad 0,3171 \quad 0,7655 \quad 0,6463 \\ X_{id} = 0,3922 \quad 0,2769 \quad 0,3171 \quad 0,7655 \quad 0,6463 \\ \text{rand} () = 0,6294 \quad -0,7460 \quad 0,2647 \quad -0,4430 \quad 0,9150 \\ D_{id} = 1 \quad 0,2515 \quad 0,3762 \quad 0 \quad 0,9812 \end{array}$$

$$D_{new} = D_{id} + \text{Stepsize} \cdot \text{rand} \cdot (|X_{Best} - X_{id}| + 1)$$

$$D_{new} = 1,6294 \quad 0,2505 \quad 0,2647 \quad -0,3611 \quad 1,6219$$

Karena solusi baru terdapat nilai  $> 1$  maka akan dilakukan tranformasi  $\frac{x}{x_{max}}$ .

Untuk nilai  $< 0$  dilakukan tranformasi  $\frac{x-x_{min}}{x_{max}-x_{min}}$

$$D_{new} = \begin{matrix} 1 & 0,3073 & 0,3114 & 0 & 0,9962 \end{matrix}$$

Kemudian di konversi menjadi biner

$$D_{new} = 10001$$

$$w = 60Kg + 0 + 0 + 0 + 50Kg = 110Kg$$

$$Z = Rp.172.000 + 0 + 0 + 0 + Rp.350.000 = Rp.522.000$$

Setelah melakukan penyelaman pertama, hasil dari  $D_{new}$  akan dibandingkan dengan solusi awal yang merujuk pada Tabel 4.7, diketahui bahwa solusi awal P4 memiliki profit sebesar Rp.1.150.000, sehingga solusi awal P4 tidak akan digantikan solusi baru yang memiliki profit lebih kecil Rp.522.000.

- Posisi baru penyelaman pertama ( $CO = 2$ ). untuk penyelaman selanjutnya  $D_{id}$  diperoleh dari  $D_{new}$  penyelaman sebelumnya.

$$X_{best} = \begin{matrix} 0,3922 & 0,2769 & 0,3171 & 0,7655 & 0,6463 \end{matrix}$$

$$X_{id} = \begin{matrix} 0,3922 & 0,2769 & 0,3171 & 0,7655 & 0,6463 \end{matrix}$$

$$\text{rand}() = \begin{matrix} 0,8116 & 0,8268 & -0,8049 & 0,0938 & 0,9298 \end{matrix}$$

$$D_{id} = \begin{matrix} 1 & 0,3073 & 0,3114 & 0 & 0,9962 \end{matrix}$$

$$D_{new} = D_{id} + \text{Stepsize} \cdot \text{rand} \cdot (|X_{Best} - X_{id}| + 1)$$

$$D_{new} = \begin{matrix} 1,8116 & 0,3073 & -0,4905 & 0,0938 & 1,8596 \end{matrix}$$

Karena solusi baru terdapat nilai  $> 1$  maka akan dilakukan tranformasi  $\frac{x}{x_{max}}$ .

Untuk nilai  $< 0$  dilakukan tranformasi  $\frac{x-x_{min}}{x_{max}-x_{min}}$

$$D_{new} = \begin{matrix} 0,9796 & 0,6913 & 0 & 0,2486 & 1 \end{matrix}$$

Kemudian di konversi menjadi biner

$$D_{new} = 11001$$

$$w = 60Kg + 20Kg + 0 + 0 + 50Kg = 130Kg$$

$$Z = Rp.172.000 + Rp.96.000 + 0 + 0 + Rp.350.000 = Rp.618.000$$

Setelah melakukan penyelaman kedua, hasil dari  $D_{new}$  akan dibandingkan dengan solusi awal yang merujuk pada Tabel 4.7, diketahui bahwa solusi awal P4 memiliki profit sebesar Rp.1.150.000, sehingga solusi awal P4 tidak akan digantikan solusi baru yang memiliki profit lebih kecil Rp.618.000.



e. Penguin 5 ,  $CO = 2$ ,  $Stepzise = 1$ ,  $iterasi = 1$

➤ Posisi baru penyelaman pertama ( $CO = 1$ ). untuk penyelaman selanjutnya  $D_{id}$  diperoleh dari  $D_{new}$  penyelaman sebelumnya.

$X_{best}$	= 0,3922	0,2769	0,3171	0,7655	0,6463
$X_{id}$	= 0,6555	0,0462	0,9502	0,2952	0,7094
rand ()	= -0,6848	0,9143	0,6006	-0,1565	0,5844
$D_{id}$	= 0,9796	0,6913	0	0,2486	1

$$D_{new} = D_{id} + Stepsize \cdot rand \cdot (|X_{Best} - X_{id}| + 1)$$

$$D_{new} = 0,1145 \quad 1,8165 \quad 0,9808 \quad 0,0185 \quad 1,6213$$

Karena solusi baru terdapat nilai  $> 1$  maka akan dilakukan tranformasi  $\frac{x}{x_{max}}$ .

