



**PENERAPAN ALGORITMA *ELEPHANT HERDING*  
*OPTIMIZATION* (EHO) PADA PERMASALAHAN *MULTIPLE*  
*CONSTRAINTS KNAPSACK 0-1***

**SKRIPSI**

Oleh

**Yulia Dewi Regita  
NIM 151810101010**

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



**PENERAPAN ALGORITMA *ELEPHANT HERDING*  
OPTIMIZATION (EHO) PADA PERMASALAHAN *MULTIPLE*  
*CONSTRAINTS KNAPSACK 0-1***

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat  
untuk menyelesaikan Program Studi Matematika (S1)  
dan mencapai gelar Sarjana Sains

Oleh

**Yulia Dewi Regita  
NIM 151810101010**

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

## PERSEMBAHAN

Puji syukur dengan menyebut nama Allah SWT yang Maha Pengasih dan Maha Penyayang serta Sholawat atas Nabi Muhammad SAW sehingga terbentuklah skripsi ini dan saya persembahkan untuk:

1. kedua orang tua saya yang tercinta yaitu Bapak Suwono dan Ibu Sumainah, kedua kakak saya Marta Murtisari dan Maria Fitriana serta seluruh keluarga yang telah memberikan doa, semangat dan perhatian terhadap saya;
2. seluruh dosen dan guru yang telah membimbing dan memberikan ilmunya kepada saya sejak sekolah dasar hingga perguruan tinggi dengan penuh kesabaran dan kasih sayang;
3. Almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember, SMA Negeri 1 Rogojampi, SMP Negeri 3 Rogojampi dan SD Negeri 1 Lemahbang Dewo;
4. teman-teman SIGMA 15 HIMATIKA Geokompstat yang selalu memberikan dukungan, bantuan dan doa;
5. teman-teman kuliah BEHAMBLO yaitu Ayu, Indy, Nanda dan Tika serta Kosan Melati yaitu Sofi dan Ingka yang telah memberikan dukungan;
6. teman-teman masa sekolah Linda dan Nuri yang telah memberikan dukungan;
7. teman-teman SEGIENAM yaitu Aulia, Wulan, Ria, Ovi dan Winda yang telah memberikan dukungan;
8. teman pejuang riset yaitu Vidiyanti Lestari yang selalu memberikan semangat dan bantuan.

**MOTO**

“Lakukanlah yang terbaik, sehingga aku tak akan menyalahkan diriku sendiri atas segalanya”

(Magdalena Neuner)

“Segala sesuatu tidak bisa terselesaikan secara instan, jika ingin lebih cepat segeralah bertindak sisanya serahkan kepada Sang Pencipta”



**PERNYATAAN**

Saya yang bertanda tangan dibawah ini:

Nama : Yulia Dewi Regita

NIM : 151810101010

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul “Penerapan Algoritma *Elephant Herding Optimization* (EHO) pada Permasalahan *Multiple Constraints Knapsack 0-1*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggungjawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, Mei 2019

Yang menyatakan,

Yulia Dewi Regita  
NIM 151810101010

**SKRIPSI**

**PENERAPAN ALGORITMA *ELEPHANT HERDING OPTIMIZATION*  
(EHO) PADA PERMASALAHAN *MULTIPLE CONSTRAINTS*  
*KNAPSACK 0-1***

Oleh

Yulia Dewi Regita  
NIM 151810101010

Pembimbing

Dosen Pembimbing Utama : Dr. Kiswara Agung Santoso, S.Si., M.Kom

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

**PENGESAHAN**

Skripsi berjudul “Penerapan Algoritma *Elephant Herding Optimization* (EHO) pada Permasalahan *Multiple Constraints Knapsack 0-1*” karya Yulia Dewi Regita telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas  
Jember

Tim Penguji,

Ketua,

Anggota I,

Dr. Kiswara A. Santoso, S.Si., M.Kom.  
NIP. 197209071998031003

Ahmad Kamsyakawuni, S.Si., M.Kom.  
NIP. 197211291998021001

Anggota II,

Anggota III,

Kusbudiono, S.Si., M.Si.  
NIP. 197704302005011001

Dr. Mohamat Fatekurohman, S.Si., M.Si.  
NIP. 196906061998031001

Mengesahkan,

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Jember

Drs. Sujito, Ph.D.  
NIP. 196102041987111001

## RINGKASAN

**Penerapan Algoritma *Elephant Herding Optimization* (EHO) pada Permasalahan *Multiple Constraints Knapsack 0-1***; Yulia Dewi Regita, 151810101010; 2019; 66 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Permasalahan optimasi sering dijumpai dalam pekerjaan sehari-hari, seperti penentuan pemilihan barang pada masalah *knapsack*. *Knapsack* merupakan suatu kantong atau tempat penyimpanan untuk memuat suatu objek yang sifatnya terbatas. Media penyimpanan tersebut dapat menyimpan objek dengan ketentuan total ukuran objek lebih kecil atau sama dengan ukuran kapasitasnya.

Penelitian ini menyelesaikan variasi permasalahan *knapsack* yaitu *multiple constraints knapsack 0-1* menggunakan algoritma *Elephant Herding Optimization* (EHO). Tujuan yang diharapkan yaitu untuk mendapatkan solusi optimum dari algoritma EHO dan membandingkannya dengan metode simpleks menggunakan *Solver Add-In* pada *Microsoft Excel*. Data yang digunakan untuk penelitian ada 2, data pertama diambil dari Toko Citra Tani terletak di Jl. PB Sudirman No.74 di Kecamatan Panti Kabupaten Jember dan data kedua diambil dari penelitian Putri, R. J. M. (2017). Terdapat tiga kendala yang digunakan yaitu kapasitas *knapsack*, volume *knapsack* dan modal.

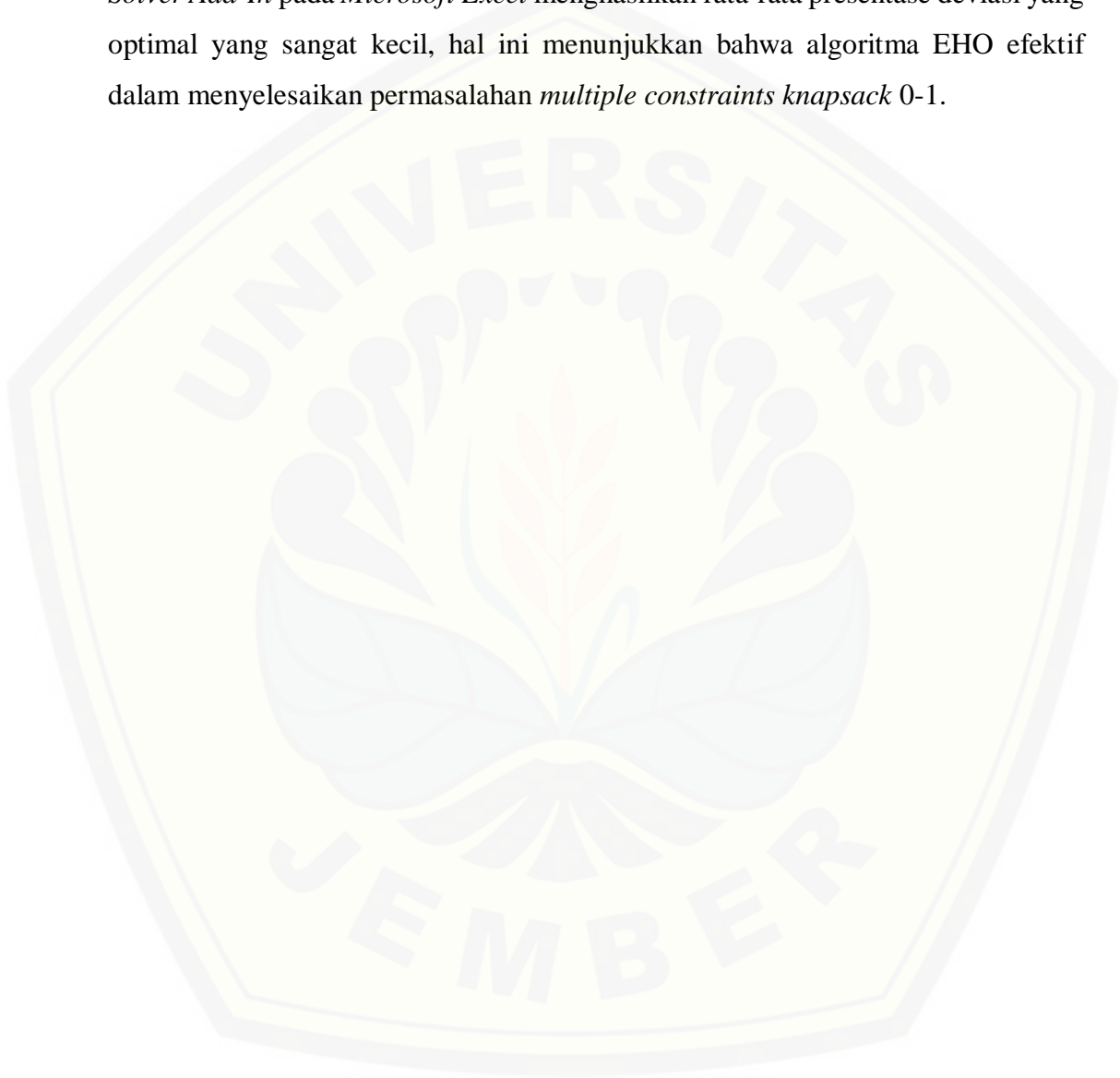
Berdasarkan hasil uji pengaruh parameter terhadap kedua data diketahui jika semakin besar nilai parameter  $nClan$ ,  $nCi$ ,  $\alpha$  dan  $\beta$  maka profit yang diperoleh akan semakin optimal. Semakin besar nilai *MaxGen* belum tentu menghasilkan profit semakin optimal. Hal ini menjadikan parameter berpengaruh terhadap profit yang dihasilkan.

Hasil percobaan akhir algoritma EHO pada data 1 mendapatkan profit terbaik yaitu sebesar Rp. 2.714.000,- dan profit terburuk yaitu sebesar Rp. 2.714.000,-. Pada data 2 mendapatkan profit terbaik yaitu sebesar Rp. 14.291.000,- dan profit terburuk yaitu sebesar Rp. 14.271.000,-. Nilai parameter yang digunakan



pada data 1 yaitu  $nClan = 25$ ;  $nCi = 25$ ,  $\alpha = 0,9$ ;  $\beta = 0,9$  dan  $MaxGen = 100$ , sedangkan pada data 2 yaitu  $nClan = 50$ ;  $nCi = 50$ ,  $\alpha = 0,9$ ;  $\beta = 0,9$  dan  $MaxGen = 1000$ .

Hasil perbandingan algoritma EHO dengan metode simpleks menggunakan *Solver Add-In* pada *Microsoft Excel* menghasilkan rata-rata presentase deviasi yang optimal yang sangat kecil, hal ini menunjukkan bahwa algoritma EHO efektif dalam menyelesaikan permasalahan *multiple constraints knapsack 0-1*.



## PRAKATA

Puji syukur dengan menyebut nama Allah SWT atas segala rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penerapan Algoritma *Elephant Herding Optimization* (EHO) pada Permasalahan *Multiple Constraints Knapsack 0-1*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusun skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Dr. Kiswara Agung Santoso, S.Si., M.Kom. selaku Dosen Pembimbing Utama dan Ahmad Kamsyakawuni, S.Si., M.Kom. selaku Dosen Pembimbing Anggota serta Dosen Pembimbing Akademik yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
2. Kusbudiono, S.Si., M.Si. dan Dr. Mohamat Fatekurohman, S.Si., M.Si. selaku Dosen Penguji yang telah memberikan kritik dan saran yang membangun demi kesempurnaan skripsi ini;
3. seluruh Dosen dan Karyawan Jurusan Matematika Fakultas MIPA Universitas Jember;
4. semua pihak yang tidak dapat disebutkan satu per satu.

Penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, Mei 2019

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL .....	i
HALAMAN PERSEMBAHAN.....	ii
HALAMAN MOTO .....	iii
HALAMAN PERNYATAAN .....	iv
HALAMAN PEMBIMBINGAN .....	v
HALAMAN PENGESAHAN .....	vi
RINGKASAN .....	vii
PRAKATA .....	ix
DAFTAR ISI.....	x
DAFTAR TABEL .....	xii
DAFTAR GAMBAR.....	xiii
<b>BAB 1. PENDAHULUAN.....</b>	<b>1</b>
<b>1.1 Latar Belakang .....</b>	<b>1</b>
<b>1.2 Rumusan Masalah .....</b>	<b>2</b>
<b>1.3 Tujuan .....</b>	<b>3</b>
<b>1.4 Manfaat .....</b>	<b>3</b>
<b>BAB 2. TINJAUAN PUSTAKA .....</b>	<b>4</b>
<b>2.1 Optimasi .....</b>	<b>4</b>
<b>2.2 Permasalahan <i>Knapsack (Knapsack Problem)</i> .....</b>	<b>4</b>
2.2.1 <i>Knapsack 0-1 Problem</i> .....	6
2.2.2 <i>Multiple Constraints Knapsack 0-1</i> .....	7
<b>2.4 Algoritma <i>Elephant Herding Optimization (EHO)</i> .....</b>	<b>8</b>
<b>2.4 Metode Simpleks.....</b>	<b>11</b>
<b>2.5 Simpleks pada <i>Solver Add-In Microsoft Excel</i>.....</b>	<b>13</b>
<b>BAB 3. METODE PENELITIAN.....</b>	<b>15</b>
<b>3.1 Data Penelitian.....</b>	<b>15</b>
<b>3.2 Langkah-langkah Penelitian .....</b>	<b>15</b>

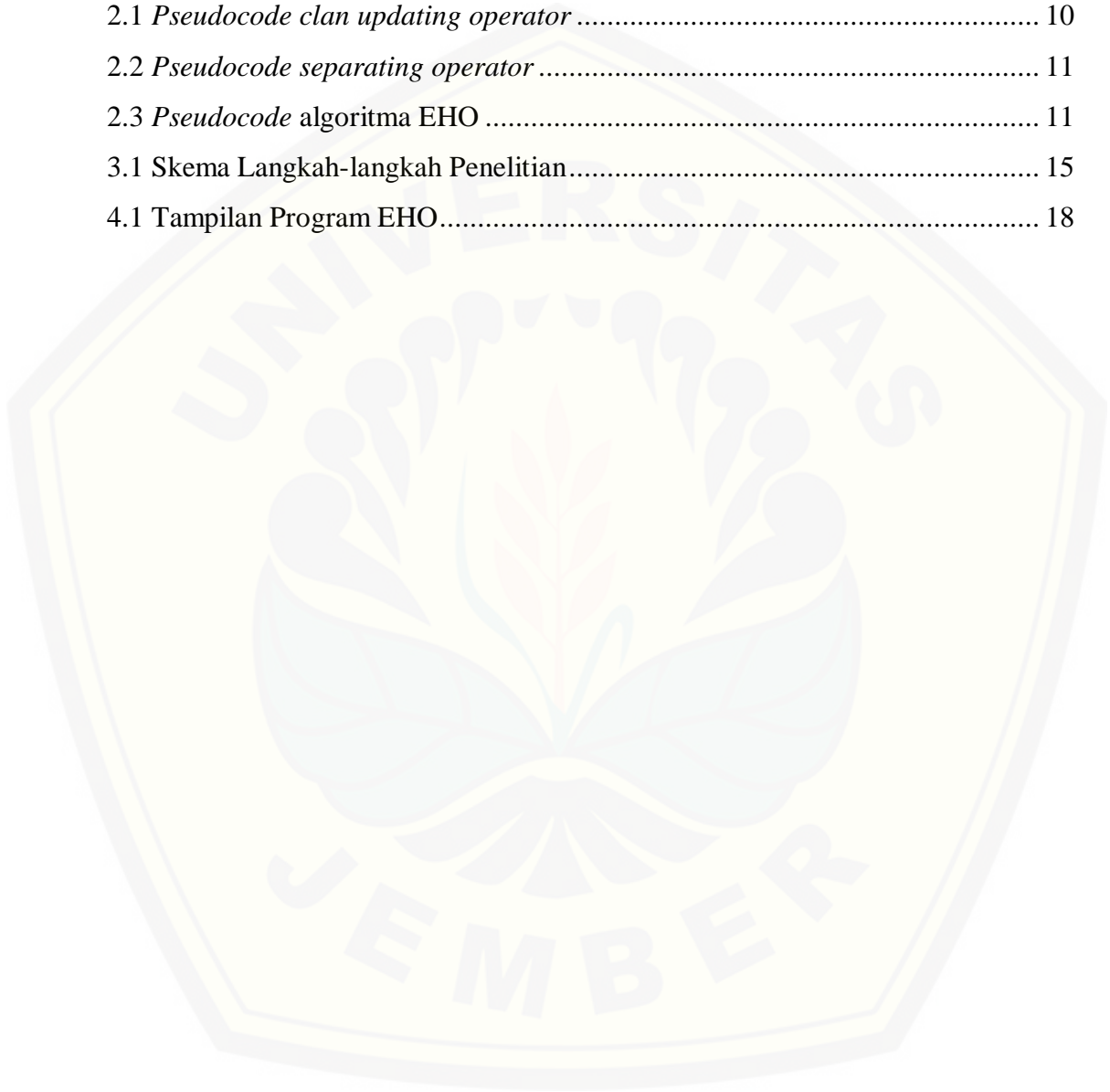
<b>BAB 4. HASIL DAN PEMBAHASAN</b> .....	18
<b>4.1 Hasil Penelitian</b> .....	18
4.1.1 Hasil Program .....	18
4.1.2 Simulasi Program.....	19
<b>4.2 Pembahasan</b> .....	24
4.2.1 Perhitungan Manual .....	24
4.2.2 <i>Pseudocode</i> Penerapan Algoritma EHO .....	34
<b>BAB 5. KESIMPULAN DAN SARAN</b> .....	42
<b>5.1 Kesimpulan</b> .....	42
<b>5.2 Saran</b> .....	42
<b>DAFTAR PUSTAKA</b> .....	43
<b>LAMPIRAN</b> .....	45

**DAFTAR TABEL**

	Halaman
4.1 Hasil uji parameter $nClan$ data 1 .....	19
4.2 Hasil uji parameter $nClan$ data 2 .....	20
4.3 Hasil uji parameter $nCi$ data 1 .....	20
4.4 Hasil uji parameter $nCi$ data 2 .....	20
4.5 Hasil uji parameter $\alpha$ dan $\beta$ data 1 .....	21
4.6 Hasil uji parameter $\alpha$ dan $\beta$ data 2 .....	21
4.7 Hasil simulasi akhir data 1 .....	22
4.8 Hasil simulasi akhir data 2 .....	22
4.9 Data Identifikasi Barang di Toko Citra Tani .....	24

**DAFTAR GAMBAR**

	Halaman
2.1 <i>Pseudocode clan updating operator</i> .....	10
2.2 <i>Pseudocode separating operator</i> .....	11
2.3 <i>Pseudocode algoritma EHO</i> .....	11
3.1 Skema Langkah-langkah Penelitian.....	15
4.1 Tampilan Program EHO.....	18



## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang Masalah

Masalah optimasi merupakan hal yang sering dijumpai dalam pekerjaan sehari-hari, seperti penentuan pemilihan barang pada permasalahan *knapsack*. Permasalahan *knapsack* merupakan permasalahan yang berhubungan dengan penyimpanan objek ke dalam media penyimpanan yang bersifat terbatas. *Knapsack* merupakan suatu kantong atau tempat yang digunakan untuk memuat sesuatu objek. Kantong atau tempat tersebut hanya dapat menyimpan beberapa objek saja dengan ketentuan total ukuran objek tersebut lebih kecil atau sama dengan ukuran kapasitasnya.

Permasalahan *knapsack* dibagi menjadi tiga jenis, yaitu 0-1 *knapsack problem*, *bounded knapsack problem* dan *unbounded knapsack problem*. Permasalahan 0-1 *knapsack* adalah menentukan pengambilan barang untuk dimuat seluruhnya atau tidak sama sekali. *Bounded knapsack problem* adalah menentukan pengambilan sebagian atau semua objek dengan jumlah objek terbatas. Sedangkan *unbounded knapsack problem* adalah menentukan pengambilan sebagian atau semua objek dengan jumlah objek tidak terbatas.

*Knapsack* memiliki beberapa variasi, yaitu *knapsack* tujuan ganda (*multi objective knapsack*), *knapsack* beberapa kendala (*multiple constraints* atau *multidimensional knapsack*), *multi knapsack* dan *quadratic knapsack*. Adanya suatu permasalahan yang kompleks dihadapkan pada lebih dari satu kendala ketika memilih objek, hal ini biasa disebut dengan permasalahan *knapsack* beberapa kendala atau *Multiple Constraints Knapsack Problem*. Permasalahan *multiple constraints knapsack* memiliki kendala lebih dari 1 bertujuan untuk memaksimalkan keuntungan yang diperoleh.

Penelitian sebelumnya tentang *knapsack*, dilakukan oleh Hilviah (2015) mengkaji masalah *multiple constraints knapsack* 0-1 diselesaikan dengan menggunakan algoritma *Dynamic Programming* dan algoritma *Backtracking*. Pada penelitiannya disimpulkan bahwa algoritma *Dynamic Programming* lebih efektif

dan efisien dibandingkan dengan algoritma *Backtracking*. Hal ini berdasarkan dari proses *running time* algoritma *Dynamic Programming* membutuhkan waktu lebih singkat dibandingkan dengan algoritma *Backtracking*, namun keduanya memiliki total keuntungan yang sama.

Selain algoritma di atas, terdapat algoritma lain yang dapat diterapkan pada permasalahan optimasi yaitu algoritma *Elephant Herding Optimization*. *Elephant Herding Optimization* (EHO) diusulkan oleh Wang *et al.* (2015) terinspirasi oleh perilaku menggiring kelompok gajah. Untuk menunjukkan keefektifannya, EHO diterapkan untuk menyelesaikan optimasi fungsi. Hasil dari EHO dibandingkan dengan BBO (*Biogeography-based Optimization*), DE (*Differential Evolution*) dan GA (*Genetic Algorithm*). Berdasarkan perbandingan menunjukkan bahwa EHO dapat menemukan solusi yang jauh lebih baik dibandingkan ketiga algoritma metaheuristik tersebut, EHO memiliki konvergensi lebih tinggi.

Berdasarkan uraian di atas, penulis tertarik untuk menerapkan algoritma EHO pada penyelesaian permasalahan *multiple constraints knapsack* 0-1. Untuk mengetahui keefektifan EHO maka perlu adanya perbandingan. Metode simpleks merupakan metode dasar yang digunakan untuk menyelesaikan permasalahan program linier termasuk *multiple constraints knapsack* 0-1. Dengan demikian, pada penelitian ini hasil dari algoritma EHO akan dibandingkan dengan hasil metode simpleks menggunakan *Solver Add-In* pada *Microsoft Excel*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan maka rumusan masalah yang akan dibahas adalah:

- a. Bagaimana solusi terbaik algoritma *Elephant Herding Optimization* (EHO) pada penyelesaian permasalahan *multiple constraints knapsack* 0-1?
- b. Bagaimana performa algoritma *Elephant Herding Optimization* (EHO) dibandingkan dengan metode simpleks pada penyelesaian permasalahan *multiple constraints knapsack* 0-1?



### 1.3 Tujuan

Tujuan penelitian ini menjawab masalah yang telah dirumuskan sebelumnya.

- a. Mendapatkan solusi terbaik algoritma *Elephant Herding Optimization* (EHO) pada penyelesaian permasalahan *multiple constraints knapsack* 0-1.
- b. Mengetahui performa algoritma *Elephant Herding Optimization* (EHO) dibandingkan dengan metode simpleks pada penyelesaian permasalahan *multiple constraints knapsack* 0-1.

### 1.4 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah dapat memberikan solusi dari permasalahan *multiple constraints knapsack* 0-1 dengan menggunakan algoritma *Elephant Herding Optimization* (EHO) dan membandingkannya dengan menggunakan metode simpleks menggunakan *Solver Add-In* pada *Microsoft Excel*.

## BAB 2. TINJAUAN PUSTAKA

### 2.1 Optimasi

Menurut Soekartawi (1992), optimasi merupakan pencapaian suatu keadaan terbaik dimana solusi masalah diarahkan pada batas maksimum atau minimum. Sedangkan menurut Tarmizi (2005) optimasi merupakan cara untuk mendapatkan nilai ekstrim secara maksimum atau minimum dari suatu fungsi tujuan dengan memperhatikan faktor-faktor fungsi kendalanya. Faktor-faktor yang menjadi kendala pada fungsi tujuan ikut menentukan nilai maksimum atau minimumnya. Berikut ini adalah beberapa persoalan yang memerlukan optimasi yang sering muncul (Munir, 2005):

- a. penentuan pemilihan barang pada masalah *knapsack*;
- b. menentukan lintasan terpendek dari suatu tempat ke tempat 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 Permasalahan *Knapsack* (*Knapsack Problem*)

*Knapsack problem* merupakan suatu permasalahan bagaimana memilih objek dari sekian banyak dan berapa besar objek tersebut akan disimpan sehingga diperoleh suatu penyimpanan yang optimal dengan memperhatikan objek yang terdiri dari  $n$  objek  $(1, 2, 3, \dots, n)$  dimana setiap objek memiliki berat  $(w_i)$ , nilai *profit* atau keuntungan  $(p_i)$  dengan memperhatikan juga kapasitas dari media penyimpanan sebesar  $C$ .

Permasalahan *knapsack* ini sering digunakan pada bidang jasa pengangkutan barang seperti pengangkutan peti kemas dalam sebuah media pengangkut. Dalam usaha tersebut, diinginkan keuntungan yang maksimal untuk mengangkut barang dengan tidak melebihi kapasitas yang ada. Berdasarkan persoalan tersebut, diharapkan ada suatu solusi yang secara otomatis dapat mengatasi persoalan itu.

*Knapsack* adalah permasalahan mengenai optimalisasi kombinatorial dimana harus mencari solusi terbaik dari banyak kemungkinan yang dihasilkan (Dimiyati dan Dimiyati, 2004).

