



**PENERAPAN *DRAGONFLY OPTIMIZATION ALGORITHM*
(DOA) PADA PERMASALAHAN *MULTIPLE
CONSTRAINTS BOUNDED KNAPSACK*
(Studi Kasus : Kerajinan Bambu Hitam Desa Pujerbaru
Kecamatan Maesan Kabupaten Bondowoso)**

SKRIPSI

Oleh

Laylatul Febriana Nilasari

NIM 151810101011

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

2020



**PENERAPAN *DRAGONFLY OPTIMIZATION ALGORITHM*
(DOA) PADA PERMASALAHAN *MULTIPLE
CONSTRAINTS BOUNDED KNAPSACK*
(Studi Kasus : Kerajinan Bambu Hitam Desa Pujerbaru
Kecamatan Maesan Kabupaten Bondowoso)**

SKRIPSI

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

Oleh

Laylatul Febriana Nilasari

NIM 151810101011

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

2020

PERSEMBAHAN

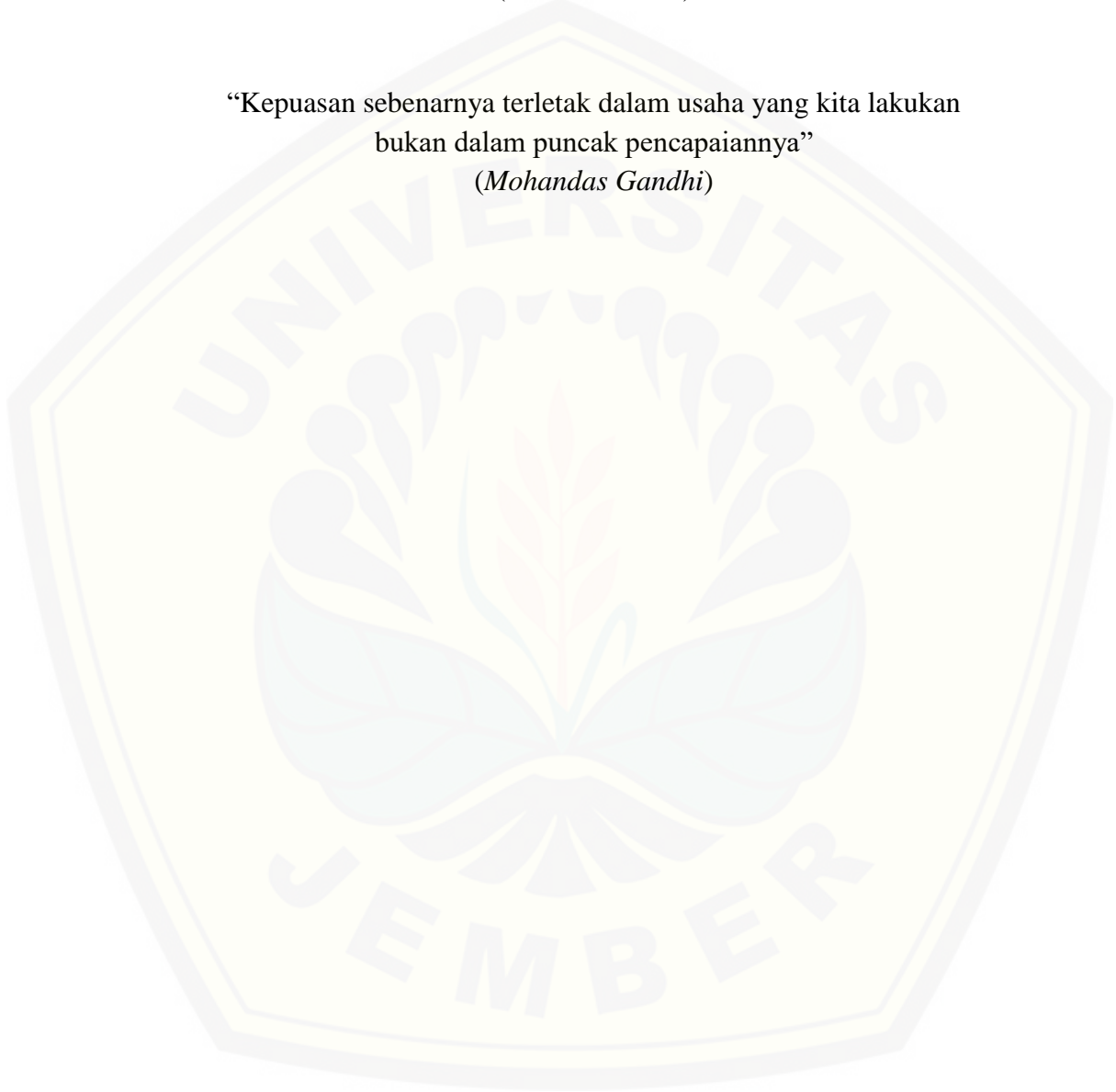
Alhamdulillah wasyukurillah, dengan segala puji bagi Allah SWT karena dengan limpahan rahmat dan hidayahnya saya dapat menyelesaikan skripsi ini dengan lancar. Skripsi ini saya persembahkan untuk:

1. Orang tua Ibunda Uun Endang Urbaniwati dan Ayahanda Samsi tercinta yang telah memberikan dukungan, semangat, kasih sayang, perhatian, dan pengorbanan yang begitu besar serta doa yang tak pernah putus untuk anaknya;
2. Guru-guru sejak dari sekolah dasar sampai dengan perguruan tinggi yang telah mendidik dan membimbing selama ini;
3. Almamater tercinta Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTTO

“Jika kamu ingin hidup bahagia, maka terikatlah pada tujuan
bukan orang atau benda”
(*Albert Einstein*)

“Kepuasan sebenarnya terletak dalam usaha yang kita lakukan
bukan dalam puncak pencapaiannya”
(*Mohandas Gandhi*)



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Laylatul Febriana Nilasari

NIM : 151810101011

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul "Penerapan *Dragonfly Optimization Algorithm* (DOA) Pada Permasalahan *Multiple Constraints Bounded Knapsack* pada Kasus Kerajinan Bambu Hitam Desa Pujer Baru Kecamatan Maesan Kabupaten Bondowoso" adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, 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 sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Januari 2020

Yang menyatakan,

Laylatul Febriana Nilasari

NIM 151810101011

SKRIPSI

**PENERAPAN *DRAGONFLY OPTIMIZATION ALGORITHM* (DOA) PADA
PERMASALAHAN *MULTIPLE CONSTRAINTS BOUNDED KNAPSACK***

**(Studi Kasus : Kerajinan Bambu Hitam Desa Pujer Baru Kecamatan
Maesan Kabupaten Bondowoso)**

Oleh

Laylatul Febriana Nilasari
NIM 15181010111

Pembimbing

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

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

PENGESAHAN

Skripsi berjudul ” Penerapan *Dragonfly Optimization Algorithm* (DOA) Pada Permasalahan *Multiple Constraints Bounded Knapsack* pada Kasus Kerajinan Bambu Hitam Desa Pujer Baru Kecamatan Maesan Kabupaten Bondowoso” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan
Alam Universitas Jember

Tim Penguji:

Ketua,

Anggota I,

Dr. Kiswara A. Santoso, S.Si., M.Kom.
NIP. 19720971998031003

Abduh Riski, S.Si., M.Si.
NIP. 199004062015041001

Anggota II,

Anggota III,

Kosala Dwidja Purnomo, S.Si., M.Si.
NIP. 196908281998021001

Prof. Drs. I Made Tirta, M.Sc., Ph.D.
NIP. 195912201985031002

Mengesahkan
Dekan,

Drs. Achmad Sjaifullah, M.Sc., Ph.D.
NIP. 195910091986021001

RINGKASAN

Penerapan *Dragonfly Optimization Algorithm* (DOA) Pada Permasalahan *Multiple Constraints Bounded Knapsack* pada Kasus Kerajinan Bambu Hitam Desa Pujer Baru Kecamatan Maesan Kabupaten Bondowoso; Laylatul Febriana Nilasari; 151810101011; 2020; 78 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Masalah *Knapsack* merupakan suatu permasalahan yang berhubungan dengan penyimpanan objek ke dalam media penyimpanan yang terbatas. Salah satu jenis permasalahan *knapsack* yaitu *Bounded Knapsack* yang merupakan permasalahan dimana objek yang dimasukkan ke dalam media penyimpanan dimensinya bisa dimasukkan sebagian atau seluruhnya namun jumlah objeknya terbatas (ketersediaan barang). Terdapat variasi dari permasalahan *knapsack* salah satunya yaitu *Multidimensional* atau *Multiple Constraints Knapsack Problem* merupakan permasalahan yang memiliki kendala lebih dari 1 untuk dapat memaksimalkan keuntungan yang diperoleh.

Penelitian ini menyelesaikan tentang Penerapan *Dragonfly Optimization Algorithm* (DOA) pada Permasalahan *Multiple Constraints Bounded Knapsack*. Data yang digunakan data primer berupa data penjualan dari rumah produksi kerajinan Bambu Hitam di Desa Pujerbaru, Kecamatan Maesan, Kabupaten Bondowoso. Kerajinan Bambu Hitam memproduksi berbagai macam jenis kerajinan yang berasal dari bambu hitam yang kemudian dibentuk dan diukir sesuai yang di inginkan. Data yang diambil berupa nama barang/benda, volume barang/benda, berat barang/benda, jumlah barang/benda, biaya produksi, harga jual dan keuntungan.

Terdapat sepuluh parameter dari DOA, yaitu parameter populasi (N_{pop}), maksimal iterasi (Max_{gen}), w_{max} , w_{min} , s , a , c , f , e dan $Range$ (R). Parameter tersebut digunakan untuk mengetahui hasil dari DOA yaitu berupa solusi optimal yang didapatkan dari beberapa percobaan. Hasil yang didapatkan dari DOA nantinya akan dibandingkan dengan hasil dari metode *Simplex*.

Berdasarkan keseluruhan hasil uji parameter, dapat diketahui bahwa nilai terbaik dari parameter-parameternya tidak terlalu besar dan tidak terlalu kecil. Jika nilai terbaiknya terlalu besar maka posisi capung akan teracak sehingga tidak jelas posisi dari capung dan jika terlalu kecil nilai terbaiknya maka perubahannya tidak terlihat.

Berdasarkan hasil simulasi akhir data dengan persentase deviasi yang menggunakan nilai parameter terbaik dari masing-masing parameter, DOA belum mampu mencapai nilai optimum *Simplex*, namun persentasenya kecil, artinya hasil DOA mendekati optimal. Profit metode *Simplex* pada data kerajinan adalah sebesar Rp 12.526.500,-. Profit terkecil yang mampu dihasilkan oleh DOA sebesar Rp 12.030.500,- dengan persentase deviasi sebesar 0,0396%, sedangkan profit terbesar yang didapat yaitu sebesar Rp 12.141.500,- dengan persentase deviasi sebesar 0,0307%. Adapun rata-rata profit dari hasil simulasi akhir data yaitu sebesar Rp 12.095.750,-, rata-rata persentase deviasi sebesar 0,0344%, dan rata-rata iterasi konvergen yaitu 592.

Berdasarkan uraian di atas, dapat dikatakan bahwa DOA kurang efektif untuk menyelesaikan permasalahan *multiple constraints bounded knapsack*, karena dari banyak percobaan tidak ada solusi yang mendekati *Simplex*. Solusi mendekati optimal yang didapatkan dari data kerajinan memiliki keuntungan sebesar Rp 12.141.500,- dengan total berat 381,75 kg, total volume 8.786.031 cm³, biaya produksi sebesar Rp 19.948.500,- dan waktu komputasi 461,1954 detik.

PRAKATA

Puji syukur ke hadirat Allah SWT, atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul " Penerapan *Dragonfly Optimization Algorithm* (Doa) Pada Permasalahan *Multiple Constraints Bounded Knapsack* pada Kasus Kerajinan Bambu Hitam Desa Pujer Baru Kecamatan Maesan Kabupaten Bondowoso ". Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

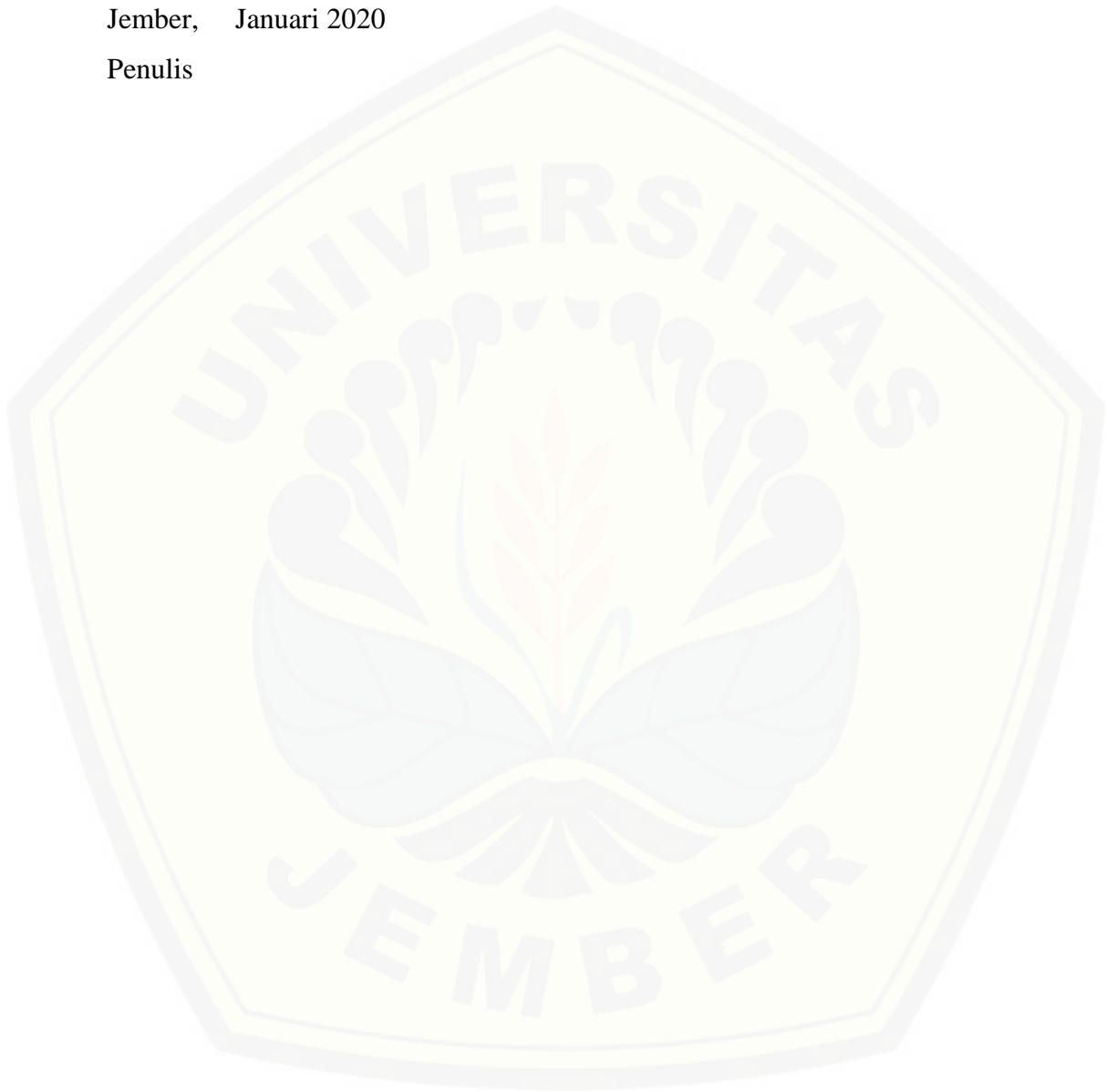
Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Bapak Dr. Kiswara Agung Santoso, S.Si., M.Kom. selaku Dosen Pembimbing Utama dan Bapak Abduh Riski, S.Si.,M.Si. selaku Dosen Pembimbing Anggota yang telah membimbing dan memberikan arahan kepada penulis;
2. Bapak Kosala Dwidja Purnomo, S,Si.,M.Si. dan Bapak Prof. Drs. I Made Tirta, M.Sc., Ph.D. selaku Dosen Penguji yang telah memberikan masukan, saran dan kritik yang membangun dalam penyusunan skripsi ini;
3. Seluruh dosen dan karyawan Jurusan Matematika Fakultas MIPA Universitas Jember;
4. Seluruh Anggota HIMATIKA yang telah memberikan ilmu dan pengalaman selama berorganisasi;
5. Adik Rilo dan Biila yang telah memberikan dukungan untuk tetap semangat;
6. Dyan Evi Susanti, Ulil Abshari, Sutitah, Rory Ronella Agustin atas keceriaan dan motivasinya;
7. Wangen Citro sebagai patner riset yang selalu mendukung satu sama lain;
8. Pintan Qorina Destianingrum dan Ucik Prastasiwi yang seperti keluarga selama di Jember;
9. Teman-teman seperjuangan Sigma Fifteen angkatan 2015 di Jurusan Matematika dan 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, Januari 2020

