

Aplikasi Algoritma Greedy Terhadap Permasalahan Integer Knapsack pada Toko Surya Muda Pekanbaru

Sri Basriati¹, Elfira Safitri², Meli Ermanita³

^{1,2,3} Jurusan Matematika, Fakultas Sains dan Teknologi, UIN Sultan Syarif Kasim Riau
Jl. HR. Soebrantas No. 155 Simpang Baru, Panam, Pekanbaru, 28293
Email: sribasriati@uin-suska.ac.id, meliermanita20@gmail.com

ABSTRAK

Permasalahan *integer knapsack* merupakan permasalahan pengangkutan atau pemilihan barang yang akan dimasukkan secara keseluruhan atau tidak sama sekali dalam satu item sehingga tidak melebihi kapasitas dari pengangkut atau wadah. Algoritma *greedy* adalah metode yang digunakan untuk menyelesaikan permasalahan optimasi. Pada penelitian ini, algoritma *greedy* digunakan untuk memilih barang yang akan diangkut oleh Toko Surya Muda Pekanbaru agar mendapatkan keuntungan yang maksimum. Berdasarkan perhitungan dari ketiga strategi dari algoritma *greedy* yaitu *greedy by profit*, *greedy by weight* dan *greedy by density* didapatkan bahwa strategi *greedy by density* menghasilkan solusi yang lebih optimal dari pada *greedy by profit* dan *greedy by weight*.

Kata Kunci: Algoritma *greedy*, *integer knapsack*, pengangkutan barang

ABSTRACT

The *integer knapsack* problem is a problem of transportation or selection of goods that will be included in whole or not in one item so that it does not exceed the capacity of the carrier or container. Greedy algorithm is a method used to solve optimization problems. In this study, the greedy algorithm is used to select goods to be transported by the Surya Muda Pekanbaru Shop in order to get the maximum profit. Based on the calculation of the three strategies of the greedy algorithm, namely *greedy by profit*, *greedy by weight* and *greedy by density*, it is found that the *greedy by density* strategy produces a solution that is more optimal than *greedy by profit* and *greedy by weight*.

Keywords: Greedy algorithm, *integer knapsack*, transportation of goods.

Pendahuluan

Permasalahan optimasi adalah permasalahan dalam pemilihan barang-barang yang akan dimasukkan ke dalam sebuah pengangkut atau wadah yang digunakan untuk mengangkut barang-barang yang akan dibeli untuk dijual kembali. Tujuan dari pemilihan barang yang akan diangkut adalah untuk memperoleh keuntungan yang maksimal. Penentuan pemilihan barang dalam permasalahan optimasi termasuk ke dalam permasalahan *knapsack*. Salah satu metode yang digunakan untuk menentukan pemilihan barang dalam permasalahan *knapsack* adalah *integer knapsack*. Permasalahan *integer knapsack* (*knapsack* 0-1) merupakan permasalahan pengangkutan barang, dimana barang akan dimasukkan secara keseluruhan atau tidak sama sekali dalam satu item, Paryati [4].

Menurut Zulhidayati [7], algoritma *greedy* adalah salah satu metode yang paling sering digunakan dalam menyelesaikan permasalahan *knapsack*. Algoritma *greedy* memecahkan masalah langkah per langkah. Setiap langkah merupakan pilihan untuk membuat langkah optimum lokal dengan harapan bahwa langkah sisanya mengarah ke solusi optimum global. Algoritma *greedy* memiliki perbedaan dengan algoritma lainnya diantaranya yaitu dari segi kecepatan dan segi ketepatan. Perhitungan algoritma *greedy* yang menggunakan komputasi akan lebih cepat dikarenakan algoritma *greedy* menggunakan prinsip pemilihan keputusan disetiap langkahnya. Algoritma *greedy* tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada sehingga algoritma *greedy* tidak selalu memberikan hasil yang optimal, akan tetapi memberikan hasil optimal ketika terdapat banyak alternatif yang diberikan seperti yang dikemukakan oleh Hasan [3].

