



**PENERAPAN *WHALE OPTIMIZATION ALGORITHM* (WOA)
PADA *QUADRATIC KNAPSACK 0-1 PROBLEM***

SKRIPSI

Oleh:

**Nia Media Anggita
NIM 161810101027**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2020**



**PENERAPAN *WHALE OPTIMIZATION ALGORITHM* (WOA)
PADA *QUADRATIC KNAPSACK 0-1 PROBLEM***

SKRIPSI

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

Oleh:

**Nia Media Anggita
NIM 161810101027**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2020**

PERSEMBAHAN

Bismillahirrahmanirrahim, puji syukur kehadiran Allah SWT atas berkat, rahmat dan hidayah-Nya sehingga saya bisa menyelesaikan tugas akhir ini dengan lancar. Shalawat serta salam selalu tucurahkan kepada junjungan kita Nabi Muhammad SAW yang telah membimbing umat islam meninggalkan zaman jahiliah. Tugas akhir ini saya persembahkan untuk orang-orang spesial, yaitu:

1. Ibu Santik Rahayu, ibu tercinta yang telah susah payah melahirkan dan membesarkan saya hingga saat ini.
2. Ayah Sugiarto dan Mama Novitasari yang selalu memberikan dukungan verbal maupun finansial.
3. Adik saya Novi Rahmadani yang selalu memberikan semangat dan menjadi tempat berbagi suka duka.
4. Sahabat saya Ari Susiana dan Melinda Asti yang selalu menjadi motivator dalam segala hal.
5. Saudara kedua saya Nowiatun Khotijah dan Rofidatun Khusniah yang sudah menjadi teman hidup selama di Jember.
6. Almamater tercinta Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam.

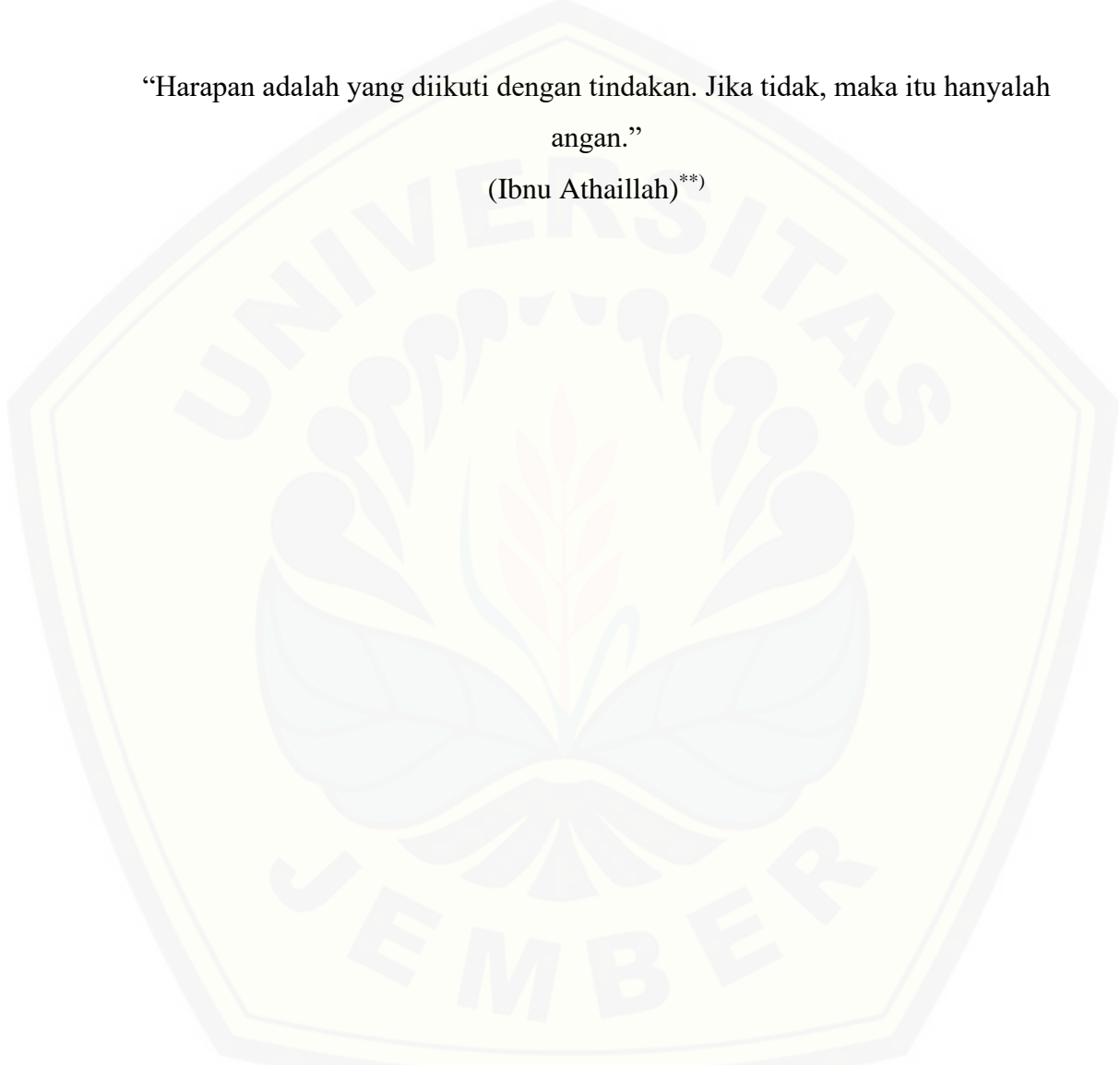
MOTO

“Ilmu tanpa agama adalah lumpuh, agama tanpa ilmu adalah buta.”

(Albert Einstein)^{*)}

“Harapan adalah yang diikuti dengan tindakan. Jika tidak, maka itu hanyalah
angan.”

(Ibnu Athaillah)^{**)}



^{*)} Rafael. 2015. Kata Mutiara Bijak Albert Einstein.
<https://8limbmuaythai.com/ini-dia-kata-mutiara-bijak-albert-einstein>. [Diakses
pada 28 April 2020]

^{**)} Fathurrohlim. 2018. Kata Mutiara Islam Ibnu Athaillah.
<https://mutiaraislam.net/kata-bijak-mutiara-islam-ibnu-athaillah/>. [Diakses 28
April 2020]

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Nia Media Anggita

NIM : 161810101027

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Penerapan *Whale Optimization Algorithm (WOA)* pada *Quadratic Knapsack 0-1 Problem*” adalah benar-benar hasil karya sendiri, kecuali kutipan-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 adanya tekanan dan paksaan dari pihak manapun dan bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, Agustus 2020

Yang menyatakan,

Nia Media Anggita

NIM 161810101027

SKRIPSI

**PENERAPAN *WHALE OPTIMIZATION ALGORITHM* (WOA) PADA
*QUADRATIC KNAPSACK 0-1 PROBLEM***

Oleh

Nia Media Anggita

NIM 161810101027

Pembimbing:

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

Dosen Pembimbing Anggota : Dr. Agustina Pradjaningsih, S.Si., M.Si.

PENGESAHAN

Skripsi berjudul “Penerapan *Whale Optimization Algorithm* (WOA) pada *Quadratic Knapsack 0-1 Problem*” karya Nia Media Anggita telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Jember

Tim Penguji:

Ketua,

Anggota I,

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

Dr. Agustina Pradjaningsih, S.Si., M.Si.
NIP. 197108022000032009

Anggota II,

Anggota III,

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

Dr. Kristiana Wijaya, S.Si., M.Si.
NIP. 197408132000032004

Mengesahkan

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Jember

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

RINGKASAN

Penerapan *Whale Optimization Algorithm* (WOA) pada *Quadratic Knapsack 0-1 Problem*; Nia Media Anggita, 161810101027; 2020; 60 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Knapsack Problem merupakan masalah dimana orang dihadapkan pada persoalan optimasi pada pemilihan benda yang dapat dimasukkan ke dalam sebuah wadah yang memiliki keterbatasan ruang atau daya tampung. Dengan adanya optimasi dalam pemilihan benda yang akan dimasukkan ke dalam wadah tersebut diharapkan dapat menghasilkan keuntungan yang maksimum. *Knapsack Problem* terbagi menjadi beberapa jenis, salah satunya adalah *Knapsack 0-1 Problem* dimana barang yang dimasukkan ke dalam media penyimpanan harus semua atau tidak dimasukkan sama sekali. Salah satu permasalahan yang termasuk di dalam permasalahan *knapsack* adalah *Quadratic Knapsack 0-1 Problem*. *Quadratic Knapsack 0-1 Problem* merupakan permasalahan *knapsack* yang tujuannya memaksimalkan fungsi tujuan dalam bentuk kuadratik untuk kendala kapasitas biner dan linier.

Penelitian ini membahas Penerapan *Whale Optimization Algorithm* (WOA) pada *Quadratic Knapsack 0-1 Problem*. *Whale Optimization Algorithm* (WOA) merupakan algoritma yang didasarkan pada perilaku paus bungkuk (*Humpback Whale*) dalam mencari makanan. Hal menarik dari paus bungkuk dalam mencari makanan adalah cara berburu yang istimewa dengan menggunakan metode perangkap gelembung. Percobaan kali ini menghitung keuntungan maksimum yang didapatkan oleh sebuah toko kebutuhan rumah tangga dengan menerapkan *Quadratic Knapsack 0-1 Problem*. Data yang digunakan adalah data berat, harga beli, dan harga jual dari 60 jenis barang.

Algoritma WOA ini digunakan untuk mengetahui solusi terbaik penyelesaian *Quadratic Knapsack 0-1 Problem*. Parameter yang digunakan untuk percobaan adalah banyaknya populasi paus (N) dan iterasi maksimum dimana istilah paus tersebut merujuk pada kombinasi barang yang akan diangkut.

Parameter N digunakan untuk menentukan banyak kandidat solusi berupa kombinasi barang yang akan dimasukkan ke dalam *knapsack*. Parameter iterasi maksimum digunakan untuk menentukan iterasi maksimum yang akan digunakan. Setiap percobaan dilakukan sebanyak 10 kali *running*. Dari percobaan yang dilakukan, nilai parameter kombinasi barang dan iterasi maksimum sangat berpengaruh terhadap nilai total profit dan *running time*-nya. Semakin banyak kombinasi barang dan iterasi maksimum maka semakin optimal hasil (total profit) yang diperoleh dan semakin lama *running time* yang diperlukan. Namun, penambahan parameter kombinasi barang lebih cepat mencapai optimal daripada parameter iterasi maksimum.

Profit optimal yang diperoleh dari penyelesaian *Quadratic Knapsack 0-1 Problem* menggunakan algoritma WOA dengan 60 data tersebut adalah sebesar Rp. 2.599.149,00 dengan total berat barang sebesar 449.310 gram. Dari beberapa percobaan yang dilakukan, parameter 1000 kombinasi barang dengan 500 iterasi maksimum memerlukan *running time* yang paling cepat mencapai optimal yaitu 194,2554 detik dan parameter 100 kombinasi barang dengan 5.000 iterasi maksimum memerlukan *running time* paling lama yaitu 504,3119 detik. Untuk mendapatkan nilai tersebut tidak membutuhkan iterasi yang lebih banyak.

Hasil tersebut menunjukkan bahwa algoritma WOA dapat dikatakan efektif karena dari setiap percobaan, algoritma selalu konvergen atau menuju profit maksimum. Selain itu, dengan parameter yang tepat, algoritma mampu mencapai hasil yang optimal. Algoritma WOA juga dapat dikatakan efisien karena untuk mendapatkan profit optimal tidak membutuhkan waktu dan iterasi yang lebih banyak.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala berkat, rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Penerapan *Whale Optimization Algorithm* (WOA) pada *Quadratic Knapsack 0-1 Problem*” ini. Penulisan tugas akhir ini disusun guna memenuhi salah satu syarat untuk menyelesaikan Program Studi Matematika (S1) dan mencapai gelar Sarjana Sains pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penulis menyadari bahwa dalam proses penulisan tugas akhir ini banyak menerima bantuan, bimbingan, motivasi, serta semangat dari berbagai pihak. Oleh karena itu, dengan segala hormat penulis ingin mengucapkan terimakasih kepada:

1. Bapak Abduh Riski, S.Si., M.Si. selaku dosen pembimbing utama dan Ibu Dr. Agustina Pradjaningsih, S.Si., M.Si. selaku dosen pembimbing anggota.
2. Bapak Kosala Dwidja Purnomo, S.Si., M.Si. selaku dosen penguji utama dan Ibu Dr. Kristiana Wijaya, S.Si., M.Si. selaku dosen penguji anggota.
3. Seluruh dosen dan karyawan Jurusan Matematika FMIPA Universitas Jember.
4. Himatika “Geokompstat” dan UKMS TITIK yang telah memberikan banyak pengalaman, pelajaran dan kenangan-kenangan yang akan saya rindukan saat berada dikampung halaman.
5. Seluruh keluarga besar saya yang telah memberikan semangat serta doa.
6. Teman-teman Misdirection, sahabat serta semua pihak yang telah membantu dan memberikan semangat.

