

Bidang Ilmu: MIPA

**LAPORAN AKHIR
HIBAH KERIS**



**OPTIMALISASI DAN PENERAPAN ARTIFICIAL INTELEAGENT
PADA PRODUK HASIL PERTANIAN
SEBAGAI PENDUKUNG SWASEMBADA PANGAN INDONESIA**

Oleh :

Dr. Kiswara Agung Santoso, M.Kom

Dr. Agustina Pradjaningsih, M.Si

Ahmad Kamsyakawuni, M.Kom

Abduh Riski, M.Si

UNIVERSITAS JEMBER

Nopember 2020

RINGKASAN

Penerapan Kombinasi Algoritma CSO-PSO pada Permasalahan *Modified Bounded Knapsack Multiple Constraints*; Muhammad Bagus Kurniawan, 151810101009; 2019; 86 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Permasalahan optimasi merupakan permasalahan yang sangat menarik untuk dibahas dalam ilmu matematika, salah satunya adalah permasalahan *knapsack*. Dalam kehidupan sehari-hari, permasalahan *knapsack* dapat ditemui pada beberapa kasus, misalnya pengadaan barang. Suatu perusahaan dalam mengadakan barang biasanya harus menyeleksi barang yang akan dibeli serta menentukan jumlahnya apabila dihadapkan kendala seperti berat, volume, dan modal. Pemilihan barang harus dilakukan sedemikian sehingga perusahaan mendapatkan keuntungan yang maksimal. Permasalahan ini biasa disebut dengan permasalahan *bounded knapsack multiple knapsack*. Karena dalam beberapa kasus beberapa perusahaan menentukan batas minimum jumlah barang yang harus dibeli, maka permasalahan ini menjadi permasalahan *modified bounded knapsack multiple constraints*.

Pada penelitian ini, penulis menyelesaikan permasalahan *modified bounded knapsack multiple constraints* dengan mengombinasikan dua algoritma metaheuristik yaitu algoritma CSO dan PSO menjadi algoritma CSO-PSO. Skema algoritma CSO-PSO. Skema algoritma CSO-PSO mengadopsi skema algoritma CSO, namun konsep perpindahan posisi pada *tracing mode* diganti dengan konsep perpindahan posisi algoritma PSO. Kemudian, konsep pembangkitan dan pemilihan kandidat posisi baru pada *seeking mode* dimodifikasi. Penelitian ini bertujuan untuk mengetahui pengaruh parameter algoritma; mengetahui perbandingan algoritma CSO-PSO dengan CSO dan PSO; dan mengetahui hasil terbaik serta perbandingannya terhadap hasil optimum metode *Simplex*.

Sebagai bahan simulasi, penelitian ini menggunakan data primer yang diambil dari toko Samudra yang terletak di dusun Sukodadi Genting, Sragen, Cluring, Banyuwangi. Data yang diambil berupa nama barang, jumlah minimum

dan maksimum, berat, volume, harga beli, harga jual, kapasitas kendaraan, dan modal. Proses perhitungan algoritma CSO-PSO, CSO, dan PSO dibantu dengan program simulasi yang dibuat menggunakan *software* MATLAB.

Berdasarkan hasil uji parameter, diketahui pengaruh parameter sebagai berikut. Semakin besar nilai *Iterasi*, *SMP*, dan *N* maka hasil yang didapatkan semakin mendekati optimal. Semakin kecil atau semakin besar nilai *CDC* maka hasil yang diperoleh semakin menjauhi nilai optimal. Semakin kecil c_1 dengan c_2 lebih besar namun tidak terlalu besar dapat menghasilkan solusi yang lebih baik. Semakin besar ω_{max} dan ω_{min} atau semakin mendekati 1, maka hasil yang didapatkan memiliki profit yang semakin besar. Nilai terbaik parameter tersebut dari nilai yang diujikan antara lain: *Iterasi* = 5000; *SMP* = 10; *CDC* = 0,2; c_1 = 0,5; c_2 = 1,5; ω_{max} = 0,9; ω_{min} = 0,7; dan *N* = 200. Nilai parameter *MR* dan *SRD* dalam penelitian ini dibuat tetap yaitu *MR* = 0,5 dan *SRD* = 0,1.

Dari hasil penelitian diketahui bahwa algoritma CSO-PSO lebih efektif dibandingkan algoritma CSO dan PSO. Hal ini terlihat dari rata-rata profit lebih besar dan rata-rata persentase deviasi yang lebih kecil meskipun menggunakan nilai parameter yang sama. Waktu komputasi yang dimiliki algoritma CSO-PSO hampir sama dengan algoritma CSO. Namun, jika dibandingkan dengan algoritma PSO, waktu komputasi algoritma CSO-PSO lebih lama apabila nilai parameter *SMP* lebih dari satu ($SMP > 1$). Dengan demikian, algoritma CSO-PSO masih cukup efisien.

Berdasarkan hasil simulasi akhir menggunakan parameter terbaik yang diujikan, didapatkan hasil terbaik algoritma CSO-PSO dengan total profit sebesar Rp 4.363.000,-. Hasil terbaik ini sama dengan nilai optimum metode *Simplex* sehingga persentasenya sama dengan nol. Rata-rata profit yang dihasilkan adalah Rp 4.360.750,- dengan rata-rata persentase deviasi 0,05157. Rata-rata waktu komputasi yang dibutuhkan untuk simulasi akhir ini adalah 258,19509 detik.

2.6.4 Langkah Algoritma CSO.....	13
2.7 Algoritma Particle Swarm Optimization	13
BAB 3. METODE PENELITIAN	16
3.1 Data Penelitian	16
3.2 Langkah-langkah Penelitian.....	16
BAB 4. HASIL DAN PEMBAHASAN	22
4.1 Hasil Penelitian	22
4.1.1 Program Simulasi	22
4.1.2 Hasil Simulasi	24
4.2 Pembahasan	30
4.2.1 Pseudocode Algoritma CSO-PSO.....	30
4.2.2 Perhitungan Manual	36
4.2.3 Pembahasan Hasil Simulasi	49
BAB 5. KESIMPULAN DAN SARAN	52
5.1 Kesimpulan.....	52
5.2 Saran	53
DAFTAR PUSTAKA	54
LAMPIRAN.....	56

DAFTAR TABEL

	Halaman
4.1 Hasil uji parameter <i>Iterasi</i>	25
4.2 Hasil uji parameter <i>SMP</i>	25
4.3 Hasil uji parameter <i>CDC</i>	26
4.4 Hasil uji parameter <i>c</i> algoritma CSO-PSO dan PSO	27
4.5 Hasil uji parameter <i>c</i> algoritma CSO	27
4.6 Hasil uji parameter ω_{max} dan ω_{min}	28
4.7 Hasil uji parameter <i>N</i>	29
4.8 Simulasi akhir algoritma CSO-PSO.....	29
4.9 Simulasi akhir algoritma CSO	29
4.10 Simulasi akhir algoritma PSO.....	30
4.11 Data perhitungan manual	36

DAFTAR GAMBAR

	Halaman
2.1 <i>Flowchart</i> algoritma PSO	15
3.1 <i>Flowchart</i> langkah-langkah penelitian.....	16
4.1 Tampilan program simulasi	22



DAFTAR LAMPIRAN

	Halaman
A. Data penelitian	56
B. Hasil metode <i>simplex</i>	57
C. Simulasi program	59
C.1 Uji parameter <i>Iterasi</i>	59
C.2 Uji parameter <i>SMP</i>	65
C.3 Uji parameter <i>CDC</i>	67
C.4 Uji parameter <i>c</i>	71
C.5 Uji parameter ω_{max} dan ω_{min}	79
C.6 Uji parameter <i>N</i>	83

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Permasalahan *knapsack* merupakan salah satu permasalahan optimasi yang sangat menarik untuk dibahas. Hal ini terlihat dari banyaknya penelitian tentang penyelesaian permasalahan *knapsack*. Menurut Gherboudj *et al.* (2012) permasalahan *knapsack* didefinisikan sebagai permasalahan penentuan atau pemilihan barang dari sekian banyak barang untuk dimasukkan ke dalam media penyimpanan (*knapsack*) dimana setiap barang memiliki berat dan profit. Total berat dari semua barang terpilih tidak boleh melebihi kapasitas media penyimpanan yang digunakan. Pemilihan barang dilakukan sedemikian sehingga total profitnya dapat dimaksimalkan.

Permasalahan *knapsack* memiliki beberapa jenis berdasarkan persoalannya, yaitu *knapsack* 0-1, *bounded knapsack*, dan *unbounded knapsack* (Pisinger, 1995). Berdasarkan tujuan, jumlah kendala dan jumlah media penyimpanan, permasalahan *knapsack* dibedakan menjadi beberapa variasi, yaitu *single objective knapsack*, *multi objective knapsack*, *multidimensional knapsack*, *multiple constraints knapsack*, dan *quadratic knapsack* (Kellerer *et al.*, 2004). Terdapat beberapa penelitian yang juga membahas kombinasi dari jenis persoalan dan variasi permasalahan *knapsack* tersebut, diantaranya sebagai berikut. Lopez, *et al.* (2018) meneliti tentang *multidimensional knapsack* 0-1. Kasyfi (2019) meneliti penyelesaian *multidimensional bounded knapsack*. Baihaki (2017) membahas tentang *multiple constraints knapsack* 0-1. Mahendra (2019) membahas tentang *multi knapsack* 0-1 with *multiple constraints*.

Penerapan permasalahan *knapsack* di kehidupan sehari-hari dapat ditemui pada beberapa kasus. Pengadaan barang pada suatu perusahaan maupun industri merupakan salah satu kasus nyata dari penerapan permasalahan *knapsack* yang sangat penting untuk dilakukan manajemen. Dalam proses pengadaan barang, apabila perusahaan atau industri memiliki beberapa kendala, tentunya sebuah perusahaan harus menyeleksi barang yang akan dibeli atau sebuah industri harus memilih barang yang akan diproduksinya, serta memperhitungkan jumlahnya.

Dengan adanya proses seleksi, perusahaan dapat memilih barang-barang yang akan menghasilkan keuntungan total maksimum. Kendala yang mungkin dihadapi dapat berupa kapasitas media penyimpanan dan keterbatasan modal. Kapasitas media penyimpanan yang menjadi pertimbangan adalah kapasitas berat dan kapasitas ruang. Dalam ilmu matematika, permasalahan ini disebut dengan permasalahan *bounded knapsack multiple constraints*. Namun, dalam beberapa kasus, perusahaan harus menentukan jumlah minimal dari setiap barang agar barang tersebut selalu tersedia. Dengan demikian, permasalahan ini akan dimasukkan ke dalam permasalahan *modified bounded knapsack multiple constraints*.

Permasalahan optimasi yang meliputi permasalahan *knapsack* ini dapat diselesaikan dengan beberapa metode atau algoritma. Metode *Simplex* merupakan salah satu metode eksak yang umum digunakan. Namun, untuk permasalahan dengan skala besar, metode eksak menjadi kurang efisien, sehingga banyak peneliti lebih memilih untuk menggunakan algoritma metaheuristik. Menurut Boussaid *et al.* (2013) algoritma metaheuristik merupakan algoritma yang didesain untuk menyelesaikan permasalahan optimasi melalui proses pendekatan yang banyak terinspirasi dari alam, seperti biologi, fisika atau etologi.

Algoritma *Cat Swarm Optimization* (CSO) merupakan salah satu algoritma metaheuristik yang pertama kali dikenalkan oleh Chu *et al.* (2006). Pembentukan algoritma CSO terinspirasi oleh perilaku kucing. Algoritma CSO telah banyak diterapkan pada beberapa permasalahan, antara lain optimasi fungsi (Chu *et al.*, 2006), *clustering* (Santosa dan Ningrum, 2009), optimasi *artificial neural networks* (Yusiong, 2013), *knapsack* 0-1 (Sharafi *et al.*, 2013), *workflow scheduling* (Bilgaiyan *et al.*, 2014). Dalam algoritma CSO, terdapat dua mode, yaitu *seeking mode* dan *tracing mode*. Pada *seeking mode*, kucing membentuk beberapa kandidat posisi (solusi) baru di lingkungan sekitarnya, kemudian memilih salah satu secara acak dan kucing berpindah. Pada *tracing mode*, kucing akan berpindah tempat berdasarkan posisi terbaik dari kawanan.

Algoritma *Particle Swarm Optimization* (PSO) juga merupakan salah satu algoritma metaheuristik yang telah banyak digunakan. Algoritma PSO dibuat oleh Kennedy dan Eberhart (1995) berdasarkan perilaku sekawanan burung atau ikan di

alam. Dalam algoritma PSO, posisi terbaik setiap partikel dan posisi terbaik global disimpan, kemudian setiap solusi diperbarui berdasarkan posisi terbaik individu dan posisi terbaik global tersebut. Permasalahan yang pernah diselesaikan dengan algoritma PSO antara lain, optimasi fungsi (Kennedy dan Eberhart, 1995), *traveling salesman problem* (Clerc, 2004), *hybrid flow-shop scheduling* (Tseng dan Liao, 2008), *vehicle routing problem* (Pornsing, 2014), *vehicle routing problem with time windows* (Fatimah, 2016).

Di samping kedua algoritma yang telah diuraikan di atas, banyak peneliti telah menggabungkan beberapa algoritma sebagai algoritma *hybrid* (kombinasi). Menurut Mehdi (2011) algoritma *hybrid* lebih efisien untuk permasalahan dalam skala besar. Algoritma *hybrid* juga mampu menyeimbangkan antara eksplorasi dan eksploitasi, serta menyeimbangkan waktu eksekusi dan kualitas solusi akhir. Algoritma *hybrid* dapat berupa gabungan metode eksak dengan algoritma metaheuristik, algoritma heuristik dengan algoritma metaheuristik, algoritma metaheuristik dengan algoritma metaheuristik lain, dan sebagainya.

Berdasarkan uraian di atas, penulis tertarik untuk menggabungkan algoritma CSO dengan algoritma PSO sebagai algoritma CSO-PSO, kemudian diterapkan pada permasalahan *modified bounded knapsack multiple constraints*. Pada penelitian ini, skema dari algoritma CSO-PSO yang diajukan sama dengan skema algoritma CSO, namun pada proses *tracing mode* penulis mengambil prinsip algoritma PSO yaitu perpindahan setiap individu didasari oleh posisi terbaik individu tersebut dan posisi terbaik global. Selain itu, penulis akan memodifikasi proses *seeking mode*, kandidat posisi baru yang dibentuk didasarkan pada posisi terbaik individu kemudian beberapa dimensi dimutasi berdasarkan posisi terbaik global. Pada algoritma CSO-PSO ini, aturan pemilihan kandidat posisi baru yang dibentuk di proses *seeking mode* adalah posisi yang terbaik. Dari penggabungan ini, diharapkan algoritma CSO-PSO dapat lebih cepat konvergen menuju solusi optimalnya (lebih efektif) pada permasalahan *modified bounded knapsack multiple constraints*. Oleh karena itu, untuk melihat performa dari algoritma CSO-PSO, penulis akan membuat program simulasi menggunakan *software* MATLAB. Program tersebut digunakan untuk membantu menguji pengaruh parameter dan

untuk melihat perbandingan performa algoritma CSO-PSO dengan algoritma CSO dan PSO. Selanjutnya, kualitas hasil (keoptimalan) akan dianalisis berdasarkan persentase deviasi perbandingan hasil algoritma dengan hasil metode *Simplex* yang dijalankan menggunakan Solver Add-In yang tersedia pada Microsoft Excel.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah yang akan dibahas pada penelitian ini sebagai berikut.

- a. Bagaimana pengaruh parameter pada algoritma CSO-PSO, CSO dan PSO dalam penyelesaian permasalahan *modified bounded knapsack multiple constraints*?
- b. Bagaimana perbandingan algoritma CSO-PSO dengan algoritma CSO dan PSO berdasarkan hasil penyelesaian permasalahan *modified bounded knapsack multiple constraints*?
- c. Bagaimana hasil terbaik algoritma CSO-PSO dalam penyelesaian permasalahan *modified bounded knapsack multiple constraints* dan dibandingkan dengan hasil metode *Simplex*?

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini antara lain sebagai berikut.

- a. Mengetahui pengaruh parameter pada algoritma CSO-PSO, CSO dan PSO dalam penyelesaian permasalahan *modified bounded knapsack multiple constraints*.
- b. Mengetahui perbandingan algoritma CSO-PSO dengan algoritma CSO dan PSO berdasarkan hasil penyelesaian permasalahan *modified bounded knapsack multiple constraints*.
- c. Mengetahui hasil terbaik algoritma CSO-PSO dalam penyelesaian permasalahan *modified bounded knapsack multiple constraints* dan dibandingkan dengan hasil metode *Simplex*.

1.4 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah dapat dijadikan acuan dalam penyeleksian barang, serta dapat dijadikan referensi untuk penelitian selanjutnya yang membahas permasalahan *knapsack*, serta penerapan algoritma algoritma CSO-PSO, CSO dan PSO pada permasalahan lain.



BAB 2. TINJAUAN PUSTAKA

2.1 Permasalahan *Knapsack*

Permasalahan *knapsack* merupakan permasalahan optimasi kombinatorik klasik yang sangat penting (Dantzig, 1957). Permasalahan *knapsack* didefinisikan sebagai permasalahan penentuan atau pemilihan barang dari sekian banyak barang untuk dimasukkan ke dalam media penyimpanan (*knapsack*) dimana setiap barang memiliki berat dan profit. Total berat dari semua barang terpilih tidak boleh melebihi kapasitas media penyimpanan yang digunakan. Pemilihan barang dilakukan sedemikian sehingga total profitnya dapat dimaksimalkan (Gherboudj *et al.*, 2012). Pemodelan dan penerapan permasalahan *knapsack* dapat ditemui dalam berbagai bidang, antara lain *resource allocations*, *capital budgets*, *investment decisions*, *industrial loading*, *economy and finance*, dan *information security* (Martello dan Toth, 1990).

Pisinger (1995) membagi permasalahan *knapsack* menjadi tiga jenis, yaitu *knapsack 0-1*, *bounded knapsack*, dan *unbounded knapsack*. Permasalahan *knapsack 0-1* diartikan sebagai permasalahan *knapsack* dimana hanya terdapat dua variabel keputusan yaitu 1 (barang dipilih) dan 0 (barang tidak dipilih). Permasalahan *bounded knapsack* diartikan sebagai permasalahan *knapsack* dimana barang dapat dipilih jumlahnya sesuai dengan batas ketersediaannya. Permasalahan *unbounded knapsack* diartikan sebagai permasalahan *knapsack* dimana jumlah barang yang dipilih tidak terbatas.

Berdasarkan tujuan, jumlah kendala dan jumlah media penyimpanan, Kellerer *et al.* (2004) membedakan permasalahan *knapsack* menjadi beberapa variasi, yaitu *single objective knapsack*, *multi objective knapsack*, *multidimensional knapsack*, *multiple constraints knapsack*, dan *quadratic knapsack*. Permasalahan *single objective knapsack* adalah permasalahan *knapsack* yang memiliki fungsi tujuan tunggal. Permasalahan *multi objective knapsack* adalah permasalahan *knapsack* yang memiliki fungsi tujuan lebih dari satu. Permasalahan *multidimensional knapsack* adalah permasalahan *knapsack* yang memiliki media penyimpanan lebih dari satu. Permasalahan *multiple constraints knapsack* adalah permasalahan

knapsack yang memiliki kendala lebih dari satu. Permasalahan *quadratic knapsack* adalah permasalahan *knapsack* yang memiliki fungsi tujuan dalam bentuk kuadrat.

2.2 Permasalahan *Bounded Knapsack*

Permasalahan *bounded knapsack* merupakan generalisasi dari permasalahan *knapsack* 0-1, dimana setiap barang memiliki jumlah ketersediaan. Permasalahan *bounded knapsack* menurut Pisinger (2000) didefinisikan sebagai permasalahan *knapsack* dimana terdapat media penyimpanan berkapasitas yang harus diisi dengan beberapa jenis barang. Setiap jenis barang memiliki profit, berat, dan jumlah ketersediaan. Permasalahan yang dihadapi adalah bagaimana memilih jumlah dari setiap jenis barang sedemikian sehingga total profitnya dimaksumalkan dengan total berat tidak melebihi kapasitas. Permasalahan *bounded knapsack* diformulasikan sebagai berikut.

Fungsi tujuan:

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

Kendala:

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

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

dengan

Z : fungsi tujuan (total keuntungan)

n : banyaknya jenis barang

p_j : profit barang jenis j

w_j : berat barang jenis j

x_j : jumlah barang jenis j yang dimasukkan ke dalam media penyimpanan

m_j : jumlah ketersediaan barang jenis j

C : kapasitas media penyimpanan

2.3 Permasalahan *Multiple Constraints Knapsack*

Permasalahan *multiple constraints knapsack* merupakan permasalahan *knapsack* dimana dalam pemilihan barang terdapat kendala lebih dari satu. Kendala tersebut adalah kapasitas media penyimpanan (berat dan ruang) dan biaya/modal. Tujuan dari permasalahan ini adalah memilih barang sedemikian sehingga total profitnya maksimal dan memenuhi kendala yang ada. Permasalahan *multiple constraints knapsack* dirumuskan secara matematis sebagai berikut.

Fungsi tujuan:

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

Kendala:

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

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

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

$$x_j \in \{0, 1\}, j = 1, 2, \dots, n \quad (2.8)$$

dengan

Z : fungsi tujuan (total keuntungan)

n : banyaknya jenis barang

p_j : profit barang jenis j

w_j : berat barang jenis j

v_j : volume barang jenis j

b_j : harga beli barang jenis j

x_j : variabel keputusan (1 jika dipilih dan 0 jika tidak dipilih)

C : kapasitas berat media penyimpanan

S : kapasitas ruang media penyimpanan

M : modal

2.4 Permasalahan *Modified Bounded Knapsack Multiple Constraints*

Permasalahan *bounded knapsack multiple constraints* merupakan permasalahan *knapsack* dimana terdapat jumlah ketersediaan dari setiap jenis barang dan terdapat kendala lebih dari satu. Kendala tersebut adalah kapasitas

media penyimpanan (berat dan ruang) dan biaya/modal. Tujuan dari permasalahan ini adalah menentukan jumlah dari setiap jenis barang sedemikian sehingga total profitnya maksimal dan memenuhi kendala yang ada.

Permasalahan *modified bounded knapsack multiple constraints* merupakan modifikasi permasalahan *bounded knapsack multiple constraints* yang terletak pada variabel keputusannya. Misalkan diasumsikan setiap barang memiliki batas minimal yang harus dibeli dan memiliki batas maksimal ketersediaan. Batas minimal ini digunakan untuk menentukan prioritas dari setiap barang yang akan dibeli. Permasalahan *modified bounded knapsack multiple constraints* dirumuskan secara matematis sebagai berikut.

Fungsi tujuan:

$$\text{Maksimalkan } Z = \sum_{j=1}^n p_j x_j \quad (2.9)$$

Kendala:

$$\sum_{j=1}^n w_j x_j \leq C \quad (2.10)$$

$$\sum_{j=1}^n v_j x_j \leq S \quad (2.11)$$

$$\sum_{j=1}^n b_j x_j \leq M \quad (2.12)$$

$$x_j \in \mathbb{Z}, \quad lb_j \leq x_j \leq ub_j, \quad j = 1, 2, \dots, n \quad (2.13)$$

dengan

Z : fungsi tujuan (total keuntungan)

n : banyaknya jenis barang

p_j : profit barang jenis j

w_j : berat barang jenis j

v_j : volume barang jenis j

b_j : harga beli barang jenis j

x_j : jumlah barang jenis j yang dimasukkan ke dalam media penyimpanan

lb_j : jumlah minimum barang jenis j yang harus dibeli

ub_j : jumlah ketersediaan barang jenis j

C : kapasitas berat media penyimpanan

S : kapasitas ruang media penyimpanan

M : modal

2.5 Algoritma Metaheuristik

Menurut Boussaid *et al.* (2013) algoritma metaheuristik merupakan algoritma yang didesain untuk menyelesaikan permasalahan optimasi melalui proses pendekatan yang banyak terinspirasi dari alam, seperti biologi, fisika, atau etologi. Algoritma metaheuristik memiliki parameter yang dibutuhkan untuk menyesuaikan dengan permasalahan yang diselesaikan. Algoritma metaheuristik diterapkan secara luas pada permasalahan yang kompleks dalam industri dan jasa.

Menurut Yang (2013) algoritma metaheuristik dianggap sebagai langkah efisien untuk menghasilkan solusi yang dapat diterima melalui proses *trial and error* pada permasalahan yang kompleks. Kompleksitas dari masalah yang dihadapi membuatnya tidak mungkin untuk dicari setiap kemungkinan atau kombinasi solusi. Tujuannya adalah mencari solusi *feasible* yang baik dalam skala waktu yang dapat diterima. Tidak ada jaminan bahwa solusi terbaik akan ditemukan, namun dari penerapan-penerapan algoritma metaheuristik yang pernah dilakukan mampu menghasilkan solusi dengan kualitas yang baik (mendekati optimal) dengan waktu yang efisien.

Komponen utama dari setiap algoritma metaheuristik adalah *intensification* dan *diversification*, atau *exploitation* dan *exploration*. *Diversification* berarti membangkitkan solusi berbeda untuk menjelajahi ruang pencarian dalam skala global, sedangkan *intensification* berarti fokus pencarian lokal dengan *exploiting* informasi bahwa solusi yang baik ditemukan dalam wilayah pencarian tersebut. Selain itu, algoritma metaheuristik juga memiliki komponen penting yaitu pengacakan, yang mana untuk membuat algoritma mampu melompat dari setiap optimum lokal sehingga mampu mencari secara global. Teknik pengacakan dapat metode sederhana dengan distribusi seragam atau Gaussian, atau metode kompleks seperti *Levy flight*.

