

Implementasi Algoritma Brudy dalam Persoalan Knapsack 0-1 di UD. Subur Tani Makmur

By Sisca Octarina



PROSIDING

SEMIRATA 2017 BIDANG MIPA

BKS-PTN WILAYAH BARAT

Jambi, Ratu Convention Center 12 - 14 Mei 2017

“Peran Sains, Teknologi dan Pendidikan MIPA dalam Menopang Sains Park, Teknopark, Serta Geopark Berbasis Argoindustri dan Lingkungan”



Penerbit: Fakultas Keguruan dan Ilmu Pendidikan (FKIP) bekerja sama dengan Fakultas Sains dan Teknologi (FST) Universitas Jambi

BUKU 1

MATEMATIKA

PROSIDING SEMIRATA 2017 BIDANG MIPA BKS-PTN WILAYAH BARAT

1 Editor:

Maison
Feri Tiona Pasaribu
Ahmad Syarkowi
Evtita
Novferma
Rosi Widia Asiani
Aulia UI Millah
Martina Asti Rahayu

Reviewer:

Maison
Evita Anggereini
Haris Effendi

Desain Sampul:

Taufan Dyusanda Putra

ISBN: 978-602-50593-0-8

Penerbit:

Fakultas Keguruan dan Ilmu Pendidikan (FKIP)
bekerjasama dengan Fakultas Sains dan Teknologi (FST) Universitas Jambi
Redaksi:

Kampus Unja Mendalo
Jl. Raya Jambi – Ma. Bulian Km. 15, Mendalo Indah
Jambi
Telp./Fax: 0741 - 583453

ISBN 978-602-50593-0-8



KATA PENGANTAR

Assalamualaikum Wr. Wb.

Alhamdulillah, segala puji bagi Allah SWT, Tuhan Yang Maha Esa, atas karunia yang telah dilimpahkan sehingga kegiatan Seminar dan Rapat Tahunan (SEMIRATA)-BKS PTN Bidang MIPA Wilayah Barat tahun 2017 dapat dilaksanakan secara baik.

Kegiatan SEMIRATA-BKS PTN Bidang MIPA Wilayah Barat tahun 2017 yang diamanahkan kepada Universitas Jambi sebagai penyelenggara dilaksanakan secara gabungan oleh Fakultas Sains dan Teknologi (FST) dan Fakultas Keguruan dan Ilmu Pendidikan (FKIP). Kegiatan telah dilaksanakan dengan sukses pada tanggal 12-14 Mei 2017 di Ratu Conference Hotel dan Swiss Bellin Hotel Jambi. Salah satu program utama adalah Seminar Nasional Sains dan Pendidikan MIPA dengan tema: “Peran Sains Teknologi dan Pendidikan MIPA dalam Menopang Sainspark, Teknopark serta Geopark berbasis Agroindustri dan Lingkungan”.

Sesi pleno seminar di Ratu Conference Center dipaparkan materi oleh dua pembicara utama yaitu akademisi Dr. Ir Yunus Kusumahbrata, M.Sc (Staf Ahli Kementerian ESDM) dan praktisi/birokrat Dr. H. Syahrial, M.P., (Bupati Tanjung Jabung Barat Prov. Jambi). Materi yang disajikan berisi topik Pengembangan Geopark, Teknopark dan Sainspark di Indonesia. Selain daripada itu, sesi paralel telah dipresentasikan secara oral lebih dari 600 judul makalah hasil penelitian yang disampaikan dalam 40 ruang seminar secara paralel. Dalam kegiatan komunikasi ilmiah secara langsung ini juga telah dimanfaatkan untuk menjalin jejaring agar lebih bersinergi dalam pengembangan Sains dan Pendidikan MIPA ke masa mendatang.

Supaya komunikasi ilmiah yang baik ini dapat juga tersampaikan ke komunitas ilmiah lain yang tidak dapat hadir pada kegiatan seminar, panitia memfasilitasi untuk menerbitkan makalah dalam bentuk Prosiding. Panitia juga tetap memberi kesempatan kepada peserta yang akan menerbitkan makalahnya di jurnal ilmiah, sehingga tidak seluruh materi yang disampaikan pada seminar diterbitkan dalam prosiding ini. Dalam proses penerbitan prosiding ini, panitia telah banyak dibantu oleh Tim Reviewer dan Tim Editor yang dikoordinir oleh Drs. Maison, M.Si., Ph.D, yang telah dengan sangat intensif mencurahkan waktu, tenaga dan pikiran untuk melakukan proses *plagiarism check*, review, dan editing. Untuk itu, panitia menyampaikan terima kasih dan penghargaan. Namun, panitia juga menyampaikan permohonan ma'af karena dengan sangat banyaknya makalah yang akan diterbitkan dalam prosiding ini, waktu yang dibutuhkan dalam proses penerbitan prosiding ini cukup lama, dan penerbitan prosiding tidak dilakukan dalam satu buku tetapi dalam empat buku prosiding. Semoga penerbitan prosiding ini selain SEMIRATA-BKS PTN Bidang MIPA Wilayah Barat tahun 2017 bermanfaat bagi para pemakalah dan penulis, juga dapat bermanfaat dalam pengembangan Sains dan Pendidikan MIPA di Indonesia.