Untuk nilai  $< 0$  dilakukan tranformasi  $\frac{x-x_{min}}{x_{max}-x_{min}}$

$$D_{new} = 0,0630 \quad 1 \quad 0,5399 \quad 0,0102 \quad 0,8925$$

Kemudian di konversi menjadi biner

$$D_{new} = 0 \ 1 \ 1 \ 0 \ 1$$

$$w = 0 + 20Kg + 12Kg + 0 + 50Kg = 82Kg$$

$$Z = 0 + Rp. 96.000 + Rp. 168.000 + 0 + Rp. 350.000 = Rp. 614.000$$

Setelah melakukan penyelaman pertama, hasil dari  $D_{new}$  akan dibandingkan dengan solusi awal yang merujuk pada Tabel 4.7, diketahui bahwa solusi awal P5 memiliki profit sebesar Rp.690.000, sehingga solusi awal P5 tidak akan digantikan solusi baru yang memiliki profit lebih kecil Rp.614.000.

➤ Posisi baru penyelaman pertama ( $CO = 2$ ). untuk penyelaman selanjutnya  $D_{id}$  diperoleh dari  $D_{new}$  penyelaman sebelumnya.

$X_{best}$	= 0,3922	0,2769	0,3171	0,7655	0,6463
$X_{id}$	= 0,6555	0,0462	0,9502	0,2952	0,7094
rand ()	= 0,9412	-0,0292	-0,7162	0,8315	0,9190
$D_{id}$	= 0,0630	1	0,5399	0,0102	0,8925



$$D_{new} = D_{id} + Stepsize \cdot rand \cdot (|X_{Best} - X_{id}| + 1)$$

$$D_{new} = 1,2520 \quad 0,9641 \quad -0,6297 \quad 1,2328 \quad 1,8694$$

Karena solusi baru terdapat nilai  $> 1$  maka akan dilakukan tranformasi  $\frac{x}{x_{max}}$ .

Untuk nilai  $< 0$  dilakukan tranformasi  $\frac{x-x_{min}}{x_{max}-x_{min}}$

$$D_{new} = 0,2529 \quad 0,6377 \quad 0 \quad 0,7452 \quad 1$$

Kemudian di konversi menjadi biner

$$D_{new} = 0 \ 1 \ 0 \ 1 \ 1$$

$$w = 0 + 20Kg + 0 + 20Kg + 50Kg = 90Kg$$

$$Z = 0 + Rp. 96.000 + 0 + Rp. 800.000 + Rp. 350.000 = Rp. 1.246.000$$

Setelah melakukan penyelaman kedua, hasil dari  $D_{new}$  akan dibandingkan dengan solusi awal yang merujuk pada Tabel 4.7, diketahui bahwa solusi awal P5 memiliki profit sebesar Rp.690.000, sehingga solusi awal P5 akan digantikan solusi baru yang memiliki profit lebih besar Rp.1.246.000.

## Lampiran D. Perhitungan manual MBO

### D1.Perhitungan Berat Total Setiap Burung Pada Populasi Awal

$$\text{Kendala : } z = \sum_{i=1}^n w_i x_i \leq 130$$

$$\begin{aligned} \text{Burung1} &= w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 \\ &= (60x1) + (20x0) + (12x0) + (20x0) + (50x0) \\ &= 60 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{Burung2} &= w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 \\ &= (60x1) + (20x0) + (12x1) + (20x0) + (50x1) \\ &= 122 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{Burung3} &= w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 \\ &= (60x1) + (20x0) + (12x1) + (20x0) + (50x0) \\ &= 72 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{Burung4} &= w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 \\ &= (60x0) + (20x1) + (12x1) + (20x1) + (50x1) \\ &= 82 \text{ kg} \end{aligned}$$

$$\begin{aligned} \text{Burung5} &= w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 \\ &= (60x1) + (20x0) + (12x1) + (20x1) + (50x0) \end{aligned}$$

$$= 92\text{kg}$$

### D2. Perhitungan Profit Setiap Burung Pada Populasi Awal

$$\text{Fungsi tujuan : } Z = \sum_{i=1}^n p_i x_i$$

$$\begin{aligned} \text{Burung1} &= p_1 x_1 + p_2 x_2 + p_3 x_3 + p_4 x_4 + p_5 x_5 \\ &= (172.000x1) + (96.000x0) + (168.000x0) + (800.000x0) + \\ &\quad (350.000x0) \\ &= \text{Rp.172.000} \end{aligned}$$

$$\begin{aligned} \text{Burung2} &= p_1 x_1 + p_2 x_2 + p_3 x_3 + p_4 x_4 + p_5 x_5 \\ &= (172.000x1) + (96.000x0) + (168.000x1) + (800.000x0) + \\ &\quad (350.000x1) \\ &= \text{Rp.578.000} \end{aligned}$$

$$\begin{aligned} \text{Burung3} &= p_1 x_1 + p_2 x_2 + p_3 x_3 + p_4 x_4 + p_5 x_5 \\ &= (172.000x1) + (96.000x0) + (168.000x1) + (800.000x0) + \\ &\quad (350.000x0) \\ &= \text{Rp.340.000} \end{aligned}$$

