ID: 39

Penyelesaian Masalah *Knapsack* (0-1) Menggunakan Algoritma Greedy

Knapsack Problem Solving (0-1) Using Greedy Algorithms

Epi Hapidah[1*], Fahrudin Muhtarulloh[2]
[1,2]UIN Sunan Gunung Djati Bandung
Jl. A.H. Nasution No. 105 Bandung 40614 Jawa Barat Indonesia
Epihapidah1@gmail.com[1*], fahrudin.math@uinsgd.ac.id[2]

Abstrak - *Seiring dengan berkembangnya zaman serta kemajuan teknologi dan informasi seperti halnya pada bidang jasa proses pengiriman barang di Indonesia sedang meningkat dikarenakan sangat membantu serta hemat waktu dan tenaga. Jasa pengiriman barang ingin mendapatkan keuntungan yang lebih besar dan juga konsumen tidak merasa terbebani oleh biaya yang besar, maka dari itu harus saling menguntungkan satu sama lain. Paket pengiriman yang paling populer dan paling digemari oleh konsumen ialah paket reguler yang disebabkan oleh faktor biaya yang murah, layanannya sangat cepat dengan memakan waktu 3-4 hari. paket reguler ini sangat cocok digunakan untuk kasus masalah knapsack. Knapsack dapat diartikan sebagai karung atau kantung yang digunakan untuk menyimpan sesuatu. Tentu tidak semua objek dapat ditampung di dalam karung. Karung tersebut hanya dapat menyimpan beberapa objek dengan total ukurannya (weight) lebih kecil atau sama dengan ukuran kapasitas karung. Rumusan masalah dalam penelitian ini adalah: 1) Bagaimana penyelesaian masalah integer knapsack menggunakan algoritma greedy; dan 2) Bagaimana algoritma greedy memberikan solusi optimal untuk permasalahan integer knapsack. Adapun tujuan dari penelitian ini adalah: 1) Mengetahui penyelesaian pada permasalahan integer knapsack problem menggunakan algoritma greedy; dan 2) Mengetahui hasil solusi optimal pada algoritma greedy dalam penyelesaian integer knapsack problem. Dari hasil akhir perhitungan dapat disimpulkan bahwa total berat dan total keuntungan menggunakan algoritma greedy solusi yang paling optimal ialah dengan total bobot sebesar 41 dan total keuntungan 90.*

Kata Kunci*: Masalah Transportasi; Solusi Optimum; Knapsack (0-1); Algoritma Greedy,*

Abstract – *Along with the development of the times and advances in technology and information as well as in the field of goods delivery services in Indonesia is increasing because it is very helpful and saves time and energy. Goods delivery services want to get a bigger profit and also consumers do not feel burdened by large costs, therefore should benefit each other. The most popular and most popular delivery package by consumers is the regular package due to the low-cost factor, the service is very fast with 3-4 days. This regular package is perfect for knapsack problem cases. A knapsack can be interpreted as a sack or pouch used to store something. Of course, not all objects can be accommodated in a sack. The sack can only store a few objects with a total size (weight) smaller or equal to the size of the sack capacity. The formulation of the problems in this study is 1) How to solve the problem of knapsack integers using greedy algorithms, and 2) How greedy algorithms provide the optimal solution to knapsack integer problems. The objectives of this study are: 1) Knowing the solution to the problem of integer knapsack problem using a greedy algorithm, and 2) Know the optimal solution results on greedy algorithms in solving the integer knapsack problem. From the final result of the calculation can be concluded that the total weight and total profit using greedy algorithm the most optimal solution is with a total weight of 41 and a total profit of 90.*

Keywords*: Transportation Problems; Optimum Solutions; Knapsack (0-1); Greedy Algorithm*

1. Pendahuluan

Seiring dengan berkembangnya zaman serta kemajuan teknologi dan informasi seperti halnya pada bidang jasa proses pengiriman barang di Indonesia sedang meningkat dikarenakan sangat membantu manusia dalam pengiriman barang serta memudahkan untuk beraktivitas serta hemat waktu dan tenaga. Biaya dalam proses pengiriman tentu harus dikeluarkan, biaya tersebut berbeda-beda tergantung dengan berat barang dan jarak antara tempat pengiriman ke alamat tujuan. Agar biaya tersebut yang dikeluarkan sedikit serta keuntungan yang didapatkan maksimal, maka barang yang didistribusikan dipilih secara cermat.