2.6 Algoritma Cat Swarm Optimization

Algoritma *Cat Swarm Optimization* (CSO) merupakan algoritma metaheuristik yang dibentuk oleh Chu *et al.* (2006) berdasarkan perilaku kucing. Kucing liar memiliki kemampuan berburu yang baik untuk bertahan hidup,

sedangkan kucing rumahan memiliki insting alami yang kuat tertarik pada benda bergerak. Meskipun kucing menghabiskan banyak waktunya untuk tidur, namun kucing memiliki kewaspadaan yang sangat tinggi. Kucing selalu tetap terjaga walaupun sedang beristirahat. Kucing tetap memperhatikan lingkungan sekitarnya untuk mengetahui adanya bahaya maupun makanan/mangsa. Dalam algoritma CSO, perilaku kucing dibagi menjadi dua mode utama, yaitu *seeking mode* dan *tracing mode*.

2.6.1 Bentuk Solusi CSO

Langkah pertama dalam algoritma CSO adalah menentukan banyak individu kucing yang akan digunakan, selanjutnya diterapkan pada permasalahan yang dihadapi. Setiap kucing memiliki posisi yang disusun dari M dimensi, kecepatan untuk setiap dimensi, nilai *fitness* dan tanda (*flag*). Nilai *fitness* digunakan untuk mengetahui kualitas solusi, sedangkan *flag* digunakan untuk mengidentifikasi kucing berada dalam *seeking mode* atau *tracing mode*. Solusi akhir dari algoritma CSO adalah posisi terbaik dari semua kucing yang disimpan selama iterasi.

2.6.2 Seeking Mode

Seeking mode digunakan untuk memodelkan situasi kucing yang sedang beristirahat. Kucing memperhatikan lingkungan sekitar dan mencari posisi baru untuk berpindah. Dalam *seeking mode*, didefinisikan empat faktor atau parameter, yaitu *seeking memory pool* (SMP), *seeking ranger of selected dimension* (SRD), *counts of dimension to change* (CDC), dan *self-position considering* (SPC). SMP digunakan untuk mendefinisikan ukuran memori pencarian dari setiap kucing. Kucing akan memilih titik baru dari memori berdasarkan aturan tertentu. SMP memiliki nilai berupa bilangan bulat positif. SRD sebagai rasio perubahan untuk dimensi terpilih. Dalam *seeking mode*, jika suatu dimensi dipilih untuk dimutasi, selisih antara nilai baru dan nilai lama tidak akan keluar dari batas yang didefinisikan oleh SRD. CDC digunakan untuk menentukan banyak dimensi yang akan dimutasi atau diubah. Nilai CDC direspresentasikan dalam interval $[0, 1]$, yang artinya berupa persentase dari banyak dimensi. Faktor ini berpengaruh besar dalam

seeking mode. SPC merupakan variabel *Boolean* (benar (1) atau salah (0)). Faktor SPC digunakan untuk menentukan kandidat posisi kucing yang baru untuk berpindah.

Proses *seeking mode* dideskripsikan dalam lima langkah berikut.

- Membuat *copy* sejumlah j dari posisi kucing ke- k , dimana $j = SMP$. Jika nilai SPC adalah benar, maka $j = SMP - 1$ dan satu kandidat posisi baru diambil sama dengan posisi lama.
- Untuk setiap *copy*, berdasarkan CDC, secara acak nilai lama diganti dengan nilai baru, dimana nilai baru dihitung menggunakan Persamaan (2.14).

$$x'_{j,d} = x_{j,d} \pm SRD \cdot r \cdot x_{j,d} \quad (2.14)$$

- Hitung nilai *fitness* (FS) dari semua kandidat.
- Jika semua nilai *fitness* dari semua kandidat tidak benar-benar sama, hitung probabilitas pemilihan berdasarkan Persamaan (2.15). Namun, jika semua nilai *fitness* adalah sama, maka probabilitas setiap kandidat diatur sama dengan satu.

$$P_i = \frac{|FS_i - FS_b|}{FS_{max} - FS_{min}} \quad (2.15)$$

dimana $FS_b = FS_{max}$ jika permasalahan minimasi, dan $FS_b = FS_{min}$ jika masalah maksimasi.

- Pilih secara acak posisi baru dari kandidat untuk menggantikan posisi kucing.

2.6.3 Tracing Mode

Tracing mode digunakan untuk memodelkan kasus dimana kucing melacak beberapa target atau mangsa. Sekali seekor kucing masuk dalam *tracing mode*, kucing berpindah berdasarkan kecepatannya sendiri untuk setiap dimensi. Proses *tracing mode* dideskripsikan dalam tiga langkah berikut.

- Perbarui kecepatan untuk setiap dimensi ($v_{k,d}$) berdasarkan Persamaan (2.16).

$$v_{k,d} = v_{k,d} + r_1 c_1 (x_{best,d} - x_{k,d}), \quad d = 1, 2, \dots, M \quad (2.16)$$

dengan

$x_{best,d}$: posisi terbaik kucing

$x_{k,d}$: posisi kucing ke- k

c_1 : koefisien percepatan

r_1 : bilangan acak dalam interval $0 \leq r_1 \leq 1$

- b. Periksa kecepatan berada di dalam batas maksimum atau keluar batas. Jika kecepatan baru keluar dari batas, maka atur kecepatan sama dengan batas.
- c. Perbarui posisi kucing ke- k berdasarkan Persamaan (2.17).

$$x_{k,d} = x_{k,d} + v_{k,d} \quad (2.17)$$

2.6.4 Langkah Algoritma CSO

Seperti yang telah diuraikan di atas, algoritma CSO memiliki dua mode, yaitu *seeking mode* dan *tracing mode*. Untuk menggabungkan kedua mode ke dalam algoritma, digunakan *mixture ratio* (MR). Faktor MR digunakan untuk membagi kucing yang masuk dalam *seeking mode* dan *tracing mode*. Proses dari algoritma CSO dideskripsikan dalam delapan langkah berikut.

- a. Bangkitkan posisi N kucing dalam ruang pencarian serta kecepatan awal-nya.
- b. Evaluasi fungsi *fitness*-nya dan simpan posisi terbaik kucing.
- c. Bangkitkan flag secara random, kemudian bagi kucing ke dalam *seeking mode* dan *tracing mode* sesuai *flag*.
- d. Jika kucing masuk dalam *seeking mode*, terapkan proses *seeking mode*.
- e. Jika kucing tidak masuk dalam *seeking mode*, terapkan proses *tracing mode*.
- f. Gabungkan kembali hasil *seeking mode* dan *tracing mode*.
- g. Evaluasi fungsi *fitness*-nya dan perbarui posisi terbaik kucing.
- h. Periksa kriteria pemberhentian. Jika kriteria pemberhentian dicapai, algoritma dihentikan dan posisi terbaik sebagai solusi akhir. Namun, jika kriteria pemberhentian belum dicapai, kembali ke langkah c.

2.7 Algoritma Particle Swarm Optimization

Algoritma *Particle Swarm Optimization* (PSO) merupakan algoritma metaheuristik yang pertama kali dikenalkan oleh James Kennedy dan Russell Eberhart pada tahun 1995. Algoritma PSO didasarkan pada perilaku sekawanan burung atau ikan di alam. Algoritma PSO telah banyak diterapkan pada hampir setiap area optimasi, kecerdasan komputasi dan aplikasi desain/penjadwalan (Yang, 2013).

Terdapat partikel-partikel dibangkitkan secara stokastik dalam ruang pencarian pada algoritma PSO. Setiap partikel merupakan kandidat solusi untuk permasalahan yang direpresentasikan dengan posisi, kecepatan dan memiliki memori untuk membantunya mengingat posisi terbaik sebelumnya. Algoritma PSO terdiri atas N partikel. Setiap kawanan partikel memiliki sejenis topologi yang mendeskripsikan hubungan antar partikel. Jenis-jenis topologi tersebut telah digunakan untuk mengidentifikasi beberapa partikel lain untuk memengaruhi individual. Dua topologi yang paling sering digunakan disebut dengan *gbest* (terbaik global) dan *lbest* (terbaik lokal). *gbest* merupakan posisi terbaik dari keseluruhan populasi, sedangkan *lbest* merupakan posisi terbaik dari setiap partikel. *gbest* digunakan sebagai pencarian cepat dan *lbest* digunakan untuk pencarian lambat (Boussaid *et al.*, 2013).

Proses inisialisasi dari algoritma PSO dilakukan dengan membangkitkan posisi dan kecepatan setiap partikel secara acak. Pada setiap iterasi, setiap partikel i memperbarui posisi X_i dan kecepatan V_i berdasarkan posisi terbaik partikel itu sendiri P_i dan posisi terbaik global P_g . Kecepatan partikel diperbarui berdasarkan Persamaan (2.18) berikut.

$$V_i(t + 1) = V_i(t) + c_1 r_1 (P_i(t) - X_i(t)) + c_2 r_2 (P_g(t) - X_i(t)) \quad (2.18)$$

dengan

c_1, c_2 : koefisien percepatan dalam interval $0 \leq c_1, c_2 \leq 2$

r_1, r_2 : bilangan acak dalam interval $0 \leq r_1, r_2 \leq 1$

Selanjutnya setiap partikel memperbarui posisinya berdasarkan Persamaan (2.19) berikut.

$$X_i(t + 1) = X_i(t) + V_i(t + 1) \quad (2.19)$$

Untuk mencegah algoritma PSO mengalami konvergen dini, banyak strategi telah dibentuk, namun yang paling umum digunakan adalah inersia. Bobot inersia ω berguna untuk menyeimbangkan pencarian global dan lokal (Shi dan Eberhart, 1998). Bobot inersia dapat berupa konstanta positif maupun fungsi positif linier atau nonlinier. Nilai bobot inersia yang besar menjamin eksplorasi global

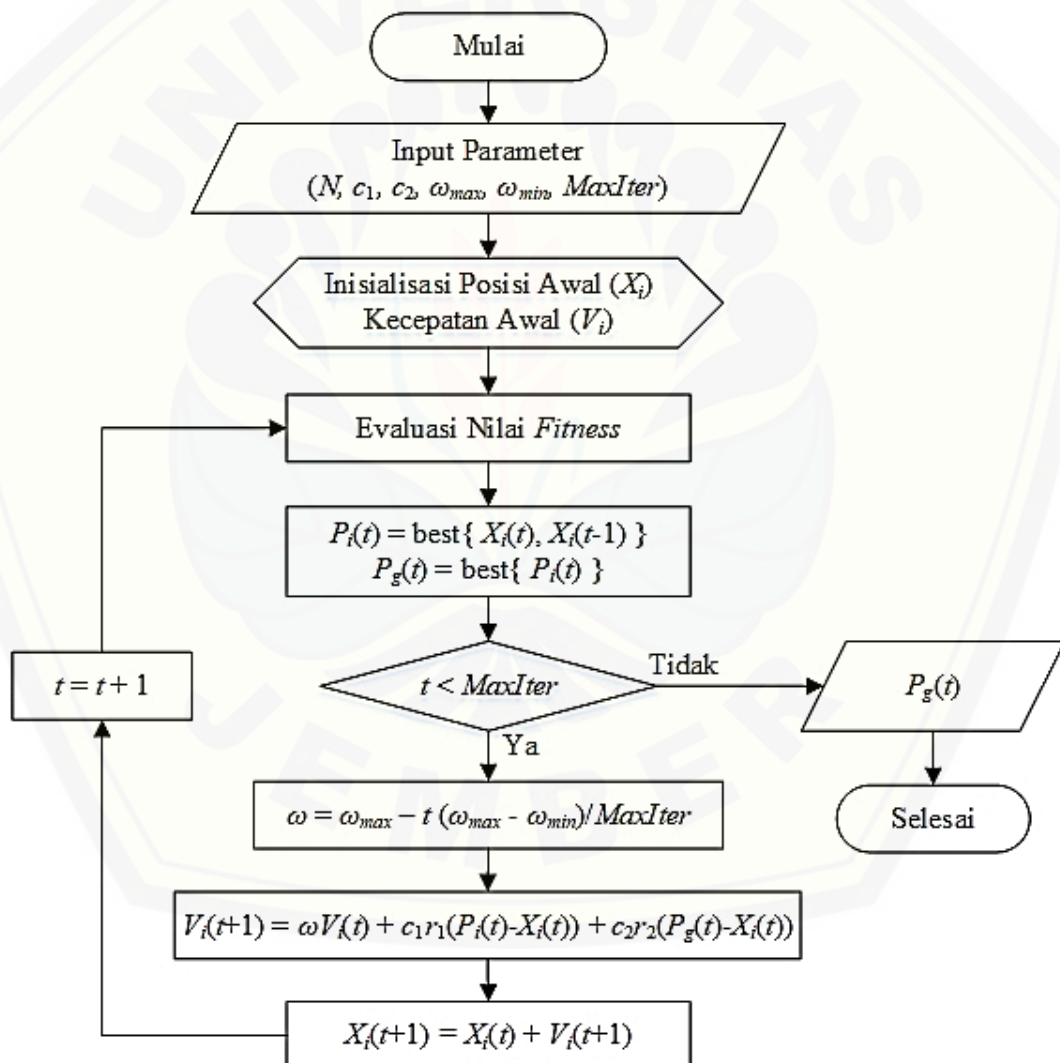
sedangkan nilai yang kecil menjamin eksploitasi lokal. Bobot inersia dalam bentuk fungsi linier dituliskan pada Persamaan (2.20).

$$\omega = \omega_{max} - t \times \frac{\omega_{max} - \omega_{min}}{MaxIter} \quad (2.20)$$

Dengan menggunakan bobot inersia, Persamaan (2.18) diubah menjadi Persamaan (2.21) berikut.

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 (P_i(t) - X_i(t)) + c_2 r_2 (P_g(t) - X_i(t)) \quad (2.21)$$

Secara garis besar, langkah-langkah algoritma PSO disajikan dalam *flowchart* pada Gambar 2.1.



Gambar 2.1 *Flowchart* algoritma PSO

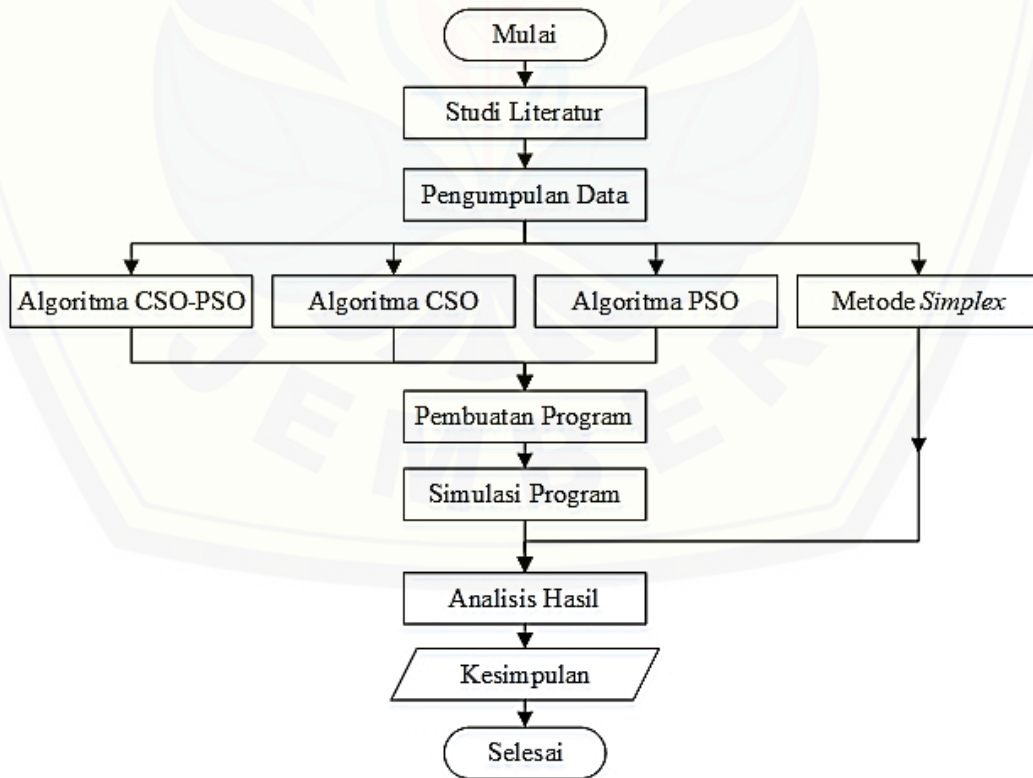
BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang akan digunakan dalam penelitian ini adalah data primer berupa data sejumlah barang, data kendaraan serta modal Toko Samudra yang terletak di dusun Sukodadi Genting RT 03 RW 02, desa Sragen, kecamatan Cluring, kabupaten Banyuwangi, Jawa Timur. Data barang yang diambil terdiri atas nama barang, jumlah minimal, jumlah maksimal, berat, volume, harga beli dan harga jual dari setiap jenis barang. Data kendaraan terdiri atas kapasitas berat yaitu sebesar 4.000 kg dan kapasitas ruang sebesar 6.000.000 cm³. Modal yang dimiliki pada data tersebut adalah Rp 30.000.000,-. Data barang dapat dilihat pada Lampiran A.

3.2 Langkah-langkah Penelitian

Adapun langkah-langkah yang akan dilakukan pada penelitian ini dapat dilihat pada Gambar 3.1



Gambar 3.1 *Flowchart* langkah-langkah penelitian

Adapun penjelasan dari *flowchart* langkah-langkah yang akan dilakukan pada penelitian ini sebagai berikut.

a. Studi Literatur

Studi literatur merupakan langkah awal yang akan dilakukan dalam penelitian ini, yang mana penulis akan mengumpulkan beberapa referensi mengenai permasalahan *modified bounded knapsack multiple constraints* serta algoritma CSO dan PSO, kemudian mempelajari literatur tersebut.

b. Pengumpulan Data

Pada langkah ini, penulis akan mengumpulkan dan mengidentifikasi data barang dan kendaraan dari Toko Samudra sebagai bahan simulasi.

c. Penerapan Algoritma

1) Penerapan CSO-PSO

Pada algoritma CSO-PSO yang diajukan ini, setiap solusi dari CSO akan disimpan terbaiknya dan terbaik globalnya berdasarkan prinsip algoritma PSO. Selanjutnya, pada proses *tracing mode*, perpindahan kucing menggunakan konsep perpindahan partikel, yang mana perpindahan didasari oleh posisi terbaik individu dan terbaik global. Selain itu, penulis akan memodifikasi proses *seeking*, kandidat posisi baru yang dibentuk didasarkan pada posisi terbaik individu kemudian beberapa dimensi dimutasi berdasarkan posisi terbaik global. Langkah-langkah algoritma CSO-PSO pada permasalahan *modified bounded knapsack multiple constraints* sebagai berikut.

- a) Bangkitkan posisi N kucing (X) secara acak dalam ruang pencarian $([0, 1])$ serta kecepatan awal (V).
- b) Ubah posisi (X) ke bentuk solusi (Y) permasalahan *modified bounded knapsack multiple constraints* melalui perkalian posisi kucing ($x_{k,d}$) dengan selisih jumlah ketersediaan barang (ub_d) dan jumlah minimal barang yang harus dibeli (lb_d) untuk setiap dimensi, kemudian dibulatkan dan ditambahkan jumlah minimal. Rumus untuk mengubah posisi (X) ke bentuk solusi (Y) dituliskan pada Persamaan (3.1).

$$y_{k,d} = [x_{k,d} \times (ub_d - lb_d)] + lb_d, \quad d = 1, 2, \dots, D \quad (3.1)$$

dengan

$y_{k,d}$: jumlah barang d yang dimasukkan *knapsack*

$x_{k,d}$: posisi kucing dimensi ke- d

lb_d : jumlah minimal barang jenis d yang harus dibeli

ub_d : jumlah ketersediaan barang jenis d

D : banyaknya jenis barang.

c) Periksa kendala dari setiap solusi.

Setiap solusi dari algoritma CSO-PSO harus memenuhi kendala permasalahan *modified bounded knapsack multiple constraints* pada Persamaan (3.2) – (3.4). Hal ini dimaksudkan agar setiap solusi selalu berada pada daerah *feasible*.

$$\sum_{d=1}^D w_d y_d \leq C \quad (3.2)$$

$$\sum_{d=1}^D v_d y_d \leq S \quad (3.3)$$

$$\sum_{d=1}^D b_d y_d \leq M \quad (3.4)$$

dengan

w_d : berat barang jenis d

v_d : volume barang jenis d

b_d : harga beli barang jenis d

C : kapasitas berat media penyimpanan

S : kapasitas ruang media penyimpanan

M : modal

Jika terdapat solusi yang tidak memenuhi kendala, maka harus dilakukan penalti untuk mengurangi jumlah barang yang dipiih. Proses penalti dilakukan terhadap posisi kucing pada dimensi yang dipilih secara acak ($x_{k,d}$), kemudian solusi ($y_{k,d}$) menyesuaikan. Namun, untuk solusi awal yang tidak memenuhi kendala, dapat dilakukan pembangkitan ulang secara acak hingga ditemukan solusi yang memenuhi. Rumus penalti menggunakan Persamaan (3.5)

$$x_{k,d} = x_{k,d} - \frac{1}{ub_d - lb_d} \quad (3.5)$$

Apabila nilai $x_{k,d}$ yang baru menjadi negatif, maka harus diganti dengan nilai lain yang dibangkitkan secara acak menggunakan Persamaan (3.6).

$$x_{k,d} = \frac{0,5 \times \text{rand}(0,1)}{ub_d - lb_d} \quad (3.6)$$

d) Evaluasi fungsi *fitness* (total profit)

Pada langkah ini dihitung nilai fungsi *fitness* dari setiap solusi berdasarkan total profit. Nilai *fitness* digunakan untuk mengetahui kualitas solusi. Total profit dihitung menggunakan Persamaan (3.7).

$$Z = \sum_{d=1}^D p_d y_d \quad (3.7)$$

dengan

Z : total profit

p_d : profit barang jenis d

e) Simpan posisi terbaik dari setiap kucing sebagai C_k dan simpan posisi terbaik global sebagai C_g .

f) Bangkitkan *flag* secara random, kemudian bagi kucing ke dalam *seeking mode* dan *tracing mode* sesuai *flag*.

g) Jika kucing masuk dalam *seeking mode*, terapkan proses *seeking mode*. Setiap kandidat posisi baru dibangkitkan dari posisi terbaik setiap kucing C_k , kemudian dimutasi beberapa dimensinya berdasarkan posisi terbaik global C_g dengan rumus mutasi pada Persamaan (3.8).

$$x_{j,d}' = C_{g,d} \pm SRD \cdot r \cdot C_{g,d} \quad (3.8)$$

h) Jika kucing tidak masuk dalam *seeking mode*, terapkan proses *tracing mode*. Untuk memperbarui kecepatan kucing digunakan Persamaan (3.9), sedangkan untuk memperbarui posisi kucing digunakan Persamaan (3.10).

$$v_{k,d} = \omega v_{k,d} + r_1 c_1 (C_{g,d} - x_{k,d}) + r_2 c_2 (C_{k,d} - x_{k,d}) \quad (3.9)$$

$$x_{k,d} = x_{k,d} + v_{k,d} \quad (3.10)$$

dengan

ω : bobot inersia

c_1, c_2 : koefisien percepatan dalam interval $0 \leq c_1, c_2 \leq 2$

r_1, r_2 : bilangan acak dalam interval $0 \leq r_1, r_2 \leq 1$

- i) Gabungkan kembali hasil *seeking mode* dan *tracing mode*. Pastikan posisi kucing berada pada ruang pencarian. Jika terdapat posisi kucing yang keluar, artinya nilai $\min(x_k) < 0$ atau $\max(x_k) > 1$, maka perlu dilakukan transformasi nilai menggunakan Persamaan (3.11).

$$x_k = \begin{cases} \frac{x_k - \min(x_k)}{\max(x_k) - \min(x_k)} & ; \text{jika } \min(x_k) < 0 \\ \frac{x_k}{\max(x_k)} & ; \text{jika } \max(x_k) > 1 \end{cases} \quad (3.11)$$

- j) Evaluasi fungsi *fitness*-nya dan perbarui posisi terbaik kucing.
 k) Periksa kriteria pemberhentian. Jika iterasi telah mencapai iterasi maksimal, algoritma dihentikan dan posisi terbaik sebagai solusi akhir. Namun, jika iterasi belum mencapai iterasi maksimal, kembali ke langkah f).

2) Penerapan CSO

Pada langkah ini algoritma CSO disesuaikan dengan permasalahan *modified bounded knapsack multiple constraints*. Representasi solusi dari algoritma CSO pada permasalahan *modified bounded knapsack multiple constraints* adalah variabel keputusan jumlah barang yang diangkut dalam *knapsack*. Jumlah barang diangkut didapatkan dari perkalian posisi kucing dengan jumlah ketersediaan barang untuk setiap dimensi, kemudian dibulatkan. Selanjutnya, dilakukan pengecekan kendala sebelum dievaluasi nilai *fitness*-nya (total profit). Pengecekan kendala dilakukan agar setiap solusi memenuhi semua kendala yang ada, yang artinya solusi berada pada daerah *feasible*. Apabila solusi tidak memenuhi kendala, maka akan dilakukan penalti yaitu dengan mengurangi posisi dari kucing (solusi) secara acak pada dimensi yang dipilih secara acak pula. Proses penalti dilakukan hingga semua kendala terpenuhi.

3) Penerapan PSO

Pada langkah ini algoritma PSO disesuaikan dengan permasalahan *modified bounded knapsack multiple constraints*. Representasi solusi dan langkah evaluasi dilakukan seperti pada algoritma CSO yang telah diuraikan di atas.

d. Pembuatan Program

Pada langkah ini, penulis akan membuat program penerapan algoritma sebagai alat bantu penyelesaian permasalahan *modified bounded knapsack multiple constraints*. Program akan dibuat dalam bentuk *Graphical User Interface* (GUI) menggunakan *software* Matlab.

e. Simulasi Program

Pada langkah ini, penulis akan menjalankan program yang telah dibuat untuk menyelesaikan data. Proses simulasi diawali dengan menguji parameter dari algoritma untuk mengetahui pengaruh dan nilai terbaiknya pada permasalahan *modified bounded knapsack multiple constraints*. Setiap pengujian, program akan dijalankan sebanyak 10 kali. Parameter terbaik diambil berdasarkan rata-rata fungsi tujuan (total profit) yang didapatkan. Selanjutnya program akan dijalankan menggunakan parameter terbaik sebagai percobaan akhir.

f. Analisis Hasil

Hasil dari percobaan akhir algoritma CSO-PSO akan dibandingkan dengan hasil dari algoritma CSO dan PSO untuk mengetahui keberhasilan hasil algoritma CSO-PSO. Untuk melihat keoptimalan hasil algoritma CSO-PSO, hasil algoritma juga akan dibandingkan dengan hasil metode *Simplex* dengan menghitung persentase deviasi (*PD*) menggunakan Persamaan (3.12).

$$PD = \frac{Exact-Z}{Exact} \times 100\% \quad (3.12)$$

dengan

Exact : hasil metode *Simplex*

Z : hasil algoritma CSO-PSO

g. Kesimpulan

Pada langkah ini, hasil analisis yang telah dilakukan akan ditarik kesimpulan akhir dari penelitian penerapan algoritma CSO-PSO pada permasalahan *modified bounded knapsack multiple constraints*.