$$\begin{aligned} \text{Burung4} &= p_1 x_1 + p_2 x_2 + p_3 x_3 + p_4 x_4 + p_5 x_5 \\ &= (172.000x0) + (96.000x1) + (168.000x1) + (800.000x0) + \\ &\quad (350.000x1) \\ &= \text{Rp.618.000} \end{aligned}$$

$$\begin{aligned} \text{Burung5} &= p_1 x_1 + p_2 x_2 + p_3 x_3 + p_4 x_4 + p_5 x_5 \\ &= (172.000x1) + (96.000x0) + (168.000x1) + (800.000x1) + \\ &\quad (350.000x0) \\ &= \text{Rp.1.028.000} \end{aligned}$$

### D3. Perhitungan Profit Setiap Burung Pada Populasi Awal

$$\text{Fungsi tujuan : } Z = \sum_{i=1}^n p_i x_i$$

$$\begin{aligned} (N_{11}) &= p_1 x_1 + p_2 x_2 + p_3 x_3 + p_4 x_4 + p_5 x_5 \\ &= (172.000x1) + (96.000x0) + (168.000x0) + (800.000x1) + \\ &\quad (350.000x1) \\ &= \text{Rp.1.322.000} \end{aligned}$$

$$\begin{aligned} (N_{12}) &= p_1 x_1 + p_2 x_2 + p_3 x_3 + p_4 x_4 + p_5 x_5 \\ &= (172.000x1) + (96.000x1) + (168.000x1) + (800.000x0) + \\ &\quad (350.000x0) \end{aligned}$$

$$= \text{Rp.436.000}$$

$$\begin{aligned}(N_{13}) &= p_1x_1 + p_2x_2 + p_3x_3 + p_4x_4 + p_5x_5 \\ &= (172.000x0) + (96.000x1) + (168.000x1) + (800.000x0) + \\ &\quad (350.000x0) \\ &= \text{Rp.264.000}\end{aligned}$$

$$\begin{aligned}(N_{21}) &= p_1x_1 + p_2x_2 + p_3x_3 + p_4x_4 + p_5x_5 \\ &= (172.000x0) + (96.000x0) + (168.000x1) + (800.000x1) + \\ &\quad (350.000x1) \\ &= \text{Rp.1.318.000}\end{aligned}$$

$$\begin{aligned}(N_{22}) &= p_1x_1 + p_2x_2 + p_3x_3 + p_4x_4 + p_5x_5 \\ &= (172.000x0) + (96.000x1) + (168.000x0) + (800.000x1) + \\ &\quad (350.000x1) \\ &= \text{Rp.1.264.000}\end{aligned}$$

$$\begin{aligned}(N_{23}) &= p_1x_1 + p_2x_2 + p_3x_3 + p_4x_4 + p_5x_5 \\ &= (172.000x1) + (96.000x1) + (168.000x1) + (800.000x0) + \\ &\quad (350.000x0) \\ &= \text{Rp.436.000}\end{aligned}$$

$$\begin{aligned}(N_{31}) &= p_1x_1 + p_2x_2 + p_3x_3 + p_4x_4 + p_5x_5 \\ &= (172.000x0) + (96.000x0) + (168.000x1) + (800.000x1) + \\ &\quad (350.000x0) \\ &= \text{Rp.968.000}\end{aligned}$$

$$\begin{aligned}(N_{32}) &= p_1x_1 + p_2x_2 + p_3x_3 + p_4x_4 + p_5x_5 \\ &= (172.000x0) + (96.000x1) + (168.000x0) + (800.000x1) + \\ &\quad (350.000x0) \\ &= \text{Rp.896.000}\end{aligned}$$

$$\begin{aligned}(N_{33}) &= p_1x_1 + p_2x_2 + p_3x_3 + p_4x_4 + p_5x_5 \\ &= (172.000x0) + (96.000x1) + (168.000x1) + (800.000x0) + \\ &\quad (350.000x0) \\ &= \text{Rp.264.000}\end{aligned}$$

$$(N_{41}) = p_1x_1 + p_2x_2 + p_3x_3 + p_4x_4 + p_5x_5$$

$$\begin{aligned}
 &= (172.000x0) + (96.000x1) + (168.0000x0) + (800.000x1) + \\
 &(350.000x1) \\
 &= \text{Rp.1.246.000}
 \end{aligned}$$

$$\begin{aligned}
 (N_{42}) &= p_1x_1 + p_2x_2 + p_3x_3 + p_4x_4 + p_5x_5 \\
 &= (172.000x1) + (96.000x0) + (168.0000x0) + (800.000x0) + \\
 &(350.000x1) \\
 &= \text{Rp.522.000}
 \end{aligned}$$

$$\begin{aligned}
 (N_{43}) &= p_1x_1 + p_2x_2 + p_3x_3 + p_4x_4 + p_5x_5 \\
 &= (172.000x0) + (96.000x1) + (168.0000x0) + (800.000x0) + \\
 &(350.000x1) \\
 &= \text{Rp.446.000}
 \end{aligned}$$