Penulis menyadari bahwa penulisan ini masih jauh dari kata sempurna. Oleh karena itu, penulis mengharap kritik dan saran demi kesempurnaan penelitian selanjutnya. Semoga bantuan, bimbingan dan motivasi beliau dicatat sebagai amal baik oleh Allah SWT. Semoga tugas akhir ini dapat bermanfaat bagi kita semua.

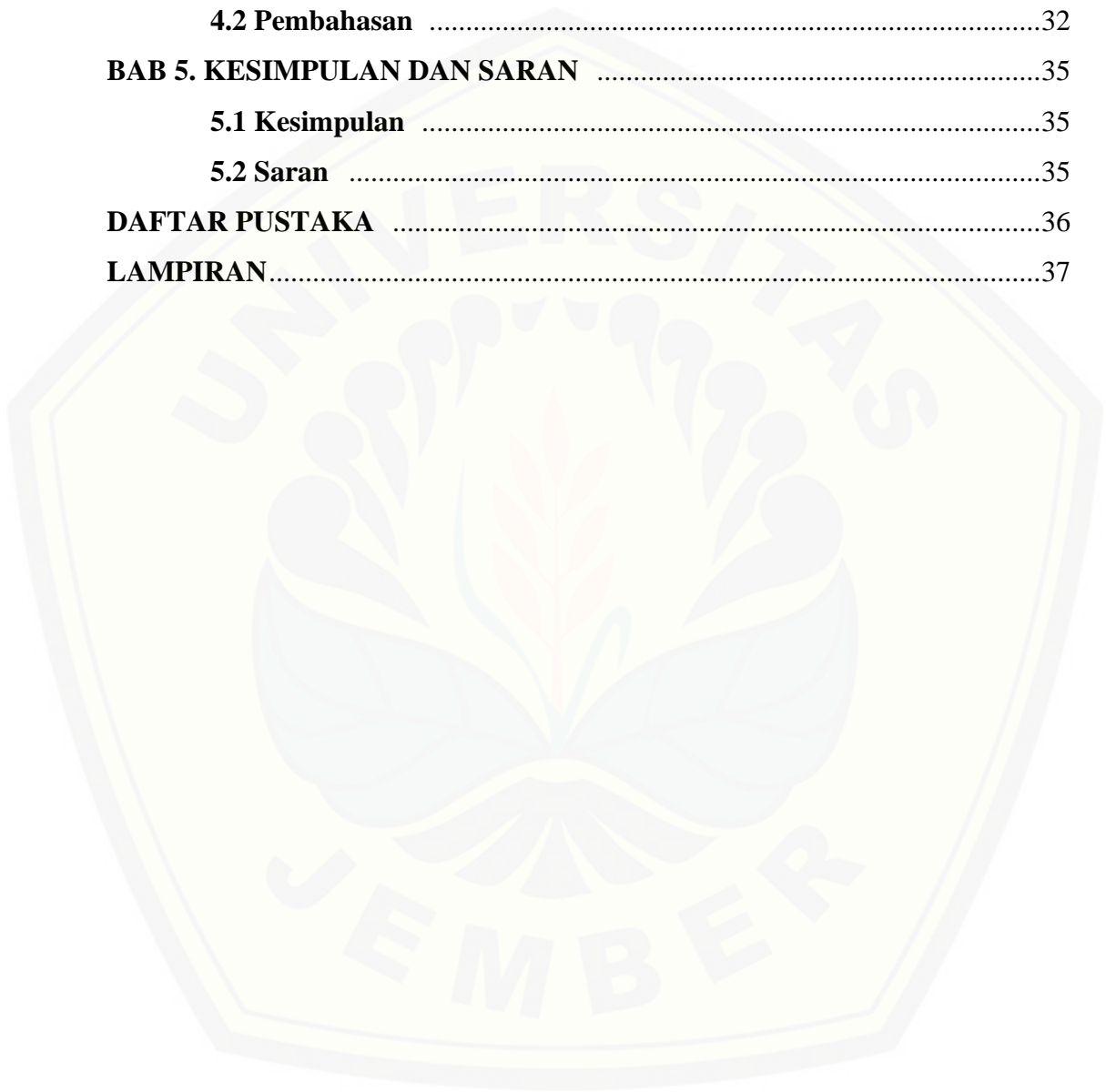
Jember, Agustus 2020

Penulis

DAFTAR ISI

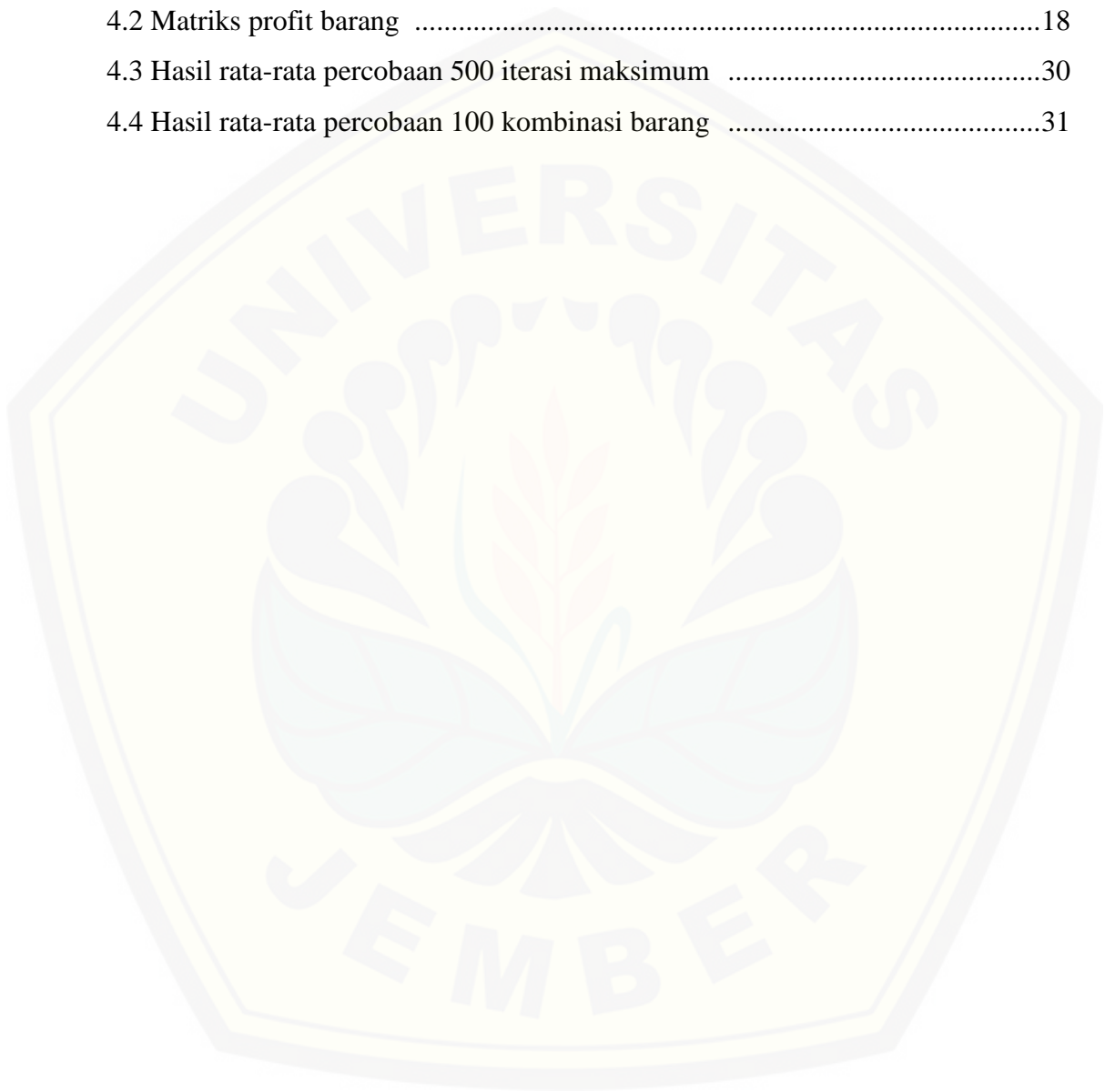
	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTO	iii
HALAMAN PERNYATAAN	iv
HALAMAN PEMBIMBINGAN	v
HALAMAN PENGESAHAN	vi
RINGKASAN	vii
PRAKATA	ix
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xiv
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	4
BAB 2. TINJAUAN PUSTAKA	5
2.1 Quadratic Knapsack 0-1 Problem	5
2.2 Whale Optimization Algorithm (WOA)	7
2.2.1 Pencarian Mangsa (Eksplorasi)	8
2.2.2 Pengepungan Mangsa	9
2.2.3 Metode Penyerangan	9
2.2.4 Pseudo Code Algoritma WOA	10
BAB 3. METODE PENELITIAN	12
3.1 Data Penelitian	12
3.2 Langkah Penelitian	12
BAB 4. HASIL DAN PEMBAHASAN	17

4.1 Hasil Penelitian	17
4.1.1 Langkah Perhitungan	17
4.1.2 Hasil Program	27
4.1.3 Hasil Percobaan	29
4.2 Pembahasan	32
BAB 5. KESIMPULAN DAN SARAN	35
5.1 Kesimpulan	35
5.2 Saran	35
DAFTAR PUSTAKA	36
LAMPIRAN	37



DAFTAR TABEL

	Halaman
4.1 Data simulasi	17
4.2 Matriks profit barang	18
4.3 Hasil rata-rata percobaan 500 iterasi maksimum	30
4.4 Hasil rata-rata percobaan 100 kombinasi barang	31



DAFTAR GAMBAR

	Halaman
2.1 Paus bungkuk mengeluarkan gelembung berbentuk spiral	7
2.2 Fungsi spiral	10
3.1 <i>Flowchart Whale Optimization Algorithm (WOA)</i>	13
3.2 Skema metode penelitian	16
4.1 Tampilan utama program	27
4.2 Tampilan program saat proses berlangsung	29
4.3 Tampilan hasil program setelah proses selesai	29
4.4 Tampilan hasil program 1000 kombinasi barang	31
4.5 Tampilan hasil program dengan $N = 100$ dan iterasi maksimum = 5.000 ...	32

DAFTAR LAMPIRAN

	Halaman
A.1 Data Barang SRC SUGIARTO	37
A.2 Data Kendaraan SRC SUGIARTO	38
A.3 Matriks Profit 60 Barang	39
B.1 Menentukan Vektor Koefisien \vec{A} dan \vec{C}	49
B.2 Menentukan Solusi Baru pada Solusi 1 (x_1)	49
B.3 Menentukan Solusi Baru pada Solusi 2 (x_2)	50
B.4 Menentukan Solusi Baru pada Solusi 3 (x_3)	51
B.5 Transformasi Solusi pada <i>knapsack</i> agar berada pada interval [0,1]	52
C.1 <i>Script</i> Utama	53
C.2 <i>Script</i> Menghitung Fungsi Tujuan <i>Quadratic Knapsack Problem</i>	55
D.1 Hasil Percobaan 10 kombinasi barang dan 500 iterasi	56
D.2 Hasil Percobaan 20 kombinasi barang dan 500 iterasi	56
D.3 Hasil Percobaan 50 kombinasi barang dan 500 iterasi	57
D.4 Hasil Percobaan 100 kombinasi barang dan 500 iterasi	57
D.5 Hasil Percobaan 250 kombinasi barang dan 500 iterasi	58
D.6 Hasil Percobaan 500 kombinasi barang dan 500 iterasi	58
D.7 Hasil Percobaan 1.000 kombinasi barang dan 500 iterasi	59
D.8 Hasil Percobaan 20 kombinasi barang dan 1.000 iterasi	59
D.9 Hasil Percobaan 20 kombinasi barang dan 2.000 iterasi	60
D.10 Hasil Percobaan 20 kombinasi barang dan 5.000 iterasi	60

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Permasalahan optimasi sering dihadapi dalam kehidupan sehari-hari. Misalnya dalam hal berbisnis, setiap orang maupun sebuah perusahaan selalu menginginkan keuntungan dan efisiensi pengiriman dalam pengemasan barang pada media penyimpanan. Kesulitan yang sering dialami dalam melakukan pemilihan barang yang akan dikirim adalah karena terbatasnya media penyimpanan untuk mengangkut semua barang meskipun media penyimpanan yang disediakan sudah lebih dari satu.

Permasalahan optimasi tersebut merupakan permasalahan *knapsack*. *Knapsack* dapat diilustrasikan sebagai karung, atau media penyimpanan. Menurut Supriadi (2016) permasalahan *knapsack* merupakan permasalahan tentang bagaimana cara memilih barang dari sekian banyak pilihan di mana setiap barang tersebut mempunyai berat dan profit masing-masing dengan memperhatikan juga kapasitas dari media penyimpanan, sehingga dari pemilihan barang tersebut didapatkan profit yang maksimum. Pemilihan barang mana saja yang akan dipilih dan disimpan dalam media penyimpanan merupakan faktor penting untuk mengoptimalkan keuntungan. Tujuan *Knapsack problem* adalah untuk mendapatkan keuntungan yang maksimum dari pemilihan barang tanpa melebihi kapasitas daya tampung media penyimpanan tersebut.

