

Optimisasi Masalah Knapsack (0-1) Menggunakan Algoritma Greedy by Density

Fujita Rahmah 120140070

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Sumatera, Jalan Terusan Ryacudu
Fujita.120140070@student.itera.ac.id

Abstract— Persoalan knapsack dapat diartikan sebagai karung atau kantong yang digunakan untuk menyimpan sesuatu. Tentu tidak semua objek dapat ditampung didalam knapsack ini. Karung tersebut hanya dapat menyimpan beberapa objek dengan bobot yang sama atau lebih kecil dari bobot maksimum kantong. Permasalahan ini yang membutuhkan penyelesaian solusi yang optimal dapat diselesaikan dengan menggunakan salah satu strategi yakni strategi Algoritma Greedy dikerucutkan kembali dengan Teknik heuristic yang menghasilkan sub-strategi yakni greedy by profit, greedy by weight dan greedy by density. Biasanya greedy by density memiliki derajat sedikit lebih tinggi dalam hal mendapatkan solusi optimalnya. Permasalahan lain ialah strategy by density ini belum tentu menghasilkan solusi optimum dalam persoalan tertentu. Makalah ini bertujuan Untuk mengetahui tahapan penyelesaian masalah knapsack 0-1 dengan menggunakan algoritma greedy by density, mengetahui suatu permasalahan berhasil mendapatkan solusi optimal atau tidak, mengetahui tahap lanjutan yang dapat digunakan untuk mencapai solusi optimal yakni dengan mengubah nilai densitas terbesar kemudian mencari kembali solusi optimum lain dan dibandingkan dengan solusi sebelumnya jika sudah memenuhi syarat maka proses berhenti. Diakhir hasil perhitungan dapat disimpulkan total berat dan total keuntungan menggunakan algoritma greedy by density yang telah diubah solusi paling optimal ialah dengan total bobot 100 dan total keuntungan 55.

Keywords— *Knapsack, Algoritma Greedy, Heuristic, solusi optimal,*

I. PENDAHULUAN

Knapsack dapat diartikan suatu kantong atau tempat yang digunakan untuk menampung sesuatu objek. Karena kantong memiliki maksimum kapasitas maka tidak semua barang dapat dimasukkan. Hanya beberapa barang yang memiliki berat yang sama atau lebih kecil dengan ukuran

kapasitas kantong. objek-objek yang ada tidak harus dimuat secara keseluruhan atau semua objek dimasukkan kedalam kantong namun dapat dimasukkan sebagian objek saja dengan cara dipilih. Untuk kegiatan pemilihan barang memiliki banyak tahap yang diperlukan untuk mendapatkan penyelesaian masalah tersebut. Tolak ukur permasalahan bukan hanya dari hasil yang optimal saja tetapi penilaian dinilai dari setiap tahapan yang diambil hingga permasalahan selesai.

Salah satu penyelesaian yang dapat dipakai ialah penyelesaian berdasarkan konsep algoritma Brute Force, dengan menggunakan algoritma ini masalah dapat dipastikan selesai dan memiliki hasil yang optimal namun pemakaian strategi ini sangat tidak efisien dalam hal waktu yang digunakan sangat lama. Beberapa yang dapat digunakan juga ialah strategi algoritma dynamic programming, dan algoritma genetic. Pada kesempatan kali ini untuk menyelesaikan permasalahan knapsack 0-1 ini karena diperlukan algoritma atau strategi yang memberikan hasil yang optimal, efektif dan efisien, maka diperkenalkanlah algoritma Greedy. Algoritma ini digunakan dalam proses analisis permasalahan, dengan konsep yang diusung ialah sesuai dengan namanya greedy (tamak, rakus) yakni membuat pilihan yang terbaik yaitu membuat pilihan optimum local yang dilakukan pada setiap langkah yang ada dengan tujuan bahwa nanti sisa dari pilihan dapat mengarah ke solusi optimum global.