Jasa pengiriman barang ingin mendapatkan keuntungan yang lebih besar dan juga konsumen tidak merasa terbebani oleh biaya yang besar, maka dari itu harus saling menguntungkan satu sama lain. Keuntungan sendiri ialah barang lebih cepat sampai ke tujuan. Kemanapun lokasi tujuan yang ingin dituju, pihak jasa pengiriman barang memiliki tanggung jawab untuk mengirimkan barang tersebut ke lokasi tujuan pengiriman dengan selamat dan sesuai dengan waktu yang telah ditentukan. Apabila mengalami kesulitan akses menuju lokasi tujuan pengiriman maka akan mempengaruhi biaya pengiriman yang harus dibayar. Terdapat banyak jenis paket pengiriman yang dimiliki oleh setiap jasa pengiriman barang yang disesuaikan dengan kebutuhan konsumen. Paket pengiriman yang paling populer dan paling digemari oleh konsumen ialah paket reguler yang disebabkan oleh faktor biaya yang murah, layanannya sangat cepat dengan memakan waktu 3-4 hari. Hal ini dikarenakan keterbatasan jumlah kurir yang tidak sebanding dengan banyaknya barang yang akan dikirimkan. Barang tersebut dikirimkan secara berangsur-angsur berdasarkan nilai keuntungan yang lebih besar dahulu. Dengan demikian paket reguler ini sangat cocok digunakan untuk kasus masalah *knapsack*.

Knapsack dapat diartikan sebagai karung atau kantung yang digunakan untuk menyimpan sesuatu. Tentu tidak semua objek dapat ditampung di dalam karung. Karung tersebut hanya dapat menyimpan beberapa objek dengan total ukurannya (*weight*) lebih kecil atau sama dengan ukuran kapasitas karung [1]. Masalah *knapsack* telah banyak dipelajari dan juga menarik perhatian para teoritikus dan pakar ilmu pengetahuan. Minat teoritis muncul terutama dari struktur sederhana mereka yang di satu sisi memungkinkan eksploitasi sejumlah sifat kombinatorial dan di sisi lain masalah optimisasi yang lebih kompleks untuk diselesaikan melalui serangkaian bagian dari masalah *knapsack*. Dari sudut pandang yang praktis, masalah tersebut dapat menjadi banyak model di bidang industri: penganggaran modal, muatan kargo, pemotongan saham, dan lainnya. Masalah *zero-one knapsack* ini dianggap sebagai masalah pemrograman linear bilangan bulat yang paling sederhana dan muncul dalam banyak aplikasi yang terdapat dalam bidang industri dan manajemen keuangan [2].

Diberikan sebuah set dalam n benda $j = 1, \dots, n$, dengan masing-masing memiliki berat w_j dan unit profit P_j, masalah *knapsack* terdiri dalam memilih beberapa barang untuk memuat ransel dengan kapasitas total W dalam pesanan untuk memaksimalkan total keuntungan [3]. Masalah *knapsack* muncul jika n buah barang yang tidak semuanya dapat dimasukkan ke dalam suatu tempat misalnya tas atau ransel. Sejumlah barang yang tersedia, masing-masing memiliki berat dan nilai yang berbeda-beda. Masalahnya adalah memilih barang-barang yang dibawa dengan keterbatasan kapasitas tempatnya dan nilai yang dihasilkan sebesar mungkin [4]. Jenis-jenis masalah *knapsack* ada tiga yaitu *unbounded knapsack problem* (tidak ada batasan jumlah barang untuk setiap barang), *integer knapsack problem* (jumlah barang untuk setiap barang hanya boleh 0 atau 1), dan *fractional knapsack problem* (jumlah barang untuk setiap barang boleh pecahan) [5]. Pada penelitian ini digunakan *integer knapsack problem* (0-1) dan menghubungkan masalah *knapsack* ke masalah transportasi linear. Metode yang dihasilkan dipecahkan oleh algoritma transportasi yang diadaptasi dan memberikan solusi yang optimal. Masalah *knapsack* sendiri dapat dipelajari dalam bab program bilangan bulat dan terdapat di riset operasi.