Suyanto (2010) menjelaskan bahwa *Knapsack Problem* terbagi menjadi tiga jenis, yaitu *Knapsack 0-1 Problem*, *Bounded Knapsack Problem* dan *Unbounded Knapsack Problem*. Pembagian tersebut berdasarkan pola penyimpanan barang dengan nilai dan bobot yang bervariasi. Pada penelitian ini penulis ingin membahas tentang *Knapsack 0-1 Problem* dimana barang akan dimasukkan secara utuh (1) atau tidak sama sekali (0) ke dalam media penyimpanan.

Knapsack Problem banyak terjadi pada perusahaan produksi maupun distribusi barang. Kendala yang banyak dihadapi adalah biaya yang dikeluarkan tidak melebihi modal yang dimiliki perusahaan. Selain itu, perusahaan juga harus memperhatikan kapasitas ruang dan berat dari media penyimpanan agar barang

yang disimpan tidak melebihi kapasitas media penyimpanan. Maka dari itu, barang harus dikombinasikan sedemikian rupa agar tidak melebihi batasan-batasan yang ada dan diperoleh keuntungan yang optimal. Agar dapat memaksimalkan keuntungan, perusahaan membutuhkan strategi dalam pengiriman barang. Pemberian variasi harga juga bisa menarik perhatian konsumen agar barang yang dijual lebih maksimal serta keuntungan yang didapat lebih besar. Seperti memberikan potongan harga untuk pembelian lebih dari satu barang.

Berdasarkan penjelasan *Knapsack Problem* tersebut, muncul variasi permasalahan *Knapsack* yaitu *Quadratic Knapsack Problem*. *Quadratic Knapsack Problem* adalah permasalahan *Knapsack* dimana fungsi tujuan dimaksimalkan dalam bentuk kuadratik untuk kendala kapasitas biner dan linier. Artinya, akan ada tambahan laba apabila dua item dipilih dari seperangkat item dengan berat, nilai dan laba masing-masing.

Banyak algoritma metaheuristik yang dapat digunakan untuk menyelesaikan *Quadratic Knapsack 0-1 Problem*. Azad *et al.* (2014) melakukan penelitian yang berjudul *A Simplified Binary Artificial Fish Swarm Algorithm (AFSA) for 0-1 Quadratic Knapsack Problems*. Algoritma AFSA ini didasarkan pada pergerakan masal gerombolan ikan dalam bermigrasi, mencari makanan, atau berkelahi dengan musuh. Penelitian ini bertujuan untuk memecahkan *Quadratic Knapsack 0-1 Problem* serta masalah fungsi objektif linear guna meningkatkan kualitas solusi. Namun penelitian ini masih memerlukan pengembangan lebih lanjut untuk mengurangi upaya komputasi dalam hal jumlah iterasi dan waktu eksekusi.

Patvardhan *et al.* (2015) juga meneliti tentang *Quadratic Knapsack 0-1 Problem*, yaitu menyelesaikan *Quadratic Knapsack 0-1 Problem* dengan *Quantum Inspired Evolutionary Algorithm* yang kompetitif. Algoritma QIEA ini terinspirasi oleh prinsip-prinsip komputasi *quantum*. Penelitian ini mendapatkan hasil yang baik yaitu menciptakan campuran fitur yang efektif dalam setiap upaya implementasi QIEA.

Whale Optimization Algorithm (WOA) merupakan salah satu algoritma yang dapat menyelesaikan *Quadratic Knapsack 0-1 Problem*. Algoritma WOA

ini didasarkan pada perilaku paus bungkuk (*Humpback Whale*) dalam mencari makanan. Mahendra (2019) melakukan penelitian tentang penerapan *Whale Optimization Algorithm* (WOA) namun pada permasalahan *Multi Knapsack 0-1 Problem* dengan *Multiple Constrains*. Mahendra membandingkan algoritma WOA dengan algoritma *Modified Variable Neighborhood Search* (MVNS) dan metode *simplex*. Hasil yang diperoleh dari penelitian tersebut adalah algoritma WOA lebih efisien daripada algoritma MVNS dan metode *simplex* berdasarkan waktu komputasi namun fungsi tujuan yang dihasilkan berbentuk linier.

Rasio penggunaan metode *Quadratic Knapsack 0-1 Problem* dalam penelitian terbilang cukup rendah jika dibandingkan dengan metode lain. Maka dari itu, penulis tertarik untuk meneliti lebih lanjut permasalahan *Quadratic Knapsack 0-1 Problem* dengan menerapkan algoritma *Whale Optimization Algorithm* yang nantinya didapatkan fungsi tujuan berbentuk kuadratik. Selain itu, penulis ingin mengetahui bagaimana efisiensi dan efektifitas algoritma *Whale Optimization Algorithm* pada *Quadratic Knapsack 0-1 Problem*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka permasalahan yang akan dibahas adalah bagaimana solusi terbaik penyelesaian *Quadratic Knapsack 0-1 Problem* menggunakan *Whale Optimization Algorithm* (WOA) sehingga diketahui keefektifan dan keefisienannya yang diukur dengan cara melihat rata-rata waktu komputasi dan iterasi konvergen saat mencapai profit optimal?

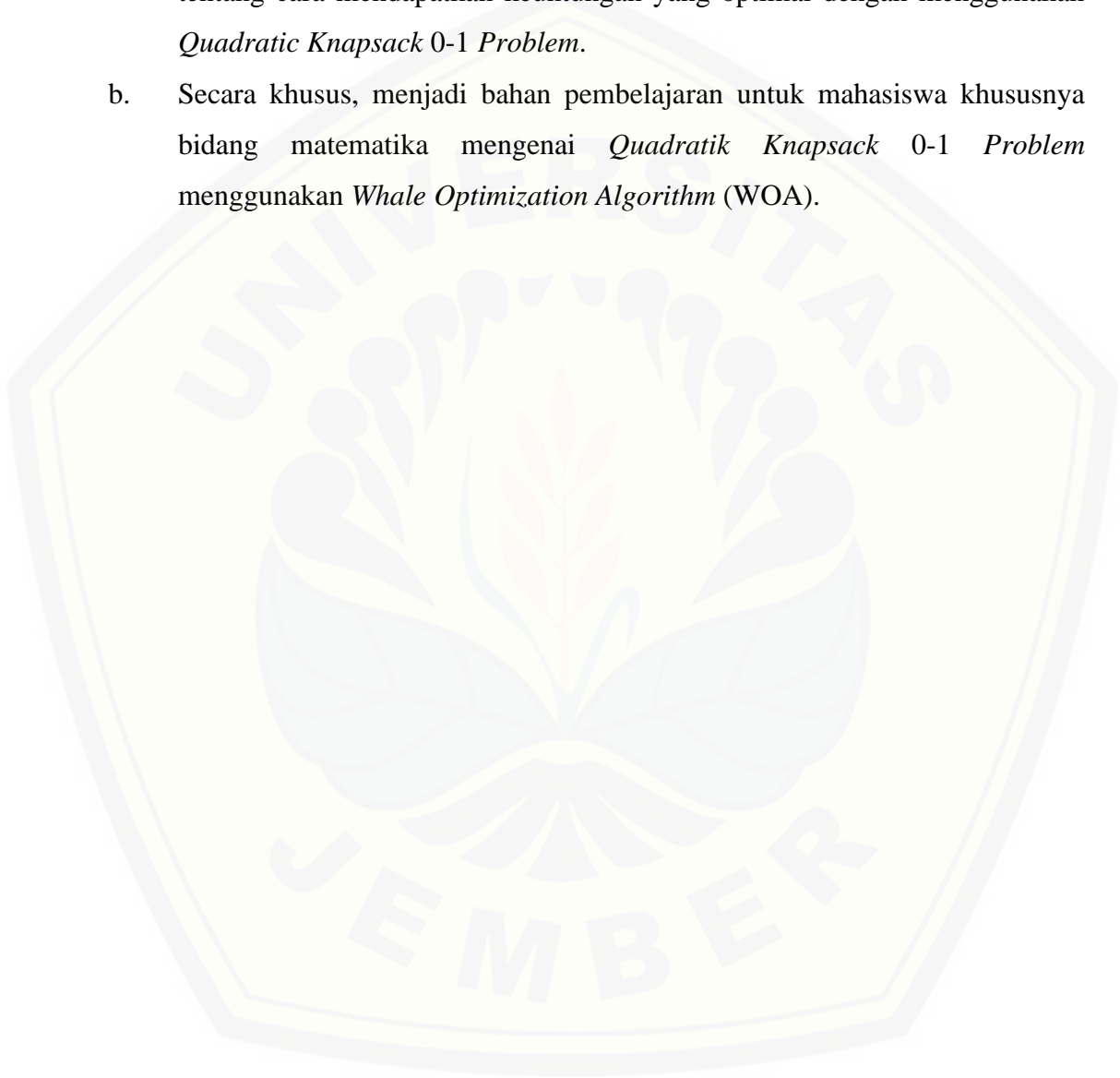
1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah mengetahui solusi terbaik penyelesaian *Quadratic Knapsack 0-1 Problem* menggunakan *Whale Optimization Algorithm* (WOA) sehingga diketahui keefektifan dan keefisienannya.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah:

- a. Secara umum, perusahaan atau pemilik usaha mendapatkan wawasan tentang cara mendapatkan keuntungan yang optimal dengan menggunakan *Quadratic Knapsack 0-1 Problem*.
- b. Secara khusus, menjadi bahan pembelajaran untuk mahasiswa khususnya bidang matematika mengenai *Quadratik Knapsack 0-1 Problem* menggunakan *Whale Optimization Algorithm (WOA)*.



BAB 2. TINJAUAN PUSTAKA

2.1 *Quadratic Knapsack 0-1 Problem*

Knapsack Problem merupakan masalah dimana orang dihadapkan pada persoalan optimasi pada pemilihan benda yang dapat dimasukkan ke dalam sebuah wadah yang memiliki keterbatasan ruang atau daya tampung. Dengan adanya optimasi dalam pemilihan benda yang akan dimasukkan ke dalam wadah tersebut diharapkan dapat menghasilkan keuntungan yang maksimum (Wicaksono dan Andy, 2007).

Dalam dunia nyata permasalahan *Knapsack* sering digunakan dalam bidang (jasa) pengangkutan barang. Dari pengangkutan barang tersebut, diharapkan diperoleh suatu keuntungan yang maksimal untuk mengangkut barang yang ada dengan memperhatikan kapasitas yang ada, sehingga diharapkan terdapat penyelesaian yang bisa memaksimalkan suatu keuntungan dalam pengangkutan barang dengan memperhatikan kendala yang ada. Menurut Dimiyati dan Dimiyati (2004), *Knapsack* merupakan permasalahan optimasi kombinatorial, dimana harus mencari solusi terbaik dari banyak kemungkinan yang dihasilkan.

Jenis *Knapsack Problem* berdasarkan permasalahannya antara lain *Knapsack Problem 0-1*, *Bounded Knapsack Problem* dan *Unbounded Knapsack Problem*. *Knapsack Problem 0-1* adalah permasalahan *Knapsack* dimana barang yang dimasukkan ke dalam media penyimpanan harus semua (1) atau tidak sama sekali (0). *Bounded Knapsack Problem* adalah permasalahan *Knapsack* dimana setiap barang tersedia sebanyak n unit dan jumlah barang yang dimasukkan ke dalam media penyimpanan terbatas. *Unbounded Knapsack Problem* adalah masalah *Knapsack* dimana setiap barang tersedia lebih dari satu unit dan jumlah barang yang dimasukkan ke dalam media penyimpanan tidak terbatas (Pisinger, 1995).

Muncul variasi dari beberapa permasalahan dasar pada *Knapsack* yang dapat diselesaikan menggunakan program pada komputer dengan jumlah data yang besar. Variasi utama terjadi dengan mengubah beberapa parameter seperti jumlah kendala, jumlah tujuan atau jumlah media penyimpanan. Variasi

permasalahan *Knapsack* berdasarkan parameternya menurut Kellerer *et al.* (2004) antara lain:

a. *Multi Objective Knapsack Problem*

Permasalahan *Knapsack* yang memiliki fungsi tujuan lebih dari satu.

b. *Multi Dimensional* atau *Multiple Constraints Knapsack Problem*

Permasalahan *Knapsack* yang memiliki kendala lebih dari satu.

c. *Multi Knapsack Problem*

Permasalahan *Knapsack* yang memiliki media penyimpanan lebih dari satu.

d. *Quadratic Knapsack Problem*

Permasalahan *Knapsack* yang tujuannya memaksimalkan fungsi tujuan dalam bentuk kuadratik untuk kendala kapasitas biner dan linier.