Penulis



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
PERSEMBAHAN.....	i
MOTTO	iii
PERNYATAAN.....	iv
SKRIPSI.....	v
HALAMAN PEMBIMBINGAN.....	v
PENGESAHAN	vi
RINGKASAN	vii
PRAKATA	ix
DAFTAR ISI.....	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN	xvi
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Manfaat	3
BAB 2. TINJAUAN PUSTAKA.....	4
2.1 Optimasi	4
2.2 Knapsack Problem	4
2.3 Bounded Knapsack.....	6
2.4 Multiple Constraints Bounded Knapsack (MCBK).....	7
2.5 Algoritma.....	8
2.6 <i>Dragonfly Optimization Algorithm (DOA)</i>	8
2.7 Metode Simpleks.....	15
2.8 Penggunaan Simpleks pada Excel.....	18
BAB 3. METODE PENELITIAN	20

3.1	Data Penelitian.....	20
3.2	Langkah-langkah Penelitian.....	20
BAB 4.	HASIL DAN PEMBAHASAN	25
4.1	Hasil Penelitian	25
4.1.1	Hasil Program	25
4.1.2	Simulasi Program.....	26
4.2	Pembahasan	32
4.2.1	Perhitungan Manual	32
4.2.2	Pembahasan Hasil Simulasi	50
BAB 5.	KESIMPULAN DAN SARAN	53
5.1	Kesimpulan	53
5.2	Saran.....	53
	DAFTAR PUSTAKA	54
	LAMPIRAN.....	56

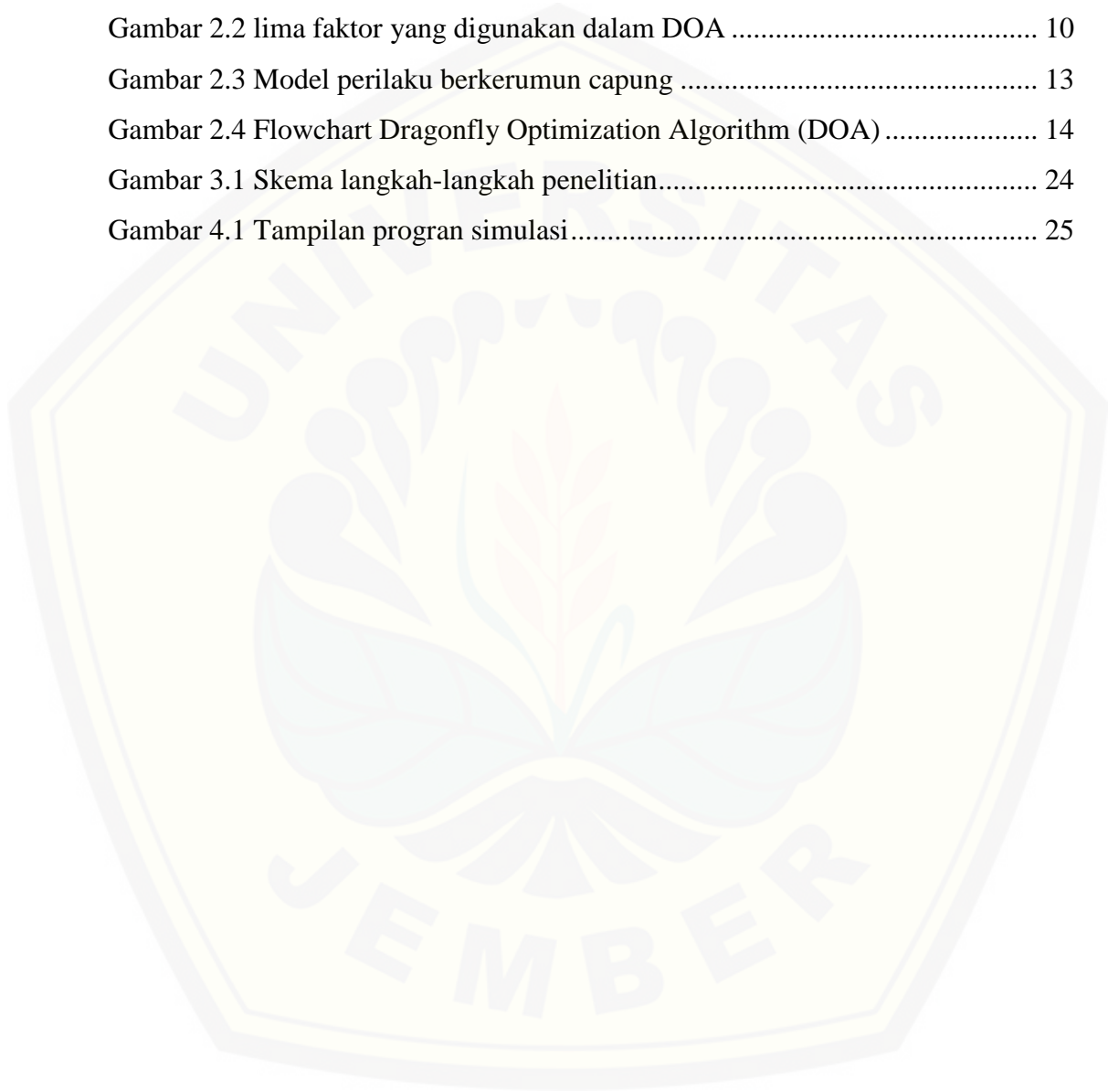
DAFTAR TABEL

	Halaman
Tabel 2.1 Tabel Simpleks pertama.....	16
Tabel 2.2 Memilih baris kunci	16
Tabel 2.3 Cara mengubah nilai baris kunci.....	17
Tabel 2.4 Mengubah nilai kolom dan baris kunci.....	17
Tabel 2.5 Melanjutkan perbaikan.....	18
Tabel 2.6 Final hasil perubahan	18
Tabel 4.1 Hasil uji parameter Populasi	27
Tabel 4.2 Hasil uji parameter Maximal Iterasi.....	27
Tabel 4.3 Hasil uji parameter w_{max}	28
Tabel 4.4 Hasil uji parameter w_{min}	28
Tabel 4.5 Hasil uji parameter berat pemisahan	29
Tabel 4.6 Hasil uji parameter berat keselarasan.....	29
Tabel 4.7 Hasil uji parameter berat kohesi.....	30
Tabel 4.8 Hasil uji parameter faktor makanan	30
Tabel 4.9 Hasil uji parameter faktor musuh.....	31
Tabel 4.10 Hasil uji parameter range	31
Tabel 4.11 Hasil Akhir Simulasi Data dengan Presentase Deviasi.....	32
Tabel 4.12 Data untuk Perhitungan Manual	32
Tabel 4.13 Nilai $random\ x$	33
Tabel 4.14 Nilai y	34
Tabel 4.15 Data keuntungan masing-masing barang	36
Tabel 4.16 Nilai $Random$ Individu ke 2,3,4 dan 5	38
Tabel 4.17 langkah menghitung r individu ke 2,3,4 dan 5.....	38
Tabel 4.18 Nilai S, A, C, F dan E	39
Tabel 4.19 Nilai $Random$ Individu ke 1,3,4 dan 5	39
Tabel 4.20 langkah menghitung r individu ke 1,3,4 dan 5.....	40
Tabel 4.21 S, A, C, F dan E	40
Tabel 4.22 Nilai $Random$ Individu ke 1,2,4 dan 5	41

Tabel 4.23 langkah menghitung r individu ke 1,2,4 dan 5.....	41
Tabel 4.24 Nilai <i>Random</i> Individu ke 1,2,3 dan 5	42
Tabel 4.25 langkah menghitung r individu ke 1,2,3 dan 5.....	42
Tabel 4.26 Nilai <i>S,A,C,F</i> dan <i>E</i>	43
Tabel 4.27 Nilai <i>Random</i> Individu ke 1,2,3 dan 4	44
Tabel 4.28 langkah menghitung r individu ke 1,2,3 dan 4.....	44
Tabel 4.29 Nilai x terbaru.....	45
Tabel 4.30 Nilai y terbaru.....	45
Tabel 4.31 Nilai x Setelah Dilakukan Pinalti	47
Tabel 4.32 Nilai y Setelah Dilakukan Pinalti	47
Tabel 4.33 Data Kerajinan yang dapat diangkut.....	52

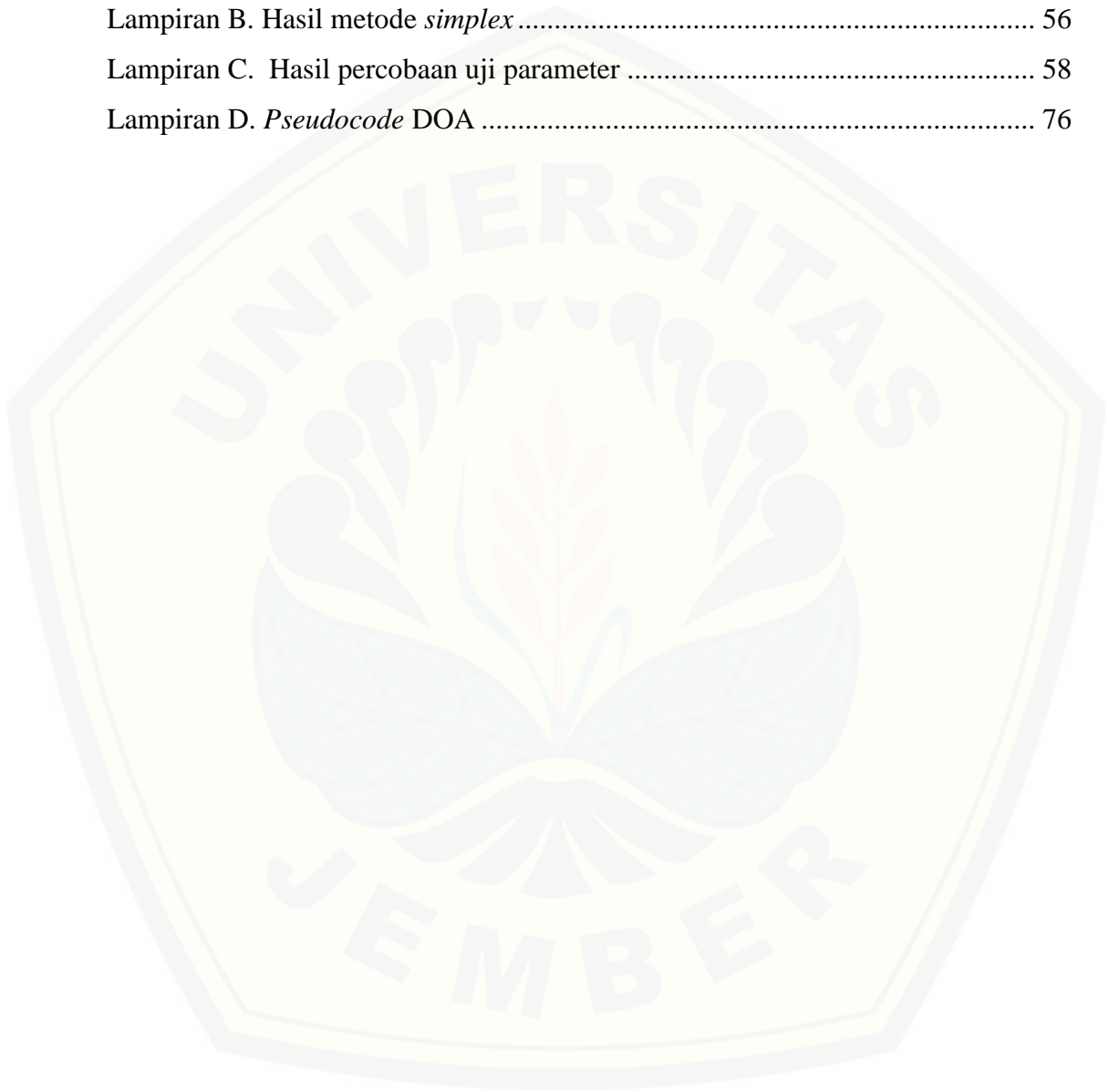
DAFTAR GAMBAR

	Halaman
Gambar 2.1 model konseptual dinamis dan statis	9
Gambar 2.2 lima faktor yang digunakan dalam DOA	10
Gambar 2.3 Model perilaku berkerumun capung	13
Gambar 2.4 Flowchart Dragonfly Optimization Algorithm (DOA)	14
Gambar 3.1 Skema langkah-langkah penelitian.....	24
Gambar 4.1 Tampilan program simulasi.....	25



DAFTAR LAMPIRAN

	Halaman
Lampiran A. Data penelitian	56
Lampiran B. Hasil metode <i>simplex</i>	56
Lampiran C. Hasil percobaan uji parameter	58
Lampiran D. <i>Pseudocode</i> DOA	76



BAB 1. PENDAHULUAN

1.1 Latar Belakang

Optimasi adalah salah satu disiplin ilmu dalam matematika yang fokus untuk mendapatkan nilai minimum atau maksimum secara sistematis dari suatu fungsi, peluang maupun pencarian nilai lainnya dalam berbagai kasus. Optimasi sangat berguna di hampir segala bidang dalam menjalankan usaha secara efektif dan efisien untuk mencapai target hasil yang ingin dicapai, sehingga optimasi sangat penting dalam persaingan di dunia industri yang sudah sangat ketat di segala bidang yang ada.

