PERBANDINGAN PENYELESAIAN KNAPSACK PROBLEM SECARA MATEMATIKA, KRITERIA GREEDY DAN ALGORITMAGREEDY

Deddy Supriadi
Manajemen Informatika, AMIK BSI Tasikmalaya
deddy.dys@bsi.ac.id

Abstract- *Knapsack problem is a problem how to select object from many object sand how weight it will be saved in order to obtain anoptimal storage with consider which consists of n objects(1,2,3, ...) where each objecthas a weight(Wi) and profit(Pi). Further more capacity from storage (M) and probability from each object must be considered. In this paper discuss how to solving knapsack problem with three ways. They are mathematic, greedy criteria and greedy algorithm. They have different way to solve knapsack problem. After the comparison, more optimized and more easily is greedy criteria. But more difficult and the result are not optimal use mathematic. Greedy algorithm will be effective if it is non decreasing. This method is faster but we must understand before hand about greedy algorithm*
Keywords: greedy, knapsack, mathematic, profit, weight

Abstrak - Masalah *Knapsack* merupakan suatu permasalahan bagaimana memilih objek dari sekian banyak dan berapa besar objek tersebut akan disimpan sehingga diperoleh suatu penyimpanan yang optimal dengan memperhatikan objek yang terdiri dari n objek (1,2,3,…) dimana setiap objek memiliki bobot (Wi) dan profit (Pi) dengan memperhatikan juga kapasitas dari media penyimpanan sebesar M dan nilai probabilitas dari setiap objek (Xi).Dalam jurnal ini membahas *metode kajian* mengenai cara menyelesaikan permasalahan *Knapsack* ini dengan membandingkan tiga cara, yaitu dengan cara matematika, kriteria greedy, dan dengan algoritma greedy. Masing-masing cara ini memiliki perbedaan dalam penyelesaiannya. Setelah dilakukan perbandingan maka lebih optimal dan lebih mudah dikerjakan yaitu dengan cara kriteria greedy. Sedangkan yang lebih sulit dan hasilnya tidak optimal digunakan secara matematika. Untuk cara algoritma greedy akan efektif apabila disusun secara tidak naik (*non descreasing*). Cara ini memang lebih cepat akan tetapi kita harus memahami terlebih dahulu tentang algoritma greedy.
Kata Kunci: greedy, *knapsack*, matematika, profit, bobot

I. PENDAHULUAN

Dalam penyimpanan beberapa objek kita kadang merasa kesulitan apabila media penyimpanannya terbatas. Hal itu sangat menyulitkan karena kita harus mengatur agar objek-objek tersebut dapat tersimpan kedalam media penyimpanan yang terbatas.Kita harus mengatur objek mana saja yang dipilih yang akan disimpan dan seberapa besar objek tersebut disimpan sesuai dengan kapasitas media penyimpanan yang ada.

Permasalahan tersebut dikenal dengan *Knapsack Problem*. *Knapsack* dapat diartikan sebagai karung, kantung, atau buntilan. Karung digunakan untuk memuat sesuatu. Tujuan *Knapsack problem* untuk mendapatkan keuntungan yang maksimum dari pemilihan barang tanpa melebihi kapasitas daya tampung media penyimpanan tersebut. 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. Setiap objek itupun tidak harus kita masukkan seluruhnya. Tetapi bisa juga sebagian saja.

Dalam menyelesaikan *Knapsack Problem* tersebut bisa menggunakan cara matematika, kriteria greedy, dan dengan algoritma greedy. Ketiga cara penyelesaian tersebut memiliki cara masing-masing dalam menyelesaikan *knapsack problem*. Dengan demikian kita bisa membandingkan 3 metode dalam menyelesaikan 1 kasus yaitu dengan cara penyelesaian *knapsack problem* secara matematika, kriteria greedy dan algoritma greedy dengan menggunakan sebuah kasus. Setelah ketiga cara tersebut dilakukan maka akan diperoleh hasil cara mana yang lebih baik dan cepat yang dapat digunakan untuk menyelesaikan *knapsack problem*.

Sebagai referensi pembahasan sebelumnya bisa dilihat di https://hendryprihandono.wordpress.com/2009/01/03/knapsack-problem-dengan-algoritma-dan-metode-greedy/.