Menurut Pisinger (1995) *Knapsack* terdiri dari beberapa persoalan antara lain sebagai berikut.

- a. *Knapsack 0-1 (Binary Knapsack)*  
Objek yang dimasukkan ke dalam media penyimpanan dimensinya harus dimasukkan semua atau tidak sama sekali.
- b. *Knapsack terbatas (Bounded Knapsack)*  
Objek yang dimasukkan ke dalam media penyimpanan dimensinya bisa dimasukkan sebagian atau seluruhnya.
- c. *Knapsack tak terbatas (Unbounded Knapsack)*  
Jumlah objek yang dimasukkan ke dalam media penyimpanan dimensinya tidak terbatas.

Ada banyak variasi dari masalah *knapsack* yang muncul dengan aplikasi jumlah besar dari masalah dasar. Variasi utama terjadi dengan mengubah jumlah beberapa parameter masalah seperti jumlah kendala, jumlah tujuan, atau bahkan jumlah medianya. Permasalahan *knapsack* tersebut terdiri dari beberapa variasi sebagai berikut (Kellerer *et al.*, 2004 dalam Hilviah, 2015):

- a. *Multi Objective Knapsack Problem*  
Permasalahan yang memiliki fungsi tujuan lebih dari 1 untuk memaksimalkan keuntungan yang diperoleh.
- b. *Multi Dimensional* atau *Multiple Constraints Knapsack Problem*  
Permasalahan yang memiliki kendala lebih dari 1 untuk memaksimalkan keuntungan yang diperoleh.
- c. *Multi Knapsack Problem*  
Permasalahan yang memiliki fungsi objektif/media penyimpanan lebih dari 1 dimana semua item harus dikemas untuk memaksimalkan keuntungan yang diperoleh.

d. *Quadratic Knapsack Problem*

Permasalahan yang tujuannya memaksimalkan fungsi objektif dalam bentuk kuadratik untuk kendala kapasitas biner dan linear.

*Knapsack* yang akan dibahas pada skripsi ini adalah jenis *Multiple Constraints Knapsack Problem* (MCKP) 0-1. Variabel keputusan yang diperoleh yaitu  $x_i$  bernilai 1 jika objek dipilih dan  $x_i$  bernilai 0 jika objek tidak pilih.

2.2.1 *Knapsack 0-1 Problem*

Dalam persoalan ini, kita diberikan  $n$  buah objek dan sebuah media penyimpanan yang memiliki daya tampung maksimal senilai  $C$ . Setiap benda memiliki berat ( $w_i$ ) dengan nilai keuntungan atau profit ( $p_i$ ). Objektif dari permasalahan ini adalah bagaimana memilih objek-objek yang dimasukkan ke dalam media penyimpanan sehingga tidak melebihi kapasitas dari media penyimpanan namun memaksimalkan total keuntungan yang diperoleh (Springer, 2005).

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. Karena itulah persoalan ini dinamakan *knapsack* 0-1 (*integer knapsack*). Sebagai contoh,  $X = \{0,1,0,1\}$  adalah sebuah solusi yang ditemukan, maka 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, persoalan *knapsack* 0-1 dapat dirumuskan sebagai berikut:

Fungsi tujuan:

$$Z = \sum_{i=1}^n p_i x_i \quad (2.1)$$

Kendala:

$$\sum_{i=1}^n w_i x_i \leq C \quad (2.2)$$

dengan  $x_i = 0$  atau  $1$ ,  $i = 1, 2, 3, \dots, n$ ,

dimana:

- $Z$  : nilai optimum dari fungsi tujuan  
 $n$  : banyak barang  
 $p_i$  : keuntungan barang  
 $w_i$  : berat (*weight*) barang  
 $x_i$  : variabel keputusan (1 jika dipilih, 0 jika tidak dipilih)  
 $C$  : kapasitas media penyimpanan (*knapsack*)

### 2.2.2 Multiple Constraints Knapsack 0-1

Pada permasalahan *Multiple Constraints Knapsack 0-1*, tiap-tiap item pilihan memiliki batasan lebih dari satu dimensi. Batasan (kendala) tersebut contohnya seperti berat, biaya, waktu, pekerja, dan lainnya (Puchinger *et al.*, 2007). Tujuannya yaitu untuk memperoleh solusi optimum dengan memilih kombinasi barang dimana semua batasan tidak melebihi kapasitas yang tersedia. *Multiple Constraints Knapsack 0-1* dapat dirumuskan sebagai berikut (Kellerer *et al.*, 2004 dalam Hilviah, 2015).

Fungsi tujuan maksimal:

$$Z = \sum_{i=1}^n p_i x_i \quad (2.3)$$

Kendala:

$$\sum_{i=1}^n w_i x_i \leq C, x_i \in \{0,1\} \quad (2.4)$$

$$\sum_{i=1}^n b_i x_i \leq M, x_i \in \{0,1\} \quad (2.5)$$

$$\sum_{i=1}^n v_i x_i \leq V, x_i \in \{0,1\} \quad (2.6)$$

dengan  $i = 1,2,3, \dots, n$ ,

dimana:

- $Z$  : nilai optimum dari fungsi tujuan  
 $i$  : indeks barang  
 $p_i$  : keuntungan tiap barang ke- $i$  ( $p_i > 0$ )  
 $x_i$  : variabel keputusan (1 jika dipilih dan 0 jika tidak)  
 $w_i$  : berat barang ke- $i$  ( $w_i \geq 0$ )  
 $b_i$  : harga beli  
 $v_i$  : volume barang ke-  $i$  ( $v_i \geq 0$ )

- $C$  : kapasitas berat maksimal *knapsack*  
 $V$  : kapasitas volume maksimal *knapsack*  
 $M$  : modal

Perumusan matematik *Multiple Constraints Knapsack* 0-1 (MCKP) diatas menjelaskan bahwa total keuntungan barang yang dibawa ( $p_i x_i$ ) dengan banyaknya barang ( $n$ ) tidak lebih dari nilai optimum ( $Z$ ). Persamaan (2.4) merumuskan variabel keputusan berat barang ( $w_i x_i$ ) dengan sejumlah barang yang dibawa ( $n$ ) tidak lebih dari kapasitas berat maksimal *knapsack* ( $C$ ). Persamaan (2.5) merumuskan total harga beli barang yang dibawa ( $b_i x_i$ ) sejumlah ( $n$ ) tidak lebih dari modal yang tersedia ( $M$ ). Sedangkan persamaan (2.6) merumuskankan total volume barang yang dibawa ( $v_i x_i$ ) sejumlah ( $n$ ) tidak lebih dari kapasitas volume maksimal *knapsack* ( $V$ ). Variabel keputusan ( $x_i$ ) bernilai 0 yang berarti barang tidak dibawa, sedangkan 1 yang berarti barang dibawa. Tujuan yang diharapkan yaitu memilih subset dari sekian banyak barang yang ada dengan total keuntungan ( $p_i$ ) maksimal dengan memperhatikan kendala-kendala atau batasan-batasan yang ada.

### 2.3 Algoritma *Elephant Herding Optimization* (EHO)

Jenis baru metode pencarian metaheuristik berbasis keramaian disebut dengan Algoritma *Elephant Herding Optimization* (EHO), diusulkan untuk menyelesaikan tugas optimasi. Metode EHO terinspirasi oleh perilaku menggiring kelompok gajah atau perpindahan gajah dari tempat satu ke tempat yang lain (*herding*) yang dibuat oleh Wang *et al* pada tahun 2015. Di alam, gajah-gajah dengan kelompok yang berbeda hidup bersama di bawah kepemimpinan *matriarch*, dan gajah jantan akan meninggalkan kelompok keluarga ketika tumbuh dewasa. Kedua perilaku ini dapat dimodelkan menjadi dua operator yaitu *clan updating operator* dan *separating operator*.

Untuk membuat perilaku menggiring gajah memecahkan semua jenis masalah pengoptimalan global, terlebih dahulu memilih elemen-elemen untuk menyederhanakannya ke dalam aturan yang diidealkan berikut ini.

- a. Populasi gajah terdiri dari beberapa kelompok, dan setiap kelompok memiliki sejumlah gajah tetap.

- b. Sejumlah gajah jantan tetap akan meninggalkan kelompok keluarga mereka dan hidup secara diam-diam jauh dari kelompok gajah utama pada setiap generasi.
- c. Gajah-gajah di masing-masing kelompok hidup bersama di bawah pimpinan (*matriarch*).

Tahap-tahap dalam EHO meliputi *clan updating operator* dan *separating operator*:

a. *Clan updating operator*

Seperti yang disebutkan sebelumnya, semua gajah hidup bersama di bawah kepemimpinan seorang *matriarch* di setiap kelompok. Oleh karena itu, untuk setiap gajah dalam kelompok, posisi berikutnya dipengaruhi oleh *matriarch ci*. Untuk gajah anggota di kelompok *ci* diperbarui berdasarkan Persamaan (2.3).

$$x_{ci,j}(t+1) = x_{ci,j}(t) + \alpha (Mat_{ci} - x_{ci,j}(t)) r \quad (2.3)$$

dengan,

$x_{ci,j}(t+1)$  = posisi baru untuk gajah *j* di kelompok *ci*

$x_{ci,j}(t)$  = posisi lama untuk gajah *j* di kelompok *ci*

$\alpha$  = faktor skala yang menentukan pengaruh *matriarch ci* pada  $x_{ci,j}(t)$   
( $\alpha \in [0,1]$ )

$Mat_{ci}$  = gajah terbaik di kelompok *ci* (*matriarch ci*)

$r$  = bilangan random pada interval  $[0,1]$ .

Pemimpin kelompok (*Matriarch*) memperbarui posisinya dengan Persamaan (2.4).

$$x_{ci,j}(t+1) = \beta x_{center,ci} + x_{ci,j}(t) \quad (2.4)$$

dengan,

$\beta$  = faktor yang menentukan pengaruh  $x_{center,ci}$  pada  $x_{ci,j}(t+1)$   
( $\beta \in [0,1]$ )

$x_{center,ci}$  = pusat kelompok dihitung berdasarkan Persamaan (2.5)

$$x_{center,ci} = \frac{1}{n_{ci}} \sum_{j=1}^{n_{ci}} x_{ci,j} \quad (2.5)$$

dengan,

$x_{center,ci}$  = pusat kelompok *ci*

$n_{ci}$  = jumlah gajah dalam kelompok  $ci$

$x_{ci,j}$  = posisi gajah  $j$  di kelompok  $ci$

Berdasarkan uraian di atas, menurut Wang *et al* (2015) *clan updating operator* dapat diformulasikan seperti yang ditunjukkan pada Gambar 2.1.

```

for  $ci=1$  to  $nClan$  (untuk semua kelompok dalam populasi gajah) do
  for  $j=1$  to  $n_{ci}$  (untuk semua gajah dalam kelompok  $ci$ ) do
    if  $x_{ci,j}(t) \neq Mat_{ci}$  then
      Perbarui  $x_{ci,j}(t)$  dan generasi  $x_{ci,j}(t + 1)$  dengan persamaan
      (2.3)
    else
      Perbarui  $x_{ci,j}(t)$  dan generasi  $x_{ci,j}(t + 1)$  dengan persamaan
      (2.4)
  
```

Gambar 2.1 Pseudocode clan updating operator

b. *Separating operator*

Dalam kelompok gajah, gajah jantan akan meninggalkan kelompok keluarga dan hidup sendiri ketika dewasa. Proses pemisahan ini dapat dimodelkan ke dalam *separating operator* ketika memecahkan masalah optimasi. Untuk lebih meningkatkan kemampuan pencarian metode EHO, diasumsikan bahwa individu gajah dengan *fitness* terburuk akan menerapkan *separating operator* pada setiap generasi seperti yang ditunjukkan dalam persamaan (2.6).

$$x_{worst,ci} = x_{min} + (x_{max} - x_{min} + 1) r \quad (2.6)$$

dengan,

$x_{max}$  = batas atas dari posisi individu gajah

$x_{min}$  = batas bawah dari posisi individu gajah

$x_{worst,ci}$  = individu gajah terburuk di kelompok  $ci$

$r$  = bilangan random dalam kisaran  $[0,1]$



Dengan demikian, *separating operator* menurut Wang *et al* (2015) dapat dibentuk seperti yang ditunjukkan pada Gambar 2.2.

```
for  $ci=1$  to  $nClan$  (semua kelompok dalam populasi gajah) do  
    Ganti gajah yang paling buruk di kelompok  $ci$  dengan persamaan (2.6)  
end for  $ci$ 
```

Gambar 2.2 Pseudocode *separating operator*

Berdasarkan deskripsi *clan updating operator* dan *separating operator*, metode EHO dikembangkan dan diringkas seperti yang ditunjukkan pada Gambar 2.3.

```
Langkah 1 : Initialization. Atur penghitung generasi  $t = 1$ ; menginisialisasi populasi;  $MaxGen$  generasi maksimum.  
Langkah 2 : While  $t < MaxGen$  do  
    Urutkan semua gajah sesuai dengan kemampuan mereka.  
    Menerapkan clan updating operator dengan Gambar 2.1.  
    Menerapkan separating operator seperti yang ditunjukkan pada Gambar 2.2.  
    Evaluasi populasi dengan posisi yang baru diperbarui.  
     $t = t + 1$ .  
Langkah 3 : end while
```

Gambar 2.3 Pseudocode algoritma EHO

## 2.4 Metode Simpleks

Menurut Nasution *et al* (2016) metode simpleks merupakan penentuan solusi optimal dilakukan dengan memeriksa titik ekstrim satu per satu dengan cara perhitungan iteratif. Sedangkan menurut Wirdasari (2009), metode simpleks digunakan untuk mencari nilai optimal dari program linier yang melibatkan banyak pembatas dan banyak variabel (lebih dari dua variabel). Beberapa ketentuan dalam penyelesaian metode simpleks antara lain:

- a. Nilai kanan fungsi tujuan harus nol (0).