Quadratic Knapsack Problem merupakan salah satu variasi permasalahan *knapsack* berdasarkan parameternya. *Knapsack* jenis ini diperkenalkan oleh Gallo *et al.* (1980). Secara umum dapat didefinisikan sebagai berikut: asumsikan bahwa diberikan m item yang setiap item ke- j memiliki bobot bilangan bulat positif w_j . Selain itu, diberikan matriks bilangan bulat *nonnegative* $P = \{p_{ij}\}$ berukuran $m \times m$, dengan p_{jj} adalah laba yang diperoleh jika item ke- j dipilih dan p_{ij} adalah laba yang diperoleh jika kedua item i dan j dipilih untuk $i < j$. Tujuan *Quadratic Knapsack 0-1 Problem* adalah untuk memilih subset item yang bobot keseluruhan tidak melebihi kapasitas *knapsack* (C) yang diberikan, sedemikian sehingga memaksimalkan keuntungan keseluruhan. *Quadratic Knapsack Problem* dapat dirumuskan sebagai berikut:

Fungsi tujuan:

$$\text{Max } Z_n = \sum_{i=1}^m \sum_{j=i}^m p_{ij} x_i x_j \quad (2.1)$$

Kendala:

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

$$x_j \in \{0,1\} (j = 1, 2, \dots, m) \quad (2.3)$$

dengan,

Z : nilai optimal dari fungsi tujuan

- n : banyaknya paus
 m : banyaknya barang
 i, j : indeks barang
 p_{ij} : keuntungan tiap barang ke- i dan ke- j
 x_i, x_j : variabel keputusan ((1) jika dipilih dan (0) jika tidak dipilih)
 w_j : berat barang ke- j
 C : kapasitas berat media penyimpanan

Fungsi tujuan (2.1) adalah nilai maksimum total keuntungan dari barang yang dimasukkan pada media penyimpanan. Kendala (2.2) merumuskan total bobot barang yang dimasukkan pada *knapsack* tidak melebihi kapasitas.

2.2 Whale Optimization Algorithm (WOA)

Whale Optimization Algorithm (WOA) dikenalkan oleh Mirjalili dan Lewis (2016). Algoritma WOA ini didasarkan pada perilaku Paus Bungkuk (*Humpback Whale*) dalam mencari makanan. Menurut Goldbogen *et al.* (2013) hal menarik dari paus bungkuk dalam mencari makanan adalah cara berburu yang istimewa dengan menggunakan metode perangkap gelembung.

Kebanyakan paus bungkuk memangsa sekelompok ikan kecil di dekat permukaan air. Paus bungkuk dapat menyelam sejauh 12 meter ke arah bawah kemudian membentuk gelembung spiral di sekitar mangsa dan kemudian naik menuju permukaan. Dijelaskan bahwa peristiwa pemangsaan dengan perangkap gelembung merupakan perilaku unik yang hanya dilakukan paus bungkuk.



Gambar 2.1 Paus bungkuk mengeluarkan gelembung berbentuk spiral

(Sumber: Mirjalili dan Lewis, 2016: 53)

Peristiwa pemangsaan mangsa pada paus bungkuk dengan mengeluarkan gelembung berbentuk spiral secara matematis dilakukan untuk melakukan optimasi.

2.2.1 Pencarian Mangsa (Eksplorasi)

Pendekatan yang sama berdasarkan dari variasi nilai vektor \vec{A} dapat dimanfaatkan untuk mencari mangsa (eksplorasi). Faktanya paus bungkuk mencari mangsa secara acak berdasarkan posisi satu sama lain. Maka dari itu, untuk memaksa nilai yang dicari pindah jauh dari paus yang dituju menggunakan \vec{A} dengan nilai acak yang lebih besar dari 1 atau lebih kecil dari -1. Posisi baru dicari dari nilai pada fase eksplorasi berdasarkan pemilihan acak dari nilai yang dicari dibanding nilai yang telah ditemukan sejauh ini. Mekanisme ini dan mekanisme $|\vec{A}| > 1$ menekan eksplorasi dan membiarkan algoritma WOA melakukan pencarian global. Model matematisnya dinyatakan dengan Persamaan (2.4) dan Persamaan (2.5).

$$\vec{D} = |\vec{C} \cdot \vec{X}_{rand} - \vec{X}| \quad (2.4)$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \vec{D} \quad (2.5)$$

dengan t merupakan iterasi perulangan, \vec{X}_{rand} merupakan posisi vektor acak (paus acak) yang dipilih pada populasi tertentu, \vec{D} merupakan jarak dari solusi terbaik yang diperoleh ke vektor posisi paus, \vec{A} dan \vec{C} merupakan vektor koefisien. Vektor \vec{A} dan \vec{C} dapat dihitung menggunakan Persamaan (2.6) dan Persamaan (2.7).

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (2.6)$$

$$\vec{C} = 2 \cdot \vec{r} \quad (2.7)$$

dengan \vec{a} secara linier menurun dari 2 ke 0 melalui pengulangan dan \vec{r} merupakan vektor acak pada [0,1].

2.2.2 Pengepungan Mangsa

Paus bungkuk dapat mengenali lokasi dari mangsanya dan mengepung mereka. Area pencarian pada posisi optimal tidak diketahui skala prioritasnya, algoritma WOA mengasumsikan bahwa kemungkinan solusi terbaik adalah mangsa tujuan atau yang mendekati optimum. Setelah pemilihan kandidat terbaik ditentukan faktor lain akan menyebabkan terjadinya perbaruan posisi menuju titik pencarian terbaik. Perilaku ini diformulasikan dengan Persamaan (2.8) dan Persamaan (2.9).

$$\vec{D} = |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)| \quad (2.8)$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \vec{D} \quad (2.9)$$

dengan \vec{X}^* menunjukkan vektor posisi dari solusi terbaik yang diperoleh.

2.2.3 Metode Penyerangan

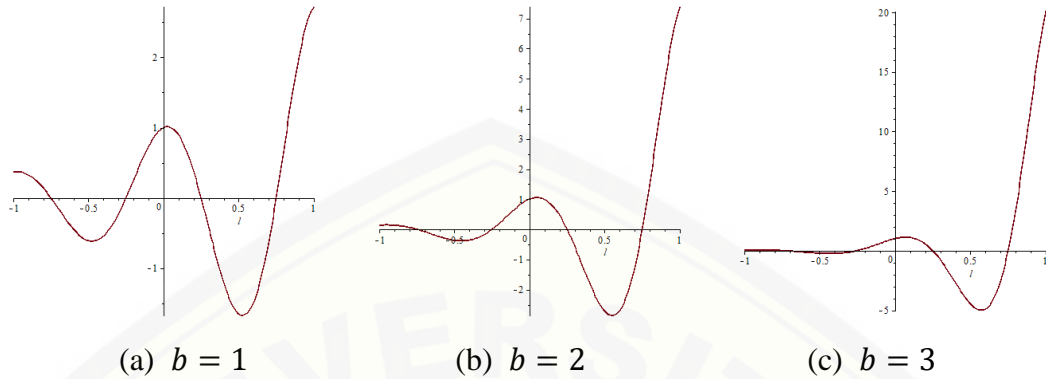
Terdapat dua pendekatan perilaku paus bungkuk dalam menggunakan perangkat gelembung yaitu:

- Mekanisme pengepungan menyusut. Mekanisme pengepungan menyusut diperoleh dari penurunan nilai \vec{a} pada Persamaan (2.6). Fluktuasi rentang \vec{A} juga menurun karena \vec{a} . Dengan kata lain \vec{A} merupakan nilai acak dalam interval $(-a, a)$ dimana penurunan dari dua ke nol selama pengulangan.
- Perbaruan posisi spiral. Pendekatan ini pertama-tama dengan memperhitungkan jarak antara paus yang terletak pada (X, Y) dan mangsa yang terletak pada (\vec{X}^*, \vec{Y}^*) . Persamaan spiral kemudian terbentuk diantara posisi dari paus dan mangsanya untuk meniru pergerakan berbentuk helix dari paus bungkuk seperti pada Persamaan (2.10).

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \quad (2.10)$$

dengan $\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)|$ merupakan jarak antara paus dengan mangsanya pada posisi spiral (solusi terbaik yang didapat), b merupakan nilai konstan dalam menentukan bentuk logaritma spiral, l merupakan

angka acak pada $(-1,1)$ dan merupakan perkalian antar elemen. Ilustrasi posisi spiral paus dapat dilihat pada Gambar 2.2 (a), (b) dan (c).



Gambar 2.2 Fungsi spiral

2.2.4 Pseudo Code Algoritma WOA

Pseudo Code Algoritma WOA adalah sebagai berikut:

Inisialisasi $X_i (i = 1, 2, \dots, n)$

Hitung nilai fitness dari setiap paus

\vec{X}^* = dicari nilai terbaik

while ($t <$ nilai maksimum pengulangan)

for setiap nilai dicari

 Perbarui $\vec{a}, \vec{A}, \vec{C}, l$ dan p

if1 ($p < 0,5$)

if2 ($|A| < 1$)

 Perbarui posisi paus menggunakan Persamaan (2.9)

else if2 ($|A| \geq 1$)

 Pilih paus secara acak (X_{rand})

 Perbarui posisi paus menggunakan Persamaan (2.5)

end if2

else if1 ($p \geq 0,5$)

 Perbarui posisi paus menggunakan Persamaan (2.10)

end if1

end for

Periksa jika paus keluar dari ruang pencarian maka lakukan perbaikan

Hitung kecocokan dari tiap nilai

Perbarui \vec{X}^* jika terdapat solusi terbaik

$t = t + 1$

end while

Kembali ke \vec{X}^*



BAB 3. METODE PENELITIAN

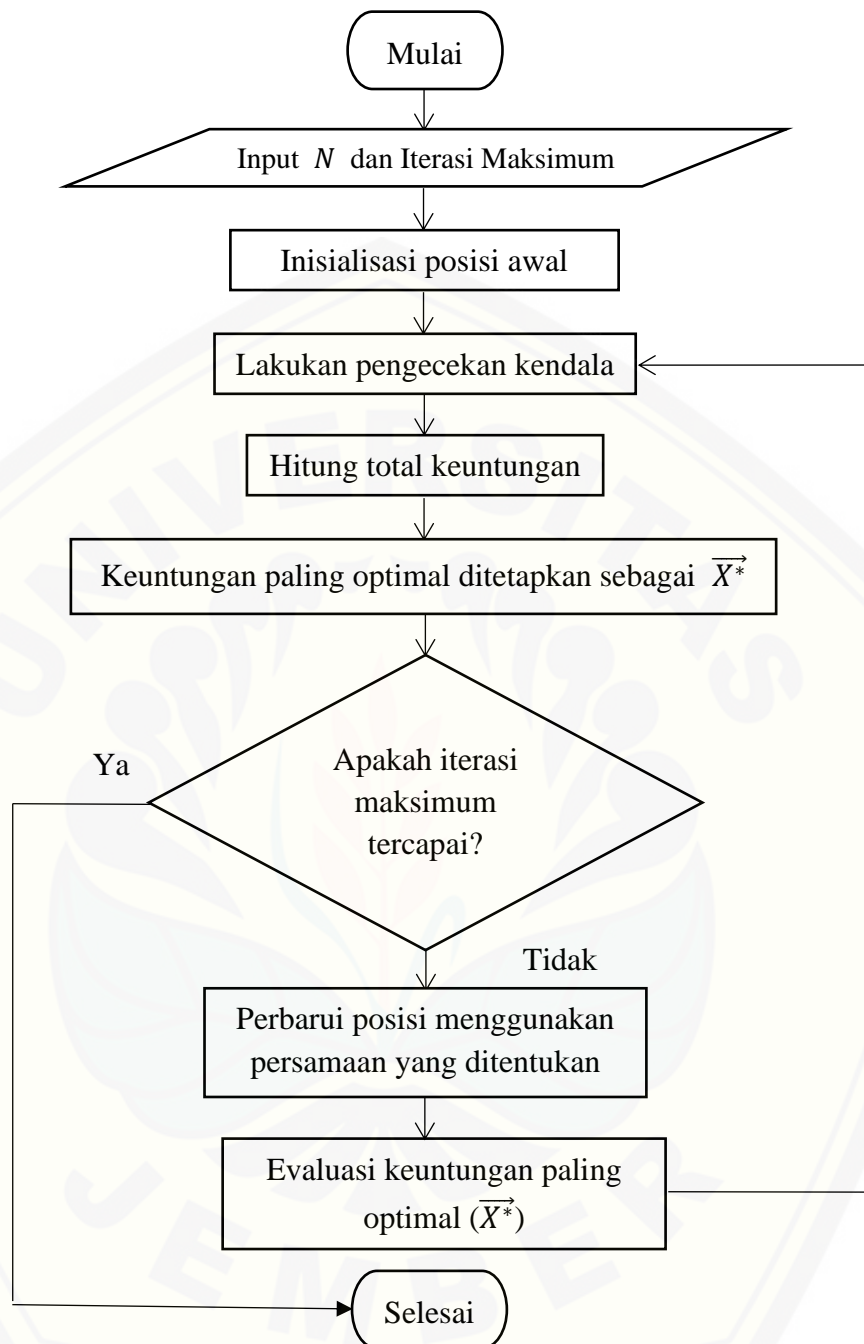
3.1 Data Penelitian

Penelitian ini menggunakan dua data yaitu data barang dan kendaraan yang digunakan untuk menerapkan *Quadratic Knapsack 0-1 Problem*. Data barang dan kendaraan diambil dari SRC SUGIARTO di Kecamatan Rogojampi, Kabupaten Banyuwangi. Toko ini menyediakan barang-barang kebutuhan sehari-hari seperti beras, minyak, sabun, dll. Data yang dibutuhkan untuk percobaan adalah data berat barang, harga beli, harga jual barang serta kapasitas kendaraan (*knapsack*). Kendaraan yang digunakan adalah Motor Tossa Hercules dengan total kapasitas beratnya sebesar 450.000 gram.