BAB 4. HASIL DAN PEMBAHASAN

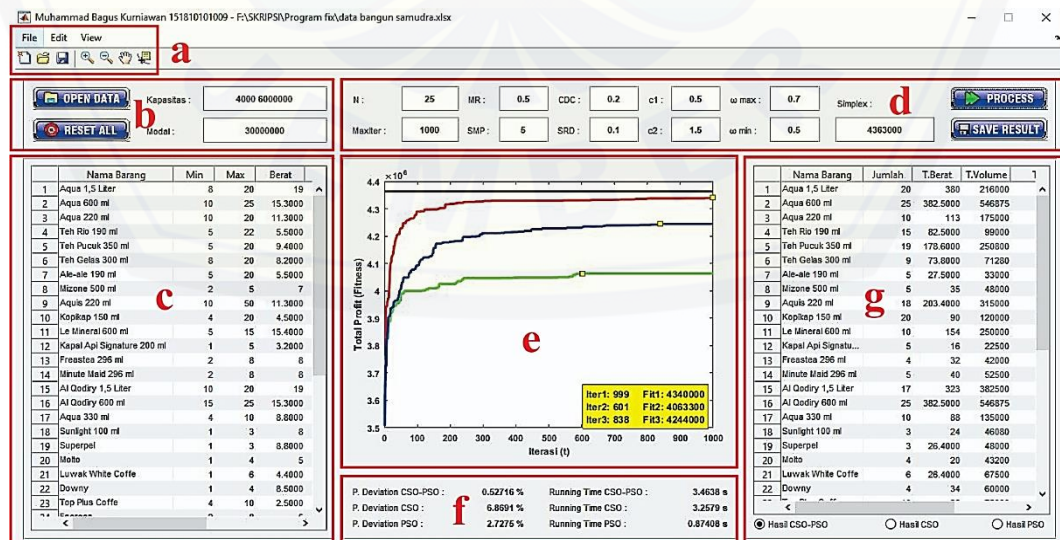
Pada bab ini akan diuraikan mengenai hasil penelitian dan pembahasannya. Hasil penelitian dimulai dengan penjelasan tentang program simulasi, kemudian dilanjutkan penjelasan tentang hasil simulasi. Pada bagian pembahasan akan dijelaskan tentang perhitungan manual, pengaruh parameter, perbandingan algoritma CSO-PSO dengan algoritma CSO dan PSO; serta hasil terbaik algoritma CSO-PSO.

4.1 Hasil Penelitian

Adapun hasil penelitian algoritma CSO-PSO yang sudah dilakukan sebagai berikut.

4.1.1 Program Simulasi

Pada penelitian ini, penulis membuat program simulasi algoritma CSO-PSO, CSO, dan PSO menggunakan *software* MATLAB dalam bentuk *Graphical User Interface* (GUI). Program tersebut digunakan untuk membantu menguji pengaruh parameter dan untuk melihat perbandingan performa algoritma CSO-PSO dengan algoritma CSO dan PSO pada penyelesaian permasalahan *modified bounded knapsack multiple constraints*. Pada Gambar 4.1 dapat dilihat tampilan program yang telah dibangun.



Gambar 4.1 Tampilan program simulasi

Program simulasi pada Gambar 4.1 dibagi menjadi tujuh bagian, antara lain sebagai berikut.

- a. Menubar dan Toolbar, berisi beberapa menu dan tool yang dapat digunakan untuk mengoperasikan program, seperti Open Data, Save Result, Reset, dan sebagainya.
- b. Input, berisi dua Tombol dan dua Edit, yaitu:
 - 1) Tombol Open Data, untuk membuka data yang akan diselesaikan. Data harus dalam format Microsoft Excel (*.xls, *.xlsx).
 - 2) Tombol Reset All, untuk mereset program ke tampilan awal.
 - 3) Edit Kapasitas, digunakan untuk mengisi kapasitas *knapsack* yang digunakan. Nilai yang dimasukkan harus ada dua yang dipisahkan dengan spasi. Nilai tersebut adalah kapasitas berat dan kapasitas volume.
 - 4) Edit Modal, digunakan untuk mengisi modal yang dimiliki pelaku usaha.
- c. Tabel Data, digunakan untuk menampilkan data yang akan diselesaikan.
- d. Utama, berisi sepuluh Edit Parameter, satu Edit *Simplex* dan dua Tombol, yaitu:
 - 1) Edit Parameter, digunakan untuk mengisi nilai-nilai parameter dari algoritma.
 - 2) Edit *Simplex*, digunakan untuk mengisi hasil *simplex* yang didapatkan dari Solver Add-In Microsoft Excel.
 - 3) Tombol Proses, untuk memulai perhitungan algoritma dalam menyelesaikan permasalahan *modified bounded knapsack multiple constraints*.
 - 4) Tombol Save Result, untuk menyimpan hasil perhitungan.
- e. Axes, untuk menampilkan grafik hasil terbaik algoritma setiap iterasi, menampilkan nilai iterasi Konvergen (iterasi dimana algoritma tidak menemukan solusi lebih baik), dan menampilkan solusi terbaik.
- f. Persentase Deviasi dan Waktu Komputasi, digunakan untuk menampilkan hasil perhitungan persentase deviasi dan waktu komputasi (*running time*) yang dibutuhkan algoritma.
- g. Tabel Output, untuk menampilkan hasil akhir dari algoritma.

4.1.2 Hasil Simulasi

Pada bagian ini, program simulasi dijalankan untuk melakukan perhitungan algoritma CSO-PSO, CSO, dan PSO dalam menyelesaikan permasalahan *modified bounded knapsack multiple constraints*. Program dijalankan pada Laptop dengan CPU AMD Ryzen 3 2200U with Radeon Vega Mobile Gfx @2.50GHz RAM 4GB. Semua parameter pada ketiga algoritma kecuali parameter *MR* dan *SRD*, diuji menggunakan beberapa nilai. Nilai *MR* yang digunakan pada penelitian ini adalah 0,5 dengan tujuan untuk menyeimbangkan banyak solusi yang masuk ke dalam *seeking mode* dan *tracing mode*. Nilai *SRD* yang digunakan relatif kecil yaitu 0,1 agar tidak keluar dari interval ruang pencarian permasalahan *modified bounded knapsack multiple constraints* [0, 1]. Pada beberapa uji parameter program dijalankan sebanyak sepuluh kali, sedangkan beberapa lainnya program dijalankan sebanyak lima kali. Setelah semua uji selesai dilakukan, program dijalankan kembali untuk simulasi akhir algoritma CSO-PSO menggunakan nilai nilai parameter terbaik. Setiap hasil uji, total profit yang didapatkan akan langsung dibandingkan dengan hasil metode *Simplex* untuk dihitung persentase desiasinya menggunakan Persamaan (3.12). Hasil optimum yang diperoleh metode *Simplex* memiliki total profit sebesar Rp 4.363.000,- (lihat Lampiran B). Secara lengkap uji parameter dan simulasi akhir sebagai berikut.

a. Uji parameter *Iterasi*

Nilai parameter *Iterasi* yang diujikan adalah 100, 200, 500, 1000, dan 2000. Nilai parameter lain yang digunakan antara lain, $N = 25$; $SMP = 1$; $CDC = 0,1$; $SRD = 0,1$; $c_1 = 2$; $c_2 = 2$; $\omega_{max} = 0,9$; dan $\omega_{min} = 0,1$. Dari setiap nilai parameter *Maxiter* yang digunakan, program simulasi dijalankan sebanyak sepuluh kali. Hasil yang didapatkan dari uji parameter *Iterasi* secara lengkap dapat dilihat pada Lampiran C.1 dan rata-rata dari sepuluh kali uji disajikan pada Tabel 4.1.

Tabel 4.1 Hasil uji parameter *Iterasi*

Algoritma	<i>Iterasi</i>	Rata-rata Profit (Rp)	Rata-Rata Deviasi (%)	Rata-rata Iterasi <i>Konvergen</i>	Rata-rata Waktu Komputasi (detik)
CSO-PSO	100	4.115.520	5,67225	94,7	0,23321
	200	4.169.460	4,43593	191,3	0,36717
	500	4.238.350	2,85698	480,3	0,77074
	1000	4.262.040	2,31401	958,2	1,56543
	2000	4.296.150	1,532202	1853,1	2,80446
	5000	4.303.850	1,355714	4611,1	7,94706
CSO	100	3.976.230	8,86478	81,6	0,1989
	200	4.003.190	8,24685	1611	0,30296
	500	4.020.020	7,8611	295,35	0,53874
	1000	4.050.230	7,16868	274,4	0,91929
	2000	4.055.270	7,05317	324,3	1,60127
	5000	4.064.970	6,83085	424	4,20707
PSO	100	4.010.600	8,07701	68,7	0,12537
	200	4.087.000	6,32592	121	0,19858
	500	4.147.650	4,93583	275,4	0,4779
	1000	4.232.160	2,99885	428,8	0,91988
	2000	4.258.980	2,38415	747,8	1,82069
	5000	4.280.450	1,892048	1298,3	4,96252

b. Uji parameter *SMP*

Nilai parameter *SMP* yang diujikan adalah 1, 5, dan 10. Nilai parameter lain yang digunakan antara lain, $N = 25$; $Iterasi = 1000$; $CDC = 0,1$; $SRD = 0,1$; $c_1 = 2$; $c_2 = 2$; $\omega_{max} = 0,9$; dan $\omega_{min} = 0,1$. Setiap nilai parameter *SMP*, program simulasi dijalankan sebanyak sepuluh kali. Hasil yang didapatkan dari uji parameter *SMP* secara lengkap dapat dilihat pada Lampiran C.2 dan rata-rata dari sepuluh kali uji disajikan pada Tabel 4.2.

Tabel 4.2 Hasil uji parameter *SMP*

Algoritma	<i>SMP</i>	Rata-rata Profit (Rp)	Rata-Rata Deviasi (%)	Rata-rata Iterasi <i>Konvergen</i>	Rata-rata Waktu Komputasi (detik)
CSO-PSO	1	4.262.040	2,31401	958,2	1,56543
	5	4.310.900	1,194118	902,6	3,75495
	10	4.326.850	0,828552	916,9	6,03177
CSO	1	4.050.230	7,16868	274,4	0,91929
	5	4.131.460	5,30689	906	3,74735
	10	4.177.460	4,25257	940,8	6,25278

c. Uji parameter *CDC*

Nilai parameter *CDC* yang diujikan adalah 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; dan 0,9. Nilai parameter lain yang digunakan antara lain, $N = 25$; $Iterasi =$

1000; $SMP = 5$; $SRD = 0,1$; $c_1 = 2$; $c_2 = 2$; $\omega_{max} = 0,9$; dan $\omega_{min} = 0,1$. Setiap nilai parameter CDC , program simulasi dijalankan sebanyak lima kali. Hasil yang didapatkan dari uji parameter CDC secara lengkap dapat dilihat pada Lampiran C.3 dan rata-rata dari lima kali uji disajikan pada Tabel 4.3.

Tabel 4.3 Hasil uji parameter CDC

Algoritma	CDC	Rata-rata Profit (Rp)	Rata-Rata Deviasi (%)	Rata-rata Iterasi $Kovergen$	Rata-rata Waktu Komputasi (detik)
CSO-PSO	0,1	4.284.500	1,7992	867	3,57272
	0,2	4.308.500	1,24914	974,4	3,46884
	0,3	4.279.400	1,9161	829,4	3,39502
	0,4	4.260.100	2,35846	850,6	3,62476
	0,5	4.243.400	2,74122	846,6	3,30268
	0,6	4.208.400	3,54342	918,6	3,17198
	0,7	4.176.600	4,2723	865	3,10554
	0,8	4.167.080	4,4905	895	3,1559
	0,9	4.119.580	5,5792	783,2	3,37388
CSO	0,1	4.075.700	6,5849	726,2	3,50808
	0,2	4.088.460	6,29246	931,4	3,6734
	0,3	4.063.300	6,86914	856,6	3,71982
	0,4	4.063.200	6,87142	852,6	4,0284
	0,5	4.033.440	7,55354	922	3,80138
	0,6	4.023.840	7,77356	576	3,5428
	0,7	4.014.000	7,99908	381	3,47784
	0,8	4.011.200	8,06326	590,4	3,52098
	0,9	3.971.180	8,9805	643,6	3,82446

d. Uji parameter c

Nilai parameter c yang diujikan adalah 0,1; 0,3; 0,5; 0,7; dan 0,9. Nilai parameter lain yang digunakan antara lain, $N = 25$; $Iterasi = 1000$; $SMP = 5$; $CDC = 0,2$; $SRD = 0,1$; $\omega_{max} = 0,9$; dan $\omega_{min} = 0,1$. Setiap nilai parameter c , program simulasi dijalankan sebanyak lima kali untuk algoritma CSO-PSO dan PSO, sedangkan untuk algoritma CSO sebanyak sepuluh kali. Hasil yang didapatkan dari uji parameter c secara lengkap dapat dilihat pada Lampiran C.4 dan rata-rata dari setiap kali uji disajikan pada Tabel 4.4 dan Tabel 4.5.

Tabel 4.4 Hasil uji parameter c algoritma CSO-PSO dan PSO

Algoritma	c_1	c_2	Rata-rata Profit (Rp)	Rata-Rata Deviasi (%)	Rata-rata Iterasi <i>Konvergen</i>	Rata-rata Waktu Komputasi (detik)
CSO-PSO	0,5	0,5	4.293.100	1,602104	682,4	3,91664
		1	4.310.000	1,21476	767	3,52676
		1,5	4.342.500	0,469852	863,8	3,62296
		2	4.335.300	0,634892	934,2	3,56792
	1	0,5	4.284.100	1,808396	559,4	3,48272
		1	4.307.600	1,269772	740	3,8333
		1,5	4.340.900	0,506532	856,6	3,47838
		2	4.337.800	0,577584	958,2	3,83734
	1,5	0,5	4.302.500	1,386676	658,2	3,65396
		1	4.324.300	0,887004	822,8	3,66976
		1,5	4.336.200	0,614256	963,6	3,96554
		2	4.334.500	0,65322	841,4	3,60428
	2	0,5	4.298.100	1,487504	723,6	3,3539
		1	4.315.700	1,084112	856,6	3,86942
		1,5	4.321.300	0,955768	905,6	4,1108
		2	4.302.300	1,391236	874,4	4,98782
PSO	0,5	0,5	4.103.660	5,94408	251,4	1,370345
		1	4.218.980	3,30092	699,4	1,012818
		1,5	4.292.500	1,61586	769,4	1,018972
		2	4.282.180	1,852392	955,2	1,05091
	1	0,5	4.178.340	4,2324	274	1,041776
		1	4.240.300	2,812292	529,2	1,09946
		1,5	4.280.800	1,88402	799,2	1,026646
		2	4.266.700	2,20718	904,2	1,11272
	1,5	0,5	4.182.440	4,13844	543,6	1,05816
		1	4.225.900	3,14232	645,4	1,05618
		1,5	4.283.500	1,822152	853	1,14154
		2	4.229.100	3,069	765,2	1,0125
	2	0,5	4.229.000	3,07128	758,8	0,97704
		1	4.249.300	2,606	841,8	1,086506
		1,5	4.250.800	2,57164	711,6	1,16078
		2	4.205.440	3,61128	420,4	1,32184

Tabel 4.5 Hasil uji parameter c algoritma CSO

Algoritma	c	Rata-rata Profit (Rp)	Rata-Rata Deviasi (%)	Rata-rata Iterasi <i>Kovergen</i>	Rata-rata Waktu Komputasi (detik)
CSO	0,5	4.139.940	5,11254	820,6	3,81702
	1	4.136.190	5,1985	765,7	3,84465
	1,5	4.107.500	5,85605	779,7	4,01645
	2	4.081.370	6,45495	854,2	4,3576

e. Uji parameter ω_{max} dan ω_{min}

Nilai parameter ω yang diujikan adalah 0,1; 0,3; 0,5; 0,7; dan 0,9. Nilai parameter lain yang digunakan antara lain, $N = 25$; $Iterasi = 1000$; $SMP = 5$; $CDC = 0,2$; $SRD = 0,1$; $c_1 = 0,5$; dan $c_2 = 1,5$. Setiap nilai parameter ω , program simulasi dijalankan sebanyak lima kali. Hasil yang didapatkan dari uji parameter ω secara lengkap dapat dilihat pada Lampiran C.5 dan rata-rata dari setiap kali uji disajikan pada Tabel 4.6.

Tabel 4.6 Hasil uji parameter ω_{max} dan ω_{min}

Algoritma	ω_{max}	ω_{min}	Rata-rata Profit (Rp)	Rata-Rata Deviasi (%)	Rata-rata Iterasi Konvergen	Rata-rata Waktu Komputasi (detik)	
CSO-PSO	0,9	0,7	4.339.000	0,55008	827,4	3,15904	
		0,5	4.334.600	0,650928	870,2	3,00742	
		0,3	4.327.900	0,804488	923,4	3,09024	
		0,1	4.311.600	1,178084	796,8	2,98266	
	0,7	0,5	4.338.300	0,566124	917,4	3,16736	
		0,3	4.332.000	0,710528	850,4	3,03778	
		0,1	4.319.900	0,98786	906,8	3,024	
		0,5	4.317.500	1,042864	678,8	2,99278	
	0,5	0,1	4.303.800	1,356872	758	3,02134	
		0,3	4.304.500	1,340816	775,6	3,2565	
	PSO	0,9	0,7	4.290.400	1,663996	919	0,867828
			0,5	4.269.800	2,13614	873,2	0,85378
0,3			4.251.600	2,55328	851,8	0,859	
0,1			4.241.000	2,79624	694,4	0,851176	
0,7		0,5	4.261.600	2,3241	904,2	0,874272	
		0,3	4.236.600	2,8971	810,2	0,877134	
		0,1	4.188.340	4,00322	883	0,844522	
		0,5	4.184.340	4,0949	958,4	0,847344	
0,5		0,1	4.083.780	6,39974	840,4	0,857236	
		0,3	3.732.820	14,4437	397	0,854766	

f. Uji parameter N

Nilai parameter N yang diujikan adalah 25, 50, 100, dan 200. Nilai parameter lain yang digunakan antara lain, $Iterasi = 25$; $SMP = 1$; $CDC = 0,2$; $SRD = 0,1$; $c_1 = 0,5$; $c_2 = 1,5$; $\omega_{max} = 0,9$; dan $\omega_{min} = 0,7$. Setiap nilai parameter N , program simulasi dijalankan sebanyak sepuluh kali. Hasil yang didapatkan dari uji parameter N secara lengkap dapat dilihat pada Lampiran C.6 dan rata-rata dari sepuluh kali uji disajikan pada Tabel 4.7.

Tabel 4.7 Hasil uji parameter N

Algoritma	N	Rata-rata Profit (Rp)	Rata-Rata Deviasi (%)	Rata-rata Iterasi <i>Konvergen</i>	Rata-rata Waktu Komputasi (detik)
CSO-PSO	25	4.304.100	1,34999	977,8	1,38088
	50	4.316.950	1,055466	939,7	2,31142
	100	4.333.400	0,678436	936,6	4,3312
	200	4.339.600	0,536334	925,3	8,05639
CSO	25	4.101.550	5,39243	696,6	1,079791
	50	4.085.450	6,36145	647,8	2,12014
	100	4.169.350	4,43847	727,3	3,61724
	200	4.204.420	3,63465	543,9	6,56325
PSO	25	4.245.800	2,686226	928,8	0,93806
	50	4.281.850	1,759952	959,1	1,768
	100	4.308.700	1,244544	887	3,60385
	200	4.323.950	0,89503	891,9	8,14069

g. Simulasi akhir

Pada simulasi akhir ini, digunakan nilai parameter terbaik untuk menguji algoritma CSO-PSO, CSO, dan PSO dalam penyelesaian permasalahan *modified bounded knapsack multiple constraints*. Program dijalankan sebanyak sepuluh kali. Hasil yang diperoleh dari program untuk masing-masing algoritma disajikan dalam Tabel 4.8 sampai Tabel 4.10.

Tabel 4.8 Simulasi akhir algoritma CSO-PSO

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi (%)	Iterasi <i>Konvergen</i>	Waktu Komputasi (detik)
1	3.998,3	5.982.325	29.990.000	4.362.500	0,01146	1938	271,3787
2	3.997,7	5.994.225	29.998.000	4.358.000	0,1146	3022	265,3057
3	3.996,7	5.996.925	29.987.000	4.360.500	0,0573	4386	266,4159
4	3.997,2	5.981.125	29.997.000	4.363.000	0	4421	269,1283
5	3.994,4	5.972.925	29.998.000	4.362.500	0,01146	4116	253,1336
6	3.999,7	5.985.275	29.976.000	4.359.500	0,08022	3632	250,6851
7	3.998,2	5.974.325	29.997.500	4.361.500	0,03438	821	254,59
8	3.998,7	5.998.375	29.991.500	4.358.000	0,1146	3017	252,7262
9	3.999,6	5.989.425	29.998.500	4.360.500	0,0573	3342	250,8685
10	3.999,3	5.981.125	29.995.000	4.361.500	0,03438	4966	247,7189
Rata-rata				4.360.750	0,05157	3366,1	258,19509

Tabel 4.9 Simulasi akhir algoritma CSO

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi (%)	Iterasi <i>Konvergen</i>	Waktu Komputasi (detik)
1	3.996,9	5.994.525	28.489.500	4.190.000	3,9652	1343	247,5591
2	3.999,4	5.985.270	29.992.500	4.232.000	3,0025	3006	261,9962
3	3.992,1	5.969.575	29.997.000	4.256.000	2,4524	2160	239,8044
4	3.973,7	5.984.480	29.989.000	4.282.500	1,8451	4824	252,7755

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi (%)	Iterasi <i>Konvergen</i>	Waktu Komputasi (detik)
5	3.990,2	5.995.840	29.919.500	4.272.000	2,0857	2446	244,5541
6	3.978	5.999.075	29.988.000	4.157.500	4,7101	400	238,8212
7	3.999,8	5.994.000	29.663.000	4.195.000	3,8506	1549	239,0895
8	3.992,4	5.996.165	29.962.000	4.131.000	5,3174	159	235,9981
9	3.996,9	5.960.095	29.996.000	4.279.500	1,9138	4024	237,3203
10	3.999,6	5.996.365	29.999.500	4.241.500	2,7848	560	228,6496
Rata-rata				4223700	3,19276	2047,1	242,6568

Tabel 4.10 Simulasi akhir algoritma PSO

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi (%)	Iterasi <i>Konvergen</i>	Waktu Komputasi (detik)
1	3.996,4	5.982.125	29.996.500	4.355.000	0,18336	2930	38,5621
2	3.999	5.993.605	29.999.000	4.343.500	0,44694	4733	39,995
3	3.999,3	5.984.345	29.985.000	4.340.500	0,5157	4595	39,2501
4	3.998,7	5.976.765	29.896.000	4.350.500	0,2865	4870	39,174
5	3.999,3	5.988.025	29.998.000	4.333.500	0,67614	4041	38,8627
6	3.997,7	5.989.045	29.996.500	4.352.000	0,25212	4635	37,9008
7	3.999,7	5.982.845	29.989.000	4.340.500	0,5157	4584	38,4197
8	3.993	5.992.805	29.974.500	4.346.000	0,38964	4681	38,2032
9	3.997,9	5.984.245	29.993.000	4.289.000	1,6961	2536	37,1348
10	3.973,4	5.999.195	29.981.000	4.333.500	0,67614	4442	36,7272
Rata-rata				4338400	0,563834	4204,7	38,42296

4.2 Pembahasan

4.2.1 Pseudocode Algoritma CSO-PSO

Input: Banyak jenis barang (D)
 Jumlah barang minimum (LB)
 Jumlah barang maksimum (UB)
 Berat barang (Wgt)
 Volume barang (Vol)
 Harga Beli (Buy)
 Harga Jual (Price)
 Kapasitas berat kendaraan (WCap)
 Kapasitas volume kendaraan (VCap)
 Modal (Modal)
 Populasi (N); Maksimal iterasi (Iterasi)
 Mixture Rasio (MR); Seeking Memory Pool (SMP);
 Counts of Dimension to Change (CDC);
 Seeking Ranger of Selected Dimension (SRD); c1; c2; Wmax;
 Wmin

```

{Pembangkitan solusi}
For k = 1 : N
  For d = 1 : D
    X[k,d] = rand(0,1)
    Y[k,d] = round(X[k,d] * (UB[d]-LB[d]))+LB[d]
  End
End

{Pemeriksaan kendala}
For k = 1 : N
  Berat[k] = 0
  Volume[k] = 0
  Beli[k] = 0
  For d = 1 : D
    Berat[k] = Berat[k] + Y[k,d]*Wgt[d]
    Volume[k] = Volume[k] + Y[k,d]*Vol[d]
    Beli[k] = Beli[k] + Y[k,d]*Buy[d]
  End
  While (Berat[k]>WCap) or (Volume[k]>VCap) or (Beli[k]>Modal)
    For d = 1 : D
      X[k,d] = rand(0,1)
      Y[k,d] = round(X[k,d] * (UB[d]-LB[d]))+LB[d]
    End
    Berat[k] = 0
    Volume[k] = 0
    Beli[k] = 0
    For d = 1 : D
      Berat[k] = Berat[k] + Y[k,d]*Wgt[d]
      Volume[k] = Volume[k] + Y[k,d]*Vol[d]
      Beli[k] = Beli[k] + Y[k,d]*Buy[d]
    End
  End
End

{Evaluasi Fitness (Total Profit)}
For k = 1 : N
  Z[k] = 0
  For d = 1 : D
    Z[k] = Z[k] + Y[k,d]*(Price[d]-Buy[d])
  End
End

{Posisi terbaik individu (Ck) & posisi terbaik global (Cg)}
For k = 1 : N
  For d = 1 : D
    Ck[k,d] = X[k,d]
  End
  ZCk[k] = Z[k]
End
Zmax = 0
For k = 1 : N
  If ZCk[k] > Zmax

