Penerapan Algoritma *Greedy* dalam Penyetokan Barang

Christian Angga - 13508008
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
ca_ers@hotmail.com

***Abstrak*—Penyetokan barang di gudang merupakan salah satu masalah khusus bagi badan usaha atau perusahaan yang bergerak di bidang perdagangan. Setiap gudang memiliki sistem dan faktor pertimbangan tersendiri untuk menentukan barang-barang apa saja yang ingin di stok. Setiap faktor tersebut pastinya mengacu pada sumber daya yang terbatas yang menyebabkan perusahaan harus menentukan barang apa saja yang ingin di stok agar *profit* yang dihasilkan maksimal dengan sumber daya yang terbatas tersebut.**

Dalam penentuan barang stok ini, kita dapat menggunakan algoritma *greedy* untuk melakukan optimasi terhadap faktor-faktor yang berhubungan langsung dengan proses penyetokan ini, diantara lain harga beli, *profit,* berat, volume, dan daya beli *customer.* Tujuan akhir penerapan algoritma *greedy* ini adalah perusahaan mendapatkan solusi yang cukup optimal dalam melakukan penyetokan barang meskipun modal terbatas, luas gudang terbatas, dan lainnya.

***Kata Kunci*—algoritma *greedy,* berat, daya beli *customer*, harga beli, *profit,* stok barang, volume.**

I. PENDAHULUAN

Pada bisnis perdagangan, diperlukan banyak strategi bisnis yang digunakan. Strategi utama dalam bisnis perdagangan adalah penyetokan barang dan penjualan barang. Strategi penjualan barang berfungsi untuk menjual sebanyak-banyaknya stok barang yang berada di gudang, sedangkan strategi penyetokan barang berfungsi untuk mensinkronkan stok barang yang berada di gudang dengan jumlah penjualan yang terjadi dengan didukung faktor-faktor yang berada di sekitarnya, seperti besar gudang, besar modal, daya beli *customer*, dan sebagainya. Dalam hal ini, algoritma *greedy* lebih berfungsi secara optimal jika diimplementasikan untuk membantu berjalannya strategi penyetokan barang.

II. TEORI DASAR

II.1. Penyetokan Barang

Penyetokan barang di gudang merupakan salah satu kunci utama dalam berbisnis di bidang perdagangan, karena di dalam perdagangan, daya jual setiap barang tidaklah sama maka diperlukan trik-trik dan perhitungan yang tepat dalam memilih barang yang akan di stok. Selain itu, terdapat faktor-faktor pembatas lainnya seperti modal, berat barang, *profit*, volume, daya beli *customer*, dan lainnya yang membuat diperlukannya banyak perhitungan dalam melakukan penyetokan barang tersebut demi mendapatkan *profit* yang maksimal.

A. Modal

Harga beli suatu barang biasa berkaitan erat dengan modal suatu perusahaan. Modal merupakan faktor pembatas utama dalam melakukan penyetokan barang, hal ini dikarenakan modal merupakan sumber daya yang terbatas.

B. Berat Barang

Berat suatu barang biasa berkaitan dengan *cost* atau ongkos kirim barang dalam melakukan penyetokan barang. Makin berat barang tersebut, makin mahal juga ongkos yang diperlukan dalam melakukan pengiriman barang tersebut. Hal ini tentu saja mempengaruhi harga beli dari barang yang akan di stok tersebut.

C. *Profit*

Profit dalam hal ini adalah keuntungan yang didapat perusahaan dari penjualan suatu barang jikalau barang tersebut berhasil terjual. *Profit* suatu barang dalam perdagangan tidaklah selalu sama persentase atau besarnya, ini semua tergantung dengan barang apa yang dijual, jenis barang yang dijual, dan sebagainya.

D. Volume

Volume suatu barang biasanya berkaitan erat dengan kemampuan gudang untuk menyimpan barang dengan kapasitas tertentu sebanyak-banyaknya. Selain mempengaruhi kapasitas gudang, terkadang volume barang pun mempengaruhi *cost* atau ongkos kirim barang tersebut juga, yang mempengaruhi harga barang itu sendiri.

E. Daya Beli *Customer*

Daya beli *customer* akan suatu barang biasanya dipengaruhi oleh banyaknya penjual yang menawarkan barang tersebut dibandingkan dengan banyaknya jumlah pembeli yang ingin membeli barang tersebut. Daya beli *customer* akan suatu barang berbeda-beda dan tidaklah selalu sama setiap waktunya, serta tidak memiliki parameter khusus yang merepresentasikan konstanta akan daya beli *customer* ini, maka disini daya beli *customer* akan diintegrasikan dengan sejumlah bilangan integer yang berupa indeks yang akan dibahas kemudian.