Persoalan masalah *knapsack*, khususnya *integer knapsack problem* dapat diselesaikan menggunakan berbagai cara. Beberapa caranya adalah algoritma *greedy*, *dynamic programming*, *brute force*, dan *genetic*. Algoritma tersebut sama-sama dapat menyelesaikan permasalahan

ISBN: 978-602-60581-2-6

integer knapsack problem dan mengahasilkan solusi optimum. Algoritma *greedy* merupakan salah satu metode dari sekian banyak metode algoritma yang dapat digunakan untuk menyelesaikan permasalahan *integer knapsack*. Contoh metode algoritma lain yang dapat digunakan untuk menyelesaikan permasalahan *knapsack* yaitu dengan algoritma *dynamic programming*, algoritma *brute force*, dan algoritma *genetic* [6].

Dari latar belakang tersebut peneliti tertarik untuk membahas penyelesaian persoalan tersebut dengan algoritma *greedy* untuk menyelesaikan permasalahan *integer knapsack* dikarenakan paling cepat dan mudah dibanding dengan algoritma yang lain serta mencari solusi layak awal dengan menggunakan metode Aproksimasi Vogel. Hasil akhir dari penelitian ini diharapkan dapat mengetahui hasil optimumnya.

Rumusan masalah dalam penelitian ini adalah: 1) Bagaimana penyelesaian masalah *integer knapsack* menggunakan algoritma *greedy*; dan 2) Bagaimana algoritma *greedy* memberikan solusi optimal untuk permasalahan *integer knapsack*. Adapun tujuan dari penelitian ini adalah: 1) Mengetahui penyelesaian pada permasalahan *integer knapsack problem* menggunakan algoritma *greedy*; dan 2) Mengetahui hasil solusi optimal pada algoritma *greedy* dalam penyelesaian *integer knapsack problem*.

2. Metode Penelitian

Penelitian ini merupakan studi literatur didasarkan pada jurnal yang dituliskan oleh Boudjellaba H, Gningue Y, dan Shamakhai H (2017) yang berjudul "*Solving The (0-1) Knapsack Problem By An Adapted Transportation Algorithm*". Dalam jurnal ini membahas mengenai penyelesaian masalah *knapsack* (0-1) dengan menggunakan algoritma transportasi yang diadaptasi untuk mencari solusi optimal, metode aproksimasi Vogel untuk mencari solusi layak awal serta pendekatan dengan algoritma *greedy*. Sehingga dalam penelitian ini penulis ingin mengkaji lebih dalam mengenai penyelesaian masalah *knapsack* dengan metode tersebut dengan menambah beberapa referensi dari jurnal lain [3].

2.31. Masalah *Knapsack* (0-1) dan Transportasi

Masalah *knapsack* (0-1) terjadi jika n buah barang tidak bisa semuanya dapat dimasukkan ke dalam suatu karung atau ransel, dikarenakan barang tersebut memiliki berat dan nilai yang berbeda-beda, sehingga harus memilih barang-barang yang akan dimasukkan ke dalam karung dengan keterbatasan kapasitas tempat agar total berat tidak melebihi kapasitas tempatnya dan nilai yang dihasilkannya sebesar mungkin [4]. Masalah *knapsack* (0-1) dapat diformulasikan sebagai masalah transportasi linear, yakni pendekatan dengan metode Vogel ekuivalen dengan algoritma *greedy* untuk masalah *knapsack* dengan urutan $O(n\ln(n))$ dari kompleksitas [3].