Ucapan terima kasih disampaikan kepada Rektor Universitas Jambi, Dekan FST dan FKIP Universitas Jambi, Ketua Forum Rektor BKS wilayah Barat, Ketua BKS-MIPA Wilayah Barat, panitia dan semua pihak yang ikut menyelesaikan acara semirata.

Jambi, 2 Oktober 2017
Ketua Panitia

Dr. Kamid, M.Si

OPTIMALISASI PORTOFOLIO DENGAN MENGGUNAKAN SEPARABLE PROGRAMMING	80
Elly Rosmaini dan Nurhalimah Pane	
MODEL MATEMATIKA PENYEBARAN PENYAKIT MALARIA	89
Syarifah Meurah Yuni, Mahmudi	
ANALISIS PERSONAL FINANCIAL LITERACY MAHASISWA DALAM MERAMALKAN JUMLAH PENGELUARAN MENGGUNAKAN METODE EXPONENTIAL SMOOTHING DAN P-SPLINE FILTER SMOOTHING (Studi Kasus : Universitas Syiah Kuala, Banda Aceh)	95
Putri Atikah, Maisarah Defadz, Siti Husna F., Miftahuddin	
3 PERBANDINGAN ESTIMASI PARAMETER PADA DISTRIBUSI EKSPONENSIAL DENGAN MENGGUNAKAN METODE MAKSIMUM LIKELIHOOD DAN METODE BAYESIAN	105
Elsa Tria Noviadi , Rado Yendra dan Arisman Adnan	
PEMODELAN DEPENDENSI DATA KATAGORI MELALUI PENDEKATAN MODEL LOG- LINIER	112
Awal Isgiyanto, Syahrul Akbar	
APPLICATION OF FOURIER SMOOTHING BASIS FOR	124
Reza Ariska, Miftahuddin	
PENERAPAN ALGORITMA DYNAMIC PROGRAMMING PADA PERMASALAHAN KNAPSACK 0-1	134
Irmeilyana, Putra Bahtera Jaya Bangun, Dian Pratamawati, Winda Herfia Septiani Fakultas MIPA, Universitas Sriwijaya	
KETERKAITAN KETAKSAMAAN NILAI SINGULAR PADA PEMETAAN LINIER	145
Rolan Pane, Asli Sirait, Aziskhan	
IMPLEMENTASI ALGORITMA BRUDY DALAM PERSOALAN KNAPSACK 0-1 DI UD. SUBUR TANI MAKMUR	154
Indrawati , Sisca Octarina, Esrawati	

IMPLEMENTASI ALGORITMA BRUDY DALAM PERSOALAN KNAPSACK 0-1 DI UD. SUBUR TANI MAKMUR

Indrawati , Sisca Octarina, Esrawati

Jurusan Matematika Fakultas MIPA Universitas Sriwijaya

email: iin10juni@yahoo.com

ABSTRACT

Knapsack Problem is a matter of good selection process, which limited by the maximum quota that can be fit in a knapsack. The selection was based on the good combination to get the optimal profit and did not exceed the maximal knapsack capacity. UD. Subur Tani Makmur determined the selection of the fertilizer that can fulfilled the consumers need and optimized the profit. This research used Brudy (Brute Force-Greedy) algorithm. The maximum profits based on Brudy algorithm was from Brudy by Profit with amount Rp 117,996,500.00 and 5,981 kg of the total fertilizers. The maximum profit by the Brute Force algorithm was Rp 118,096,500.00 and 5,981 kg of the total fertilizers. The maximum profits based on Greedy algorithm was from Greedy by Density with amount Rp 118,087,500.00 and 5,975 kg of the total fertilizers. So, the maximum profit was obtained by Brute Force algorithm.

Keywords: *Knapsack, Brudy, Brute Force, Greedy.*

PENDAHULUAN

Setiap perusahaan menginginkan keuntungan yang besar, namun memiliki kendala pada keterbatasan sumber daya dan transportasi. Transportasi tidak hanya fokus pada penentuan rute dari sumber ke tempat tujuan tetapi juga pada pengaturan tempat penyimpanan barang sehingga memudahkan proses pengangkutan.

Proses pengangkutan barang harus memperhatikan pemilihan atau kombinasi susunan barang agar ruang yang terbatas dapat menampung barang secara maksimal. Permasalahan serupa dihadapi oleh UD. Subur Tani Makmur, perusahaan unit daerah di Jambi yang menjual pupuk dan kebutuhan pertanian. Berbagai macam jenis pupuk dan kebutuhan pertanian harus dibeli oleh perusahaan ini. UD. Subur Tani Makmur harus menentukan pemilihan barang yang tepat sehingga kebutuhan konsumen dapat terpenuhi, biaya transportasi minimal dan mendapatkan keuntungan yang maksimal.