II. KAJIAN LITERATUR

A. *Knapsack Problem*

Menurut Komang (2010) *Knapsack problem* merupakan salah satudari persoalan klasik yang banyak ditemukan dalam literatur-literatur lama dan hingga kini permasalahan tersebut masih sering ditemukan dalam kehidupan sehari-hari. Contoh nyata dari Knapsack Problem ini misalnya, jika ada seorang pedagang barang kebutuhan rumah tangga yang berkeliling menggunakan gerobak. Tentu saja gerobaknya memiliki kapasitas maksimum, sehingga ia tidak bisa memasukkan semua barang dagangannya dengan seenak hatinya. Pedagang tersebut harus memilih barang-barang mana saja yang harus ia angkut, dengan pertimbangan berat dari barang yang dibawanya tidak melebihi kapasitas maksimum gerobak dan memaksimalkan profit dari barang-barang yang dibawa.

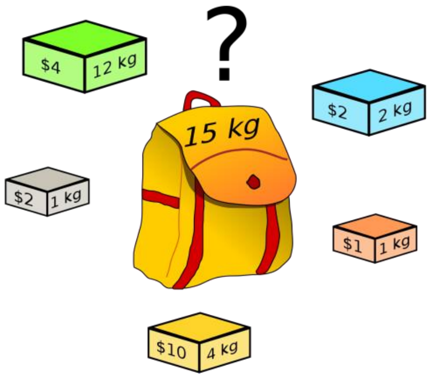
Menurut Dian(2013) *Knapsack* adalah tas atau karung. Karung digunakan untuk memuat sesuatu. Dan tentunya tidak semua objek dapat ditampung di dalam karung tersebut. Karung tersebut hanya dapat menyimpan beberapa objek dengan total ukurannya (*weight*) lebih kecil atau sama dengan ukuran kapasitas karung. Ilustrasi permasalahan dapat dilihatpada gambar 1.

Pada gambar 1 terlihat terdapat sebuah tas berkapasitas 15 kg. Dan terdapat 5 barang dengan berat dan keuntungannya masing - masing. Yang menjadi persoalan adalah barang mana saja yang harus dimasukan ke dalam tas.

Knapsack Problem secara matematis dapat ditulis sebagai berikut:

Diberikan bobot knapsack adalah M. Diketahui n buah objek yang masing-masing bobotnya adalah w1, w2, …, wn. Tentukan nilai bi sedemikian sehingga:

$$M = b1w1 + b2w2 + \ldots + bnwn$$



Gambar 1. Ilustrasi Permasalahan *Knapsack Problem*
Sumber: Dian (2013:186)

yang dalam hal ini, bi bernilai 0 atau 1. Jika bi = 1, berarti objek i dimasukkan ke dalam *knapsack*, sebaliknya jika bi = 0, objek i tidak dimasukkan. Berhubung nilai bi 0 dan 1 maka masalah ini sering juga disebut sebagai *Knapsack* 0/1.

Dalam teori algoritma, persoalan *knapsack* termasuk ke dalam kelompok NP-complete. Persoalan yang termasuk NPcomplete tidak dapat dipecahkan dalam orde waktu polinomial

B. Secara Matematika

Pada cara ini, kita harus memperhatikan nilai probabilitas dari setiap barang, karena nilai inilah sebagai penentunya dengan memperhatikan nilai probabilitas (X*i*) yaitu 0≤Xi≤1.Nilai Xi bisa sangat beragam, dari 0, 0.1, 0.01, 0.001, ... , 1.

Menurut Yulikuspartono (2001) dalam beberapa literatur, istilah lain dari fungsi tujuan dapat disebut sebagai fungsi utama atau juga fungsi objektif yaitu fungsi yang menjadi penyelesaian permasalahan dengan mendapatkan solusi yang optimal.

Solusi dimaksud= menemukan nilai/*profit* yang maks. untuk jumlah obyek yang dimuat dalam ransel sehingga sesuai kapasitas.

Fungsi TujuanMaksimum : $\sum_{I=1}^{n} Pi\ Xi$

Fungsi pembatas= fungsi *subyektif* = fungsi yang bertujuan untuk memberikan batas maksimal dari setiap obyek untuk dapat dimuat dalam ransel sehingga kapasitasnya tidak melebihi dari jumlah maksimal daya tampung ransel.

