

## PERBANDINGAN ALGORITMA GREEDY DAN METODE BRANCH AND BOUND PADA PENYELESAIAN KNAPSACK 0-1 UNTUK MENGOPTIMALKAN MUATAN BARANG

D. Wungguli<sup>1</sup>, S. S. Ibrahim<sup>2</sup>, dan L. Yahya<sup>3</sup>

<sup>1,2,3</sup>Program Studi Matematika FMIPA Universitas Negeri Gorontalo

Jln. Jendral Sudirman No.6, Kota Gorontalo 96128, Indonesia

<sup>1</sup>septiaibrahim@gmail.com, <sup>2</sup>lailany.yahya@ung.ac.id, <sup>3</sup>djihad@ung.ac.id

### ABSTRACT

This research was conducted to determine the cargo of goods to be sold by using cargo media that has a limited capacity in obtaining maximum profit. In data collection, it was required weight, profit, and maximum capacity of the truck. The calculations were done in the form of total weight and total profit of each item and then modeled into a mathematical form. The method used in this research is the greedy algorithm and the branch and bound method. Obtained several optimal solutions using the greedy algorithm, greedy by weight, greedy by profit, and greedy by density. The results show that by using the greedy algorithm, the maximum result is obtained in the greedy by profit calculation with a profit of Rp. 13952000 and a total weight of 15319Kg. While using the branch and bound method, the profit is Rp. 141689000 with a total weight of 15599Kg.

**Keywords** : Knapsack Problem 0-1, Greedy Algorithm, Branch and Bound Method

### ABSTRAK

Penelitian ini dilakukan untuk menentukan muatan barang yang akan dijual dengan menggunakan media muatan yang memiliki kapasitas terbatas dalam memperoleh keuntungan maksimum. Data yang digunakan yaitu berupa berat barang, keuntungan, dan kapasitas maksimum media muatan. Selanjutnya dilakukan perhitungan berupa berat total dan keuntungan total dari setiap barang kemudian dimodelkan kedalam bentuk matematika. Metode yang digunakan dalam penelitian ini yaitu algoritma *greedy* dan metode *branch and bound*. Diperoleh beberapa solusi optimal dalam menggunakan algoritma *greedy* yaitu *greedy by weight*, *greedy by profit*, dan *greedy by density*. Hasil penelitian menunjukkan bahwa dengan menggunakan algoritma *greedy* diperoleh hasil yang maksimal pada perhitungan *greedy by profit* dengan keuntungan yaitu Rp13952000 dan berat total 15319Kg. Sedangkan menggunakan metode *branch and bound* diperoleh keuntungan Rp141689000 dengan berat total 15599Kg.

**Kata kunci** : Knapsack Problem 0-1, Algoritma Greedy, Metode Branch and Bound.

## I. PENDAHULUAN

*Knapsack* dapat diartikan sebagai karung, kantong, atau buntilan yang digunakan untuk memuat sesuatu dengan tujuan untuk mendapatkan keuntungan yang maksimum dari pemilihan objek tanpa melebihi kapasitas daya tampung media penyimpanan tersebut (Supriana, 2016). *Knapsack Problem* adalah suatu permasalahan optimasi dengan tujuan untuk memaksimalkan total nilai dari barang-barang yang dimasukkan kedalam wadah tanpa melewati kapasitas wadah tersebut (Kosasi, 2013). Permasalahan *knapsack* terdiri dari beberapa kasus yaitu integer *knapsack*, bounded *knapsack*, dan unbounded *knapsack* yaitu permasalahan dalam menentukan objek yang dimuat dengan kapasitas tidak terbatas (Ambarwari and Yanto, 2016). Untuk *integer knapsack* permasalahannya bersyarat yaitu 0 atau 1 artinya barang tersebut dimasukkan kedalam *knapsack* atau tidak sama sekali (Abdurrahman Rois, 2019). Menurut (Arista, 2013) terdapat beberapa persoalan optimasi dalam kehidupan sehari-hari yaitu permasalahan menentukan barang, menentukan jalur lintasan terpendek dari suatu tempat ketempat yang lain, dan menentukan jumlah pekerja seminimal mungkin untuk melakukan suatu proses produksi agar pengeluaran biaya pekerja dapat diminimalkan dan hasil produksi tetap maksimal. Dalam menentukan nilai optimum dari suatu persoalan linear dapat menggunakan metode program linear (Prasetya, 2015).