Masalah *Knapsack* merupakan suatu permasalahan yang berhubungan dengan penyimpanan objek ke dalam media penyimpanan yang terbatas. *Knapsack* merupakan suatu kantong atau tempat yang digunakan untuk memuat suatu 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 berdasarkan persoalannya, yaitu *0-1 knapsack problem*, *bounded knapsack problem* dan *unbounded knapsack problem*. Pengelompokan tersebut didasarkan pada pola penyimpanan barang dengan bobot dan nilai yang bervariasi. Permasalahan ini bertambah kompleks, ketika tiap-tiap pilihan yang ada masing-masing memiliki lebih dari satu dimensi batasan yang lebih dikenal *Multiple Constraints Knapsack Problem*. Pada bidang bisnis ekonomi, tiap-tiap objek memiliki lebih dari satu dimensi batasan sebagai bahan pertimbangan.

Penelitian sebelumnya mengenai *knapsack*, dilakukan oleh Hadi (2015) membahas tentang *bounded knapsack problem* diselesaikan dengan menggunakan *hybrid genetic algorithm*, dalam penelitian tersebut disimpulkan bahwa penyelesaian *bounded knapsack problem* memiliki hasil yang lebih baik apabila diselesaikan dengan metode *hybrid genetic algorithm* daripada algoritma genetik biasa. Kemudian Hayyu (2016) membahas tentang *bounded knapsack problem* diselesaikan dengan menggunakan Algoritma *Artificial Bee Colony*, dalam penelitian tersebut disimpulkan bahwa penyelesaian *bounded knapsack problem*

juga memiliki hasil yang lebih baik apabila diselesaikan dengan Algoritma *Artificial Bee Colony* daripada algoritma genetik biasa, dan masih banyak algoritma yang dapat diterapkan dalam kasus optimasi salah satunya *Dragonfly Optimization Algorithm* (DOA).

DOA adalah salah satu algoritma optimasi yang dapat digunakan untuk pengambilan keputusan. Algoritma ini terinspirasi dari spesies capung dalam menangkap mangsa. Algoritma ini memiliki dua fase, yaitu fase *Eksplorasi* dan fase *Eksplotasi*. *Capung* menciptakan sub kawanan dan terbang di atas wilayah yang berbeda dalam kawanan dinamis, yang merupakan tujuan utama dari fase *Eksplorasi*. Namun, dalam kelompok statis, capung terbang dalam kawanan yang lebih besar dan sepanjang satu arah, yang menguntungkan dalam fase *Eksplotasi*. Perlu diketahui bahwa terdapat lima faktor yang digunakan dalam algoritma ini, antara lain: Pertama Faktor Pemisahan yaitu faktor yang menentukan seekor capung akan berpisah dari kelompoknya, kedua Faktor Penggabungan yaitu Faktor yang menentukan seekor capung bergabung dengan kelompok baru, ketiga Faktor Penyesuaian yaitu Faktor yang menentukan seekor capung menyesuaikan arah terbang pada kelompok yang baru, keempat Faktor Sumber Makanan yaitu Faktor yang menentukan capung dalam kelompok secara bersama-sama menuju sumber makanan, dan kelima Faktor Predator yaitu Faktor yang menentukan capung dalam kelompok secara bersama-sama berpencar untuk menghindari predator.

Berdasarkan uraian diatas, penulis tertarik untuk meneliti lebih lanjut permasalahan *Multiple Constraints Bounded Knapsack* dengan menerapkan DOA. Diharapkan dari penelitian ini DOA dapat menghasilkan solusi yang optimal dengan waktu komputasi lebih efisien.

1.2 Rumusan Masalah

Adapun permasalahan yang akan dibahas pada penelitian ini adalah sebagai berikut.

- a. Bagaimana penyelesaian masalah *Multiple Constraints Bounded Knapsack* menggunakan *Dragonfly Optimization Algorithm* (DOA)?
- b. Bagaimana keefektifan *Dragonfly Optimization Algorithm* (DOA) dibandingkan dengan hasil metode *Simplex*?

1.3 Tujuan

Tujuan yang ingin didapatkan dari penelitian ini adalah sebagai berikut.

- a. Menyelesaikan masalah *Multiple Constraints Bounded Knapsack* menggunakan *Dragonfly Optimization Algorithm* (DOA).
- b. Mengetahui keefektifan *Dragonfly Optimization Algorithm* (DOA) dibandingkan dengan hasil metode *Simplex*.

1.4 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah dapat memberikan solusi dari permasalahan *Multiple Constraints Bounded Knapsack* dengan menggunakan *Dragonfly Optimization Algorithm* (DOA) dan sebagai rujukan bagi peneliti yang ingin mempelajari lebih jauh tentang *Dragonfly Optimization Algorithm* (DOA).

BAB 2. TINJAUAN PUSTAKA

2.1 Optimasi

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 dan 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 Knapsack Problem

Knapsack 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, . . . , n) dimana setiap objek memiliki bobot (w_i), nilai *profit* atau keuntungan (p_i) dengan memperhatikan juga kapasitas dari media penyimpanan sebesar M dan nilai probabilitas dari setiap objek (X_i).

Knapsack Problem merupakan sebuah persoalan yang menarik. Dalam dunia nyata permasalahan *Knapsack* ini sering sekali digunakan pada bidang jasa pengangkutan barang seperti pengangkutan peti kemas dalam sebuah media pengangkut. Dalam usaha tersebut, diinginkan keuntungan yang maksimal untuk mengangkut barang yang ada 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 kita harus mencari solusi terbaik dari banyak kemungkinan yang dihasilkan (Dimiyanti dan Dimiyanti, 2004).

Masalah *Knapsack* dapat digambarkan sebagai sebuah wadah dengan kapasitas (M) akan diisi dengan benda sebanyak (n), masing-masing benda memiliki keuntungan berupa *value* (v) dan berat berupa *weight* (w). Sehingga masalah *Knapsack* dapat digambarkan sebuah proses memilih barang dengan berat dan keuntungan yang maksimal dari sejumlah n objek yang memilih kapasitas terbatas agar mendapatkan hasil yang optimal (Kellerer, *et al.*, 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 masalah *Knapsack* yang muncul dengan jumlah besar dari masalah dasar seperti jumlah kendala, jumlah tujuan atau bahkan medianya. Adapun variasi permasalahan *Knapsack* diatas antara lain (Kellerer, *et al.*, 2004) :

a. *Multi Objective Knapsack Problem*

Permasalahan yang memiliki fungsi tujuan lebih dari satu untuk memaksimalkan keuntungannya.

b. *Multidimensional atau Multiple Constrains Knapsack Problem*

Permasalahan yang memiliki kendala lebih dari satu untuk memaksimalkan keuntungannya.

c. *Multi Knapsack Problem*

Permasalahan yang memiliki fungsi objek/media penyimpanan lebih dari satu dimana semua item harus dikemas untuk memaksimalkan keuntungannya.

d. *Quadratic Knapsack Problem*

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

Knapsack yang akan dibahas adalah *multiple constraints bounded knapsack* dimana tiap item memiliki kendala dengan jumlah unit barangnya terbatas dan jika dimasukkan ke dalam media penyimpanan barang tersebut bisa dimasukkan sebagian atau seluruhnya (Hayyu, 2016).

2.3 Bounded Knapsack

Bounded Knapsack merupakan generalisasi dari *knapsack* 0-1 dimana jumlah dari setiap jenis barang yang tersedia dibatasi. Misalkan, diberikan n buah objek untuk diangkut dalam sebuah media penyimpanan yang memiliki daya tampung M . Setiap objek j memiliki berat w_j , nilai keuntungan p_j dan batas kesediaan barang m_j . Permasalahannya adalah bagaimana memilih barang dengan jumlah tertentu x_j , dimana jumlah barang tidak melebihi batas ketersediaan (Martello *et al.*, 2006).

Secara matematis, permasalahan *bounded knapsack* dapat didefinisikan sebagai berikut :

Fungsi tujuan maksimum

$$Z = \sum_{j=1}^n p_j x_j \quad (2.1)$$

Kendala

$$\sum_{j=1}^n w_j x_j \leq C \quad (2.2)$$

$$x_j \in \{0, 1, \dots, m_j\}, j = 1, \dots, n \quad (2.3)$$

dimana

Z : nilai optimum fungsi tujuan

p_j : keuntungan/*profit* barang ke- j per unit

x_j : jumlah unit barang ke- j yang dimasukkan ke media penyimpanan

n : banyak barang

w_j : berat barang ke- j

m_j : jumlah kesediaan barang ke- j

C : kapasitas berat media penyimpanan

fungsi tujuan semua koefisien adalah bilangan bulat positif. Tanpa menghilangkan sifat umumnya, dapat diasumsikan bahwa $m_j w_j \leq C$ untuk $j = 1, 2, \dots, n$ sehingga semua jenis barang dapat dimasukkan kedalam media penyimpanan (Pisinger, 1995).

2.4 Multiple Constraints Bounded Knapsack (MCBK)

Multiple constraints knapsack problem (MCKP) adalah suatu permasalahan *knapsack* yang sering disebut juga dengan *multidimensional knapsack problem* (MKP) dimana MCKP merupakan permasalahan optimasi kombinatorial NP-hard yang terdapat pada beragam aplikasi. Pada MCKP, tiap-tiap objek/barang memiliki batasan lebih dari satu dimensi. Batasan-batasan bisa berupa waktu, biaya dan pekerja. Tujuan dari masalah ini yaitu memperoleh solusi optimal dengan mengambil kombinasi barang sedemikian hingga semua batasan juga tidak melewati kapasitas yang tersedia (Lamine, et al., 2012).

Secara matematik, *multiple constraints bounded knapsack* dapat dirumuskan sebagai formulasi MCBK berikut :

Fungsi tujuan maksimum

$$Z = \sum_{j=1}^n p_j x_j \quad (2.4)$$

Kendala

$$\sum_{j=1}^n w_j x_j \leq C \quad (2.5)$$

$$\sum_{j=1}^n v_j x_j \leq S \quad (2.6)$$

$$\sum_{j=1}^n b_j x_j \leq M \quad (2.7)$$

$$x_j \in \{1, 2, \dots, m_j\}, \quad j = 1, 2, \dots, n \quad (2.8)$$

dimana,

Z : nilai optimum fungsi tujuan

p_j : keuntungan/*profit* barang ke- j per unit

x_j : jumlah barang ke- j yang dimasukkan ke media penyimpanan

n : banyaknya jenis barang

m_j : jumlah ketersediaan barang ke- j

w_j	: berat barang ke- j
v_j	: volume barang ke- j
b_j	: harga beli barang ke- j
C	: kapasitas berat media penyimpanan
S	: kapasitas ruang media penyimpanan
M	: modal

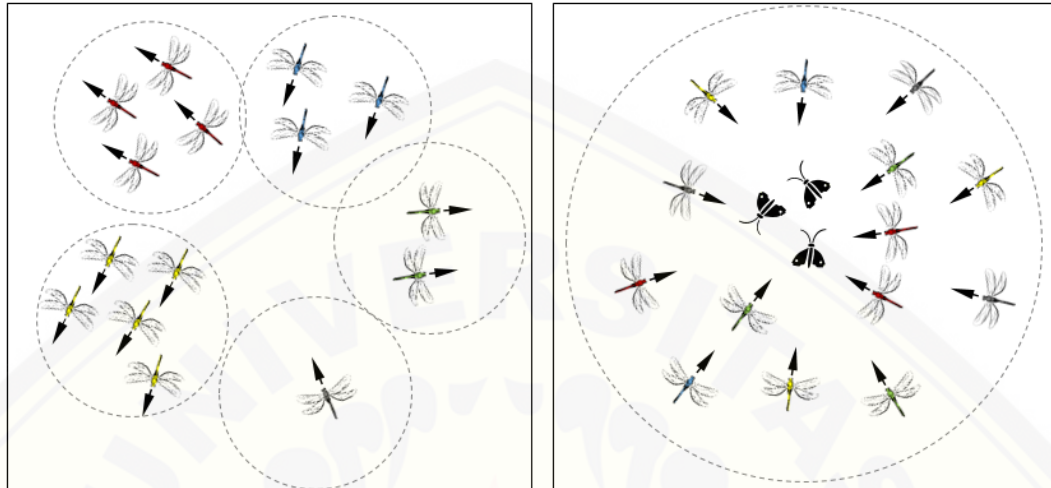
2.5 Algoritma

Algoritma merupakan urutan langkah-langkah untuk menyelesaikan masalah yang disusun secara sistematis. Algoritma dibuat dengan tanpa memperhatikan bentuk yang akan digunakan sebagai implementasinya, sehingga suatu algoritma dapat menjelaskan “bagaimana” cara menggunakan fungsi yang dapat diekspresikan dengan suatu program (Tony, 2007). Pertimbangan dalam pemilihan algoritma adalah algoritma haruslah benar artinya algoritma akan memberikan keluaran yang dikehendaki dari sejumlah masukan yang diberikan. Tidak peduli sebegus apapun algoritma kalau memberikan keluaran yang salah, pastilah algoritma tersebut bukanlah algoritma yang baik. Algoritma yang baik adalah algoritma yang dapat meminimumkan kebutuhan ruang dan waktu. Dalam suatu algoritma yang menjadi perhatian adalah ruang memori yang dibutuhkan dan lamanya waktu tempuh untuk menjalankan suatu algoritma tersebut. Meskipun suatu algoritma memberikan hasil yang mendekati optimal tetapi waktu yang dibutuhkan sangat lama maka algoritma tersebut biasanya jarang dipakai (Munir, 2005).