Knapsack dihubungkan dengan vektor:

$$X_{1,j} = X_j \; ; j = 1, \dots, n. \tag{1}$$

Dalam pemesanan untuk memiliki formulasi transportasi, dilanjutkan ke perubahan variabel

$$Y_{ij} = w_j X_{ij} \tag{2}$$

Keterangan:

$Y_{i,j}$ = Variabel *Knapsack* baris *i* kolom *j*; $i = 1, \dots, n \;\; ; j = 1, \dots, n$

w_j = Bobot/ berat barang ; $\quad j = 1, \dots, n$

$X_{i,j}$ = Vektor *Knapsack* baris *i* kolom *j*; $\; i = 1, \dots, n \;\; ; j = 1, \dots, n$

n = Jumlah barang

Koefisien tersebut dalam tujuan memaksimalkan adalah sama dengan nol, $K_{2,j} = 0 \; ; j = 1, \dots, n$ yang membuat variabel-variabel tersebut kurang menarik. Karena masalah *knapsack* biasanya tidak seimbang, maka ditambahkan surplus *dummy* barang $n + 1$ dengan koefisien nol dan beratnya sama dengan buffer *B* yang tidak diketahui yang merupakan kapasitas *knapsack* yang tidak dialokasikan yaitu

$$w_{(n+1)} = B. \tag{3}$$

Total kapasitas dari *knapsack* dan *dummy knapsack 2* yaitu sama dengan:

$$W_1 = W, \text{ dan} \tag{4}$$

ISBN: 978-602-60581-2-6

$$W_2 = B + \sum_{j=1}^{n} w_j - W. \quad (5)$$

Keterangan:
$w_{(n+1)}$ = Kapasitas *kna psack* yang tidak dialokasikan
B = *Buffer*
W_1 = Kapasitas *knapsack* 1
W = Total kapasitas *knapsack*
w_j = Bobot/ berat barang ;$j = 1, \dots, n$
n = Jumlah barang

Semua transformasi ini menyiratkan masalah *knapsack* yang seimbang

$$Masalah\ Knapsack \begin{cases} \text{maks } Z = \sum_{j=1}^{n} \frac{P_j}{w_j} Y_{1,j} \\ \sum_{j=1}^{n+1} Y_{1,j} = W_1 \ dan \ \sum_{j=1}^{n+1} Y_{2,j} = W_2 \\ \sum_{i=1}^{2} Y_{1,j} = w_j \ ; \ j = 1, \dots, (n+1) \\ Y_{1,j} \in \{0, w_j\} \ ; \ i = 1,2 \ ; \ j = 1, \dots, (n+1) \end{cases} \quad (6)$$

Keterangan:
$maks\ Z$ = Nilai optimum dari fungsi tujuan
n = Jumlah barang
P_j = *Profit* barang ; $j = 1, \dots, n$
w_j = Bobot/ berat barang ; $j = 1, \dots, n$
$Y_{1,j}$ = Variabel *knapsack* pada baris 1 kolom j ; $j = 1, \dots, n$
$Y_{2,j}$ = Variabel *knapsack* pada baris 2 kolom j ; $j = 1, \dots, n$
W_1 = Kapasitas *knapsack 1*
W_2 = Kapasitas *knapsack 2*

Barang diurutkan berdasarkan penurunan tingkat efisiensi dengan yang terbesar $K_{1,1} = \underset{j}{maks} \left\{ \frac{P_j}{w_j} \right\}$, kemudian mengurangi semua koefisien dari nilai terbesar untuk mendapatkan koefisien baru

$$K_{1,j} = K_{1,1} - \frac{P_j}{w_j} \ dan\ K_{2,j} = K_{1,1} \ ; \ j = 1, \dots, n \quad (7)$$

Keterangan:
$K_{1,j}$ = Koefisien tingkat efisiensi pada baris 1 kolom j ; $j = 1, \dots, n$
$K_{2,j}$ = Koefisien tingkat efisiensi pada baris 2 kolom j ; $j = 1, \dots, n$
$K_{1,1}$ = Koefisien tingkat efisiensi pada baris 1 kolom1
P_j = *Profit* barang ; $j = 1, \dots, n$
w_j = Bobot/ berat barang ; $j = 1, \dots, n$
n = Jumlah barang