II.2. Teori Algoritma *Greedy*

Algoritma *greedy* merupakan metode yang paling populer untuk memecahkan masalah optimasi. Secara harfiah, *greedy* berarti tamak atau rakus. Prinsip dari algoritma *greedy* adalah mengambil setiap kesempatan yang ada saat itu juga, tanpa memperhatikan konsekuensi kedepannya. Algoritma *greedy* membentuk solusi dari langkah demi langkah, dan pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Di setiap langkahnya algoritma *greedy* mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi kedepan. Setiap keputusan yang diambil diharapkan merupakan langkah optimum pada langkah tersebut, dikenali sebagai solusi optimum lokal, kemudian dengan setiap langkah yang ditempuh diharapkan dapat memperoleh solusi optimum di akhir proses, yaitu solusi optimum global.

Algoritma *greedy* tidak selalu memperoleh solusi optimum untuk keseluruhan permasalahan yang ditangani, dikarenakan algoritma *greedy* tidak melakukan operasi secara exhaustive kepada semua data, dan seringkali tidak mementingkan solusi optimum global. Akan tetapi, algoritma *greedy* merupakan solusi yang baik digunakan dikarenakan algoritma *greedy* bekerja dengan cepat dan sering memberikan perkiraan nilai optimum yang baik di setiap langkahnya. Dan tidak jarang dapat menghasilkan solusi optimum global pada suatu permasalahan dari pengambilan solusi optimum lokal di setiap langkahnya.

Elemen-elemen algoritma *greedy* dalam persoalan optimasi adalah sebagai berikut :

1. Himpunan kandidat
   Himpunan ini berisi elemen-elemen pembentuk solusi.
2. Himpunan solusi
   Himpunan ini berisi kandidat-kandidat yang terpilih sebagai solusi persoalan. Himpunan solusi merupakan himpunan bagian dari himpunan kandidat.
3. Fungsi seleksi
   Fungsi seleksi dinyatakan sebagai predikat SELEKSI merupakan fungsi yang pada setiap langkah memilih kandidat yang paling mungkin untuk mendapatkan solusi optimal.
4. Fungsi kelayakan
   Fungsi kelayakan dinyatakan sebagai predikat LAYAK merupakan fungsi yang memeriksa apakah suatu kandidat yang dipilih dapat memberikan solusi yang layak, yaitu kandidat tersebut tidak melanggar aturan yang ada.
5. Fungsi objektif
   Fungsi objektif merupakan fungsi yang memaksimumkan atau meminimumkan nilai solusi.

III. PENERAPAN ALGORITMA *GREEDY* DALAM PENYETOKAN BARANG

III.1. *Greedy* yang melibatkan 1 buah atribut saja

Algoritma *greedy* dalam penyetokan barang ini dapat diterapkan dalam mencapai solusi penyetokan barang apa saja yang dianggap paling optimal sesuai dengan prioritas faktor yang diinginkan. Pada penerapan algoritma *greedy* kali ini, terdapat contoh kasus yang memuat 20 buah data barang yang masing-masing memiliki 5 faktor yang sudah terukur untuk digunakan dalam penentuan barang-barang mana saja yang patut dipilih dalam penyetokan barang dimana pada kasus ini memuat persoalan *knapsack* dengan bobot modal = 200.000

no	modal	profit	berat	volume	daya jual
1	20000	5000	70	600	2
2	90000	30000	80	800	2
3	30000	10000	50	600	6
4	10000	5000	10	800	6
5	20000	2000	40	300	5
6	40000	30000	70	500	2
7	80000	20000	10	600	6
8	40000	10000	40	800	9
9	60000	5000	20	600	5
10	10000	5000	50	500	9
11	30000	3000	50	800	5
12	30000	15000	10	300	2
13	60000	15000	20	300	9
14	90000	20000	20	200	9
15	70000	30000	40	500	5
16	20000	10000	70	200	6
17	10000	5000	10	200	2
18	50000	25000	80	300	6
19	80000	20000	50	200	5
20	60000	6000	80	800	2