2.6 *Dragonfly Optimization Algorithm (DOA)*

Inspirasi utama dari *Dragonfly Optimization Algorithm (DOA)* berasal dari perilaku sibuk statis dan dinamis. Kedua perilaku ini sangat mirip dengan dua fase utama optimasi menggunakan metaheuristic yaitu fase eksplorasi dan fase eksploitasi. Capung menciptakan sub kawanan dan terbang di atas wilayah yang berbeda dalam kawanan dinamis, yang merupakan tujuan utama dari fase eksplorasi. Namun, dalam kelompok statis, capung terbang dalam kawanan yang

lebih besar dan sepanjang satu arah, yang menguntungkan dalam fase eksploitasi. Sebuah model konseptual dari kawanan dinamis dan statis diilustrasikan dalam Gambar 2.1

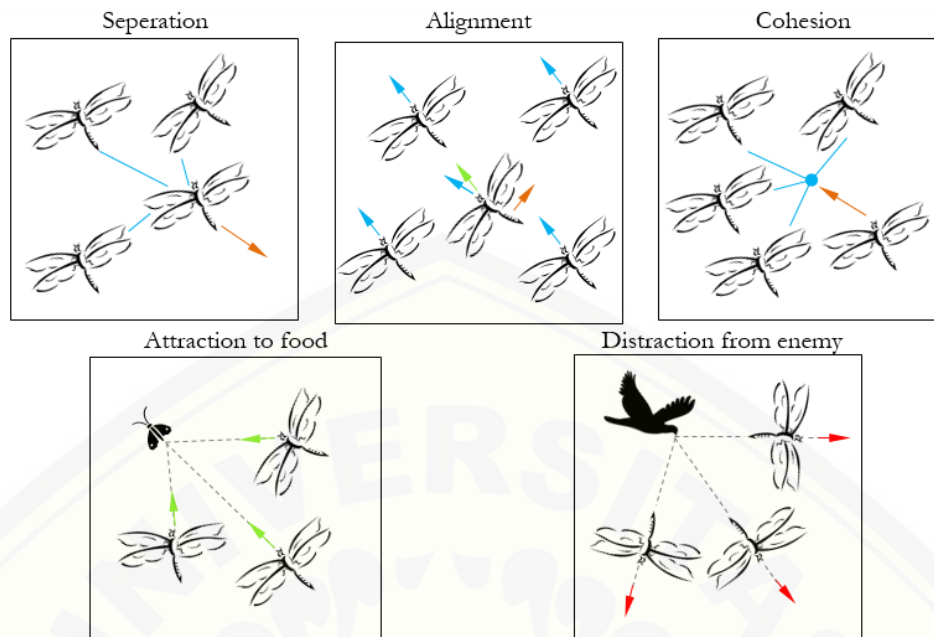


Gambar 2.1 model konseptual dinamis dan statis

(Sumber : Mirjalili, 2016)

Capung hanya menunjukkan dua jenis kawanan: statis dan dinamis seperti yang ditunjukkan pada Gambar 2.1. Seperti yang dilihat pada gambar diatas, capung cenderung menyelaraskan terbangnya sambil mempertahankan pemisahan dan kohesi yang tepat dalam kerumunan yang dinamis. Namun, dalam gerombolan statis, keberpihakan sangat rendah sementara kohesi tinggi untuk menyerang mangsa. Oleh karena itu, kami menetapkan capung dengan penjajaran tinggi dan bobot kohesi rendah ketika menjelajahi ruang pencarian dan penyelarasan rendah dan kohesi tinggi ketika mengeksploitasi ruang pencarian. Untuk transisi antara eksplorasi dan eksploitasi, jari-jari lingkungan meningkat sebanding dengan jumlah iterasi (Mirjalili, 2016).

Untuk simulasi perilaku capung ada lima faktor penting, tiga prinsip primitif dari mengerumuni serangga serta dua konsep baru lainnya seperti yang ditunjukkan pada Gambar 2.2



Gambar 2.2 lima faktor yang digunakan dalam DOA

(Sumber : Mirjalili, 2016)

Setiap perilaku partikel yang diekspresikan dirumuskan menggunakan beberapa model. Model-model ini dibahas dalam bagian berikut (Amini, *et al.*, 2018) :

1. *Separation* atau pemisahan mengacu pada mekanisme yang diikuti capung untuk menghindari tabrakan dengan capung lainnya. Perilaku ini dirumuskan seperti pada Persamaan (2.9)

$$S_i = \sum_{j=1}^n X - X_j \quad (2.9)$$

2. *Alignment* atau keselarasan menunjukkan kecocokan kecepatan capung menurut capung terdekat lainnya. Perilaku ini dirumuskan seperti pada Persamaan (2.10)

$$A_i = \frac{\sum_{j=1}^n V_j}{n} \quad (2.10)$$

3. Kohesi mengacu pada kecenderungan capung pusat massa di lingkungan itu. Perilaku ini dirumuskan seperti pada Persamaan (2.11)

$$C_i = \frac{\sum_{j=1}^n X_j}{n} - X \quad (2.11)$$

dimana

S_i : pemisahan capung ke- i

A_i : keselarasan capung ke- i

C_i : kohesi capung ke- i

V_j : kecepatan capung ke- j

X : posisi capung saat ini

X_j : posisi capung ke- j

n : jumlah capung

Daya tarik terhadap sumber makanan dan melarikan diri dari musuh adalah dua perilaku kunci lainnya yang masing-masing capung berperilaku untuk bertahan hidup.

4. Capung yang melompat menuju sumber makanan dirumuskan menggunakan Persamaan (2.12)

$$F_i = X^+ - X \quad (2.12)$$

5. Melarikan diri dari musuh dirumuskan menggunakan Persamaan (2.13)

$$E_i = X^- - X \quad (2.13)$$

di mana

F_i : *Food* atau makanan ke- i

E_i : *Enemy* atau musuh ke- i

X^+ : menunjukkan posisi sumber makanan

X^- : menunjukkan posisi musuh

X : menunjukkan posisi individu saat ini

(Mirjalili, 2016).

DOA menggunakan dua vektor untuk menyelesaikan masalah optimasi yaitu vektor langkah dan vektor posisi. Vektor langkah didefinisikan seperti pada Persamaan (2.14)

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \quad (2.14)$$

dimana

s : berat pemisahan

S_i : pemisahan capung ke- i

a : berat keselarasan

- A_i : keselarasan dari capung ke-i
 c : berat kohesi
 C_i : kohesi dari capung ke-i
 f : faktor makanan
 F_i : sumber makanan dari capung ke-i
 e : faktor musuh
 E_i : posisi musuh dari capung ke-i
 w : berat inersia
 t : jumlah iterasi algoritma

Setelah menghitung vektor langkah, vektor posisi dirumuskan seperti pada Persamaan (2.15)

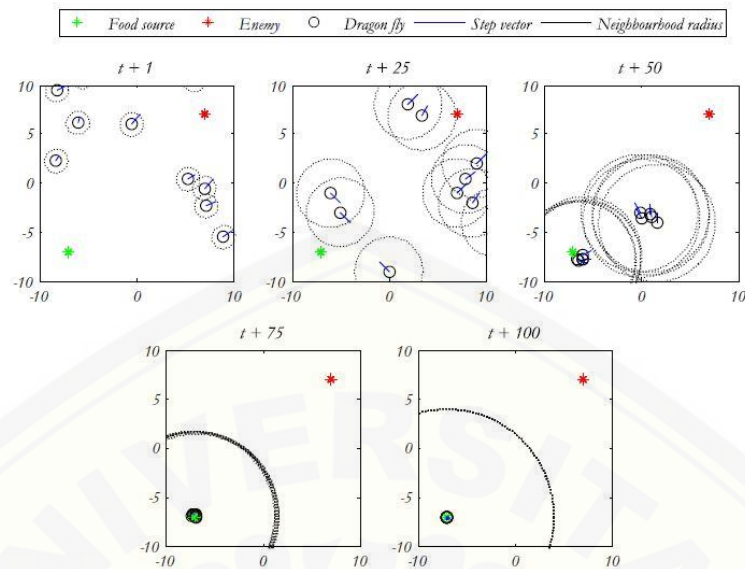
$$X_{t+1} = X_t + \Delta X_{t+1} \quad (2.15)$$

dimana t menunjukkan lokasi saat ini (Amini, *et al.*, 2018).

Vektor langkah dan vektor posisi digunakan ketika X_i mempunyai tetangga, tetapi jika X_i tidak mempunyai tetangga maka X_i berpindah secara acak, dirumus seperti pada Persamaan (2.16)

$$X_{t+1} = X_t + rand(-1,1)(X^+ - X_t) \quad (2.16)$$

Dengan pemisahan, keselarasan, kohesi, faktor makanan, dan faktor musuh (s , a , c , f , dan e), berbagai perilaku eksploratif dan eksploitatif dapat dicapai selama optimisasi. Tetangga capung sangat penting, sehingga suatu lingkungan (lingkaran dalam 2D, bola dalam ruang 3D, atau hypersphere dalam ruang nD) dengan radius tertentu diasumsikan di sekitar masing-masing capung buatan. Contoh perilaku berkerumun capung dengan peningkatan radius lingkungan menggunakan model matematika yang diusulkan diilustrasikan pada Gambar (2.3)

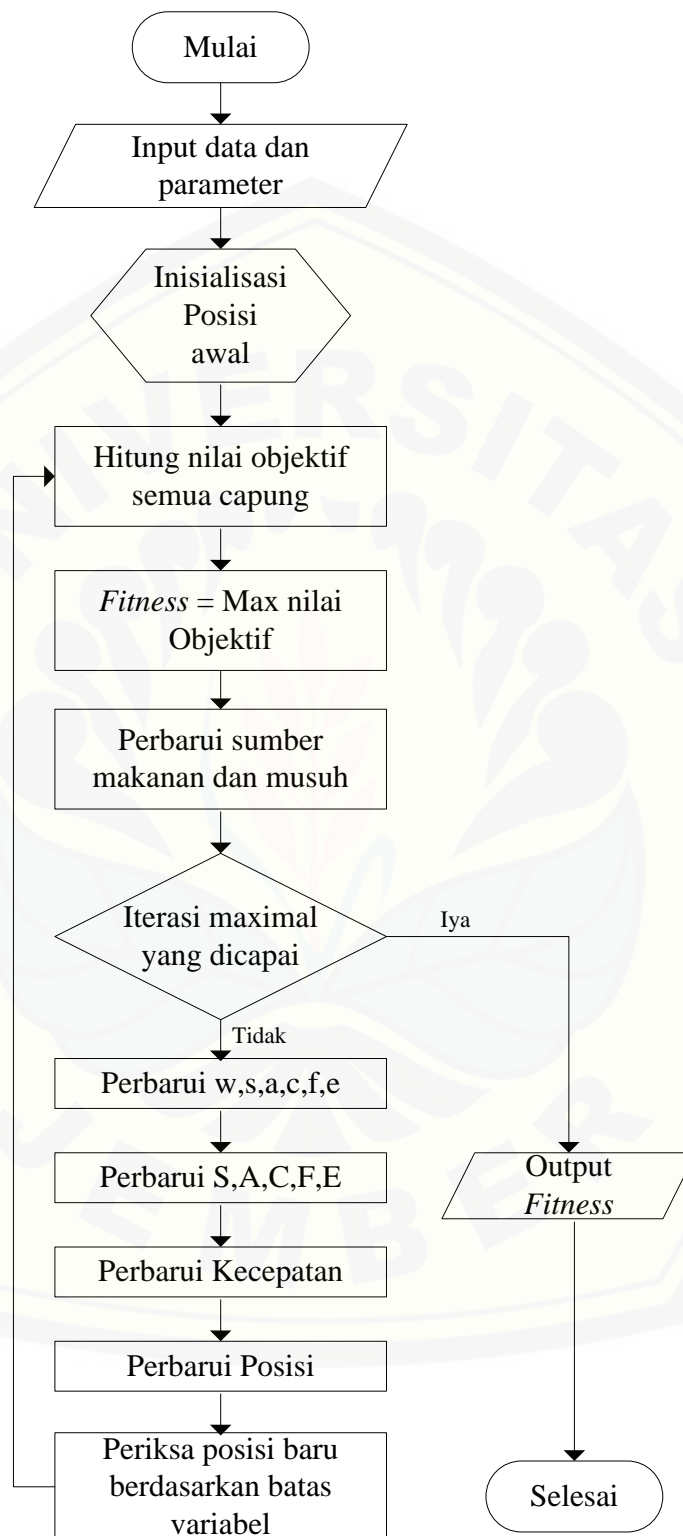


Gambar 2.3 Model perilaku berkerumun capung

(Sumber : Mirjalili, 2016)

Pada Gambar 2.3 model bagaimana perilaku berkerumun capung dimana terdapat beberapa tahapan berkerumun capung untuk mendapatkan mangsa, dijelaskan bahwa pertama capung akan berpencar untuk mencari atau menghindari musuh jika salah satu capung menemukan mangsa maka capung yang lain juga akan mengikuti mangsa yang telah ditemukan tersebut dan berkerumun pada mangsa untuk memakan mangsa tersebut. Jika banyak capung yang berkerumun pada satu titik yang sama maka radius lingkungannya juga akan meningkat.

Pada flowchart berikut cara menjalankan DOA



Gambar 2.4 Flowchart Dragonfly Optimization Algorithm (DOA)

(Sumber : Amini, *et al.*, 2018)

2.7 Metode Simpleks

Menurut Siang (2014) metode simpleks merupakan metode untuk mencari nilai fungsi di titik ujung daerah fisibel dengan melakukan pencarian iteratif secara numerik sehingga terhindar dari keterbatasan jumlah variabel. 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 pada ruas kiri. 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 kedua ruas 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).

Contoh :

$$\text{Maksimum } Z = 3x_1 + 5x_2$$

Batasan (constraints)

$$2x_1 \leq 8$$

$$3x_2 \leq 15$$

$$6x_1 + 5x_2 \leq 30$$

Penyelesaian :

Langkah a

Fungsi tujuan :

$$Z = 3x_1 + 5x_2 \text{ diubah menjadi } Z - 3x_1 - 5x_2 = 0$$

Langkah b dan c

Fungsi batasan :

$$2x_1 \leq 8 \quad \text{menjadi} \quad 2x_1 + x_3 = 8$$

$$3x_2 \leq 15 \quad \text{menjadi} \quad 3x_2 + x_4 = 15$$

$$6x_1 + 5x_2 \leq 30 \quad \text{menjadi} \quad 6x_1 + 5x_2 + x_5 = 30$$

Tabel 2.1 Tabel Simpleks pertama

Variabel	Z	x_1	x_2	x_3	x_4	x_5	NK
Z	1	-3	-5	0	0	0	0
x_3	0	2	0	1	0	0	8
x_4	0	0	3	0	1	0	15
x_5	0	6	5	0	0	1	30

Dimana NK adalah Nilai kanan. Kemudian pilih kolom yang mempunyai nilai pada garis fungsi tujuan yang bernilai negatif dengan angka terbesar. Dalam hal ini kolom x_2 dengan nilai pada baris persamaan tujuan -5.

