Majalah Ilmiah Matematika dan Statistika Volume 19 Nomor 2, 2019, 65 – 74
https://jurnal.unej.ac.id/index.php/MIMS/index ISSN 1411-6669

IMPLEMENTASI ALGORITMA *GREY WOLF OPTIMIZER* (GWO) DI TOKO CITRA TANI JEMBER

(*Implementation of the Grey Wolf Optimizer* (*GWO) Algorithm at Citra Tani Jember Store*)

Vidiyanti Lestari, Ahmad Kamsyakawuni, Kiswara Agung Santoso

Jurusan Matematika, Fakultas MIPA, Universitas Jember
Jl. Kalimantan 37 Jember 68121, Indonesia
E-mail: vidiyanti.lestari10@gmail.com, {kamsyakawuni, kiswara}.fmipa@unej.ac.id

Abstract. Generally, optimization is defined as the process of determining the minimum or maximum value that depends on the function of the goal, even now there are many problems regarding optimization. One of them is the problem regarding the selection of goods to be included in a limited storage medium called Knapsack problem. Knapsack problems have different types and variations. This study will solve the problem of bounded knapsack multiple constraints by implementing the Grey Wolf Optimizer (GWO) algorithm. The problem of bounded knapsack multiple constraints has more than one subject with the items that are inserted into the dimension storage media can be partially or completely inserted, but the number of objects is limited. The aim of this study is to determine the results of using the Grey Wolf Optimizer (GWO) algorithm for solving the problem of multiple constraints bounded knapsack and compare the optimal solutions obtained by the simplex method using the Solver Add-In in Microsoft Excel. The data used in this study is primary data. There are two parameters to be tested, namely population parameters and maximum iteration. The test results of the two parameters show that the population parameters and maximum iterations have the same effect, where the greater the value of the population parameters and the maximum iteration, the results obtained are also getting closer to the optimal value. In addition, based on the results of the final experiment it is known that the comparison of the results of the GWO algorithm and the simplex method has a fairly small percentage deviation which indicates that the GWO algorithm produces results that are close to the optimal value.

Keywords: GWO algorithm, Knapsack, Multiple Constraints Bounded Knapsack.
MSC 2010: 65C20

1. Pendahuluan

Umumnya, optimasi didefinisikan sebagai proses menentukan nilai minumum atau maksimum yang bergantung pada fungsi tujuannya. Saat ini banyak ditemukan permasalahan yang menyangkut permasalahan optimasi. Sebagai contoh yaitu penyusunan jadwal perkuliahan, penentuan rute kendaraan umum, dan penentuan jumlah pekerja pada sebuah proses produksi. Masalah optimasi lain yang menarik untuk dibahas adalah permasalahan yang berhubungan dengan pemilihan barang-barang untuk

dimasukkan ke dalam media penyimpanan yang bersifat terbatas. Oleh karena itu, untuk mengatasi permasalahan tersebut diperlukan sebuah cara pemilihan barang-barang yang telah ditentukan, dimana permasalahan ini dinamakan permasalahan *knapsack*. Pisinger [1] membagi permasalahan *knapsack* dalam tiga jenis, yaitu permasalahan *knapsack* 0-1, permasalahan *bounded knapsack*, dan permasalahan *unbounded knapsack*. Pengelompokan tersebut didasarkan pada pola penyimpanan barang dengan bobot dan nilai yang bervariasi. Berdasarkan permasalahan diatas nantinya dapat ditemukan sebuah cara pemilihan barang yang akan dimasukkan dalam suatu media dengan tujuan memberikan hasil optimum, tapi tidak melebihi kemampuan suatu media untuk menampungnya. Selain permasalahan *knapsack* diatas, juga terdapat berbagai variasi dari permasalahan *knapsack.* Variasi dari permasalahan *knapsack* dibagi menjadi empat jenis, yaitu *multi objective knapsack problem*, *multidimensional* atau *multiple constraints knapsack problem*, *multi knapsack problem*, dan *quadratic knapsack problem*.