$$\begin{aligned}
 (N_{51}) &= p_1x_1 + p_2x_2 + p_3x_3 + p_4x_4 + p_5x_5 \\
 &= (172.000x0) + (96.000x1) + (168.0000x1) + (800.000x1) + \\
 &(350.000x1) \\
 &= \text{Rp.1.414.000}
 \end{aligned}$$

$$\begin{aligned}
 (N_{52}) &= p_1x_1 + p_2x_2 + p_3x_3 + p_4x_4 + p_5x_5 \\
 &= (172.000x1) + (96.000x1) + (168.0000x1) + (800.000x1) + \\
 &(350.000x0) \\
 &= \text{Rp.1.236.000}
 \end{aligned}$$

$$\begin{aligned}
 (N_{53}) &= p_1x_1 + p_2x_2 + p_3x_3 + p_4x_4 + p_5x_5 \\
 &= (172.000x0) + (96.000x1) + (168.0000x0) + (800.000x1) + \\
 &(350.000x0) \\
 &= \text{Rp.896.000}
 \end{aligned}$$

#### Lampiran E. Hasil 10 kali *running* setiap perbandingan parameter

##### a. Uji simulasi dengan populasi 21

Pengujian dengan populasi 21 dilakukan dengan parameter  $k = 3, 5, 7$  dan  $x = 1, 2, 3$   $CO = 3, 5, 7$  dan Stepzise = 0.1. dilakukan *running* program sebanyak 10 kali untuk mengetahui jumlah iterasi yang dibutuhkan.  $r.t = \text{running time}$ ,  $w = \text{berat}$ .

Parameter  $k = 3, x = 1, CO = 3$ , dan Stepzise = 0.1

No.	MBO				PeSOA			
	Iterasi ke-	r.t	profit	w	Iterasi ke-	r.t	profit	w
1	182	339,273	10.364.000	479	1079,000	339,194	10.364.000	479
2	182	342,473	10.364.000	479	1079,000	341,832	10.364.000	479
3	111	467,359	10.364.000	479	401,000	466,282	10.364.000	479
4	99	468,502	10.364.000	479	417,000	467,446	10.364.000	479
5	114	496,134	10.364.000	479	1112,000	495,039	10.364.000	479
6	97	536,387	10.364.000	479	193,000	535,224	10.364.000	479
7	198	842,131	10.364.000	479	383,000	840,675	10.364.000	479
8	121	607,296	10.364.000	479	270,000	605,063	10.364.000	479
9	187	606,293	10.364.000	479	137,000	604,999	10.364.000	479
10	63	606,231	10.364.000	479	433,000	605,984	10.364.000	479
<b>Rt2</b>	<b>130,2222</b>	<b>531</b>	<b>10.364.000</b>	<b>479</b>	<b>550,400</b>	<b>530,174</b>	<b>10.364.000</b>	<b>479</b>

Parameter  $k = 5, x = 2, CO = 5$ , dan Stepzise = 0.1

No.	MBO				PeSOA			
	Iterasi ke-	r.t	profit	w	Iterasi ke-	r.t	profit	w
1	47	497,583	10.364.000	479	507	499,936	10.364.000	479
2	73	528,761	10.364.000	479	232	537,111	10.364.000	479
3	60	489,190	10.364.000	479	616	493,095	10.364.000	479
4	61	464,964	10.364.000	479	768	468,390	10.364.000	479
5	55	465,464	10.364.000	479	333	467,100	10.364.000	479
6	44	476,411	10.364.000	479	605	479,832	10.364.000	479
7	37	464,041	10.364.000	479	475	466,863	10.364.000	479
8	40	480,984	10.364.000	479	781	485,434	10.364.000	479
9	164	522,381	10.364.000	479	160	527,498	10.364.000	479
10	50	543,828	10.364.000	479	741	547,676	10.364.000	479
<b>Rt2</b>	<b>63,100</b>	<b>493,361</b>	<b>10.364.000</b>	<b>479</b>	<b>521,800</b>	<b>497,294</b>	<b>10.364.000</b>	<b>479</b>

Parameter  $k = 7, x = 3, CO = 7$ , dan Stepzise = 0.1

No.	MBO				PeSOA			
	Iterasi ke-	r.t	profit	w	Iterasi ke-	r.t	profit	w
1	41,000	583,543	10.364.000	479	799,000	589,902	10.364.000	479