Model matematika untuk masalah *knapsack* terdiri dari tiga unsur yaitu variable keputusan, fungsi tujuan dan fungsi pembatas (Dimiyati and Dimiyati, 2018). Dalam menemukan solusi *knapsack problem* dapat diselesaikan dengan menggunakan metode *branch and bound* dan metode algoritma *greedy* (Rachmawati and Candra, 2013). Algoritma *greedy* merupakan salah satu metode populer untuk memecahkan persoalan optimasi karena akan banyak solusi kemungkinan yang ada (Sabaruddin, 2016). Cara kerja metode algoritma *greedy* yaitu dilakukan secara bertahap dan memperhatikan masing-masing input data pada setiap keadaan (Ammar, 2019). Metode *branch and bound* adalah salah satu metode untuk menyelesaikan masalah *integer linear programming* (Supatimah, Farida and Andriani, 2019). Sebelum melakukan perhitungan dengan menggunakan metode *branch and bound* terlebih dahulu menyelesaikan program linear relaksasi (Setiawani, 2017). Prinsip kerja dari metode *branch and bound* adalah mencabangkan variabel keputusan yang tidak memiliki penyelesaian bulat, pencabangan dilakukan terus hingga diperoleh penyelesaian bulat sehingga semua variabel keputusannya bernilai bulat dan menghasilkan keuntungan maksimal pada perusahaan. Sedangkan dalam menyelesaikan *knapsack problem* dengan menggunakan algoritma *greedy* terdapat beberapa beberapa jenis *greedy* yang dapat digunakan yaitu *greedy by weight*, *greedy by profit*, dan *greedy By density*.

Beberapa penelitian sebelumnya pernah menggunakan metode *branch and bound* dan algoritma *greedy* diantaranya yaitu penelitian yang dilakukan oleh (Pratiwi, Mulyono and Rochmad, 2014) pada studi kasus di sebuah CV. Pangestu Interaksi Semarang dengan menggunakan metode *branch and bound* dan penelitian (Ammar, 2019) untuk studi kasus sebuah jasa pengiriman PT Citra Van Titipan Kilat dengan menggunakan metode algoritma *greedy*.

PK. Murni Sanjaya adalah salah satu perusahaan yang bergerak dalam bidang jual beli jenis dan bentuk kayu. PK Murni Sanjaya ini terleas di Kabupaten Gorontalo yang telah berdiri sejak tahun 2010. Sama halnya dengan perusahaan lain, perusahaan kayu Murni Sanjaya juga ingin melakukan hal yang terbaik dan tepat dalam memilih barang untuk menjamin kebutuhan bagi kelancaran usaha. Dalam menjamin kelangsungan hidup usahanya setiap perusahaan perlu menentukan masing-masing barang yang akan dijual, karenajika tidak diadakannya penentuan pemilihan barang perusahaan akan dihadapkan pada resiko bahwa suatu waktu perusahaan akan mengalami kerugian. Berdasarkan permasalahan tersebut maka dalam penelitian ini peneliti tertarik untuk menentukan solusi optimal dari permasalahan *knapsack* dengan menggunakan metode algoritma *greedy* dan metode *branch and bound*.