3.2 Langkah Penelitian

Penelitian penerapan *Whale Optimization Algorithm* (WOA) pada *Quadratic Knapsack 0-1 Problem* dilakukan dengan langkah-langkah sebagai berikut:

- a. Studi Literatur dilakukan dengan mencari dan mempelajari referensi yang berkaitan dengan *Quadratic Knapsack 0-1 Problem* dan *Whale Optimization Algorithm* (WOA) pada jurnal, buku dan skripsi.
- b. Pengumpulan data diperoleh dari SRC SUGIARTO yang bertempat di Kecamatan Rogojampi Kabupaten Banyuwangi.
- c. Mengidentifikasi data sesuai dengan permasalahan *Quadratic Knapsack 0-1 Problem*. Data yang digunakan meliputi berat barang, harga beli barang, harga jual barang, dan kapasitas berat kendaraan.
- d. Setelah data diidentifikasi, data digunakan untuk mencari fungsi tujuan (profit optimal) menggunakan *Whale Optimization Algorithm*. Berikut merupakan *Flowchart Whale Optimization Algorithm*:



Gambar 3.1 Flowchart Whale Optimization Algorithm (WOA)

e. Pembuatan program *Whale Optimization Algorithm* (WOA). Langkah-langkah penyelesaian *Quadratic Knapsack 0-1 Problem* menggunakan algoritma WOA adalah sebagai berikut:

- 1) Inisialisasi posisi awal paus (kandidat posisi awal). Istilah paus tersebut merujuk pada kombinasi barang yang akan diangkut.

$$X = \begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,j} \\ x_{2,1} & x_{2,2} & \dots & x_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i,1} & x_{i,2} & \dots & x_{i,j} \end{pmatrix}$$

$$x_{i,j} \in \{0,1\}; i = 1,2, \dots, n; j = 1,2, \dots, m$$

dengan,

i : indeks paus (kombinasi barang)

j : indeks barang

Ubah kombinasi barang (X) ke dalam bentuk solusi *knapsack* 0-1 (Y)

$$Y = \begin{pmatrix} y_{1,1} & y_{1,2} & \dots & y_{1,j} \\ y_{2,1} & y_{2,2} & \dots & y_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ y_{i,1} & y_{i,2} & \dots & y_{i,j} \end{pmatrix}$$

$y_{i,j} = \text{round}(x_{i,j})$, misal $y_{1,1} = 1, y_{1,2} = 0$ berarti barang 1 diangkut pada solusi 1 dan barang 2 tidak diangkut pada solusi 1.

2) Pengecekan kendala

Setiap solusi pada masing-masing *knapsack* harus memenuhi kendala berikut:

$$\sum_{j=1}^m w_j y_j \leq C \quad (y_j \in \{0,1\}; j = 1,2, \dots, m)$$

dengan,

w_j : berat barang ke- j

C : kapasitas berat media penyimpanan

Jika solusi tidak memenuhi kendala, pilih salah satu barang yang akan ditinggalkan atau tidak diangkut (diganti nilai 0).

$$x'_{i,j} = x_{i,j} - 0,5$$

3) Hitung keuntungan

Nilai keuntungan dari setiap solusi dihitung berdasarkan total profitnya.

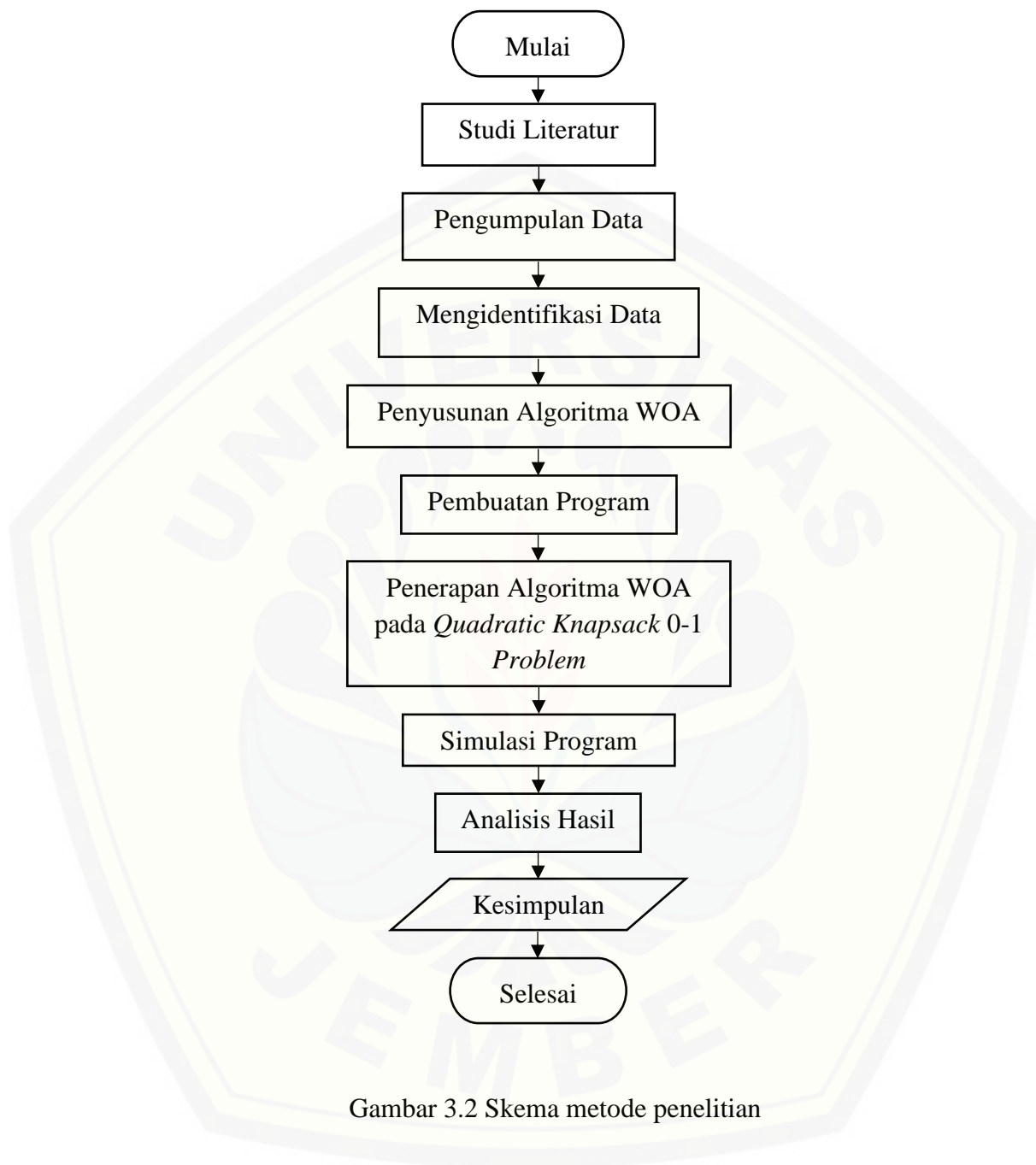
$$\text{Max } Z_n = \sum_{i=1}^m \sum_{j=i}^m p_{ij} x_i x_j \quad (i = 1,2, \dots, m)$$

dengan,

p_{ij} : keuntungan tiap barang ke- i dan ke- j

4) Tetapkan keuntungan yang paling maksimum sebagai kombinasi barang terbaik ($\overline{X^*}$).

- 5) Perbarui nilai $\vec{a}, \vec{A}, \vec{C}, l$ dan p .
 - 6) Lakukan perpindahan kombinasi barang sesuai dengan aturan pada *pseudo-code* algoritma WOA.
 - 7) Pastikan kombinasi barang berada pada ruang pencarian ($x_{i,j} \in \{0,1\}$).
 - 8) Ubah posisi kombinasi barang (X) ke dalam bentuk solusi (Y) kemudian lakukan pengecekan kendala dan hitung keuntungan.
 - 9) Perbarui kombinasi barang terbaik (\vec{X}^*).
 - 10) Ulangi langkah 5) - 8) hingga iterasi maksimum tercapai.
- f. Menerapkan program yang telah dibuat untuk menyelesaikan permasalahan *Quadratic Knapsack 0-1 Problem* menggunakan data yang dikumpulkan dan diidentifikasi.
- g. Mensimulasikan program dengan $N = 10, 20, 50, 100, 250, 500, 1.000$ dan banyak iterasi mulai 500, 1.000, 2.000 dan 5.000.
- h. Membuat kesimpulan berdasarkan hasil simulasi menggunakan program. Sehingga mendapatkan kesimpulan mengenai keefektifan dan keefisienan algoritma WOA pada *Quadratic Knapsack 0-1 Problem*.



Gambar 3.2 Skema metode penelitian

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan, kesimpulan secara umum yang dapat diperoleh adalah algoritma *Whale Optimization Algorithm* efektif dan efisien untuk menyelesaikan permasalahan *Quadratic Knapsack 0-1 Problem*. Dikatakan efektif karena dari setiap percobaan, algoritma selalu konvergen atau menuju profit maksimum. Selain itu, dengan parameter yang tepat, algoritma mampu mencapai hasil yang optimal. Dikatakan efisien karena untuk mendapatkan profit optimal tidak membutuhkan waktu dan iterasi yang lebih banyak. Parameter kombinasi barang dan iterasi maksimum sangat berpengaruh terhadap nilai total profit dan *running time*-nya. Namun, penambahan parameter kombinasi barang lebih cepat mencapai optimal daripada parameter iterasi maksimum.

Sedangkan secara khusus dari hasil dan pembahasan pada penelitian ini adalah profit optimal yang diperoleh dari penyelesaian *Quadratic Knapsack 0-1 Problem* menggunakan algoritma WOA dengan 60 jenis barang adalah sebesar Rp. 2.599.149,00 dengan total berat barang sebesar 449.310 gram. Profit optimal diperoleh pada parameter $N = 1.000$ dengan 500 iterasi maksimum dan parameter $N = 100$ dengan 5.000 iterasi maksimum.

5.2 Saran

Rasio penyelesaian permasalahan *knapsack* menggunakan kasus *Quadratic Knapsack Problem* dapat dikatakan cukup rendah. Masih terbuka bagi peneliti selanjutnya untuk menerapkan kasus tersebut. Penelitian selanjutnya diharapkan dapat menyelesaikan kasus *Quadratic Knapsack Problem* menggunakan algoritma lain yang lebih efisien. Selain itu, penelitian selanjutnya diharapkan dapat menambah fungsi kendala dan jumlah *knapsack* sehingga permasalahannya lebih bervariasi.

DAFTAR PUSTAKA

- Azad, A. K., A. M. A. C. Rocha dan M. G. P. Fernandes. 2014. A simplified binary artificial fish swarm algorithm for 0-1 quadratic knapsack problems. *Journal of Computational and Applied Mathematics*, 259: 897-904.
- Dimiyati, T. T. dan A. Dimiyati. 2004. *Operations Research Model-model Pengambilan Keputusan*. Bandung: Sinar Baru Algensindo.
- Gallo, G., P. Hammer dan B. Simeone. 1980. Quadratic knapsack problems, *Math. Programming Study*, 12: 132-149.
- Goldbogen, J. A., A.S. Friedlaender, J. Calambokidis, M.F. Mckenna, M. Simon dan D.P. Nowacek. 2013. Integrative approaches to the study of baleen whale diving behavior, feeding performance, and foraging ecology. *BioScience*, 63: 90-100.
- Kellerer, H., D. Pisinger dan U. Pferschy. 2004. *Knapsack Problem*. Berlin: Springer.
- Mahendra, M. A. 2019. Penerapan Algoritma Whale Optimization Algorithm (WOA) Pada Multi Knapsack 0-1 Problem With Multiple Constraints. *Skripsi*. Jember:Jurusan Matematika FMIPA Universitas Jember.
- Mirjalili, S. dan A. Lewis. 2016. The whale optimization algorithm. *Advances in Engineering Software*, 95: 51-67.
- Patvardhan, C., S. Bansal dan A. Srivastav. 2015. Solving the 0-1 quadratic knapsack problem with a competitive quantum inspired evolutionary algorithm. *Journal of Computational and Applied Mathematics*, 285: 86-99.
- Pisinger, D. 1995. A minimal algorithm for the multiple-choise knapsack problem. *Europen Journal of Operational Research*, 83(2): 349-410.
- Supriadi, D. 2016. Perbandingan penyelesaian knapsack problem secara matematika, kriteria greedy dan algoritma greedy. *Indonesian Journal on Computer and Information Technology*, 1(2): 91.
- Suyanto. 2010. *Algoritma Optimasi : Deterministik atau Probabilistik*. Yogyakarta: Graha Ilmu.
- Wicaksono, Prasetyo Andy. 2007. *Makalah IF2251 Strategi Algoritmik: Eksplorasi Algoritma Brute Force, Greedy dan Pemrograman Dinamis pads Penyelesaian Masalah 0/1 Knapsack*. Bandung: STEI, Institut Teknologi Bandung.