Persoalan knapsack 0-1 ini yang telah dipaparkan bahwa untuk menghasilkan solusi optimal maka penyelesaian masalah ini menggunakan algoritma Greedy. Untuk pengerjaan nya Algoritma Greedy terpecah memiliki beberapa cara atau strategi yang heuristic yang dapat dimanfaatkan untuk memilih objek yang akan dimasukkan kedalam kantong yakni strategi greedy by profit, greedy by weight, greedy density. Perlu diketahui bahwa tidak semua sub-strategi ini dalam satu permasalahan dapat menghasilkan solusi optimum dapat saja salah satu atau salah dua dari sub-strategi ini yang mendapatkan hasil yang optimal ataupun tidak ketiga nya. Greedy by density biasanya memiliki keunggulan yakni seringnya menghasilkan solusi yang optimal tetapi dapat pula tidak memberikan solusi yang optimal. Untuk menyelesaikan permasalahan ini dapat diberikan sentuhan sedikit

perubahan atau tambahan dalam tahapan yang digunakan sesuai konsep greedy by density.

A. Rumusan masalah

1. Bagaimana penyelesaian masalah knapsack 0-1 menggunakan greedy by density
2. Apakah greedy by density mendapatkan solusi optimal
3. Bagaimana tahap lanjutan yang dapat dilakukan agar mendapat solusi optimal

B. Tujuan penelitian

1. Untuk mengetahui tahapan penyelesaian masalah knapsack 0-1 dengan menggunakan algoritma greedy by density
2. Mengetahui suatu permasalahan berhasil mendapatkan solusi optimal atau tidak
3. Mengetahui tahap lanjutan yang dapat digunakan untuk mencapai solusi optimal

C. Metode penelitian

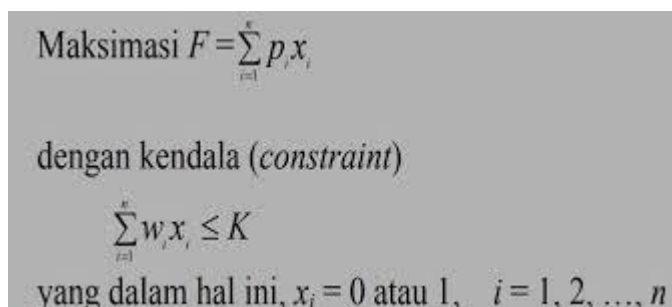
Makalah ini dibuat menggunakan metode pengumpulan data kualitatif yakni berdasarkan observasi, analisis visual, studi pustaka, jurnal lain di internet. Ada pula menggunakan percobaan untuk menyelesaikan masalah ini.

II. PEMBAHASAN

Dasar teori

Knapsack problem merupakan permasalahan mengenai diberikannya n buah objek dan sebuah kantong dengan kapasitas maksimum bobot tertentu K . setiap objek memiliki proeperti bobot (weight) W dan keuntungan (profit) P . permasalahan ini terjadi jika n buah barang tidak dapat dimasukkan semuanya kedalam kantong karena keterbatasan muatan sehingga harus dipilih barang mana saja yang akan dimasukkan kedalam kantong dengan berat tidak melebihi kapasitas kantong dan mendapatkan profit sebesar mungkin. Sehingga pemilihan tiap tahap mengambil pilihan terbaik yakni optimum local agar diakhir menghasilkan optimum global. Disebut 1/0 atau 0-1 knapsack problem karena suatu objek esensi nya boleh diambil atau tidak, angka atau simbol 0 menandakan bahwa barang atau objek tersebut tidak diambil dan symbol 1 menandakan bahwa barang tersebut diambil atau dipilih untuk dimasukkan dalam kantong.

Berikut formulasi matematis persoalan integer knapsack:



Maksimasi $F = \sum_{i=1}^n p_i x_i$
dengan kendala (*constraint*)
 $\sum_{i=1}^n w_i x_i \leq K$
yang dalam hal ini, $x_i = 0$ atau 1 , $i = 1, 2, \dots, n$

Gambar 1.1 Formulasi matematis pada persoalan Integer Knapsack Problem.Sumber:

[http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Brute-Force-\(2016\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Brute-Force-(2016).pdf)

Algoritma Greedy adalah algoritma untuk menyelesaikan permasalahan secara bertahap(Brassard G, 1996). Tahapan penyelesaiannya adalah :

1. Mengambil pilihan terbaik yang dapat diperoleh tiap tahapan nya tanpa memperhatikan konsekuensi kedepan.
2. Berharap bahwa dengan memilih optimum local pada setiap langkah atau tahapan dapat berakhir mendapatkan hasil dengan optimum global.