## II. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini yaitu menggunakan studi literature dengan mempelajari jurnal-jurnal dan buku serta beberapa sumber yang berkaitan dengan *knapsack problem* 0-1. Langkah-langkah yang dilakukan yaitu :

1. Mengumpulkan data
2. Mendefinisikan data yang diperoleh
3. Memodelkan kedalam bentuk matematika.
4. Menerapkan perhitungan menggunakan algoritma *greedy* dan metode *branch and bound*.
5. Membandingkan solusi optimal dari berbagai solusi yang didapatkan.
6. Menentukan Solusi optimal dari kedua metode yang digunakan.
7. Representasi hasil.

## III. HASIL DAN PEMBAHASAN

### 3.1. Pendefinisian Variabel

Dalam menyelesaikan permasalahan knapsack 0-1 dengan membandingkan metode greedy dan metode branch and bound maka diperlukan data berupa jenis kayu, berat kayu, harga jual dan harga beli di PK. Murni Sanjaya dengan kapasitas angkut maksimum dari 3 truk yang akan mengangkut kayu tersebut di PK. Murni Sanjaya sebesar 15600. Perlu dilakukan identifikasi dari data berupa berat total dan keuntungan total dari setiap barang. Untuk menentukan nilai dari berat total kayu dapat diidentifikasi dengan cara mengalikan berat dari masing-masing barang  $w_i$  dengan banyaknya barang. Sedangkan untuk menentukan nilai dari keuntungan  $v_i$  total dari masing-masing kayu yaitu diidentifikasi dengan cara menghitung selisih antara harga jual dengan harga beli kemudian dikalikan dengan banyaknya jumlah barang kemudian didefinisikan.

Tabel 1 : Pendefinisian Variabel

No	Jenis Kayu	Ukuran Kayu	Berat Total ( $w_i$ )	Variabel $x_i$	Keuntungan Total ( $v_i$ )	Rasio $\frac{v_i}{w_i}$
1	KC	3 × 30	775	$x_1$	1250000	1612.9
2	KC	3 × 20	660	$x_2$	1050000	1590
3	KC	3 × 25	750	$x_3$	1125000	1500
4	KCN	3 × 20	480	$x_4$	600000	1250
5	KA	2 × 25	560	$x_5$	630000	1125
6	KC	6 × 12	630	$x_6$	630000	1000
7	KM	3 × 5	500	$x_7$	500000	1000
8	KC	6 × 16	840	$x_8$	800000	952.38
9	KM	4 × 25	640	$x_9$	600000	937.5
10	KCN	2.5 × 25	770	$x_{10}$	700000	909
11	KM	4 × 5	575	$x_{11}$	517000	900
12	KM	6 × 16	700	$x_{12}$	625000	892.86
13	KCN	2 × 25	720	$x_{13}$	630000	875
14	KM	5 × 5	450	$x_{14}$	375000	833.33
15	KN	2 × 25	760	$x_{15}$	600000	789.47
16	KM	2 × 20	605	$x_{16}$	440000	727.27
17	KA	2.5 × 25	704	$x_{17}$	480000	681.81
18	KM	2 × 25	675	$x_{18}$	450000	666.67
19	KM	5 × 7	540	$x_{19}$	360000	666.67
20	KM	6 × 12	720	$x_{20}$	450000	625
21	KM	2.5 × 25	648	$x_{21}$	360000	555.55
22	KN	2.5 × 25	682	$x_{22}$	372000	545.45
23	KM	5 × 10	570	$x_{34}$	304000	533.33
24	KM	6 × 10	650	$x_{24}$	325000	500
25	KD	6 × 12	300	$x_{25}$	150000	500
26	KM	3 × 25	400	$x_{26}$	199900	499
27	KD	3 × 25	820	$x_{27}$	408000	498
28	KM	6 × 8	125	$x_{28}$	62000	495
29	KD	6 × 16	455	$x_{29}$	224000	492
30	KM	3 × 20	120	$x_{30}$	59000	491

Pendefinisian setiap barang dapat dilihat pada Tabel 1, sedangkan variabel pada *knapsack problem* didefinisikan sebagai :

$$x_i = \begin{cases} 1, & \text{jika barang ke } i \text{ dimasukkan kedalam truk} \\ 0, & \text{jika barang ke } i \text{ tidak dimasukkan kedalam truk} \end{cases} \quad \forall i = 1, 2, 3, \dots, 30$$

Permasalahan *knapsack* dapat dimodelkan kedalam bentuk matematika sehingga memperoleh fungsi tujuan dan fungsi pembatas untuk memaksimalkan keuntungan.