Terdapat berbagai metode yang dapat digunakan untuk menyelesaikan masalah *knapsack*. Adapun penelitian yang telah membahas permasalahan *bounded knapsack* adalah penelitian dari Hadi [4] menggunakan algoritma genetika *hybrid*. Menurut Hadi [4], *bounded knapsack* adalah permasalahan pengambilan sebagian atau semua objek dari beberapa objek yang jumlahnya terbatas. Hasil penelitiannya menunjukkan bahwa algoritma genetika *hybrid* dapat diimplementasikan pada permasalahan *bounded knapsack* dengan baik. Jika dibandingkan dengan algoritma genetika, algoritma genetika *hybrid* mampu menghasilkan profit lebih maksimal dan *running time* lebih cepat. Sementara penelitian yang telah membahas mengenai permasalahan *multiple consraints bounded knapsack* adalah penelitian dari Hayyu [2] menggunakan algoritma *artificial bee colony*. Algoritma ini memberikan keuntungan yang maksimal dimana semakin besar jumlah iterasi serta jumlah populasi maka semakin mudah mendapatkan solusi optimal yang diinginkan. Oleh karena itu, parameter jumlah populasi dan jumlah iterasi sangat berpengaruh terhadap solusi optimal. Selain algoritma genetika, algoritma genetika *hybrid*, serta algoritma *articicial bee colony* tersebut, terdapat algoritma baru yang dapat digunakan untuk menyelesaikan permasalahan *knapsack* yaitu algoritma *Grey Wolf Optimizer* (GWO). Algoritma *Grey Wolf Optimizer* (GWO) merupakan algoritma metaheuristik baru yang mampu memberikan hasil kompetitif, dimana eksplorasi *search* spesiesnya lebih luas dan juga dapat menghindari terjebaknya lokal optimum. Berdasarkan uraian diatas, peneliti akan mengkaji implementasi algoritma *Grey Wolf Optimizer* (GWO) untuk menyelesaikan permasalahan *multiple constraints bounded knapsack*. Untuk mengetahui keefektifan algoritma *Grey Wolf Optimizer* maka diperlukan adanya pembanding, dimana dalam penelitian ini menggunakan metode simpleks sebagai pembandingnya. Metode simpleks merupakan metode dasar yang digunakan untuk menyelesaikan permasalahan program linier, salah satunya pada permasalahan *multiple constraints bounded knapsack*. Oleh karena itu, hasil dari penyelesaian menggunakan algoritma *Grey Wolf Optimizer* nantinya akan dibandingkan dengan hasil dari metode simpleks.

Multiple Constraints Bounded Knapsack

Multiple constraints knapsack problem atau yang sering disebut pula dengan *multidimensional knapsack problem* adalah suatu permasalahan optimasi kombinatorial NP-hard yang terdapat pada beragam aplikasi. Pada multiple constraints knapsack problem, tiap-tiap objek atau barang memiliki batasan lebih dari satu dimensi. Batasan-batasan tersebut dapat berupa waktu, biaya, maupun pekerja. Tujuan dari adanya permasalahan ini adalah agar dapat memperoleh solusi optimal dengan mengambil kombinasi objek atau barang dengan semua batasan tidak melewati kapasitas yang tersedia. Sedangkan pada multiple constraints bounded knapsack, tiap-tiap objek atau barang memiliki batasan lebih dari satu dimensi dengan objek dapat diambil sebagian atau seluruhnya [2].

Secara matematis, formulasi multiple constraints bounded knapsack dapat dirumuskan sebagai berikut:
Fungsi Tujuan dan kendala

$$\text{Maks } Z = \sum_{j=1}^{n} p_j x_j \tag{1}$$

$$\sum_{j=1}^{n} w_j x_j \le C \tag{2}$$

$$\sum_{j=1}^{n} v_j x_j \le S \tag{3}$$

$$\sum_{j=1}^{n} b_j x_j \le M \tag{4}$$

$$x_j \in \{0,1,2,...,m\},\ j = 1,2,...,n$$

Algoritma *Grey Wolf Optimizer* (GWO)

Algoritma *Grey Wolf Optimizer* merupakan algoritma yang terinspiras oleh perilaku berburu serigala di alam. Serigala abu-abu dianggap sebagai predator puncak, yang berarti bahwa serigala abu-abu berada di puncak rantai makanan. Serigala abu-abu juga memiliki hirarki dominan sosial yang tinggi. Para pemimpin yang merupakan tingkatan pertama akan disebut sebagai alfa, tingkatan kedua yaitu beta, tingkatan ketiga yaitu delta, sementara tingkatan terakhir yaitu omega. Selain hirarki sosial serigala, berburu secara berkelompok adalah perilaku menarik lain dari serigala abu-abu. Menurut Muro, *dkk.,* [3], fase utama berburu serigala abu-abu adalah sebagai berikut.