```

```

    Zmax = ZCk[k]
    g = k
  End
End
For d = 1 : D
  Cg[d] = Ck[g,d]
End
ZCg = Zmax

{Iterasi}
t = 0
w = Wmax
While t < Iterasi
  {Pembangkitan flag}
  For k = 1 : N
    Flag[k] = rand(0,1)
  End

  {Pengurutan flag}
  For k = 1 : N
    Indeks[k] = i
  End
  For k = 1 : N-1
    For m = k+1 : N
      If Flag[k] > Flag[m]
        Temp = Flag[k]
        Flag[k] = Flag[m]
        Flag[m] = Temp
        Temp2 = Indeks[k]
        Indeks[k] = Indeks[m]
        Indeks[m] = Temp2
      End
    End
  End

  {Pembagian mode}
  Numtrac = ceil(MR*N)
  Numseek = N - Numtrac
  For k = 1 : Numseek
    Kseek[k] = Indeks[k]
  End
  For k = 1 : Numtrac
    Ktrac[k] = Indeks[Numseek+k]
  End

  {Seeking mode}
  For k = 1 : Numseek
    For m = 1 : SMP
      For d = 1 : D
        Copy[m,d] = Ck[Kseek[k],d]
      End
      JmlMut = ceil(CDC*D)
    End
  End
End

```

```

For j = 1 : JmlMut
    p = ceil(rand(0,1)*D)
    Copy[m,p] = Cg[p] + (2*rand(0,1)-1)*SRD*Cg[p]
End

{Transformasi}
Xmin = inf
Xmax = -inf
For d = 1 : D
    If Copy[m,d] < Xmin
        Xmin = Copy[m,d]
    End
    If Copy[m,d] > Xmax
        Xmax = Copy[m,d]
    End
End
If Xmin < 0
    For d = 1 : D
        Copy[m,d] = (Copy[m,d]-Xmin)/(Xmax-Xmin)
    End
ElseIf Xmax > 1
    For d = 1 : D
        Copy[m,d] = Copy[m,d]/Xmax
    End
End

For d = 1 : D
    YCopy[m,d] = round(Copy[m,d]*(UB[d]-LB[d]))+LB[d]
End

{Pemeriksaan kendala}
BeratC = 0
VolumeC = 0
BeliC = 0
For d = 1 : D
    BeratC = BeratC + YCopy[m,d]*Wgt[d]
    VolumeC = VolumeC + YCopy[m,d]*Vol[d]
    BeliC = BeliC + YCopy[m,d]*Buy[d]
End
While (BeratC>WCap) or (VolumeC>VCap) or (BeliC>Modal)
    p = ceil(rand(0,1)*D)
    Copy[m,p] = Copy[m,p] - 1/(UB[p]-LB[p])
    If Copy[m,p] < 0
        Copy[m,p] = (0.5*rand(0,1))/(UB[p]-LB[p])
    End
    YCopy[m,p] = ceil(Copy[m,p]*(UB[p]-LB[p]))+LB[p]
    BeratC = 0
    VolumeC = 0
    BeliC = 0
    For d = 1 : D
        BeratC = BeratC + YCopy[m,d]*Wgt[d]
        VolumeC = VolumeC + YCopy[m,d]*Vol[d]

```

```

        BeliC = BeliC + YCopy[m,d]*Buy[d]
    End
End

{Evaluasi Fitness (Total Profit)}
ZCopy[m] = 0
For d = 1 : D
    ZCopy[m] = ZCopy[m] + YCopy[m,d]*(Price[d]-Buy[d])
End
End

{Pemilihan}
Zmax = 0
For m = 1 : SMP
    If ZCopy[m] > Zmax
        Best = m
        Zmax = ZCopy[m]
    End
End
For d = 1 : D
    X[Kseek[k],d] = Copy[best,d]
    Y[Kseek[k],d] = YCopy[best,d]
End
Z[Kseek[k]] = ZCopy[best]
End

{Tracing Mode}
For k = 1 : Numtrac
    For d = 1 : D
        r1 = rand(0,1)
        r2 = rand(0,1)
        V[Ktrac[k],d] = w*V[Ktrac[k],d] + c1*r1*(Ck[Ktrac[k],d]-
            X[Ktrac[k],d]) + c2*r2*(Cg[Ktrac[k],d]-X[Ktrac[k],d])
        X[Ktrac[k],d] = X[Ktrac[k],d] + V[Ktrac[k],d]
    End
End

{Transformasi}
Xmin = inf
Xmax = -inf
For d = 1 : D
    If X[Ktrac[k],d] < Xmin
        Xmin = X[Ktrac[k],d]
    End
    If X[Ktrac[k],d] > Xmax
        Xmax = X[Ktrac[k],d]
    End
End
If Xmin < 0
    For d = 1 : D
        X[Ktrac[k],d] = (X[Ktrac[k],d]-Xmin)/(Xmax-Xmin)
    End
ElseIf Xmax > 1

```



```

    For d = 1 : D
        X[Ktrac[k],d] = X[Ktrac[k],d]/Xmax
    End
End

For d = 1 : D
    Y[Ktrac[k],d] = round(X[Ktrac[k],d]*(UB[d]-LB[d]))+LB[d]
End

{Pemeriksaan kendala}
Berat[Ktrac[k]] = 0
Volume[Ktrac[k]] = 0
Beli[Ktrac[k]] = 0
For d = 1 : D
    Berat[Ktrac[k]] = Berat[Ktrac[k]] + Y[Ktrac[k],d]*Wgt[d]
    Volume[Ktrac[k]] = Volume[Ktrac[k]] + Y[Ktrac[k],d]*Vol[d]
    Beli[Ktrac[k]] = Beli[Ktrac[k]] + Y[Ktrac[k],d]*Buy[d]
End
While (Berat[Ktrac[k]]>WCap) or (Volume[Ktrac[k]]>VCap) or
    (Beli[Ktrac[k]]>Modal)
    p = ceil(rand(0,1)*D)
    X[Ktrac[k],p] = X[Ktrac[k],p] - 1/(UB[p]-LB[p])
    If X[Ktrac[k],p] < 0
        X[Ktrac[k],p] = (0.5*rand(0,1))/(UB[p]-LB[p])
    End
    Y[Ktrac[k],p] = ceil(X[Ktrac[k],p]*(UB[p]-LB[p])+LB[p])
    Berat[Ktrac[k]] = 0
    Volume[Ktrac[k]] = 0
    Beli[Ktrac[k]] = 0
    For d = 1 : D
        Berat[Ktrac[k]] = Berat[Ktrac[k]] + Y[Ktrac[k],d]*Wgt[d]
        Volume[Ktrac[k]] = Volume[Ktrac[k]]+Y[Ktrac[k],d]*Vol[d]
        Beli[Ktrac[k]] = Beli[Ktrac[k]] + Y[Ktrac[k],d]*Buy[d]
    End
End

{Evaluasi Fitness (Total Profit)}
Z[Ktrac[k]] = 0
For d = 1 : D
    Z[Ktrac[k]] = Z[Ktrac[k]] + Y[Ktrac[k],d]*(Price[d]-Buy[d])
End
End

{Update Ck & Cg}
For k = 1 : N
    If Z[k] > ZCk[k]
        For d = 1 : D
            Ck[k,d] = X[k,d]
        End
        ZCk[k] = Z[k]
    End
End
End

```

```

Zmax = 0
For k = 1 : N
  If ZCk[k] > Zmax
    Zmax = ZCk[k]
    g = k
  End
End
For d = 1 : D
  Cg[d] = Ck[g,d]
End
ZCg = Zmax

{Update w}
w = w - (Wmax-Wmin)/Iterasi

t = t + 1
End

{Solusi akhir}
For d = 1 : D
  Solusi[d] = round(Cg[d]*(UB[d]-LB[d]) + LB[d])
End

```

4.2.2 Perhitungan Manual

Pada bagian ini, akan diberikan contoh langkah perhitungan Algoritma CSO-PSO pada permasalahan *modified bounded knapsack multiple constraints*. Data yang akan diselesaikan pada perhitungan manual ini (lihat Tabel 4.9) adalah data enam jenis barang yang diambil dari data utama penelitian ini (lihat Lampiran A). Kapasitas kendaraan (*knapsack*) yang digunakan pada perhitungan manual ini adalah 300 kg (berat) dan 640.000 cm³ (volume). Besar modal yang digunakan yaitu Rp 3.000.000,-.

Tabel 4.11 Data perhitungan manual

No	Nama Barang	LB	UB	Berat	Volume	H. Beli	H. Jual
1	Aqua 220 ml	10	20	11,3	17.500	28.000	30.000
2	Mizone 500 ml	2	5	7	9.600	38.000	42.000
3	Kopikap 150 ml	4	20	4,5	19.000	19.000	24.000
4	Luwak White Coffee	1	6	4,4	11.250	193.000	200.000
5	Sariwangi	1	2	2,25	26.250	240.000	280.000
6	Susu SGM	1	2	8,2	48.000	624.000	720.000

Data perhitungan manual dapat dikonversi ke dalam model matematis *modified bounded knapsack multiple constraints* sebagai berikut.

Fungsi tujuan:

$$\begin{aligned} \text{Maksimumkan } Z = & (30.000 - 28.000)y_1 + (42.000 - 38.000)y_2 + \\ & (24.000 - 19.000)y_3 + (200.000 - 193.000)y_4 + \\ & (280.000 - 240.000)y_5 + (720.000 - 624.000)y_6 \end{aligned} \quad (4.1)$$

Kendala:

$$11,3y_1 + 7y_2 + 4,5y_3 + 4,4y_4 + 2,25y_5 + 8,2y_6 \leq 300 \quad (4.2)$$

$$\begin{aligned} 17.500y_1 + 9.600y_2 + 19.000y_3 + 11.250y_4 + 26.250y_5 \\ + 48.000y_6 \leq 640.000 \end{aligned} \quad (4.3)$$

$$\begin{aligned} 28.000y_1 + 38.000y_2 + 19.000y_3 + 193.000y_4 + 240.000y_5 \\ + 624.000y_6 \leq 3.000.000 \end{aligned} \quad (4.4)$$

$$y_j \in \mathbb{Z}, \quad lb_j \leq y_j \leq ub_j, \quad j = 1, 2, \dots, 6 \quad (4.5)$$

Algoritma CSO-PSO memiliki beberapa parameter. Nilai yang digunakan pada perhitungan manual ini adalah $N = 4$, $MR = 0,5$, $SMP = 2$, $CDC = 0,3$, $SRD = 0,1$, $c_1 = 1$, $c_2 = 1$, $\omega_{max} = 0,9$, $\omega_{min} = 0,1$ dan $MaxIter = 1$. Langkah-langkah perhitungan algoritma CSO-PSO untuk menyelesaikan data pada Tabel 4.1 atau yang telah dikonversi dalam model matematis *modified bounded knapsack multiple constraints* pada Persamaan (4.1) sampai (4.5) adalah sebagai berikut.

a. Pembangkitan posisi kucing (X) dan kecepatan kucing (V)

Posisi kucing (X) dibangkitkan secara acak dalam bentuk matriks dengan ukuran $N \times D$, dengan D adalah banyak jenis barang. Nilai posisi kucing harus berada pada ruang pencarian. Dalam permasalahan *modified bounded knapsack multiple constraints* ini, ruang pencariannya berada pada interval $[0, 1]$. Setiap posisi kucing merepresentasikan persentase dari selisih jumlah barang maksimal dan minimal. Hasil pembangkitan posisi kucing sebagai berikut.

$X =$	0,7435	0,8838	0,2856	0,0735	0,8889	0,4656
	0,3785	0,4451	0,5937	0,9927	0,8641	0,3716
	0,4748	0,6682	0,3705	0,7760	0,6976	0,0830
	0,1063	0,7956	0,7956	0,0074	0,8850	0,9972

Selanjutnya, kecepatan kucing (V) dibangkitkan dalam bentuk matriks dengan ukuran yang sama dengan posisi kucing, namun nilainya diawali dengan nol, yang artinya kucing diawali pada posisi diam.

$$V = \begin{array}{|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

- b. Ubah posisi (X) ke bentuk solusi (Y)

Permasalahan *modified bounded knapsack multiple constraints* memiliki bentuk solusi ($y_{k,d}$) adalah banyak barang yang dipilih dengan syarat harus berada pada interval $[lb_d, ub_d]$. Oleh karena itu, posisi kucing (X) harus diubah ke bentuk solusi (Y) melalui perkalian posisi kucing ($x_{k,d}$) dengan selisih jumlah ketersediaan barang (ub_d) dan jumlah minimal barang yang harus dibeli (lb_d) untuk setiap dimensi, kemudian dibulatkan dan ditambahkan jumlah minimal. Langkah untuk mengubah posisi (X) ke bentuk solusi (Y) didasarkan pada Persamaan (3.1), sehingga diperoleh hasil sebagai berikut.

$$Y = \begin{array}{|c|c|c|c|c|c|} \hline 17 & 5 & 9 & 1 & 2 & 1 \\ \hline 14 & 3 & 13 & 6 & 2 & 1 \\ \hline 15 & 4 & 10 & 5 & 2 & 1 \\ \hline 11 & 4 & 17 & 1 & 2 & 2 \\ \hline \end{array}$$

- c. Periksa kendala dari setiap solusi

Setiap solusi dari algoritma CSO-PSO harus memenuhi kendala permasalahan *modified bounded knapsack multiple constraints* pada Persamaan (3.2) – (3.4). Hal ini dimaksudkan agar setiap solusi selalu berada pada daerah *feasible*.

$$\begin{aligned} \sum_{j=1}^D w_j y_{1,j} &= 11,3 \times 17 + 7 \times 5 + 4,5 \times 9 + 4,4 \times 1 + 2,25 \times 2 + 8,2 \times 1 \\ &= 284,7 \text{ kg (memenuhi)} \end{aligned}$$

$$\begin{aligned} \sum_{j=1}^D w_j y_{2,j} &= 11,3 \times 14 + 7 \times 3 + 4,5 \times 13 + 4,4 \times 6 + 2,25 \times 2 + 8,2 \times 1 \\ &= 276,8 \text{ kg (memenuhi)} \end{aligned}$$

$$\begin{aligned} \sum_{j=1}^D w_j y_{3,j} &= 11,3 \times 15 + 7 \times 4 + 4,5 \times 10 + 4,4 \times 5 + 2,25 \times 2 + 8,2 \times 1 \\ &= 284,7 \text{ kg (memenuhi)} \end{aligned}$$

$$\begin{aligned}\sum_{j=1}^D w_j y_{4,j} &= 11,3 \times 11 + 7 \times 4 + 4,5 \times 17 + 4,4 \times 1 + 2,25 \times 2 + 8,2 \times 2 \\ &= 254,1 \text{ kg (memenuhi)}\end{aligned}$$

$$\begin{aligned}\sum_{j=1}^D v_j y_{1,j} &= 17.500 \times 17 + 9.600 \times 5 + 19.000 \times 9 + 11.250 \times 1 + \\ &26.250 \times 2 + 48.000 \times 1 \\ &= 628.250 \text{ cm}^3 \text{ (memenuhi)}\end{aligned}$$

$$\begin{aligned}\sum_{j=1}^D v_j y_{2,j} &= 17.500 \times 14 + 9.600 \times 3 + 19.000 \times 13 + 11.250 \times 6 + \\ &26.250 \times 2 + 48.000 \times 1 \\ &= 688.800 \text{ cm}^3 \text{ (tidak memenuhi)}\end{aligned}$$

$$\begin{aligned}\sum_{j=1}^D v_j y_{3,j} &= 17.500 \times 15 + 9.600 \times 4 + 19.000 \times 10 + 11.250 \times 5 + \\ &26.250 \times 2 + 48.000 \times 1 \\ &= 647.650 \text{ cm}^3 \text{ (tidak memenuhi)}\end{aligned}$$

$$\begin{aligned}\sum_{j=1}^D v_j y_{4,j} &= 17.500 \times 11 + 9.600 \times 4 + 19.000 \times 17 + 11.250 \times 1 + \\ &26.250 \times 2 + 48.000 \times 2 \\ &= 713.650 \text{ cm}^3 \text{ (tidak memenuhi)}\end{aligned}$$

$$\begin{aligned}\sum_{j=1}^D b_j y_{1,j} &= 28.000 \times 17 + 38.000 \times 5 + 19.000 \times 9 + 193.000 \times 1 + \\ &240.000 \times 2 + 624.000 \times 1 \\ &= \text{Rp } 2.134.000 \text{ (memenuhi)}\end{aligned}$$

$$\begin{aligned}\sum_{j=1}^D b_j y_{2,j} &= 28.000 \times 14 + 38.000 \times 3 + 19.000 \times 13 + 193.000 \times 6 + \\ &240.000 \times 2 + 624.000 \times 1 \\ &= \text{Rp } 3.015.000 \text{ (tidak memenuhi)}\end{aligned}$$

$$\begin{aligned}\sum_{j=1}^D b_j y_{3,j} &= 28.000 \times 15 + 38.000 \times 4 + 19.000 \times 10 + 193.000 \times 5 + \\ &240.000 \times 2 + 624.000 \times 1 \\ &= \text{Rp } 2.831.000 \text{ (memenuhi)}\end{aligned}$$

$$\begin{aligned}\sum_{j=1}^D b_j y_{4,j} &= 28.000 \times 11 + 38.000 \times 4 + 19.000 \times 17 + 193.000 \times 1 + \\ &240.000 \times 2 + 624.000 \times 2 \\ &= \text{Rp } 2.704.000 \text{ (memenuhi)}\end{aligned}$$

Dari perhitungan total berat, volume dan harga beli barang yang dipilih, terlihat bahwa solusi 2, 3 dan 4 tidak memenuhi kendala kapasitas ruang (S), serta solusi 2 tidak memenuhi kendala modal (M). Oleh karena itu, ketiga solusi tersebut

harus dipenalti. Namun, karena solusi tersebut masih merupakan solusi awal, maka dapat diganti dengan solusi baru dengan pembangkitan ulang posisi awal hingga diperoleh solusi yang memenuhi. Misalkan hasil pembangkitan baru adalah sebagai berikut.

$$X = \begin{array}{cccccc} \hline 0,7435 & 0,8838 & 0,2856 & 0,0735 & 0,8889 & 0,4656 \\ \hline 0,1785 & 0,1118 & 0,5937 & 0,7927 & 0,8641 & 0,3716 \\ \hline 0,4748 & 0,6682 & 0,3080 & 0,7760 & 0,6976 & 0,0830 \\ \hline 0,0063 & 0,7956 & 0,6706 & 0,0074 & 0,1024 & 0,9972 \\ \hline \end{array}$$

Selanjutnya posisi tersebut dikonversi ke dalam bentuk solusi.

$$Y = \begin{array}{cccccc} \hline 17 & 5 & 9 & 1 & 2 & 1 \\ \hline 12 & 2 & 13 & 5 & 2 & 1 \\ \hline 15 & 4 & 9 & 5 & 2 & 1 \\ \hline 10 & 4 & 15 & 1 & 1 & 2 \\ \hline \end{array}$$

Kemudian dari setiap kandidat solusi, dihitung kembali total berat, volume dan harga belinya.

Solusi	Total Berat (kg)	Total Volume (cm ³)	Total Harga Beli (Rp)
1	284,7	628.250	2.134.000
2	242,8	632.950	2.728.000
3	272,7	628.650	2.812.000
4	231,55	631.900	2.398.000

Dari hasil perhitungan, terlihat bahwa semua solusi sudah memenuhi kendala berat, volume dan harga beli.

d. Evaluasi fungsi *fitness* (total profit)

Setelah semua solusi memenuhi kendala yang ada, selanjutnya dievaluasi fungsi *fitness* (total profit) dari setiap solusi. Nilai *fitness* dihitung menggunakan Persamaan (3.7).

$$\begin{aligned} Z_1 &= (30.000 - 28.000) \times 17 + (42.000 - 38.000) \times 5 + (24.000 - \\ &\quad 19.000) \times 9 + (200.000 - 193.000) \times 1 + (280.000 - 240.000) \times \\ &\quad 2 + (720.000 - 624.000) \times 1 \\ &= \text{Rp } 282.000 \end{aligned}$$

$$\begin{aligned}
 Z_2 &= (30.000 - 28.000) \times 12 + (42.000 - 38.000) \times 2 + (24.000 - \\
 &\quad 19.000) \times 13 + (200.000 - 193.000) \times 5 + (280.000 - 240.000) \times \\
 &\quad 2 + (720.000 - 624.000) \times 1 \\
 &= \text{Rp } 308.000
 \end{aligned}$$

$$\begin{aligned}
 Z_3 &= (30.000 - 28.000) \times 15 + (42.000 - 38.000) \times 4 + (24.000 - \\
 &\quad 19.000) \times 9 + (200.000 - 193.000) \times 5 + (280.000 - 240.000) \times \\
 &\quad 2 + (720.000 - 624.000) \times 1 \\
 &= \text{Rp } 302.000
 \end{aligned}$$

$$\begin{aligned}
 Z_4 &= (30.000 - 28.000) \times 10 + (42.000 - 38.000) \times 4 + (24.000 - \\
 &\quad 19.000) \times 15 + (200.000 - 193.000) \times 1 + (280.000 - 240.000) \times \\
 &\quad 1 + (720.000 - 624.000) \times 2 \\
 &= \text{Rp } 350.000
 \end{aligned}$$

- e. Simpan posisi terbaik dari setiap kucing sebagai C_k dan simpan posisi terbaik global sebagai C_g .

Posisi terbaik dari setiap kucing (C_k) pada iterasi awal sama dengan posisi awal kucing (X). Dengan demikian, posisi terbaik dari setiap kucing (C_k) sebagai berikut.

$C_k =$	0,7435	0,8838	0,2856	0,0735	0,8889	0,4656
	0,1785	0,1118	0,5937	0,7927	0,8641	0,3716
	0,4748	0,6682	0,3080	0,7760	0,6976	0,0830
	0,0063	0,7956	0,6706	0,0074	0,1024	0,9972

$Z_{C_k} =$	Rp 282.000
	Rp 308.000
	Rp 302.000
	Rp 350.000

Posisi terbaik global (C_g) diambil dari C_k yang memiliki nilai *fitness* (total profit) paling besar. Oleh karena itu, C_g diambil dari C_3 dengan total profit adalah Rp 350.000,-.

$C_g =$	0,0063	0,7956	0,6706	0,0074	0,1024	0,9972
---------	--------	--------	--------	--------	--------	--------

- f. Bangkitkan *flag* secara random, kemudian bagi kucing ke dalam *seeking mode* dan *tracing mode* sesuai *flag*.

Pembangkitan *flag* dilakukan secara acak untuk masing-masing solusi dengan nilai pada interval $[0, 1]$. Nilai *flag* ini digunakan untuk menentukan solusi yang masuk *seeking mode* dan *tracing mode*. Jumlah pembagian ditentukan berdasarkan parameter *mix ratio* (*MR*). Banyaknya solusi yang masuk *tracing mode* (*num_trac*) didapat dari perkalian *MR* dan *Npop*, kemudian dibulatkan ke atas. Banyaknya solusi yang masuk *seeking mode* (*num_seek*) sama dengan *N* dikurangi banyak solusi yang masuk *tracing mode*. Sejumlah *num_seek* solusi dengan nilai *flag* terkecil dimasukkan ke dalam *seeking mode* terlebih dahulu, kemudian sisanya dimasukkan ke dalam *tracing mode*.

Solusi	<i>flag</i>	<i>mode</i>
1	0,9986	<i>tracing</i>
2	0,7563	<i>seeking</i>
3	0,8765	<i>tracing</i>
4	0,4896	<i>seeking</i>

- g. *Seeking mode*

Setiap kandidat posisi baru (*Copy*) dibangkitkan dari posisi terbaik setiap kucing C_k , kemudian dimutasi beberapa dimensinya berdasarkan posisi terbaik global C_g dengan rumus mutasi pada Persamaan (3.8). Banyak kandidat yang dibangkitkan tergantung pada parameter *SMP* dan *SPC*. Jika *SPC* bernilai benar, artinya solusi merupakan yang terbaik dalam populasi, maka banyak solusi yang dibangkitkan adalah $SMP - 1$. Namun, jika *SPC* bernilai salah atau solusi bukan yang terbaik, maka banyaknya kandidat baru adalah *SMP*. Banyak dimensi yang akan dimutasi (*num_dim*) adalah *CDC* dikalikan *D*, kemudian dibulatkan ke atas. Dimensi yang dimutasi dipilih secara acak.

$$\begin{aligned}
 num_dim &= [CDC \times D] \\
 &= [0,3 \times 6] \\
 &= 2
 \end{aligned}$$

Untuk solusi 2

$Copy_1 =$	0,1785	0,1118	0,5937	0,7927	0,8641	0,3716
$Copy_2 =$	0,1785	0,1118	0,5937	0,7927	0,8641	0,3716

Misalkan dipilih dimensi 2 dan 6 untuk $Copy_1$, dan dimensi 4 dan 6 untuk $Copy_2$. Selanjutnya dimensi tersebut dimutasi menggunakan rumus pada Persamaan (3.8) dan diperoleh hasil sebagai berikut.

$Copy_1 =$	0,1785	0,8743	0,5937	0,7927	0,8641	0,8987
$Copy_2 =$	0,1785	0,1118	0,5937	0,0080	0,8641	0,9809

Kemudian kedua kandidat posisi baru dikonversi ke bentuk solusi dan dievaluasi kendala serta nilai *fitness*-nya.