Fungsi Pembatas : $\sum_{i=1}^{n} W_i X_i \leq M$

dimana : $0 \leq X_i \leq 1$; $P_i > 0$; $W_i > 0$

Dengan cara ini sulit untuk menentukan yang paling optimal sebab kita harus mencari nilai probabilitas yang tersebar antara 0 dan 1, $0 \leq X_i \leq 1$ untuk setiap objek.

C. Kriteria Greedy

Menurut Dian dan Ade dalam *paper* Implementasi Algoritma Greedy untuk “Menyelesaikan Masalah *Knapsack Problem*” Pada penyelesaian Knapsack Problem dengan kriteria greedy terdapat 3 tahapan yang dapat digunakan yaitu :

1. *Greedy by Weight*
   Pada setiap langkah pilih objek yang mempunyai berat teringan. Mencoba memaksimumkan keuntungan dengan memasukkan sebanyak mungkin objek ke dalam *knapsack*.
   Pertama kali yang dilakukan adalah program mengurutkan secara menaik objek-objekberdasarkan weightnya. Kemudian baru diambil satu-persatu objek yang dapat ditampung oleh *knapsack* sampai *knapsack* penuh atau sudah tidak ada objek lagi yang bisa dimasukkan.
2. *Greedy by profit*
   Pada setiap langkah, pilih objek yang mempunyai keuntungan terbesar. Mencoba memaksimumkan keuntungan dengan memilih objek yang paling menguntungkan terlebih dahulu.
   Pertama kali yang dilakukan adalah program mengurutkan secara menurun objek-objek berdasarkan profitnya. Kemudian baru diambil satu-persatu objek yang dapat ditampung oleh *knapsack* sampai *knapsack* penuh atau sudah tidak ada objek lagi yang bisa dimasukkan.
3. *Greedy By Density*
   Pada setiap langkah *knapsack* di isi dengan objek yang mempunyai pi/wi terbesar, dimana p adalah keuntungan dan w adalah berat barang. Mencoba memaksimumkan keuntugan dengan memilih objek yang mempunyai *density* per unit berat terbesar.
   Pertama kali yang dilakukan adalah program mencari nilai profit per berat tiap - tiap unit (*density*) dari tiap- tiap objek. Kemudian objek-objek tersebut diurutkan berdasarkan *density*-nya. Kemudian baru diambil satu-persatu objek yang dapat ditampung oleh *knapsack* sampai knapsack penuh atau sudah tidak ada objek lagi yang bias dimasukkan.

D. Algoritma Greedy

Menurut Dian (2013) Algoritma Greedy merupakan algoritma yang lazim untuk memecahkan persoalan optimasi meskipun hasilnya tidak selalu merupakan solusi yang optimum. Sesuai arti harfiah, Greedy berarti tamak. Prinsip utama dari algoritma ini adalah mengambil sebanyak mungkin apa yang dapat diperoleh sekarang. Untuk memecahkan persoalan dengan algoritma Greedy, kita memerlukan elemen-elemen sebagai berikut.

1. Himpunan Kandidat (C)
   Himpunan ini berisi elemen-elemen pembentuk solusi.
2. Himpunan Solusi, (S)
   Himpunan ini berisi kandidat yang terpilih sebagai solusi persoalan. Dengan kata lain, himpunan solusi adalah himpunan bagian dari himpunan kandidat.
3. Fungsi Seleksi
   Fungsi seleksi merupakan fungsi yang ada pada setiap langkah memilih kandidat yang paling memungkinkan guna mencapai solusi optimal.
4. Fungsi Kelayakan (*Feasible*)
   Fungsi kelayakan adalah fungsi yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak dan tidak melanggar batasan atau constraints yang ada.
5. Fungsi Objektif
   Fungsi objektif adalah fungsi yang memaksimumkan atau meminimumkan nilai solusi.

Skema umum algoritma Greedy adalah sebagai berikut:

1. Inisialisasi S dengan kosong.
2. Pilih sebuah kandidat C dengan fungsi seleksi.
3. Kurangi C dengan kandidat yang sudah dipilih dari langkah (b) di atas.
4. Periksa apakah kandidat yang dipilih tersebut bersama-sama dengan

himpunan solusi membentuk solusi yang layak atau feasible (dengan fungsi kelayakan).
5. Periksa apakah himpunan solusi sudah memberikan solusi yang lengkap serta optimal (dengan fungsi objektif).