Penelitian tentang masalah *knapsack* telah dilakukan oleh beberapa peneliti, diantaranya penelitian masalah *knapsack* oleh Setemen [6] dengan studi kasus pemilihan buah kemasan kotak dengan

menggunakan algoritma genetik dan memberikan hasil yang optimal. Selanjutnya Pratiwi [5] mengemukakan bahwa dalam pengoptimalan muatan barang dengan menggunakan algoritma *branch and bound* yang memberikan keuntungan penjualan maksimal. Menurut Ambarwari [1] pula, penerapan algoritma *greedy* pada permasalahan *knapsack* untuk optimasi pengangkutan peti kemas yang memberikan kesimpulan bahwa algoritma *greedy* memberikan keuntungan yang besar pada pemilihan peti kemas. Berdasarkan latar belakang tersebut penulis tertarik untuk mengulas kembali permasalahan *integer knapsack* dengan menggunakan algoritma *greedy* kemudian mengaplikasikannya di salah satu toko di Kota Pekanbaru.

Metode Penelitian

1. Permasalahan *Knapsack*

Knapsack diartikan sebagai sebuah tas atau wadah, permasalahan *knapsack* merupakan permasalahan tentang pemilihan barang dari sejumlah barang yang memiliki bobot atau nilai yang berbeda-beda, sehingga diperoleh keuntungan yang maksimal tanpa melebihi kapasitas dari tas atau wadah yang tersedia.

Bentuk umum dari permasalahan *integer knapsack* adalah sebagai berikut:

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

dengan kendala:

$$\sum_{i=1}^n w_i x_i \leq M, x_i \in \{0, 1\}$$

Keterangan:

Z : Nilai optimum dari nilai fungsi tujuan

n : Banyak barang

p_i : Keuntungan barang ke- i keseluruhan dimana $i = 1, 2, 3, \dots, n$

w_i : Berat barang ke- i keseluruhan, dimana $i = 1, 2, 3, \dots, n$

x_i : Variabel keputusan (1 jika dipilih, 0 jika tidak dipilih)

M : Kapasitas media pengangkut (*knapsack*)

2. Algoritma *Greedy*

Algoritma *greedy* digunakan untuk memperoleh penyelesaian dari suatu permasalahan optimasi. Suatu permasalahan dengan n masukkan data dilakukan secara bertahap. Pertama dilakukan pemilihan solusi dari ketiga strategi kemudian himpunan solusi dari ketiga strategi tersebut akan diperoleh solusi yang paling optimal, Hasan [3]. Menurut Brassard [2], algoritma *greedy* adalah algoritma untuk menyelesaikan permasalahan secara bertahap. Terdapat beberapa strategi *Greedy* yang dapat digunakan untuk memilih objek yang akan dimasukkan ke dalam M antara lain:

a. *Greedy by profit*

Knapsack diisi dengan objek yang mempunyai keuntungan terbesar. Strategi ini mencoba memaksimalkan keuntungan dengan memilih objek yang paling menguntungkan terlebih dahulu. Pertama kali dilakukan adalah mengurutkan secara menurun objek-objek berdasarkan *profit*nya. Kemudian objek-objek yang dapat ditampung oleh *knapsack* diambil satu persatu sampai *knapsack* penuh atau sudah tidak ada objek lagi yang bisa dimasukkan.

b. *Greedy by weight*

Knapsack diisi dengan objek yang mempunyai berat paling ringan. Strategi ini mencoba memaksimalkan keuntungan dengan memasukkan sebanyak mungkin objek ke dalam *knapsack*. Pertama kali dilakukan adalah mengurutkan secara menaik objek-objek berdasarkan beratnya. Kemudian objek-objek yang dapat ditampung oleh *knapsack* diambil satu persatu sampai *knapsack* atau sudah tidak ada objek lagi yang bisa dimasukkan.

c. *Greedy by density*

Knapsack diisi dengan objek yang mempunyai p_i/w_i terbesar dimana p adalah keuntungan dan w adalah berat barang. Strategi ini mencoba memaksimalkan keuntungan dengan memilih objek yang mempunyai keuntungan per unit (*density*) berat terbesar. Pertama kali yang dilakukan adalah mencari nilai profit per *weight density* dari tiap-tiap objek. Kemudian objek-objek diurutkan berdasarkan *density* terbesar. Kemudian objek-objek yang dapat ditampung oleh *knapsack* diambil satu persatu sampai *knapsack* penuh atau sudah tidak ada objek lagi yang bisa dimasukkan.