Fungsi Tujuan :

Maksimumkan :

$$Z = 1250x_1 + 1050x_2 + 1125x_3 + 600x_4 + 630x_5 + 630x_6 + 500x_7 + 800x_8 + 600x_9 + 700x_{10} + 517x_{11} + 625x_{12} + 630x_{13} + 375x_{14} + 600x_{15} + 440x_{16} + 480x_{17} + 450x_{18} + 360x_{19} + 450x_{20} + 360x_{21} + 372x_{22} + 304x_{23} + 325x_{24} + 150x_{25} + 199.9x_{26} + 408x_{27} + 62x_{28} + 224x_{29} + 59x_{30} \quad (1)$$

Fungsi Pembatas :

$$775x_1 + 660x_2 + 750x_3 + 480x_4 + 560x_5 + 560x_6 + 500x_7 + 840x_8 + 640x_9 + 770x_{10} + 575x_{11} + 700x_{12} + 720x_{13} + 450x_{14} + 760x_{15} + 605x_{16} + 704x_{17} + 675x_{18} + 540x_{19} + 720x_{20} + 648x_{21} + 682x_{22} + 570x_{23} + 650x_{24} + 300x_{25} + 400x_{26} + 820x_{27} + 125x_{28} + 455x_{29} + 120x_{30} \leq 15600 \quad (2)$$

$$x_i \geq 0 \text{ untuk } i = 1, 2, 3, \dots, 30.$$

### 3.2. Perhitungan Algoritma Greedy

Perhitungan algoritma greedy terdapat 3 strategi greedy yang akan digunakan untuk menentukan barang yang akan dipilih untuk dimasukkan kedalam knapsack yaitu greedy by weight, greedy by profit, dan greedy by density (Supriadi, 2016).

#### 3.1.1. Greedy by weight

Greedy by weight adalah mengurutkan barang pada Tabel 1 berdasarkan berat teringan kemudian dimasukkan satu persatu barang kedalam truk dengan kapasitas truk sebesar 15600Kg sampai kapasitas truk penuh atau sudah tidak ada lagi yang bisa dimasukkan. Menurut perhitungan dengan menggunakan strategi greedy by weight barang yang akan dimasukkan kedalam truk yang bernilai 1 yaitu  $\{x_2, x_3, x_4, x_5, x_6, x_7, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}, x_{20}, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}, x_{28}, x_{29}, x_{30}\}$  sedangkan untuk  $\{x_1, x_8, x_{27}\}$  bernilai 0 karena apabila ketiga barang tersebut dimasukkan maka akan melebihi kapasitas truk. Menurut perhitungan greedy by weight diperoleh solusi optimal yaitu  $X = \{0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1\}$ . Berdasarkan fungsi tujuan pada Persamaan (1) maka keuntungan yang diperoleh dari perhitungan greedy by weight dengan total berat barang yang diangkut sebesar 15389Kg yaitu Rp.12817900.