III. METODE PENELITIAN

Penelitian ini dilakukan dengan langkah-langkah sebagai berikut:

1. Menyelesaikan permasalahan *knapsack* dengan menggunakan cara matematika, kriteria greedy dan algoritma greedy dengan menggunakan suatu kasus.
2. Melakukan perbandingan dari ketiga cara yaitu secara matematika, kriteria greedy, algoritma greedy setelah menyelesaikan suatu kasus *knapsack problem*.
3. Menghasilkan cara mana yang lebih baik yang digunakan dalam menyelesaikan *knapsack problem*.

IV. PEMBAHASAN

Knapsack Problem merupakan masalah optimasi kombinatorial. Sebagai contoh dalam dunia nyata, permasalahan Knapsack ini sering sekali digunakan terutama pada bidang (jasa) pengangkutan barang (seperti pengangkutan peti kemas dalam sebuah kapal), pengisian barang di bagasi, pengisian barang di suatu perusahaan, pengoptimalisasi karyawan dalam suatu badan usaha.

Terdapat beberapa algoritma yang digunakan untuk menyelesaikan permasalahan tersebut. Pemilihan algoritma dalam penyelesaian *knapsack problem* ini sangat penting. Penggunaan algoritma yang tepat akan membantu penyelesaikan kasus Knapsack dengan baik. Dalam *paper* ini dibahas tentang penyelesaian *knapsack problem* dengan cara matematika, kriteria greedy dan algoritma greedy. Ketiga cara tersebut diterapkan dalam sebuah contoh kasus. Contoh kasusnya adalah:

Diketahui 3 barang yang akan disimpan pada suatu tempat yang memiliki kapasitas maksimal sebesar 25 Kg. Berat masing-masing barang adalah:
Barang pertama:20 Kg
Barang kedua :17 Kg
Barang ketiga : 12 Kg
dimana setiap barang memiliki profit sebesar masing-masing:
Barang pertama: 27
Barang kedua : 26
Barang ketiga : 17
Sesuai dengan contoh kasus diatas maka Kemudian menentukan barang mana saja yang dapat disimpan ke dalam tempat penyimpanan sehingga diperoleh nilai profit yang maksimal.

Tabel 1. Keterangan Barang

Barang ke-	Berat (W)	Profit (P)
1	20	27
2	17	26
3	12	17

A. Secara Matematika

Dengan cara matematika ini nilai probabilitas harus diperhatikan. Sebagai penentunya dengan memperhatikan nilai probabilitas (X*i*) yaitu 0≤Xi≤1.
Nilai Xi bisa sangat beragam, dari 0, 0.1, 0.01, 0.001, ... , 1.
Diketahui: n = 3, (1, 2, 3)
kapasitas, M = 25
Untuk berat masing-masing barang:
W_1 : 20
W_2 : 17
W_3 : 12
Untuk Profit masing-masing barang:
P_1: 27
P_2: 26
P_3: 17
Nilai probabilitas $0 \le Xi \le 1$
Solusi ke Nilai Probabilitas Fungsi Pembatas Fungsi Tujuan

$$\sum Wi.Xi \le M \sum Pi.Xi \text{ (Maximum)}$$

$$(X_1, X_2, X_3)\ (W_1.\ X_1) + (W_2.\ X_2) + (W_3.\ X_3) \le M\ (P_1.\ X_1) + (P_2.\ X_2) + (P_3.\ X_3)$$

Untuk profit yang terbesar, nilai probabilitasnya diberi nilai 1 dan yang terkecil diberi nilai 0.

P_1 : 27 → X_1 : 1 karena nilainya terbesar
P_2 : 26 → X_2 : ?
P_3 : 17 → X_3 : 0 karena nilainya terkecil

$$\sum_{i=1}^{3} = Wi.\ Xi \le M$$

$= W_1.X_1 + W_2.X_2 + W_3.X_3 \leq M$
$(20.1) + (17X_2) + (12.0) \leq 25$
$20 + 17X_2 \leq 25$
$17X_2 \leq 25\text{-}20$
$17X_2 \leq 5$
$X_2 \leq \frac{5}{17}$

Jadi, nilai X_2 nya adalah $\frac{5}{17}$

Untuk tahap ke-1
(X_1, X_2, X_3) $(W_1. X_1) + (W_2. X_2) + (W_3. X_3) \leq M$ $(P_1. X_1) + (P_2. X_2) + (P_3. X_3)$
$(1, \frac{5}{17}, 0)$ $(20.1) + (17.\frac{5}{17}) + (12.0) \leq 25$ $(27.1) + (26.\frac{5}{17}) + (17.0)$