Hasil dan Pembahasan

1. Data Pengangkutan Barang Toko Surya Muda Pekanbaru

Data yang diambil dari Toko Surya Muda Pekanbaru berupa data berat barang perunit, banyaknya barang yang dibeli, harga beli barang dan harga jual barang. Data tersebut dapat dilihat pada Tabel 1 sebagai berikut:

Tabel 1 Daftar Barang yang diangkut

No.	Nama Barang ke- i	Berat Barang Perunit (Kg)	Banyak Barang (unit)	Harga Beli (Rp)	Harga Jual (Rp)
1	GULA	50	1	570.000	650.000
2	TEPUNG TERIGU	25	1	195.000	212.500
3	GARAM	50	1	575.000	690.000
⋮	⋮	⋮	⋮	⋮	⋮
38	½ LUSIN TEH PUCUK	4,5	2	144.000	192.000
39	1 LUSIN M-150	2	1	39.600	60.000
40	PLASTIK KACA	¼	20	150.000	180.000

Sumber: Toko Surya Muda Pekanbaru (2019)

Berdasarkan Tabel 1, untuk mempermudah penyelesaian permasalahan integer *knapsack* maka tabel dapat disederhanakan dengan memisalkan dengan simbol, seperti berikut ini:

Nama barang ke-1 disimbolkan dengan angka **1**;

Nama barang ke-2 disimbolkan dengan angka **2**;

Nama barang ke-3 disimbolkan dengan angka **3**;

⋮

Nama barang ke-40 disimbolkan dengan angka **40**.

2. Mengidentifikasi Data Berdasarkan Berat dan Keuntungan

Identifikasikan data untuk memperoleh berat w_i dan keuntungan p_i . w_i diperoleh dari mengalikan bobot berat barang perunit dengan banyak barang, dan p_i diperoleh dari selisih harga jual dan harga beli, hasil identifikasi dapat dilihat pada Tabel 2 sebagai berikut:

Tabel 2. Daftar Identifikasi dari Tabel 1

No.	Barang ke- i	w_i (Kg)	p_i (Rp)
1	1	50	80.000
2	2	25	17.500
3	3	50	115.000
⋮	⋮	⋮	⋮
38	38	9	48.000
39	39	2	20.400
40	40	5	30.000

3. Penyusunan Model Permasalahan Integer Knapsack

Berdasarkan Tabel 2, dapat disusun model permasalahan *integer knapsack*. Permasalahan *integer knapsack* terdapat beberapa variabel yang akan digunakan, fungsi kendala dan fungsi tujuan. Adapun variabel yang akan digunakan, fungsi tujuan dan fungsi kendala adalah sebagai berikut:

a. Menentukan variabel yang akan digunakan

p_i : Keuntungan barang ke- i keseluruhan

w_i : Berat barang ke- i keseluruhan

x_i : Barang ke- i yang dipilih (1 jika dipilih, 0 jika tidak dipilih)

M : Kapasitas media pengangkut (*knapsack*)

- b. Menentukan fungsi tujuan
 Maksimum $Z = \sum_{i=1}^n p_i x_i$
 c. Menentukan fungsi kendala
 $w_i x_i \leq M$

Berdasarkan model permasalahan *integer knapsack*, sehingga diperoleh formulasi untuk permasalahan *integer knapsack* adalah sebagai berikut:

Fungsi Tujuan:

Maksimum

$$Z = 80.000x_1 + 17.500x_2 + 115.000x_3 + 24.000x_4 + 15.000x_5 + 15.000x_6 + 10.000x_7 + 20.000x_8 + 20.000x_9 + 5.000x_{10} + 5.000x_{11} + 20.000x_{12} + 24.000x_{13} + 20.000x_{14} + 12.000x_{15} + 20.000x_{16} + 2.000x_{17} + 49.000x_{18} + 6.000x_{19} + 20.000x_{20} + 18.000x_{21} + 105.000x_{22} + 42.500x_{23} + 50.000x_{24} + 148.000x_{25} + 8.000x_{26} + 4.000x_{27} + 12.000x_{28} + 3.000x_{29} + 6.000x_{30} + 9.000x_{31} + 24.000x_{32} + 24.000x_{33} + 12.000x_{34} + 24.000x_{35} + 12.000x_{36} + 12.000x_{37} + 48.000x_{38} + 20.400x_{39} + 30.000x_{40}$$

