TEKNIK MENENTUKAN KOMPOSISI BUAH PADA MASALAH PENGANGKUTAN DENGAN MENGGUNAKAN GREEDY KNAPSACK

Oleh: Faisal P., M.Kom
Jurusan Teknik Komputer Universitas Islam "45"
faisal_piliang@yahoo.co.id

Abstrak

Metode Greedy adalah salah satu cara atau teknik merancang suatu algoritma. Metode Greedy digunakan untuk mendapatkan solusi optimal dari suatu permasalahan. Salah satu permasalahan yang dapat diselesaikan dalam metode Greedy adalah masalah Knapsack atau ransel untuk tempat penampungan. Masalah Knapsack atau ransel adalah bagaimana memilih atau menentukan dari sekian banyak objek dari beberapa objek yang ada yang dapat dimuat ke dalam ransel sedemikian sehingga mendapatkan nilai kumulatif yang paling maksimum dan sesuai dengan nilai kapasitas maksimum ransel. Dalam penulisan ini diambil suatu kasus permasalahan mengenai pengiriman buah-buahan yang pada setiap jenisnya memiliki harga dan berat yang berbeda atau bervariasi yang dalam proses pengirimannya dibutuhkan sebuah alat pengangkutan berupa mobil truk Dalam hal ini, permasalahan yang timbul adalah bagaimana cara yang dipergunakan untuk dapat menentukan komposisi setiap jenis buah yang ada sesuai dengan nilai dan beratnya masing-masing dengan perbandingan dari nilai (profit) dengan beratnya yang terbesar, dan bagaimana dalam satu kali pengiriman dapat memuat buah secara optimal tanpa harus mengulangi pengangkutan kembali barang yang sama Dengan demikian apabila dalam satu kali pengiriman barang sudah dapat mengangkut secara optimal sesuai dengan kapasitas alat angkut, maka biaya dari pengangkutan barang tersebut bisa diminimalkan.

Kata kunci: *Algoritma, Metode Greedy, Knapsack Problem.*

PENDAHULUAN

Dalam kehidupan sehari-hari, kita sering dipusingkan dengan kapasitas media penyimpanan yang terbatas padahal kita diharuskan menyimpan beberapa objek kedalam media tersebut. Bagaimana kita mengatur objek apa saja yang dipilih dan seberapa besar objek tersebut disimpan? Dari permasalahan tersebut, munculan suatu permasalahan yang dikenal dengan "Permasalahan *Knapsack* atau lebih dikenal dengan *"Knapsack* Problem". 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).

Permasalahan ini dapat diselesaikan dengan 3 cara, yaitu:

1. Matematika
2. Kriteria Greedy dan
3. Algoritma Greedy.

Dalam kasus ini penulis mencoba menyelesaikan dengan 3 cara di atas. Metode Greedy merupakan salah satu cara untuk mendapatkan solusi optimal dalam proses penyimpanan. Pada metode ini untuk mendapatkan solusi optimal dari permasalahan yang mempunyai dua kriteria yaitu Fungsi Tujuan/Utama dan Nilai Pembatas (Constrain). Fungsi Tujuan hanya terdiri atas satu fungsi sedangkan Fungsi Pembatas dapat terdiri atas lebih dari satu fungsi.

PROSES KERJA METODE GREEDY

Menyelesaikan suatu masalah dengan beberapa fungsi pembatas untuk mencapai satu fungsi tujuan. Jadi dalam penyelesaiannya harus ditentukan mana sebagai fungsi pembatas dan mana sebagai fungsi tujuan. Cara menyelesaikan masalah *Knapsack* adalah:

1. Tentukan Fungsi Tujuan, yaitu mencari nilai maximum dari jumlah hasil perkalian antara nilai profit (Pi) dengan nilai probabilitas (Xi) Maximum XPi.Xi
2. Tentukan Fungsi Pembatas, yang merupakan hasil penjumlahan dari perkalian antara bobot (Wi) dengan nilai probabilitas (Xi) yang tidak boleh melebihi dari kapasitas media penyimpanan (M) £Wi.Xi<M, dimana 0<Xi<l, Pi>0, Wi>0

Dari kedua cara di atas berarti kita harus mengetahui:

1. Jumlah objek (n)
2. Bobot setiap objek (Wi)
3. Profit setiap objek (Pi)
4. Probabilitas setiap objek (Xi), dan
5. Kapasitas media penyimpanan (M)

Seperti penulis sudah sampaikan di atas bahwa permasalahan *Knapsack* ini bisa diselesaikan dengan 3 cara, yaitu:

1. Matematika
2. Kriteria Greedy dan
3. Algoritma Greedy.

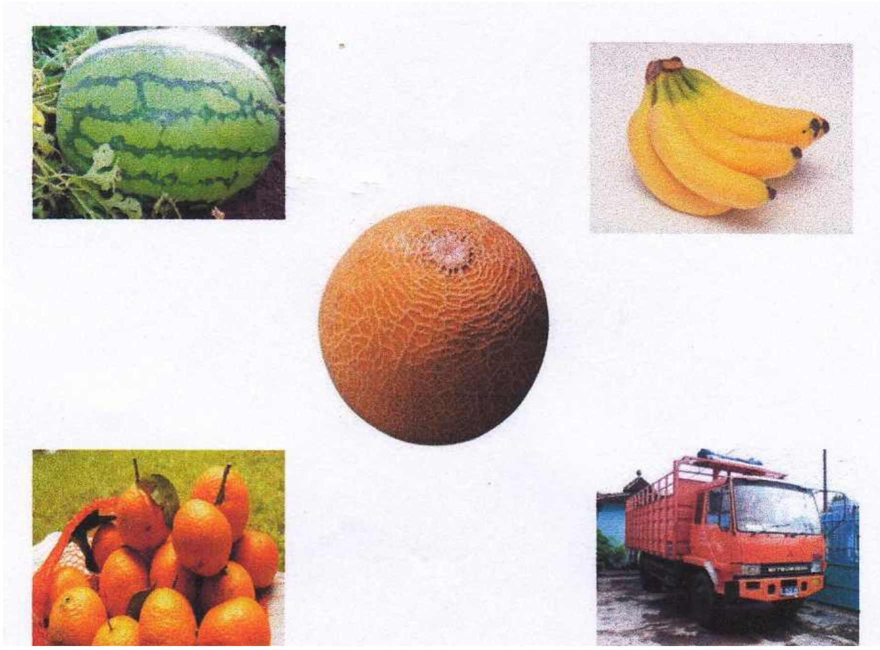
1. Cara Matematika, kita harus memperhatikan nilai probabilitas dari setiap barang, karena nilai inilah sebagai penentunya dengan memperhatikan nilai probabilitas (Xi) yaitu 0<Xi<l. Disini nilai Xi kisarannya sangat banyak bisa 0,0.1,0.01,0.001,1.

2. Kriteria greedy dengan memperhatikan:

   a. Pilih objek dengan nilai profit terbesar (Pi)
   b. Pilih objek dengan bobot terkecil (Wi)
   c. Pilih objek dengan nilai perbandingan profit dengan bobot yang terbesar (Pi/Wi)
3. Algortima greedy, yaitu
   PROCEDURE GREEDY *KNAPSACK* (P, W, X, n) REAL P(l :n), W(l
   :n), X(l :n), M, isi INTEGER i, n X(l:n) = 0 isi = M
   FOR i = 1 TO n DO
   IF W(i) > isi THEN EXIT ENDIF
   X(i)=l
   isi = isi - W(i)
   REPEAT
   IF i < n THEN X(i) = isiAV(i) ENDIF
   END GREEDY *KNAPSACK* Teknik yang ke-3 ini akan efektif jika objekdisusun secara tidak naik (non increasing) berdasarkan nilai Pi/Wi.

KNAPSACK PROBLEM

Diketahui 4 jenis buah-buahan yang akan dikirimkan dengan menggunakan mobil truk dengan kapasitas maksimal sebesar 15 ton. Berat masing-masing buah yang akan dikirimkan adalah 6 ton, 5 ton, 10 ton dan 4 ton dimana setiap buah memiliki profit sebesar masing-masing 12, 15, 50
dan 8. Bagaimana caranya untuk menentukan dalam satu kali pengiriman dapat memuat buah-buah mana saja yang akan dibawa secara optimal tanpa harus mengulangi pengangkutan kembali barang yang sama.



PEMBAHASAN MASALAH

1. Cara Matematika

Objek (n) = (1,2, 3, 4) Kapasitas (M) =15

(Wl, W2, W3, W4) = (6, 5,10,4)

(Pl, P2, P3,W4) = (12,15, 50,10)

Nilai probabilitas 0 < Xi < 1

Solusi ke Nilai Probabilitas Fungsi Pembatas dan Fungsi Tujuan yaitu:

No	(X1, X2, X3, X4)	(W1. X1) +	(W2. X2) +	(W3. X3) +	(W4. X4)	∑ Wi.Xi ≤ M	(P1. X1) +	(P2. X2) +	(P3. X3) +	(P4. X4	∑ Pi.Xi (Max)
1	(1; 1, 2/5, 0)	(6*1)	(5*1)	(10*(2/5))	(4*0)	15	(12*1)	(15*1)	(50*(2/5))	(10*0)	47
2	(1, 1, 0, 1)	(6*1)	(5*1)	(10*0)	(4*1)	15	(12*1)	(15*1)	(50*0)	(10*1)	37
3	(0, 1, 1, 0)	(6*0)	(5*1)	(10*1)	(4*0)	15	(12*0)	(15*1)	(50*1)	(10*0)	65
4	(5/6, 0, 1, 0)	(6*(5/6))	(5*1)	(10*1)	(4*0)	15	(12*(5/6))	(15*0)	(50*1)	(10*0)	60
5	(0, 1, 3/5, 1)	(6*0)	(5*1)	(10*(3/5))	(4*1)	15	(12*0)	(15*1)	(50*(3/5))	(10*1)	55
	...	...	...								