Kemudian nilai probabilitasnya diganti,
Nilai $X_1 = 1$
Nilai $X_2 = 0$
Nilai X_3 = ?

$$\sum_{i=1}^{3} = Wi. Xi \leq M$$
$= W_1.X_1 + W_2.X_2 + W_3.X_3 \leq M$
$(20.1) + (17.0) + (12.X_3) \leq 25$
$20 + 12X_3 \leq 25$
$12X_3 \leq 25\text{-}20$
$12X_3 \leq 5$
$X_3 \leq \frac{5}{12}$

Jadi, nilai X_2 nya adalah $\frac{5}{12}$

Untuk tahap ke-2
(X_1, X_2, X_3) $(W_1. X_1) + (W_2. X_2) + (W_3. X_3) \leq M$ $(P_1. X_1) + (P_2. X_2) + (P_3. X_3)$

$(1,0,\frac{5}{12})$ $(20.1) + (17.0)+(12.\frac{5}{12}) \leq 25$ $(27.1) + (26.0)+(17.\frac{5}{12})$

Untuk tahap ke-3 nilai probabilitasnya menjadi:
Nilai $X_1 = 0$
Nilai $X_2 = 1$
Nilai X_3 = ?

$$\sum_{i=1}^{3} = Wi. Xi \leq M$$
$= W_1.X_1 + W_2.X_2 + W_3.X_3 \leq M$
$(20.0) + (17.1) + (12.X_3) \leq 25$
$17 + 12X_3 \leq 25$
$12X_3 \leq 25\text{-}17$
$12X_3 \leq 8$
$X_3 \leq \frac{8}{12} = \frac{2}{3}$

Jadi, nilai X_3 nya adalah $\frac{2}{3}$

(X_1, X_2, X_3) $(W_1. X_1) + (W_2. X_2) + (W_3. X_3) \leq M$ $(P_1. X_1) + (P_2. X_2) + (P_3. X_3)$

$(0,1,\frac{2}{3})$ $(20.0) + (17.1)+(12.\frac{2}{3}) \leq 25$ $(27.0) + (26.1)+(17.\frac{2}{3})$

Kemudian seterusnya hitung kembali dengan nilai probabilitas yang dirubah. Dengan cara ini sulit untuk menentukan yang paling optimal sebab kita harus mencari nilai probabilitas yang tersebar antara 0 dan 1, 0 ≤ Xi ≤ 1 untuk setiap objek

Tabel 2. Hasil Perhitungan Secara Matematika

Solusi ke	**(X1, X2, X3)**	**∑WiXi**	**∑PiXi**
1	$(1, \frac{5}{17}, 0)$	25	34,6
2	$(1, 0, \frac{5}{12})$	25	34,1
3	$(0, 1, \frac{2}{3})$	25	**37,3**
...	...	...	...

Maka nilai profit terbesarnya adalah 37,3. Dengan demikian diperoleh pengaturan barang yang akan dimasukkan adalah:
X_1: 0
X_2:1
X_2: $\frac{2}{3}$
Dengan artian bahwa:
Barang kesatu dengan nilai 0 maka barang tersebut tidak usah dimasukkan.
Barang kedua diambil 1 bagian yaitu 17Kg

Barang ketiga dengan nilai $\frac{2}{3}$ maka barang tersebut
Diambil $\frac{2}{3}$ bagian dengan artian: $\frac{2}{3}$ X 12 = 8Kg
Jadi, total barang yang disimpan adalah 0+17+8=25Kg sesuai dengan berat maksimal tempat tersebut.

B. Kriteria Greedy

Berdasarkan cara kriteria greedy maka dilakukan dengan 3 tahap:

1. *Greedy by Profit*
   Dengan cara ini maka dicari terlebih dahulu *Profit* terbesar dari ketiga barang tersebut.
   P_1 : 27
   P_2 : 26
   P_3 : 17
   Dari ketiga barang tersebut maka barang yang profitnya terbesar adalah barang ke-1. Barang dengan profit terbesar diberi nilai 1 dan dengan profit terkecil diberi nilai 0 sesuai dengan fungsi pembatas. Sedangkan barang lainnya nilai probabilitasnya harus dihitung terlebih dahulu.
   Nilai probabilitasnya $0 \le xi \le 1$
   P_1 : 27 ➔ X_1 : 1 karena nilainya terbesar
   P_2 : 26 ➔ X_2 : ?
   P_3 : 17 ➔ X_3 : 0 karena nilainya terkecil
   Untuk nilai X_2 harus dicari terlebih dahulu dengan rumus :
   $$\sum_{i=1}^{3} = Wi.\ Xi \le M$$
   $$= W_1.X_1 + W_2.X_2 + W_3.X_3 \le M$$
   $$(20.1) + (17X_2) + (12.0) \le 25$$
   $$20 + 17X_2 \le 25$$
   $$17X_2 \le 25-20$$
   $$17X_2 \le 5$$
   $$X_2 \le \frac{5}{17}$$
   Jadi, nilai X_2 nya adalah $\frac{5}{17}$

2. *Greedy by Weight*
   Untuk greedy by weight maka dicari berat yang paling minimal. Untuk nilai probabilitasnya maka apabila berat yang terkecil diberi nilai 1, yang berat terbesar maka diberi nilai 0.
   Nilai probabilitasnya $0 \le xi \le 1$
   W_1 : 20 ➔ X_1 : 0 karena beratnya terbesar
   W_2 : 17 ➔ X_2 : ?
   W_3 : 12➔ X_3 : 1 karena beratnya terkecil
   Untuk nilai X_2 harus dicari terlebih dahulu dengan rumus :
   $$\sum_{i=1}^{3} = Wi.\ Xi \le M$$
   $$= W_1.X_1 + W_2.X_2 + W_3.X_3 \le M$$
   $$(20.0) + (17X_2) + (12.1) \le 25$$
   $$12+ 17X_2 \le 25$$
   $$17X_2 \le 25-12$$
   $$17\ X_2 \le 13$$
   $$X_2 \le \frac{13}{17}$$
   Jadi, nilai X_2 nya adalah $\frac{13}{17}$

3. *Greedy by Density*
   Greedy by density ini dicari nilai perbandingan yang terbesar antara Profit dengan berat. Untuk nilai perbandingan yang terbesar diberi nilai 1, untuk nilai yang terkecil diberi nilai 0.
   Nilai probabilitasnya $0 \le xi \le 1$
   $\frac{P_1}{W_1}$: $\frac{27}{20}$: 1,35 ➔ X_1 : 0 Karena nilainya terkecil
   $\frac{P_2}{W_2}$: $\frac{26}{17}$: 1,5 ➔ X_2 :1 Karena nilainya terbesar
   $\frac{P_3}{W_3}$: $\frac{17}{12}$: 1,4➔ X_3 : ?
   Untuk nilai X_2 harus dicari terlebih dahulu dengan rumus :
   $$\sum_{i=1}^{3} = Wi.\ Xi \le M$$
   $$= W_1.X_1 + W_2.X_2 + W_3.X_3 \le M$$
   $$(20.0) + (17.1) + (12.X_3) \le 25$$
   $$17 + 12X_3 \le 25$$
   $$12X_3 \le 25-17$$
   $$12X_3 \le 8$$
   $$X_3 \le \frac{8}{12} = \frac{2}{3}$$
   Jadi, nilai X_3nya adalah $\frac{2}{3}$

Berdasarkan dari *greedy by profit, greedy by weight, greedy by density* maka dihasilkan

Tabel 3. Hasil Perhitungan Kriteria Greedy ke-1

Solusi ke-	(X_1,X_2,X_3)	∑Wi.Xi	∑Pi. Xi
PiMax	$(1, \frac{5}{17}, 0)$	25	?
WiMin	$(0, \frac{13}{17}, 1)$	25	?
$\frac{Pi}{Wi}$ Max	$(0,1,\frac{2}{3})$	25	?

Untuk mencari nilai ∑Pi.Xi dari masing-masing solusi maka harus dimasukkan

terlebih dahulu profitnya dan nilai probabilitasnya.
Untuk Pi Max:

$\sum Pi.Xi = (P_1.X_1) + (P_2.X_2) + (P_3.X_3)$
$= (27.\ 1) + (26.\frac{5}{17}) + (17.0)$
$= 27 + 7{,}6 + 0$
$= 34{,}6$

Untuk Wi Min

$\sum Pi.Xi = (P_1.X_1) + (P_2.X_2) + (P_3.X_3)$
$= (27.\ 0) + (26.\frac{13}{17}) + (17.1)$
$= 0 + 19{,}8 + 17$
$= 36{,}8$