Pelacakan, mengejar, mendekati, dan mengacau mangsa sampai berhenti bergerak lalu menyerang mangsa.Adapun secara rinci perilaku serigala abu-abu dalam berburu dibagi kedalam beberapa tahapan yaitu, melingkari mangsa, berburu, menyerang mangsa (ekploitasi dan menyerang mangsa (eksplorasi). Seperti disebutkan diatas, serigala abu-

abu mengelilingi mangsanya selama berburu. Adapun model matematisnya adalah sebagai berikut.

$$D = \left| C \cdot X_p(t) - X(t) \right| \quad (5)$$

$$X(t+1) = X_p(t) - A \cdot D \quad (6)$$

Vektor A dan C dihitung mengikuti

$$A = 2a \cdot r_1 - a \quad (7)$$

$$C = 2a \cdot r_2 \quad (8)$$

Sementara untukberburu, model matematisnya adalah sebagai berikut:

$$D_\alpha = \left| C_1 \cdot X_\alpha - X \right|, D_\beta = \left| C_2 \cdot X_\beta - X \right|, D_\delta = \left| C_3 \cdot X_\delta - X \right| \quad (10)$$

$$X_1 = X_\alpha - A_1 \cdot (D_\alpha), X_2 = X_\beta - A_2 \cdot (D_\beta), X_3 = X_\delta - A_3 \cdot (D_\delta) \quad (11)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad (12)$$

Menurut Mirjalili., *et al.* [7] *pseudocode* dari algoritma GWO dapat digambarkan sebagai berikut:

Inisialisasi populasi serigala abu-abu $X_i(1,2,..., n)$

Inisialisasi a, A, dan C

Hitung kesesuaian setiap perantara pencari

X_α = perantara pencari terbaik

X_β = perantara pencari terbaik kedua

X_δ = perantara pencari terbaik ketiga

while (t<jumlah maksimal iterasi)

for setiap perantara pencari

Perbarui posisi perantara pencari saat ini dengan persamaan (12)

end for

Perbarui a, A, dan C

Hitung $X_\alpha, X_\beta, X_\delta$

end while

2. Metodologi

Data penelitian yang digunakan pada penelitian ini adalah data sejumlah barang yang dijual di Toko Citra Tani. Toko Citra Tani adalah toko yang menjual bermacam-macam bahan bangunan seperti cat, tinner, semen dan masih banyak lagi lainnya. Toko Citra Tani terletak di Jl. PB Sudirman No.74 Kecamatan Panti Kabupaten Jember. Data yang diambil berupa nama barang, jumlah barang, satuan, berat satuan, volume barang, harga beli, dan harga jual Kapasitas maksimum dari knapsack adalah sebesar 5.000 kg, volume

maksimum dari knapsack adalah sebesar 9.000.000 , dan modalnya sebesar Rp 20.000.000,-. Sementara data harga jual dan harga beli nantinya digunakan untuk menghitung keuntungan.

Langkah-langkah penerapan algoritm *Grey Wolf Optimizer* (GWO) adalah sebagai berikut:

1) Inisialisasi populasi awal
   Menginputkan jenis barang(brg), jumlah barang (n), maksimal iterasi(maks iter), populasi(pop), volume(V), berat(W), modal(M), beli(B), jual(J). Lalu kita bangkitkan bilangan acak
   X_i= random (0,1), i = 1....n
   y_i= round (X_i), i = 1...n

2) Pengecekan sulusi awal dengan menghitung berat
   Memeriksa hasil awal dari W, I, dan V. Apabila ada yang tidak memenuhi kendala maka akan dilakukan pinalti. Pemeriksaan solusi awal dilakukan dengan menggunakan rumus:

$$\sum_{j=1}^{n} w_i y_i \leq C \quad (13)$$

$$\sum_{j=1}^{n} v_i y_i \leq S \quad (14)$$

$$\sum_{j=1}^{n} b_i y_i \leq M \quad (15)$$

$$y_i \in \{0,1,2,...,m\}, i = 1,2,...,n \quad (16)$$

3) Menghitung nila*i fitne*ss.
   Menghitung total keuntungan dari masing-masing barang yang dibawa dengan menggunakan persamaan (1).