- b. Nilai kanan fungsi kendala harus positif. Apabila negatif, nilai tersebut harus dikali dengan -1.
- c. Fungsi kendala dengan tanda " $\leq$ " harus diubah ke bentuk " $=$ " dengan menambahkan variabel slack/surplus. Variabel slack/surplus disebut juga variabel dasar. Penambahan variabel slack menyatakan kapasitas yang tidak digunakan atau tersisa pada sumber daya tersebut. Hal ini karena ada kemungkinan kapasitas yang tersedia tidak semua digunakan dalam proses produksi.
- d. Fungsi kendala dengan tanda " $\geq$ " diubah ke bentuk " $\leq$ " dengan cara mengalikan dengan -1, lalu diubah ke bentuk persamaan ( $=$ ) dengan ditambah variabel slack. Kemudian karena nilai kanannya negatif, dikalikan lagi dengan -1 dan ditambah variabel *artificial* (M). Variabel *artificial* ini secara fisik tidak mempunyai arti, dan hanya digunakan untuk kepentingan perhitungan saja.
- e. Fungsi kendala dengan tanda " $=$ " harus ditambah variabel *artificial* (M).

Menurut Firmansyah *et al* (2018) langkah-langkah metode simpleks dalam menganalisis data yaitu sebagai berikut:

1. Mengidentifikasi variabel keputusan dan memformulasikannya ke dalam simbol matematika.
2. Mengidentifikasi fungsi tujuan yang akan dicapai dan fungsi batasan ke dalam model matematika.
3. Fungsi tujuan dan fungsi batasan diformulasikasikan ke dalam bentuk baku metode simpleks dengan menambahkan variabel slack.
4. Membuat tabel awal metode simpleks.
5. Memasukkan nilai masing-masing variabel ke dalam tabel awal simpleks.
6. Menentukan kolom kunci berdasarkan nilai z terkecil.
7. Menentukan rasio solusi =  $\frac{s}{\text{nilai kolom kunci}}$
8. Menentukan baris kunci berdasarkan rasio terkecil (tanpa baris z).
9. Menentukan *elemen cell* yang merupakan irisan dari kolom kunci dan baris kunci.

10. Melakukan tahapan (iterasi) yang diawali dengan menentukan baris kunci yang baru.

$$\text{baris kunci baru} = \frac{\text{baris kunci lama}}{\text{elemen cell}}$$

11. Mentransformasikan baris yang lain selain baris kunci baris baru selain baris kunci = baris lama – [(nilai kolom lama) x (baris kunci baru)]  
(jika koefisien pada baris z masih ada yang bernilai negatif, maka kembali ke nomor 6 - 11)
12. Melakukan uji optimalitas, hingga semua koefisien pada baris z sudah tidak ada lagi yang bernilai negatif, yang berarti tabel sudah optimal.

## 2.5 Simpleks pada *Solver Add-In* Microsoft Excel

Menurut Lestari (2011) untuk menyelesaikan masalah mengenai jawaban fungsi tujuan, fungsi kendala serta analisis sensitivitas digunakan *solver* yang ada pada salah satu menu *Excel* caranya klik menu *Tools* kemudian pilih *solver*. *Solver* merupakan salah satu fasilitas tambahan pada *Excel* fungsinya untuk menyelesaikan kombinasi variabel dalam meminimalkan atau memaksimalkan satu sel target. *Solver* juga dapat mendefinisikan sendiri suatu batasan atau kendala yang harus dipenuhi agar penyelesaian masalah dianggap benar.

Langkah pertama yang harus dilakukan sebelum memasuki *solver* adalah mempersiapkan dan menyusun *worksheet* dengan mendefinisikan dan memilih sel yang akan berisi variabel keputusan, kendala, dan fungsi tujuan dari suatu masalah. Langkah kedua, memasukkan data fungsi tujuan, kendala, dan variabel keputusan ke dalam *worksheet Excel*. Langkah ketiga, menuliskan rumus untuk menghitung nilai fungsi tujuan dan nilai variabel slack.

Langkah selanjutnya adalah menjalankan *Solver* dengan menentukan atau mengisi *Target Cell*, *Equal To*, *By Changing Cells*, dan *Subject to Constraints*. *Target Cell* merupakan sel yang mencerminkan nilai yang akan dimaksimalkan atau diminimalkan pada parameter ke dua. *Equal to* mencerminkan tujuan yang akan dicapai. Ada tiga pilihan, yaitu Max (maksimum), Min (minimum), dan *Value of*. *By Changing Cells* berisi nilai yang akan menentukan nilai fungsi tujuan atau marjin total. Sel ini mencerminkan variabel keputusan atau variabel yang dicari.

Sedangkan *Subject to Constraints* berisi kendala-kendala yang akan membatasi pengoptimalan nilai fungsi tujuan atau marjin total. Setelah langkah tersebut klik *Solve* dan *solver* akan bekerja.



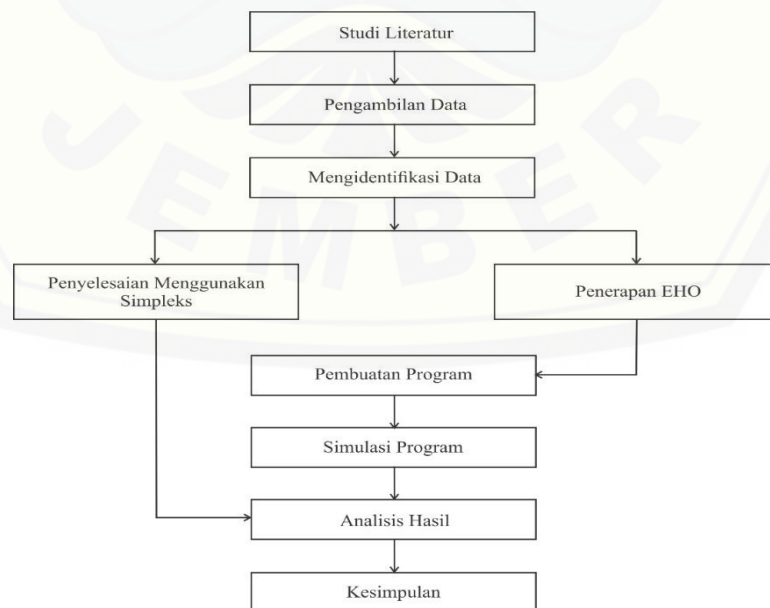
### BAB 3. METODE PENELITIAN

#### 3.1 Data Penelitian

Terdapat dua data yang digunakan untuk penelitian ini. Data satu diambil dari Toko Citra Tani terletak di Jl. PB Sudirman No.74 di Kecamatan Panti Kabupaten Jember, yaitu usaha dibidang perdagangan jenis barang bangunan seperti cat, semen, dan masih banyak lagi (lihat Lampiran A.1). Sedangkan data kedua diambil dari penelitian Putri, R. J. M. (2017) (lihat Lampiran A.2). Data yang digunakan berupa nama barang, jumlah barang, satuan, berat satuan, harga beli, harga jual serta volume maksimumnya. Adapun media penyimpanan (*knapsack*) yang digunakan untuk data 1 memiliki kapasitas berat 5.000 kg dan kapasitas volume 9.000.000  $cm^3$ . Modal yang dimiliki toko Rp. 20.000.000,-. Media penyimpanan (*knapsack*) yang digunakan untuk data 2 memiliki kapasitas berat 4.000 kg dan kapasitas volume 8.540.000  $cm^3$ . Modal yang dimiliki toko Rp. 10.000.000,-.

#### 3.2 Langkah-langkah Penelitian

Berikut skema untuk menyelesaikan permasalahan *knapsack* 0-1 menggunakan algoritma *Elephant Herding Optimization* (EHO).



Gambar 3.1 Skema Langkah-Langkah Penelitian

Penelitian penerapan algoritma *Elephant Herding Optimization* (EHO) pada permasalahan *knapsack* 0-1 akan dilakukan dengan langkah-langkah berikut:

a. Studi Litelatur

Langkah awal dilakukan dengan mencari dan mengumpulkan informasi mengenai algoritma *Elephant Herding Optimization* (EHO) dan permasalahan *knapsack* pada jurnal internasional dan buku-buku yang berkaitan dengan dua hal tersebut.

b. Pengambilan Data

Pengambilan diperoleh dari data Toko Citra Tani dan penelitian Putri, R. J. M (2017).

c. Identifikasi Data

Mengidentifikasi data dilakukan dengan memilih data yang bersesuaian dengan permasalahan *knapsack* 0-1 untuk mencari keuntungan dan berat dari masing-masing barang.

d. Penyelesaian *Solver*

Menyelesaikan data yang telah diidentifikasi menggunakan metode simpleks menggunakan *Solver Add-In* pada *Microsoft Excel*.

e. Penerapan Algoritma *Elephant Herding Optimization* (EHO)

Menerapkan algoritma *Elephant Herding Optimization* pada permasalahan *knapsack* 0-1 yaitu dengan langkah sebagai berikut.

- 1) Inialisasi Populasi Awal
- 2) Pengecekan Solusi
- 3) Menghitung *Fitness*
- 4) Simpan *Matriarch*
- 5) *Clan Updating Operator*
  - a) Didapatkan posisi baru
  - b) Cek solusi baru
  - c) Menghitung *Fitness*
- 6) *Separating Operator*
- 7) *Update Matriarch*

#### 8) Cek Kriteria Pemberhentian

Jika iterasi maksimal belum tercapai kembali ke langkah 5.

#### f. Pembuatan Program

Membuat program berdasarkan permasalahan yang akan dibahas mengenai *multiple constraints knapsack 0-1* menggunakan algoritma *Elephant Herding Optimization* (EHO) dengan program yang digunakan adalah *software* Matlab.

#### g. Simulasi Program

Setelah program dibuat langkah selanjutnya yaitu mensimulasi program.

#### h. Analisis Hasil

Setelah langkah penyelesaian menggunakan *Solver Add-In* dan pembuatan program adalah menganalisis hasil keduanya dan membandingkannya.

#### i. Kesimpulan

Langkah terakhir adalah membuat kesimpulan dari hasil yang telah diperoleh.

## BAB 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan hasil dan pembahasan dapat diambil kesimpulan sebagai berikut.

- a. Permasalahan *multiple constraints knapsack* 0-1 menggunakan algoritma *Elephant Herding Optimization* (EHO) pada kedua data menghasilkan profit terbaik sebesar Rp. 2.714.000,- (data 1) dengan nilai  $nClan = 25$ ;  $nCi = 25$ ,  $\alpha = 0,9$ ;  $\beta = 0,9$  dan  $MaxGen = 100$ , serta profit sebesar Rp. 14.291.000,- (data 2) dengan nilai  $nClan = 50$ ;  $nCi = 50$ ,  $\alpha = 0,9$ ;  $\beta = 0,9$  dan  $MaxGen = 1000$ . Rata-rata iterasi konvergen yang didapatkan sebesar 25,4 (data 1) dan 536,3 (data 2). Pengaruh parameter pada algoritma *Elephant Herding Optimization* (EHO) dalam menyelesaikan permasalahan *multiple constraints knapsack* 0-1 yaitu jika nilai parameternya semakin besar maka profit yang didapatkan akan semakin optimal.
- b. Hasil algoritma EHO dibandingkan dengan metode Simpleks pada *Microsoft Excel* memiliki hasil rata-rata presentase deviasi yang optimal yaitu sebesar 0% (data 1) dan 0,038486% (data 2), hal ini menunjukkan bahwa algoritma EHO efektif dalam menyelesaikan permasalahan *multiple constraints knapsack* 0-1.

### 5.2 Saran

Saran untuk penelitian selanjutnya dapat mengembangkan algoritma *Elephant Herding Optimization* (EHO) pada permasalahan optimasi lain, seperti penentuan rute terpendek, penjadwalan, dan lain-lain.