Untuk Pi/Wi Max

$\sum Pi.Xi = (P_1.X_1) + (P_2.X_2) + (P_3.X_3)$
$= (27.\ 0) + (26.1) + (17.\frac{2}{3})$
$= 0 + 26 + 11{,}3$
$= 37{,}3$

Maka hasil dari ketiga tahapan tersebut adalah:

Tabel 4. Hasil Akhir Perhitungan Kriteria Greedy

Solusi ke-	(X_1,X_2,X_3)	$\sum Wi.Xi$	$\sum Pi.Xi$
Pi Max	$(1, \frac{5}{17}, 0)$	25	34,6
Wi Min	$(0, \frac{13}{17}, 1)$	25	36,8
$\frac{Pi}{Wi}$ Max	$(0,1, \frac{2}{3})$	25	37,3

Maka nilai profit terbesarnya adalah 37,3. Dengan demikian diperoleh pengaturan barang yang akan dimasukkan adalah:

X_1: 0
X_2: 1
X_3: $\frac{2}{3}$

Dengan artian bahwa:
Barang kesatu dengan nilai 0 maka barang tersebut tidak usah dimasukkan.
Barang kedua diambil 1 bagian yaitu 17Kg

Barang ketiga dengan nilai 2/3 maka barang tersebut

Diambil 2/3 bagian dengan artian: 2/3 X 12 = 8Kg

Jadi, total barang yang disimpan adalah 0+17+8=25Kg sesuai dengan berat maksimal tempat tersebut.

C. Algoritma Greedy

Teknik ini akan efektif jika objek disusun secara tidak naik (non increasing) berdasarkan nilai Pi/Wi.
Data yang diketahui:
n = 3, (1, 2, 3)
kapasitas tempat, M = 25
(W1, W2, W3) = (20, 17, 12)
(P1, P2, P3) = (27, 26, 17)
Nilai probabilitas $0 \le Xi \le 1$

perbandingan profit dengan bobot
$\frac{P_1}{W_1} : \frac{27}{20}$: 1,35 menjadi urutan ke 3
$\frac{P_2}{W_2} : \frac{26}{17}$: 1,5 menjadi urutan ke 1
$\frac{P_3}{W_3} : \frac{17}{12}$: 1,4 menjadi urutan ke 2

Susun data sesuai kriteria (*non increasing*), sehingga menghasilka pola yang baru
$(P_2, P_3, P_1) = (26, 17, 27)$
$(W_2, W_3, W_1) = (17, 12, 20)$

Masukkan nilai kriteria di atas ke dalam algoritma greedy

```
Algoritma GREEDY KNAPSACK.
PROCEDURE GREEDY
KNAPSACK(W,x,n)
float W[n], x[n], M, isi;
Int i, n;
x(1:1) ← 0; isi ← M ;
FOR i ← 1 TO n
{
IF W[i] > M ; EXIT ENDIF
x[i] ← 1
isi ← isi – W[i]
}
IF i ≤n ; x[i] ← isi / W[i] ENDIF
END_GREEDY KNAPSACK
```

Setelah itu masukkan data-data tersebut ke dalam algoritma diatas.
x(1:n) ← 0 ; isi ← 25
untuk i = 1
W(1) > isi ? → berat W1 adalah 17
17>25, kondisi salah karena 17 lebih kecil dari 25
Dengan demikian x(1) = 1 yang artinya barang tersebut dapat dimuat seluruhnya.
Isi = 25 -17= 8 kapasitas tempat menjadi berkurang karena sudah diiisi barang pertama, sisaya menjadi 8Kg lagi.

Untuk i=2
W(2) > isi ? → berat W2 adalah 12

12>8, kondisi benar karena 12 lebih besar dari 8

Dengan demikian $x(2) = \frac{8}{12}$

yang artinya barang tersebut dapat dimuat8/12 Bagian saja.

$\frac{8}{12}$ x 12kg(berat barang ke-2) = 8kg

Untuk i=3

Endif,

berakhir karena kapasitas tempat sudah maximal yaitu 25 kg, dengan berat barang ke-1 = 17Kg dan berat barang ke-2 adalah 8Kg.