LAMPIRAN

Lampiran A. Data Penelitian

A.1 Data Barang SRC SUGIARTO

No	Data Barang	Berat (g)	Profit (Rp)
1	Beras Bunga Matahari	25.000	37.500
2	Beras Grand Master	25.000	32.500
3	Gula	50.000	30.000
4	Daia 850 g	10.700	18.000
5	Daia 300 g	7.100	16.000
6	Rinso 770 g	9.750	12.000
7	Rinso Cair Refill 800 ml	10.300	9.500
8	Soklin Cair Refill 800 ml	5.000	7.000
9	Soklin Lantai Refil 800 ml	9.600	7.500
10	Soklin Rapika Refill 450 ml	3.500	12.500
11	Wipol Refil 450 ml	5.500	19.200
12	Sunlight 435 ml revil	10.200	28.000
13	Wing ekonomi	3.500	12.000
14	Ciptadent 65 g	14.900	49.000
15	Pepsodent 75 g	29.200	58.000
16	Pepsodent 190 g	14.900	48.000
17	Sinzui	7.000	26.500
18	Giv	1.000	43.000
19	Lifebuoy	7.000	72.000
20	Sampo Sunsilk hijab 170 ml	6.200	86.000
21	Sampo Sunsilk Renceng	3.500	12.800
22	Rexona Roll On 50 ml	3.500	12.000
23	Domestos Nomos Jumbo	11.300	36.000
24	Baygon Bakar Max Jumbo	9.400	53.000
25	Charm Night 29 cm renceng	5.000	14.000
26	Tissu Paseo 250	15.000	51.000
27	Mie Sedap Goreng	3.700	11.000
28	Mie Sedap Soto	3.000	7.000
29	Mie Indomie Goreng	3.500	9.000
30	Sarimi 2 Goreng Kecap	3.500	2.500
31	Mie Burung Dara Pipih	3.400	7.000
32	Kecap ABC Refill 80 ml	5.200	25.000
33	Teh Sariwangi Sachet	3.000	30.000

No	Data Barang	Berat (g)	Profit (Rp)
34	Teh Sariwangi Kotak	1.000	10.000
35	Lavenia 900 g	10.500	20.000
36	Lavenia 1800 g	10.500	20.000
37	Bimoli 1800 ml	7.000	10.000
38	ABC Plus	2.660	9.000
39	ABC Susu	4.300	9.000
40	Kapal Api 6,5 g	1.800	18.000
41	Luwak White Coffee	4.500	32.000
42	Susu Kremer	10.000	69.000
43	Carnation 425 g	21.000	51.000
44	Dancow Sachet Putih	4.800	123.000
45	Extra Joss	3.500	6.000
46	Kuku Bima Anggur	4.000	18.500
47	Ale-ale	4.800	7.000
48	Aqua Gelas	12.000	7.600
49	You C-1000	3.000	39.000
50	Pulpy Orange	6.500	16.000
51	Aqua Botol 600 ml	3.500	18.500
52	Aqua Botol 1500 ml	6.500	12.000
53	Tepung Terigu Segitiga Biru 25 kg	25.000	16.500
54	Tepung Terigu DUS Segitiga Biru	12.500	15.000
55	Garam Daun	10.000	28.000
56	Blue Band Mentega 15 kg	15.000	35.000
57	Permen Kiss	3.500	22.000
58	Energen	3.000	43.000
59	Minyak Kayu Putih Cap Lang	3.000	11.000
60	Minyak Gosok GPU	3.600	28.000

A.2 Data Kendaraan SRC SUGIARTO

No	Data Kendaraan	Kapasitas Berat (g)
1	Tossa	450.000

A.3 Matriks Profit 60 Barang

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16
B1	37500	18500	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B2	18500	32500	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B3	0	0	30000	0	0	0	0	0	0	0	0	0	0	0	0	0
B4	0	0	0	18000	5500	0	0	5500	6000	5000	0	0	5000	8000	0	0
B5	0	0	0	5500	16000	0	0	3500	4000	2500	0	0	2500	5500	0	0
B6	0	0	0	0	0	12000	4000	0	0	0	4300	4000	0	0	6900	6000
B7	0	0	0	0	0	4000	9500	0	0	0	4300	4000	0	0	6900	6000
B8	0	0	0	5500	3500	0	0	7000	4000	2500	0	0	2500	5500	0	0
B9	0	0	0	6000	4000	0	0	4000	7500	3000	0	0	3000	6000	0	0
B10	0	0	0	5000	2500	0	0	2500	3000	12500	0	0	2000	5000	0	0
B11	0	0	0	0	0	4300	4300	0	0	0	19200	4300	0	0	7200	6300
B12	0	0	0	0	0	4000	4000	0	0	0	4300	28000	0	0	6900	6000
B13	0	0	0	5000	2500	0	0	2500	3000	2000	0	0	12000	5000	0	0
B14	0	0	0	8000	5500	0	0	5500	6000	5000	0	0	5000	49000	0	0
B15	0	0	0	0	0	6900	6900	0	0	0	7200	6900	0	0	57600	9400
B16	0	0	0	0	0	6000	6000	0	0	0	6300	6000	0	0	9400	48000
B17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B18	0	0	0	6500	4500	0	0	4500	4500	3500	0	0	3500	6500	0	0
B19	0	0	0	0	0	5500	5500	0	0	0	5800	5500	0	0	8400	7500
B20	0	0	0	0	0	8500	8500	0	0	0	8800	8500	0	0	11400	11000
B21	0	0	0	0	0	5700	5200	0	0	0	5500	5200	0	0	8600	7700
B22	0	0	0	0	0	5000	5000	0	0	0	4800	5000	0	0	7900	7000
B23	0	0	0	0	0	3500	3500	0	0	0	3800	3500	0	0	6900	6000

Digital Repository Universitas Jember

B24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B27	0	0	0	5500	3500	0	0	3500	4000	2500	0	0	2500	5500	0	0
B28	0	0	0	5500	3500	0	0	3500	3500	2500	0	0	2500	5500	0	0
B29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B33	0	0	0	0	0	4500	4000	0	0	0	4300	4000	0	0	7400	6500
B34	0	0	0	0	0	4000	4000	0	0	0	4300	4000	0	0	7400	6500
B35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B47	0	0	0	4000	2000	0	0	2000	2500	1000	0	0	1000	4000	0	0
B48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

B49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B56	0	0	0	0	0	5110	5075	0	0	0	5158	5070	0	0	7734	7270
B57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

B17	B18	B19	B20	B21	B22	B23	B24	B25	B26	B27	B28	B29	B30	B31	B32	B33
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	6500	0	0	0	0	0	0	0	0	5500	5500	0	0	0	0	0
0	4500	0	0	0	0	0	0	0	0	3500	3500	0	0	0	0	0
0	0	5500	8500	5700	5000	3500	0	0	0	0	0	0	0	0	0	4500
0	0	5500	8500	5200	5000	3500	0	0	0	0	0	0	0	0	0	4000
0	4500	0	0	0	0	0	0	0	0	3500	3500	0	0	0	0	0
0	4500	0	0	0	0	0	0	0	0	4000	3500	0	0	0	0	0
0	3500	0	0	0	0	0	0	0	0	2500	2500	0	0	0	0	0
0	0	5800	8800	5500	4800	3800	0	0	0	0	0	0	0	0	0	4300

Digital Repository Universitas Jember

0	0	5500	8500	5200	5000	3500	0	0	0	0	0	0	0	0	0	4000
0	3500	0	0	0	0	0	0	0	0	2500	2500	0	0	0	0	0
0	6500	0	0	0	0	0	0	0	0	5500	5500	0	0	0	0	0
0	0	8400	11400	8600	7900	6900	0	0	0	0	0	0	0	0	0	7400
0	0	7500	11000	7700	7000	6000	0	0	0	0	0	0	0	0	0	6500
26500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	43000	0	0	0	0	0	0	0	0	4500	4000	0	0	0	0	0
0	0	72000	10000	6700	6500	5000	0	0	0	0	0	0	0	0	0	5500
0	0	10000	86000	10200	9500	8000	0	0	0	0	0	0	0	0	0	9000
0	0	6700	10200	12800	6200	5200	0	0	0	0	0	0	0	0	0	5700
0	0	6500	9500	6200	12000	4500	0	0	0	0	0	0	0	0	0	5000
0	0	5000	8000	5200	4500	36000	0	0	0	0	0	0	0	0	0	4000
0	0	0	0	0	0	0	53000	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	14000	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	51000	0	0	0	0	0	0	0
0	4500	0	0	0	0	0	0	0	0	11000	3000	0	0	0	0	0
0	4000	0	0	0	0	0	0	0	0	3000	7000	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	9000	4500	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	4500	2500	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	7000	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25000	0
0	0	5500	9000	5700	5000	4000	0	0	0	0	0	0	0	0	0	30000
0	0	5500	9000	5700	5000	4000	0	0	0	0	0	0	0	0	0	4500
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Digital Repository Universitas Jember

0	0	0	0	0	0	0	0	0	0	0	0	6500	6000	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	13500	12500	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	3000	0	0	0	0	0	0	0	0	2000	2000	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	3500	2500	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	4050	3405	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	4170	3525	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	7860	7215	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	5880	5235	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	6550	9650	6502	5950	4690	0	0	0	0	0	0	0	0	0	5290
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

B34	B35	B36	B37	B38	B39	B40	B41	B42	B43	B44	B45	B46	B47	B48	B49	B50
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	4000	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	2000	0	0	0
4000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	2000	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	2500	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1000	0	0	0
4300	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1000	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	4000	0	0	0
7400	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	3000	0	0	0
5500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5700	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Digital Repository Universitas Jember

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	2000	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	2000	0	0	0
0	0	0	6500	0	0	0	0	13500	0	0	0	0	0	3500	0	0
0	0	0	6000	0	0	0	0	12500	0	0	0	0	0	2500	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	20000	8500	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	8500	20000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	10000	0	0	0	0	14500	0	0	0	0	0	4500	0	0
0	0	0	0	9000	4500	3500	0	0	0	0	0	0	0	0	0	0
0	0	0	0	4500	9000	5500	0	0	0	0	0	0	0	0	0	0
0	0	0	0	3500	5500	18000	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	32000	0	0	0	0	0	0	0	0	0
0	0	0	14500	0	0	0	0	69000	0	0	0	0	0	11500	0	0
0	0	0	0	0	0	0	0	0	51000	38000	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	38000	123000	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	6000	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	18500	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	7000	0	0	0
0	0	0	4500	0	0	0	0	11500	0	0	0	0	0	7600	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39000	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16000
0	0	0	5340	0	0	0	0	12210	0	0	0	0	0	2100	0	0
0	0	0	5460	0	0	0	0	12330	0	0	0	0	0	2220	0	0
0	0	0	9150	0	0	0	0	16020	0	0	0	0	0	5910	0	0
0	0	0	7170	0	0	0	0	14050	0	0	0	0	0	3930	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5250	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

B51	B52	B53	B54	B55	B56	B57	B58	B59	B60
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	5110	0	0	0	0
0	0	0	0	0	5075	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	5158	0	0	0	0
0	0	0	0	0	5070	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	7734	0	0	0	0
0	0	0	0	0	7270	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	6550	0	0	0	0
0	0	0	0	0	9650	0	0	0	0
0	0	0	0	0	6502	0	0	0	0
0	0	0	0	0	5950	0	0	0	0
0	0	0	0	0	4690	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
4050	4170	7860	5880	0	0	0	0	0	0
3405	3525	7215	5235	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	5290	0	0	0	0
0	0	0	0	0	5250	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
5340	5460	9150	7170	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
12210	12330	16020	14050	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
2100	2220	5910	3930	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
18500	2760	6450	4470	0	0	0	0	0	0
2760	12000	6570	4590	0	0	0	0	0	0
6450	6570	16500	8280	0	0	0	0	0	0
4470	4590	8280	15000	0	0	0	0	0	0
0	0	0	0	28000	0	0	0	0	0
0	0	0	0	0	35000	0	0	0	0
0	0	0	0	0	0	22000	0	0	0
0	0	0	0	0	0	0	43000	0	0
0	0	0	0	0	0	0	0	11000	0
0	0	0	0	0	0	0	0	0	28000