Dengan fungsi kendala :

$$50x_1 + 25x_2 + 50x_3 + 12x_4 + 10x_5 + 10x_6 + 10x_7 + 10x_8 + 10x_9 + 5x_{10} + 5x_{11} + 5x_{12} + 12x_{13} + 5x_{14} + 4x_{15} + 5x_{16} + 2x_{17} + 28x_{18} + 6x_{19} + 10x_{20} + 18x_{21} + 95x_{22} + 55x_{23} + 55x_{24} + 28x_{25} + 9x_{26} + 4x_{27} + 7x_{28} + 10x_{29} + 17x_{30} + 3x_{31} + 7x_{32} + 9x_{33} + 6x_{34} + 9x_{35} + 6x_{36} + 4x_{37} + 9x_{38} + 2x_{39} + 5x_{40} \leq 500$$

$$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}, x_{20}, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}, x_{27}, x_{28}, x_{29}, x_{30}, x_{31}, x_{32}, x_{33}, x_{34}, x_{35}, x_{36}, x_{37}, x_{38}, x_{39}, x_{40} \in \{0,1\}$$

4. Pengaplikasian Algoritma Greedy

Langkah 1: Barang-barang yang terdapat pada Tabel 2 diurutkan berdasarkan *profit* (p_i) terbesar, selanjutnya dengan mengurutkan daftar barang-barang yang ada pada Tabel 2 sehingga diperoleh tabel sebagai berikut:

Tabel 3. Urutan Daftar Barang Berdasarkan *Profit* Terbesar

No	Barang ke- i	w_i (Kg)	p_i (Rp)
1	25	28	148.000
2	3	50	115.000
3	22	95	105.000
⋮	⋮	⋮	⋮
38	27	4	4.000
39	29	10	3.000
40	17	2	2.000

Langkah 2: Mengambil satu persatu barang yang terdapat pada Tabel 3 yang dapat diangkut tanpa melebihi kapasitas *knapsack*. Diketahui bahwa kapasitas maksimum pengangkut sebesar **500 kg** dengan menggunakan strategi *Greedy by profit* maka diperoleh tabel pengangkutan barang ke dalam permasalahan *knapsack* sebagai berikut:

Tabel 4. Pengangkutan Barang ke dalam Permasalahan *Knapsack* Menggunakan Strategi *Greedy by Profit*

No	Barang ke- i	w_i (Kg)	p_i (Rp)	Status	x_i
1	25	28	148.000	Diambil	1
2	3	50	115.000	Diambil	1
3	22	95	105.000	Diambil	1
⋮	⋮	⋮	⋮	⋮	⋮
38	27	4	4.000	Tidak	0

39	29	10	3.000	Tidak	0
40	17	2	2.000	Tidak	0

Berdasarkan Tabel 4 diperoleh

$x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1, x_5 = 1, x_6 = 0, x_7 = 0, x_8 = 1, x_9 = 1, x_{10} = 0, x_{11} = 0,$
 $x_{12} = 1, x_{13} = 1, x_{14} = 1, x_{15} = 0, x_{16} = 1, x_{17} = 0, x_{18} = 1, x_{19} = 0, x_{20} = 1, x_{21} = 1,$
 $x_{22} = 1, x_{23} = 1, x_{24} = 1, x_{25} = 1, x_{26} = 0, x_{27} = 0, x_{28} = 0, x_{29} = 0, x_{30} = 0, x_{31} = 0,$
 $x_{32} = 1, x_{33} = 1, x_{34} = 0, x_{35} = 1, x_{36} = 0, x_{37} = 0, x_{38} = 1, x_{39} = 1, x_{40} = 1$

Solusi optimal permasalahan ini adalah

$X = \{1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0,$
 $0, 1, 1, 0, 1, 0, 0, 1, 1, 1\}$

yang artinya barang ke-1, 3, 4, 5, 8, 9, 12, 13, 14, 16, 18, 20, 21, 22, 23, 24, 25, 32, 33, 35, 38, 39 dan 40 diambil dengan total keuntungan maksimum sebesar Rp 960.900,-.

a. Menggunakan strategi *Greedy by Weight* dalam menyelesaikan permasalahan *knapsack*

Langkah 1: Barang-barang yang ada pada Tabel 2 diurutkan berdasarkan *weight*-nya atau berat terkecil. Selanjutnya dengan mengurutkan daftar barang-barang yang ada pada Tabel 2 sehingga diperoleh tabel sebagai berikut:

Tabel 5. Urutan Daftar Barang Berdasarkan Berat Terkecil

No.	Barang ke- i	w_i (Kg)	p_i (Rp)
1	17	2	2.000
2	39	2	20.400
3	31	3	9.000
⋮	⋮	⋮	⋮
38	23	55	42.500
39	24	55	50.000
40	22	95	105.000