**DAFTAR PUSTAKA**

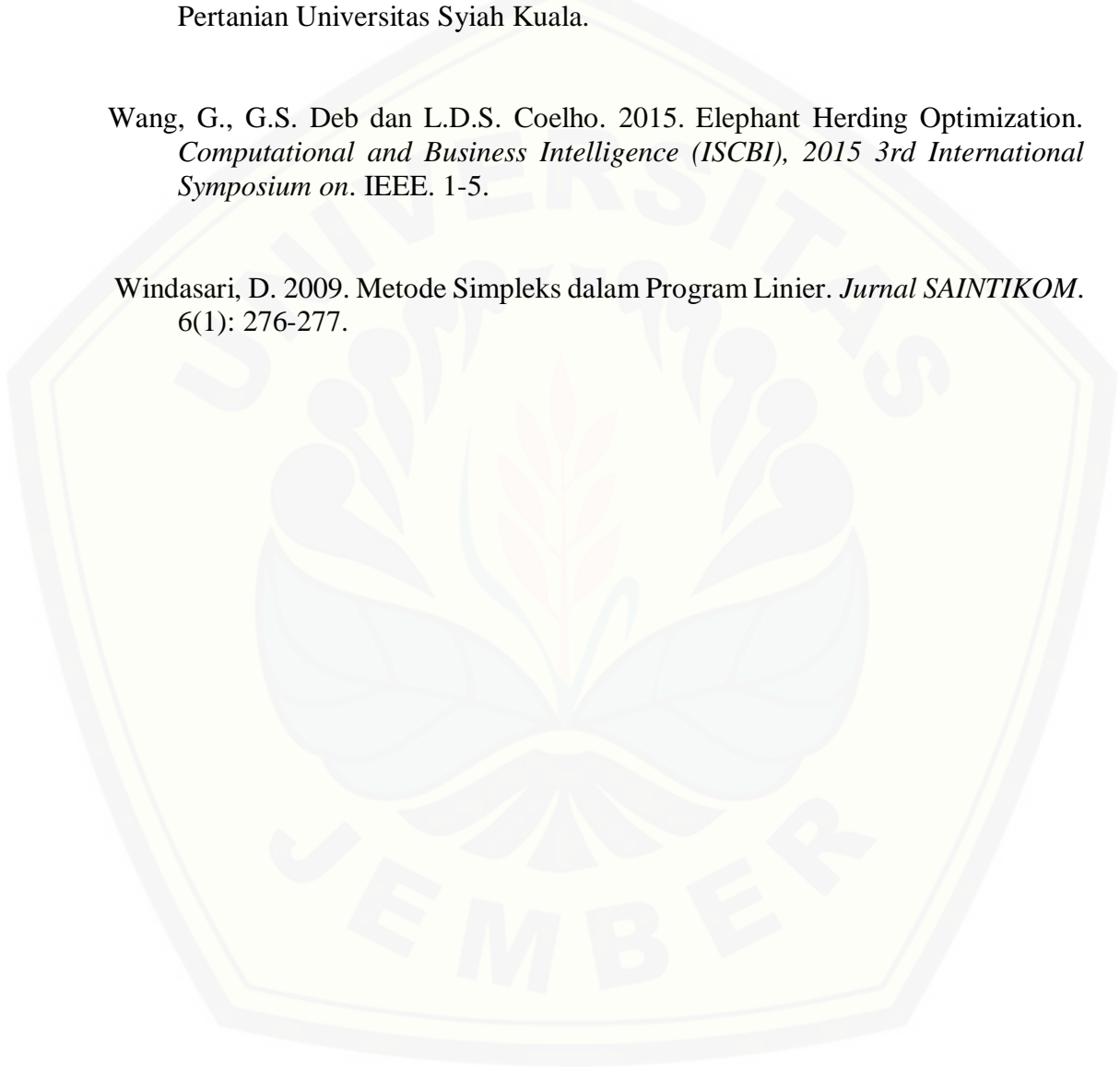
- Dimiyanti, T.T. dan A. Dimiyanti. 2004. *Operations Research: Model-model Pengambilan Keputusan*. Bandung: Sinar Baru Agesindo.
- Firmansyah., D.J. Panjaitan., M. Salayan dan A.D. Silalahi. 2018. Pengoptimalan Keuntungan Badan Usaha Karya Tani di Deli Serdang dengan Metode Simpleks. *JISTech*, 3(1): 2528-5718.
- Hilviah, F. 2015. Penerapan Algoritma Dynamic Programming dan Algoritma Backtracking pada Permasalahan Multiple Constraints Knapsack 0-1. *Skripsi*. Jember: Universitas Jember.
- Lestari, H.P. 2011. Pemanfaatan *Excel Solver* dalam Pembelajaran Pemrograman Linier. *Prosiding Seminar Nasional Penelitian, Pendidikan dan Penerapan MIPA, Fakultas MIPA, Universitas Negeri Yogyakarta*, 1-7.
- Munir, R. 2005. *Strategi Algoritmik*. Yogyakarta: Graha Ilmu.
- Nasution, Z., H. Sunandar, I. Lubis dan L.T. Sianturi. 2016. Penerapan Metode Simpleks untuk Menganalisa Persamaan Linier dalam Menghitung Keuntungan Maksimum. *Jurnal Riset Komputer (JURIKOM)*. 3(4): 42.
- Pisinger, D. 1995. *A minimal Algorithm for the Multiple-Choice Knapsack Problem*. *European Journal of Operational Research*, 83(2): 349-410.
- Putri, R. J. M. 2017. Penerapan Algoritma *Modified Variable Neighborhood Search (MVNS)* pada Permasalahan Multiple Constraints Knapsack 0-1. *Skripsi*. Jember: Jurusan Matematika FMIPA Universitas Jember.
- Puchinger, D., G.R. Raidl dan U. Pferschy. 2007. The Multidimensional Knapsack Problem: Structure And Algorithms. *INFORMS Journal on Computing*. 22: 250-265.
- Springer, V. 2005. *Knapsack 0-1 Problem*. Yogyakarta: Graha Ilmu.

Soekartawi. 1992. *Linier Programming Teori dan Aplikasinya Khususnya dalam Bidang Pertanian*. Jakarta: Rajawali Pers.

Tarmizi. 2005. *Optimasi Usaha Tani dalam Pemanfaatan Air Irigasi Embung Leubuk Aceh Besar*. Skripsi. Banda Aceh: Jurusan Teknik Pertanian Fakultas Pertanian Universitas Syiah Kuala.

Wang, G., G.S. Deb dan L.D.S. Coelho. 2015. Elephant Herding Optimization. *Computational and Business Intelligence (ISCBI), 2015 3rd International Symposium on*. IEEE. 1-5.

Windsari, D. 2009. Metode Simpleks dalam Program Linier. *Jurnal SAINTIKOM*. 6(1): 276-277.



## LAMPIRAN

## Lampiran A. Data Penelitian

A.1 Data 1 (Kapasitas berat *knapsack* = 5.000 kg, kapasitas volume *knapsack* = 9.000.000  $cm^3$ , dan modal = Rp. 20.000.000,-)

No	Nama Barang (i)	Satuan	Jumlah	Berat (kg)	Volume ( $cm^3$ )	Harga Beli	Harga Jual
1	Altex Emulsion Paint 1 kg	Kaleng	30	1	2156	21.600	24.000
2	Altex Emulsion Paint 5 kg	Kaleng	20	5	6583.5	89.800	99.800
3	Altex Cat Synthetic - Kaleng Merah	Kaleng	30	0,5	855.6429	29.700	33.000
4	Altex - Meni kayu, Meni Besi	Kaleng	30	0,5	905.1429	16.700	18.600
5	Altex - Plamir Kayu 0,5 kg	Kaleng	30	0,5	905.1429	16.700	18.600
6	Altex - Plamir Kayu 1 kg	Kaleng	30	1	1331	29.300	32.600
7	Altex - Warna Khusus 1 kg	Kaleng	50	1	1331	22.200	24.700
8	Altex - Warna Khusus 5 kg	Kaleng	30	5	6583.5	99.400	110.400
9	Paku 3/4"	Dus	30	1	2704	16.000	20.000
10	Paku 1"	Dus	30	1	3120	15.500	20.000
11	Paku 1 1/4"	Dus	30	1	2730	15.000	20.000
12	Paku 1 1/2"	Dus	30	1	22176	15.500	20.000
13	Paku 2"	Dus	30	1	22176	13.000	18.000
14	Paku 3"	Dus	30	1	22176	13.000	17.000
15	Paku 4"	Dus	30	1	22176	13.000	17.000
16	Semen Gresik	Sak	20	40	23560	48.000	50.000
17	Semen Holsim	Sak	20	40	23560	43.500	45.000
18	Semen Bosowa	Sak	20	40	23560	40.500	43.000
19	Semen Puger	Sak	20	40	23560	38.000	40.000
20	Semen Putih Gresik	Sak	20	40	23560	82.500	85.000
21	Mowilex	Kaleng	15	1	1584	65.000	70.000
22	Kawat	Kg	50	1	31360	18.000	20.000
23	Bendrat	Kg	40	1	31360	16.000	22.000
24	Kalsium	Sak	5	30	31200	32.500	38.000
25	Flintkote Oil	Kaleng	5	1	1357.714	20.000	23.500
26	Lem Rajawali	Dus	10	12	13500	12.000	13.500
27	Karbit	Kg	30	1	25600	23.000	25.000
28	Japen Cat	Kaleng	20	1	1470.857	52.000	55.000
29	Paragon	Kaleng	20	5	5672.857	95.000	100.000
30	Paragon Genteng	Kaleng	20	5	5672.857	120.000	125.000

A.2 Data 2 (Kapasitas berat *knapsack* = 4.000 kg, volume *knapsack* = 8.540.000  $cm^3$ , dan modal = Rp. 10.000.000,-)

No	Nama Barang	Jumlah Ketersediaan	Berat (kg)	Volume ( $cm^3$ )	Harga Beli (Rp)	Harga Jual (Rp)
1	Tempat Tisu A	5	0,4	2475	47000	100000
2	Tempat Tisu B	5	0,3	2475	25000	90000
3	Tempat Tisu C	5	0,25	2475	17500	35000
4	Asbak	3	0,2	36	15000	35000
5	Tempat Perhiasan A	4	0,2	810	17500	40000
6	Tempat Perhiasan B	4	0,2	810	17500	35000
7	Nampan A	5	0,4	4725	35000	100000
8	Nampan D	5	0,4	4725	35000	80000
9	Tempat Minuman	4	0,25	1320	32000	65000
10	Tempat Pensil	5	0,025	441	4000	6500
11	Tempat Sendok	5	0,1	588	17500	40000
12	Tempat Sedotan	5	0,02	392	6000	15000
13	Gerbong Maut	3	0,2	5500	62000	150000
14	Rak Hiasan Dinding	4	1,5	10800	20000	80000
15	Tempat Payung	2	3	96000	85000	150000
16	Tempat Koran	2	1,5	39375	60000	125000
17	Rak Sepatu	2	7	120000	150000	300000
18	Lampu Tempel B + I	4	0,3	23625	90000	250000
19	Lampu Tempel E + I	4	0,5	23625	80000	150000
20	Lampu Tempel F + G + I	4	0,2	9800	75000	150000
21	Lampu Tempel H + I	4	0,5	12250	90000	200000
22	Lampu Gantung B1	4	0,3	10115	60000	90000
23	Lampu Gantung J2	4	0,3	36000	55000	125000
24	Lampu Gantung J1	4	0,2	12250	30000	90000
25	Lampu Gantung K2	4	0,2	36000	40000	75000
26	Lampu Gantung K1	4	0,2	12250	40000	75000
27	Lampu Duduk L1	4	1,2	5625	55000	100000
28	Lampu Duduk L2	4	1,2	7875	55000	100000
29	Lampu Duduk M****	3	5,5	64000	200000	400000
30	Lampu Duduk M***	3	5,5	64000	170000	250000

No	Nama Barang	Jumlah Ketersediaan	Berat (kg)	Volume (cm <sup>3</sup> )	Harga Beli (Rp)	Harga Jual (Rp)
31	Lampu Duduk M**	3	5,5	64000	120000	200000
32	Lampu Duduk M*	3	5,5	64000	75000	150000
33	Lampu Duduk Na	3	0,3	13781,25	30000	50000
34	Lampu Duduk Nb	3	0,3	20000	35000	75000
35	Lampu Duduk Nc	3	0,3	20000	40000	90000
36	Lampu Duduk O*	3	2	26250	17500	50000
37	Lampu Duduk O**	3	2,5	26250	60000	125000
38	Lampu Duduk O***	3	4,5	26250	70000	200000
39	Lampu Duduk Pa	3	0,5	13781,25	15000	75000
40	Lampu Duduk Pb	3	0,4	20000	15000	125000
41	Lampu Duduk Pc	3	0,4	20000	35000	200000
42	Lampu Duduk Qa	3	0,2	12000	35000	65000
43	Lampu Duduk Qb	3	0,25	16000	40000	90000
44	Lampu Duduk Qc	3	0,3	37500	50000	175000
45	Lampu Duduk Ra	3	4	540000	225000	400000
46	Lampu Duduk Rb	3	5	675000	250000	600000
47	Lampu Duduk Rc	3	6	810000	375000	700000
48	Patung Kuda 3	4	3,5	90000	200000	600000
49	Patung Kuda 2	4	3	56250	175000	400000
50	Patung Rusa 3	4	5	112500	150000	350000
51	Patung Rusa 2	4	4	97500	100000	250000
52	Patung Rusa 1	4	3	82500	95000	200000
53	Patung Domba 3	4	3	31500	55000	100000
54	Patung Domba 2	4	2,75	27000	35000	75000
55	Patung Domba 1	4	2,5	22500	30000	65000

## Lampiran B. Hasil Metode *Simpleks* dengan *Solver Add-In Ms. Excel*

### B.1 Hasil *Simpleks* Data 1

**Microsoft Excel 16.0 Answer Report**

**Worksheet:** [Data 1 Simpleks.xlsx]Sheet1

**Report Created:** 11/03/2019 17:03:54

**Result:** Solver found a solution. All Constraints and optimality conditions are satisfied.

**Solver Engine**

Engine: Simplex LP

Solution Time: 0,407 Seconds.