2	30,000	633,023	10.364.000	479	715,000	645,502	10.364.000	479
3	57,000	431,950	10.364.000	479	285,000	441,123	10.364.000	479
4	50,000	395,391	10.364.000	479	2028,000	401,307	10.364.000	479
5	94,000	545,192	10.364.000	479	358,000	554,738	10.364.000	479
6	45,000	513,132	10.364.000	479	1066,000	522,403	10.364.000	479
7	50,000	349,954	10.364.000	479	2018,000	354,345	10.364.000	479
8	94,000	470,016	10.364.000	479	358,000	477,059	10.364.000	479
9	50,000	340,407	10.364.000	479	2018,000	344,368	10.364.000	479
10	94,000	527,640	10.364.000	479	358,000	535,235	10.364.000	479
<b>Rt2</b>	<b>60,500</b>	<b>479,024</b>	<b>10.364.000</b>	<b>479</b>	<b>1000,300</b>	<b>486,598</b>	<b>10.364.000</b>	<b>479</b>

b. Uji simulasi dengan populasi 31

Pengujian dengan populasi 31 dilakukan dengan parameter  $k = 3, 5, 7$  dan  $x = 1, 2, 3$   $CO = 3, 5, 7$  dan Stepzise = 0.1. dilakukan running program sebanyak 10 kali untuk mengetahui jumlah iterasi yang dibutuhkan.  $r.t = \text{running time}$ ,  $w = \text{berat}$ .

Parameter  $k = 3$ ,  $x = 1$ ,  $CO = 3$ , dan Stepzise = 0.1

No.	MBO				PeSOA			
	Iterasi ke-	r.t	profit	w	Iterasi ke-	r.t	profit	w
1	141,000	461,851	10.364.000	479	777,000	462,388	10.364.000	479
2	60,000	458,437	10.364.000	479	572,000	459,263	10.364.000	479
3	50,000	464,030	10.364.000	479	750,000	465,136	10.364.000	479
4	103,000	461,605	10.364.000	479	356,000	461,550	10.364.000	479
5	93,000	466,055	10.364.000	479	601,000	466,695	10.364.000	479
6	59,000	526,285	10.364.000	479	596,000	528,202	10.364.000	479
7	164,000	525,951	10.364.000	479	347,000	526,674	10.364.000	479
8	68,000	474,442	10.364.000	479	537,000	474,435	10.364.000	479
9	103,000	29,241	10.364.000	479	136,000	29,221	10.364.000	479
10	101,000	44,654	10.364.000	479	192,000	44,605	10.364.000	479
<b>Rt2</b>	<b>94,200</b>	<b>391,255</b>	<b>10.364.000</b>	<b>479</b>	<b>486,400</b>	<b>391,817</b>	<b>10.364.000</b>	<b>479</b>



Parameter  $k = 5, x = 2, CO = 5$ , dan Stepzise = 0.1

No.	MBO				PeSOA			
	Iterasi ke-	r.t	profit	w	Iterasi ke-	r.t	profit	w
1	35,000	473,228	10.364.000	479	359,000	479,152	10.364.000	479
2	66,000	469,442	10.364.000	479	561,000	474,906	10.364.000	479
3	36,000	453,809	10.364.000	479	580,000	458,399	10.364.000	479
4	66,000	532,265	10.364.000	479	108,000	538,661	10.364.000	479
5	69,000	482,945	10.364.000	479	210,000	490,547	10.364.000	479
6	73,000	423,265	10.364.000	479	708,000	429,972	10.364.000	479
7	63,000	453,161	10.364.000	479	1339,000	460,844	10.364.000	479
8	101,000	427,180	10.364.000	479	468,000	432,854	10.364.000	479
9	57,000	434,071	10.364.000	479	1088,000	439,991	10.364.000	479
10	51,000	467,609	10.364.000	479	432,000	474,343	10.364.000	479
<b>Rt2</b>	<b>61,700</b>	<b>461,698</b>	<b>10.364.000</b>	<b>479</b>	<b>585,300</b>	<b>467,967</b>	<b>10.364.000</b>	<b>479</b>

Parameter  $k = 7, x = 3, CO = 7$ , dan Stepzise = 0.1

No.	MBO				PeSOA			
	Iterasi ke-	r.t	profit	w	Iterasi ke-	r.t	profit	w
1	45,000	376,885	10.364.000	479	626,000	387,114	10.364.000	479
2	88,000	402,279	10.364.000	479	208,000	417,270	10.364.000	479
3	66,000	456,256	10.364.000	479	612,000	467,501	10.364.000	479
4	66,000	456,256	10.364.000	479	612,000	467,501	10.364.000	479
5	22,000	485,917	10.364.000	479	546,000	497,956	10.364.000	479
6	22,000	486,186	10.364.000	479	280,000	498,113	10.364.000	479
7	118,000	485,911	10.364.000	479	903,000	497,438	10.364.000	479
8	35,000	557,937	10.364.000	479	544,000	571,296	10.364.000	479
9	37,000	558,199	10.364.000	479	645,000	571,059	10.364.000	479
10	36,000	485,808	10.364.000	479	1038,000	500,262	10.364.000	479
<b>Rt2</b>	<b>53,500</b>	<b>475,163</b>	<b>10.364.000</b>	<b>479</b>	<b>601,400</b>	<b>487,551</b>	<b>10.364.000</b>	<b>479</b>

## c. Uji simulasi dengan populasi 41

Pengujian dengan populasi 41 dilakukan dengan parameter  $k = 3, 5, 7$  dan  $x = 1, 2, 3$   $CO = 3, 5, 7$  dan Stepzise = 0.1. dilakukan running program sebanyak 10 kali untuk mengetahui jumlah iterasi yang dibutuhkan. r.t = *running time*, w = berat