Dari contoh kasus diatas, bisa dilakukan perhitungan dengan algoritma *greedy* berdasarkan banyak faktor. Beberapa strategi *greedy* yang dapat digunakan untuk memilih objek/barang yang akan dimasukkan ke *knapsack* adalah:

A. *Greedy by* Modal

Pada setiap langkah, *knapsack* diisi dengan objek yang mempunyai harga beli atau modal terkecil. Strategi ini mencoba memaksimumkan jumlah barang yang dibeli dengan cara memilih objek-objek yang paling murah terlebih dahulu.

no	modal	profit	berat	volume	daya jual	modal
1	20000	5000	70	600	2	20000
2	90000	30000	80	800	2	
3	30000	10000	50	600	6	30000
4	10000	5000	10	800	6	10000
5	20000	2000	40	300	5	20000
6	40000	30000	70	500	2	
7	80000	20000	10	600	6	
8	40000	10000	40	800	9	
9	60000	5000	20	600	5	
10	10000	5000	50	500	9	10000
11	30000	3000	50	800	5	30000
12	30000	15000	10	300	2	30000
13	60000	15000	20	300	9	
14	90000	20000	20	200	9	
15	70000	30000	40	500	5	
16	20000	10000	70	200	6	20000
17	10000	5000	10	200	2	10000
18	50000	25000	80	300	6	
19	80000	20000	50	200	5	
20	60000	6000	80	800	2	
				total harga		180000

B. *Greedy by Profit*

Pada setiap langkah, *knapsack* diisi dengan objek yang mempunyai keuntungan terbesar (bukan persen keuntungan terbesar). Strategi ini mencoba memaksimumkan jumlah *profit* yang didapat jikalau barang tersebut terjual habis dengan cara memilih objek-objek yang memiliki nilai *profit* terbesar terlebih dahulu.

no	modal	profit	berat	volume	daya jual	profit
1	20000	5000	70	600	2	
2	90000	30000	80	800	2	90000
3	30000	10000	50	600	6	
4	10000	5000	10	800	6	
5	20000	2000	40	300	5	
6	40000	30000	70	500	2	40000
7	80000	20000	10	600	6	
8	40000	10000	40	800	9	
9	60000	5000	20	600	5	
10	10000	5000	50	500	9	
11	30000	3000	50	800	5	
12	30000	15000	10	300	2	
13	60000	15000	20	300	9	
14	90000	20000	20	200	9	
15	70000	30000	40	500	5	70000
16	20000	10000	70	200	6	
17	10000	5000	10	200	2	
18	50000	25000	80	300	6	
19	80000	20000	50	200	5	
20	60000	6000	80	800	2	
				total harga		200000

C. *Greedy by* Berat

Pada setiap langkah, *knapsack* diisi dengan objek yang mempunyai berat teringan. Strategi ini mencoba meminimumkan berat objek agar *cost* dalam pengiriman barang pun ikut minimum dengan cara memilih objek-objek yang memiliki nilai berat terkecil terlebih dahulu.

no	modal	profit	berat	volume	daya jual	berat
1	20000	5000	70	600	2	
2	90000	30000	80	800	2	
3	30000	10000	50	600	6	
4	10000	5000	10	800	6	10000
5	20000	2000	40	300	5	
6	40000	30000	70	500	2	
7	80000	20000	10	600	6	80000
8	40000	10000	40	800	9	
9	60000	5000	20	600	5	60000
10	10000	5000	50	500	9	10000
11	30000	3000	50	800	5	
12	30000	15000	10	300	2	30000
13	60000	15000	20	300	9	
14	90000	20000	20	200	9	
15	70000	30000	40	500	5	
16	20000	10000	70	200	6	
17	10000	5000	10	200	2	10000
18	50000	25000	80	300	6	
19	80000	20000	50	200	5	
20	60000	6000	80	800	2	
				total harga		200000

D. *Greedy by* Volume

Pada setiap langkah, *knapsack* diisi dengan objek yang mempunyai volume terkecil. Strategi ini mencoba

meminimumkan volume objek agar kapasistas gudang menjadi maksimal serta *cost* dalam pengiriman barang pun ikut minimum dengan cara memilih objek-objek yang memiliki nilai volume terkecil terlebih dahulu.