Strategi Algoritma Greedy

Barang yang sudah diambil tidak dapat dikembalikan lagi dan untuk memilih barang yang akan dimasukkan kedalam knapsack terdapat beberapa strategi greedy yang heuristic.

Heuristic merupakan Teknik yang dirancang untuk memecahkan persoalan dengan mengabaikan apakah Teknik tersebut terbukti benar secara sistematis. Asal mula heuristic jika diturunkan dari Bahasa Yunani yaitu “eureka” yang berarti menemukan. Acuan yang dipakai ialah berbasis pengalaman, dari proses pembelajaran, dan catatan metode ini tidak dijamin mendapatkan solusi optimal dalam semua percobaannya.

Perbedaan dengan algoritma dan heuristic ialah heuristic berlaku sebagai panduan dan algoritma sebagai urutan langkah penyelesaian persoalan. Walaupun secara matematis tidak dapat dibuktikan namun teknik ini sangat berguna. Heuristic yang baik dapat membantu menangani dalam hal waktu pengerjaan dapat memangkas waktu dalam proses tersebut dengan cara mengeliminir kebutuhan untuk mempertimbangkan kemungkinan solusi yang tidak perlu. Tetapi perlu diingat bahwa Teknik ini tidak menjamin selalu dapat memecahkan persoalan tetapi cukup baik memecahkan beberapa persoalan dan seringkali lebih cepat dalam hal pencarian solusi.

Pada kesempatan kali ini untuk menyelesaikan masalah knapsack 0-1 digunakan Algoritma Greedy dengan Teknik Heuristik sebagai berikut:

a. Greedy by profit

Knapsack diisi dengan barang yang memiliki keuntungan terbesar yang dipilih tiap tahapnya. Proses ini memilih objek yang paling menguntungkan akan dipilih terlebih dahulu untuk memaksimumkan keuntungan. Adapaun langkahnya ialah :

1. Mencari nilai keuntungan (profit) dari tiap-tiap barang
2. Barang atau objek-objek tersebut diurutkan berdasarkan profitnya
3. Mengambil satu persatu barang berdasarkan urutan yang dapat ditampung oleh kantong sampai kantong mencapai batas maksimum

kapasitas atau sudah tidak mampu menampung objek lagi.

b. Greedy by weight

Knapsack diisi dengan objek yang memiliki berat yang paling ringan pada setiap tahap. Dengan memaksimalkan jumlah objek yang dapat dimasukkan kedalam kantong untuk memaksimumkan keuntungan. Adapun tahapannya yakni :

1. Mencari nilai berat (weight) dari tiap-tiap barang yang ada .
2. Objek – objek tersebut dilakukan pengurutan berdasarkan nilai bobotnya.
3. Mengambil satu barang tiap tahap berdasarkan urutan yang dapat ditampung oleh kantong sampai kantong penuh atau sudah tidak dapat menerima barang lagi.

c. Greedy by density

Knapsack diisi dengan objek yang mempunyai densitas terbesar pada setiap tahapnya. Mendapatkan nilai densitas dari tiap barang dilakukan dengan cara barang yang memiliki P_j / W_j dimana P_j adalah value atau profit dan W_j ialah weight atau bobot serta $i = (1,2,3,...,n)$. Adapun tahapannya yakni :

1. Mencari nilai profit per-berat (density) dengan membagi profit indeks ke-i dengan weight indeks ke-i dari tiap barang atau objek yang ada
2. Barang-barang tersebut diurutkan berdasarkan density nya.
3. Mengambil barang satu persatu berdasarkan urutan yang dapat ditampung oleh kantong sampai kantong penuh atau sudah tidak dapat menampung barang lagi.