LAMPIRAN B. Proses Perhitungan Manual Algoritma WOA

B.1 Menentukan Vektor Koefisien \vec{A} dan \vec{C}

Bangkitkan nilai b, p, l, \vec{r} dan \vec{a} secara acak. Diperoleh nilai sebagai berikut:

$$b = 1, \quad l = 0,5126; -0,1987; -0,6102, \quad p = 0,5628; 0,0321; 0,6392, \quad \vec{a} = 2$$

$$\vec{r} = \begin{pmatrix} 0,1244 & 0,7866 & 0,8254 & 0,1213 & 0,0029 & 0,6801 & 0,8711 & 0,0123 \\ 0,5401 & 0,2479 & 0,6711 & 0,9987 & 0,0958 & 0,1111 & 0,0098 & 0,7777 \\ 0,7676 & 0,0301 & 0,0012 & 0,6745 & 0,1234 & 0,5463 & 0,8092 & 0,0104 \end{pmatrix}$$

$$\vec{A} = \begin{pmatrix} -1,5024 & 1,1464 & 1,3016 & -1,5148 & -1,9884 & 0,7204 & 1,4844 & -1,9508 \\ 0,1604 & -1,0084 & 0,6844 & 1,9948 & -1,6168 & -1,5556 & -1,9608 & 1,1108 \\ 1,0704 & -1,8796 & -1,9952 & 0,6980 & -1,5064 & 0,1852 & 0,2368 & -1,9584 \end{pmatrix}$$

$$\vec{C} = \begin{pmatrix} 0,2488 & 1,5732 & 1,6508 & 0,2426 & 0,0058 & 1,3602 & 1,7422 & 0,0246 \\ 1,0802 & 0,4958 & 1,3422 & 1,9974 & 0,1916 & 0,2222 & 0,0196 & 1,5554 \\ 1,5352 & 0,0602 & 0,0024 & 1,349 & 0,2468 & 1,0926 & 1,6184 & 0,0208 \end{pmatrix}$$

Mencari nilai $|\vec{A}^i|$ untuk mencari persamaan yang akan digunakan untuk mencari solusi iterasi kedua.

$$|\vec{A}^1| = \sqrt{(-1,5024)^2 + (1,1464)^2 + (1,3016)^2 + (-1,5148)^2 + (-1,9884)^2 + (0,7204)^2 + (1,4844)^2 + (-1,9508)^2}$$

$$|\vec{A}^1| = 4,2476$$

$$|\vec{A}^2| = \sqrt{(0,1604)^2 + (-1,0084)^2 + (0,6844)^2 + (1,9948)^2 + (-1,6168)^2 + (-1,5556)^2 + (-1,9608)^2 + (1,1108)^2}$$

$$|\vec{A}^2| = 3,95$$

$$|\vec{A}^3| = \sqrt{(1,0704)^2 + (-1,8796)^2 + (-1,9952)^2 + (0,6980)^2 + (-1,5064)^2 + (0,1852)^2 + (0,2368)^2 + (-1,9584)^2}$$

$$|\vec{A}^3| = 3,9168$$

B.2 Menentukan Solusi Baru pada Solusi 1 (x_1)

Solusi 1: $p = 0,5628$ ($p \geq 0,5$) .

Gunakan Persamaan (2.10) untuk mencari kombinasi barang baru pada solusi ke-1 (x_1). Gunakan solusi ke-2 (x_2) sebagai \vec{X}^* yaitu total profit yang paling optimal.

$$x_{1,1} = 0,3125$$

$$\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)| = |0,6012 - 0,3125| = 0,2887$$

$$\begin{aligned} x_{1,1}(t) &= \vec{D}' \cdot e^{bt} \cdot \cos(2\pi t) + \vec{X}^*(t) \\ &= 0,2887 \cdot e^{0,5126t} \cdot \cos(2\pi(0,5126t)) + 0,6012 = 1,0825 \end{aligned}$$

$$x_{1,2} = 0,5012$$

$$\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)| = |0,0177 - 0,5012| = 0,4835$$

$$x_{1,2}(1) = 0,4835 \cdot e^{0,5126} \cdot \cos(2\pi(0,5126)) + 0,0177 = 0,8237$$

$$x_{1,3} = 0,3444$$

$$\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)| = |0,5138 - 0,3444| = 0,1694$$

$$x_{1,3}(1) = 0,1694 \cdot e^{0,5126} \cdot \cos(2\pi(0,5126)) + 0,5138 = 0,7962$$

$$x_{1,4} = 0,0101$$

$$\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)| = |0,2458 - 0,0101| = 0,2357$$

$$x_{1,4}(1) = 0,2357 \cdot e^{0,5126} \cdot \cos(2\pi(0,5126)) + 0,2458 = 0,6387$$

$$x_{1,5} = 0,8125$$

$$\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)| = |0,7998 - 0,8125| = 0,0127$$

$$x_{1,5}(1) = 0,0127 \cdot e^{0,5126} \cdot \cos(2\pi(0,5126)) + 0,7998 = 0,821$$

$$x_{1,6} = 0,7102$$

$$\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)| = |0,3412 - 0,7102| = 0,369$$

$$x_{1,6}(1) = 0,369 \cdot e^{0,5126} \cdot \cos(2\pi(0,5126)) + 0,3412 = 0,9563$$

$$x_{1,7} = 0,0129$$

$$\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)| = |0,1319 - 0,0129| = 0,119$$

$$x_{1,7}(1) = 0,119 \cdot e^{0,5126} \cdot \cos(2\pi(0,5126)) + 0,1319 = 0,3446$$

$$x_{1,8} = 0,1011$$

$$\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)| = |0,7182 - 0,1011| = 0,6171$$

$$x_{1,8}(1) = 0,6171 \cdot e^{0,5126} \cdot \cos(2\pi(0,5126)) + 0,7182 = 1,7469$$

B.3 Menentukan Solusi Baru pada Solusi 2 (x_2)

Solusi 2: $p = 0,0321$ ($p < 0,5$) dan $|\vec{A}^2| = 3,95$ ($|\vec{A}^2| \geq 1$)

Gunakan persamaan (2.5) untuk mencari kombinasi barang yang baru pada solusi

ke-2 (x_2). Ambil solusi ke-3 (x_3) sebagai \vec{X}_{rand} .

$$x_{2,1} = 0,6012$$

$$\vec{D} = |\vec{C} \cdot \vec{X}_{rand} - \vec{X}| = |(1,0802 \cdot 0,1298) - 0,6012| = 0,4610$$

$$x_{2,1}(1) = \vec{X}_{rand} - \vec{A} \vec{D} \\ = 0,1298 - (0,1604 \cdot 0,4610) = 0,0559$$

$$x_{2,2} = 0,0177$$

$$\vec{D} = |(0,4958 \cdot 0,3912) - 0,0177| = 0,1763$$

$$x_{2,2}(1) = 0,3912 - ((-1,0084) \cdot 0,1763) = 0,5690$$

$$x_{2,3} = 0,5138$$

$$\vec{D} = |(1,3422 \cdot 0,7788) - 0,5138| = 0,5315$$

$$x_{2,3}(1) = 0,7788 - (0,6844 \cdot 0,5315) = 0,4150$$

$$x_{2,4} = 0,2458$$

$$\vec{D} = |(1,9974 \cdot 0,4415) - 0,2458| = 0,6361$$

$$x_{2,4}(1) = 0,4415 - (1,9948 \cdot 0,6361) = -0,8274$$

$$x_{2,5} = 0,7998$$

$$\vec{D} = |(0,1916 \cdot 0,9011) - 0,7998| = 0,6272$$

$$x_{2,5}(1) = 0,9011 - ((-1,6168) \cdot 0,6272) = 1,9152$$

$$x_{2,6} = 0,3412$$

$$\vec{D} = |(0,2222 \cdot 0,1111) - 0,3412| = 0,3165$$

$$x_{2,6}(1) = 0,1111 - ((-1,5556) \cdot 0,3165) = 0,6035$$

$$x_{2,7} = 0,1319$$

$$\vec{D} = |(0,0196 \cdot 0,9998) - 0,1319| = 0,6165$$

$$x_{2,7}(1) = 0,9998 - ((-1,9608) \cdot 0,6165) = 2,2086$$

$$x_{2,8} = 0,7182$$

$$\vec{D} = |(1,5554 \cdot 0,6379) - 0,7182| = 0,274$$

$$x_{2,8}(1) = 0,6379 - (1,1108 \cdot 0,274) = 0,3335$$

B.4 Menentukan Solusi Baru pada Solusi 3 (x_3)

Solusi 3: $p = 0,6392$ ($p \geq 0,5$)

Gunakan Persamaan (2.10) untuk mencari kombinasi barang yang baru pada solusi ke-3 (x_3). Gunakan solusi ke-2 (x_2) sebagai \vec{X}^* yaitu total keuntungan yang paling optimal.

$$x_{3,1} = 0,1298$$

$$\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)| = |0,6012 - 0,1298| = 0,4714$$

$$\begin{aligned}x_{3,1}(t) &= \vec{D}^i \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \\ &= 0,4714 \cdot e^{(-0,6102)} \cdot \cos(2\pi(-0,6102)) + 0,6012 = 0,8567\end{aligned}$$

$$\begin{aligned}x_{3,2} &= 0,3912 \\ \vec{D}^i &= |\vec{X}^*(t) - \vec{X}(t)| = |0,0177 - 0,3912| = 0,3735 \\ x_{3,2}(1) &= 0,3735 \cdot e^{(-0,6102)} \cdot \cos(2\pi(-0,6102)) + 0,0177 = 0,2202\end{aligned}$$

$$\begin{aligned}x_{3,3} &= 0,7788 \\ \vec{D}^i &= |\vec{X}^*(t) - \vec{X}(t)| = |0,5138 - 0,7788| = 0,265 \\ x_{3,3}(1) &= 0,265 \cdot e^{(-0,6102)} \cdot \cos(2\pi(-0,6102)) + 0,5138 = 0,6574\end{aligned}$$

$$\begin{aligned}x_{3,4} &= 0,4415 \\ \vec{D}^i &= |\vec{X}^*(t) - \vec{X}(t)| = |0,2458 - 0,4415| = 0,1957 \\ x_{3,4}(1) &= 0,1957 \cdot e^{(-0,6102)} \cdot \cos(2\pi(-0,6102)) + 0,2458 = 0,3519\end{aligned}$$

$$\begin{aligned}x_{3,5} &= 0,9011 \\ \vec{D}^i &= |\vec{X}^*(t) - \vec{X}(t)| = |0,7998 - 0,9011| = 0,1013 \\ x_{3,5}(1) &= 0,1013 \cdot e^{(-0,6102)} \cdot \cos(2\pi(-0,6102)) + 0,7998 = 0,8547\end{aligned}$$

$$\begin{aligned}x_{3,6} &= 0,1111 \\ \vec{D}^i &= |\vec{X}^*(t) - \vec{X}(t)| = |0,3412 - 0,1111| = 0,2301 \\ x_{3,6}(1) &= 0,2301 \cdot e^{(-0,6102)} \cdot \cos(2\pi(-0,6102)) + 0,3412 = 0,4659\end{aligned}$$

$$\begin{aligned}x_{3,7} &= 0,9998 \\ \vec{D}^i &= |\vec{X}^*(t) - \vec{X}(t)| = |0,1319 - 0,9998| = 0,8679 \\ x_{3,7}(1) &= 0,8679 \cdot e^{(-0,6102)} \cdot \cos(2\pi(-0,6102)) + 0,1319 = 0,6023\end{aligned}$$

$$\begin{aligned}x_{3,8} &= 0,6379 \\ \vec{D}^i &= |\vec{X}^*(t) - \vec{X}(t)| = |0,7182 - 0,6379| = 0,0803 \\ x_{3,8}(1) &= 0,0803 \cdot e^{(-0,6102)} \cdot \cos(2\pi(-0,6102)) + 0,7182 = 0,7617\end{aligned}$$

B.5 Transformasi solusi pada *knapsack* agar berada pada interval [0,1].