Hal ini mengubah masalah *knapsack* menjadi minimisasi masalah *knapsack* transportasi

$$Masalah\ Knapsack\ Transportasi \begin{cases} minZ_L = \sum_{i=1}^{2} \sum_{j=1}^{n+1} K_{1,j} Y_{1,j} \\ \sum_{j=1}^{n+1} Y_{1,j} = W_1 \ dan \ \sum_{j=1}^{n+1} Y_{2,j} = W_2 \\ \sum_{i=1}^{2} Y_{1,j} = w_j \ ; \ j = 1, .., (n+1) \\ Y_{1,j} \in \{0, w_j\} \ ; \ i = 1,2 \ ; j = 1, \dots, (n+1) \end{cases} \quad (8)$$

Keterangan:
min Z_L = Nilai minium dari fungsi tujuan
n = Jumalah barang
$K_{1,j}$ = Koefisien tingkat efisiensi pada baris 1 kolom j ; $j = 1, \dots, n$
$Y_{1,j}$ = Variabel *knapsack* pada baris 1 kolom j ; $j = 1, \dots, n$
$Y_{2,j}$ = Variabel *knapsack* pada baris 2 kolom j ; $j = 1, \dots, n$
W_1 = Kapasitas *knapsack 1*
W_2 Kapasitas *knapsack 2*

ISBN: 978-602-60581-2-6

w_j = Bobot/ berat barang ; $j = 1, \dots, n$

Hal ini dapat diselesaikan oleh algoritma transportasi yang diadaptasi yang disajikan di bagian berikutnya.

2.32. Algoritma Greedy

Algoritma *greedy* merupakan metode yang populer untuk memecahkan masalah optimasi. Masalah optimasi sendiri terdiri dari dua macam yaitu maksimasi dan minimasi. Algoritma *greedy* membentuk solusi langkah demi langkah. Pada setiap langkah, terdapat banyak pilihan yang perlu dieksplorasi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Pada setiap langkahnya merupakan pilihan untuk membuat langkah optimum lokal dengan harapan bahwa langkah sisanya mengarah ke solusi optimum global. Algoritma *greedy* merupakan metode untuk menyelesaikan masalah *integer knapsack*.

Menurut Pan dan Zhang, menyelesaikan masalah *integer knapsack* dengan algoritma *greedy* mempunyai kompleksitas waktu $O(n \log n)$. Metode algoritma ini tidak selalu menyelesaikan dengan hasil yang optimal, tetapi dapat menghasilkan solusi optimal lokal yang mendekati solusi optimal global dengan waktu yang cepat. Untuk memilih barang yang akan dimasukkan ke dalam *knapsack* terdapat beberapa strategi dari metode algoritma *greedy* menurut Juwita, Susanto dan Haloman adalah:

a) *Greedy by Profit*

Setiap langkah di masalah *knapsack* diisi dengan barang yang mempunyai keuntungan terbesar. Strategi ini bertujuan untuk memaksimalkan keuntungan dengan memilih barang yang paling menguntungkan terlebih dahulu. Tahap pertamanya ialah barang-barang berdasarkan *value/profit* barang diurutkan secara menurun. Kemudian barang-barang diambil satu persatu sampai kapasitas tempatnya penuh atau sudah tidak ada yang dapat dimasukkan lagi.

b) *Greedy by Weight*

Setiap langkah di masalah *knapsack* diisi dengan barang yang mempunyai berat yang paling ringan. Strategi ini bertujuan untuk memaksimalkan keuntungan denganmemasukkan barang sebanyak mungkin. Tahap pertamanya ialah barang-barang berdasarkan berat barang yang paling ringan diurutkan secara menurun. Kemudian barang-barang diambil satu persatu sampai kapasitas tempatnya penuh atau sudah tidak ada yang dapat dimasukkan lagi.

c) *Greedy by Density*

Setiap langkah di masalah *knapsack* diisi dengan barang yang mempunyai $\frac{P_j}{w_j}$ dimana P_j adalah *value/profit*, w_j adalah *weight* (berat) dan $i = (1,2,3, \dots, n)$. Strategi ini bertujuan untuk memaksimalkan keuntungan dengan memilih barang yang mempunyai $\frac{P_j}{w_j}$ (*density*) terbesar. Takap pertamanya ialah mencari dan mengurutkan secara menurun barang-barang berdasarkan $\frac{P_j}{w_j}$ barang. Kemudian barang-barang diambil satu persatu sampai kapasitas tempatnya penuh atau sudah tidak ada yang dapat dimasukkan lagi.