4) Mencari $X_\alpha, X_\beta, X_\delta$
   Mengurutkan nilai *fitness* yang telah didapatkan dari langkah sebelumnya menggunakan persamaan . Nilai *fitness* terbesar pertama akan dipilih sebagai X_α, nilai *fitness* terbesar kedua akan dipilih sebagai X_β, nilai *fitness* terbesar ketiga akan dipilih sebagai X_δ.

5) Menghitung a, A, dan C
   Menghitung a, A, dan C dengan menuliskan bilangan random $r_1$dan $r_2$agar dapat menghitung

6) Memperbarui posisi
   Menghitung arah perpindahan (pergerakan serigala) sesuai persamaan (10), (11), (12) lalu memperbarui posisi serigala.

7) Pengecekan solusi baru.
   Memeriksa solusi baru dari W, V, dan B. Apabila ada yang tidak memenuhi kendala maka akan dilakukan pinalti. Namun apabila sudah memenuhi kendala maka bisa melanjutkan ke langkah selanjutnya.
8) Menghitung nilai *fitness.*
   Menghitung total keuntungan terbaru dari masing-masing barang yang dibawa setelah semua diperbarui.
9) Memperbarui $X_\alpha, X_\beta, X_\delta$
   Memperbarui posisi serigala dari *fitness* terbaru yang telah didapatkan. Apabila posisi serigala yang telah diperbarui memberikan *fitness*yang lebih baik dari alfa, maka posisi alfa diperbarui. Namun jika ternyata tidak lebih baik maka posisi alfa tetap. Sama halnya dengan beta dan delta.
10) Cek kriteria pemberhentian.
    Melihat iterasi maksimal yang digunakan. Apabila sudah memenuhi maka iterasi dapat dihentikan.

3. Hasil dan Pembahasan

Penelitian ini akan membahas hasil dari implementasi algoritma *Grey Wolf Optimizer* (GWO) pada permasalahan *Multiple Constraints Bounded Knapsack* yang diterapkan pada data yang telah diambil dari Toko Citra Tani dengan menggunakan aplikasi Matlab R2015b. Program dijalankan pada Laptop dengan CPU Intel(R) Celeron(R) CPU N2840 @2,16GHz 2,16GHz 2,00GB 64 bit OS. Langkah awalnya yaitu menguji parameter untuk mengetahui hasilnya. Terdapat dua parameter yang akan diuji yaitu populasi (*pop*) dan maksimal iterasi (*max* iter). Sementara hasil dari pengujian parameternya adalah sebagai berikut:

a. Uji Parameter Populasi

Uji parameter populasi (*pop*) dilakukan dengan menggunakan lima nilai, yaitu 25, 75, 150, 250, dan 500. Populasi merepresentsikan banyaknya kandidat solusi (serigala). Sementara maksimal iterasi yang dipakai untuk pengujian populasi ini adalah 1000. Setiap kombinasi nilai parameter dilakukan 10 kali *running* program untuk kemudian dihitung rata-rata dari profit, rata-rata iterasi konvergen, dan rata-rata waktu komputasi.

Tabel 1. Hasil uji parameter populasi data

pop	Profit Rata-rata	Profit Terbaik	Profit Terburuk	Rata-rata Iterasi Konvergen	Rata-rata Waktu Komputasi
25	2568850	2625100	2433100	985,5	54,4748
50	2585140	2624100	2531400	988,5	57,1807
100	2607860	2658500	2581900	989,8	63,9717
250	2635130	2677500	2592900	990,6	67,9609
500	2636590	2669200	2553900	990,1	84,3505

b. Uji Parameter Maksimal Iterasi

Uji parameter maksimal iterasi (*max* iter) dilakukan dengan menggunakan empat nilai, yaitu 1000, 2000, 5000, dan 7000. Pengujian parameter maksimal iterasi ini menggunakan parameter populasi sebesar 500 yang merupakan nilai populasi dengan profit rata-rata terbaik. Setiap kombinasi nilai parameter dilakukan 10 kali *running* program sama halnya dengan pengujian terhadap parameter populasi. Langkah selanjutnya yaitu menghitung rata-rata profit, rata-rata iterasi saat Z maksimum lokal, dan rata-rata waktu komputasi.