no	modal	profit	berat	volume	daya jual	volume
1	20000	5000	70	600	2	
2	90000	30000	80	800	2	
3	30000	10000	50	600	6	
4	10000	5000	10	800	6	
5	20000	2000	40	300	5	20000
6	40000	30000	70	500	2	
7	80000	20000	10	600	6	
8	40000	10000	40	800	9	
9	60000	5000	20	600	5	
10	10000	5000	50	500	9	
11	30000	3000	50	800	5	
12	30000	15000	10	300	2	
13	60000	15000	20	300	9	
14	90000	20000	20	200	9	90000
15	70000	30000	40	500	5	
16	20000	10000	70	200	6	
17	10000	5000	10	200	2	10000
18	50000	25000	80	300	6	
19	80000	20000	50	200	5	80000
20	60000	6000	80	800	2	
				total harga		200000

E. *Greedy by* Daya Jual

Pada setiap langkah, *knapsack* diisi dengan objek yang mempunyai daya jual tertinggi. Strategi ini mencoba memaksimalkan penjualan dengan cara memilih objek-objek yang memiliki nilai daya jual terbesar terlebih dahulu.

no	modal	profit	berat	volume	daya jual	daya jual
1	20000	5000	70	600	2	
2	90000	30000	80	800	2	
3	30000	10000	50	600	6	
4	10000	5000	10	800	6	
5	20000	2000	40	300	5	
6	40000	30000	70	500	2	
7	80000	20000	10	600	6	
8	40000	10000	40	800	9	40000
9	60000	5000	20	600	5	
10	10000	5000	50	500	9	10000
11	30000	3000	50	800	5	
12	30000	15000	10	300	2	
13	60000	15000	20	300	9	60000
14	90000	20000	20	200	9	90000
15	70000	30000	40	500	5	
16	20000	10000	70	200	6	
17	10000	5000	10	200	2	
18	50000	25000	80	300	6	
19	80000	20000	50	200	5	
20	60000	6000	80	800	2	
				total harga		200000

III.2. *Greedy* yang melibatkan 2 buah atribut

Pada dasarnya, *greedy* yang melibatkan 1 buah atribut saja tidaklah cukup, ini dikarenakan terlalu banyak aspek yang perlu dikelola agar hasil yang didapat lebih optimal. Di dalam dunia bisnis, aspek utama yang selalu menjadi prioritas utama adalah *profit*. Karena demikian, maka atribut-atribut yang ada akan dibandingkan dengan *profit* agar menghasilkan sesuatu yang optimal dengan *profit* yang maksimal.

A. *Greedy by* Modal dan *Profit*

Sebelum melakukan penentuan akan objek yang akan dimasukkan ke dalam *knapsack*, dilakukan perbandingan terlebih dahulu antara modal dan *profit* objek tersebut. Setelah mendapatkan perbandingannya, pada setiap langkah *knapsack* diisi dengan objek yang mempunyai perbandingan modal dengan *profit* terbesar. Strategi ini mencoba memaksimumkan jumlah barang yang dibeli serta memaksimumkan *profit* yang didapat dengan cara memilih objek-objek yang paling murah dengan persentase *profit* yang tinggi terlebih dahulu.

no	modal	profit	berat	volume	daya jual	% modal (dg profit)	modal
1	20000	5000	70	600	2	0.25	
2	90000	30000	80	800	2	0.3333333	
3	30000	10000	50	600	6	0.3333333	30000
4	10000	5000	10	800	6	0.5	10000
5	20000	2000	40	300	5	0.1	
6	40000	30000	70	500	2	0.75	40000
7	80000	20000	10	600	6	0.25	
8	40000	10000	40	800	9	0.25	
9	60000	5000	20	600	5	0.0833333	
10	10000	5000	50	500	9	0.5	10000
11	30000	3000	50	800	5	0.1	
12	30000	15000	10	300	2	0.5	30000
13	60000	15000	20	300	9	0.25	
14	90000	20000	20	200	9	0.2222222	
15	70000	30000	40	500	5	0.4285714	
16	20000	10000	70	200	6	0.5	20000
17	10000	5000	10	200	2	0.5	10000
18	50000	25000	80	300	6	0.5	50000
19	80000	20000	50	200	5	0.25	
20	60000	6000	80	800	2	0.1	
				total harga			200000

B. *Greedy by* Berat dan *Profit*

Sebelum melakukan penentuan akan objek yang akan dimasukkan ke dalam *knapsack*, dilakukan perbandingan terlebih dahulu antara berat dan *profit* objek tersebut. Setelah mendapatkan perbandingannya, pada setiap langkah *knapsack* diisi dengan objek yang mempunyai perbandingan berat dengan *profit* terkecil. Strategi ini mencoba meminimumkan berat barang yang dibeli serta memaksimumkan *profit* yang didapat dengan cara memilih objek-objek yang paling ringan dengan persentase *profit* yang tinggi terlebih dahulu.