Iterations: 1 Subproblems: 38

**Solver Options**

Max Time 600 sec, Iterations Unlimited, Precision 0,000001, Use Automatic Scaling

Max Subproblems Unlimited, Max Integer Sols Unlimited, Integer Tolerance 0%, Assume NonNegative

Objective Cell (Max)

Cell	Name	Original Value	Final Value
\$M\$33	Profit	0	2714000

Variable Cells

Cell	Name	Original Value	Final Value	Integer
\$I\$2	Altex Emulsion Paint solusi	0	1	Binary
\$I\$3	Altex Emulsion Paint solusi	0	1	Binary
\$I\$4	Altex Cat Synthetic - Kaleng Merah solusi	0	1	Binary
\$I\$5	Altex - Meni kayu, Meni Besi solusi	0	1	Binary
\$I\$6	Altex - Plamir Kayu solusi	0	1	Binary
\$I\$7	Altex - Plamir Kayu solusi	0	1	Binary
\$I\$8	Altex - Warna Khusus solusi	0	1	Binary
\$I\$9	Altex - Warna Khusus solusi	0	1	Binary
\$I\$10	Paku 3/4" solusi	0	1	Binary
\$I\$11	Paku 1" solusi	0	1	Binary
\$I\$12	Paku 1 1/4" solusi	0	1	Binary
\$I\$13	Paku 1 1/2" solusi	0	1	Binary
\$I\$14	Paku 2" solusi	0	1	Binary
\$I\$15	Paku 3" solusi	0	1	Binary
\$I\$16	Paku 4" solusi	0	1	Binary
\$I\$17	Semen Gresik solusi	0	0	Binary
\$I\$18	Semen Holsim solusi	0	0	Binary
\$I\$19	Semen Bosowa solusi	0	1	Binary
\$I\$20	Semen Puger solusi	0	0	Binary
\$I\$21	Semen Putih Gresik solusi	0	0	Binary
\$I\$22	Mowilex solusi	0	1	Binary
\$I\$23	Kawat solusi	0	1	Binary
\$I\$24	Bendrat solusi	0	1	Binary
\$I\$25	Kalsium solusi	0	1	Binary
\$I\$26	Flintkote Oil solusi	0	1	Binary
\$I\$27	Lem Rajawali solusi	0	1	Binary
\$I\$28	Karbit solusi	0	1	Binary
\$I\$29	Japen Cat solusi	0	1	Binary
\$I\$30	Paragon solusi	0	1	Binary
\$I\$31	Paragon Genteng solusi	0	0	Binary

### B.2 Hasil *Simpleks* Data 2

**Microsoft Excel 16.0 Answer Report**

**Worksheet:** [Data 2 Simpleks.xlsx]Sheet1

**Report Created:** 11/03/2019 21:30:23

**Result:** Solver found a solution. All Constraints and optimality conditions are satisfied.

**Solver Engine**

Engine: Simplex LP

Solution Time: 2,078 Seconds.

Iterations: 2 Subproblems: 470

**Solver Options**

Max Time 600 sec, Iterations Unlimited, Precision 0,000001, Use Automatic Scaling  
 Max Subproblems Unlimited, Max Integer Sols Unlimited, Integer Tolerance 0%, Assume NonNegative

Objective Cell (Max)

Cell	Name	Original Value	Final Value
\$P\$58	Profit	0	14291000

Variable Cells

Cell	Name	Original Value	Final Value	Integer
\$L\$2	Tempat Tisu A Solusi	0	1	Binary
\$L\$3	Tempat Tisu B Solusi	0	1	Binary
\$L\$4	Tempat Tisu C Solusi	0	0	Binary
\$L\$5	Asbak Solusi	0	1	Binary
\$L\$6	Tempat Perhiasan A Solusi	0	1	Binary
\$L\$7	Tempat Perhiasan B Solusi	0	1	Binary
\$L\$8	Nampan A Solusi	0	1	Binary
\$L\$9	Nampan D Solusi	0	1	Binary
\$L\$10	Tempat Minuman Solusi	0	1	Binary
\$L\$11	Tempat Pensil Solusi	0	0	Binary
\$L\$12	Tempat Sendok Solusi	0	1	Binary
\$L\$13	Tempat Sedotan Solusi	0	1	Binary
\$L\$14	Gerbong Maut Solusi	0	1	Binary
\$L\$15	Rak Hiasan Dinding Solusi	0	1	Binary
\$L\$16	Tempat Payung Solusi	0	0	Binary
\$L\$17	Tempat Koran Solusi	0	1	Binary
\$L\$18	Rak Sepatu Solusi	0	1	Binary
\$L\$19	Lampu Tempel B + I Solusi	0	1	Binary
\$L\$20	Lampu Tempel E + I Solusi	0	1	Binary
\$L\$21	Lampu Tempel F + G + I Solusi	0	1	Binary
\$L\$22	Lampu Tempel H + I Solusi	0	1	Binary
\$L\$23	Lampu Gantung B1 Solusi	0	0	Binary
\$L\$24	Lampu Gantung J2 Solusi	0	1	Binary
\$L\$25	Lampu Gantung J1 Solusi	0	1	Binary
\$L\$26	Lampu Gantung K2 Solusi	0	1	Binary
\$L\$27	Lampu Gantung K1 Solusi	0	1	Binary
\$L\$28	Lampu Duduk L1 Solusi	0	0	Binary
\$L\$29	Lampu Duduk L2 Solusi	0	1	Binary
\$L\$30	Lampu Duduk M**** Solusi	0	1	Binary
\$L\$31	Lampu Duduk M*** Solusi	0	0	Binary
\$L\$32	Lampu Duduk M** Solusi	0	0	Binary
\$L\$33	Lampu Duduk M* Solusi	0	1	Binary
\$L\$34	Lampu Duduk Na Solusi	0	0	Binary
\$L\$35	Lampu Duduk Nb Solusi	0	1	Binary
\$L\$36	Lampu Duduk Nc Solusi	0	1	Binary
\$L\$37	Lampu Duduk O* Solusi	0	1	Binary
\$L\$38	Lampu Duduk O** Solusi	0	1	Binary
\$L\$39	Lampu Duduk O*** Solusi	0	1	Binary
\$L\$40	Lampu Duduk Pa Solusi	0	1	Binary
\$L\$41	Lampu Duduk Pb Solusi	0	1	Binary
\$L\$42	Lampu Duduk Pc Solusi	0	1	Binary
\$L\$43	Lampu Duduk Qa Solusi	0	1	Binary
\$L\$44	Lampu Duduk Qb Solusi	0	1	Binary
\$L\$45	Lampu Duduk Qc Solusi	0	1	Binary
\$L\$46	Lampu Duduk Ra Solusi	0	0	Binary
\$L\$47	Lampu Duduk Rb Solusi	0	1	Binary
\$L\$48	Lampu Duduk Rc Solusi	0	0	Binary
\$L\$49	Patung Kuda 3 Solusi	0	1	Binary
\$L\$50	Patung Kuda 2 Solusi	0	1	Binary
\$L\$51	Patung Rusa 3 Solusi	0	1	Binary
\$L\$52	Patung Rusa 2 Solusi	0	1	Binary
\$L\$53	Patung Rusa 1 Solusi	0	1	Binary
\$L\$54	Patung Domba 3 Solusi	0	0	Binary
\$L\$55	Patung Domba 2 Solusi	0	1	Binary
\$L\$56	Patung Domba 1 Solusi	0	1	Binary

**Lampiran C. Hasil Percobaan Uji Parameter****C.1 Uji Parameter  $nClan$  data 1** $nClan = 5, nCi = 5$ 

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	3375	8024075,711	19675500	2714000	107	102,8301
2	3375	8024075,711	19675500	2714000	115	98,0326
3	3375	8024075,711	19675500	2714000	36	98,2219
4	3375	8024075,711	19675500	2714000	45	97,1951
5	3375	8024075,711	19675500	2714000	268	116,6108
6	3375	8024075,711	19675500	2714000	260	97,689
7	3375	8024075,711	19675500	2714000	233	97,1456
8	3375	8024075,711	19675500	2714000	107	98,0117
9	3375	8024075,711	19675500	2714000	158	98,0713
10	3375	8024075,711	19675500	2714000	122	98,8113
Rata-rata				2714000	145,1	100,26194

 $nClan = 10, nCi = 5$ 

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	3375	8024075,711	19675500	2714000	305	105,2411
2	3375	8024075,711	19675500	2714000	210	105,3468
3	3375	8024075,711	19675500	2714000	159	104,7178
4	3375	8024075,711	19675500	2714000	86	104,4057
5	3375	8024075,711	19675500	2714000	214	104,4262
6	3375	8024075,711	19675500	2714000	83	104,7412
7	3375	8024075,711	19675500	2714000	66	103,9587
8	3375	8024075,711	19675500	2714000	67	105,1503
9	3375	8024075,711	19675500	2714000	98	104,6793
10	3375	8024075,711	19675500	2714000	76	104,3338
Rata-rata				2714000	136,4	104,70009

 $nClan = 15, nCi = 5$ 

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	3375	8024075,711	19675500	2714000	61	114,8788
2	3375	8024075,711	19675500	2714000	95	110,929
3	3375	8024075,711	19675500	2714000	38	111,6249
4	3375	8024075,711	19675500	2714000	47	111,3633



Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
5	3375	8024075,711	19675500	2714000	81	110,4945
6	3375	8024075,711	19675500	2714000	55	111,466
7	3375	8024075,711	19675500	2714000	85	110,9079
8	3375	8024075,711	19675500	2714000	90	110,4896
9	3375	8024075,711	19675500	2714000	144	110,4812
10	3375	8024075,711	19675500	2714000	79	111,1437
Rata-rata				2714000	77,5	111,37789

$nClan = 20, nCi = 5$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	3375	8024075,711	19675500	2714000	42	116,0882
2	3375	8024075,711	19675500	2714000	77	115,9433
3	3375	8024075,711	19675500	2714000	84	117,3394
4	3375	8024075,711	19675500	2714000	57	115,9203
5	3375	8024075,711	19675500	2714000	55	116,5685
6	3375	8024075,711	19675500	2714000	63	116,4287
7	3375	8024075,711	19675500	2714000	79	115,8935
8	3375	8024075,711	19675500	2714000	56	116,111
9	3375	8024075,711	19675500	2714000	42	115,5792
10	3375	8024075,711	19675500	2714000	66	115,1083
Rata-rata	3375	8024075,711	19675500	2714000	62,1	116,09804

$nClan = 25, nCi = 5$

Percobaan	berat	volume	Harga Beli	profit	Iterasi Konvergen	Waktu komputasi
1	3375	8024075,711	19675500	2714000	69	120,5534
2	3375	8024075,711	19675500	2714000	43	121,2796
3	3375	8024075,711	19675500	2714000	27	121,0942
4	3375	8024075,711	19675500	2714000	39	120,3824
5	3375	8024075,711	19675500	2714000	53	119,8517
6	3375	8024075,711	19675500	2714000	55	121,155
7	3375	8024075,711	19675500	2714000	132	120,5623
8	3375	8024075,711	19675500	2714000	50	132,6868
9	3375	8024075,711	19675500	2714000	23	119,759
10	3375	8024075,711	19675500	2714000	82	121,0091
Rata-rata	3375	8024075,711	19675500	2714000	57,3	121,83335

C.2 Uji Parameter  $nClan$  data 2 $nClan = 5, nCi = 5$ 

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	425	6405391,75	9976500	14221000	345	95,7048
2	420	6212841,75	9996500	14251000	848	96,4425
3	420	6212841,75	9996500	14251000	729	96,1571
4	425	6268341,75	9986500	14261000	904	97,0564
5	412,5	6326730,5	9981500	14238500	404	97,7815
6	407,5	6295105,5	9969000	14221000	273	95,577
7	412,5	6280695,5	9976500	14231000	918	96,268
8	420	6135591,75	9966500	14231000	988	96,1459
9	412,5	6152926,75	9979000	14153500	499	95,816
10	402,5	6212355,5	9989000	14251000	248	96,7998
Rata-rata	415,75	6250282,25	9981750	14231000	615,6	96,3749

 $nClan = 10, nCi = 5$ 

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	422,5	6184185,5	9996500	14201000	387	103,7892
2	410	6217571,75	9968500	14229000	862	106,15
3	427,5	6154935,5	9996500	14251000	384	103,2244
4	397,5	6238011,75	9994000	14241000	463	101,5346
5	415	6370780,5	9981000	14231500	792	114,7919
6	405	6425646,75	9981500	14248500	369	103,0986
7	395	6232646,75	9981500	14248500	997	103,2107
8	420	6087435,5	9991500	14261000	689	106,2103
9	405	6238256,75	9989000	14258500	645	101,6365
10	420	6339327,5	9991500	14251000	711	102,1377
Rata-rata	411,75	6248879,825	9987150	14242100	629,9	104,57839

 $nClan = 15, nCi = 5$ 

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	395	6216146,75	9988500	14241500	481	110,6655
2	410	6227351,75	9976500	14221000	749	109,8864
3	405	6356886,75	9991500	14278500	865	110,8596
4	422,5	6405435,5	9986500	14261000	771	112,794