Tabel 2.2 Memilih baris kunci

Variabel	Z	x_1	x_2	x_3	x_4	x_5	NK	Keterangan (Indeks)
Z	1	-3	-5	0	0	0	0	
x_3	0	2	0	1	0	0	8	$8/0 = \infty$
x_4	0	0	3	0	1	0	15	$15/3 = 5$
x_5	0	6	5	0	0	1	30	$30/5 = 6$

Pilih baris yang mempunyai indeks positif terkecil. Dalam hal ini batasan ke-2 yang terpilih sebagai baris kunci. Nilai baris kunci diubah dengan cara membaginya dengan angka kunci, seperti tabel 2.3 bagian bawah.

Tabel 2.3 Cara mengubah nilai baris kunci

Variabel	Z	x_1	x_2	x_3	x_4	x_5	NK	Keterangan (Indeks)
Z	1	-3	-5	0	0	0	0	
x_3	0	2	0	1	0	0	8	$8/0 = \infty$
x_4	0	0	3	0	1	0	15	$15/3 = 5$
x_5	0	6	5	0	0	1	30	$30/5 = 6$
Z								
x_3								
x_2	0	0	1	0	1/3	0	5	
x_5								

Bagian selanjutnya menggunakan cara yang sama seperti sebelumnya.

Tabel 2.4 Mengubah nilai kolom dan baris kunci

Variabel	Z	x_1	x_2	x_3	x_4	x_5	NK
Z	1	-3	-5	0	0	0	0
x_3	0	2	0	1	0	0	8
x_4	0	0	3	0	1	0	15
x_5	0	6	5	0	0	1	30
Z	1	-3	0	0	5/3	0	25
x_3	0	2	0	1	0	0	8
x_2	0	0	1	0	1/3	0	5
x_5	0	6	0	0	-5/3	1	5

Ulangi langkah sebelumnya untuk memperbaiki tabel-tabel yang telah diubah nilainya. perubahan baru berhenti setelah pada baris pertama (fungsi tujuan) tidak ada yang bernilai negatif.

Tabel 2.5 Melanjutkan perbaikan

Variabel Dasar	Z	x_1	x_2	x_3	x_4	x_5	NK	Keterangan (Indeks)
Z	1	-3	0	0	5/3	0	25	
x_3	0	2	0	1	0	0	8	8/2=4
x_4	0	0	1	0	1/3	0	5	
x_5	0	6	0	0	-5/3	1	5	5/6
Z	1							
x_3	0							
x_2	0							
x_1	0	1	0	0	-5/18	1/6	5/6	

Menggunakan cara yang sama sampai mendapatkan hasil optimal. Baris pertama (Z) tidak ada lagi yang bernilai negatif, sehingga tabel tidak dapat dioptimalkan lagi dan tabel berikut merupakan hasil optimal.

Tabel 2.6 Final hasil perubahan

Variabel Dasar	Z	x_1	x_2	x_3	x_4	x_5	NK
Z	1	0	0	0	5/6	1/2	27 1/2
x_3	0	0	0	1	5/9	-1/3	6 1/3
x_2	0	0	1	0	1/3	0	5
x_1	0	1	0	0	-5/18	1/6	5/6

Dari tabel final didapatkan

$$x_1 = 5/6$$

$$x_2 = 5$$

$$Z_{maksimum} = 27\frac{1}{2}$$

2.8 Penggunaan Simpleks pada Excel

Solver adalah program *add-in* yang berada di bawah program Excel. Program *solver* ini berisi perintah-perintah yang berfungsi untuk melakukan analisis terhadap masalah optimasi. Jika kita instal Microsoft Excel, maka tidak secara otomatis *solver* ini terinstal, jadi harus di instal secara khusus setelah program Excel terinstal pada komputer. Program *solver* ini cukup baik untuk menyelesaikan masalah optimasi (Dwijanto, 2008).

Menurut Herry (2008) terdapat 3 algoritma atau metode penyelesaian pada solver, yaitu: *Standard GRG Nonlinear*, *Standard Simpleks LP*, dan *Standard Evolutionary*. Dalam penelitian ini yang digunakan adalah dengan *Standard Simpleks LP*. Adapun langkah-langkah dalam penggunaan simpleks pada *solver add-in* Microsoft Excel adalah sebagai berikut :

1. *Set Target Cell: The Target Cell* berisi kuantitas atau jumlah yang dioptimumkan yaitu nilai fungsi tujuan. Untuk menentukan letak *Target Cell*, klik pada sel yang sudah ditentukan atau ketik nama selnya.
2. *Equal To*: Menentukan arah optimasi, bila kasusnya biaya maka dipilih Min sedangkan apabila kasusnya keuntungan maka dipilih Max.
3. *By Changing Cells*: Disini diisi dengan sel pada *sheet* dimana akan ditampilkan nilai variabel. Nilai variabel ini lah yang akan mengoptimumkan fungsi tujuan.
4. *Subject to the Constraints*: Tentukan kendala dengan mengklik tombol *Add*, kemudian masukkan fungsi kendala dengan mengisi sel sebelah kiri, pilih = 18 atau \leq atau \geq (sesuai fungsi pembatas/kendala) kemudian isi sel sebelah kanan. Setelah seluruh kendala fungsional dimasukkan, tekan tombol OK. Pastikan pula telah menyentang pembatas non-negatif.
5. *Solving Method* : Memilih metode penyelesaian. Dalam penelitian ini digunakan *Standard Simpleks LP*.

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang akan digunakan dalam penelitian ini adalah data primer berupa data penjualan dari rumah produksi kerajinan Bambu Hitam di Desa Pujerbaru, Kecamatan Maesan, Kabupaten Bondowoso. Kerajinan Bambu Hitam memproduksi berbagai macam jenis kerajinan yang berasal dari bambu hitam yang kemudian dibentuk dan diukir sesuai yang diinginkan. Data yang diambil berupa nama barang/benda, volume barang/benda, berat barang/benda, jumlah barang/benda, biaya produksi, harga jual dan keuntungan.

3.2 Langkah-langkah Penelitian

Penelitian mengenai penerapan *Dragonfly Optimization Algorithm* (DOA) untuk menyelesaikan masalah *multiple constraints bounded knapsack* akan dilakukan dengan langkah-langkah sebagai berikut:

a. Studi Literatur

Langkah awal yang dilakukan yaitu mengumpulkan data dengan cara mencari, membaca, serta mempelajari buku-buku maupun jurnal internasional yang berkaitan dengan *Dragonfly Optimization Algorithm* (DOA) dan permasalahan *multiple constraints bounded knapsack*.

b. Pengumpulan Data

Langkah kedua yaitu mengumpulkan dan mengidentifikasi data barang dari rumah produksi kerajinan Bambu Hitam di Desa Pujerbaru, Kecamatan Maesan, Kabupaten Bondowoso.

c. Penerapan *Dragonfly Optimization Algorithm* (DOA)

Menerapkan *Dragonfly Optimization Algorithm* (DOA) pada data yang telah diidentifikasi dengan langkah-langkah berikut :

- 1) Input parameter dan data
- 2) Inisialisasi populasi awal
 - a) Kandidat solusi awal

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,d} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,d} \\ \vdots & \vdots & & \vdots \\ X_{n,1} & X_{n,2} & \cdots & X_{n,d} \end{bmatrix}$$

Nilai X dipilih dari nilai $[0,1]$

$$Y = \begin{bmatrix} Y_{1,1} & Y_{1,2} & \cdots & Y_{1,d} \\ Y_{2,1} & Y_{2,2} & \cdots & Y_{2,d} \\ \vdots & \vdots & & \vdots \\ Y_{n,1} & Y_{n,2} & \cdots & Y_{n,d} \end{bmatrix}$$

dengan $Y_{i,j} = X_{i,j} \times m_j$ kemudian dibulatkan ke pembulatan terdekat

dimana,

n : banyak capung

d : banyak barang

m : jumlah barang

b) Pengecekan kendala

Setiap solusi harus memenuhi Persamaan

$$\sum_{j=1}^n w_j y_j \leq C \quad (3.1)$$

$$\sum_{j=1}^n v_j y_j \leq S \quad (3.2)$$

$$\sum_{j=1}^n b_j y_j \leq M \quad (3.3)$$

dengan

w_j : berat barang ke- j

v_j : volume barang ke- j

b_j : harga beli barang ke- j

C : kapasitas berat media penyimpanan

S : kapasitas ruang media penyimpanan

M : modal

Jika tidak memenuhi, maka perlu dipinjalty dengan menggunakan rumus

$$x_{i,j} = x_{i,j} - \frac{1}{m_j} \quad (3.4)$$

c) Evaluasi fungsi

Hitung total profit dari setiap kandidat solusi. Total profit dihitung menggunakan Persamaan (3.5)

$$Z = \sum_{j=1}^n p_j y_j \quad (3.5)$$

dengan

Z : total profit

p_j : profit barang jenis j

3) Inisialisasi langkah Δx

Karena capung diawali dari posisi diam maka :

$$\Delta X = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

4) Memperbarui sumber makanan dan musuh

X^+ untuk sumber makanan diambil dari posisi terbaik

X^- untuk musuh diambil dari posisi terburuk

5) Memperbarui $w, s, a, c, f,$ dan e

6) Menghitung $S, A, C, F,$ dan E

Untuk menghitung $S, A, C, F,$ dan E didasarkan pada Persamaan (2.9) sampai Persamaan (2.13).

7) Perbarui Kecepatan dan posisi

Untuk memperbarui kecepatan dan posisi didasarkan pada Persamaan (2.14) dan Persamaan (2.15).

8) Memastikan X_i berada pada ruang pencarian $[0,1]$

a) Jika $X_i > 0$ maka menggunakan rumus pada persamaan (3.6)

$$X_i = \frac{X_i - \min(X_i)}{\max(X_i) - \min(X_i)} \quad (3.6)$$

b) Jika $X_i < 0$ maka menggunakan rumus pada persamaan (3.7)

$$X_i = \frac{X_i}{\max(X_i)} \quad (3.7)$$

d. Penerapan Metode Simpleks

Data yang dikumpulkan, selain akan diselesaikan menggunakan *Dragonfly Optimization Algorithm*, juga akan diselesaikan dengan metode simpleks. Penyelesaian metode simpleks dibantu dengan *Solver Tool* yang tersedia pada Microsoft Excel.

e. Pembuatan Program

Membuat program berdasarkan permasalahan yang akan dibahas yaitu mengenai *multiple constraints bounded knapsack* yang akan diselesaikan dengan *Dragonfly Optimization Algorithm* (DOA) dengan menggunakan *software* Matlab.

f. Simulasi Program

Setelah pembuatan program, langkah selanjutnya yaitu pengujian program yang telah dibuat.

g. Analisis Hasil

Hasil dari percobaan akhir DOA akan dibandingkan dengan hasil metode *Simplex* dengan menggunakan presentase deviasi menggunakan Persamaan (3.8).

$$\text{Presentase deviasi} = \frac{\text{simplex} - z_i}{\text{simplex}} \times 100\% \quad (3.8)$$

dengan

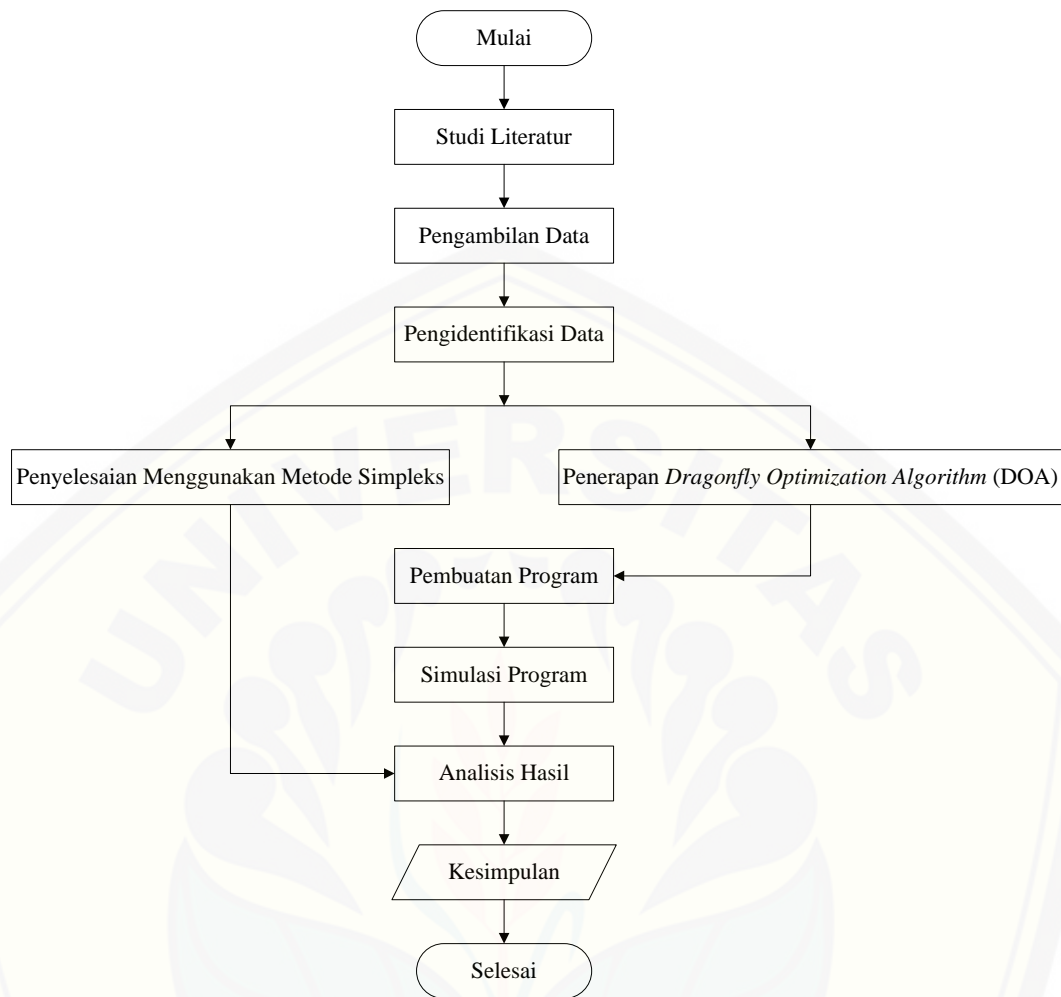
Simplex : hasil metode *Simplex*

z_i : hasil profit DOA

h. Kesimpulan

Langkah terakhir yaitu membuat kesimpulan dari permasalahan yang dibahas.

Berikut adalah skema untuk menyelesaikan permasalahan *multiple constraint bounded knapsack* dengan menggunakan *Dragonfly Optimization Algorithm* (DOA).