no	modal	profit	berat	volume	daya jual	% berat (dg profit)	berat
1	20000	5000	70	600	2	0.014	
2	90000	30000	80	800	2	0.0026667	
3	30000	10000	50	600	6	0.005	
4	10000	5000	10	800	6	0.002	
5	20000	2000	40	300	5	0.02	
6	40000	30000	70	500	2	0.0023333	
7	80000	20000	10	600	6	0.0005	80000
8	40000	10000	40	800	9	0.004	
9	60000	5000	20	600	5	0.004	
10	10000	5000	50	500	9	0.01	
11	30000	3000	50	800	5	0.0166667	
12	30000	15000	10	300	2	0.0006667	30000
13	60000	15000	20	300	9	0.0013333	
14	90000	20000	20	200	9	0.001	90000
15	70000	30000	40	500	5	0.0013333	
16	20000	10000	70	200	6	0.007	
17	10000	5000	10	200	2	0.002	
18	50000	25000	80	300	6	0.0032	
19	80000	20000	50	200	5	0.0025	
20	60000	6000	80	800	2	0.0133333	
				total harga			200000

C. *Greedy by* Volume dan *Profit*

Sebelum melakukan penentuan akan objek yang akan dimasukkan ke dalam *knapsack*, dilakukan perbandingan terlebih dahulu antara volume dan *profit* objek tersebut. Setelah mendapatkan perbandingannya, pada setiap langkah *knapsack* diisi dengan objek yang mempunyai perbandingan volume dengan *profit* terkecil. Strategi ini mencoba meminimumkan volume barang yang dibeli serta memaksimumkan *profit* yang didapat dengan cara memilih objek-objek bervolume paling kecil dengan persentase *profit* yang tinggi terlebih dahulu.

no	modal	profit	berat	volume	daya jual	% volume (dg profit)	volume
1	20000	5000	70	600	2	0.12	
2	90000	30000	80	800	2	0.0266667	
3	30000	10000	50	600	6	0.06	
4	10000	5000	10	800	6	0.16	
5	20000	2000	40	300	5	0.15	
6	40000	30000	70	500	2	0.0166667	
7	80000	20000	10	600	6	0.03	
8	40000	10000	40	800	9	0.08	
9	60000	5000	20	600	5	0.12	
10	10000	5000	50	500	9	0.1	
11	30000	3000	50	800	5	0.2666667	
12	30000	15000	10	300	2	0.02	30000
13	60000	15000	20	300	9	0.02	
14	90000	20000	20	200	9	0.01	90000
15	70000	30000	40	500	5	0.0166667	
16	20000	10000	70	200	6	0.02	
17	10000	5000	10	200	2	0.04	
18	50000	25000	80	300	6	0.012	
19	80000	20000	50	200	5	0.01	80000
20	60000	6000	80	800	2	0.1333333	
				total harga			200000

IV. KESIMPULAN

Algoritma *greedy* adalah algoritma yang sederhana, cepat, dan tergolong praktis. Penerapan algoritma *greedy* dalam penyetokan barang memberikan hasil yang sangat variatif, tergantung dengan apa yang menjadi faktor penentu dalam menggunakan algoritma *greedy* tersebut. Tidak ada satupun hasil dari penerapan algoritma *greedy* ini yang dianggap paling benar atau dianggap paling salah, karena semua hasil tersebut tergantung dengan apa yang diprioritaskan oleh si penyetok barang.

REFERENSI

Munir, Rinaldi. Diktat Kuliah Strategi Algoritmik. 2007. Bandung : Program Studi Teknik Informatika ITB.

Wikipedia
http://en.wikipedia.org/wiki/Greedy_algorithm
Waktu akses : 26 November 2010, pukul 00.00 – 02.00

Wikibooks
http://en.wikibooks.org/wiki/Algorithms/Greedy_Algorithms
Waktu akses : 26 November 2010, pukul 00.00 – 02.00

Blogspot
http://delly-kapila.blogspot.com/2009/01/implementasi-rencana-pengadaan-barang.html
Waktu akses : 26 November 2010, pukul 00.00 – 02.00

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 26 November 2010

Christian Angga
13508008