Parameter  $k = 3, x = 1, CO = 3$ , dan Stepzise = 0.1

No.	MBO				PeSOA			
	Iterasi ke-	r.t	profit	w	Iterasi ke-	r.t	profit	w
1	75,000	529,876	10.364.000	479	97,000	532,199	10.364.000	479
2	81,000	843,697	10.364.000	479	994,000	844,206	10.364.000	479
3	77,000	859,234	10.364.000	479	552,000	857,995	10.364.000	479
4	75,000	876,370	10.364.000	479	130,000	876,184	10.364.000	479
5	65,000	871,743	10.364.000	479	218,000	868,723	10.364.000	479
6	72,000	876,853	10.364.000	479	147,000	877,146	10.364.000	479
7	65,000	217,195	10.364.000	479	576,000	217,147	10.364.000	479
8	52,000	532,023	10.364.000	479	166,000	539,547	10.364.000	479
9	80,000	968,928	10.364.000	479	678,000	977,622	10.364.000	479
10	38,000	495,046	10.364.000	479	172,000	497,217	10.364.000	479
<b>Rt2</b>	68,000	707,096	<b>10.364.000</b>	<b>479</b>	373,000	708,799	<b>10.364.000</b>	<b>479</b>

Parameter  $k = 5, x = 2, CO = 5$ , dan Stepzise = 0.1

No.	MBO				PeSOA			
	Iterasi ke-	r.t	profit	w	Iterasi ke-	r.t	profit	w
1	37,000	544,397	10.364.000	479	500,000	551,769	10.364.000	479
2	50,000	505,113	10.364.000	479	448,000	512,848	10.364.000	479
3	60,000	423,250	10.364.000	479	658,000	438,352	10.364.000	479
4	34,000	565,134	10.364.000	479	225,000	576,800	10.364.000	479
5	41,000	519,469	10.364.000	479	1266,000	532,073	10.364.000	479
6	79,000	487,077	10.364.000	479	424,000	497,192	10.364.000	479
7	49,000	488,071	10.364.000	479	530,000	500,146	10.364.000	479
8	51,000	491,007	10.364.000	479	54,000	499,971	10.364.000	479

9	30,000	487,700	10.364.000	479	308,000	496,210	10.364.000	479
10	56,000	489,335	10.364.000	479	436,000	498,681	10.364.000	479
<b>Rt2</b>	<b>48,700</b>	<b>500,055</b>	<b>10.364.000</b>	<b>479</b>	<b>484,900</b>	<b>510,404</b>	<b>10.364.000</b>	<b>479</b>

Parameter  $k = 7, x = 3, CO = 7$ , dan Stepzise = 0.1

No.	MBO				PeSOA			
	Iterasi ke-	r.t	Profit	w	Iterasi ke-	r.t	profit	w
1	55,000	487,554	10.364.000	479	55,000	487,554	10.364.000	479
2	51,000	536,811	10.364.000	479	51,000	536,811	10.364.000	479
3	30,000	548,532	10.364.000	479	30,000	548,532	10.364.000	479
4	62,000	489,398	10.364.000	479	62,000	489,398	10.364.000	479
5	35,000	488,212	10.364.000	479	35,000	488,212	10.364.000	479
6	33,000	491,049	10.364.000	479	33,000	491,049	10.364.000	479
7	54,000	539,586	10.364.000	479	54,000	539,586	10.364.000	479
8	47,000	513,580	10.364.000	479	47,000	513,580	10.364.000	479
9	21,000	512,177	10.364.000	479	21,000	512,177	10.364.000	479
10	48,000	543,148	10.364.000	479	48,000	543,148	10.364.000	479
<b>Rt2</b>	<b>43,600</b>	<b>515,005</b>	<b>10.364.000</b>	<b>479</b>	<b>43,600</b>	<b>515,005</b>	<b>10.364.000</b>	<b>479</b>