Gambar 3.1 Skema langkah-langkah penelitian

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan yang telah diuraikan dapat diambil kesimpulan sebagai berikut :

- a. Solusi terbaik yang dihasilkan oleh *Dragonfly Optimization Algorithm* dalam penyelesaian masalah *multiple constraint bounded knapsack* adalah sebesar Rp 12.141.500,-. Solusi terbaik tersebut diperoleh dari profit paling maksimal hasil akhir simulasi data dengan parameter $pop = 200$, $maxgen = 1000$, $wmax = 0,5$, $wmin = 0,15$, $s = 0,5$, $a = 0,01$, $c = 0,2$, $f = 0,5$, $e = 0,1$, $Range = 1,25$ dan iterasi konvergen 613. Parameter-parameter tersebut memiliki pengaruh yang sama yaitu untuk mendapatkan hasil yang optimal, dimana semakin besar nilai dari parameter tersebut maka hasil yang didapatkan juga semakin mendekati nilai optimal.
- b. Hasil dari *Dragonfly Optimization Algorithm* yang memiliki keuntungan Rp 12.141.500,- dibandingkan dengan hasil *simplex* yang memiliki keuntungan Rp 12.526.500,-, yang artinya *Dragonfly Optimization Algorithm* kurang efektif untuk menyelesaikan permasalahan *multiple constraint bounded knapsack*. *Dragonfly Optimization Algorithm* mendekati optimal, dilihat dari deviasi yang cukup kecil.

5.2 Saran

Dragonfly Optimization Algorithm (DOA) dapat diterapkan pada permasalahan *multiple constraint bounded knapsack* akan tetapi hasil yang diperoleh belum optimal, untuk penelitian selanjutnya disarankan memodif *Dragonfly Optimization Algorithm* (DOA) atau di Hybrid dengan algoritma lain agar mampu menemukan solusi optimal dengan kata lain lebih efektif.

DAFTAR PUSTAKA

- Amini, Z., Maeen, M., dan Jahangir, M.R. 2018. Providing A Load Balancing Method Based on Dragonfly Optimization Algorithm for Resource Allocation in Cloud Computing. *International Journal of Networked and Distributed Computing*. 6 (1).
- Dimiyanti, T.T. dan A. Dimiyanti. 2004. *Operation Research: Model-model Pengambilan Keputusan*. Bandung: Sinar Baru Agesindo.
- Dwijanto. 2008. *Program Linear Berbantu Komputer: Lindo, Lingo, dan Solver*. Semarang: UNNES Press.
- Hadi, I.S. 2015. Penerapan Algoritma Genetika *Hybrid* pada Permasalahan *Bounded Knapsack*. *Skripsi*. Jember: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.
- Hayyu, A. N. 2016. Penerapan Algoritma *Artificial Bee Colony* pada Permasalahan *Multiple Constraints Bounded Knapsack*. *Skripsi*. Jember: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.
- Herry, D.Y. 2008. *Riset Operasi dengan Excel*. Yogyakarta: ANDI.
- Kellerer, H., Pferschy, U., dan Pisinger, D. 2004. *Knapsack Problem*. New York: Springer-Verlag Berlin Heidelberg.
- Lamine, A., Khemakhem, M., dan Chabchoud, H. 2012. Knapsack Problem Involving Dimensions, Demands and Multiple Choice Constraints. *International Journal of Advanced Science and Technology*. 46 (4).

- Martello, S., D. Pisinger dan P. Toth. 2000. New Trends in Exact Algorithms for the 0-1 Knapsack Problem. *European Journal of Operation Research*. 123: 325-332.
- Mirjalili, S. 2016. Dragonfly Algorithm : a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput & Applications*. 27 (4):1053-1073.
- Munir, R. 2005. *Strategi Algoritmik*. Yogyakarta: Graha Ilmu.
- Pisinger, D. 1995. A minimal Algorithm for the Multiple-Choice Knapsack Problem. *European Journal of Operation Research*, 83(2):349-410.
- Pisinger, D. 1995. *Algorithms for Knapsack Problems*. Denmark: University of Copenhagen.
- Siang, J.J. 2014. *Riset Operasi dalam Pendekatan Algoritmik*. Yogyakarta: Andi Offset.
- Tony, H. 2007. *Algoritma dan Pemrograman*. Seri: Modul Diskusi Fakultas Ilmu Komputer. Surabaya : Universitas Narotama. Rincikembang.dosen.narotama.ac.id/files/2011/04/Algo2007.pdf. [3 Desember 2018].
- Windasari, D. 2009. Metode Simpleks dalam Program Linier. *Jurnal SAINTKOM*. 6(1): 276-277.

LAMPIRAN

Lampiran A. Data penelitian

No	Nama Barang	Jumlah	Berat (Kg)	Volume	Biaya Produksi (item)	Harga Jual (item)
1	Meja Biasa	8	8	127756	165500	200000
2	Meja Kaca	4	10	146071	196000	250000
3	Meja Sudut	4	10	370260	248000	300000
4	Nampan 1	20	0,5	1632	28000	35000
5	Nampan 2	18	1	4400	43000	55000
6	Nampan 3	16	1	6600	48500	65000
7	Nampan 4	15	1	10500	46000	75000
8	Peti Harta Besar	20	2	18480	210000	350000
9	Peti Harta Kecil	20	1	4200	163000	250000
10	Vas Bunga Duduk	18	0,5	57024	32500	70000
11	Vas Bunga Lantai	18	0,75	96800	36000	100000
12	Lampu Gantung	6	3,5	225000	245000	300000
13	Lampu Dinding 1	9	2,5	51714	70000	100000
14	Lampu Dinding 2	9	2,5	91260	98000	150000
15	Hiasan Dinding	20	0,5	30550	20000	25000
16	Pigura 20R	12	1	3000	45500	50000
17	Pigura 85x65	12	1,5	8287,5	88000	200000
18	Kere Gambar 1	15	1,5	2400	46000	75000
19	Kere Gambar 2	10	1,75	2975	57500	100000
20	Kere Gambar 3	8	2	3000	80000	150000
21	Kere Gambar 4	8	2,25	8000	96000	200000

Lampiran B. Hasil metode *simplex*

Microsoft Excel 14.0 Answer Report

Worksheet: [Book3]Sheet1

Report Created: 03/12/2019 21:39:47

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

Solver Engine

Engine: Simplex LP

Solution Time: 0,64 Seconds.

Iterations: 3 Subproblems: 124

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
\$N\$24	profit	0	12526500

Variable Cells

Cell	Name	Original Value	Final Value	Integer
\$J\$2	Meja Biasa solusi	0	1	Integer
\$J\$3	Meja Kaca solusi	0	4	Integer
\$J\$4	Meja Sudut solusi	0	2	Integer
\$J\$5	Nampan 1 solusi	0	18	Integer
\$J\$6	Nampan 2 solusi	0	18	Integer
\$J\$7	Nampan 3 solusi	0	16	Integer
\$J\$8	Nampan 4 solusi	0	15	Integer
\$J\$9	Peti Harta Besar solusi	0	20	Integer
\$J\$10	Peti Harta Kecil solusi	0	20	Integer
\$J\$11	Vas Bunga Duduk solusi	0	18	Integer
\$J\$12	Vas Bunga Lantai solusi	0	18	Integer
\$J\$13	Lampu Gantung solusi	0	6	Integer
\$J\$14	Lampu Dinding 1 solusi	0	9	Integer
\$J\$15	Lampu Dinding 2 solusi	0	9	Integer
\$J\$16	Hiasan Dinding solusi	0	20	Integer
\$J\$17	Pigura 20R solusi	0	0	Integer
\$J\$18	Pigura 85x65 solusi	0	12	Integer
\$J\$19	Kere Gambar 1 solusi	0	15	Integer
\$J\$20	Kere Gambar 2 solusi	0	10	Integer
\$J\$21	Kere Gambar 3 solusi	0	8	Integer
\$J\$22	Kere Gambar 4 solusi	0	8	Integer

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
\$K\$24	berat	376,5	\$K\$24<=\$K\$26	Not Binding	3623,5
\$L\$24	volume	8547634	\$L\$24<=\$L\$26	Not Binding	452366
\$M\$24	biaya	19993500	\$M\$24<=\$M\$26	Not Binding	6500
\$J\$2	Meja Biasa solusi	1	\$J\$2<=\$C\$2	Binding	0
\$J\$3	Meja Kaca solusi	4	\$J\$3<=\$C\$3	Binding	0
\$J\$4	Meja Sudut solusi	2	\$J\$4<=\$C\$4	Binding	0
\$J\$5	Nampan 1 solusi	18	\$J\$5<=\$C\$5	Binding	0
\$J\$6	Nampan 2 solusi	18	\$J\$6<=\$C\$6	Binding	0
\$J\$7	Nampan 3 solusi	16	\$J\$7<=\$C\$7	Binding	0
\$J\$8	Nampan 4 solusi	15	\$J\$8<=\$C\$8	Binding	0
\$J\$9	Peti Harta Besar solusi	20	\$J\$9<=\$C\$9	Binding	0
\$J\$10	Peti Harta Kecil solusi	20	\$J\$10<=\$C\$10	Binding	0
\$J\$11	Vas Bunga Duduk solusi	18	\$J\$11<=\$C\$11	Binding	0
\$J\$12	Vas Bunga Lantai solusi	18	\$J\$12<=\$C\$12	Binding	0
\$J\$13	Lampu Gantung solusi	6	\$J\$13<=\$C\$13	Binding	0
\$J\$14	Lampu Dinding 1 solusi	9	\$J\$14<=\$C\$14	Binding	0
\$J\$15	Lampu Dinding 2 solusi	9	\$J\$15<=\$C\$15	Binding	0
\$J\$16	Hiasan Dinding solusi	20	\$J\$16<=\$C\$16	Binding	0
\$J\$17	Pigura 20R solusi	0	\$J\$17<=\$C\$17	Binding	0
\$J\$18	Pigura 85x65 solusi	12	\$J\$18<=\$C\$18	Binding	0
\$J\$19	Kere Gambar 1 solusi	15	\$J\$19<=\$C\$19	Binding	0
\$J\$20	Kere Gambar 2 solusi	10	\$J\$20<=\$C\$20	Binding	0
\$J\$21	Kere Gambar 3 solusi	8	\$J\$21<=\$C\$21	Binding	0

\$J\$22	Kere Gambar 4 solusi	8 \$J\$22<=\$C\$22	Binding	0
\$J\$2:\$J\$22=Integer				

Lampiran C. Hasil percobaan uji parameter

C.1 Uji parameter populasi

Pop = 25

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	386,75	8681900	19589000	11921000	331	31,3954
2	388	8741684	19850500	11849500	234	31,6578
3	379,5	8922628	19282000	11803000	419	31,3436
4	394	8170039	19601500	11733500	356	31,3066
5	403,25	8883542	19852000	11803000	236	31,3045
6	386,5	7850824	19319500	11625500	289	31,3331
7	395,75	8266355	19667500	11652500	266	31,3472
8	412	8546928	19769500	11795500	485	31,3319
9	382,25	8529505	19634000	11766000	456	33,9098
10	375,25	8916465	19523000	11657000	192	31,3034
Rata-rata				11760650	326,4	31,62333

Pop = 50

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	388,25	8625988	19828000	11932000	133	42,808
2	410,75	8475937	19902500	12057500	114	46,9433
3	369,25	8492326	19469000	11821000	209	46,7939
4	387	7795269,5	19472000	11778000	407	46,9356
5	389,25	8164945	19718000	11787000	245	46,9985
6	391,5	8192562,5	19583500	11791500	224	46,99
7	402	8946388,5	19787000	11803000	203	47,2118
8	402,5	8843738,5	19786500	11828500	178	47,0393
9	383,5	8360501	19239500	11665500	275	47,2627
10	390,75	8882243	19550500	11884500	119	46,9126
Rata-rata				11834850	210,7	46,58957

Pop = 100

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	414,75	8528454	19993500	11821500	140	106,7247
2	383,5	7768172	19532500	11957500	398	105,8302
3	396,5	8693952	19671000	11899000	307	105,8308
4	403	8657723	19996500	11853500	276	108,6451
5	377	7934931	19471500	11863500	226	106,1159
6	394	8385252	19569000	11901000	260	105,528
7	398,75	8893016	19718000	11847000	191	105,8569
8	382	8753913	19894500	12090500	422	107,1211
9	392	8692714	19956500	11883500	392	106,4407
10	384,75	7679935	19863000	12002000	154	106,2719
Rata-rata				11911900	276,6	106,43653

Pop = 200

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	394,25	8983238	19938500	12111500	313	338,4447
2	397,5	8148748,5	19738500	11801500	420	339,7543
3	391	8088342,5	19885500	11969500	136	340,1189
4	407,5	8222419	19410000	11995000	162	347,2913
5	406,5	8977421	19772000	11923000	208	341,332
6	403,75	7912763	19739000	12056000	228	340,8733
7	388	8700615	19860500	11829500	268	340,7597
8	400,25	8975105	19819500	11970500	351	339,9021
9	389,25	8847239,5	19806500	12013500	245	341,6288
10	392,25	8640156,5	19796000	12054000	396	339,7866
Rata-rata				11972400	272,7	340,98917

C.2 Uji parameter maksimum iterasi

Maxgen = 100

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	384,5	8838401	19137000	11608000	27	21,2656
2	398	8498988	19598000	11512000	51	21,302
3	412,25	8920791	19774000	11631000	66	22,5584
4	376	7706351	19410000	11530000	90	21,4971
5	405,75	8861758	19728500	11796500	70	21,2724

6	380,75	8630069	19464000	11371000	88	21,183
7	349,5	7901269	18693500	11611500	50	21,0619
8	409	8246553	19964000	11881000	43	21,1578
9	388,75	8742473	19477500	11592500	55	21,0763
10	406	8522088	19688500	11761500	74	21,2625
Rata-rata				11629500	61,4	21,3637

Maxgen = 200

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	396,25	8951559	19940500	11954500	142	42,487
2	374,75	8550852	19280500	11824500	100	42,9919
3	399,75	8501835	19905500	11964500	195	41,9838
4	412,75	8878840	19998000	11837000	175	42,1055
5	411	8846064	19920000	11785000	139	42,3004
6	393,5	8250060	19524000	11636000	137	42,099
7	394	7641494,5	19163000	11722000	150	42,396
8	411,25	8176341	19337000	11728000	110	42,4989
9	367,75	8384257	18968500	11791500	110	42,2262
10	399,25	8928333,5	19409500	11675500	121	42,6498
Rata-rata				11791850	137,9	42,37385

Maxgen = 500

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	398,5	8754780	19761500	11853500	367	106,1054
2	408,5	8700881	19959000	11781000	327	104,9198
3	393,25	8286012	19864500	11735500	274	105,2134
4	391,25	8724594	19831500	11753500	451	105,463
5	365,75	8570245	19546500	11823500	273	105,5879
6	414,5	8269926	19626500	11848500	450	104,3755
7	405	8125651	19402500	11847500	292	105,0903
8	386,5	8854166	19535500	11804500	269	105,3514
9	405,25	8966977	19565500	11779500	454	105,2083
10	387	7921611	19881000	11929000	265	105,715
Rata-rata				11815600	342,2	105,303