### 3.1.2. Greedy by profit

Greedy by profit adalah mengurutkan barang-barang pada Tabel 1 berdasarkan keuntungan terbesar dengan tetap memperhatikan berat barang kemudian dimasukkan satu persatu barang kedalam truk sampai kapasitas truk penuh atau sudah tidak ada lagi yang bisa dimasukkan. pada perhitungan dengan menggunakan strategi greedy by profit barang yang akan dimasukkan kedalam truk yang bernilai 1 yaitu  $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}, x_{20}, x_{21}, x_{22}, x_{27}\}$  sedangkan untuk  $\{x_{23}, x_{24}, x_{25}, x_{26}, x_{28}, x_{29}, x_{30}\}$  bernilai 0 karena apabila ketiga barang tersebut dimasukkan maka akan melebihi kapasitas truk. Menurut perhitungan greedy by profit diperoleh solusi optimal yaitu  $X = \{1, 0, 0, 0, 0, 0, 0, 0, 0\}$ . Berdasarkan fungsi tujuan pada Persamaan (1) maka keuntungan yang diperoleh dari perhitungan greedy by profit dengan total berat barang yang diangkut sebesar 15319Kg yaitu Rp.13952000.

### 3.1.3. Greedy by density

Greedy by density adalah strategi mengurutkan barang-barang berdasarkan rasio terbesar dengan tetap memperhatikan berat barang kemudian dimasukkan satu persatu barang kedalam truk sampai kapasitas truk penuh atau sudah tidak ada lagi yang bisa dimasukkan. Pada perhitungan dengan menggunakan greedy by density barang yang akan dimasukkan kedalam truk yang bernilai 1 yaitu  $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}, x_{20}, x_{21}, x_{22}, x_{23}\}$  sedangkan untuk  $\{x_{24}, x_{25}, x_{26}, x_{27}, x_{28}, x_{29}, x_{30}\}$  bernilai 0 karena apabila ketiga barang tersebut dimasukkan maka akan melebihi kapasitas truk. Menurut perhitungan greedy by density diperoleh solusi optimal yaitu  $X = \{1, 0, 0, 0, 0, 0, 0, 0, 0\}$ . Berdasarkan fungsi tujuan pada Persamaan (1) maka keuntungan yang diperoleh dari perhitungan greedy by density dengan total berat barang yang diangkut sebesar 149549Kg yaitu Rp.13848000.

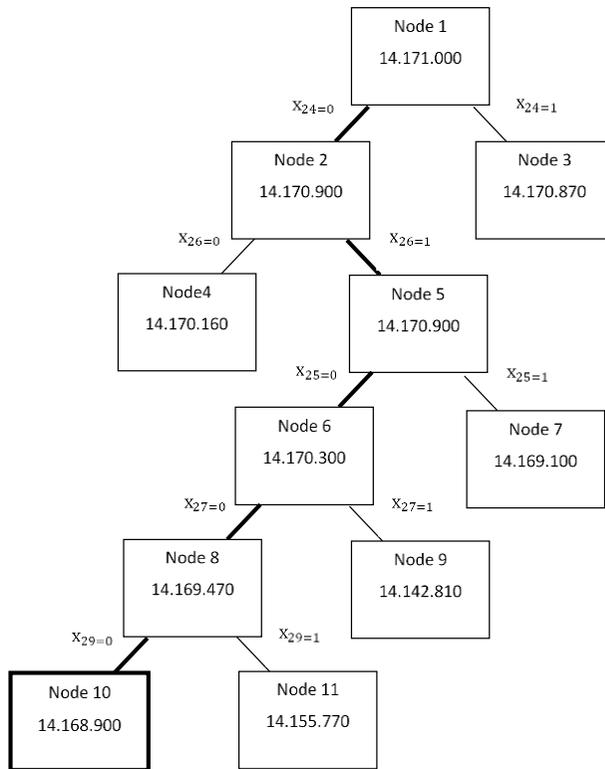
## 3.3. Perhitungan Metode *Branch and Bound*