## LAMPIRAN F.Skrip Program

```
tic;
%Data
data001 = get(handles.uitable1, 'UserData');
data=data001{1};
dataWi=data(:,1)';
dataPi=data(:,2)';
NamaBarang=data001{2};
Kapasitas=str2num(get(handles.edit7, 'string'));
Dim=size(data,1);

%Parameter
Pop=str2num(get(handles.edit1, 'string'));
kval=str2num(get(handles.edit2, 'string'));
xval=str2num(get(handles.edit3, 'string'));
NO2=str2num(get(handles.edit4, 'string'));
StepSize=str2num(get(handles.edit5, 'string'));
MaxIter=str2num(get(handles.edit6, 'string'));
```

```

%Inisialization
Swarm=rand(Pop,Dim);
PeSOA=round(Swarm);
for i=1:Pop
    BeratPeSOA(i)=sum(PeSOA(i,:).*dataWi);
    while BeratPeSOA(i)>Kapasitas
        pss=find(PeSOA(i,:)==1);
        r1=ceil(rand*length(pss));
        Swarm(i,pss(r1))=Swarm(i,pss(r1))/2;
        PeSOA(i,:)=round(Swarm(i,:));
        BeratPeSOA(i)=sum(PeSOA(i,:).*dataWi);
    end
    FitnessPeSOA(i)=sum(PeSOA(i,:).*dataPi);
end

MBO=PeSOA;
BeratMBO=BeratPeSOA;
FitnessMBO=FitnessPeSOA;

%Best Swarm
best=find(FitnessPeSOA==max(FitnessPeSOA));
SwarmBest=Swarm(best(1),:);
FitnessBest=max(FitnessPeSOA);

bsfMBO(1)=max(FitnessMBO); inon1=0;
bsfPeSOA(1)=FitnessBest; inon2=0;

%Waktu Komputasi
rt=toc;
rtMBO=rt;
rtPeSOA=rt;

t=0;
while t<MaxIter
    %%MBO
    tic;
    for i=1:Pop
        if i==1
            for j=1:kval
                Ni(j,:,i)=MBO(i,:);
                in0=find(Ni(j,:,i)==0);
                pr0=ceil(rand*length(in0));
                Ni(j,in0(pr0),i)=1;
                BeratN(i,j)=sum(Ni(j,:,i).*dataWi);
                while BeratN(i,j)>Kapasitas
                    in1=find(Ni(j,:,i)==1);
                    pr1=ceil(rand*length(in1));
                    Ni(j,in1(pr1),i)=0;
                    BeratN(i,j)=sum(Ni(j,:,i).*dataWi);
                end
                FitN(i,j)=sum(Ni(j,:,i).*dataPi);
            end
            [sortNi,indek]=sort(FitN(i,:), 'descend');
            Ni(:, :, i)=Ni(indek, :, i);
            BeratN(i, :)=BeratN(i, indeks);
            FitN(i, :)=sortNi;
            if FitN(i,1)>FitnessMBO(i)

```

```

        MBO(i,:) = Ni(1,:,i);
        BeratMBO(i) = BeratN(i,1);
        FitnessMBO(i) = FitN(i,1);
    end
    for k=1:xval
        Share1(k,:) = Ni(2*k,:,i);
        Share2(k,:) = Ni(2*k+1,:,i);
    end
elseif mod(i,2)==0
    for k=1:xval
        Ni(k,:,i) = Share1(k,:);
        BeratN(i,k) = sum(Ni(k,:,i).*dataWi);
        FitN(i,k) = sum(Ni(k,:,i).*dataPi);
    end
    for j=xval+1:kval
        Ni(j,:,i) = MBO(i,:);
        in0 = find(Ni(j,:,i)==0);
        pr0 = ceil(rand*length(in0));
        Ni(j,in0(pr0),i) = 1;
        BeratN(i,j) = sum(Ni(j,:,i).*dataWi);
        while BeratN(i,j) > Kapasitas
            in1 = find(Ni(j,:,i)==1);
            pr1 = ceil(rand*length(in1));
            Ni(j,in1(pr1),i) = 0;
            BeratN(i,j) = sum(Ni(j,:,i).*dataWi);
        end
        FitN(i,j) = sum(Ni(j,:,i).*dataPi);
    end
    [sortNi,indek] = sort(FitN(i,:), 'descend');
    Ni(:, :, i) = Ni(indek, :, i);
    BeratN(i, :) = BeratN(i, indek);
    FitN(i, :) = sortNi;
    if FitN(i,1) > FitnessMBO(i)
        MBO(i,:) = Ni(1,:,i);
        BeratMBO(i) = BeratN(i,1);
        FitnessMBO(i) = FitN(i,1);
    end
    for k=1:xval
        Share1(k,:) = Ni(k+1,:,i);
    end
else
    for k=1:xval
        Ni(k,:,i) = Share2(k,:);
        BeratN(i,k) = sum(Ni(k,:,i).*dataWi);
        FitN(i,k) = sum(Ni(k,:,i).*dataWi);
    end
    for j=xval+1:kval
        Ni(j,:,i) = MBO(i,:);
        in0 = find(Ni(j,:,i)==0);
        pr0 = ceil(rand*length(in0));
        Ni(j,in0(pr0),i) = 1;
        BeratN(i,j) = sum(Ni(j,:,i).*dataWi);
        while BeratN(i,j) > Kapasitas
            in1 = find(Ni(j,:,i)==1);
            pr1 = ceil(rand*length(in1));
            Ni(j,in1(pr1),i) = 0;
            BeratN(i,j) = sum(Ni(j,:,i).*dataWi);
        end
        FitN(i,j) = sum(Ni(j,:,i).*dataPi);
    end
end