Tabel 2. Hasil uji parameter maksimal iterasi data

Max Iter	Profit Rata-rata	Profit Terbaik	Profit Terburuk	Rata-rata Iterasi Konvergen	Rata-rata Waktu Komputasi
1000	2636590	2669200	2598000	990,1	84,3505
2000	2647370	2674500	2531400	1973,7	172,5286
5000	2648010	2681000	2607400	4927,5	457,3635
10000	2667320	2690800	2620000	9835,4	881,0332

Setelah dilakukan pengujian parameter, langkah selanjutnya yang harus dilakukan yaitu menhitung persentase deviasi dari data yang digunakan. Persentase ini dihitung untuk mengetahui seberapa optimal algoritma yang digunakan dalam penelitian ini yaitu algoritma *Grey Wolf Optimizer* (GWO) yang diterapkan pada permasalahan *multiple constraints bounded knapsack*. Persentasie deviasi dapat dihitung dengan cara sebagai berikut:

$$\%\text{Dev} = \frac{Simplex - Z_i}{Simplex} \times 100\%$$

Hasil dari *running time* dengan populasi sebesar 1000 dan maksimal iterasi sebesar 10000 dapat dilihat pada Tabel 3.

Algoritma *Grey Wolf Optimizer* (GWO) memiliki dua parameter yaitu populasi dan maksimal iterasi. Berdasarkan hasil uji pengaruh parameter terhadap data yang digunakan

diketahui bahwa parameter populasi dan maksimal iterasi memiliki pengaruh yang sama terhadap hasil yang didapatkan. Semakin besar nilai dari parameter populasi, maka hasil yang didapatkan juga semakin mendekati nilai optimal. Sama halnya dengan parameter maksimal iterasi, semakin besar nilai dari parameter maksimal iterasi, maka hasil yang didapatkan juga semakin mendekati nilai optimal. Hal ini dapat dilihat pada Tabel 1 dan Tabel 2 yang menunjukkan bahwa semakin meningkat nilai parameter populasi dan maksimal iterasi, nilai dari profit rata-rata juga semakin meningkat apabila diterapkan pada permasalahan *multiple constraints bounded knapsack*.

Tabel 3. Hasil akhir simulasi data dengan persentase deviasi

No	Harga Beli	Profit	Iterasi Konvergen	Waktu Komputasi	%Dev
1	19984200	2678400	9859	1302,3471	1,926034
2	19989500	2679000	9908	1311,3876	1,904064
3	19979000	2680000	9880	1193,5162	1,867448
4	19984200	2678400	9859	1175,4208	1,926034
5	19989500	2679000	9908	1406,8451	1,904064
6	19985500	2689000	9794	1201,0167	1,537898
7	19989500	2679000	9908	1446,8643	1,904064
8	19984200	2678400	9859	1186,5067	1,926034
9	19989500	2679000	9908	1182,2869	1,904064
10	19990100	2704700	9893	1166,7700	0,963017
Rata-rata		2669580	9877,6	1257,2961	1,776272

Berdasarkan hasil simulasi akhir data yang menggunakan parameter populasi sebesar 1000 dan maksimal iterasi sebesar 10000, algoritma GWO memiliki solusi yang hampir mendekati nilai optimal yang didapat dari *Simpleks Solver Add-In* pada Microsoft Excel. Profit optimal yang dihasilkan dari *Simpleks Solver Add-In* sebesar Rp 2.731.000,- Sementara profit yang didapatkan pada algoritma GWO pada tabel hasil akhir simulasi data yaitu profit terkecil sebesar Rp 2.678.400,- dengan persentase deviasi 1,926034%. Sedangkan profit terbesar yang didapatkan yaitu sebesar Rp 2.704.700,- dengan persentase deviasi sebesar 0,963017%. Adapun rata-rata profit dari hasil simulasi akhir data yaitu sebesar Rp 2.682.490,-, rata-rata persentase deviasi sebesar 1,776272%, dan rata-rata iterasi saat Z maksimum lokal yaitu 9877,6.