Maxgen = 1000

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	403,25	8897127	19724500	11875500	580	210,735
2	413	8613235	19926500	11933500	311	211,5676
3	396,25	8822758,5	19239500	11880500	670	211,6392
4	417,25	8834766	19999000	11856000	218	210,9255
5	374,25	8812265	19242000	11908000	508	211,5874
6	402,25	8999610	19876000	11929000	601	210,0918
7	364,5	8556948	19374000	11851000	812	211,9518
8	417,75	8975612	19822500	11857500	848	211,0517
9	401,25	8487530	19872000	11828000	788	211,2686
10	393,25	7906707,5	19898500	12071500	752	218,0263
Rata-rata				11899050	608,8	211,88449

C.3 Uji parameter wmax

Wmax = 0,3

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	375,25	7803665	19140000	11735000	263	107,8325
2	404,5	8940339	19897000	12083000	349	107,3387
3	373,25	8827746	19701000	11969000	150	107,0501
4	379,5	7942771	19716000	11909000	387	106,7897
5	400	8069240	19776000	11784000	146	106,7477
6	394,75	8305934	19966000	11989000	222	106,7879
7	399,75	8885206	19645000	11820000	455	106,6846
8	407	8817951	19724000	11951000	159	107,1982
9	402,25	8730794	19635000	11725000	326	106,8184
10	387,25	8664979	19564500	11745500	447	106,8237
Rata-rata				11871050	290,4	107,00715

Wmax = 0,5

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	420,75	8016906	19957500	12112500	132	105,8733
2	395	8999014	19885000	11880000	450	106,6127
3	386,5	8544881	19822500	11877500	423	105,4995
4	380,5	8960596	19745500	12104500	154	105,5206
5	384	8359096,5	19842000	11978000	432	105,6713

6	400,25	8418245	19994000	12111000	462	105,609
7	429,25	8588199	19825500	11949500	282	106,0236
8	396	8915278	19392500	11837500	379	105,8126
9	404,5	8632707	19796500	11963500	475	105,5429
10	416	8334361,5	19705500	12044500	407	105,7642
Rata-rata				11985850	359,6	105,79297

$W_{max} = 0,7$

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	407	8801772	19980500	11914500	232	106,8976
2	385,25	8895519	19936500	12013500	142	105,9017
3	402,25	8253714	19187000	11853000	212	104,8405
4	396	8809267	19438500	11836500	414	105,582
5	381,5	7638065	19659500	11970500	183	106,1353
6	400,75	8853301	19807500	11937500	421	105,2544
7	399,25	8911547	19564000	11986000	174	105,6273
8	381,25	8663176	19794500	11895500	170	107,2321
9	397,25	8030441	19712500	11802500	326	105,2857
10	401,5	8700423	19959000	11961000	200	105,5241
Rata-rata				11917050	247,4	105,82807

$W_{max} = 0,9$

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	394,25	8823077,5	19720000	11900000	348	105,1771
2	404,5	8926230	19836000	12029000	210	105,0564
3	401,5	8785224	19703000	11852000	205	105,5502
4	366,75	8409184,5	19337500	11812500	213	105,2439
5	404	8576379	19949000	12091000	384	105,1187
6	391,75	8176195	19730000	11910000	295	105,1012
7	401,75	8992073	19711500	11983500	148	105,5335
8	408	8842673,5	19986500	11678500	480	105,0522
9	374,25	8497256,5	19662500	12017500	446	105,1484
10	414,75	8869669	19909500	11905500	435	107,0546
Rata-rata				11917950	316,4	105,40362

C.4 Uji parameter wmin

Wmin = 0,1

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	401,75	8151772	19887500	11812500	242	110,5617
2	405,75	8524895	19968000	11737000	265	109,7189
3	396	8379023	19885000	12005000	348	110,6345
4	401,75	8151772	19887500	11812500	242	113,2049
5	405,75	8524895	19968000	11737000	265	108,0279
6	366,25	8193679	19313000	11967000	338	107,5972
7	405,75	7938820	19291500	11938500	257	110,7135
8	380,25	8530991	19773500	11781500	357	108,5103
9	408,25	8578400	19755000	11800000	219	107,2215
10	414,75	8841377	19980500	11754500	423	110,1655
Rata-rata				11834550	295,6	109,63559

Wmin = 0,15

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	413	8966287	19908000	11877000	441	107,5059
2	398,5	8270638	19839500	11880500	338	106,9581
3	398	8095610,5	19886500	11883500	266	106,1873
4	420,75	8490048,5	19769000	12021000	179	106,6145
5	396,25	8292887	19965000	11855000	406	107,4322
6	394,5	8776286	19699000	11736000	430	108,0166
7	377	8426139,5	19805500	12039500	316	107,1009
8	397,75	8950039,5	19532500	11757500	406	107,2358
9	383,5	8859528	19893000	11882000	313	107,9404
10	400,25	8678920	19587000	11898000	155	106,8864
Rata-rata				11883000	307,1	107,18781

Wmin = 0,2

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	399,75	8691359	19989000	11826000	408	107,5331
2	404	8054643	19803000	11862000	160	106,981
3	380,75	8995263	19731000	11759000	373	106,7328
4	404	8273554	19962500	11907500	422	110,6162
5	366,5	8570675	19575500	11939500	300	112,2539

6	410	8980977	19819000	11821000	453	107,6044
7	407,5	8680351	19974500	11885500	249	107,7417
8	378,75	8811552	19867500	11952500	383	107,3584
9	402	8168847	19800500	11929500	461	107,7911
10	397,5	8677326	19759000	11746000	185	107,8137
Rata-rata				11862850	339,4	108,24263

Wmin = 0,25

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	401,75	8930902,5	19749500	11810500	216	107,6463
2	394,75	8807998	19770500	11809500	462	107,2615
3	390,5	8632542,5	19564000	11946000	260	107,7795
4	400	8931467,5	19706500	11803500	416	107,2366
5	403,75	8889581	19770000	11745000	238	107,5617
6	376	8644396	19582000	11733000	149	107,5527
7	380,75	7620237	19359500	11880500	170	107,5165
8	381,75	8846033	19994000	12076000	241	107,3398
9	406,25	8857770,5	19709500	11970500	378	106,755
10	397,25	8176463,5	19983000	11967000	389	107,2841
Rata-rata				11874150	291,9	107,39337

C.5 Uji parameter s

s = 0,01

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	390	8783249	19606500	11743500	366	113,3277
2	408,75	8680955	19972500	11877500	490	111,0956
3	406,5	8989417	19976500	11943500	213	111,3471
4	382,25	8914047	19462000	11893000	239	110,6502
5	404,75	8744216	19789000	11916000	461	110,9188
6	394,75	8878070	19250500	11809500	454	110,5848
7	372,5	8040578	19232500	11667500	212	110,947
8	369,5	8246623	19030000	11710000	145	111,3598
9	406,5	8170830	19428500	11926500	471	111,2914
10	400	8571767	19726000	11859000	163	111,2925
Rata-rata				11834600	321,4	111,28149

s = 0,1

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	405,5	8703469,5	19339000	11871000	119	107,0904
2	409	8773786	19478500	11826500	386	112,0962
3	405,25	8677748	19561000	11914000	182	107,5078
4	396,25	8451742	19854500	11910500	404	107,2387
5	404,25	8944620,5	19973500	11906500	381	107,4974
6	394,75	8720371,5	19557000	11843000	447	108,3365
7	398,25	8897028	19604000	11816000	352	107,1022
8	389,25	8927154	19859000	11866000	112	106,7689
9	383,25	7830481	19396000	11759000	122	107,4413
10	376,25	8003439	19728000	11947000	194	106,2393
Rata-rata				11865950	269,9	107,73187

s = 0,2

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	399,5	8495893	19891500	11913500	442	107,3619
2	410	8609134	19986000	12054000	473	107,3802
3	364,75	8376706	19648000	11822000	261	108,8205
4	392	8787489	19693000	11847000	315	107,2082
5	381,5	8947872	19177500	11742500	229	109,5284
6	399,25	8914672	19884500	11940500	392	109,4321
7	396,75	8975579	19582500	11847500	100	107,0552
8	403,75	8898229	19655500	11929500	394	107,5168
9	397	8154337	19716000	11804000	474	107,2384
10	397,75	8824543	19627500	11822500	114	107,1614
Rata-rata				11872300	319,4	107,87031

s = 0,5

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	Komputasi
1	398	8898907	19671500	11873500	242	109,7377
2	393,25	7809048	19738500	11996500	115	107,1212
3	403,25	8906413	19889000	12166000	278	106,9498
4	414,75	8680986,5	19975000	12020000	380	106,6296
5	384,5	7912522	19403000	11802000	318	107,1139
6	387	8914151,5	19975000	11895000	258	106,9773

7	393	7986331	19761500	11958500	495	107,1427
8	375,5	8149222	19415000	12095000	189	107,5291
9	404,75	8918240	19925500	11894500	133	107,3928
10	381,5	8796412	19576000	11869000	486	107,1264
Rata-rata				11957000	289,4	107,37205

s = 1

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	402,5	8809352	19626500	11818500	274	108,1884
2	387,25	7763616,5	19856500	11968500	374	108,4244
3	389,25	8637664	19408000	11732000	178	108,3723
4	376,5	8305361	19838500	11891500	476	108,0115
5	403,5	8719554	19967000	11873000	368	107,7525
6	402,75	8454022	19771000	11834000	342	107,0056
7	405,75	8730034	19917000	11813000	281	107,5233
8	395,75	7981850,5	19819000	11921000	317	107,8794
9	405,75	8908856,5	19930500	11804500	332	107,5521
10	400,5	8537575,5	19896000	11844000	447	107,7623
Rata-rata				11850000	338,9	107,84718

C.6 Uji parameter c

c = 0,01

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	390,25	7927940	19727000	11948000	114	110,5867
2	387,75	7902923	19952000	12083000	154	109,6209
3	400,25	8459737	19989500	12120500	213	113,9434
4	401,5	8853768	19433000	11787000	132	107,4182
5	402	8701487	19507000	11878000	211	107,2566
6	397,75	8931610	19611500	11888500	248	107,7679
7	408	8672713	19950500	11854500	270	107,4323
8	404,5	8893954	19549000	11846000	193	108,2075
9	404,25	8439097	19913000	11887000	327	108,2636
10	390	8928955	19959500	11875500	388	107,6536
Rata-rata				11916800	225	108,81507

c = 0,1

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	403	8636482	19973000	11872000	276	107,3148
2	401	8874788	19731000	11959000	469	107,5495
3	375,25	8710126,5	19591000	11864000	140	107,935
4	388,5	8064180,5	19593500	11921500	202	107,1401
5	381,5	8717467	19798000	12022000	264	107,9707
6	398	8932552	19618000	11782000	313	107,7064
7	408,5	8394702,5	19852500	11757500	336	108,9801
8	397	8437356	19977000	11898000	193	108,1109
9	402,25	8941899,5	19656500	11798500	243	107,6953
10	406	8713446,5	19809000	11806000	120	107,7704
Rata-rata				11868050	255,6	107,81732

c = 0,2

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	398,75	8083489	19805000	12040000	330	108,5993
2	411,5	8786988	19987000	11893000	411	108,8582
3	417,25	8226680	19806000	11929000	383	109,4544
4	386,5	8724646	19860000	11970000	229	108,3139
5	406,25	8482682	19828500	11841500	349	108,0234
6	382	8505795	19753500	11826500	349	107,8426
7	397	8980958	19672500	11937500	124	107,5134
8	401,25	8557411	19946500	11883500	242	109,0378
9	382,75	8739284	19898000	12132000	440	108,2637
10	371,75	8464800	19431500	11758500	363	108,1264
Rata-rata				11921150	322	108,40331

c = 0,5

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	364	8870192,5	19491000	11869000	432	107,9458
2	424	8280473	19845500	11959500	117	107,6046
3	394,75	8565247	19429000	11821000	361	107,6419
4	417	8516596	19500500	11944500	175	109,2813
5	385	8801980	19869000	12011000	120	107,7165
6	416,25	8934097	19812000	11823000	101	107,8206

7	408,75	8841976	19914500	11805500	271	108,2411
8	406,75	7997690	19258500	11766500	367	107,7977
9	415	8769220,5	19901000	11744000	419	108,1991
10	405,25	8822980	19783500	11926500	347	107,8977
Rata-rata				11867050	271	108,01463

c = 1

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	398,75	8945685,5	19863500	11826500	241	108,9006
2	392,25	8569628	19811500	11843500	177	108,1944
3	402	7950862	19216000	11834000	376	108,3178
4	388,75	8537105	19270500	11769500	211	108,2705
5	414	8730498	19962000	12033000	398	107,9967
6	378,75	8466653,5	19913000	12062000	363	107,9407
7	406,5	8439278,5	19762000	11938000	305	108,0335
8	412,5	8325811	19506000	11979000	227	108,0746
9	407,75	8774263	19896000	11849000	261	107,7834
10	383,75	8729695,5	19641500	11753500	420	108,2266
Rata-rata				11888800	297,9	108,17388

C.7 Uji parameter a

a = 0,01

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	393,75	8967584	19734500	11750500	108	107,6307
2	369,75	8419856	19595500	11994500	139	108,3944
3	405,75	8480421	19984500	11865500	423	107,8
4	408	8879221	19977500	11907500	140	107,2478
5	397	8100249	19829500	11920500	283	107,801
6	391	8987098	19799000	11846000	304	108,0969
7	430,5	8492162	19918500	12121500	354	107,6311
8	399,25	8270552	19795500	11939500	187	108,5162
9	397,5	8921680	19774000	12016000	224	107,9738
10	401	8133308	19841500	11988500	238	107,6158
Rata-rata				11935000	240	107,87077

a = 0,1

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	406,5	8391440,5	19938500	11886500	173	107,6809
2	400,25	8452499	19752000	11868000	147	107,6692
3	386,5	8742615	19833500	11881500	168	107,775
4	401,5	8995176,5	19714000	11981000	240	107,9374
5	388	8637339,5	19960000	11935000	203	107,3109
6	390,75	8607095	19537500	11822500	135	107,331
7	385,5	8938126	19342500	11827500	262	108,1559
8	354	7955888	19089000	11836000	333	108,2623
9	401,5	8983355,5	19691000	11974000	429	107,8245
10	408,25	8935172,5	19832500	11962500	111	107,7491
Rata-rata				11897450	220,1	107,76962

a = 0,2

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	410,5	8980863	19887000	11898000	282	107,7492
2	406,75	8040811	19143000	11897000	215	107,6285
3	388,5	8040093	19807000	11963000	338	107,1039
4	409,25	8995172	19592500	11957500	123	107,5977
5	394,5	8990097	19939500	11790500	365	107,6525
6	416,75	8968323	19951000	11854000	447	107,9202
7	390,75	7986972	19723000	11972000	280	107,7297
8	402	8830251	19807500	11807500	313	107,0741
9	404,5	8857710	19878500	11946500	161	107,3546
10	391,25	8860629	19522500	11742500	486	107,0944
Rata-rata				11882850	301	107,49048

a = 0,5

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	410,25	8973068,5	19871500	11888500	218	107,3323
2	388	7859638	19532500	11802500	187	107,6525
3	377,25	8463632	19709500	12000500	154	107,4622
4	407,75	8378023,5	19831500	11878500	255	107,8274
5	395,25	8909065	19604500	11870500	136	107,5668
6	394	8970197	19965000	11870000	324	107,1196