Pemilihan dan pengangkutan barang pada perusahaan sering terkendala pada batasan kapasitas angkut maksimal dari sebuah kendaraan. Tiap barang memiliki berat masing-masing. Permasalahan pemilihan penyusunan barang dalam dunia Optimasi dikenal dengan istilah *Knapsack (Knapsack Problem)*. *Knapsack* adalah suatu permasalahan bagaimana memilih objek dari sekian banyak objek dan berapa besar objek tersebut akan disimpan sehingga diperoleh suatu penyimpanan yang optimal.

Knapsack terdiri atas beberapa permasalahan, yaitu masalah *knapsack 0-1 (integer knapsack)*, masalah *knapsack terbatas (bounded knapsack)*, masalah *knapsack ganda (multiple knapsack problem)*, dan masalah pengepakan (*bin packing problem*). Masalah *knapsack 0-1* merupakan masalah *knapsack tunggal*, karena sejumlah barang hanya akan dimasukkan ke

dalam satu *knapsack*. *Knapsack* 0-1 ini merupakan dasar untuk pengembangan beberapa masalah *knapsack* lainnya (Martello & Toth, 1990).

Permasalahan *knapsack* dapat diselesaikan dengan beberapa algoritma, seperti algoritma *Brute Force*, algoritma *Greedy* dan algoritma *Brudy (Brute Force-Greedy)*. Penelitian terdahulu tentang *knapsack* telah dilakukan. Arista (2013) menyimpulkan bahwa perhitungan menggunakan algoritma *Dynamic Programming* lebih optimal dibandingkan dengan algoritma *Greedy*. Septiani (2015) menyimpulkan bahwa penggunaan metode *Branch and Bound* pada permasalahan *knapsack* 0-1 di UD. Subur Tani Makmur menghasilkan keuntungan yang lebih maksimum dibandingkan dengan algoritma *Greedy*. Rusli (2008) menyimpulkan bahwa algoritma *Brudy* dan *Brute Force* lebih optimal dibandingkan dengan algoritma *Greedy*.

Algoritma *Brudy (Brute Force-Greedy)* merupakan gabungan dari algoritma *Brute Force* dan algoritma *Greedy*, dan lebih mengacu pada algoritma *Greedy*. Bedanya, algoritma *Greedy* mencari optimum lokal pada tiap langkahnya, sedangkan algoritma *Brudy* mencari optimum lokal pada tiap b langkah, dengan b sebagai nilai batas, b lebih dari 1 dan lebih kecil dari jumlah tahap.

Berdasarkan latar belakang tersebut, maka penelitian ini mengimplementasikan algoritma *Brudy (Brute Force-Greedy)* dalam persoalan *knapsack* 0-1 di UD. Subur Tani Makmur sehingga dapat memaksimalkan keuntungan. Permasalahan yang dibahas pada penelitian ini dibatasi pada penentuan keuntungan optimum masalah *knapsack* 0-1 dengan data berupa jenis-jenis pupuk, jumlah, berat, harga jual dan harga beli pupuk yang diangkut dengan sebuah truk di UD. Subur Tani Makmur. Penelitian ini mengabaikan ukuran pupuk, biaya transportasi, dan biaya bongkar muat. Data pada penelitian ini berdasarkan penelitian Septiani (2015).

KAJIAN LITERATUR

Knapsack

Knapsack merupakan suatu permasalahan bagaimana memilih objek dari sekian banyak objek dan berapa besar barang tersebut akan disimpan sehingga diperoleh suatu penyimpanan yang optimum. *Knapsack* memperhatikan objek yang terdiri dari n objek (1, 2, 3, ..., n) dimana setiap objek memiliki berat (w_i) dan keuntungan (v_i) serta memperhatikan kapasitas maksimum *knapsack* (Martello & Toth, 1990).

Knapsack terdiri dari beberapa persoalan, yaitu :

1. *Knapsack* 0-1 (*integer knapsack*) yaitu barang yang dimasukkan ke dalam media penyimpanan harus dimasukkan semua atau tidak sama sekali.
2. *Knapsack* terbatas (*bounded knapsack*) yaitu barang yang dimasukkan ke dalam penyimpanan dapat dimasukkan sebagian atau seluruhnya.
3. *Knapsack* tak terbatas (*unbounded knapsack*) yaitu jumlah barang yang dimasukkan ke dalam media penyimpanan macamnya tidak terbatas. (Surjawan & Susanto, 2015)

Persoalan *integer knapsack* mensyaratkan barang yang diangkut haruslah seluruhnya diangkut atau tidak sama sekali. Permasalahan *integer knapsack* mempunyai solusi persoalan yang dinyatakan sebagai himpunan :

$$X = \{x_1, x_2, x_3, \dots, x_n\} \quad (1)$$

$$x_i = \begin{cases} 0, & \text{jika jenis pupuk ke-}i \text{ tidak dimasukkan ke dalam media penyimpanan.} \\ 1, & \text{jika jenis pupuk ke-}i \text{ dimasukkan ke dalam media penyimpanan.} \end{cases}$$

$$i = \{1, 2, 3, \dots, 19\}$$

n = Jumlah pupuk dan x_i = Jenis pupuk ke- i . (Surjawan & Susanto, 2015)