Solusi 1 = $\frac{x}{x_{max}}$, dimana $x_{max} = 1,7469$

$$x_{1,1} = 1,0825$$

$$x_{1,5} = 0,821$$

$$x_{1,1} = \frac{1,0825}{1,7469} = 0,6197$$

$$x_{1,5} = \frac{0,821}{1,7469} = 0,47$$

$$x_{1,2} = 0,8237$$

$$x_{1,6} = 0,9563$$

$$x_{1,2} = \frac{0,8237}{1,7469} = 0,4715$$

$$x_{1,6} = \frac{0,9563}{1,7469} = 0,5474$$

$$x_{1,3} = 0,7962$$

$$x_{1,7} = 0,3446$$

$$x_{1,3} = \frac{0,7962}{1,7469} = 0,4558$$

$$x_{1,7} = \frac{0,3446}{1,7469} = 0,1973$$

$$x_{1,4} = 0,6387$$

$$x_{1,8} = 1,7469$$

$$x_{1,4} = \frac{0,6387}{1,7469} = 0,3656$$

$$x_{1,8} = \frac{1,7469}{1,7469} = 1$$

Solusi 2 = $\frac{X-X_{min}}{X_{max}-X_{min}}$, dimana $X_{min} = -0,8274$; $X_{max} = 2,2086$

$$x_{2,1} = 0,0559$$

$$x_{2,5} = 1,9563$$

$$x_{2,1} = \frac{0,0559-(-0,8274)}{2,2086-(-0,8274)} = 0,2909$$

$$x_{2,5} = \frac{1,9152-(-0,8274)}{2,2086-(-0,8274)} = 0,9034$$

$$x_{2,2} = 0,569$$

$$x_{2,6} = 0,6035$$

$$x_{2,2} = \frac{0,569-(-0,8274)}{2,2086-(-0,8274)} = 0,46$$

$$x_{2,6} = \frac{0,6035-(-0,8274)}{2,2086-(-0,8274)} = 0,4713$$

$$x_{2,3} = 0,415$$

$$x_{2,7} = 2,2086$$

$$x_{2,3} = \frac{0,415-(-0,8274)}{2,2086-(-0,8274)} = 0,4092$$

$$x_{2,7} = \frac{2,2086-(-0,8274)}{2,2086-(-0,8274)} = 1$$

$$x_{2,4} = -0,8274$$

$$x_{2,8} = 0,3335$$

$$x_{2,4} = \frac{-0,8274-(-0,8274)}{2,2086-(-0,8274)} = 0$$

$$x_{2,7} = \frac{0,3335-(-0,8274)}{2,2086-(-0,8274)} = 0,3824$$

Solusi 3 sudah berada pada interval [0,1] maka tidak perlu dilakukan transformasi.

LAMPIRAN C. Script Program

C.1 Script Utama

```
tic;
%Data
data001 = get(handles.uitable1, 'UserData');
dataprofit = get(handles.uitable2, 'UserData');
databerat = data001{1}';
NamaBarang = data001{2};
n = length(databerat);
Cap = str2num(get(handles.edit4, 'string')); %Kapasitas Knapsack

%Parameter
N = str2num(get(handles.edit1, 'string')); %Populasi
Maxiter = str2num(get(handles.edit2, 'string')); %Maks. Iterasi
a = 2;
b = str2num(get(handles.edit3, 'string'));

%Inisialisasi
```

```

for i = 1:N
    X(i,:) = rand(1,n);
    Y(i,:) = round(X(i,:));
    Berat(i) = sum(Y(i,:).*databerat);
    while Berat(i) > Cap
        X(i,:) = rand(1,n);
        Y(i,:) = round(X(i,:));
        Berat(i) = sum(Y(i,:).*databerat);
    end
    Profit(i) = hitungprofit(Y(i,:), dataprofit);
end

%Find X* (solusi terbaik)
best = find(Profit==max(Profit));
Xbest = X(best(1),:);
Ybest = Y(best(1),:);
Beratbest = Berat(best(1));
Profitbest = Profit(best(1));
%Best-so-far
bsf(1) = Profitbest;
inon = 0;
%Iterasi
for t = 1:Maxiter
    A = 2*a*rand(N,n)-a;
    C = 2*rand(N,n);
    l = rand(N,1)*2-1;
    p = rand(N,1);
    sh = find(p<0.5); %Shrinking
    sp = find(p>=0.5); %Spiral

    %Shrinking
    for i = 1:length(sh)
        if norm(A(sh(i),:))<1
            X(sh(i),:) =
Xbest-A(sh(i),:).*abs(C(sh(i),:).*Xbest-X(sh(i),:));
        else
            Xrand = ceil(rand*N);
            X(sh(i),:) =
X(Xrand,:)-A(sh(i),:).*abs(C(sh(i),:).*X(Xrand,:)-X(sh(i),:));
        end
        if min(X(sh(i),:)) < 0
            X(sh(i),:) =
(X(sh(i),:)-min(X(sh(i),:)))/(max(X(sh(i),:))-min(X(sh(i),:)));
        elseif max(X(sh(i),:)) > 1
            X(sh(i),:) = X(sh(i),:)/max(X(sh(i),:));
        end
        Y(sh(i),:) = round(X(sh(i),:));
        Berat(sh(i)) = sum(Y(sh(i),:).*databerat);
        while Berat(sh(i)) > Cap
            soll = find(Y(sh(i),:)==1);
            ps = ceil(rand*length(soll));
            j = soll(ps);
            X(sh(i),j) = X(sh(i),j)/2;
            Y(sh(i),:) = round(X(sh(i),:));
            Berat(sh(i)) = sum(Y(sh(i),:).*databerat);
        end
    end
end

```

```

        Profit(sh(i)) = hitungprofit(Y(sh(i),:), dataprofit);
    end

    %Spiral
    for i = 1:length(sp)
        X(sp(i),:) =
        abs(Xbest-X(sp(i),:))*exp(b*l(sp(i)))*cos(2*pi*l(sp(i)))+Xbest;
        if min(X(sp(i),:)) < 0
            X(sp(i),:) =
            (X(sp(i),:)-min(X(sp(i),:)))/(max(X(sp(i),:))-min(X(sp(i),:)));
        elseif max(X(sp(i),:)) > 1
            X(sp(i),:) = X(sp(i),:)/max(X(sp(i),:));
        end
        Y(sp(i),:) = round(X(sp(i),:));
        Berat(sp(i)) = sum(Y(sp(i),:).*databerat);
        while Berat(sp(i)) > Cap
            soll = find(Y(sp(i),:)==1);
            ps = ceil(rand*length(soll));
            j = soll(ps);
            X(sp(i),j) = X(sp(i),j)/2;
            Y(sp(i),:) = round(X(sp(i),:));
            Berat(sp(i)) = sum(Y(sp(i),:).*databerat);
        end
        Profit(sp(i)) = hitungprofit(Y(sp(i),:), dataprofit);
    end

    %Update X*
    if max(Profit) > Profitbest
        best = find(Profit==max(Profit));
        Xbest = X(best(1),:);
        Ybest = Y(best(1),:);
        Beratbest = Berat(best(1));
        Profitbest = Profit(best(1));
    end

    a=a-2/Maxiter;

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

```

C.2 Script Menghitung Fungsi Tujuan *Quadratic Knapsack Problem*

```

function [Profit, Matriks] = hitungprofit(solusi, dataprofit)
n = length(solusi);
Matriks = zeros(n);
for i = 1:n
    for j = i:n
        Matriks(i,j) = dataprofit(i,j)*solusi(i)*solusi(j);
    end
end
Profit = sum(sum(Matriks));

```

LAMPIRAN D. Hasil Penelitian Algoritma WOA**D.1 Hasil percobaan dengan 10 kombinasi barang dan 500 iterasi**

N	Total Berat (g)	Total Profit (Rp)	Waktu Komputasi (s)	Iterasi Non-Improvement
1	448.210	2.488.269	27,2853	20
2	445.110	2.494.149	27,2486	451
3	438.310	2.492.649	27,4747	22
4	448.610	2.495.515	27,6121	25
5	440.310	2.536.544	27,4044	284
6	447.610	2.547.194	27,3973	383
7	449.110	2.329.730	27,2262	27
8	446.310	2.481.559	27,1895	78
9	447.410	2.544.149	26,9852	298
10	449.210	2.577.149	27,1405	69

D.2 Hasil percobaan dengan 20 kombinasi barang dan 500 iterasi

N	Total Berat (g)	Total Profit (Rp)	Waktu Komputasi (s)	Iterasi Non-Improvement
1	440.810	2.561.389	28,7108	290
2	449.110	2.563.649	28,5361	340
3	445.710	2.552.149	28,5072	27
4	449.410	2.593.149	28,5466	144
5	449.310	2.566.194	28,503	117
6	442.510	2.529.674	28,7098	15
7	449.310	2.541.189	28,7277	169
8	446.710	2.564.649	28,2097	26
9	449.310	2.508.247	28,532	198
10	449.410	2.585.149	28,5605	168

D.3 Hasil percobaan dengan 50 kombinasi barang dan 500 iterasi

N	Total Berat (g)	Total Profit (Rp)	Waktu Komputasi (s)	Iterasi Non-Improvement
1	449.810	2.562.649	33,77	11
2	448.810	2.509.999	33,396	8
3	449.810	2.566.149	33,4101	9
4	446.810	2.545.149	33,6888	46
5	445.810	2.578.649	33,4855	311
6	446.310	2.589.149	33,3637	393
7	449.910	2.517.649	33,5652	178
8	449.110	2.557.649	33,2298	46
9	449.410	2.598.149	33,5285	209
10	449.310	2.594.149	33,5523	66

D.4 Hasil percobaan dengan 100 kombinasi barang dan 500 iterasi

N	Total Berat (g)	Total Profit (Rp)	Waktu Komputasi (s)	Iterasi Non-Improvement
1	449.910	2.594.649	43,1366	32
2	447.810	2.591.149	41,2087	11
3	449.310	2.538.149	41,6989	343
4	449.310	2.566.194	41,5135	103
5	449.410	2.598.149	41,4221	7
6	449.310	2.583.149	41,5153	149
7	449.910	2.594.649	41,4841	221
8	449.810	2.594.149	41,3883	228
9	448.510	2.586.149	41,4818	208
10	449.410	2.598.149	41,6762	34

D.5 Hasil percobaan dengan 250 kombinasi barang dan 500 iterasi

N	Total Berat (g)	Total Profit (Rp)	Waktu Komputasi (s)	Iterasi Non-Improvement
1	449.310	2.599.149	66,484	40
2	449.810	2.594.149	65,815	376
3	449.310	2.599.149	68,8576	26
4	449.410	2.576.149	66,5309	220
5	449.210	2.577.149	65,5677	74
6	449.310	2.599.149	66,4942	6
7	449.910	2.594.649	65,9678	4
8	447.810	2.591.149	66,1846	22
9	449.310	2.599.149	65,5471	136
10	449.110	2.557.649	65,5663	10

D.6 Hasil percobaan dengan 500 kombinasi barang dan 500 iterasi

N	Total Berat (g)	Total Profit (Rp)	Waktu Komputasi (s)	Iterasi Non-Improvement
1	449.910	2.594.649	130,4011	50
2	449.910	2.594.649	106,7134	250
3	449.310	2.599.149	113,69	21
4	449.310	2.599.149	113,9743	103
5	449.310	2.599.149	112,6048	13
6	449.310	2.583.149	105,122	87
7	449.410	2.572.389	111,723	47
8	449.910	2.594.649	110,8358	6
9	449.310	2.594.149	110,6337	27
10	449.410	2.598.149	108,824	48

D.7 Hasil percobaan dengan 1.000 kombinasi barang dan 500 iterasi

N	Total Berat (g)	Total Profit (Rp)	Waktu Komputasi (s)	Iterasi Non-Improvement
1	449.310	2.599.149	187,1592	30
2	449.310	2.599.149	192,69	19
3	449.310	2.599.149	193,9558	195
4	449.310	2.599.149	192,5347	71
5	449.310	2.599.149	195,624	4
6	449.310	2.599.149	195,559	5
7	449.310	2.599.149	194,2555	21
8	449.310	2.599.149	198,0071	7
9	449.310	2.599.149	198,0946	228
10	449.310	2.599.149	194,6745	5

D.8 Hasil percobaan dengan 100 kombinasi barang dan 1.000 iterasi

N	Total Berat (g)	Total Profit (Rp)	Waktu Komputasi (s)	Iterasi Non-Improvement
1	449.310	2.599.149	83,905	110
2	447.810	2.591.149	83,9215	17
3	449.310	2.599.149	90,6052	176
4	440.210	2.565.194	87,4262	158
5	447.610	2.575.149	110,3638	569
6	448.810	2.579.649	83,8263	611
7	448.310	2.587.649	83,8017	211
8	449.310	2.599.149	84,8245	316
9	449.810	2.594.149	83,7795	393
10	448.810	2.586.649	83,9936	476

D.9 Hasil percobaan dengan 100 kombinasi barang dan 2.000 iterasi

N	Total Berat (g)	Total Profit (Rp)	Waktu Komputasi (s)	Iterasi Non-Improvement
1	449.410	2.593.149	213,5918	825
2	449.310	2.599.149	208,541	153
3	449.810	2.566.149	215,2658	605
4	449.810	2.581.149	197,29	1319
5	449.310	2.599.149	215,8469	689
6	449.310	2.563.649	214,5158	657
7	449.810	2.581.149	205,8766	1403
8	448.810	2.586.649	213,9194	293
9	448.610	2.571.649	203,7671	13
10	449.310	2.599.149	210,0427	891

D.10 Hasil percobaan dengan 100 kombinasi barang dan 5.000 iterasi

N	Total Berat (g)	Total Profit (Rp)	Waktu Komputasi (s)	Iterasi Non-Improvement
1	449.310	2.599.149	479,2898	1087
2	449.310	2.599.149	540,1584	3239
3	449.310	2.599.149	539,827	632
4	449.310	2.599.149	530,4809	289
5	449.310	2.599.149	510,9114	76
6	449.310	2.599.149	493,7159	486
7	449.310	2.599.149	510,9344	12
8	449.310	2.599.149	537,0458	12
9	449.310	2.599.149	467,5926	637
10	449.310	2.599.149	433,1628	745