Analisis dan Pembahasan

Analisis kasus:

Diberikan 6 buah objek sebagai berikut :

$(W_1, P_1) = (100, 40);$

$(W_2, P_2) = (50, 35);$

$(W_3, P_3) = (45, 18);$

$(W_4, P_4) = (20, 4);$

$(W_5, P_5) = (10, 10);$

$(W_6, P_6) = (5, 2);$

Dan sebuah knapsack dengan kapasitas $K = 100$. Solusi dengan algoritma Greedy adalah sebagai berikut :

Greedy by profit

Pertama kali dilakukan ialah program mengurutkan secara menurun barang-barang berdasarkan profitnya dari yang terbesar hingga yang terkecil dengan mengabaikan weight satu dengan yang lain dalam arti jika weight lebih besar dan profit

kecil maka tetap diletakkan dibawah yang weight lebih kecil tetapi profit besar.

Kemudian dilakukan proses pemilihan pengambilan barang satu persatu yang dapat ditampung oleh knapsack sampai knapsack penuh atau sudah tidak ada objek lagi yang bisa dimasukkan. Tidak lupa menggunakan symbol 0 atau 1 sebagai penanda barang tidak diambil atau diambil.

Berikut table terkait greedy by profit :

Table 1. Greedy by profit

Property objek				
i	Wi	Pi	Simbol	Status
1	100	40	1	Diambil
2	50	35	0	Tidak Diambil
3	45	18	0	Tidak Diambil
5	10	10	0	Tidak Diambil
4	20	4	0	Tidak Diambil
6	5	2	0	Tidak Diambil

Greedy by weight

Pertama kali dilakukan ialah program mengurutkan secara menurun barang-barang berdasarkan weight nya dari yang terbesar hingga yang terkecil dengan mengabaikan profit satu dengan yang lain dan urutan penomoran penginputan data dalam arti jika weight lebih kecil dan profit lebih besar kecil maka tetap diletakkan dibawah yang weight lebih besar tetapi profit lebih kecil.

Kemudian dilakukan proses pemilihan pengambilan barang satu persatu yang dapat ditampung oleh knapsack sampai knapsack penuh atau sudah tidak ada objek lagi yang bisa dimasukkan. Tidak lupa menggunakan symbol 0 atau 1 sebagai penanda barang tidak diambil atau diambil.

Berikut table terkait greedy by weight :

Table 2. Greedy by weight

Property objek				
i	Wi	Pi	Simbol	Status
1	100	40	0	Tidak Diambil
2	50	35	0	Tidak

				Diambil
3	45	18	1	Diambil
4	20	4	1	Diambil
5	10	10	1	Diambil
6	5	2	1	Diambil

Greedy by density

Pertama kali dilakukan ialah program mencari nilai density tiap barang dengan membagi profit indeks ke-I dengan bobot indeks ke-I.

1. $40/100 = 0,4$
2. $35/50 = 0,7$
3. $18/45 = 0,4$
4. $4/20 = 0,2$
5. $10/10 = 1$
6. $2/5 = 0,4$

Kemudian mengurutkan secara menurun barang-barang berdasarkan weight nya dari yang terbesar hingga yang terkecil Kemudian dilakukan proses pemilihan pengambilan barang satu persatu yang dapat ditampung oleh knapsack sampai knapsack penuh atau sudah tidak ada objek lagi yang bisa dimasukkan. Tidak lupa menggunakan symbol 0 atau 1 sebagai penanda barang tidak diambil atau diambil.

Berikut table terkait greedy by density :

Table 3. Greedy by density

Property objek					
i	W _i	P _i	P _i /W _i	Simbol	Status
5	10	10	1,0	1	Diambil
2	50	35	0,7	1	Diambil
3	45	18	0,4	0	Tidak Diambil
1	100	40	0,4	0	Tidak Diambil
6	5	2	0,4	1	Diambil
4	20	4	0,2	1	Diambil

Berikut lampiran semua nilai yang telah diproses dengan memperhatikan indeks :

Properti objek				Greedy by			Solusi
i	w _i	p _i	p _i /w _i	profit	weight	density	Optimal
1	100	40	0,4	1	0	0	0
2	50	35	0,7	0	0	1	1
3	45	18	0,4	0	1	0	1
4	20	4	0,2	0	1	1	0
5	10	10	1,0	0	1	1	0
6	5	2	0,4	0	1	1	0
Total bobot				100	80	85	100
Total keuntungan				40	34	51	55

Gambar 1.2 mengenai table penyelesaian dengan algoritma Greedy

[http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Brute-Force-\(2016\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Brute-Force-(2016).pdf)

Dapat dipahami bahwa persoalan diatas tidak dapat diselesaikan dengan algoritma greedy baik dari algoritma greedy by profit, by weight maupun by density. Merujuk dari kolom yang menampilkan solusi optimal yang seharusnya dipilih maka pada permasalahan atau soal ini algoritma greedy tidak optimal.