Secara Matematika, persoalan *integer knapsack* dapat dirumuskan sebagai berikut :

Fungsi tujuan :

Maksimumkan atau Minimumkan

$$Z = \sum_{j=1}^n p_j x_j \quad (2)$$

dengan kendala :

$$\sum_{j=1}^n w_j x_j \leq c \quad (3)$$

$$x_j = 0 \text{ atau } 1, j = 1, \dots, n \quad (4)$$

dengan :

Z = fungsi tujuan dalam kasus ini adalah mencari keuntungan maksimal

p_j = keuntungan jenis pupuk ke- j , dengan

$$j = 1, 2, \dots, n$$

w_j = berat jenis pupuk ke- j , dengan $j = 1, 2, \dots, n$

n = banyak pupuk

c = kapasitas maksimum media penyimpanan

x_j = pembatas tanda yang nilainya 0 atau 1. (Martello & Toth, 1990)

2

Algoritma *Brute Force*

Brute Force adalah sebuah pendekatan langsung (*straight forward*) untuk memecahkan suatu masalah yang biasanya didasarkan pada pernyataan masalah dan definisi konsep yang dilibatkan. Dasarnya algoritma *Brute Force* adalah alur penyelesaian suatu permasalahan dengan cara berpikir yang sederhana dan tidak membutuhkan suatu pemikiran yang lama.

Algoritma *Brute Force* pada penyelesaian masalah *Knapsack* dilakukan dengan menghitung satu persatu keuntungan yang diperoleh dari semua kemungkinan pemilihan barang yang ada. Banyaknya kemungkinan pemilihan barang tersebut dapat dirumuskan sebagai 2^n . Adapun n adalah banyak barang yang akan dikirim. Jadi, seandainya banyak barang yang akan dikirim 5 buah, maka untuk mencari solusi optimal diperlukan $2^5 = 32$ kemungkinan. Memang, akan didapatkan hasil yang sangat optimal mengingat akan ditelusuri satu per satu kemungkinan yang ada, tetapi akan sangat membutuhkan waktu yang sangat lama (perhitungan manual) dan memori yang besar (jika menggunakan program komputer) untuk jumlah barang yang ada sangat banyak. (Rusli, 2008)

Algoritma *Greedy*

Algoritma *Greedy* membentuk solusi langkah per langkah. Terdapat banyak pilihan yang perlu dieksplorasi pada setiap langkah solusi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Pada setiap langkah dibuat pilihan optimum lokal dengan harapan bahwa langkah sisanya mengarah ke solusi optimum global. Keputusan yang telah diambil pada suatu langkah tidak dapat diubah lagi pada langkah selanjutnya.

Pencarian solusi permasalahan *integer knapsack* menggunakan algoritma *Greedy*, dimana masalah dipecahkan dengan memasukkan objek satu per satu ke dalam *knapsack*. Sekali objek tersebut dimasukkan ke dalam *knapsack*, maka objek tersebut tidak dapat dikeluarkan lagi. Ada beberapa strategi algoritma *Greedy* yang dapat digunakan untuk pemecahan masalah ini, tergantung pada properti objek yang akan dijadikan parameter *Greedy* :

1. *Greedy by weight*

Setiap langkah pada *Greedy by weight, knapsack* diisi dengan objek yang memiliki bobot lebih ringan terlebih dahulu. Tujuannya adalah untuk memaksimalkan jumlah kuantitas objek yang dapat masuk ke dalam *knapsack*.

2. Greedy by profit

Setiap langkah pada *Greedy by profit, knapsack* diisi dengan objek yang memiliki keuntungan lebih besar terlebih dahulu. Tujuannya adalah untuk memaksimalkan jumlah keuntungan dari objek yang dimasukkan ke dalam *knapsack*.

3. Greedy by density

Setiap langkah pada *Greedy by density, knapsack* diisi dengan objek yang memiliki rasio keuntungan dibagi dengan bobot (p_i/w_i) yang paling besar. Tujuannya adalah untuk memaksimalkan keuntungan dari objek yang dimasukkan, namun *knapsack* tetap diisi dengan jumlah objek sebanyak mungkin. (Wicaksono, 2007)

Algoritma Brudy

Algoritma *Brudy* adalah penggabungan dari algoritma *Brute Force* dan algoritma *Greedy*, lebih mengacu pada algoritma *Greedy*. Bedanya, algoritma *greedy* mencari optimum lokal pada tiap langkahnya, sedangkan algoritma *Brudy* mencari optimum lokal pada tiap b langkah, b adalah nilai batas, dengan catatan $1 < b < \text{jumlah tahap}$.