Langkah 2: Mengambil satu persatu barang yang terdapat pada Tabel 5 yang dapat diangkat tanpa melebihi kapasitas *knapsack*. Diketahui bahwa kapasitas maksimum pengangkut sebesar **500 kg** dengan menggunakan strategi *Greedy by Weight* maka diperoleh tabel pengangkutan barang ke dalam permasalahan *knapsack* sebagai berikut:

Tabel 6. Pengangkutan Barang ke dalam Permasalahan *Knapsack* Menggunakan Strategi *Greedy by Weight*

No	Barangke- i	w_i (Kg)	p_i (Rp)	Status	x_i
1	17	2	2.000	Diambil	1
2	39	2	20.400	Diambil	1
3	31	3	9.000	Diambil	1
⋮	⋮	⋮	⋮	⋮	⋮
38	23	55	42.500	Diambil	1
39	24	55	50.000	Tidak	0
40	22	95	105.000	Tidak	0

Berdasarkan Tabel 6 diperoleh

$x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1, x_5 = 1, x_6 = 1, x_7 = 1, x_8 = 1, x_9 = 1, x_{10} = 1, x_{11} = 1,$
 $x_{12} = 1, x_{13} = 1, x_{14} = 1, x_{15} = 1, x_{16} = 1, x_{17} = 1, x_{18} = 1, x_{19} = 1, x_{20} = 1, x_{21} = 1,$
 $x_{22} = 0, x_{23} = 1, x_{24} = 0, x_{25} = 1, x_{26} = 1, x_{27} = 1, x_{28} = 1, x_{29} = 1, x_{30} = 1, x_{31} = 1,$
 $x_{32} = 1, x_{33} = 1, x_{34} = 1, x_{35} = 1, x_{36} = 1, x_{37} = 1, x_{38} = 1, x_{39} = 1, x_{40} = 1$

Solusi optimal permasalahan ini adalah

$X = \{1, 1,$
 $1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$

yang artinya barang ke-1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39 dan 40 diambil dengan total keuntungan maksimum sebesar Rp 956.400,- .

b. Menggunakan strategi *Greedy by Density* dalam menyelesaikan permasalahan *knapsack*

Langkah 1: Mencari nilai rasio p_i/w_i dari setiap barang, dengan mencari nilai rasio perberat atau biasa disebut *density* di setiap barang. Sehingga didapatkan tabel sebagai berikut:

Tabel 7. Nilai *Density* dari Tiap Barang

No.	Barang ke- i	w_i (Kg)	p_i (Rp)	p_i/w_i
1	1	50	80.000	1.600,000
2	2	25	17.500	700,000
3	3	50	115.000	2.300,000
⋮	⋮	⋮	⋮	⋮
38	38	9	48.000	5.333,333
39	39	2	20.400	10.200,000
40	40	5	30.000	6.000,000

Langkah 2: Barang-barang yang terdapat pada Tabel 7 diurutkan berdasarkan nilai *Density*nya terbesar. Selanjutnya dengan mengurutkan barang-barang pada Tabel 7 sehingga diperoleh tabel sebagai berikut:

Tabel 8. Urutan Barang-barang dari *Density* Terbesar

No.	Barang ke- i	w_i (Kg)	p_i (Rp)	p_i/w_i
1	39	2	20.400	10.200,000
2	40	5	30.000	6.000,000
3	38	9	48.000	5.333,333
⋮	⋮	⋮	⋮	⋮
38	2	25	17.500	700,000
39	30	17	6.000	352,941
40	29	10	3.000	300,000

Langkah 3: Mengambil satu persatu barang yang terdapat pada Tabel 8 yang dapat diangkat tanpa melebihi kapasitas *knapsack*. Diketahui bahwa kapasitas maksimum pengangkut sebesar **500 kg** dengan menggunakan strategi *Greedy by Density* maka diperoleh tabel pengangkutan barang ke dalam permasalahan *knapsack* sebagai berikut:

Tabel 9. Pengangkutan Barang ke dalam Permasalahan *Knapsack* Menggunakan Strategi *Greedy by Density*

No.	Barang ke- i	w_i (Kg)	p_i (Rp)	p_i/w_i	Status	x_i
1	39	2	20.400	10.200,000	Diambil	1
2	40	5	30.000	6.000,000	Diambil	1
3	38	9	48.000	5.333,333	Diambil	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
38	2	25	17.500	700,000	Diambil	1
39	30	17	6.000	352,941	Tidak	0
40	29	10	3.000	300,000	Tidak	0