Sedangkan profit nilai yang didapat adalah:

$$\sum Pi.Xi = (P_1.X_1) + (P_2.X_2) + (P_3.X_3)$$
$$= (26.\ 1) + (17.\ \tfrac{8}{12}) + (27.0)$$
$$= 26 + 11{,}3 + 0$$
$$= 37{,}3$$

Maka profit nilai yang didapat adalah 37,3

D. Perbandingan cara matematika, kriteria greedy dan algoritma greedy

Setelah contoh kasus yang diberikan selesai dihitung dengan cara matematika, kriteria greedy, algoritma greedy maka diperoleh hasil akhir yang sama akan tetapi dengan cara pengerjaan yang berbeda. Tiap cara memiliki kelebihan dan kekurangan masing-masing.

Untuk cara yang pertama yaitu cara matematika, kita harus memperhatikan nilai probabilitas dari setiap barang, karena nilai inilah sebagai penentunya dengan memperhatikan nilai probabilitas (Xi) yaitu 0≤Xi≤1. Disini nilai Xi kisarannya sangat banyak bisa 0, 0.1, 0.01, 0.001,... 1. Dengan demikian cara matematika ini ini sulit untuk menentukan yang paling optimal sebab kita harus mencari nilai probabilitas yang tersebar antara 0 dan 1, 0 ≤ Xi ≤ 1 untuk setiap objek. Cara ini disarankan tidak digunakan, karena dianggap lebih sulit dan lebih rumit dibanding kriteria greedy dan algoritma greedy.

Untuk cara yang kedua yaitu dengankriteria greedy.Dengan cara ini kita harus mencari terlebih dahulu greedy by profit berdasarkan profit yang paling besar, kemudian mencari greedy by weight yaitu berdasarkan berat yang paling kecil dan terakhir greedy by density yaitu perbandingan profit dengan berat yang nilainya paling besar. Cara ini dianggap lebih gampang dan lebih baik karena nilai yang dihasilkan lebih optimal, meskipun kita harus menyelesaikan beberapa tahapan terlebih dahulu.

Untuk cara yang ketiga yaitu dengan algoritma greedy. Dengan cara ini sebenarnya untuk perhitungan lebih mudah dan lebih cepat karena kita langsung mengaplikasikan perhitungannya dengan algoritma yang ada. Namun kita harus tau algoritmanya terlebih dahulu, tidak semua orang paham dan bisa menterjemahkan algoritma tersebut. Selain itu kelemahannya dengan algoritma greedy adalah teknik ini akan efektif jika objek disusun secara tidak naik (*non increasing*) berdasarkan nilai Pi/Wi.

V. KESIMPULAN

Berdasarkan perhitungan yang telah dilakukan yaitu untuk menyelesaikan permasalahan *knapsack*, maka dapat disimpulkan *Knapsack problem* dapat diselesaikan dengan cara matematika, kriteria greedy dan algoritma greedy. Tiap teknik mempunyai cara penyelesaian masing-masing. Cara matematika dianggap lebih rumit dan tidak cocok untuk digunakan. Cara kriteria greedy dianggap lebih mudah dan lebih optimal dibanding cara yang lain meskipun kekurangannya kita harus mengerjakan beberapa tahapan terlebih dahulu. Cara algoritma greedy lebih cepat penyelesaiannya namun kita harus tahu algoritma dan harus paham cara penterjemahan algoritma tersebut. Selain itu teknik ini akan efektif jika objek disusun secara tidak naik (*non increasing*) berdasarkan nilai Pi/Wi.

Sebagai saran penulis menyarankan untuk membuat perbandingan dengan penyelesaian dengan metode algoritma genetika, sehingga akan didapat penyelesaian yang paling optimal.

DAFTAR PUSTAKA

Dian dan Ade. 2013. *Implementasi Algoritma Greddy untuk Menyelesaikan Masalah Knapsack Problem*. Jurnal Ilmiah Saintikom. Jurusan Ilmu Komputer Universitas Sumatera Utara. Vol. 12, No. 3, September 2013.

Komang. 2010. *Implementasi Algoritma Genetika pada knapsack problem untuk optimasi pemilihan buah kemasan kotak*. Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010). Yogyakarta, 19 Juni 2010.

Knapsack problem dengan algoritma dan metode greedy by Hendry Prihandono https://hendryprihandono.wordpress.com/2009/01/03/knapsack-problem-dengan-algoritma-dan-metode-greedy/ diunduh (9 Oktober 2016)

Yulikuspartono. 2001. Pengantar Logika dan Algoritma. Yogyakarta: Andi.