Sebelum menerapkan metode *branch and bound* terlebih dahulu menentukan batas *dantzig*. Batas *dantzig* adalah batas akhir barang yang jika semua barangnya dimasukkan maka akan melebihi kapasitas *knapsack* dan apabila tidak dimasukkan maka *knapsack* akan bersisa (Supatimah dkk, 2019). Langkah awal dalam menentukan batas *dantzig* yaitu memasukan barang kedalam *knapsack* dengan memilih barang dengan memperhatikan rasio dari setiap barang, dan apabila yang memiliki nilai rasio yang sama maka yang dipilih terlebih dahulu yang memiliki keuntungan yang lebih besar . maka didapatkan :

$$s = \min (j: w_1 + w_2 + w_3 + \dots + w_{24} = 15604 > 15600) .$$

Berdasarkan hasil yang diperoleh sebelumnya dapat dilihat bahwa batas *dantzig* terletak pada barang  $x_{24}$  untuk semua barang yang dimasukkan sebelum batas *dantzig* bernilai 1 yaitu :





Gambar 1 : Pencabangan *Node*

### 3.4. Rekapitulasi Solusi Optimal

Setelah dilakukan perhitungan dengan menggunakan algoritma *greedy* dan metode *branch and bound* di PK. Murni Sanjaya maka diperoleh solusi optimal. Untuk *greedy by weight* diperoleh hasil optimal sebanyak 27 kayu dimuat dengan total berat 15389Kg yaitu Rp.12817900, pada *greedy by profit* solusi yang dihasilkan yaitu sebanyak 23 kayu dimuat diperoleh keuntungan Rp.13952000 dengan total berat 15319 Kg, dan *greedy by density* diperoleh solusi sebanyak 23 kayu dimuat dengan keuntungan yaitu Rp.13952000 dan berat total 14954 Kg. Sedangkan untuk solusi optimal dengan menggunakan metode *branch and bound* yaitu sebanyak 26 jenis kayu yang dimuat dengan keuntungan untuk total berat 15599 Kg yaitu Rp14168900.

Tabel 2 : Rekapitulasi Perbandingan Solusi

Barang Ke	Nilai			
	<i>Greedy By Weight</i>	<i>Greedy By Profit</i>	<i>Greedy By Density</i>	<i>Branch and Bound</i>
$x_1$	0	1	1	1
$x_2$	1	1	1	1
$x_3$	1	1	1	1

$x_4$	1	1	1	1
$x_5$	1	1	1	1
$x_6$	1	1	1	1
$x_7$	1	1	1	1
$x_8$	0	1	1	1
$x_9$	1	1	1	1
$x_{10}$	1	1	1	1
$x_{11}$	1	1	1	1
$x_{12}$	1	1	1	1
$x_{13}$	1	1	1	1
$x_{14}$	1	1	1	1
$x_{15}$	1	1	1	1
$x_{16}$	1	1	1	1
$x_{17}$	1	1	1	1
$x_{18}$	1	1	1	1
$x_{19}$	1	1	1	1
$x_{20}$	1	1	1	1
$x_{21}$	1	1	1	1
$x_{22}$	1	1	1	1
$x_{23}$	1	0	1	1
$x_{24}$	1	0	0	0
$x_{25}$	1	0	0	0
$x_{26}$	1	0	0	1
$x_{27}$	0	1	0	0
$x_{28}$	1	0	0	1
$x_{29}$	1	0	0	0
$x_{30}$	1	0	0	1

#### IV. KESIMPULAN

Untuk permasalahan *knapsack* 0-1 di PK. Murni Sanjaya ini dapat dimodelkan kedalam bentuk matematika . Perhitungan solusi optimal dan keuntungan total yang ada di PK. Murni Sanjaya dengan menggunakan algoritma *greedy* diperoleh hasil yang optimal pada perhitungan *greedy by profit* untuk keuntungan dengan jumlah berat yaitu 15319Kg yaitu Rp.13952000. Sedangkan untuk solusi optimal dengan menggunakan metode *branch and bound* dengan keuntungan untuk jumlah berat 15599Kg yaitu Rp14168900. Perhitungann solusi optimal yang dihasilkan dengan menggunakan perhitungan *branch and bound* lebih besar keuntungannya dibandingkan dengan menggunakan algoritma *greedy*.