Misalkan suatu permasalahan mempunyai pengerjaan n tahap. Tahap ke-0 adalah kondisi awal. Algoritma *Brute Force* akan mencari semua cara sampai tahap ke- $(n-1)$. Algoritma *Greedy* akan mencari optimum dari tahap ke- i menuju tahap ke- $(i+1)$, sedangkan algoritma *Brudy* akan mencari optimum dari tahap ke- i menuju tahap ke- $(i+b)$ dan b bebas ditentukan pengguna. Misalkan dipilih b adalah k , berarti akan dicari optimum tahap ke-1 menuju tahap ke- $(1+k)$, selanjutnya dicari optimum tahap ke- $(1+k)$ menuju tahap ke- $(1+2k)$, dan seterusnya. Inilah kemiripan algoritma *Brudy* dengan algoritma *Brute Force*. Berbeda dengan algoritma *Greedy*, apabila tahap ke- $(i+b)$ melebihi solusi akhir, kurangilah b dengan 1 dan lakukan lagi untuk langkah selanjutnya.

Langkah-langkah penyelesaian *knapsack* menggunakan algoritma *Brudy* sebagai berikut :

1. Memilih nilai b dengan catatan $1 < b < \text{jumlah tahap}$.
2. Menentukan semua himpunan bagian dari banyak pupuk (n), dengan rumus :
$$C_{(n,r)} = \frac{n!}{(n-r)!r!} \quad (5) \quad n = \text{jumlah pupuk}, r = b = \text{nilai batas}$$
3. Mencari keuntungan (*profit*), berat (*weight*) dan perbandingan keuntungan dan berat (*density*) untuk tiap 2 benda sebanyak semua himpunan.
4. Menggunakan strategi *brudy by weight*, *brudy by profit* dan *brudy by density* untuk menentukan keuntungan yang paling maksimal. (Rusli, 2008)

3

METODE PENELITIAN

Data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dari penelitian Septiani, 2015 di UD. Subur Tani Makmur. Tahapan-tahapan yang dilakukan dalam menyelesaikan penelitian ini adalah sebagai berikut :

1. Mendeskripsikan data yang dibutuhkan yaitu data berat pupuk, banyak pupuk, harga beli pupuk dan harga jual pupuk.
2. Mengidentifikasi data yang didapat untuk mencari berat total berdasarkan banyak pupuk dan berat pupuk serta keuntungan berdasarkan harga jual dan harga beli dari masing-masing pupuk.
3. Mendefinisikan jenis-jenis pupuk dengan variabel x_1, x_2, \dots, x_i dengan $i = \{1, 2, 3, \dots, 19\}$.
4. Membentuk fungsi tujuan dan kendala dari permasalahan *knapsack*. Pada kasus ini didefinisikan :
 - a. Fungsi tujuannya yaitu memaksimalkan keuntungan penjualan dari tiap pupuk.

- b. Kendalanya yaitu berat total masing-masing barang dan dibatasi oleh kapasitas maksimum kendaraan.
5. Menghitung solusi optimum masalah *knapsack* 0-1 pada bongkar muat pupuk berdasarkan algoritma *Brute Force*.
6. Menghitung solusi optimum masalah *knapsack* 0-1 pada bongkar muat pupuk berdasarkan algoritma *Brudy (Brute Force-Greedy)*.
7. Menganalisis hasil akhir perhitungan menggunakan algoritma *Brudy* dan algoritma *Brute Force*.

HASIL DAN PEMBAHASAN

Permasalahan yang dihadapi adalah memaksimalkan daya angkut pupuk pada perusahaan dengan sebuah truk yang memiliki kapasitas maksimum angkut sebesar 6.000 kg. UD. Subur Tani Makmur mempunyai 19 jenis pupuk yang diangkut dan dikombinasikan agar mendapat keuntungan maksimum dan tidak melebihi kapasitas angkut kendaraan. Data pupuk yang diangkut disajikan dalam bentuk Tabel 1.

Penerapan algoritma *Brudy (Brute Force-Greedy)* dan algoritma *Brute Force* dalam mencari keuntungan maksimum, memerlukan data berupa berat total barang dan keuntungan dari masing-masing pupuk. Nilai berat total setiap pupuk diperoleh melalui pengidentifikasian data dengan mengalikan berat dari tiap pupuk (p_i) dengan banyaknya pupuk. Sedangkan untuk mencari nilai keuntungan dari tiap pupuk (v_i) dilakukan klarifikasi data dengan menghitung selisih antara harga jual dan harga beli. Tabel hasil klarifikasi data disajikan pada Tabel 1.