Solusi paling optimal yang didapatkan dari data Toko Citra Tani memiliki keuntungan sebesar Rp 2.704.700,- dengan total berat 2853 kg, total volume 8.510.527 dan total harga beli sebesar Rp 19.990.100. Adapun barang-barang yang diangkut adalah 30 Altex Emulsion Paint 1 kg, 20 Altex Emulton Paint 5 kg, 30 Altex Cat Synthetic - Kaleng Merah, 28 Altex - Meni kayu, Meni Besi, 30 Altex - Plamir Kayu 0,5 kg, 30 Altex - Plamir Kayu 1 kg, 50 Altex - Warna Khusus 1 kg, 30 Altex - Warna Khusus 5 kg, 30 Paku 3/4",

30 Paku 1", 30 Paku 1 1/4", 30 Paku 1 1/2", 30 Paku 2", 30 Paku 3", 30 Paku 4", 3 Semen Gresik, 9 Semen Holsim, 17 Semen Bosowa, 15 Semen Puger, 1 Semen Putih Gresik, 15 Mowilex, 50 Kawat, 40 Bendrat, 4 Kalsium, 4 Flintkote Oil, 8 Lem Rajawali, 30 Karbit, 19 Japen Cat, 10 Paragon, 3 Paragon Genteng.

4. Kesimpulan

Berdasarakan hasil dan pembahasan, maka dapat diperoleh kesimpulan sebagai berikut:

a. Solusi terbaik yang didapatkan oleh algoritma *Grey Wolf Optimizer* (GWO) dalam menyelesaikan permasalahan *multiple constraints bounded knapsack* adalah sebesar Rp 2.704.700,-. Solusi terbaik tersebut diperoleh dari profit paling maksimal hasil akhir simulasi data dengan parameter populasi 1000, parameter maksimal iterasi 10000 dan rata-rata iterasi saat Z maksimum lokal 9877,6. Parameter populasi dan maksimal iterasi dalam penyelesaian masalah ini memiliki pengaruh yang sama untuk mendapatkan hasil yang optimal, dimana semakin besar nilai dari parameter tersebut maka hasil yang didapatkan juga semakin mendekati nilai optimal.

b. Hasil dari algoritma *Grey Wolf Optimizer* (GWO) dibandingkan dengan Simpleks *Solver Add-In* pada Microsoft Excel memiliki rata-rata persentase deviasi sebesar 1,776272%. Namun jika dilihat dari profit terbaik, persentase deviasi yang didapat yaitu sebesar 0,963017% yang menunjukkan bahwa algoritma GWO memberikan hasil yang mendekati nilai optimal.

Daftar Pustaka

[1] Pisinger, D. (1995). *Algorithms for Knapsack Problems.* Denmark: University of Copenhagen.

[2] Hadi, I. S. (2015). Penerapan Algoritma Genetika *Hybrid* pada Permasalahan *Bounded Knapsack*. *Skripsi*. Jember: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

[3] Hayyu, A. N. (2016). Penerapan Algoritma *Artificial Bee Colony* pada Permasalahan *Multiple Constraints Bounded Knapsack*. *Skripsi*. Jember: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

[4] Muro, C., Escobedo, R., Spector, L. dan Coppinger, R. (2011). Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations, Behav Process, 88 : 192-7.

[5] Mirjalili, S., Mirjalili, M. S. dan Lewis, A. (2014). Grey Wolf Optimizer, *Advances in Engineering Software 69,* 46-61.

[6] Yazdani, Danial, Toosi, Nadjaran, A. dan Meybodi, M. R. (2010). "Fuzzy Adaptive Artificial Fish Swarm Algorithm". Qazvin, Mashhad, Tehran, Iran: IAU of Qazvin, IAU of Mashhad and Amirkabir University of Technology

[7] Zhou, W. dan Li, Y. (2010). An Improved Genetic Algorithm for Multiple Traveling Salesman Problem. pp.493– 495.

[8] Hilviah, F. (2015). Penerapan Algoritma *Dynamic Programming* dan Algoritma *Backtracking* pada Permasalahan *Multiple Constraints Knapsack* 0-1. *Skripsi*. Jember: Jurusan Matematika FMIPA Universitas Jember.

[9] Kellerer, H., Pferschy, U. dan Pisinger, D. (2004). *Knapsack Problem*. Verlag Berlin Heidelberg: Springer.

[10] Pisinger, D. (1995). *A minimal Algorithm for the Multiple-Choise Knapsack Problem*. *Europan Journal of Operational Research*, 83(2): 349-410.