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. Cara ini disarankan tidak digunakan.

2. Cara Kriteria Greedy

Objek (n) = (1,2, 3,4) Kapasitas (M) =15 (Wl, W2, W3, W4) = (6, 5,10, 4) (Pl, P2, P3,W4) = (12, 15, 50, 10) Nilai probabilitas 0 < Xi < 1

Kriteria Greedy:

a. Pilih objek dengan nilai profit terbesar (Pi) Susun data
sesuai kriteria:
(Pl, P2, P3,W4) = (50,15, 12,10)
(Wl, W2, W3, W4) = (10, 5, 6, 4)
Solusi ke Nilai Probabilitas Fungsi Pembatas dan Fungsi Tujuan yaitu:

No	(X1, X2, X3, X4)	(W1. X1) +	(W2. X2) +	(W3. X3) +	(W4. X4)	∑ Wi.Xi ≤ M	(P1. X1) +	(P2. X2) +	(P3. X3) +	(P4. X4)	∑ Pi.Xi (Max)
1	(1, 1, 0, 0)	(10*1)	(5*1)	(6*0)	(4*0)	15	(50*1)	(15*1)	(12*0)	(10*0)	65

b. Pilih objek dengan bobot terkecil (Wi) Susun data
sesuai kriteria:
(P4, P3, P2, Pl) = (10, 12,15, 50)
(W4, W3, W2, Wl) = (4, 5, 6, 10)
Solusi ke Nilai Probabilitas Fungsi Pembatas dan Fungsi Tujuan yaitu:

No	(X4, X3, X2, X1)	(W4. X4) +	(W3. X3) +	(W2. X1) +	(W1. X1)	∑ Wi.Xi ≤ M	(P4. X4) +	(P3. X3) +	(P2. X2) +	(P1. X1)	∑ Pi.Xi (Max)
1	(1, 1, 1, 0)	(4*1)	(5*1)	(6*1)	(10*0)	15	(10*1)	(12*1)	(15*1)	(50*0)	52

c. Pilih objek dengan nilai perbandingan profit dengan bobot yang terbesar (Pi/Wi)
Data yang diketahui:
Objek (n) = (1,2, 3,4)
Kapasitas (M) = 15
(Wl, W2, W3, W4) = (6, 5, 10,4)
(Pl, P2, P3,W4) = (12, 15, 50, 10)
perbandingan profit dengan bobot
P1/W1 = 12/6 = 2
P2/W2= 15/5 = 3
P3/W3 = 50/10 = 5
P4/W4= 10/4 -2,5
Susun data sesuai kriteria, secara tidak naik *(non increasing):*
(P3, P2, P4, Pl) = (50, 15,10,12)
(W3, W2, W4, Wl) = (10, 5, 4, 6)
Solusi ke Nilai Probabilitas Fungsi Pembatas Fungsi Tujuan:

No	(X3, X2, X4, X1)	(W3. X3) +	(W2. X2) +	(W4. X4) +	(W1. X1)	∑ Wi.Xi ≤ M	(P3. X3) +	(P2. X2) +	(P4. X4) +	(P1. X1)	∑ Pi.Xi (Max)
1	(1, 1, 0, 0)	(10*1)	(5*1)	(4*0)	(6*0)	15	(50*1)	(15*1)	(10*0)	(12*0)	65

Dari 3 kriteria di atas dapat disimpulkan bahwa fimgsi tujuan yang bernilai maximum adalah 65 dengan fungsi pembatasnya adalah 15 dan nilai probabilitasnya adalah (X3, X2, X4, Xl) : (1, 1, 0, 0), jadi disini yang memeberikan hasil optimal pada kriteria yang ke-3 yaitu Pilih objek dengan nilai perbandingan profit dengan bobot yang terbesar (Pi/Wi) .