Variabel pada persoalan *knapsack* didefinisikan sebagai :

$$x_i, \quad i = 1, 2, 3, \dots, 19$$

$$x_i = \begin{cases} 0, & \text{jika jenis pupuk ke-}i \text{ tidak dimasukkan ke dalam media penyimpanan.} \\ 1, & \text{jika jenis pupuk ke-}i \text{ dimasukkan ke dalam media penyimpanan.} \end{cases}$$

Tabel 1. Data Jenis Pupuk di UD. Subur Tani Makmur

No	Jenis Pupuk	Berat Pupuk (kg)	Unit Satuan	Harga Beli (Rp)	Harga Jual (Rp)	Berat Total (kg)	Keuntungan Tiap Jenis Pupuk (Rp)	Keuntungan Total (Rp)
1	Pupuk NPK Basf	50	20	415.000	450.000	1.000	35.000	35.000.000
2	Pupuk Borax	25	30	270.000	300.000	750	30.000	22.500.000
3	KCI Merauke	50	15	275.000	300.000	750	25.000	18.750.000
4	Plastik Mulsa 25 kg	25	15	275.000	300.000	375	25.000	9.375.000
5	Pupuk Yaramila Mutiara	50	15	425.000	440.000	750	15.000	11.250.000
6	Plastik Mulsa 10 kg	10	10	185.000	200.000	100	15.000	1.500.000
7	Pupuk Merauke TPS 46	50	15	297.500	310.000	750	12.500	9.375.000

8	Antracol 70 WP	1	150	100.000	110.000	150	10.000	1.500.000
9	Amcozeb	1	200	90.000	100.000	200	10.000	2.000.000
10	Dithome M-45	1	100	110.000	120.000	100	10.000	1.000.000
11	Ferterra 0,4 gr	2	250	78.000	85.000	500	7.000	3.500.000
12	Kresnakum	1	100	35.000	40.000	100	5.000	500.000
13	Furadan 3 gr Kecil	1	150	10.750	15.000	150	4.250	637.500
14	Polybag 30 * 35 cm	25	12	16.000	20.000	300	4.000	1.200.000
15	Primodon 2 kg	20	20	21.000	25.000	400	4.000	1.600.000
16	Furadan 3 gr Besar	2	100	21.000	25.000	200	4.000	800.000
17	Mineral Feed	1	150	9.000	12.000	150	3.000	450.000
18	Sidafur	1	150	13.000	15.000	150	2.000	300.000
19	Top Mix (Pakan Ayam)	0,5	12	7.000	8.500	6	1.500	9.000
	Total					6.881	222.250	121.246.500

Sumber : (Septiani, 2015)

Langkah awal yang dilakukan adalah memilih nilai batas b dengan syarat $1 < b <$ jumlah tahap. Besar kecil nilai b tidak mempengaruhi nilai akhir atau hasil optimalnya, tapi mempengaruhi lama singkatnya waktu yang digunakan dalam mengerjakan permasalahan *knapsack*. Semakin besar nilai batas b maka iterasi yang digunakan semakin banyak. Hal ini terlihat bahwa jika nilai batas $b = 2$ menghasilkan 171 kombinasi, untuk nilai batas $b = 3$ menghasilkan 969 kombinasi, dan untuk nilai batas $b = 4$ menghasilkan 3.876 kombinasi, sehingga dengan cara pengerjaan yang sama untuk tiap nilai batas b yang berbeda maka dipilih nilai batas $b = 2$. Pada penelitian ini digunakan nilai batas $b = 2$ yang memiliki kombinasi lebih sedikit dibandingkan nilai batas $b = 3$ atau $b = 4$, sehingga lebih praktis dan efisien dari segi langkah dan waktu.

Cari banyaknya himpunan bagian dari 19 jenis pupuk tersebut yang mempunyai banyak anggota 2. Banyaknya himpunan bagian tersebut adalah 171.

Proses perhitungan selanjutnya diselesaikan dengan tiga strategi *Brudy* yang cara penyelesaiannya seperti algoritma *Greedy*. Adapun strategi tersebut adalah *Brudy by Weight*, *Brudy by Profit* dan *Brudy by Density*.

Perhitungan *Brudy by Weight*

Perhitungan *Brudy by Weight* adalah mengurutkan pupuk-pupuk berdasarkan berat (*weight*). Langkah pertama adalah mengurutkan pupuk-pupuk yang memiliki berat teringan dengan memilih i , kemudian diambil tiap dua pupuk yang terpilih sampai tidak melebihi kapasitas truk 6.000 kg. Apabila di dalam indeks i melebihi kapasitas truk 6.000 kg maka kurangi b dengan 1 dan pilih pupuk yang sesuai sampai $b = 0$ dan tidak ada lagi pupuk yang dapat dimasukkan ke dalam truk.

Berdasarkan perhitungan, pupuk yang akan diangkut ke dalam truk adalah jenis pupuk yang bernilai 1 pada solusi optimalnya, yaitu jenis $x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}$. Semua pupuk ini bernilai 1 sedangkan untuk jenis pupuk ke-1 (x_1) bernilai 0 karena jenis pupuk ke-1 sudah melebihi kapasitas truk, sehingga tidak

keuntungan yang paling besar dengan syarat total berat tidak melebihi kapasitas truk, maka untuk tiap kombinasi pupuk dipilih pupuk yang hanya memiliki keuntungan terbesar dengan syarat berat total kurang dari 6.000 kg. Perhitungan menggunakan algoritma *Brute Force* dilakukan untuk tiap pupuk, dimulai dari pengambilan 0 jenis pupuk dari 19 jenis pupuk, 1 jenis pupuk dari 19 jenis pupuk, 2 jenis pupuk dari 19 jenis pupuk, sampai 19 jenis pupuk dari 19 jenis pupuk. Jumlah kemungkinan untuk tiap jenis pupuk yang harus dicoba berbeda-beda jumlah caranya. Cara yang memiliki keuntungan paling besar dan berat totalnya tidak melebihi kapasitas truk tersebutlah yang dipilih. Hasil perhitungan menggunakan algoritma *Brute Force* dengan tiap-tiap barang dapat dilihat pada Tabel 2.