Berdasarkan Tabel 9 diperoleh

$$x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1, x_5 = 1, x_6 = 1, x_7 = 1, x_8 = 1, x_9 = 1, x_{10} = 1, x_{11} = 1, x_{12} = 1, x_{13} = 1, x_{14} = 1, x_{15} = 1, x_{16} = 1, x_{17} = 1, x_{18} = 1, x_{19} = 1, x_{20} = 1, x_{21} = 1, x_{22} = 1, x_{23} = 0, x_{24} = 0, x_{25} = 1, x_{26} = 1, x_{27} = 1, x_{28} = 1, x_{29} = 0, x_{30} = 0, x_{31} = 1, x_{32} = 1, x_{33} = 1, x_{34} = 1, x_{35} = 1, x_{36} = 1, x_{37} = 1, x_{38} = 1, x_{39} = 1, x_{40} = 1$$

Solusi optimal permasalahan ini adalah

$X = \{1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$

yang artinya barang ke-1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 25, 26, 27, 28, 31, 32, 33, 34, 35, 36, 37, 38, 39 dan 40 diambil dengan total keuntungan maksimum sebesar Rp 1.009.900,- .

Berdasarkan dari ketiga strategi *Greedy* didapatkan hasil bahwa barang yang akan diangkut oleh Toko Surya Muda Pekanbaru adalah gula, tepung terigu, garam, dolpin, beras topikoki, beras anak daro, interco, plastik royal 15, plastik royal 19, plastik royal 24, kantong HD PE, masako, tepung beras, kopi bubuk sepeda, sambal abc, minyak rose brand, ajinomoto, minyak curah, boom, kantong plastik P.E, gas, aqua besar, aqua sedang, ale-ale, torpedo, o'caffe, kratindeng, kecap bango, Fanta, mizone, sprite, pocari, lasegar, teh pucuk, M-150 dan plastik kaca. Sedangkan barang yang tidak diangkut adalah muraqua, sms, x-teh dan teh gelas. Sehingga keuntungan maksimum yang diperoleh oleh Toko Surya Muda Pekanbaru sebesar Rp 1.009.900,- .

Kesimpulan

Berdasarkan hasil pembahasan, pengangkutan barang untuk permasalahan *knapsack* pada Toko Surya Muda Pekanbaru dengan menggunakan algoritma *greedy* mendapatkan keuntungan maksimum. Hasil menggunakan algoritma *greedy* dengan tiga strategi yaitu *Greedy by Profit* diperoleh keuntungan maksimum sebesar Rp 960.900, *Greedy by Weight* diperoleh keuntungan maksimum sebesar Rp 956.400, *Greedy by Density* diperoleh keuntungan maksimum sebesar Rp 1.009.900. Berdasarkan ketiga strategi tersebut, penyelesaian *Greedy by Density* menghasilkan keuntungan yang lebih besar bagi Toko Surya Muda Pekanbaru.

Daftar Pustaka

- [1] Ambarwari, A., dan Yanto, N. W., "Penerapan Algoritma Greedy pada Permasalahan Knapsack untuk Optimasi Pengangkutan Peti Kemas", Research Gate, 2016.
- [2] Brassard, G., "Fundamentals of Algorithms", New Jersey: Prentice Hall, 1996.
- [3] Hasan. M., "Implementasi Algoritma Greedy dalam Menyelesaikan Kasus Knapsack Problem", Makassar: Jurusan Matematika Sains dan Teknologi UIN Alaudinn Makassar, 2016.
- [4] Paryati, "Optimasi Strategi Algoritma Greedy untuk Menyelesaikan Permasalahan Knapsack 0-1", In Seminar Nasional Informatika (SEMNASIF), 2015.
- [5] Pratiwi, A., dkk. "Implementasi Algoritma Branch and Bound pada 0-1 Knapsack Problem untuk Mengoptimalkan Muatan Barang". Jurnal Matematika Unnes, 2014.
- [6] Setemen, K., "Implementasi Algoritma Genetik pada Knapsack Problem untuk Optimasi Pemilihan Buah Kemasan Kotak", Seminar Nasional Aplikasi Teknologi Informasi, Yogyakarta, 2010.
- [7] Zulhidayati, I., "Aplikasi Algoritma Greedy dan Program Dinamis (Dynamic Programming) pada Permainan Greddy Spiders", Bandung: Universitas Pendidikan Indonesia, 2013.