7	387,5	7905899,5	19603500	11886500	285	107,8801
8	405,25	8411611,5	19869500	11865500	333	107,4049
9	400,5	8668280,5	19616500	11803500	455	107,8288
10	384,5	8822253,5	19860000	11860000	394	108,003
Rata-rata				11872600	274,1	107,60776

a = 1

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	408,5	8570841	19946000	11824000	326	107,1829
2	397,5	8785096,5	19859500	11735500	485	107,3999
3	401,25	8977889,5	19985500	11839500	403	108,9937
4	382	7790292	19733500	12096500	155	107,8369
5	404,75	8943244,5	19550000	11760000	330	108,0542
6	400	8627869	19901500	11913500	198	107,0635
7	393,5	7933745	19749000	11796000	203	107,8082
8	375,25	8595118	19541000	11914000	464	107,2846
9	379,5	7918445	19639000	11866000	352	108,3968
10	411,75	8839589,5	19637000	11863000	353	114,9195
Rata-rata				11860800	326,9	108,49402

C.8 Uji parameter f

f = 0,01

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	373,75	7798008	19429000	11801000	257	108,7293
2	382	7950524	19510000	11945000	397	106,3774
3	405,75	8332231	19603500	11911500	397	110,0825
4	398,75	8435851	19941000	12049000	217	107,1465
5	380,25	8706579	19088500	11706500	457	106,8431
6	411	8953162	19632500	11802500	335	106,6136
7	394,25	8765777	19531000	11874000	256	107,4162
8	401,75	8870532	19897000	11933000	212	107,2064
9	395	8994307	19934500	11915500	347	106,718
10	408,75	8844985	19772000	11743000	210	106,6481
Rata-rata				11868100	308,5	107,37811

f = 0,1

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	415,75	8640082,5	19996000	11689000	468	107,6297
2	397,5	8958301	19551500	11878500	252	107,594
3	397	7667791	19540000	11935000	202	107,5878
4	397	8815316,5	19858000	11907000	187	106,875
5	415,5	8245346	19499500	11865500	305	106,9347
6	407,5	8365241,5	19765000	11895000	430	106,8761
7	398,25	8257725	19726000	11894000	474	107,2726
8	405,5	8248451	19206500	11938500	477	106,9392
9	386	8881943	19554000	11801000	241	111,5907
10	405,75	8528805	19842500	12052500	104	106,8364
Rata-rata				11885600	314	107,61362

f = 0,2

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	418,25	8809061	19976000	11754000	229	107,0998
2	399,25	8090103	19855500	11944500	237	107,2614
3	390	8843911	19828500	11861500	251	110,1113
4	362,75	8187577	19234500	11845500	267	111,3646
5	412,75	8840579	19996500	11933500	472	107,1285
6	391	8952854	19750500	11849500	327	107,0344
7	393	8017296	19735500	12039500	359	107,5291
8	382	8906115	19862000	12033000	312	107,2171
9	369,5	8315196	19749000	11851000	362	107,0963
10	374,75	8640680	19551500	11758500	389	107,7175
Rata-rata				11887050	320,5	107,956

f = 0,5

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	406,5	8959715,5	19634500	11985500	244	108,4142
2	355,5	7806906	19319500	11875500	129	108,1091
3	399,75	7781942	19861500	12108500	402	108,5036
4	404,25	8724605	19849500	11780500	416	115,5716
5	413,25	8343369,5	19545000	11845000	297	112,8608
6	389,75	8948944	19938500	11996500	282	112,6199

7	357,5	8194808,5	19036500	11833500	101	111,8014
8	400	8189345,5	19063500	11751500	474	113,2938
9	385	8985853	19635500	11809500	108	113,0633
10	368,25	8474923	19756500	11978500	319	115,1206
Rata-rata				11896450	277,2	111,93583

f = 1

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	380,5	8615006	19762000	11943000	400	113,202
2	397	8644166	19737500	11902500	203	113,3939
3	374,75	8676913	19901500	12003500	499	115,986
4	411,25	8883707,5	19764500	11850500	278	113,4087
5	387	8717240,5	19450500	11714500	499	113,535
6	398	8086505,5	19691000	11854000	232	115,2557
7	399,5	8908842,5	19571500	11838500	320	111,077
8	397	8904435	19408000	11787000	376	110,5448
9	393,5	8345318	19728500	11951500	321	120,4552
10	396,25	8607311,5	19736500	11963500	107	114,9569
Rata-rata				11880850	323,5	114,18152

C.9 Uji parameter e

e = 0,01

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	385	8021073	19946500	11943500	117	111,3802
2	409	8773786	19478500	11826500	386	108,6779
3	371,75	8454838	19324000	11786000	268	107,9587
4	391	8840705	19470500	11889500	444	118,0634
5	398,25	8002388	19777000	11858000	250	117,9224
6	376,25	8444437	19923500	12061500	216	118,3984
7	413,25	8770657	19912000	11918000	404	118,0879
8	403,25	8889833	19638000	11742000	223	107,6219
9	426	8713312	19791500	12168500	325	107,1285
10	410,5	8262771	19958000	11887000	444	107,0421
Rata-rata				11908050	307,7	112,22814

e = 0,1

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	417,75	8739404,5	19950500	11964500	204	106,3583
2	403	8789513,5	19958000	11892000	243	106,5151
3	396,5	8202525,5	19596500	11813500	440	106,8097
4	399,5	8985153,5	19836500	11848500	291	106,6525
5	388	8253087,5	19429500	11925500	336	106,3188
6	397,75	8125417	19997500	12012500	210	106,998
7	403,5	8297335,5	19974500	11920500	402	106,3313
8	394,25	7994184	19939000	12106000	119	106,8452
9	406,25	8943672	19986000	11714000	387	107,4082
10	398,75	8181674	19917500	11977500	371	107,1732
Rata-rata				11917450	300,3	106,74103

e = 0,2

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	373,25	8473280	19743500	11751500	210	107,1616
2	392,25	8053950	19702500	11822500	202	108,7131
3	403,75	8728198	19804500	11830500	243	107,3458
4	393,25	8952124	19715000	11955000	207	109,0979
5	408	8679291	19898000	11867000	493	108,5566
6	399,25	8718728	19618000	11967000	248	107,473
7	404	8980682	19817500	11812500	361	107,4555
8	390	8935110	19754000	11901000	395	107,5789
9	386	8968775	19580000	11815000	344	107,8218
10	409,5	8930180	19804000	11966000	201	107,6817
Rata-rata				11868800	290,4	107,88859

e = 0,5

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	374,25	7646294,5	19276500	11728500	373	108,0958
2	402,25	7699880,5	19414000	11881000	436	108,5071
3	399,75	8695253	19883000	11917000	243	108,3455
4	399,75	8794850	19965000	11985000	100	107,9695
5	399,75	8696582	19794500	11760500	272	108,5571
6	401,25	8710309,5	19858500	11931500	187	108,8572

7	400,5	8504230,5	19954000	11826000	473	107,8384
8	410,25	8471238,5	19926000	11829000	185	108,2651
9	406,75	8770487,5	19476000	11709000	448	107,946
10	409,5	8877860	19786500	11843500	195	108,3377
Rata-rata				11841100	291,2	108,27194

e = 1

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	389,5	8462380	19569500	11800500	351	109,3232
2	401,5	8912417	19686500	11813500	335	110,7862
3	400	8921522	19345500	11809500	456	108,961
4	390,75	8490958	19902500	11702500	360	109,0308
5	366,25	8570107	19159500	11600500	280	108,789
6	408,75	8862802	19777500	11762500	408	108,7771
7	395	8213973,5	19680000	11660000	446	108,6757
8	410,25	8795839,5	19586500	11683500	177	108,6688
9	407,75	8996181	19517500	11632500	241	108,1475
10	400,75	8789855,5	19522000	11783000	311	107,7911
Rata-rata				11724800	336,5	108,89504

C.10 Uji parameter Range

R = 0,5

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	410,25	8729348	19902500	11877500	460	62,0446
2	419,25	8512316	19611000	11904000	439	60,0961
3	410,5	8809713	19981000	11849000	379	59,9322
4	412	8734471	19873500	11656500	150	59,412
5	401,5	8859704	19791000	11889000	358	59,3836
6	406,75	8680555	19964500	11960500	269	60,5594
7	416,5	8770977	19968000	11792000	349	59,6123
8	415	8891871	19995000	11855000	154	60,0138
9	413	8818244	19969500	11675500	246	59,7145
10	405,75	8937354	19923000	11852000	367	60,3195
Rata-rata				11831100	317,1	60,1088

R = 1,25

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	379	8776589	19766000	12104000	486	71,8065
2	405,25	8795536	19983500	12046500	123	70,4659
3	397,25	8276470	19909500	12075500	349	73,3109
4	421	8260012	19690500	12039500	447	71,9718
5	409,75	8356190	19963500	12081500	488	71,2829
6	390,25	7875875,5	19933000	12132000	452	71,9205
7	385,75	8613886	19948000	12052000	344	71,5092
8	391,25	8289589,5	19809000	12221000	176	71,6546
9	400,25	8974411,5	19784500	11895500	333	71,4692
10	379,75	8993994,5	19974000	12086000	118	72,2252
Rata-rata				12073350	331,6	71,76167

R = 2

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	396,25	8065490	19954500	12040500	468	111,0498
2	403,25	8353928	19810000	11835000	340	111,4292
3	362,5	8149107	19430500	11829500	158	112,231
4	408	8869439	19845500	11824500	455	110,6264
5	381,5	8464222	19625000	11895000	457	111,7053
6	407,75	8868744	19931500	11908500	245	113,5492
7	387,25	8642429	19476500	11863500	287	116,1772
8	408,75	8994752	19882000	11883000	447	110,9956
9	400	8979362	19743000	11832000	142	113,866
10	392,5	8572180	19948500	12071500	329	111,5715
Rata-rata				11898300	332,8	112,32012

R = 2,75

No	Berat	Volume	Biaya Produksi	Profit	iterasi konvergen	komputasi
1	374,75	8521926	19607000	11893000	127	115,9858
2	395,75	8700736	19779000	11796000	224	114,8906
3	379,25	8703996	19411500	11763500	409	116,929
4	387	8003240	19920500	12019500	108	119,3463
5	405,25	8988534	19748500	11826500	284	119,7684
6	384,75	8373437	19714500	12055500	268	116,4817

7	402,25	8915233	19704000	11826000	274	114,1601
8	408,5	8680728,5	19972500	11842500	350	114,6117
9	415	8789893	19969000	11836000	282	114,6922
10	386	8868367	19689500	11820500	345	114,6387
Rata-rata				11867900	267,1	116,15045

Lampiran D. Pseudocode DOA

```

Input : Populasi (n)
         Maksimal iterasi (Iter)
         Banyak barang (brg)
         Jumlah barang (Jml)
         Berat barang (brt)
         Volume barang (vol),
         Biaya produksi (By)
         Harga Jual (Jl)
         berat pemisahan (s)
         berat keselarasan (a)
         berat kohesi (c)
         faktor makanan (f)
         faktor musuh (e)
         berat inersia maksimum (wmax)
         berat inersia minimum (wmin)
         Kapasitas berat (max_berat)
         Kapasitas volume (max_volume)
         Modal (M)

{Inisialisasi populasi}
For i=1:n
  For j=1:brg
    X[i,j]=rand(0,1)
    Y[i,j]=round(X[i,j]*Jml(j))
    dX[i,j]=0
  End
End

{Periksa kendala}
For i=1:n
  For j=1:brg
    Berat[i]=berat(i)+brt(j)*Y[i,j]
    Volume[i]=volume(i)+vol(j)*Y[i,j]
    Biaya[i]=biaya(i)+By(j)*Y[i,j]
  End
  While
    (berat(i)>Max_berat) and (volume(i)>Max_volume) and (Biaya(i)>M)
    For j=1:brg
      Y[i,j]=round(rand(0,1)*Jml(j))
      Berat[i]=berat(i)+brt(j)*Y[i,j]
    
```

```

        Volume[i]=volume(i)+vol(j)*Y[i,j]
        Biaya[i]=biaya(i)+By(j)*Y[i,j]
    End
End
End
{Hitung Profit}
For i=1:n
    Zmax=Jl(1)-By(1)*Y(1,1)
    Zmin=Zmax
    InMax=1
    InMin=1
    For j=1:brg
        Profit(i)=(Jl(j)-By(j))*Y[i,j]+Profit(i)
        If Profit(i)>Zmax
            Zmax=Profit(i)
            InMax=i
        End
        If Profit(i)<Zmin
            Zmin=Profit(i)
            InMin=i
        End
    End
End
End

{Sumber makanan}
Zmax = 0
idx1 = 0
For i = 1 : n
    If Z[i] > Zmax
        Zmax = Z[i]
        idx1 = i
    End
    For j = 1 : brg
        X+[j] = X[idx1,j]
    End
End
Z+ = Zmax

{Sumber Musuh}
Zmin = inf
idx2 = 0
For i = 1 : n
    If Z[i] < Zmin
        Zmin = Z[i]
        idx2 = i
    End
    For j = 1 : brg
        X-[j] = X[idx2,j]
    End
End
End
Z- = Zmin

```

```

{Tetangga}
For i = 2:n
  For j = 1:brg
    r[i]=r[i]+sqrt[(X[i,j]-X[1,j])^2]
  End
End
End

{parameter}
Lbh=0;krq=0
For i = 1:n-1
  If r[i]<0,5
    Krg=krq+1
    For j = 1:brg
      S[i]=X[1,j]-X[i,j]+S[i]
    End
  End
  If r[i]>=0,5
    For j = 1:brg
      X[I,j]=rand[-1,1]*X+[j]
    End
  End
  For i =1:n
    For j =1:brg
      If r[i]<0,5
        C[i]=((X[I,j]/krq)-X[i,j])+C[i]
        F[i]=X+[j]-X[1,j]
        E[i]=X-[j]+X[1,j]
        dX[i,j]=(s*S[i]+c*C[i]+f*F[i]+e*E[i])+w*dX[i,j]
        X[i,j] =X[i,j]+dX[i,j]
      End
    End
  End
End
End
End

```