Tabel 2. Hasil Perhitungan *Brute Force*

Banyak Himpunan Bagian (nC)	Himpunan Solusi	Berat Total ($\sum w_i$)	Keuntungan Total ($\sum p_i$)
${}^{19}_0C = 1$	{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}	0	0
${}^{19}_1C = 19$	{1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}	1.000	35.000.000
${}^{19}_2C = 171$	{1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}	1.750	57.500.000
${}^{19}_3C = 969$	{1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}	2.500	76.250.000
${}^{19}_4C = 3.876$	{1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0}	3.250	87.500.000
${}^{19}_5C = 11.628$	{1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}	3.625	96.875.000
${}^{19}_6C = 27.132$	{1,1,1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0}	4.375	106.250.000
${}^{19}_7C = 50.388$	{1,1,1,1,1,0,1,0,0,0,1,0,0,0,0,0,0,0,0}	4.875	109.750.000
${}^{19}_8C = 75.582$	{1,1,1,1,1,0,1,0,1,0,1,0,0,0,0,0,0,0,0}	5.075	111.750.000
${}^{19}_9C = 92.378$	{1,1,1,1,1,0,1,0,1,0,1,0,0,0,1,0,0,0,0}	5.475	113.350.000
${}^{19}_{10}C = 92.378$	{1,1,1,1,1,1,1,0,1,0,1,0,0,0,1,0,0,0,0}	5.575	114.850.000
${}^{19}_{11}C = 75.582$	{1,1,1,1,1,1,1,1,0,1,0,0,0,1,0,0,0,0,0}	5.725	116.350.000
${}^{19}_{12}C = 50.388$	{1,1,1,1,1,1,1,1,1,1,0,0,0,1,0,0,0,0,0}	5.825	117.350.000
${}^{19}_{13}C = 27.132$	{1,1,1,1,1,1,1,1,1,1,0,1,0,1,0,0,0,0,0}	5.975	117.987.500
${}^{19}_{14}C = 11.628$	{1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0}	5.975	118.087.500
${}^{19}_{15}C = 3.876$	{1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,1}	5.981	118.096.500
${}^{19}_{16}C = 969$	{1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1}	5.981	117.646.500
${}^{19}_{17}C = 171$	{1,1,1,1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,1}	5.981	116.146.500
${}^{19}_{18}C = 19$	{0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}	5.881	86.246.500
${}^{19}_{19}C = 1$	{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}	6.881	121.246.500

Analisis Hasil Akhir Berdasarkan Algoritma *Brudy (Brute Force-Greedy)* dan Algoritma *Brute Force*

Keuntungan pada UD. Subur Tani Makmur berdasarkan perhitungan menggunakan algoritma *Brudy (Brute Force-Greedy)* dan algoritma *Brute* dapat dilihat pada Tabel 3.

Tabel 3. Hasil Akhir Perhitungan Total Keuntungan

	Algoritma <i>Brudy (Brudy by Profit)</i>	Algoritma <i>Brute Force</i>	Algoritma <i>Greedy (Greedy by)</i>
--	--	------------------------------	-------------------------------------

			<i>Density)</i>
Keuntungan Total ($\sum p_i$)	Rp 117.996.500,00	Rp 118.096.500,00	Rp 118.087.500,00
Berat Total ($\sum w_i$)	5.981 kg	5.981 kg	5.975 kg

Tabel 4. Pupuk yang Tidak Diangkut Berdasarkan Solusi Optimal

No	Algoritma	Pupuk yang Tidak Diangkut
1	<i>Brudy by Weight</i>	Pupuk NPK Basf
2	<i>Brudy by Profit</i>	Kresnakum, Polybag 30 * 35 cm, Furadan 3 gr Besar, Sidafur
3	<i>Brudy by Density</i>	Polybag 30 * 35 cm, Primodon 2 kg, Furadan 3 gr Besar
4	<i>Brute Force</i>	Primodon 2 kg, Furadan 3 gr Besar, Mineral Feed, Sidafur

Berdasarkan Tabel 3. dapat dilihat bahwa keuntungan terbesar diperoleh dengan menggunakan algoritma *Brute Force*. Pupuk yang tidak diangkut dari hasil perhitungan kedua algoritma dapat dilihat pada Tabel 4. Pupuk yang tidak diangkut pada perhitungan algoritma *Brudy by Weight* adalah pupuk yang mempunyai keuntungan yang paling besar. Sedangkan berdasarkan algoritma *Brudy by Profit* pupuk yang tidak diangkut adalah pupuk yang mempunyai keuntungan yang kecil $\{(p_i) \leq 1,2 \text{ juta}\}$. Pada algoritma *Brudy by Density* pupuk yang tidak diangkut adalah barang-barang yang memiliki densitas $\left(\frac{p_i}{w_i}\right) \leq 4.000$. Selain itu pupuk-pupuk yang memiliki keuntungan paling kecil tidak diangkut berdasarkan perhitungan dengan algoritma *Brute Force*.