3. Hasil dan Pembahasan

Berdasarkan contoh pada masalah *knapsack* [3] dengan menggunakan algoritma *greedy* untuk menentukan solusi optimumnya. Parameter w_j (berat), P_j (*value/profit*), W (kapasitas maksimum). Data yang sudah dicontohkan oleh Boudjellaba, Gningue dan Shamakhai [3] awalnya diketahui:

$$w_1 = 31, w_2 = 10, w_3 = 20, w_4 = 18, w_5 = 4, w_6 = 3, w_7 = 6$$
$$P_1 = 70, P_2 = 20, P_3 = 39, P_4 = 35, P_5 = 7, P_6 = 5, P_7 = 9$$
$$W = 50$$

a) Greedy by Profit

Langkah pertama yang dilakukan ialah barang-barang diurutkan secara menurun berdasarkan *value/profit*. Kemudian diambil satu persatu barang sampai kapasitas maksimum tempatnya terpenuhi atau sudah tidak ada barang yang bisa dimasukkan lagi.

ISBN: 978-602-60581-2-6

Tabel 1. *Greedy by Profit*

J	w_j	P_j	$\frac{P_j}{w_j}$	**Status**
1	31	70	$\frac{70}{31} = 2{,}258$	Diambil
3	20	39	$\frac{39}{20} = 1{,}95$	Tidak
4	18	35	$\frac{35}{18} = 1{,}944$	Tidak
2	10	20	$\frac{20}{10} = 2$	Tidak
7	6	9	$\frac{9}{6} = 1{,}5$	Tidak
5	4	7	$\frac{7}{4} = 1{,}75$	Tidak
6	3	5	$\frac{5}{3} = 1{,}667$	Tidak

b) Greedy by Weight

Langkah pertama yang dilakukan ialah barang-barang diurutkan secara menaik berdasarkan *weight*/ beratnya. Kemudian diambil satu persatu barang sampai kapasitas maksimum tempatnya terpenuhi atau sudah tidak ada barang yang bisa dimasukkan lagi.

Tabel 2 *Greedy by Weight*

J	w_j	P_j	$\frac{P_j}{w_j}$	**Status**
6	3	5	$\frac{5}{3} = 1{,}667$	Diambil
5	4	7	$\frac{7}{4} = 1{,}75$	Diambil
7	6	9	$\frac{9}{6} = 1{,}5$	Diambil
2	10	20	$\frac{20}{10} = 2$	Diambil
4	18	35	$\frac{35}{18} = 1{,}944$	Diambil
3	20	39	$\frac{39}{20} = 1{,}95$	Tidak
1	31	70	$\frac{70}{31} = 2{,}258$	Tidak

c) Greedy by Density

Langkah pertama yang dilakukan ialah mencari $\frac{P_j}{w_j}$ terbesar, kemudian diurutkan secara menurun berdasarkan $\frac{P_j}{w_j}$ (*density*). Kemudian diambil satu persatu barang sampai kapasitas maksimum tempatnya terpenuhi atau sudah tidak ada barang yang bisa dimasukkan lagi.

Tabel 3. *Greedy by Density*

J	w_j	P_j	$\frac{P_j}{w_j}$	*Status*
1	31	70	$\frac{70}{31} = 2{,}258$	Diambil

ISBN: 978-602-60581-2-6

J	w_j	P_j	$\frac{P_j}{w_j}$	*Status*
2	10	20	$\frac{20}{10} = 2$	Diambil
3	20	39	$\frac{39}{20} = 1{,}95$	Tidak
4	18	35	$\frac{35}{18} = 1{,}944$	Tidak
5	4	7	$\frac{7}{4} = 1{,}75$	Tidak
6	3	5	$\frac{5}{3} = 1{,}667$	Tidak
7	6	9	$\frac{9}{6} = 1{,}5$	Tidak

Maka hasil akhir dari permasalahan di atas dan dianalisis menggunakan tiga metode dari algoritma *greedy* mendapatkan hasil data seperti di bawah ini:

Tabel 4. Hasil akhir algoritma *greedy*

Barang				***Greedy***		
J	w_j	P_j	$\frac{P_j}{w_j}$	***Profit***	***Weight***	***Density***
1	31	70	$\frac{70}{31} = 2{,}258$	1	0	1
2	10	20	$\frac{20}{10} = 2$	0	1	1
3	20	39	$\frac{39}{20} = 1{,}95$	0	0	0
4	18	35	$\frac{35}{18} = 1{,}944$	0	1	0
5	4	7	$\frac{7}{4} = 1{,}75$	0	1	0
6	3	5	$\frac{5}{3} = 1{,}667$	0	1	0
7	6	9	$\frac{9}{6} = 1{,}5$	0	1	0
Total Bobot				**31**	**41**	**41**
Total Keuntungan				**70**	**76**	**90**

Greedy dengan 1 berarti barang diambil dan 0 berarti barang tidak diambil. Hasil akhir tersebut dapat disimpulkan bahwa total berat dan total keuntungan menggunakan algoritma *greedy* yang paling optimum ialah dengan total bobot sebesar 41 dan total keuntungan 90.

4. Kesimpulan

Berdasarkan data, hasil dan pembahasan yang telah dilakukan sebelumnya, maka dapat diperoleh kesimpulan dari penyelesaian masalah *integer knapsack* dengan menggunakan algoritma *greedy* yang terdiri dari *greedy by profit*, *greedy by weight*, dan *greedy by density* dengan ditampilkan pada Tabel 2.4 sebagai berikut:

ISBN: 978-602-60581-2-6

Tabel 5. Hasil akhir algoritma *greedy*

Barang				Greedy		
J	w_j	P_j	$\frac{P_j}{w_j}$	*Profit*	*Weight*	*Density*
1	31	70	$\frac{70}{31} = 2{,}258$	1	0	1
2	10	20	$\frac{20}{10} = 2$	0	1	1
3	20	39	$\frac{39}{20} = 1{,}95$	0	0	0
4	18	35	$\frac{35}{18} = 1{,}944$	0	1	0
5	4	7	$\frac{7}{4} = 1{,}75$	0	1	0
6	3	5	$\frac{5}{3} = 1{,}667$	0	1	0
7	6	9	$\frac{9}{6} = 1{,}5$	0	1	0
Total Bobot				**31**	**41**	**41**
Total Keuntungan				**70**	**76**	**90**

Greedy dengan 1 berarti barang diambil dan 0 berarti barang tidak diambil. Hasil akhir tersebut dapat disimpulkan bahwa total berat dan total keuntungan menggunakan algoritma *greedy* solusi yang paling optimal ialah dengan total bobot sebesar 41 dan total keuntungan 90.

Ucapan Terima Kasih

Penulis mengucapkan banyak terima kasih kepada para pihak yang telah membantu menyelesaikan penelitian ini.

References

[1] E. Gautama, "Menyelesaikan Knapsack Problem dengan Menggunakan Algoritma Greedy," PERBANAS INSTITUTE, 31 Mei 2017. [Online]. Available: https://dosen.perbanas.id/menyelesaikan-knapsack-problem-dengan-menggunakan-algoritma-greedy/. [Diakses 15 Oktober 2020].

[2] S. Martello and P. Toth, Knapsack Problems Algorithms and Computer Implementations, England: John Wiley & Sons Ltd. Baffins Lane, Chichester West Sussex PO19 IUD, 1990.

[3] B. H, G. Y and S. H, "Solving The (0-1) Knapsack Problem By An Adapted Transportation Algorithm," *International Journal of Latest Research in Science and Technology*, vol. 6, no. 3, pp. 20-24, 2017.

[4] J. J. Siang, Riset Operasi dalam Pendekatan Algoritmis. Edisi 2, Yogyakarta: ANDI, 2014.

[5] M. A. Rois, Penyelesaian Integer Knapsack Problem Menggunakan Eksplorasi Algoritma Greedy, Dynamic Programming, Brute Force dan Genetic, Semarang: UIN Walisongo, 2019.

[6] X. Pan and T. Zhang, "Comparison and Analysis of Algorithms for The 0/1 Knapsack Problem," *IOP Journal of Phisics*, pp. 1-8, 2018.

ISBN: 978-602-60581-2-6