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
5	427,5	6286185,5	9986500	14211000	443	109,5751
6	407,5	6224195,5	9993500	14224000	443	110,0512
7	412,5	6329886,75	9996500	14268500	686	110,3461
8	410	6272216,75	9974000	14253500	955	110,1443
9	400	6160146,75	9971500	14228500	443	110,2668
10	412,5	6195351,75	9991500	14271000	809	109,4224
Rata-rata	410,25	6267380,375	9985650	14245850	664,5	110,40114

$nClan = 20, nCi = 5$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	405	6332511,75	9999000	14261000	263	117,9168
2	420	6157935,5	9991500	14231000	341	120,018
3	417,5	6379915,5	9998500	14249000	921	117,2668
4	415	6025851,75	9986500	14226000	945	116,5327
5	420	6190935,5	9991500	14261000	629	115,3403
6	425	6444591,75	9956500	14241000	697	116,8807
7	425	6173841,75	9986500	14261000	938	115,8886
8	422,5	6299091,75	9986500	14241000	710	114,5845
9	405	6332511,75	9999000	14291000	541	117,1295
10	422,5	6195591,75	9986500	14241000	878	116,2087
Rata-rata	417,75	6253277,875	9988200	14250300	686,3	116,77666

$nClan = 25, nCi = 5$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	410	6205386,75	9981500	14288500	695	122,156
2	425	6164841,75	9986500	14261000	734	122,5005
3	430	6280435,5	9991500	14261000	876	121,3019
4	410	6308886,75	9981500	14288500	527	122,4898
5	430	6185935,5	9991500	14261000	792	123,0495
6	415	6409851,75	9986500	14271000	571	122,597
7	427,5	6262435,5	9976500	14271000	812	122,2051
8	400	6286256,75	9999000	14248500	924	122,7796
9	405	6332511,75	9999000	14291000	717	127,5321
10	405	6331146,75	9981500	14248500	711	126,9359
Rata-rata	415,75	6276768,875	9987500	14269000	735,9	123,35474

C.3 Uji Parameter  $nCi$  data 1 $nClan = 25, nCi = 5$ 

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	3375	8024075,711	19675500	2714000	69	120,5534
2	3375	8024075,711	19675500	2714000	43	121,2796
3	3375	8024075,711	19675500	2714000	27	121,0942
4	3375	8024075,711	19675500	2714000	39	120,3824
5	3375	8024075,711	19675500	2714000	53	119,8517
6	3375	8024075,711	19675500	2714000	55	121,155
7	3375	8024075,711	19675500	2714000	132	120,5623
8	3375	8024075,711	19675500	2714000	50	132,6868
9	3375	8024075,711	19675500	2714000	23	119,759
10	3375	8024075,711	19675500	2714000	82	121,0091
Rata-rata	3375	8024075,711	19675500	2714000	57,3	121,83335

 $nClan = 25, nCi = 10$ 

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	3375	8024075,711	19675500	2714000	60	126,8092
2	3375	8024075,711	19675500	2714000	59	132,6488
3	3375	8024075,711	19675500	2714000	23	127,8468
4	3375	8024075,711	19675500	2714000	41	126,2534
5	3375	8024075,711	19675500	2714000	48	126,5546
6	3375	8024075,711	19675500	2714000	41	126,8572
7	3375	8024075,711	19675500	2714000	72	129,3597
8	3375	8024075,711	19675500	2714000	19	126,6849
9	3375	8024075,711	19675500	2714000	36	128,3776
10	3375	8024075,711	19675500	2714000	47	130,4947
Rata-rata				2714000	44,6	128,18869

 $nClan = 25, nCi = 15$ 

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	3375	8024075,711	19675500	2714000	16	151,0808
2	3375	8024075,711	19675500	2714000	35	141,6705

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
3	3375	8024075,711	19675500	2714000	41	142,2610
4	3375	8024075,711	19675500	2714000	28	174,9135
5	3375	8024075,711	19675500	2714000	9	146,0764
6	3375	8024075,711	19675500	2714000	21	145,5775
7	3375	8024075,711	19675500	2714000	88	140,9178
8	3375	8024075,711	19675500	2714000	11	155,9248
9	3375	8024075,711	19675500	2714000	46	144,8748
10	3375	8024075,711	19675500	2714000	21	138,9390
Rata-rata				2714000	31,6	148,22361

$nClan = 25, nCi = 20$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	3375	8024075,711	19675500	2714000	35	161,7638
2	3375	8024075,711	19675500	2714000	28	151,2682
3	3375	8024075,711	19675500	2714000	29	153,192
4	3375	8024075,711	19675500	2714000	15	157,6913
5	3375	8024075,711	19675500	2714000	16	155,0133
6	3375	8024075,711	19675500	2714000	7	155,2954
7	3375	8024075,711	19675500	2714000	25	158,3650
8	3375	8024075,711	19675500	2714000	46	152,4721
9	3375	8024075,711	19675500	2714000	26	150,1311
10	3375	8024075,711	19675500	2714000	19	151,4495
Rata-rata				2714000	24,6	154,66417

$nClan = 25, nCi = 25$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	3375	8024075,711	19675500	2714000	8	178,7802
2	3375	8024075,711	19675500	2714000	13	166,5545
3	3375	8024075,711	19675500	2714000	26	167,1527
4	3375	8024075,711	19675500	2714000	10	166,4182
5	3375	8024075,711	19675500	2714000	12	164,7549
6	3375	8024075,711	19675500	2714000	17	168,8411
7	3375	8024075,711	19675500	2714000	14	170,8227
8	3375	8024075,711	19675500	2714000	22	169,4076

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
9	3375	8024075,711	19675500	2714000	40	171,6681
10	3375	8024075,711	19675500	2714000	9	172,579
Rata-rata				2714000	17,1	169,6979

#### C.4 Uji Parameter $nCi$ data 2

$nClan = 25, nCi = 5$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	410	6205387	9981500	14288500	695	122,156
2	425	6164842	9986500	14261000	734	122,5005
3	430	6280436	9991500	14261000	876	121,3019
4	410	6308887	9981500	14288500	527	122,4898
5	430	6185936	9991500	14261000	792	123,0495
6	415	6409852	9986500	14271000	571	122,597
7	427,5	6262436	9976500	14271000	812	122,2051
8	400	6286257	9999000	14248500	924	122,7796
9	405	6332512	9999000	14291000	717	127,5321
10	405	6331147	9981500	14248500	711	126,9359
Rata-rata				14269000	735,9	123,35474

$nClan = 25, nCi = 10$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	425	6173842	9986500	14261000	642	137,5098
2	395	6327147	9981500	14248500	910	142,7873
3	405	6357387	9991500	14278500	763	144,4767
4	430	6289436	9991500	14271000	339	142,9793
5	405	6341947	9999000	14266000	800	143,6132
6	402,5	6196279	9996500	14268500	377	142,8637
7	412,5	6281352	9991500	14271000	259	143,3955
8	402,5	6187779	9996500	14268500	619	138,9994
9	410	6308887	9981500	14288500	816	139,207
10	405	6238012	9999000	14291000	869	138,7173
Rata-rata				14271250	639,4	141,45492

$nClan = 25, nCi = 15$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	417,5	6272887	9986500	14278500	703	158,5905
2	402,5	6196279	9996500	14268500	599	159,6707
3	417,5	6272887	9986500	14278500	461	159,125
4	405	6238012	9999000	14291000	947	158,2526
5	422,5	6405436	9986500	14261000	940	157,5251
6	405	6238012	9999000	14291000	648	160,5026
7	430	6289436	9991500	14261000	235	157,5767
8	412,5	6281352	9991500	14271000	286	158,5744
9	417,5	6281887	9986500	14278500	359	157,8622
10	395	6113272	9989000	14261000	326	157,5191
Rata-rata				14274000	550,4	158,51989

$nClan = 25, nCi = 20$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	427,5	6262436	9976500	14271000	934	173,4683
2	427,5	6262436	9976500	14271000	617	173,958
3	417,5	6281887	9986500	14278500	918	172,4913
4	407,5	6183387	9986500	14278500	610	171,9133
5	405	6229012	9999000	14291000	782	173,3755
6	405	6238012	9999000	14291000	180	172,0673
7	410	6205387	9981500	14288500	220	173,2275
8	412,5	6186852	9991500	14271000	157	179,2508
9	407,5	6183387	9986500	14278500	320	175,6684
10	417,5	6272887	9986500	14278500	798	173,5942
Rata-rata				14279750	553,6	173,90146

$nClan = 25, nCi = 25$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	410	6205387	9981500	14288500	243	190,6957
2	410	6214387	9981500	14288500	809	189,4324
3	405	6238012	9999000	14291000	437	185,9757
4	410	6214387	9981500	14288500	554	185,636

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
5	405	6238012	9999000	14291000	244	189,9707
6	405	6238012	9999000	14291000	818	190,0159
7	405	6332512	9999000	14291000	835	187,7109
8	405	6238012	9999000	14291000	661	190,9889
9	407,5	6183387	9986500	14278500	728	190,4399
10	405	6229012	9999000	14291000	537	188,116
Rata-rata				14289000	586,6	188,89821

### C.5 Uji Parameter $\alpha$ dan $\beta$ data 1

$\alpha = 0,1$  dan  $\beta = 0,1$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	3375	8024075,711	19675500	2714000	52	163,1541
2	3375	8024075,711	19675500	2714000	48	163,3398
3	3375	8024075,711	19675500	2714000	39	161,4633
4	3375	8024075,711	19675500	2714000	52	163,1541
5	3375	8024075,711	19675500	2714000	34	160,8763
6	3375	8024075,711	19675500	2714000	49	162,3663
7	3375	8024075,711	19675500	2714000	40	164,3107
8	3375	8024075,711	19675500	2714000	42	161,2821
9	3375	8024075,711	19675500	2714000	50	164,7857
10	3375	8024075,711	19675500	2714000	41	163,9274
Rata-rata				2714000	44,7	162,86598

$\alpha = 0,3$  dan  $\beta = 0,3$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	3375	8024075,711	19675500	2714000	45	177,128
2	3375	8024075,711	19675500	2714000	23	168,1432
3	3375	8024075,711	19675500	2714000	18	169,2931
4	3375	8024075,711	19675500	2714000	45	167,8703
5	3375	8024075,711	19675500	2714000	14	165,5169
6	3375	8024075,711	19675500	2714000	46	166,0217
7	3375	8024075,711	19675500	2714000	23	169,3595



Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
8	3375	8024075,711	19675500	2714000	14	169,0914
9	3375	8024075,711	19675500	2714000	19	165,5268
10	3375	8024075,711	19675500	2714000	12	165,7245
Rata-rata				2714000	25,9	168,36754

$\alpha = 0,5$  dan  $\beta = 0,5$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	3375	8024075,711	19675500	2714000	8	178,7802
2	3375	8024075,711	19675500	2714000	13	166,5545
3	3375	8024075,711	19675500	2714000	26	167,1527
4	3375	8024075,711	19675500	2714000	10	166,4182
5	3375	8024075,711	19675500	2714000	12	164,7549
6	3375	8024075,711	19675500	2714000	17	168,8411
7	3375	8024075,711	19675500	2714000	14	170,8227
8	3375	8024075,711	19675500	2714000	22	169,4076
9	3375	8024075,711	19675500	2714000	40	171,6681
10	3375	8024075,711	19675500	2714000	9	172,579
Rata-rata				2714000	17,1	169,6979

$\alpha = 0,7$  dan  $\beta = 0,7$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	3375	8024075,711	19675500	2714000	29	174,8086
2	3375	8024075,711	19675500	2714000	11	164,1999
3	3375	8024075,711	19675500	2714000	16	166,1206
4	3375	8024075,711	19675500	2714000	19	167,8562
5	3375	8024075,711	19675500	2714000	26	170,9315
6	3375	8024075,711	19675500	2714000	18	170,6683
7	3375	8024075,711	19675500	2714000	27	168,6277
8	3375	8024075,711	19675500	2714000	10	168,6427
9	3375	8024075,711	19675500	2714000	12	168,2776
10	3375	8024075,711	19675500	2714000	15	170,09
Rata-rata				2714000	18,3	169,02231

$\alpha = 0,9$  dan  $\beta = 0,9$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	3375	8024075,711	19675500	2714000	6	179,2376
2	3375	8024075,711	19675500	2714000	47	172,5542
3	3375	8024075,711	19675500	2714000	41	171,2906
4	3375	8024075,711	19675500	2714000	22	169,6437
5	3375	8024075,711	19675500	2714000	6	172,6122
6	3375	8024075,711	19675500	2714000	7	170,6813
7	3375	8024075,711	19675500	2714000	72	170,1605
8	3375	8024075,711	19675500	2714000	5	173,5638
9	3375	8024075,711	19675500	2714000	53	172,8922
10	3375	8024075,711	19675500	2714000	54	167,4231
Rata-rata				2714000	31,3	172,00592