KESIMPULAN

Keuntungan optimal UD. Subur Tani Makmur didapat dari perhitungan *Brudy by Profit* yaitu sebesar Rp 117.996.500,00 dengan total berat pupuk yang diangkut adalah 5.981 kg dan berdasarkan algoritma *Brute Force* yaitu sebesar Rp 118.096.500,00 dengan total berat pupuk yang diangkut adalah 5.981 kg. Keuntungan yang lebih optimal diperoleh menggunakan algoritma *Brute Force*.

DAFTAR PUSTAKA

- Aminudin. (2005). *Prinsip-Prinsip Riset Operasi*. Erlangga.
- W. M. Arista. (2013). Algoritma *greedy* dan *dynamic program* pada permasalahan *integer knapsack*. *Skripsi*. Jember: Universitas Jember.
- S. Martello & P. Toth, (1990). *Knapsack Problem*. John Wiley & Sons, Singapore.
- T.K. Rusli. (2008). Perancangan program aplikasi penyelesaian masalah knapsack dengan metode algoritma *brudy* (Studi Kasus: PT. Siam Square Buanatrans). *Skripsi*. Jakarta: Universitas Bina Nusantara.
- W.H. Septiani. (2015). Implementasi algoritma *greedy* dan metode *branch And bound* dalam persoalan *knapsack* 0-1 Di UD. Subur Tani Makmur. *Skripsi*. Universitas Sriwijaya.(Tidak Dipublikasikan).
- Sudradjat. (2010). *Pendahuluan Penelitian Operasional*. Diktat Kuliah. Jurusan Matematika FMIPA Universitas Padjadjaran.

- D.J. Surjawan dan I. Susanto. (2015). Aplikasi optimalisasi muat barang dengan penerapan algoritma dynamic programming pada persoalan integer knapsack. *Jurnal Teknik Informatika dan Sistem Informasi*, Vol. 1, No. 2, 152.
- P.A. Wicaksono. (2007). Eksplorasi algoritma *brute force*, *greedy* dan pemrograman dinamis pada penyelesaian masalah 0/1 knapsack. *Makalah IF2251 Strategi Algoritmik*, ITB.

Sertifikat

Diberikan Kepada:

SISCA OCTARINA

Atas Partisipasinya Sebagai

PENAKKALAH

Pada Kegiatan

SEMIRATA 2017 Bidang MIPA BKS-PTN Barat

"Peran Sains, Teknologi dan Pendidikan MIPA dalam Menopang
Sains Park, Teknopark, serta Geopark berbasis
Agroindustri dan Lingkungan"

Ratu Convention Center, Jambi 12-14 Mei 2017

Ketua BKS PTN Bidang MIPA



Dr. Teuku M. Iqbalisyah, M.Sc

NIP. 1971101019977031003
BKS PTN Barat
Bidang MIPA

Ketua Panitia


/102
ATA-RIMES
BKS-PTN Barat Bidang MIPA
FKIP & IST UNIVERSITAS JAMBI
Dr. Kamid, M.Si.
NIP. 196609041992031002



SEMIRATA 2017
BIDANG MIPA BKS-PTN
WILAYAH BARAT
UNIVERSITAS JAMBI



Implementasi Algoritma Brudy dalam Persoalan Knapsack 0-1 di UD. Subur Tani Makmur

ORIGINALITY REPORT

2%

SIMILARITY INDEX

PRIMARY SOURCES

1 Fitriani Jingga, Muliadi Muliadi, Risiko Risiko. "Kondisi Arus Musim Barat di Perairan Pantai Kijing Kabupaten Mempawah Kalimantan Barat", Jurnal Laut Khatulistiwa, 2021 30 words — 1%

[Crossref](#)

2 Condro Wibawa. "OPTIMALISASI RUTE WISATA DI YOGYAKARTA MENGGUNAKAN METODE TRAVELLING SALESMAN PERSON DAN ALGORITMA BRUTE FORCE", Jurnal Teknik dan Science, 2022 26 words — 1%

[Crossref](#)

3 Eviana Hikamudin. "ESTIMASI KEMAMPUAN SISWA DALAM UJIAN NASIONAL MENGGUNAKAN METODE BAYES", Jurnal Penelitian Kebijakan Pendidikan, 2018 25 words — 1%

[Crossref](#)

EXCLUDE QUOTES ON

EXCLUDE SOURCES < 1%

EXCLUDE BIBLIOGRAPHY ON

EXCLUDE MATCHES < 10 WORDS