$YCopy_1 =$	12	5	13	5	2	2
$YCopy_2 =$	12	2	13	1	2	2

$Copy$	Total Berat (kg)	Total Volume (cm ³)	Total Harga Beli (Rp)
1	272	709.750	3.466.000
2	233,4	635.950	2.580.000

Dari hasil perhitungan total berat, volume dan harga beli, terlihat bahwa $Copy_1$ tidak memenuhi kendala kapasitas ruang dan kendala modal. Oleh karena itu, $Copy_1$ harus dipenalti, artinya harus ada barang yang dikurangi hingga memenuhi semua kendala. Dalam hal ini, proses pengurangan dilakukan terlebih dahulu terhadap posisi kucing ($Copy_1$), kemudian solusi ($YCopy_1$) menyesuaikan. Rumus pengurangan yang digunakan dituliskan pada Persamaan (3.5). Misalkan dipilih dimensi 4, maka didapatkan posisi baru sebagai berikut.

$Copy_1 =$	0,1785	0,8743	0,5937	0,5927	0,8641	0,8987
$YCopy_1 =$	12	5	13	4	2	2

Selanjutnya akan dihitung kembali total berat, volume dan harga beli. Hasil perhitungan yang didapatkan sebagai berikut.

Total Berat (kg)	Total Volume (cm ³)	Total Harga Beli (Rp)
267,6	698.500	3.273.000

Dari hasil perhitungan total berat, volume dan harga beli, terlihat bahwa $Copy_1$ masih tidak memenuhi kendala kapasitas ruang dan kendala modal. Dengan demikian, proses penalti dilakukan kembali. Misalkan dipilih dimensi 2, maka didapatkan posisi baru sebagai berikut.

$Copy_1 =$	0,1785	0,5410	0,5937	0,5927	0,8641	0,8987
$YCopy_1 =$	12	4	13	4	2	2

Selanjutnya akan dihitung kembali total berat, volume dan harga beli. Hasil perhitungan yang didapatkan sebagai berikut.

Total Berat (kg)	Total Volume (cm ³)	Total Harga Beli (Rp)
260,6	688.900	3.235.000

Dari hasil perhitungan total berat, volume dan harga beli, terlihat bahwa $Copy_1$ masih tidak memenuhi kendala kapasitas ruang dan kendala modal. Dengan demikian, proses penalti masih harus dilakukan kembali. Proses penalti akan berhenti jika semua kendala telah dipenuhi. Dengan cara penalti yang sama, diperoleh hasil sebagai berikut.

$Copy_1 =$	0,0185	0,5410	0,5312	0,3927	0,8641	0,8987
$YCopy_1 =$	10	4	12	3	2	2

Selanjutnya akan dihitung kembali total berat, volume dan harga beli. Hasil perhitungan yang didapatkan sebagai berikut.

Total Berat (kg)	Total Volume (cm ³)	Total Harga Beli (Rp)
229,1	623.650	2.967.000

Dari hasil perhitungan total berat, volume dan harga beli, terlihat bahwa $Copy_1$ sudah memenuhi semua kendala. Dengan demikian, proses perhitungan dilanjutkan ke tahap evaluasi nilai *fitness*.

$$Z_{Copy_1} = \underline{\underline{\text{Rp } 389.000}}$$

$$Z_{Copy_2} = \underline{\underline{\text{Rp } 376.000}}$$

Berdasarkan hasil perhitungan nilai *fitness* (total profit), $Copy_1$ memiliki nilai yang lebih besar. Dengan demikian, kandidat posisi baru $Copy_1$ menggantikan posisi x_2 .

$$x_2 = \begin{array}{|c|c|c|c|c|c|} \hline 0,0185 & 0,5410 & 0,5312 & 0,3927 & 0,8641 & 0,8987 \\ \hline \end{array}$$

Untuk solusi 4

$$\begin{array}{|c|c|c|c|c|c|} \hline Copy_1 = & 0,0063 & 0,7956 & 0,6706 & 0,0074 & 0,1024 & 0,9972 \\ Copy_2 = & 0,0063 & 0,7956 & 0,6706 & 0,0074 & 0,1024 & 0,9972 \\ \hline \end{array}$$

Misalkan dipilih dimensi 2 dan 3 untuk $Copy_1$, dan dimensi 4 dan 6 untuk $Copy_2$. Selanjutnya dimensi tersebut dimutasi menggunakan rumus pada Persamaan (3.8) dan diperoleh hasil sebagai berikut.

$$\begin{array}{|c|c|c|c|c|c|} \hline Copy_1 = & 0,0063 & 0,7173 & 0,7293 & 0,0074 & 0,1024 & 0,9972 \\ Copy_2 = & 0,0063 & 0,7956 & 0,6044 & 0,0074 & 0,1114 & 0,9972 \\ \hline \end{array}$$

Kemudian kedua kandidat posisi baru dikonversi ke bentuk solusi dan dievaluasi kendala serta nilai *fitness*-nya.

$$\begin{array}{|c|c|c|c|c|c|} \hline YCopy_1 = & 10 & 4 & 16 & 1 & 1 & 2 \\ YCopy_2 = & 10 & 4 & 14 & 1 & 1 & 2 \\ \hline \end{array}$$

<i>Copy</i>	Total Berat (kg)	Total Volume (cm ³)	Total Harga Beli (Rp)
1	236,05	650.900	2.417.000
2	227,05	612.900	2.379.000

Dari hasil perhitungan total berat, volume dan harga beli, terlihat bahwa $Copy_1$ tidak memenuhi kendala kapasitas ruang. Oleh karena itu, $Copy_1$ harus dipenalti. Dengan cara penalti yang sama seperti sebelumnya, diperoleh hasil sebagai berikut.

$$\begin{array}{|c|c|c|c|c|c|} \hline Copy_1 = & 0,0063 & 0,7173 & 0,6668 & 0,0074 & 0,1024 & 0,9972 \\ Copy_2 = & 0,0063 & 0,7956 & 0,6044 & 0,0074 & 0,1114 & 0,9972 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|c|c|} \hline YCopy_1 = & 10 & 4 & 15 & 1 & 1 & 2 \\ YCopy_2 = & 10 & 4 & 14 & 1 & 1 & 2 \\ \hline \end{array}$$

<i>Copy</i>	Total Berat (kg)	Total Volume (cm ³)	Total Harga Beli (Rp)
1	231,55	631.900	2.398.000
2	233,4	635.950	2.580.000

Dari hasil perhitungan total berat, volume dan harga beli, terlihat bahwa *Copy*₁ sudah memenuhi semua kendala. Dengan demikian, proses perhitungan dilanjutkan ke tahap evaluasi nilai *fitness*.

$$Z_{Copy_1} = \underline{\text{Rp } 350.000}$$

$$Z_{Copy_2} = \underline{\text{Rp } 345.000}$$

Berdasarkan hasil perhitungan nilai *fitness* (total profit), *Copy*₁ memiliki nilai yang lebih besar. Dengan demikian, kandidat *Copy*₁ menggantikan posisi x_4 .

$$x_4 = \underline{\begin{array}{cccccc} 0,0063 & 0,7173 & 0,6668 & 0,0074 & 0,1024 & 0,9972 \end{array}}$$

h. *Tracing mode*

Solusi (kucing) yang masuk proses *tracing mode* adalah 1 dan 3. Kecepatan kedua kucing tersebut diperbarui menggunakan Persamaan (3.9). Selanjutnya, untuk memperbarui posisi kucing digunakan Persamaan (3.10). Nilai ω awal sama dengan ω_{max} yaitu 0,9.

Untuk solusi x_1

$$\begin{array}{l} x_1 = \underline{\begin{array}{cccccc} 0,7435 & 0,8838 & 0,2856 & 0,0735 & 0,8889 & 0,4656 \end{array}} \\ v_1 = \underline{\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \end{array}} \\ C_g = \underline{\begin{array}{cccccc} 0,0063 & 0,7956 & 0,6706 & 0,0074 & 0,1024 & 0,9972 \end{array}} \\ C_{k2} = \underline{\begin{array}{cccccc} 0,7435 & 0,8838 & 0,2856 & 0,0735 & 0,8889 & 0,4656 \end{array}} \\ r_1 = \underline{\begin{array}{cccccc} 0,8886 & 0,5436 & 0,7654 & 0,3421 & 0,9768 & 0,9867 \end{array}} \\ r_2 = \underline{\begin{array}{cccccc} 0,2341 & 0,3456 & 0,9878 & 0,9878 & 0,6453 & 0,9768 \end{array}} \end{array}$$

Dengan menggunakan Persamaan (3.9), diperoleh kecepatan baru untuk solusi 1 sebagai berikut.

$$v_1 = \underline{\begin{array}{cccccc} -0,6551 & -0,0479 & 0,2947 & -0,0226 & -0,7683 & 0,5245 \end{array}}$$

Selanjutnya, dengan menggunakan Persamaan (3.10), didapatkan posisi baru untuk solusi 1 sebagai berikut.

$$x_1 = \underline{\begin{array}{cccccc} 0,0884 & 0,8359 & 0,5803 & 0,0509 & 0,1206 & 0,9901 \end{array}}$$

Untuk solusi x_3

$x_3=$	0,4748	0,6682	0,3080	0,7760	0,6976	0,0830
$v_3=$	0	0	0	0	0	0
$C_g=$	0,0063	0,7956	0,6706	0,0074	0,1024	0,9972
$C_{k3}=$	0,4748	0,6682	0,3080	0,7760	0,6976	0,0830
$r_1=$	0,9987	0,8767	0,8678	0,5643	0,6434	0,9867
$r_2=$	0,8678	0,8976	0,9867	0,9876	0,9853	0,5364

Dengan menggunakan Persamaan (3.9), diperoleh kecepatan baru untuk solusi 1 sebagai berikut.

$v_3=$	-0,4679	0,1117	0,3147	-0,4337	-0,3830	0,9020
--------	---------	--------	--------	---------	---------	--------

Selanjutnya, dengan menggunakan Persamaan (3.10), didapatkan posisi baru untuk solusi 1 sebagai berikut.

$x_3=$	0,0069	0,7799	0,6227	0,3423	0,3146	0,9850
--------	--------	--------	--------	--------	--------	--------

i. Penggabungan kembali hasil *seeking mode* dan *tracing mode*

Pada langkah ini, hasil perpindahan posisi di *seeking mode* dan *tracing mode* digabungkan kembali.

$X=$	0,0884	0,8359	0,5803	0,0509	0,1206	0,9901
	0,0185	0,5410	0,5312	0,3927	0,8641	0,8987
	0,0069	0,7799	0,6227	0,3423	0,3146	0,9850
	0,0063	0,7173	0,6668	0,0074	0,1024	0,9972
$Y=$	11	5	13	1	1	2
	10	4	12	3	2	2
	10	4	14	3	1	2
	10	4	15	1	1	2

j. Evaluasi fungsi *fitness* dan perbarui posisi terbaik kucing

Pada tahap ini, setiap solusi dievaluasi nilai *fitness*-nya. Namun, terlebih dahulu harus dipastikan solusi-solusi tersebut harus memenuhi kendala berat, ruang dan modal, sehingga solusi berada pada daerah *feasible*.

Solusi	Total Berat (kg)	Total Volume (cm ³)	Total Harga Beli (Rp)
1	240,85	621.000	2.426.000
2	229,1	623.650	2.967.000

3	235,85	635.400	2.765.000
4	231,55	631.900	2.398.000

Berdasarkan hasil perhitungan, terlihat bahwa semua solusi telah memenuhi kendala yang ada. Dengan demikian, langkah perhitungan dilanjutkan dengan menghitung nilai *fitness* (total profit).

$$Z = \begin{array}{r} \hline \text{Rp } 346.000 \\ \hline \text{Rp } 389.000 \\ \hline \text{Rp } 359.000 \\ \hline \text{Rp } 350.000 \\ \hline \end{array}$$

Selanjutnya, posisi terbaik setiap kucing (C_k) dan posisi terbaik global (C_g) diperbarui. Jika posisi baru setiap kucing lebih baik daripada posisi terbaik kucing tersebut, maka posisi baru menggantikan posisi terbaik. Namun, jika posisi baru tidak lebih baik, maka posisi terbaik sebelum dipertahankan.

$$C_k = \begin{array}{r} \hline 0,0884 \quad 0,8359 \quad 0,5803 \quad 0,0509 \quad 0,1206 \quad 0,9901 \\ \hline 0,0185 \quad 0,5410 \quad 0,5312 \quad 0,3927 \quad 0,8641 \quad 0,8987 \\ \hline 0,0069 \quad 0,7799 \quad 0,6227 \quad 0,3423 \quad 0,3146 \quad 0,9850 \\ \hline 0,0063 \quad 0,7956 \quad 0,6706 \quad 0,0074 \quad 0,1024 \quad 0,9972 \\ \hline \end{array}$$

$$Z_{C_k} = \begin{array}{r} \hline \text{Rp } 346.000 \\ \hline \text{Rp } 389.000 \\ \hline \text{Rp } 359.000 \\ \hline \text{Rp } 350.000 \\ \hline \end{array}$$

Setelah posisi terbaik setiap kucing diperbarui, selanjutnya diambil posisi paling baik untuk disimpan sebagai posisi terbaik global (C_g). Dari hasil tersebut, terlihat posisi 2 memiliki total profit terbesar yaitu Rp 389.000, sehingga posisi tersebut digunakan untuk memperbarui C_g .

$$C_g = \begin{array}{r} \hline 0,0185 \quad 0,5410 \quad 0,5312 \quad 0,3927 \quad 0,8641 \quad 0,8987 \\ \hline \end{array}$$

k. Pemeriksaan kriteria pemberhentian

Jika iterasi telah mencapai iterasi maksimal, algoritma dihentikan dan posisi terbaik global digunakan untuk solusi akhir. Karena iterasi maksimal yang

digunakan pada perhitungan manual ini adalah 1, maka proses perhitungan dihentikan.

$$\text{Solusi} = \begin{array}{cccccc} \hline & 10 & 4 & 12 & 3 & 2 & 2 \\ \hline \end{array}$$

Solusi tersebut memiliki arti bahwa jumlah barang yang dipilih yaitu 10 Aqua 220ml, 4 Mizone 500ml, 12 Kopikap 150ml, 3 Luwak White Coffee, 2 Sariwangi dan 2 Susu SGM. Total profit yang akan diperoleh sebesar Rp 389.000,-.

4.2.3 Pembahasan Hasil Simulasi

Berdasarkan hasil uji parameter *Iterasi* (lihat Tabel 4.1), dapat diketahui bahwa semakin besar jumlah iterasi atau batas maksimal iterasi yang digunakan, maka rata-rata profit semakin besar dan rata-rata persentase deviasi semakin kecil. Dalam tugas akhir ini, digunakan istilah iterasi konvergen. Istilah ini memiliki arti yaitu iterasi yang menandakan algoritma tidak mampu menemukan solusi lebih baik hingga iterasinya mencapai batas maksimal yang ditentukan. Berdasarkan rata-rata iterasi konvergen, dapat diketahui bahwa algoritma CSO-PSO mampu menghindari terjebak lokal optimum meskipun nilai parameter N dan SMP kecil. Hal ini terlihat dari nilai rata-rata tersebut selalu mendekati batas iterasinya, berbeda dengan algoritma CSO dan PSO yang rata-rata iterasi konvergensinya relatif kecil dan jauh dari batas maksimal. Nilai terbaik *Iterasi* yang diujikan adalah 5000.

Dari hasil uji parameter SMP (lihat Tabel 4.2), dapat diketahui bahwa semakin besar nilai SMP , maka semakin besar pula rata-rata profitnya dan semakin kecil rata-rata persentase deviasinya. Selain itu, pada algoritma CSO, terlihat rata-rata iterasi konvergensinya mendekati nilai *Iterasi* yang digunakan, yang artinya nilai parameter SMP yang semakin besar mampu menghindarkan algoritma dari terjebak lokal optimum. Hal tersebut sesuai dengan fungsi parameter SMP , yaitu mengatur banyaknya kandidat solusi baru pada proses *seeking mode*. Semakin banyak kandidat solusi baru yang dibangkitkan, maka semakin besar pula peluang mendapatkan solusi yang lebih baik, sehingga hasil yang didapatkan semakin

optimal dan algoritma dapat keluar dari lokal optimum. Nilai terbaik SMP yang diujikan adalah 10.

Parameter CDC digunakan untuk mengontrol banyaknya dimensi yang dimutasi pada proses *seeking mode* algoritma CSO-PSO dan CSO. Berdasarkan hasil uji parameter ini yang disajikan pada Tabel 4.3, dapat dilihat bahwa nilai CDC terbaik yang digunakan untuk menyelesaikan data penelitian ini adalah $CDC = 0,2$. Nilai tersebut memiliki arti yaitu banyak dimensi yang dimutasi adalah 8. Pengaruh perubahan nilai CDC terhadap hasil adalah semakin kecil atau semakin besar nilai CDC , maka solusi yang didapatkan semakin buruk. Hal ini dikarenakan semakin kecil nilai CDC maka semakin sedikit dimensi yang dimutasi sehingga dimungkinkan kandidat solusi baru sama dengan solusi sebelum. Semakin besar nilai CDC maka semakin banyak dimensi yang dimutasi, sehingga kandidat solusi bar uterus teracak dan tidak dapat konvergen. Oleh karena itu, solusi sulit mendapatkan kandidat yang lebih baik.

Parameter c berfungsi untuk mengatur besar kecil pengaruh posisi terbaik terhadap posisi sebelum pada proses perpindahan. Pada algoritma CSO-PSO dan PSO, terdapat dua parameter c yaitu c_1 untuk posisi terbaik global dan c_2 untuk posisi terbaik individu, sedangkan pada algoritma CSO hanya terdapat satu parameter c untuk posisi terbaik global. Berdasarkan hasil uji parameter c_1 dan c_2 algoritma CSO-PSO dan PSO (lihat Tabel 4.4), dari kombinasi nilai yang diujikan terlihat bahwa untuk setiap nilai c_1 , solusi terbaik selalu diperoleh dengan nilai $c_2 = 1,5$. Dari hasil uji, dapat dilihat bahwa nilai c_1 yang menghasilkan solusi terbaik adalah $c_1 = 0,5$. Dengan demikian, dapat diketahui bahwa meskipun nilai c_1 kecil, posisi terbaik global sudah cukup berpengaruh, sedangkan posisi terbaik individu membutuhkan nilai c_2 yang lebih besar namun tidak terlalu besar. Selanjutnya, berdasarkan hasil uji parameter c algoritma CSO (lihat Tabel 4.5), dapat diketahui bahwa pengaruh parameter c algoritma CSO terhadap hasil sama dengan pengaruh parameter c_1 algoritma CSO-PSO dan PSO.

Berdasarkan hasil uji parameter ω_{max} dan ω_{min} dengan kombinasi nilai yang digunakan (lihat Tabel 4.6), dapat diketahui bahwa untuk setiap nilai ω_{max} , nilai ω_{min} yang menghasilkan solusi semakin optimal adalah ω_{min} yang semakin besar.

Dari hasil-hasil tersebut kombinasi terbaik yang didapatkan adalah $\omega_{max}=0,9$ dan $\omega_{min}=0,7$ baik algoritma CSO-PSO maupun algoritma PSO. Dengan demikian, diketahui bahwa nilai terbaik ω_{min} adalah semakin mendekati 1.

Parameter N mempresentasikan banyaknya individu (kandidat solusi). Berdasarkan hasil uji parameter N (lihat Tabel 4.7), dapat diketahui semakin besar nilai N maka semakin mendekati optimal hasil yang didapatkan. Hal ini jelas karena semakin banyak kandidat solusi maka peluang mendapatkan solusi yang baik juga semakin besar. Nilai terbaik N yang diujikan adalah 200.

Berdasarkan keseluruhan hasil uji parameter, dapat diketahui bahawa dengan nilai parameter yang sama algoritma CSO-PSO selalu menghasilkan solusi yang lebih baik daripada algoritma CSO dan PSO. Hal ini terlihat dari rata-rata profit dan presentase deviasi. Waktu komputasi algoritma CSO-PSO relatif sama dengan waktu komputasi algoritma CSO. Namun, jika dibandingkan dengan algoritma PSO, waktu komputasi algoritma CSO-PSO memiliki selisih yang semakin besar apabila nilai parameter SMP semakin besar.

Berdasarkan hasil simulasi akhir menggunakan parameter terbaik yang diujikan dengan jumlah *running* program sebanyak sepuluh kali, untuk algoritma CSO-PSO didapatkan rata-rata profit Rp 4.360.750,- dengan rata-rata persentase deviasi profit terhadap *simplex* 0,05157. Kemudian, algoritma PSO memperoleh rata-rata profit sebesar Rp 4.338.400,- dengan rata-rata presentase deviasi terhadap *simplex* 0,563834. Selanjutnya, algoritma CSO mendapatkan rata-rata profit Rp 4.223.700,- dengan rata-rata presentase deviasi terhadap *simplex* 3,19276. Rata-rata waktu komputasi yang dibutuhkan untuk simulasi akhir adalah 258,19509 detik (CSO-PSO), 38,42296 detik (PSO) dan 242,6568 detik (CSO). Dari sepuluh kali simulasi akhir tersebut, diperoleh hasil terbaik algoritma CSO-PSO dengan total profit sebesar Rp 4.363.000,-, algoritma PSO sebesar Rp 4.355.000,- dan algoritma CSO sebesar Rp 4.282.500,-. Hasil terbaik algoritma CSO-PSO mencapai nilai optimum metode *Simplex* sehingga persentase deviasinya sama dengan nol.

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

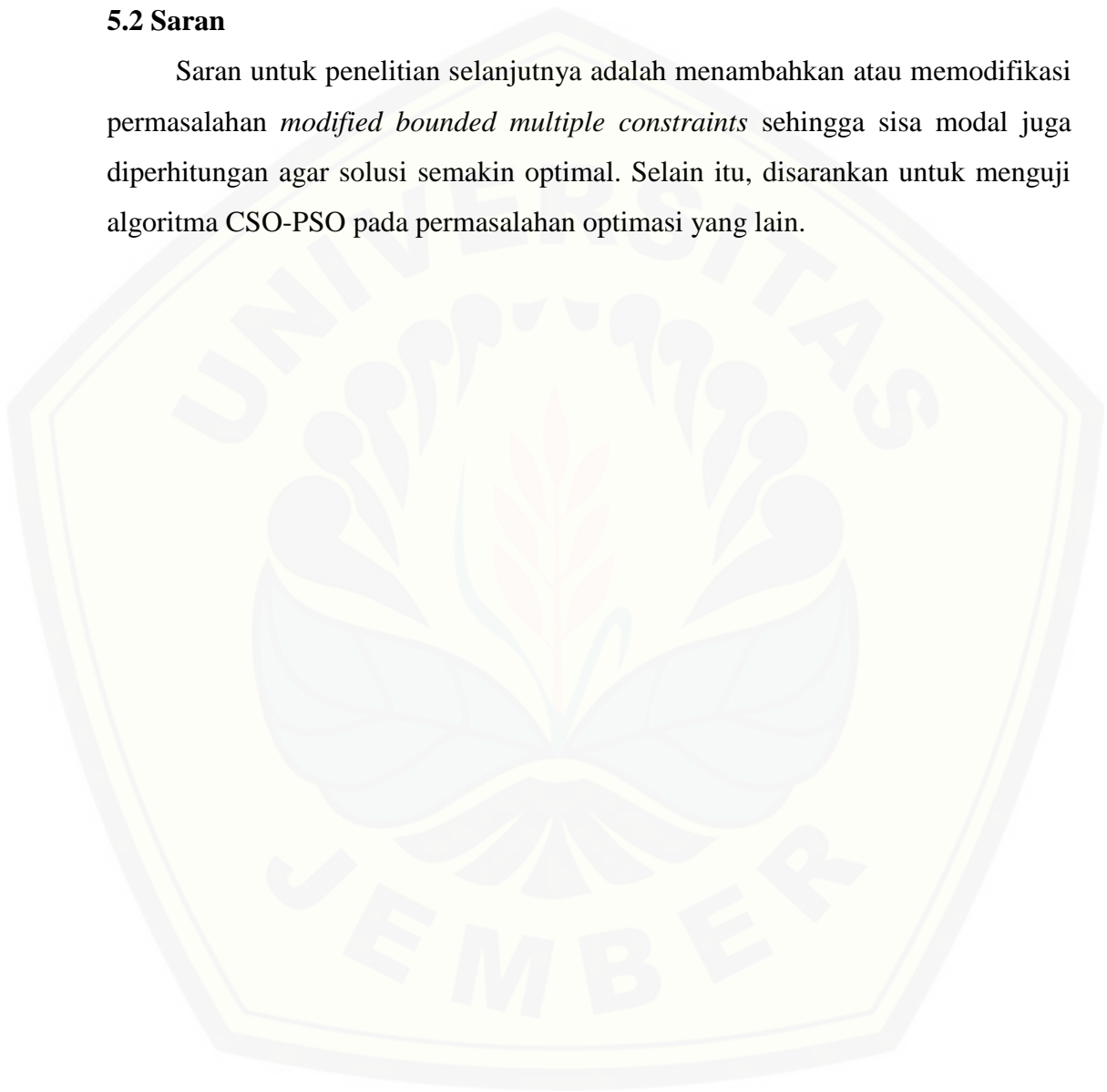
Berdasarkan hasil dan pembahasan yang telah diuraikan pada Bab 4, diambil kesimpulan sebagai berikut.

- a. Pengaruh parameter algoritma CSO-PSO, CSO dan PSO yang diuji antara lain.
 - 1) Semakin banyak *Iterasi*, maka semakin optimal hasil yang diperoleh. Nilai *Iterasi* yang diujikan adalah mendekati 5000.
 - 2) Semakin besar *Seeking Memory Pool (SMP)*, maka total profit semakin besar pula. Nilai *SMP* yang diujikan adalah 10 dari interval [1,10],
 - 3) *Counts of Dimension Change (CDC)* digunakan untuk menentukan banyak dimensi yang dimutasi. Nilai *CDC* yang diujikan adalah 0,2 dari interval [0,1].
 - 4) c_1 dan c_2 digunakan untuk mengontrol pengaruh kandidat solusi terbaik setiap *Iterasi* terhadap masing-masing kandidat solusi mampu menghasilkan solusi yang lebih baik. Kombinasi nilai c yang diujikan adalah $c_1 = 0,5$ dan $c_2 = 1,5$.
 - 5) Semakin besar nilai ω_{max} dan ω_{min} , maka semakin baik solusi yang diperoleh. Kombinasi nilai yang diujikan adalah $\omega_{max} = 0,9$ dan $\omega_{min} = 0,7$ dari interval [0,1]
 - 6) Semakin besar N atau semakin banyak kandidat solusi yang digunakan, maka hasilnya semakin mendekati optimal. Nilai N yang diujikan adalah 200.
- b. Algoritma CSO-PSO lebih efektif dibandingkan algoritma CSO dan PSO karena memiliki rata-rata persentase deviasi yang lebih kecil. Algoritma CSO-PSO memiliki waktu komputasi yang hampir sama dengan algoritma CSO. Namun, jika dibandingkan dengan algoritma PSO, waktu komputasi algoritma CSO-PSO lebih lama apabila nilai parameter *SMP* lebih dari satu ($SMP > 1$). Dengan demikian, algoritma CSO-PSO masih cukup efisien.