#### C.6 Uji Parameter $\alpha$ dan $\beta$ data 2

$\alpha = 0,1$  dan  $\beta = 0,1$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	410	6205387	9981500	14288500	327	174,246
2	407,5	6363701	9993500	14216500	826	173,109
3	395	6216772	9989000	14261000	254	171,571
4	420	6159352	9996500	14261000	279	171,8593
5	405	6238012	9999000	14291000	263	173,2064
6	405	6332512	9999000	14291000	688	172,2956
7	422,5	6310436	9986500	14261000	71	172,3803
8	407,5	6163012	9994000	14246000	996	173,1725
9	407,5	6172012	9994000	14246000	741	171,9973
10	395	6381039	9981500	14238500	127	172,1132
Rata-rata				14260050	457,2	172,59506

$\alpha = 0,3$  dan  $\beta = 0,3$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	405	6229012	9999000	14291000	456	180,3333
2	412,5	6091852	9991500	14271000	654	182,8758

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
3	415	6305092	9996500	14241000	605	180,8259
4	405	6238012	9999000	14291000	683	179,5241
5	420	6149828	9991500	14251000	83	180,5285
6	427,5	6262436	9976500	14271000	387	180,7756
7	430	6280436	9991500	14261000	999	180,9483
8	417,5	6272887	9986500	14278500	439	178,0926
9	425	6141592	9996500	14256000	135	180,4904
10	412,5	6240244	9991500	14261000	881	180,137
Rata-rata				14267250	532,2	180,45311

$\alpha = 0,5$  dan  $\beta = 0,5$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	410	6205387	9981500	14288500	243	190,6957
2	410	6214387	9981500	14288500	809	189,4324
3	430	6185936	9991500	14261000	268	190,0388
4	402,5	6282279	9996500	14268500	331	188,9073
5	405	6238012	9999000	14291000	244	189,9707
6	405	6238012	9999000	14291000	818	190,0159
7	405	6332512	9999000	14291000	835	187,7109
8	405	6238012	9999000	14291000	661	190,9889
9	407,5	6183387	9986500	14278500	728	190,4399
10	405	6229012	9999000	14291000	537	188,116
Rata-rata				14284000	547,4	189,63165

$\alpha = 0,7$  dan  $\beta = 0,7$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	405	6356887	9991500	14278500	579	188,6387
2	405	6229012	9999000	14291000	494	199,1203
3	410	6308887	9981500	14288500	440	184,7334
4	405	6332512	9999000	14291000	680	190,8893
5	412,5	6186352	9991500	14271000	565	186,1716
6	407,5	6079887	9986500	14278500	356	195,0959
7	405	6229012	9999000	14291000	473	189,6294

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
8	410	6214387	9981500	14288500	837	197,9919
9	405	6229012	9999000	14291000	340	186,9668
10	405	6253387	9991500	14278500	707	186,4658
Rata-rata				14284750	547,1	190,57031

$\alpha = 0,9$  dan  $\beta = 0,9$

Percobaan	Berat	Volume	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi
1	410	6205387	9981500	14288500	402	187,9485
2	405	6238012	9999000	14291000	899	204,7156
3	410	6308887	9981500	14288500	830	186,9351
4	405	6332512	9999000	14291000	507	184,8684
5	405	6229012	9999000	14291000	664	191,9905
6	417,5	6178387	9986500	14278500	586	195,9127
7	405	6229012	9999000	14291000	789	190,6737
8	405	6238012	9999000	14291000	836	191,0514
9	410	6308887	9981500	14288500	982	191,7723
10	405	6238012	9999000	14291000	335	197,031
Rata-rata				14289000	683	192,28992

#### Lampiran D. Skrip Program EHO

```
tic;
%Data
data001 = get(handles.uitable1, 'UserData');
data=data001{1};
NamaBarang=data001{2};
Kapasitas=str2num(get(handles.edit6, 'string'));
Modal=str2num(get(handles.edit7, 'string'));
dim=size(data,1);

%Parameter
nClan=str2num(get(handles.edit1, 'string'));
nci=str2num(get(handles.edit2, 'string'));
alpha=str2num(get(handles.edit3, 'string'));
beta=str2num(get(handles.edit4, 'string'));
MaxGen=str2num(get(handles.edit5, 'string'));

%Inisialisasi
X=rand(nClan*nci, dim);
Y=round(X);
for i=1:nClan
```

```

    for j=1:nci
        Berat(i,j)=sum(Y((i-1)*nci+j,:).*(data(:,1).*data(:,2))');
        Volume(i,j)=sum(Y((i-
1)*nci+j,:).*(data(:,1).*data(:,3))');
        Beli(i,j)=sum(Y((i-1)*nci+j,:).*(data(:,1).*data(:,4))');
        while Berat(i,j)>Kapasitas(1) || Volume(i,j)>Kapasitas(2)
|| Beli(i,j)>Modal
            pss=find(Y((i-1)*nci+j,:)==1);
            r1=ceil(rand*length(pss));
            X((i-1)*nci+j,pss(r1))=X((i-1)*nci+j,pss(r1))/2;
            Y((i-1)*nci+j,:)=round(X((i-1)*nci+j,:));
            Berat(i,j)=sum(Y((i-
1)*nci+j,:).*(data(:,1).*data(:,2))');
            Volume(i,j)=sum(Y((i-
1)*nci+j,:).*(data(:,1).*data(:,3))');
            Beli(i,j)=sum(Y((i-
1)*nci+j,:).*(data(:,1).*data(:,4))');
        end
        Fitness(i,j)=sum(Y((i-1)*nci+j,:).*(data(:,1).*data(:,5)-
data(:,4))');
    end
    best=find(Fitness(i,:)==max(Fitness(i,:)));
    Matriarch(i,:)=X((i-1)*nci+best(1),:);
    FitMat(i)=max(Fitness(i,:));
end

bsf(1)=max(FitMat);
inon=0;
t=0;
while t<MaxGen
    %Update posisi
    for i=1:nClan
        Xctr=sum(X((i-1)*nci+(1:nci),:))/nci;
        X((i-1)*nci+(1:nci),:)=X((i-
1)*nci+(1:nci),:)+alpha*(repmat(Matriarch(i,:),nci,1)-X((i-
1)*nci+(1:nci),:)).*rand(nci,dim);
        for j=1:nci
            if Y((i-1)*nci+j,:)==round(Matriarch(i,:))
                X((i-1)*nci+j,:)=X((i-
1)*nci+j,:)+beta*Xctr.*rand(1,dim);
            end
            if min(X((i-1)*nci+j,:))<0
                X((i-1)*nci+j,:)=(X((i-1)*nci+j,:)-min(X((i-
1)*nci+j,:)))/(max(X((i-1)*nci+j,:))-min(X((i-1)*nci+j,:)));
            else
                X((i-1)*nci+j,:)=X((i-1)*nci+j,:)/max(X((i-
1)*nci+j,:));
            end
            Y((i-1)*nci+j,:)=round(X((i-1)*nci+j,:));
            Berat(i,j)=sum(Y((i-
1)*nci+j,:).*(data(:,1).*data(:,2))');
            Volume(i,j)=sum(Y((i-
1)*nci+j,:).*(data(:,1).*data(:,3))');
            Beli(i,j)=sum(Y((i-
1)*nci+j,:).*(data(:,1).*data(:,4))');

```

```

        while Berat(i,j)>Kapasitas(1) ||
Volume(i,j)>Kapasitas(2) || Beli(i,j)>Modal
        pss=find(Y((i-1)*nci+j, :)==1);
        r1=ceil(rand*length(pss));
        X((i-1)*nci+j, pss(r1))=X((i-1)*nci+j, pss(r1))/2;
        Y((i-1)*nci+j, :)=round(X((i-1)*nci+j, :));
        Berat(i,j)=sum(Y((i-
1)*nci+j, :).*(data(:,1).*data(:,2))');
        Volume(i,j)=sum(Y((i-
1)*nci+j, :).*(data(:,1).*data(:,3))');
        Beli(i,j)=sum(Y((i-
1)*nci+j, :).*(data(:,1).*data(:,4))');
        end
        Fitness(i,j)=sum(Y((i-
1)*nci+j, :).*(data(:,1).(data(:,5)-data(:,4))));
        end
        %Replace worst
        worst=find(Fitness(i, :)==min(Fitness(i, :)));
        if length(worst)==nci
            worst=ceil(0.75*nci):nci;
        end
        X((i-1)*nci+worst, :)=repmat(min(X((i-
1)*nci+(1:nci), :)), length(worst), 1)+...
            repmat((max(X((i-1)*nci+(1:nci), :))-min(X((i-
1)*nci+(1:nci), :)))+1, length(worst), 1).*rand(length(worst), dim);
        for k=1:length(worst)
            if min(X((i-1)*nci+worst(k), :))<0
                X((i-1)*nci+worst(k), :)=(X((i-1)*nci+worst(k), :)-
min(X((i-1)*nci+worst(k), :)))/(max(X((i-1)*nci+worst(k), :))-
min(X((i-1)*nci+worst(k), :)));
            else
                X((i-1)*nci+worst(k), :)=X((i-
1)*nci+worst(k), :)/max(X((i-1)*nci+worst(k), :));
            end
            Y((i-1)*nci+worst(k), :)=round(X((i-
1)*nci+worst(k), :));
            Berat(i, worst(k))=sum(Y((i-
1)*nci+worst(k), :).*(data(:,1).*data(:,2))');
            Volume(i, worst(k))=sum(Y((i-
1)*nci+worst(k), :).*(data(:,1).*data(:,3))');
            Beli(i, worst(k))=sum(Y((i-
1)*nci+worst(k), :).*(data(:,1).*data(:,4))');
            while Berat(i, worst(k))>Kapasitas(1) ||
Volume(i, worst(k))>Kapasitas(2) || Beli(i, worst(k))>Modal
                pss=find(Y((i-1)*nci+worst(k), :)==1);
                r1=ceil(rand*length(pss));
                X((i-1)*nci+worst(k), pss(r1))=X((i-
1)*nci+worst(k), pss(r1))/2;
                Y((i-1)*nci+worst(k), :)=round(X((i-
1)*nci+worst(k), :));
                Berat(i, worst(k))=sum(Y((i-
1)*nci+worst(k), :).*(data(:,1).*data(:,2))');
                Volume(i, worst(k))=sum(Y((i-
1)*nci+worst(k), :).*(data(:,1).*data(:,3))');
                Beli(i, worst(k))=sum(Y((i-
1)*nci+worst(k), :).*(data(:,1).*data(:,4))');

```

```

        end
        Fitness(i,worst(k))=sum(Y((i-
1)*nci+worst(k,:).*(data(:,1).*(data(:,5)-data(:,4)))'));
    end
    %Update Matriarch
    if max(Fitness(i,:))>FitMat(i)
        best=find(Fitness(i,:)==max(Fitness(i,:)));
        Matriarch(i,:)=X((i-1)*nci+best(1),:);
        FitMat(i)=max(Fitness(i,:));
    end
end

%Plot
bsf(t+2)=max(FitMat);
if bsf(t+2)~=bsf(t+1)
    inon=t+1;
end
plot(0:length(bsf)-1,bsf,'r','LineWidth',2);

line(inon,bsf(inon+1),'Marker','s','MarkerEdgeColor','k','MarkerFaceColor','b','MarkerSize',6);
if inon<t*0.75
    text(inon,bsf(inon+1),sprintf(['\n\n Iter: ' num2str(inon)
'\n Fit: ' sprintf('%12d',bsf(inon+1)) ',-' ]));
else
    text(inon,bsf(inon+1),sprintf([' Iter: ' num2str(inon) '\n
Fit: ' sprintf('%12d',bsf(inon+1)) ',-'
'\n\n']),'HorizontalAlignment','Right');
end
ylim([min(bsf) max(bsf)*1.08]);
set(handles.axes1,'XColor',[1 1 1],'YColor',[1 1 1],
'FontSize',8,'FontWeight','bold');
xlabel('Iterasi'); ylabel('Total Profit (Fitness)');
pause(0.00001);

%Next iter
t=t+1;
end
best=find(FitMat==max(FitMat));
FinalSol=round(Matriarch(best(1),:));
%Hasil
for i=1:dim
    output{i,1}=NamaBarang{i};
    output{i,2}=FinalSol(i)*data(i,1);
    output{i,3}=output{i,2}*data(i,2);
    output{i,4}=output{i,2}*data(i,3);
    output{i,5}=output{i,2}*data(i,4);
    output{i,6}=output{i,2}*(data(i,5)-data(i,4));
end
output{dim+2,1}='Total';
output{dim+2,2}=sum(FinalSol.*data(i,1)');
output{dim+2,3}=sum(FinalSol.*(data(:,1).*data(i,2))');
output{dim+2,4}=sum(FinalSol.*(data(:,1).*data(:,3))');
output{dim+2,5}=sum(FinalSol.*(data(:,1).*data(:,4))');
output{dim+2,6}=sum(FinalSol.*(data(:,1).*(data(:,5)-
data(:,4))))');

```

```
set(handles.uitable2,'data',output,'userdata',FinalSol,...  
    'columnname',{ 'Nama  
Barang','Jumlah','T.Berat','T.Volume','T.Beli','T.Profit'},'rowname',  
    'numbered',...  
    'columnwidth',{100,60,60,60,80,80});  
set(handles.text10,'string',[num2str(toc) ' detik']);  
set(handles.axes1,'userdata',inon);  
set(handles.pushbutton4,'enable','on');
```