3. Cara algoritma greedy

Teknik ini akan efektif jika objek disusrm secara tidak naik *(non increasing)*
berdasarkan nilai Pi/Wi.
Data yang diketahui:

Objek (n) : (1, 2,3,4)
Kapasitas (M): 15
(Wl, W} W3, W4) : (6,5,10, 4)
(P1, P2, P3,\ry4) : (12,15, 50, l0)
Nilai probabilitas 0 <Xi < I
Perbandingan profit dengan bobot:
Pl/ Wl : 1216:2
PA W2: 15/5 :3
P3/ W3 : 50/10:5
P4lW4: l0l4 :2,5
Susun data sesuai kriteria (non increasing):
(P3,P2, P4, Pl) : (50, 15,12,10) atau
(Pl,P2, P3, P4) : (50, 15,12,70)
(W3, W2, W4, Wl): (10,5,4,6) atau
(W1, W2" W3, W4) : (10, 5,4,6)

Masukkan nilai kdteria di atas ke dalam algoritma greedy
l. nama prosedur/proses PROCEDURE GREEDY KNAPSACK (P, W, X, n)
2. variabel yang digunakan REAL P(l:n), W(1:n), X(l:n), M, isi
3. vmiabel yang digunakan INTEGER i, n
4. X(l:n):0
5. isi:M
6.FOR I = l TO n DO
7. IF w(i) > isi THEN EXIT ENDIF
8. X(i): 1
9. isi = isi -W(i)
10. REPEAT
11. IF i < n THEN X(i) : isi/W(i) ENDIF
12. akhir prosedur/proses END GREEDY KNAPSACK
Proses kegiatan dimulai dari langkah ke- 4 sampai dengan 11.
X(l:4) = 0, artinya X(1):0, X(2H, X(3)=0, X(4):0
isi = M = 20
Pengulangan untuk i = 1 sampai dengan 4 Untuk i = 1
Apakah W(l) > isi
Apakah 10 > 15, jawabnya tidak, karena tidak maka perintah dibawah IF dikerjakan.
nilai probabilitas untuk objek pada urutan pertama (XI) X(l) = l isi -15 -
10 = 5

mengulang untuk perulangan FOR REPEAT Untuki
= 2 Apakah W(2) > isi
Apakah 5 > 5, jawabnya tidak, karena tidak maka perintah dibawah IF dikerjakan.
nilai probabilitas untuk objek pada urutan pertama (X2)
X(2)=l
isi = 5 - 5 - 0
mengulang untuk perulangan FOR REPEAT Untuk i
= 3 Apakah W(3) > isi
Apakah 0 > 0, jawabnya ya, karena ya maka perintah EXIT dikerjakan, yaitu
keluar dari pengulangan/FOR dan mengerjakan perintah di bawah REPEAT.
nilai probabilitas untuk objek pada urutan kedua (X3).
Apakah 3 < 4, jawabnya ya, karena ya maka X(3) = 0/0 = 0.
nilai probabilitas untuk objek pada urutan kedua (X4).
Apakah 4 < 4, jawabnya ya, karena ya maka X(4) = 0/0 = 0.
Selesai (akhir dari prosedur greedy Knapsack).
Berarti untuk nilai X(3) = 0 dan X(4) = 0, sebab nilai probabilitas untuk objek ke-3 dan ke-4 tidak pernah dicari.

Jadi susunan:
(P3, P2, P4, PI) - (50,15,12,10) atau (PI, P2, P3, P4) = (50,15,12,10)
(W3, W2, W4, Wl) = (10, 5,4, 6) atau (Wl, W2, W3, W4) = (10, 5, 4, 6)

No	(X1, X2, X3, X4)	(W1. X1) +	(W2. X2) +	(W3. X3) +	(W4. X4)	∑ Wi.Xi ≤ M	(P1. X1) +	(P2. X2) +	(P3. X3) +	(P4. X4)	∑ Pi.Xi (Max)
1	(1, 1, 0, 0)	(10*1)	(5*1)	(6*0)	(4*0)	15	(50*1)	(15*1)	(12*0)	(10*0)	65

DAFTAR PUSTAKA

Ellis, Horrowitz & Sahni, Sartaj, "Fundamental of computer algorithmic",
Pitman publis. Limited, London, 1988. Gary Kroehnert, 1993, 100 Training games, National Library of Australia, Australia.
HS, Suryadi, Teori Graph Dasar, 1994, Gunadarma, Jakarta.
HS, Suryadi, Pengantar Algoritma dan Pemrograman, 1991, Gunadarma,
Jakarta.
MT, Suryadi, Pengantar Analisa Algoritma, 1992, Gunadarma, Jakarta. Munir, Rinaldi, 2002, Logika dan Algoritma Buku I, Edisi keempat, Informatika, Bandung.
Munir, Rinaldi, 2005, Logika dan Algoritma Buku II, Edisi ketiga, Informatika, Bandung.
Munir, Rinaldi, Strategi AJgoritmik,Bandung, 2007.
Binanto Iwan, 2003, Pemrograman C++ di Linux, Andi Offset, Yogyakarta
Frieyadie, 2006, Pemrograman C++, Elex Media, Jakarta.