- c. Total profit dari hasil terbaik yang diperoleh algoritma CSO-PSO menggunakan parameter yang diujikan adalah Rp. 4.363.000,-. Hasil ini sama dengan nilai optimum metode *Simplex*.

5.2 Saran

Saran untuk penelitian selanjutnya adalah menambahkan atau memodifikasi permasalahan *modified bounded multiple constraints* sehingga sisa modal juga diperhitungkan agar solusi semakin optimal. Selain itu, disarankan untuk menguji algoritma CSO-PSO pada permasalahan optimasi yang lain.



DAFTAR PUSTAKA

- Baihaki, A. 2017. Penerapan Algoritma Cuckoo Search (CS) pada Permasalahan Multiple Constraints Knapsack 0-1. *Skripsi*. Jember: Jurusan Matematika FMIPA Universitas Jember.
- Bilgaiyan, S., S. Sagnika dan M. Das. 2014. Workflow Scheduling in Cloud Computing Environment using Cat Swarm Optimization. *IEEE International Advance Computing Conference*, 680-685.
- Boussaid, I., J. Lepagnot dan P. Siarry. 2013. A survey on optimization metaheuristics. *Information Sciences*, 237: 82-117.
- Chu, S.C., P.W. Tsai dan J.S. Pan. 2006. Cat Swarm Optimization. *LNAI 4099*, 3(1): 854-858. Berlin Heidelberg: Springer-Verlag.
- Clerc, M. 2004. Discrete Particle Swarm Optimization, Illustrated by Traveling Salesman Problem. Dalam Babu, B.V. dan G.C. Onwubolu. *New Optimization Thechniques in Engineering*, 219-239. Berlin: Springer.
- Dantzig, G.B. 1957. Discrete variable extremum problems. *Operation Research*, 5(2): 266-277.
- Fatimah, Z.M. 2016. Peneraoan Algoritma Particle Swarm Optimization untuk Vehicle Routing Problem with Time Windows pada Kasus Pendistribusian Barang. *Skripsi*. Jember: Jurusan Matematika FMIPA Universitas Jember.
- Gherboudj, A., A. Layeb dan S. Chikhi. 2012. Solving 0-1 Knapsack Problem by a Discrete Binary Version of Cuckoo Search Algorithm. *International Journal Bio-Inspired Computation*, 4(4): 229-236.
- Kasyfi, R.B. 2019. Penerapan Algoritma Centripetal Accelerated Particle Swarm Optimization pada Permasalahan Multiple Bounded Knapsack. *Skripsi*. Jember: Jurusan Matematika FMIPA Universitas Jember.
- Kellerer, H., D. Pisinger dan U. Pferschy. 2004. *Knapsack Problem*. Berlin: Springer.
- Mahendra, M.A. 2019. Penerapan Whale Optimization Algorithm pada Permasalahan Multi Knapsack Multiple Constrains. *Skripsi*. Jember: Jurusan Matematika FMIPA Universitas Jember.

- Martello, S. dan P. Toth. 1990. *Knapsack Problems: Algorithms and Computer Implementations*. New York: John Wiley & Sons, Inc.
- Mehdi, M. 2011. *Parallel Hybrid Optimization Methods for Permutation Based Problems*. Universite des Sciences et Technologie de Lille.
- Pisinger, D. 1995. A Minimal Algorithm for the Multiple-Choice Knapsack Problem. *European Journal of Operational Research*, 83(2): 394-410.
- Pisinger, D. 2000. A Minimal Algorithm for the Bounded Knapsack Problem. *INFORMS Journal on Computing*, 2(1): 75-82.
- Pornsing, C. 2014. *A Particle Swarm Optimization for the Vehicle Routing Problem*. Rhode Island: University of Rhode Island.
- Santosa, B. dan M.K. Ningrum. 2009. Cat Swarm Optimization for Clustering. *International Conference of Soft Computing and Pattern Recognition*, 54-59.
- Sharafi, Y., M.A. Khanesar dan M. Teshnehlab. 2013. Discrete Binary Cat Swarm Optimization Algorithm. *IEEE*, 1-6.
- Shi, Y. dan E. Eberhart. 1998. A Modified Particle Swarm Optimizer. *Proceedings of IEEE International Conference on Evolutionary Computation*, 69-73. Washington DC: IEEE Computer Society.
- Tseng, C.T. dan C.J. Liao. 2008. A Particle Swarm Optimization Algorithm for Hybrid Flow-Shop Scheduling with Multiprocessor Tasks. *International Journal of Production Research*, 46(17): 4655-5670.
- Yang, X.S. 2013. Metaheuristic Optimization: Nature-Inspired Algorithms and Applications. *Artificial Intelligent, Evolutionary Computation and Metaheuristics*, 427: 405-420.
- Yusiong, J.P.T. 2013. Optimizing Artificial Neural Networks using Cat Swarm Optimization Algorithm. *I.J. Intelligent Systems and Applications*, 1: 69-80.

LAMPIRAN

Lampiran A. Data penelitian

No	Nama Barang	Min (LB)	Max (UB)	Berat (kg)	Volume (cm ³)	Harga Beli (Rp)	Harga Jual (Rp)
1	Aqua 1,5 Liter	8	20	19	10800	46000	57600
2	Aqua 600 ml	10	25	15,3	21875	42000	60000
3	Aqua 330 ml	4	10	8,8	13500	33000	48000
4	Aqua 220 ml	10	20	11,3	17500	28000	30000
5	Teh Rio 190 ml	5	22	5,5	6600	17500	18500
6	Teh Pucuk 350 ml	5	20	9,4	13200	51000	54000
7	Teh Gelas 300 ml	8	20	8,2	7920	17500	18500
8	Ale-ale 190 ml	5	20	5,5	6600	17500	18500
9	Mizone 500 ml	2	5	7	9600	38000	42000
10	Aquis 220 ml	10	50	11,3	17500	13500	15000
11	Kopikap 150 ml	4	20	4,5	6000	19000	24000
12	Le Mineral 600 ml	5	15	15,4	25000	38000	42000
13	Kapal Api Signature 200 ml	1	5	3,2	4500	55000	60000
14	Freastea 296 ml	2	8	8	10500	38000	40000
15	Minute Maid 296 ml	2	8	8	10500	38000	40000
16	Al Qodiry 1,5 Liter	10	20	19	22500	50000	54000
17	Al Qodiry 600 ml	15	25	15,3	21875	40000	48000
18	Luwak White Coffe	1	6	4,4	11250	193000	200000
19	Top Plus Coffe	4	10	2,5	7500	48000	60000
20	ABC Plus Coffe	1	3	2,5	7500	41000	60000
21	Energen	2	8	5	12000	188000	240000
22	Sariwangi	1	2	2,25	26250	240000	280000
23	Sunlight 100 ml	1	3	8	15360	122500	144000
24	Superpel	1	3	8,8	16000	122500	144000
25	Molto	1	4	5	10800	150000	180000
26	Downy	1	4	8,5	15000	300000	360000
27	Mie Sedaap Kuah	2	7	9,2	15000	84000	100000
28	Mie Sedaap Goreng	2	7	9,2	15000	88000	100000
29	Sarimi Isi 2 Kuah	2	7	3,5	15000	69000	84000
30	Sarimi Isi 2 Goreng	2	7	3,3	15000	69000	84000
31	GIV	1	3	5,7	15400	120000	144000
32	LifeBioy	1	3	13	17745	374000	432000
33	Shampo Clear	1	3	5,5	13200	380000	480000
34	Filma	2	5	12	17500	132000	144000
35	Beras SB	10	30	5	6000	53000	56000
36	Beras SB	5	15	25	48000	257500	265000
37	Beras Osing	10	30	10	13500	99000	104000
38	Beras Osing	5	15	25	48000	240000	250000
39	Susu SGM	1	2	8,2	48000	624000	720000
40	Kecap Bango	1	4	3,2	15000	190000	220000

Lampiran B. Hasil metode *simplex***Microsoft Excel 16.0 Answer Report**

Worksheet: [data bangun samudra - Simplex.xlsx]Sheet1

Report Created: 3/11/2019 8:27:00 PM

Result: Solver found an integer solution within tolerance. All Constraints are satisfied.**Solver Engine**

Engine:

Simplex LP

Solution Time: 8.25 Seconds.

Iterations: 1 Subproblems: 2164

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\$43	Profit	0	4363000

Variable Cells

Cell	Name	Original Value	Final Value	Integer
\$J\$2	Aqua 1,5 Liter	0	20	Integer
\$J\$3	Aqua 600 ml	0	25	Integer
\$J\$4	Aqua 220 ml	0	11	Integer
\$J\$5	Teh Rio 190 ml	0	5	Integer
\$J\$6	Teh Pucuk 350 ml	0	20	Integer
\$J\$7	Teh Gelas 300 ml	0	8	Integer
\$J\$8	Ale-ale 190 ml	0	5	Integer
\$J\$9	Mizone 500 ml	0	5	Integer
\$J\$10	Aquis 220 ml	0	11	Integer
\$J\$11	Kopikap 150 ml	0	20	Integer
\$J\$12	Le Mineral 600 ml	0	15	Integer
\$J\$13	Kapal Api Signature 200 ml	0	5	Integer
\$J\$14	Freastea 296 ml	0	2	Integer
\$J\$15	Minute Maid 296 ml	0	5	Integer
\$J\$16	Al Qodiry 1,5 Liter	0	20	Integer
\$J\$17	Al Qodiry 600 ml	0	25	Integer
\$J\$18	Aqua 330 ml	0	10	Integer
\$J\$19	Sunlight 100 ml	0	3	Integer
\$J\$20	Superpel	0	3	Integer
\$J\$21	Molto	0	4	Integer
\$J\$22	Luwak White Coffe	0	6	Integer
\$J\$23	Downy	0	4	Integer
\$J\$24	Top Plus Coffe	0	10	Integer
\$J\$25	Energen	0	8	Integer
\$J\$26	ABC Plus Coffe	0	3	Integer

\$J\$27	Mie Sedap Kuah	0	7	Integer
\$J\$28	Mie Sedap Goreng	0	7	Integer
\$J\$29	Sarimi Isi 2 Kuah	0	7	Integer
\$J\$30	Sarimi Isi 2 Goreng	0	7	Integer
\$J\$31	GIV	0	3	Integer
\$J\$32	LifeBioy	0	3	Integer
\$J\$33	Shampo Clear	0	3	Integer
\$J\$34	Sariwangi	0	2	Integer
\$J\$35	Filma	0	5	Integer
\$J\$36	Beras SB	0	30	Integer
\$J\$37	Beras SB	0	5	Integer
\$J\$38	Beras Osing	0	30	Integer
\$J\$39	Beras Osing	0	11	Integer
\$J\$40	Susu SGM	0	2	Integer
\$J\$41	Kecap Bango	0	4	Integer

Lampiran C. Simulasi program

C.1 Uji parameter *Iterasi*

Algoritma PSO

Iterasi = 100

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3981,4	5997030	28915500	4056500	7,025	53	0,2977
2	3998,1	5961230	28388000	4019400	7,8753	60	0,1162
3	3961,15	5983770	28617500	4060000	6,9448	89	0,1069
4	3988,8	5960440	27930500	3937500	9,7525	90	0,1126
5	3991,6	5947970	27839500	3974700	8,8998	44	0,1073
6	3998,9	5928475	28023000	3963000	9,168	73	0,0974
7	3993,4	5926045	28045000	3977500	8,8357	58	0,0947
8	3954,45	5929990	27864500	4070900	6,6949	76	0,1096
9	3992,6	5971960	27905500	4007000	8,1595	71	0,1045
10	3986,9	5981785	27952500	4039500	7,4146	73	0,1068
Rata-rata				4010600	8,07701	68,7	0,12537

Iterasi = 200

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3993,8	5987725	27640500	4090700	6,2411	129	0,1973
2	3905,2	5984070	27852500	3963100	9,1657	50	0,1951
3	3994,6	5970265	28396500	4092500	6,1999	158	0,1976
4	3974,7	5917215	29119500	4092000	6,2113	107	0,1928
5	3968,8	5994755	28600500	4104000	5,9363	130	0,2049
6	3936,6	5991530	29649500	4098100	6,0715	127	0,1884
7	3999,1	5970665	26876500	4078500	6,5207	121	0,2096
8	3999,6	5984330	27690500	4090900	6,2365	133	0,2026
9	3998,35	5995445	29926500	4098700	6,0578	114	0,1982
10	3936,5	5960475	28721000	4161500	4,6184	141	0,1993
Rata-rata				4087000	6,32592	121	0,19858

Iterasi = 500

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3949,9	5998630	29281000	4180900	4,1737	493	0,4895
2	3995,2	5990865	29179000	4196900	3,807	132	0,4648
3	3974	5981610	29023500	4182800	4,1302	357	0,4705
4	3954	5980340	27958000	4160000	4,6528	232	0,4928
5	3966	5952445	29275000	4177000	4,2631	244	0,4723
6	3962,8	5940455	28575500	4137400	5,1708	241	0,5097
7	3996,7	5997100	28043500	4189000	3,9881	284	0,4742
8	3990,6	5975755	29041000	4047500	7,2313	321	0,4874
9	3989,3	5963720	28491500	4205500	3,6099	276	0,4498
10	3980,1	5983610	29397000	3999500	8,3314	174	0,468
Rata-rata				4147650	4,93583	275,4	0,4779

Iterasi = 1000

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3998,25	5996885	29247000	4170800	4,4052	352	0,9079
2	3986,4	5938650	29671500	4215000	3,3922	350	0,8998
3	3993,4	5954595	29168500	4284400	1,8015	448	0,9302
4	3983,6	5996895	29263000	4240500	2,8077	608	0,9484
5	3957,2	5977990	29946000	4289500	1,6846	380	0,9459
6	3981,9	5972930	29534000	4198500	3,7703	571	0,9319
7	3985,3	5921965	28780500	4261000	2,3378	485	0,918
8	3996,6	5990525	28116500	4219000	3,3005	397	0,9021
9	3999	5981825	28874500	4230000	3,0484	206	0,9137
10	3936,4	5987270	28889000	4212900	3,4403	491	0,9009
Rata-rata				4232160	2,99885	428,8	0,91988

Iterasi = 2000

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3981,9	5993430	29668500	4313000	1,146	961	1,8217
2	3992,1	5995090	29419500	4299500	1,4554	748	1,8352
3	3980,9	5990315	29393000	4281500	1,868	554	1,8736
4	3995,5	5979665	29619000	4252900	2,5235	479	1,8341
5	3991,3	5999865	29247000	4247000	2,6587	314	1,8238
6	3995	5940575	29997500	4291500	1,6388	1150	1,8018
7	3995,5	5966145	28293500	4206900	3,5778	1455	1,8397
8	3997,5	5945000	28400500	4257500	2,4181	879	1,7928
9	3999,2	5997725	29859500	4185000	4,0798	362	1,7411
10	3996,3	5989165	29019500	4255000	2,4754	576	1,8431
Rata-rata				4258980	2,38415	747,8	1,82069

Iterasi = 5000

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3980,2	5993560	29420000	4275500	2,0055	1184	5,3812
2	3976,7	5995845	29937000	4214500	3,4036	1249	5,9495
3	3999,6	5981305	29810500	4304000	1,3523	942	5,7486
4	3992,75	5971475	29633500	4264500	2,2576	1219	4,5927
5	3956,8	5996515	29924500	4266500	2,2118	1533	4,5213
6	3995,5	5998765	29808000	4300000	1,444	1043	4,5773
7	3993,7	5990875	28998000	4280500	1,8909	2214	4,7387
8	3985,1	5983615	29978000	4317000	1,0543	808	4,7076
9	3994,15	5975315	29152500	4260500	2,3493	1057	4,7746
10	3993,8	5994265	29997500	4321500	0,95118	1734	4,6337
Rata-rata				4280450	1,892048	1298,3	4,96252

Algoritma CSO

Iterasi = 100							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3988,2	5996860	29384500	4104700	5,9202	73	0,3897
2	3996,4	5985545	28604500	4015500	7,9647	84	0,1951
3	3998,25	5997040	28284000	3971500	8,9732	51	0,2611
4	3985,4	5979215	28082500	3931300	9,8946	100	0,2527
5	3976,15	5982205	28071000	3924900	10,0413	98	0,1227
6	3977,2	5999830	27890500	4024500	7,7584	88	0,2027
7	3995,2	5999060	27494000	3970700	8,9915	84	0,1191
8	3903,7	5999190	28673000	4027500	7,6897	57	0,1268
9	3998,65	5947485	26985000	3862500	11,4715	81	0,1668
10	3916,5	5994205	27942500	3929200	9,9427	100	0,1523
Rata-rata				3976230	8,86478	81,6	0,1989

Iterasi = 200							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3997,55	5972695	26894000	3865100	11,4119	182	0,2558
2	3863,1	5998105	27923500	3868600	11,3317	100	0,2059
3	3966,3	5999295	27507000	3832000	12,1705	188	0,3214
4	3966,9	5998880	27431000	4098300	6,0669	179	0,285
5	3978,5	5995245	28368000	4062000	6,8989	170	0,4058
6	3986,2	5996935	29471000	4033800	7,5453	64	0,2362
7	3989,4	5992620	28973500	4070500	6,7041	179	0,3609
8	3996,45	5981515	28067500	3981900	8,7348	198	0,375
9	3996	5984430	29822500	4202500	3,6787	181	0,2287
10	3973,9	5999250	28607500	4017200	7,9257	170	0,3549
Rata-rata				4003190	8,24685	161,1	0,30296

Iterasi = 500							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3999,95	5949235	28756000	3971500	8,9732	361	0,6371
2	3997,2	5985270	27348500	3923300	10,0779	40,5	0,5531
3	3972,2	5999005	27600000	3996600	8,3979	405	0,617
4	3947,6	6000000	27650500	4130600	5,3266	215	0,459
5	3995,2	5989030	28436000	4047700	7,2267	469	0,4547
6	3998,4	5984070	28511500	4004500	8,2168	240	0,4874
7	3999,05	5999515	27491000	4065200	6,8256	162	0,5667
8	3974,8	5992160	28474000	4138500	5,1455	206	0,4827
9	3997,7	5990835	29061500	4084800	6,3763	476	0,5893
10	3971,8	5993070	29828000	3837500	12,0445	379	0,5404
Rata-rata				4020020	7,8611	295,35	0,53874

Iterasi = 1000

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3999,9	5981665	28671500	4021000	7,8386	305	1,2855
2	3964,3	5949865	29999500	4173500	4,3433	545	0,945
3	3996,65	5998130	28853000	4046000	7,2656	244	0,9074
4	3990,9	5995665	29049500	4100000	6,028	280	0,9568
5	3944,1	5999270	28625500	3990000	8,5492	101	0,7837
6	3999,5	5940695	27765000	4052200	7,1235	569	0,9683
7	3956,6	5999230	29664500	4032700	7,5705	109	0,7783
8	3972	5997430	29171500	4179800	4,1989	123	0,7954
9	3982,6	5997930	28475000	3956700	9,3124	177	0,8499
10	3999,85	5998060	27726500	3950400	9,4568	291	0,9226
Rata-rata				4050230	7,16868	274,4	0,91929

Iterasi = 2000

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3998,3	5982535	28776000	4067200	6,7797	490	1,649
2	3975,4	5995630	28950500	4090000	6,2572	130	1,5192
3	3911,1	5994120	28453500	3989000	8,5721	342	1,6905
4	3998,9	5952970	27951000	4004500	8,2168	251	1,5404
5	3998,7	5983130	29537500	4142000	5,0653	123	1,4749
6	3993,75	5999915	29242000	4098300	6,0669	165	1,5248
7	3927,1	5999950	28121000	4000600	8,3062	685	1,8068
8	3998,3	5965055	28785500	4034500	7,5292	595	1,731
9	3995,8	5931155	28159000	3995100	8,4323	219	1,4977
10	3965,9	5997790	28608500	4131500	5,306	243	1,5784
Rata-rata				4055270	7,05317	324,3	1,60127

Iterasi = 5000

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3916,3	5998635	29160500	4097100	6,0944	207	4,46
2	3999,9	5912185	28089500	4034900	7,5201	293	4,5653
3	3967,1	5999315	27997500	4003000	8,2512	223	4,5167
4	3998,3	5965400	27903500	4082000	6,4405	1241	3,8915
5	3996,35	5981825	29651500	4117000	5,6383	656	4,2267
6	3979,75	5998920	27536500	3873300	11,2239	236	3,8692
7	3997,3	5998445	27533000	3954000	9,3743	475	4,7195
8	3998,7	5999680	29269000	4206500	3,587	332	3,9751
9	3999,15	5999150	29680500	4118900	5,5948	354	3,965
10	3934	5999095	29743000	4163000	4,584	223	3,8817
Rata-rata				4064970	6,83085	424	4,20707

Algoritma CSO-PSO

Iterasi = 100							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3936,9	5965905	29776500	4127700	5,3931	98	0,4294
2	3998,9	59646665	27941000	4125900	5,4343	71	0,2247
3	3976,2	5999500	28788000	4157400	4,7124	90	0,3047
4	3988,5	5996260	28981500	4147400	4,9416	97	0,2844
5	3944,6	5980960	29846000	4162000	4,6069	99	0,1549
6	3945,3	5865105	27963000	4076500	6,5666	96	0,2243
7	3998,7	5991465	28259000	4013600	8,0083	99	0,1546
8	3972,4	5991800	27487000	4127900	5,3885	100	0,1753
9	3942,7	5999885	29149500	4119500	5,581	97	0,1992
10	3972,6	5979570	27879000	4097300	6,0898	100	0,1806
Rata-rata				4115520	5,67225	94,7	0,23321

Iterasi = 200							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3943,7	5975585	29604000	4197000	3,8047	199	0,3209
2	3991,6	5967965	28603000	4132000	5,2945	186	0,2731
3	3983,1	5999460	28982000	4112900	5,7323	182	0,3878
4	3941,8	5924575	27800000	4110500	5,7873	197	0,4346
5	3952,1	5988005	29255500	4145500	4,9851	194	0,4531
6	3951,7	5929495	29875500	4194500	3,862	197	0,3355
7	3996,9	5996935	29871500	4230500	3,0369	179	0,4029
8	3956,5	5883045	27598000	4125400	5,4458	199	0,3862
9	3968,4	5936695	29005500	4223300	3,2019	192	0,2701
10	3990,4	5992310	28708500	4223000	3,2088	188	0,4075
Rata-rata				4169460	4,43593	191,3	0,36717

Iterasi = 500							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3981,3	5982950	29809000	4267500	2,1889	474	0,8697
2	3944,9	5988445	29146000	4205500	3,6099	499	0,6886
3	3997,5	5985705	29983000	4255000	2,4754	470	0,7344
4	3965,5	5985955	28898500	4202000	3,6901	454	0,6689
5	3995,3	5967780	29103000	4230500	3,0369	474	0,792
6	3995,9	5978515	29491000	4248500	2,6243	494	0,7024
7	3998,45	5985170	28823500	4199500	3,7474	495	0,8617
8	3975,4	5999100	29110500	4257500	2,4181	454	0,9332
9	3986,9	5993650	29834000	4241000	2,7962	495	0,7331
10	3985	5997195	29678500	4276500	1,9826	494	0,7234
Rata-rata				4238350	2,85698	480,3	0,77074

Iterasi = 1000

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3993,8	5981720	29325000	4196500	3,8162	944	1,9609
2	3971,1	5904475	29977500	4284500	1,7992	997	1,4143
3	3992,5	5936265	29396000	4304000	1,3523	993	1,5915
4	3991	5997475	29128000	4226500	3,1286	933	1,548
5	3999,4	5998010	29930500	4244000	2,7275	969	1,3246
6	3994,1	5920780	28413500	4189000	3,9881	938	1,5792
7	3997,5	5949530	29718500	4312500	1,1575	950	1,7285
8	3998,3	5965695	29790500	4265500	2,2347	993	1,6714
9	3998,3	5985105	29965000	4321000	0,9626	885	1,3955
10	3999,4	5986005	29468500	4276900	1,9734	980	1,4404
Rata-rata				4262040	2,31401	958,2	1,56543

Iterasi = 2000

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3992,8	5997225	29982500	4326000	0,84804	1634	2,906
2	3994,3	5994905	29995500	4268000	2,1774	1734	2,6058
3	3992,4	5981375	29998000	4301000	1,421	1908	3,8802
4	3999,7	5986025	29970500	4310000	1,2148	1935	2,5669
5	3985,5	5999265	29295500	4249500	2,6014	1964	2,5527
6	3995,85	5983635	29993000	4307500	1,2721	1864	2,6302
7	3993,5	5998725	29867000	4329000	0,77928	1857	2,8121
8	3994,6	5995005	29805500	4233000	2,9796	1880	2,7551
9	3999	5987645	29987500	4300500	1,4325	1812	2,6504
10	3998,3	5989725	29971000	4337000	0,5959	1943	2,6852
Rata-rata				4296150	1,532202	1853,1	2,80446

Iterasi = 5000

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3988	5992225	29993500	4272500	2,0743	4116	7,4179
2	3996,6	5999985	29995000	4285500	1,7763	4730	7,6452
3	2991,55	5944355	29999500	4282500	1,8451	4108	7,4488
4	3997,1	5978625	29998000	4318500	1,0199	4715	7,1328
5	3988,9	5954295	29991000	4302000	1,3981	4732	8,1556
6	3988,15	5958285	29998000	4311000	1,1918	4974	7,2914
7	3975,5	5996895	29929500	4299500	1,4554	4312	13,5825
8	3998,3	5986345	29974500	4285500	1,7763	4904	7,0062
9	3997,3	5993305	29984000	4341000	0,50424	4540	7,0687
10	3998,5	5995505	29929000	4340500	0,5157	4980	6,7215
Rata-rata				4303850	1,355714	4611,1	7,94706