## DAFTAR PUSTAKA

- [1]. Abdurrahman Rois, M., Penyelesaian Integer Knapsack Problem Menggunakan Algoritma Greedy, Dynamic Programming, Brute Force dan Genetic DY, DYNAMIC PROGRAMMING, BRUTE FORCE DAN GENETIC, *Telematika*, 2019, 12(2), pp. 87–97. doi: 10.35671/telematika.v12i2.841.
- [2]. Ambarwari, A. and Yanto, N. W., Penerapan Algoritma Greedy pada Permasalahan Knapsack untuk Optimasi Pengangkutan Peti Kemas, *Research Gate*, 2016.
- [3]. Ammar, M., Implementasi Algoritma Greedy Dalam Menyelesaikan Knapsack Problem Pada Jasa Pengiriman PT. Citra Van Titipan Kilat (TIKI) Kota Makassar, *Axiomath : Jurnal Matematika dan Aplikasinya*, 2019, 1(September), pp. 26–32.
- [4]. Arista, W. M., *Penerapan Algoritma Greedy Dan Dynamic Programming Pada Permasalahan Integer Knapsack*. Universitas Jember, 2013, Available at: <http://repository.unej.ac.id/handle/123456789/6944>.
- [5]. Dimiyati, T. T. and Dimiyati, A., *Operation Research : Model-model pengambilan keputusan*, Sinar Baru Algensindo, 2018, Bandung.
- [6]. Kosasi, S., Penyelesaian Bounded Knapsack Problem menggunakan Dynamic Programming, *Jurnal Ilmiah Ilmu Komputer*, 2013, 8(2), pp. 35–43. Available at: <http://e-journals.unmul.ac.id/index.php/JIM/article/view/107>.
- [7]. Prasetya, A. B., *Permasalahan Integer Knapsack dengan Algoritma Dynamic Programming dan Algoritma Branch and Bound*, Universitas Jember, 2015, Jember, Available at: <http://repository.unej.ac.id/handle/123456789/65320>.
- [8]. Pratiwi, A., Mulyono and Rochmad, Implementasi Algoritma Branch and Bound Pada 0-1 Knapsack Problem Untuk Mengoptimalkan Muatan Barang, *Unnes Journal of Mathematics*, 2014, 3(2), pp. 90–96. doi: 10.15294/ujm.v3i2.4302.
- [9]. Rachmawati, D. and Candra, A., Implementasi Algoritma Greedy Untuk Menyelesaikan, *Saintikom*, 2013, 12(3), p. 185-192.
- [10]. Sabaruddin, R., Solusi Optimum Minmax 0/1 Knapsack Menggunakan Algoritma Greedy, *Evolusi*, 2016, 4(2), pp. 76–82. doi: <https://doi.org/10.31294/evolusi.v4i2.703>.
- [11]. Setiawani, S., Metode Relaksasi Lagrange Untuk Menentukan Solusi Program Bilangan Cacah, *Majalah Ilmiah Matematika dan Statistika*, 2017, 17(2), pp. 49–60.
- [12]. Supatimah, S. S., Farida and Andriani, S., Optimasi Keuntungan dengan Metode Branch and Bound, 2019, 10(1), p. 23.

- [13]. Supriadi, D., Perbandingan Penyelesaian Knapsack Problem Secara Matematika, Kriteria Greedy Dan Algoritma Greedy, *Indonesian Journal on Computer and Information Technology*, 2016, 1(2), pp. 91–99.
- [14]. Supriana, I. W., Optimalisasi Penyelesaian Knapsack Problem Dengan Algoritma Genetika, *Lontar Komputer: Jurnal Ilmiah Teknologi Informasi*, 2016, p. 182. doi: 10.24843/LKJITI.2016.v07.i03.p06.