```

```

end
[sortNi,indek]=sort (FitN(i,:), 'descend');
Ni(:, :, i)=Ni (indek, :, i);
BeratN(i, :)=BeratN(i, indek);
FitN(i, :)=sortNi;
if FitN(i,1)>FitnessMBO(i)
    MBO(i, :)=Ni (1, :, i);
    BeratMBO(i)=BeratN(i, 1);
    FitnessMBO(i)=FitN(i, 1);
end
for k=1:xval
    Share2(k, :)=Ni (k+1, :, i);
end
end
end
bsfMBO(t+2)=max (FitnessMBO);
if bsfMBO(t+2)~=bsfMBO(t+1)
    inon1=t+1;
end
rtMBO=rtMBO+toc;

%%PeSOA
tic;
%Diving
for i=1:Pop
    O2=NO2; lev=1;
    Div(lev, :)=Swarm(i, :);
    while O2>0

Div(lev+1, :)=Div(lev, :)+StepSize*(rand(1, Dim)).*(abs(SwarmBest-
Swarm(i, :))+1);
    if min(Div(lev+1, :))<0
        Div(lev+1, :)=(Div(lev+1, :)-
min(Div(lev+1, :)))/(max(Div(lev+1, :))-min(Div(lev+1, :)));
    elseif max(Div(lev+1, :))>1
        Div(lev+1, :)=Div(lev+1, :)/max(Div(lev+1, :));
    end
    PeSOAD(lev, :)=round(Div(lev+1, :));
    BeratPeSOAD(lev)=sum (PeSOAD(lev, :).*dataWi);
    while BeratPeSOAD(lev)>Kapasitas
        pss=find (PeSOAD(lev, :)==1);
        r1=ceil (rand*length (pss));
        Div(lev+1, pss(r1))=Div(lev+1, pss(r1))/2;
        PeSOAD(lev, :)=round (Div(lev+1, :));
        BeratPeSOAD(lev)=sum (PeSOAD(lev, :).*dataWi);
    end
    FitnessD(lev)=sum (PeSOAD(lev, :).*dataPi);
    O2=O2-1; lev=lev+1;
end
if max (FitnessD)>FitnessPeSOA(i)
    bestD=find (FitnessD==max (FitnessD));
    Swarm(i, :)=Div (bestD(1)+1, :);
    FitnessPeSOA(i)=max (FitnessD);
end
end
end
%Best Swarm
if max (FitnessPeSOA)>FitnessBest
    best=find (FitnessPeSOA==max (FitnessPeSOA));
    SwarmBest=Swarm (best(1), :);
end

```



```

        FitnessBest=max(FitnessPeSOA);
    end
    bsfPeSOA(t+2)=FitnessBest;
    if bsfPeSOA(t+2)~=bsfPeSOA(t+1)
        inon2=t+1;
    end
    rtPeSOA=rtPeSOA+toc;

    %%Plot
    tic;
    plot(0:length(bsfMBO)-1,bsfMBO,'r',0:length(bsfPeSOA)-
1,bsfPeSOA,'b','LineWidth',2);

    line(inon1,bsfMBO(inon1+1),'Marker','s','MarkerEdgeColor','k','Mar
kerFaceColor','y','MarkerSize',6);

    line(inon2,bsfPeSOA(inon2+1),'Marker','s','MarkerEdgeColor','k','M
arkerFaceColor','y','MarkerSize',6);
    xlval=get(handles.axes1,'XLim');
    ylval=get(handles.axes1,'YLim');
    text(xlval(2)-xlval(2)/25,(ylval(2)-
ylval(1))*0.1+ylval(1),sprintf(['IterMBO: ' num2str(inon1) '\n'
'FitMBO: ' num2str(bsfMBO(inon1+1)) ...
'\n' 'IterPeSOA: ' num2str(inon2) '\n' 'FitPeSOA: '
num2str(bsfPeSOA(inon2+1))]),...

'HorizontalAlignment','right','VerticalAlignment','bottom','FontSi
ze',9,'FontWeight','bold',...
'BackgroundColor','y','EdgeColor','k');
    xlabel('Iterasi'); ylabel('Total Profit (Fitness)');
    set(handles.axes1,'FontSize',9,'FontWeight','bold');
    pause(0.00001);
    rt=toc;
    rtMBO=rtMBO+rt;
    rtPeSOA=rtPeSOA+rt;

    %Next iter
    t=t+1;
end
legend('MBO','PeSOA','Location','East');
set(handles.axes1,'UserData',[inon1,inon2]);
%Solusi Akhir
best1=find(FitnessMBO==max(FitnessMBO));
Solusi1=MBO(best1(1),:);
Berat1=BeratMBO(best1(1));
Solusi2=round(SwarmBest);
Berat2=sum(Solusi2.*dataWi);
for i=1:Dim
    colname{i}=num2str(i);
    colwidth{i}=30;
end
colname{Dim+1}='Berat'; colname{Dim+2}='T.Profit';
colwidth{Dim+1}=60; colwidth{Dim+2}=80;
set(handles.uitable2,'data',[Solusi1,Berat1,max(FitnessMBO);Solusi
2,Berat2,FitnessBest],...
'userdata',[Solusi1;Solusi2],...

```

```
'rowname',{'MBO';'PeSOA'},'columnname',colname,'columnwidth',colwidth);
```

```
set(handles.text9,'string',[num2str(rtMBO) ' detik']);  
set(handles.text11,'string',[num2str(rtPeSOA) ' detik']);
```