C.2 Uji parameter *SMP*

Algoritma CSO

SMP = 1

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3999,9	5981665	28671500	4021000	7,8386	305	1,2855
2	3964,3	5949865	29999500	4173500	4,3433	545	0,945
3	3996,65	5998130	28853000	4046000	7,2656	244	0,9074
4	3990,9	5995665	29049500	4100000	6,028	280	0,9568
5	3944,1	5999270	28625500	3990000	8,5492	101	0,7837
6	3999,5	5940695	27765000	4052200	7,1235	569	0,9683
7	3956,6	5999230	29664500	4032700	7,5705	109	0,7783
8	3972	5997430	29171500	4179800	4,1989	123	0,7954
9	3982,6	5997930	28475000	3956700	9,3124	177	0,8499
10	3999,85	5998060	27726500	3950400	9,4568	291	0,9226
Rata-rata				4050230	7,16868	274,4	0,91929

SMP = 5

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3992,75	5988205	29219000	4231000	3,0254	963	4,1963
2	3942,5	5999880	29125000	4143600	5,0287	718	3,829
3	3997,5	5998490	28831500	4119500	5,581	970	3,9199
4	3967,9	5984320	29020000	4097700	6,0807	933	3,7468
5	3994,85	5958115	29387500	4206500	3,587	948	3,685
6	3942,3	5994845	28275000	3972000	8,9617	988	3,4966
7	3996,9	5966770	28763000	4132500	5,2831	855	3,6566
8	3942,05	5999660	29451500	4037300	7,465	737	3,5236
9	3997,1	5993845	28978000	4213500	3,4265	968	3,6222
10	3997,3	5985895	28577000	4161000	4,6298	980	3,7975
Rata-rata				4131460	5,30689	906	3,74735

SMP = 10

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3998	5996410	29581500	4244500	2,716	872	6,1099
2	3997,5	5988340	29107000	4193000	3,8964	981	6,4629
3	3999,3	4998480	29406000	4163500	4,5725	967	6,2723
4	3990,7	5961920	29986500	4254500	2,4868	921	6,1856
5	3951,5	5997085	29654500	4109600	5,8079	831	6,3049
6	3996,05	5987075	28961500	4118500	5,6039	994	6,1888
7	3951,7	5987630	28252500	4139000	5,1341	983	6,2042
8	3997,4	5999145	28813000	4174000	4,3319	999	6,1583
9	3973,5	5997105	29996500	4194000	3,8735	875	6,2849
10	3993,4	5979190	28130500	4184000	4,1027	985	6,356
Rata-rata				4177460	4,25257	940,8	6,25278

Algoritma CSO-PSO

SMP = 1

No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3993,8	5981720	29325000	4196500	3,8162	944	1,9609
2	3971,1	5904475	29977500	4284500	1,7992	997	1,4143
3	3992,5	5936265	29396000	4304000	1,3523	993	1,5915
4	3991	5997475	29128000	4226500	3,1286	933	1,548
5	3999,4	5998010	29930500	4244000	2,7275	969	1,3246
6	3994,1	5920780	28413500	4189000	3,9881	938	1,5792
7	3997,5	5949530	29718500	4312500	1,1575	950	1,7285
8	3998,3	5965695	29790500	4265500	2,2347	993	1,6714
9	3998,3	5985105	29965000	4321000	0,9626	885	1,3955
10	3999,4	5986005	29468500	4276900	1,9734	980	1,4404
Rata-rata				4262040	2,31401	958,2	1,56543

SMP = 5

No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3999,3	5988825	29999500	4327500	0,81366	942	4,21
2	3992,5	5985985	29995000	4331000	0,73344	989	3,6516
3	3995,4	5962645	29983000	4276000	1,994	700	3,812
4	3996,1	5991445	29916500	4322500	0,92826	954	3,6883
5	3999,3	5964935	29997500	4331000	0,73344	697	3,7178
6	3998	5953105	29975500	4309500	1,2262	967	3,6323
7	3988,8	5993345	29991500	4320000	0,98556	885	3,773
8	3999,6	5989225	29880500	4273500	2,0513	960	3,4319
9	3996	5991625	29988000	4324500	0,88242	945	3,6269
10	3997,2	5988600	29996500	4293500	1,5929	987	4,0057
Rata-rata				4310900	1,194118	902,6	3,75495

SMP = 10

No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3990,3	5990485	29990500	4345000	0,41256	908	6,2015
2	3993	5995385	29996000	4297000	1,5127	999	5,8743
3	3996	5966845	29992000	4340000	0,52716	798	5,96
4	3986,3	5991885	29999500	4325000	0,87096	926	5,8638
5	3997,1	5952945	30000000	4340500	0,5157	829	5,8962
6	3996,75	5996625	29988500	4309500	1,2262	955	5,9042
7	3997,9	5998725	29974500	4337500	0,58446	968	6,2704
8	3999,8	5989825	29986500	4351500	0,26358	848	6,1522
9	3998,8	5950865	29992000	4306500	1,295	960	6,2207
10	3998,7	5997315	29976500	4316000	1,0772	978	5,9744
Rata-rata				4326850	0,828552	916,9	6,03177

C.3 Uji parameter *CDC*

Algoritma CSO

CDC = 0,1							
No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3991,4	5934005	28335000	4116500	5,6498	999	3,5941
2	3975,9	5999135	29431000	4156000	4,7444	952	3,5468
3	3994,85	5843335	27049000	3645500	16,4451	217	3,4102
4	3971,2	5999685	29829000	4269500	2,143	887	3,4578
5	3981,6	5998455	29671500	4191000	3,9422	576	3,5315
Rata-rata				4075700	6,5849	726,2	3,50808
CDC = 0,2							
No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3998,6	5958585	28841500	4207300	3,5686	975	3,5686
2	3969,5	5995705	28732500	3982500	8,7211	828	3,7641
3	3998,1	5996190	29907000	4201500	3,7016	984	3,7057
4	3942,7	5996280	29016500	3987500	8,6065	986	3,6529
5	3995,4	5963385	29953500	4063500	6,8645	884	3,6757
Rata-rata				4088460	6,29246	931,4	3,6734
CDC = 0,3							
No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3996,3	5980010	28933000	3997500	8,3773	944	3,6604
2	3979,1	5996505	28663500	4120500	5,5581	876	3,7037
3	3994,5	5953635	28765500	4165000	4,5382	941	3,6777
4	3990,25	5999355	28699500	3960700	9,2207	552	3,7756
5	3929,5	5994315	29972000	4072800	6,6514	970	3,7817
Rata-rata				4063300	6,86914	856,6	3,71982
CDC = 0,4							
No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3988,6	5980685	28750500	4039000	7,4261	973	3,7158
2	3967,5	5998075	29406000	3951500	9,4316	993	3,5438
3	3991,8	5973705	29675500	4127000	5,4091	994	3,829
4	4996,3	5916445	27409000	4045500	7,2771	910	4,4301
5	3995	5994685	28132000	4153000	4,8132	393	4,6233
Rata-rata				4063200	6,87142	852,6	4,0284
CDC = 0,5							
No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3914,5	5991850	29295500	3956500	9,317	975	4,9737
2	3999,45	5990380	28003000	4105000	5,9134	863	4,1789
3	3950,6	5994450	28895000	4056300	7,0296	936	3,2488
4	3986	5929210	27301500	3934500	9,8212	943	3,2722
5	3937,8	5999775	29307000	4114900	5,6865	893	3,3333
Rata-rata				4033440	7,55354	922	3,80138

CDC = 0,6

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3994,6	5974875	28732500	4110900	5,7781	716	3,3493
2	3956,2	5994165	27395000	3944500	9,592	796	3,3418
3	3985,65	5999125	28951500	4076300	6,5712	937	3,692
4	3999,1	5997450	28412500	3930000	9,9244	325	3,4509
5	3964,6	5996365	28336000	4057500	7,0021	106	3,88
Rata-rata				4023840	7,77356	576	3,5428

CDC = 0,7

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3995,7	5999710	28047000	3995000	8,4346	131	3,5338
2	3999,45	5933275	29295500	4056500	7,025	252	3,5022
3	3956,1	5995090	28489000	3935500	9,7983	877	3,459
4	3971,7	5993805	27674500	4061000	6,9218	172	3,307
5	3997,8	5970940	28560500	4022000	7,8157	473	3,5872
Rata-rata				4014000	7,99908	381	3,47784

CDC = 0,8

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Deviasi Perbedaan Profit (%)	Deviasi (%)	Iterasi Konvergen	Waktu Komputasi
1	3977,55	5986745	29133500	4080600	6,4726	904	3,2991
2	3995,75	5931205	29315500	3985000	8,6638	187	3,3857
3	3999,5	5970530	27751000	3988500	8,5835	456	3,7266
4	3996,9	5998505	28332000	4025400	7,7378	544	3,5503
5	3983,8	5999775	27879000	3976500	8,8586	861	3,6432
Rata-rata				4011200	8,06326	590,4	3,52098

CDC = 0,9

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3997,2	5998570	28939000	3945000	9,5806	179	3,4902
2	3964,4	5997595	27355500	3861900	11,4852	317	3,8861
3	3974,05	5999895	28653500	4023500	7,7813	1000	3,786
4	3996,2	5992040	28740500	4056000	7,0364	963	4,4228
5	3994,65	5995795	28323500	3969500	9,019	759	3,5372
Rata-rata				3971180	8,9805	643,6	3,82446

Algoritma CSO-PSO

CDC = 0,1

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3995,5	5979675	29594000	4238500	2,8535	867	3,6463
2	3999,85	5949975	29983000	4296000	1,5356	800	3,6798
3	3999,3	5994085	29996500	4296500	1,5242	774	3,5512
4	3989,8	5988500	29998500	4282000	1,8565	973	3,4691
5	3998,9	5988985	29996500	4309500	1,2262	921	3,5172
Rata-rata				4284500	1,7992	867	3,57272

CDC = 0,2

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3976,6	5993160	29988500	4253500	2,5097	985	3,391
2	3993	5996435	29987500	4319500	0,99702	998	3,5586
3	3975,6	5949885	29997000	4292500	1,6159	997	3,5239
4	3993,6	5993615	29967000	4348500	0,33234	929	3,4288
5	3992,9	5997665	29999500	4328500	0,79074	963	3,4419
Rata-rata				4308500	1,24914	974,4	3,46884

CDC = 0,3

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3996,8	5998505	2998505	4323000	0,9168	648	3,3633
2	3990,3	5966770	29997500	4268500	2,1659	995	3,4222
3	3998,5	5989925	29939500	4245500	2,6931	663	3,3133
4	3952,9	5992775	29987500	4268500	2,1659	954	3,4563
5	3991,2	5987480	29970500	4291500	1,6388	887	3,42
Rata-rata				4279400	1,9161	829,4	3,39502

CDC = 0,4

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3961,2	5949595	29998000	4269500	2,143	845	3,4454
2	3968,8	5999265	29599500	4252500	2,5327	993	3,2331
3	3953,6	5986505	29984000	4253000	2,5212	985	3,5734
4	3986,8	5984365	29977500	4276000	1,994	637	4,1201
5	3948,1	5995065	29871500	4249500	2,6014	793	3,7518
Rata-rata				4260100	2,35846	850,6	3,62476

CDC = 0,5

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3978,1	5991010	29764500	4224000	3,1859	889	3,9425
2	3996,7	5962395	29337500	4228500	3,0827	890	3,66
3	3996,7	5960725	29676000	4262000	2,3149	621	2,943
4	3999,25	5983295	29849000	4251000	2,567	844	3,0045
5	3977,2	5992790	29981000	4251500	2,5556	989	2,9634
Rata-rata				4243400	2,74122	846,6	3,30268

CDC = 0,6

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3972,6	5979175	29729500	4199500	3,7474	994	3,0521
2	3959,3	5963165	28057000	4170500	4,4121	961	2,9974
3	3985,3	5995925	29555000	4213000	3,438	948	3,4008
4	3997,6	5963075	28599000	4175500	4,2975	989	3,1155
5	3976,1	5954050	29242000	4283500	1,8221	701	3,2941
Rata-rata				4208400	3,54342	918,6	3,17198

CDC = 0,7

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3995,1	5986600	29907500	4260500	2,3493	986	3,1412
2	3958,8	5929070	29636000	4089000	6,2801	985	3,1328
3	3935,2	5989315	28479000	4125000	5,455	941	3,047
4	3923,6	5993015	29643000	4134000	5,2487	478	2,9906
5	3993	5999000	29416000	4274500	2,0284	935	3,2161
Rata-rata				4176600	4,2723	865	3,10554

CDC = 0,8

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3972,7	5990500	29256000	4178500	4,2287	905	3,0348
2	3968,6	5988485	29463500	4149000	4,9049	976	3,0193
3	3967,6	5969380	29466000	4237500	2,8765	786	3,3839
4	3987,9	5961625	28699000	4077500	6,5437	824	3,1713
5	3982,7	5983435	29565000	4192900	3,8987	984	3,1702
Rata-rata				4167080	4,4905	895	3,1559

CDC = 0,9

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3994,7	5980755	29808000	4090500	6,2457	912	3,1773
2	3983,1	5975345	28135000	4187500	4,0225	992	3,5844
3	3928,1	598135	28857000	4221400	3,2455	807	3,0489
4	3996,5	5988370	29712500	4016000	7,9532	394	3,9074
5	3998,05	5989465	29508500	4082500	6,4291	811	3,1514
Rata-rata				4119580	5,5792	783,2	3,37388

C.4 Uji parameter c

Algoritma PSO

 $(c_1; c_2)$ 0,5; 0,5

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3943,8	5983345	29821000	4167600	4,4786	381	2,5784
2	3972,6	5998105	29534500	4115500	5,6727	208	0,98201
3	3949,6	5995345	29585500	4214300	3,4082	339	1,0005
4	3996,7	5965955	27185000	4060800	6,9264	220	1,0052
5	3981	5997365	28294000	3960100	9,2345	109	0,89728
Rata-rata				4103660	5,94408	251,4	1,292678

 $(c_1; c_2)$ 0,5; 1

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3993,45	5990890	29673500	4200500	3,7245	736	1,095
2	3991,2	5996955	28927000	4224000	3,1859	941	1,024
3	3996,4	5995980	29403000	4265900	2,2255	713	1,0414
4	3999,9	5923775	28799000	4166000	4,5152	626	0,95286
5	3974,2	5992015	29990000	4238500	2,8535	481	0,95083
Rata-rata				4218980	3,30092	699,4	1,012818

 $(c_1; c_2)$ 0,5; 1,5

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3982,5	5994715	29965000	4285500	1,7763	885	1,0561
2	3987,4	5997575	29208500	4301000	1,421	711	0,99468
3	3997,2	5986505	29660500	4271000	2,1086	642	0,99558
4	3995,4	5999005	29870000	4307500	1,2721	967	1,0399
5	3995	5987385	29515000	4297500	1,5013	642	1,0086
Rata-rata				4292500	1,61586	769,4	1,018972

 $(c_1; c_2)$ 0,5; 2

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3994,9	5981320	28718000	4275900	1,9963	962	1,0067
2	3995,3	5974660	29405500	4255500	2,4639	993	0,99865
3	3991,4	5990425	29986000	4342500	0,46986	967	1,1277
4	3991,1	5985690	29333500	4270000	2,1316	926	1,0719
5	3999,5	5996810	29964500	4267000	2,2003	928	1,0496
Rata-rata				4282180	1,852392	955,2	1,05091

 $(c_1; c_2)$ 1; 0,5

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3947,3	5995565	29063000	4209000	3,5297	338	1,0457
2	3992,3	5980635	29457000	4170000	4,4236	353	1,1769
3	3952,2	5997475	28655500	4181000	4,1714	126	1,0106
4	3991,8	5999065	28872500	4103200	5,9546	132	0,97188
5	3999,1	5998115	29458000	4228500	3,0827	421	1,0038
Rata-rata				4178340	4,2324	274	1,041776

(c_1, c_2) 1; 1							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3994,1	5999485	28257500	4250500	2,5785	752	1,1458
2	3986,4	5990030	29888000	4325000	0,87096	551	1,0189
3	3975,8	5999145	29177500	4198500	3,7703	488	1,1834
4	3998,65	5997740	29698000	4247500	2,6473	377	1,0062
5	3990,4	5998525	28832500	4180000	4,1944	478	1,143
Rata-rata				4240300	2,812292	529,2	1,09946
(c_1, c_2) 1; 1,5							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3979,7	5991030	29987500	4258000	2,4066	797	1,1721
2	3984,8	5991125	29326000	4282000	1,8565	999	1,0207
3	3968,2	5999125	29559000	4271000	2,1086	623	0,95035
4	3997,35	5997525	29972000	4286500	1,7534	850	0,96238
5	3991,5	5997955	29980000	4306500	1,295	727	1,0277
Rata-rata				4280800	1,88402	799,2	1,026646
(c_1, c_2) 1; 2							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3958	5997025	29823000	4276000	1,994	885	1,0613
2	3994,25	5989035	29996500	4303500	1,3637	961	1,0557
3	3991,6	5993840	29958500	4274500	2,0284	780	1,1881
4	3998,3	5997025	29335000	4225000	3,163	943	1,0073
5	3999,8	5963395	29323500	4254500	2,4868	952	1,2512
Rata-rata				4266700	2,20718	904,2	1,11272
(c_1, c_2) 1,5; 0,5							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3998,1	5985175	28855500	4206500	3,587	540	1,2092
2	3895,5	5999365	29462500	4215700	3,3761	779	1,067
3	3999,7	5940950	28216000	4100500	6,0165	952	1,0108
4	3997,25	5998075	29706000	4137500	5,1685	156	0,9574
5	3988,1	5986825	29975000	4252000	2,5441	291	1,0464
Rata-rata				4182440	4,13844	543,6	1,05816
(c_1, c_2) 1,5; 1							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3992,9	5998345	29521500	4300500	1,4325	917	1,0426
2	3984,7	5993685	29890000	4283500	1,8221	731	1,0367
3	3983,5	5995495	29986500	4221500	3,2432	319	0,9743
4	3997,8	5985185	28417000	4198000	3,7818	590	1,2179
5	3995,6	5954635	28652500	4126000	5,432	670	1,0094
Rata-rata				4225900	3,14232	645,4	1,05618

(c_1, c_2) 1,5; 1,5							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3999,4	5970805	30000000	4319000	1,0085	757	1,0613
2	3998,9	5996275	29917500	4317500	1,0429	909	1,1126
3	3998,8	2996845	29557000	4202000	3,6901	846	1,1802
4	3989,4	5998625	29004000	4251500	2,5556	954	1,1801
5	3999,7	5990225	29836000	4327500	0,81366	799	1,1735
Rata-rata				4283500	1,822152	853	1,14154
(c_1, c_2) 1,5; 2							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3997,5	5997750	28908000	4230500	3,0369	630	1,0496
2	3998,9	5999995	29209500	4277500	1,9597	820	1,0593
3	3916,5	5992720	29980000	4186500	4,0454	825	1,0665
4	3978,6	5987010	29498000	4191000	3,9422	774	0,9715
5	3999,05	5975820	29243500	4260000	2,3608	777	0,9696
Rata-rata				4229100	3,069	765,2	1,0125
(c_1, c_2) 2; 0,5							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3997,6	5992460	29993000	4152000	4,8361	636	0,8644
2	3996,3	5950340	28463000	4229000	3,0713	793	0,8703
3	3967,6	5995265	29528500	4257000	2,4295	464	1,0547
4	3997,1	5993585	29995000	4307000	1,2835	972	1,0071
5	3960,35	5998940	29994500	4200000	3,736	929	1,0887
Rata-rata				4229000	3,07128	758,8	0,97704
(c_1, c_2) 2; 1							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3999	5973325	29633000	4257000	2,4295	931	1,4059
2	3965,1	5998475	28585000	4235500	2,9223	523	1,0904
3	3994,5	5967985	29957000	4217000	3,3463	958	0,86656
4	3943,5	5992945	29905500	4257000	2,4295	848	0,87627
5	3995,3	5996525	29957000	4280000	1,9024	949	1,1934
Rata-rata				4249300	2,606	841,8	1,086506
(c_1, c_2) 2; 1,5							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3998,2	5976495	29067500	4285000	1,7878	692	1,4658
2	3978,6	5976105	29996000	4229500	3,0598	598	1,3148
3	3967,8	5978585	29614500	4237500	2,8765	901	1,3629
4	3977,1	5967710	29187000	4232500	2,9911	750	0,8329
5	3996,8	5994630	29771000	4269500	2,143	617	0,8275
Rata-rata				4250800	2,57164	711,6	1,16078

(c_1, c_2) 2; 2

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3922,3	5992140	29451000	4142200	5,0607	538	1,5456
2	3997,1	5990645	29445500	4225000	3,163	424	1,8281
3	3985	5956420	28543000	4210000	3,5068	515	1,0332
4	3956	5967425	29755500	4246500	2,6702	213	1,0642
5	3983,8	5977735	29669500	4203500	3,6557	412	1,1381
Rata-rata				4205440	3,61128	420,4	1,32184

Algoritma CSO-PSO

(c_1, c_2) 0,5; 0,5

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3997,6	5998045	29959500	4304000	1,3523	461	5,5081
2	3998,3	5987305	29997500	4319500	0,99702	990	3,6455
3	3999	5967805	29994500	4280000	1,9024	756	3,7285
4	3995,1	5972465	28989000	4248500	2,6243	786	3,5541
5	3977,7	5998265	29994500	4313500	1,1345	419	3,147
Rata-rata				4293100	1,602104	682,4	3,91664

(c_1, c_2) 0,5; 1

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3998,5	5977545	29979500	4341000	0,50424	868	3,8584
2	3998,6	5988645	29996000	4325000	0,87096	839	3,492
3	3999	5995565	29997000	4272000	2,0857	508	3,283
4	3988,9	5996385	29999500	4297500	1,5013	846	3,6131
5	3998,8	5972625	30000000	4314500	1,1116	774	3,3873
Rata-rata				4310000	1,21476	767	3,52676

(c_1, c_2) 0,5; 1,5

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3973,8	5999165	29990000	4316000	1,0772	840	4,1805
2	3999,7	5979635	29999500	4348500	0,33234	775	3,473
3	3994,2	5974165	29990500	4351500	0,26358	794	3,3817
4	3981,7	5992465	29999000	4347000	0,36672	944	3,4724
5	3996,7	5996465	29998000	4349500	0,30942	966	3,6072
Rata-rata				4342500	0,469852	863,8	3,62296

(c_1, c_2) 0,5; 2

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3992,4	5996325	29992000	4347500	0,35526	830	3,4696
2	3980,3	5990015	29995500	4340000	0,52716	980	3,2936
3	3969,4	5996325	29999000	4315000	1,1002	920	3,9597
4	3983,1	5971895	29965000	4330500	0,7449	993	3,6078
5	3996,6	5983675	29989500	4343500	0,44694	948	3,5089
Rata-rata				4335300	0,634892	934,2	3,56792

(c_1, c_2) 1; 0,5							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3999,1	5999325	29992000	4329000	0,77928	537	3,4544
2	3983,5	5998950	29997000	4271500	2,0972	422	3,9342
3	3999,9	5994295	29756000	4222500	3,2203	575	3,3106
4	3981	5993165	29997000	4314500	1,1116	980	3,3079
5	3997,3	5985865	29998500	4283000	1,8336	283	3,4065
Rata-rata				4284100	1,808396	559,4	3,48272
(c_1, c_2) 1; 1							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3994,9	5982065	29995500	4330500	0,7449	827	4,042
2	3999	5999405	29993000	4323500	0,90534	881	3,4172
3	3995,5	5948545	29990500	4349500	0,30942	575	4,2159
4	3992,75	5990480	29994500	4251500	2,5556	973	3,535
5	3975	5999505	29999000	4283000	1,8336	444	3,9564
Rata-rata				4307600	1,269772	740	3,8333
(c_1, c_2) 1; 1,5							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3999,2	5990685	29984000	4338500	0,56154	786	3,7119
2	3993,1	5994245	29990500	4342500	0,46986	944	3,4133
3	3977,8	5999565	29939000	4342000	0,48132	788	3,4057
4	3994,9	5999625	29986000	4343500	0,44694	928	3,3805
5	3991,9	5996125	29995000	4338000	0,573	837	3,4805
Rata-rata				4340900	0,506532	856,6	3,47838
(c_1, c_2) 1; 2							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3998,7	5984085	29988500	4336500	0,60738	986	3,8102
2	3999,8	5995925	29998000	4321500	0,95118	976	3,5998
3	3976	5995845	29993500	4348000	0,3438	902	4,0458
4	3997,6	5996545	29992500	4340500	0,5157	973	3,6076
5	3999,7	5976935	29982500	4342500	0,46986	954	4
Rata-rata				4337800	0,577584	958,2	3,83734
(c_1, c_2) 1,5; 0,5							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3974,6	5982985	29985500	4312500	1,1575	996	4,4017
2	3996,4	5998765	29987500	4334000	0,66468	542	3,571
3	3998,5	5999745	29956500	4259500	2,3722	591	3,2845
4	3995,4	5986465	29998000	4312500	1,1575	572	3,4479
5	3984,7	5996525	29992000	4294000	1,5815	590	3,5647
Rata-rata				4302500	1,386676	658,2	3,65396

(c_1, c_2) 1,5; 1							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3964,4	5991825	29996500	4325500	0,8595	899	3,5184
2	3974,6	5978605	29992500	4325000	0,87096	883	3,3713
3	3991,1	5997665	29997500	4319500	0,99702	928	4,0023
4	3990	5998325	29997000	4323500	0,90534	655	4,0086
5	3998,8	5995425	29998000	4328000	0,8022	749	3,4482
Rata-rata				4324300	0,887004	822,8	3,66976
(c_1, c_2) 1,5; 1,5							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3994,5	5982825	29987500	4339500	0,53862	983	3,5918
2	3989,2	5976985	29954500	4341000	0,50424	1000	3,8095
3	3989	5975165	29988000	4333000	0,6876	995	4,2674
4	3997,7	5984665	29988500	4336500	0,60738	969	4,0452
5	3994,3	5968945	29999500	4331000	0,73344	871	4,1138
Rata-rata				4336200	0,614256	963,6	3,96554
(c_1, c_2) 1,5; 2							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3989,8	5988865	29895500	4329000	0,77928	980	3,5325
2	3998	5987525	29999000	4339500	0,53862	546	3,5653
3	3993,9	5924845	29989000	4333500	0,67614	953	3,6462
4	3978,6	5994125	29911500	4341000	0,50424	986	3,4273
5	3993,2	5959745	29992500	4329500	0,76782	742	3,8501
Rata-rata				4334500	0,65322	841,4	3,60428
(c_1, c_2) 2; 0,5							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3989,1	5937210	29989500	4230500	3,0369	994	3,0217
2	3998,4	5987255	29993500	4332000	0,71052	427	2,9648
3	3999,4	5998085	29996000	4335500	0,6303	812	3,8373
4	3963,4	5978525	29997500	4301000	1,421	648	3,3144
5	3998,9	5997285	29989000	4291500	1,6388	737	3,6313
Rata-rata				4298100	1,487504	723,6	3,3539
(c_1, c_2) 2; 1							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3999,9	5985445	29989000	4299500	1,4554	989	4,2995
2	3985,5	5999025	29983000	4315500	1,0887	900	3,9542
3	3999,8	5978705	29953000	4310500	1,2033	986	3,3955
4	3999,6	5999915	29985500	4308000	1,2606	647	3,5695
5	3986,2	5995625	29995000	4345000	0,41256	761	4,1284
Rata-rata				4315700	1,084112	856,6	3,86942

(c_1, c_2) 2; 1,5

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3999	5997765	29931000	4241500	2,7848	991	5,3906
2	3990,6	5991725	29971000	4348500	0,33234	988	4,3846
3	3996,7	5998885	29892500	4323000	0,9168	958	4,6436
4	3998	5997015	29995500	4356500	0,14898	604	3,1522
5	3986,6	5968225	29959000	4337000	0,59592	987	2,983
Rata-rata				4321300	0,955768	905,6	4,1108

(c_1, c_2) 2; 2

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3998,1	5951565	29999000	4282000	1,8565	868	6,3406
2	3991,6	5996785	29992000	4302000	1,3981	907	6,7196
3	3995,7	5994465	29967000	4276500	1,9826	987	3,5699
4	3996,5	5985165	29995500	4336500	0,60738	731	3,8589
5	3995,9	5998935	29997500	4314500	1,1116	879	4,4501
Rata-rata				4302300	1,391236	874,4	4,98782

Algoritma CSO

$c_1 = 0,5$

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3992,6	5999920	29255000	4143000	5,0424	947	4,1816
2	3961,8	5994545	29569500	4136500	5,1914	631	3,5649
3	3997,1	5963730	28788500	4079000	6,5093	913	3,9839
4	3970,4	5962015	29997000	4089900	6,2595	988	3,6492
5	3994,6	5985295	29443000	4157000	4,7215	918	3,5662
6	3991,5	5995465	29750500	4194500	3,862	749	3,8224
7	3995,9	5985715	28499500	4166500	4,5038	990	3,8114
8	3995,9	5989805	29982500	4281000	1,8794	642	3,5558
9	3998,15	5987905	29267000	3964500	9,1336	488	4,1469
10	3997,8	5989195	29542000	4187500	4,0225	940	3,8879
Rata-rata				4139940	5,11254	820,6	3,81702

$c_1 = 1$

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3961,15	5993735	28232000	3937500	9,7525	954	4,003
2	3940,7	5998240	29456000	4129000	5,3633	975	3,7799
3	3966,1	5997625	29430500	4166500	4,5038	651	3,695
4	3956,7	5988590	29186000	4224900	3,1653	947	4,7051
5	3994,45	5970820	29077500	4168000	4,4694	1000	3,7874
6	3991,2	5996265	28356000	4107000	5,8675	564	3,5394
7	3998,05	5984395	29995500	4244500	2,716	558	3,4324
8	3999,7	5959510	28778000	4108000	5,8446	885	3,6557
9	3995,4	5977025	29988000	4179000	4,2173	696	3,5951
10	3998,6	5982535	28520500	4097500	6,0853	427	4,2535
Rata-rata				4136190	5,1985	765,7	3,84465

$c_1 = 1,5$

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3999,8	5949435	28213500	4038500	7,4375	999	4,6943
2	3964,2	5998185	29126000	4196500	3,8162	972	3,9223
3	3997,7	5989475	28849500	4009000	8,1137	967	3,8469
4	3971,5	5998705	29373000	4130600	5,3266	977	3,7729
5	3963,2	5997145	29544000	4128400	5,377	666	3,758
6	3930,75	5998135	29324000	3978500	8,8127	183	3,5205
7	3998,2	5948155	29237500	4031000	7,6094	983	4,3809
8	3997,2	5983995	28877000	4155500	4,7559	442	3,9225
9	3975,2	5996135	29556000	4273000	2,0628	833	4,1544
10	3966,8	5995365	29179500	4134000	5,2487	775	4,1918
Rata-rata				4107500	5,85605	779,7	4,01645

$c_1 = 2; 0,5$

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3997,7	5944360	29966500	4053500	7,0937	753	3,0094
2	3996,5	5974765	29987000	4226500	3,1286	360	4,1749
3	3994,75	5970590	29384000	4133500	5,2601	984	3,6403
4	3997,5	5949015	29259500	4067000	6,7843	905	3,8043
5	3993,1	5988870	29325000	4138000	5,157	849	3,2761
6	3996,7	5975655	28956000	4002500	8,2627	861	3,3604
7	3929,4	5992885	29117500	3771700	13,5526	989	4,5822
8	3993,2	5971160	28224500	4077000	6,5551	968	4,7373
9	3994,15	5993600	29552000	4102000	5,9821	898	6,3034
10	3992,5	5992010	29071000	4242000	2,7733	975	6,6877
Rata-rata				4081370	6,45495	854,2	4,3576

C.5 Uji parameter ω_{max} dan ω_{min}

Algoritma PSO

 $(\omega_{max}, \omega_{min})$ 0,9 ; 0,7

No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3946,7	5982610	29998500	4286500	1,7534	939	0,88149
2	3970,5	5998135	29409000	4264000	2,2691	968	0,89455
3	3989,6	5994795	28832000	4289500	1,6846	931	0,86242
4	3956,3	5985125	29407500	4290500	1,6617	985	0,8474
5	3998,9	5973315	29974500	4321500	0,95118	772	0,85328
Rata-rata				4290400	1,663996	919	0,867828

 $(\omega_{max}, \omega_{min})$ 0,9 ; 0,5

No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3974,25	5999665	28662500	4220500	3,2661	859	0,85607
2	3990,9	5992005	29986500	4308500	1,2491	756	0,83819
3	3991,9	5970330	29387000	4315500	1,0887	946	0,85907
4	3994,9	5996245	29541500	4264500	2,2576	837	0,83847
5	3997,7	5961750	27961500	4240000	2,8192	968	0,8771
Rata-rata				4269800	2,13614	873,2	0,85378

 $(\omega_{max}, \omega_{min})$ 0,9 ; 0,3

No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3998,45	5936945	27756500	4163500	4,5725	916	0,83883
2	3993,5	5996065	29919000	4304000	1,3523	789	0,85751
3	3976,8	5999365	29729500	4251500	2,5556	699	0,84735
4	3979	5988140	29975500	4236000	2,9108	995	0,90302
5	3934,2	5998450	29939000	4303000	1,3752	860	0,84829
Rata-rata				4251600	2,55328	851,8	0,859

 $(\omega_{max}, \omega_{min})$ 0,9 ; 0,1

No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3981	5999185	29933000	4224500	3,1744	822	0,87071
2	3993,2	5998310	29943000	4217000	3,3463	763	0,84571
3	3972,7	5983450	29898000	4274500	2,0284	538	0,85361
4	3997,75	5993370	28466000	4230000	3,0484	690	0,81969
5	3998,1	5996685	28492500	4259000	2,3837	659	0,86616
Rata-rata				4241000	2,79624	694,4	0,851176

 $(\omega_{max}, \omega_{min})$ 0,7 ; 0,5

No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3988,5	5998710	29090500	4244000	2,7275	838	0,87408
2	3994,2	5989155	29983000	4290500	1,6617	951	0,89356
3	3998,2	5984165	29105500	4259000	2,3837	941	0,87951
4	3983,9	5999765	29969000	4298000	1,4898	952	0,85879
5	3994,85	5993530	28950000	4216500	3,3578	839	0,86542
Rata-rata				4261600	2,3241	904,2	0,874272

$(\omega_{max}, \omega_{min})$ 0,7 ; 0,3							
No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3935,5	5986270	29963500	4241500	2,7848	941	0,90023
2	3966	5997645	29753000	4256500	2,441	787	0,86537
3	3980,1	5995770	28781500	4238500	2,8535	680	0,88166
4	3934,5	5997865	29667000	4199000	3,7589	980	0,861
5	3993,6	5994575	29514000	4247500	2,6473	663	0,87741
Rata-rata				4236600	2,8971	810,2	0,877134
$(\omega_{max}, \omega_{min})$ 0,7 ; 0,1							
No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3949,9	5958655	28387500	4070500	6,7041	843	0,89706
2	3949,4	5996480	28637000	4202600	3,6764	943	0,81673
3	3996,2	5987785	29580500	4222500	3,2203	989	0,85492
4	3934,9	5997205	29554500	4218100	3,3211	710	0,82037
5	3995,5	5964245	28693000	4228000	3,0942	930	0,83353
Rata-rata				4188340	4,00322	883	0,844522
$(\omega_{max}, \omega_{min})$ 0,5 ; 0,3							
No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3864,1	5986705	28533000	4131800	5,2991	931	0,79712
2	3983,5	5980300	29358000	4220500	3,2661	967	0,86961
3	3999,4	5955330	27170500	4079000	6,5093	912	0,79973
4	3957,1	5997630	29999000	4291400	1,6411	986	0,87336
5	3956,9	5987865	28795500	4199000	3,7589	996	0,8969
Rata-rata				4184340	4,0949	958,4	0,847344
$(\omega_{max}, \omega_{min})$ 0,5 ; 0,1							
No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3987,2	5986195	29429000	4117500	5,6269	999	0,8222
2	3999,2	5998310	28057000	4115800	5,6658	990	0,83554
3	3979,55	5987525	27914500	4103700	5,9432	617	0,80704
4	3989,75	5994620	28494000	4023600	7,7791	837	0,8377
5	3963,85	5994805	28743000	4058300	6,9837	759	6,9837
Rata-rata				4083780	6,39974	840,4	0,857236
$(\omega_{max}, \omega_{min})$ 0,3 ; 0,1							
No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3991,7	5913840	26865500	3785800	13,2294	902	0,82937
2	3922,4	5989235	27454500	3711900	14,9232	590	0,80283
3	3828,9	5786195	25705000	3627100	16,8668	189	0,87039
4	3904,9	5928595	26590500	3745800	14,1462	232	0,87035
5	3927	5899290	27174000	3793500	13,0529	72	0,90089
Rata-rata				3732820	14,4437	397	0,854766

Algoritma CSO-PSO

 $(\omega_{max}, \omega_{min})$ 0,9 ; 0,7

No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3999,3	5996955	29991500	4350000	0,29796	638	3,3602
2	3992,9	5996565	29993000	4328000	0,8022	674	2,9933
3	3977,3	5979505	29990500	4333500	0,67614	956	3,0118
4	3999,8	5980265	29993000	4338000	0,573	940	3,1114
5	3996,4	5985665	29998500	4345500	0,4011	929	3,3185
Rata-rata				4339000	0,55008	827,4	3,15904

 $(\omega_{max}, \omega_{min})$ 0,9 ; 0,5

No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3983,2	5990625	29999500	4343500	0,44694	992	2,9566
2	3983,7	5994085	29946500	4329500	0,76782	793	3,0127
3	3999,4	5997585	29986500	4343500	0,44694	962	3,001
4	3995,5	5967135	29976500	4331000	0,73344	844	3,0474
5	3998,7	5972095	29997500	4325500	0,8595	760	3,0194
Rata-rata				4334600	0,650928	870,2	3,00742

 $(\omega_{max}, \omega_{min})$ 0,9 ; 0,3

No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3997,6	5998975	29984500	4331000	0,73344	869	2,9925
2	3993,9	5999705	29991000	4332500	0,69906	876	3,1081
3	3997,5	5998295	29972000	4309500	1,2262	994	3,0304
4	3998,1	5972725	29993500	4330500	0,7449	953	3,3749
5	3998,1	5997645	29892000	4336000	0,61884	925	2,9453
Rata-rata				4327900	0,804488	923,4	3,09024

 $(\omega_{max}, \omega_{min})$ 0,9 ; 0,1

No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3998,1	5971295	29998000	4339500	0,53862	912	3,0492
2	3970,2	5989925	29987500	4299000	1,4669	829	3,0555
3	3994,8	5980545	29995500	4318500	1,0199	712	2,9752
4	3992,8	5999405	29983500	4320500	0,9741	578	2,9444
5	3998,65	5967715	29995000	4280500	1,8909	953	2,889
Rata-rata				4311600	1,178084	796,8	2,98266

 $(\omega_{max}, \omega_{min})$ 0,7 ; 0,5

No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3997,8	5989145	29978500	4340000	0,52716	999	3,4638
2	3998,1	5985675	29998000	4343500	0,44694	937	3,0262
3	3999,9	5994105	29997500	4337500	0,58446	796	3,3446
4	3975,3	5998505	29996000	4330500	0,7449	940	3,0426
5	3995,7	5973445	29943000	4340000	0,52716	915	2,9596
Rata-rata				4338300	0,566124	917,4	3,16736

$(\omega_{max}, \omega_{min})$ 0,7 ; 0,3							
No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3988	5998845	29938000	4351500	0,26358	966	2,9396
2	3998,4	5997645	29986000	4351500	0,26358	990	3,1345
3	3999	5987005	29997500	4320000	0,98556	739	3,0665
4	3997,1	5992785	29999500	4332000	0,71052	886	3,0462
5	3970,2	5998080	29984500	4305000	1,3294	671	3,0021
Rata-rata				4332000	0,710528	850,4	3,03778
$(\omega_{max}, \omega_{min})$ 0,7 ; 0,1							
No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3989,9	5999525	29986500	4267500	2,1889	914	3,085
2	3998,1	5972875	29971500	4337500	0,58446	899	2,9553
3	3999,2	5972545	29992500	4324000	0,89388	739	3,1602
4	3979,3	5998545	29992000	4340000	0,52716	989	2,9545
5	3997,7	5997545	29992000	4330500	0,7449	993	2,965
Rata-rata				4319900	0,98786	906,8	3,024
$(\omega_{max}, \omega_{min})$ 0,5 ; 0,3							
No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3987,4	5996745	29993500	4290500	1,6617	920	2,9944
2	3991,5	5952325	29999500	4312000	1,1689	420	2,8989
3	3982,4	5996755	29997500	4345000	0,41256	655	2,9297
4	3997,2	5976765	29996500	4330000	0,75636	576	3,1737
5	3993,6	5999235	29982000	4310000	1,2148	823	2,9672
Rata-rata				4317500	1,042864	678,8	2,99278
$(\omega_{max}, \omega_{min})$ 0,5 ; 0,1							
No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3991	5999825	29686500	4301500	1,4096	958	2,9928
2	3997,7	5993645	29981500	4305500	1,3179	881	3,0077
3	3997,3	5982765	29999500	4331000	0,73344	711	2,9285
4	3997,8	5999645	29987000	4329500	0,76782	730	3,1205
5	3975	5999005	30000000	4251500	2,5556	510	3,0572
Rata-rata				4303800	1,356872	758	3,02134
$(\omega_{max}, \omega_{min})$ 0,3 ; 0,1							
No	Berat (kg)	Volume (cm3)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3997,3	5991125	29997000	4329000	0,77928	930	3,2981
2	3999,4	5998285	29999500	4325500	0,8595	854	3,106
3	3994,6	5981105	29996500	4335500	0,6303	751	3,3152
4	3989,6	5995635	29561500	4254500	2,4868	526	3,1772
5	3997,3	5983365	29999000	4278000	1,9482	817	3,386
Rata-rata				4304500	1,340816	775,6	3,2565

C.6 Uji parameter N

Algoritma PSO

N 25							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3994,75	5995115	29095500	4153000	4,8132	952	0,92993
2	3973,6	5999755	29880000	4279500	1,9138	831	0,90797
3	3988,8	5988530	29988000	4247500	2,6473	989	0,96533
4	3989,95	5956110	29511000	4222500	3,2203	999	0,94736
5	3965,6	5998490	29994000	4263000	2,292	894	0,91612
6	3999,1	5988130	29300500	4293500	1,5929	824	0,89912
7	3992,8	5984925	29941000	4320000	0,98556	991	0,94702
8	3976,6	5999175	29985500	4263500	2,2805	866	0,94899
9	3999,9	5994055	28245500	4140000	5,1112	981	0,98934
10	3934,2	5998865	29638500	4275500	2,0055	961	0,92942
Rata-rata				4245800	2,686226	928,8	0,93806
N 50							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3982,6	5996135	29988500	4318500	1,0199	910	1,7373
2	3951,8	5992820	29992500	4251500	1,5556	901	1,8216
3	3955,2	5997225	28920000	4275500	2,0055	950	1,6944
4	3997,65	5998395	29276500	4259500	2,3722	929	1,7945
5	3995	5998925	29644000	4317000	1,0543	982	1,7627
6	3994,8	5949005	29958000	4319500	0,99702	968	1,7522
7	3995,8	5992170	29563500	4283000	1,8336	984	1,7296
8	3984,5	5998325	29331000	4300500	1,4325	980	1,7328
9	3997,9	5984285	28833000	4247500	2,6473	999	1,7479
10	3995,1	5998210	29677000	4246000	2,6816	988	1,907
Rata-rata				4281850	1,759952	959,1	1,768
N 100							
No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3995,3	5956965	29894000	4300500	1,4325	990	3,5712
2	3999,6	5978305	29992500	4296000	1,5356	972	3,356
3	3998,3	5973525	29997000	4321500	0,95118	734	3,4657
4	3959,9	5983195	29962000	4306000	1,3064	973	3,4364
5	3995,3	5999955	29993500	4263000	2,292	903	3,5437
6	3974,1	5999525	29487500	4293500	1,5929	898	3,4603
7	3998,7	5985455	30000000	4316500	1,0658	894	3,8782
8	3997,95	5983555	29982000	4317000	1,0543	694	3,7122
9	3966,1	5993245	29999500	4321000	0,96264	837	3,6898
10	3997,3	5986725	29987000	4352000	0,25212	975	3,925
Rata-rata				4308700	1,244544	887	3,60385

N 200

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3993,7	5982185	29953500	4320500	0,9741	959	7,5575
2	3994,5	5957565	29895000	4337000	0,59592	994	7,71
3	3994,5	5957565	29895000	4337000	0,59592	994	7,71
4	3998,7	5991625	29976000	4293000	1,6044	908	8,7582
5	3987,9	5995365	29995000	4335500	0,6303	845	8,4749
6	3984,1	5998955	29985000	4320000	0,98556	813	8,1431
7	3997,8	5986925	29958500	4334500	0,65322	946	8,174
8	3993,4	5999405	29983500	4314000	1,1231	878	8,473
9	3996,1	5994205	29999500	4331500	0,72198	731	8,174
10	3940,9	5998245	29984500	4316500	1,0658	851	8,2322
Rata-rata				4323950	0,89503	891,9	8,14069

Algoritma CSO

N 25

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3984,85	5999585	29455500	4000500	8,3085	993	1,0764
2	3950,9	5998655	29685500	4164800	4,5427	975	1,0951
3	3942,9	5996505	29330000	4168800	4,4511	913	1,1382
4	3999,25	5991780	28848000	4090500	6,2457	959	1,1354
5	3977,9	5994565	28833500	4038000	7,449	452	1,046
6	3989,95	5911790	26886500	4077000	6,5551	996	1,0246
7	3987,9	5992540	27620000	4106500	5,879	205	0,82861
8	3999,9	5978585	28402000	4057000	1,0135	204	1,3104
9	3995,3	5997465	28036500	4086900	6,3282	434	1,0441
10	3981,3	5999475	29731500	4225500	3,1515	835	1,0991
Rata-rata				4101550	5,39243	696,6	1,079791

N 50

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3999,6	5995190	29463500	4184000	4,1027	812	2,0153
2	3980,2	5999085	27898000	3971500	8,9732	323	1,5634
3	3959,5	5986230	27822000	3999600	8,3291	990	4,9589
4	3985,1	5998875	29401000	4071500	6,6812	409	1,8453
5	3985,25	5994780	29598000	4118000	5,6154	620	1,788
6	3997,1	5955225	28813500	4162400	4,5978	965	1,958
7	3990,2	5994270	29382500	4123000	5,5008	1000	1,8262
8	3991,5	5996860	29963000	4083000	6,4176	434	1,8429
9	3997,1	5938155	27558500	4052000	7,1281	672	1,8227
10	3988,4	5994215	28588500	4089500	6,2686	253	1,5807
Rata-rata				4085450	6,36145	647,8	2,12014

N 100

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3998,2	5978300	28626500	4132500	5,2831	259	2,8172
2	3999,2	5985185	29788000	4177000	4,2631	570	3,2476
3	3979,6	5993070	29334500	4150500	4,8705	974	3,8196
4	3998,2	5971930	29328500	4208000	3,5526	789	3,5486
5	3985,8	5997735	29750500	4125000	5,455	833	3,7387
6	3998	5988780	29653000	4127500	5,3977	906	3,679
7	3983,4	5994225	29993500	4269500	2,143	941	4,0022
8	3998,9	5990215	29144000	4164000	4,5611	668	3,8116
9	3987,2	5989435	29130500	4249500	2,6014	995	3,9752
10	3982,95	5995335	29888000	4090000	6,2572	338	3,5327
Rata-rata				4169350	4,43847	727,3	3,61724

N 200

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3998	5956995	28717000	4158000	4,6986	514	6,0935
2	3977	5993425	28617500	4082100	6,4382	343	5,5157
3	3977	5993425	28617500	4082100	6,4382	343	5,5157
4	3999,4	5989415	29952000	4157000	4,7215	616	7,5703
5	3990,1	5997945	29825000	4263500	2,2805	236	5,7735
6	3998,8	5986935	29933000	4307500	1,2721	732	6,8518
7	3991,7	5998690	29989000	4222000	3,2317	939	7,3595
8	3998,5	5964355	29909000	4267500	2,1889	657	7,3451
9	3994,8	5990855	29294000	4257500	2,4181	437	6,3402
10	3987,6	5995935	29667000	4247000	2,6587	622	7,2672
Rata-rata				4204420	3,63465	543,9	6,56325

Algoritma CSO-PSO

N 25

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3983	5989885	29950000	4308000	1,2606	978	1,2845
2	3969	5995475	29831500	4290000	1,6732	974	1,2997
3	3994,9	5994155	29894000	4294500	1,57	977	1,2878
4	3995,6	5991585	29796500	4292000	1,6273	977	1,2909
5	3990,8	5994835	29890000	4314000	1,1231	997	1,4476
6	3996,5	5983860	29929500	4280000	1,9024	999	1,2837
7	3937,4	5925325	29883000	4308000	1,2606	999	1,589
8	3990,8	5984085	29998000	4338000	0,573	900	1,6228
9	3993,2	5987505	29558000	4304500	1,3408	996	1,387
10	3996,6	5991065	29479000	4312000	1,1689	981	1,3158
Rata-rata				4304100	1,34999	977,8	1,38088

N 50

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3992,7	5979165	29859000	4329500	0,76782	911	2,3198
2	3997,7	5982075	29931000	4313500	1,1345	950	2,2573
3	3999	5992395	29951000	4307500	1,2721	1000	2,3293
4	3999,95	5987875	29926000	4289000	1,6961	929	2,5609
5	3968	5999185	29985500	4332000	0,71052	893	2,2586
6	3994,8	5960985	29939000	4323000	0,9168	977	2,1768
7	3997,6	5973795	29928500	4303000	1,3752	971	2,281
8	3998,8	5983265	29938500	4335000	0,64176	859	2,2441
9	3994,9	5992085	29977500	4340000	0,52716	963	2,2247
10	3958,5	5998365	29998000	4297000	1,5127	944	2,4617
Rata-rata				4316950	1,055466	939,7	2,31142

N 100

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3995,3	5997715	29979000	4338500	0,56154	990	4,1278
2	3994,3	5974365	29987000	4339000	0,55008	962	4,0357
3	3995,9	5994925	29999500	4331000	0,73344	979	4,2173
4	3997,7	5983525	29958000	4334500	0,65322	831	4,0428
5	3985,6	5997475	29733500	4310000	1,2148	966	4,3552
6	3996,6	5993725	29919500	4345000	0,41256	1000	4,1291
7	3994,6	5948225	29954000	4335000	0,64176	1000	4,318
8	3997	5995545	29980500	4321500	0,95118	792	4,6153
9	3980,6	5982585	29983500	4336500	0,60738	999	4,341
10	3998,8	5995305	29984000	4343000	0,4584	847	5,1298
Rata-rata				4333400	0,678436	936,6	4,3312

N 200

No	Berat (kg)	Volume (cm ³)	Beli (Rp)	Profit (Rp)	Deviasi Perbedaan Profit (%)	Iterasi Konvergen	Waktu Komputasi
1	3997,9	5980295	29971000	4339000	0,55008	933	7,8388
2	3998,5	5999045	29974000	4339500	0,53862	889	7,6
3	3998,5	5999045	29974000	4339500	0,53862	889	7,6
4	3997,5	5990305	29988500	4340000	0,52716	997	7,9454
5	3989,6	5994085	29985000	4337500	0,58446	958	8,803
6	3998,5	5998865	29998500	4338500	0,56154	969	7,8163
7	3993,3	5961105	29990000	4356500	0,14898	885	8,2238
8	3964,5	5986205	29982000	4327500	0,81366	947	7,8809
9	3979,2	5975215	29998000	4348500	0,3324	962	8,3933
10	3993,1	5986845	29995500	4329500	0,76782	824	8,4624
Rata-rata				4339600	0,536334	925,3	8,05639