Dapat dianalisis solusi optimal dihasilkan jika barang yang diambil ialah barang ke-2 dan ke-3 dengan weight $50 + 45 + 5 = 100$ kemudian profit yakni $35 + 18 + 2 = 55$. Terdapat kesalahan dalam table yakni indeks ke-6 bernilai 1 atau diambil. Walaupun memiliki kesalahan namun secara konsep yakni mendapatkan keuntungan yang paling besar dengan bobot kantong dapat sama dengan bobot maksimum kantong maupun dapat mendekati bobot maksimum knapsack, sehingga solusi optimum yang seharusnya dipilih yakni barang kedua dan ketiga.

Focus yang dibahas dalam kesempatan ini ialah greedy by density karena telah disinggung diawal bahwa greedy by density memiliki tingkatan sedikit lebih tinggi daripada greedy by profit dan greedy by weight dalam hal mendapatkan penyelesaian yang optimal. Tapi dalam soal ini menunjukkan salah satu contoh kelemahan greedy by density yang tidak selalu optimal walau tingkatannya lebih dari by profit dan by weight.

Greedy by density pada persoalan ini yang diambil ialah nilai yang telah diurutkan sebelumnya dan dipilih yang paling besar hingga terkecil dengan tetap memperhatikan pertambahan bobot agar tidak melebihi bobot maksimum knapsack. Dapat dilihat yang diambil dengan density bernilai

Density	1.0	0.7	0.4	0.2	Stop
Bobot	10	50	5	20	Total = 85

Jika dianalisis untuk solusi optimal dengan density keterkaitan nya, solusi optimal mengambil density yang bernilai 0.7 dengan bobot 50 dengan density bernilai 0.4 dengan bobot 45 dan bernilai 0.4 dengan bobot 5.

Jika merujuk konsep density yang mengambil nilai terbesar maka nilai terbesar adalah 1 dan nilai kedua terbesar adalah 0.7 sehingga dapat pahami jika diberi perubahan sedikit dalam hal

pengambilan solusi optimum pada persoalan ini dan mungkin persoalan lain yang mirip dengan persoalan ini, dapat menambahkan kondisi atau syarat maupun aturan yakni jika solusi optimum pertama telah berhasil didapatkan, proses pencarian solusi dapat dilakukan atau dilanjutkan ke proses kedua pencarian solusi optimum yang kedua.

Setelah mendapatkan hasil perhitungan penjumlahan total bobot dengan total keuntungan maka dapat membandingkan solusi optimal pertama yang telah didapatkan disandingkan dengan solusi optimal yang kedua, lalu diambil solusi yang memiliki nilai terbesar baik dilihat dari bobot maupun keuntungan.

Jika pada solusi kedua ternyata lebih besar keuntungan serta bobot nya daripada solusi yang pertama atau dapat dikatakan solusi kedua lebih optimal daripada solusi yang pertama, maka proses pencarian solusi dapat dilanjutkan. Mengapa? Karena ada saja kemungkinan walaupun kecil solusi yang dihasilkan tahap ketiga lebih optimal daripada solusi yang ditawarkan pada tahap kedua. Proses ini dilakukan terus menerus dengan mengambil nilai terbesar secara mundur untuk dijadikan patokan, jika solusi ke-n sudah lebih optimal daripada solusi ke-n+1 maka proses dapat berhenti dan menampilkan solusi optimal yang telah didapatkan.

Dengan catatan tidak ada perubahan urutan penambahan bobot atau pun profit sesuai dengan urutan awal density ditampilkan sehingga yang berubah disini ialah patokan nilai terbesar yang dapat mundur (nilai n+1 dapat menjadi nilai terbesar pertama jika nilai n memiliki solusi yang tidak optimum dari solusi kedua) hingga menemukan solusi terbaik dengan membandingkan nilai yang didapat solusi ke-n+1 dengan solusi ke-n.

Sehingga dapat hasil dapat ditampilkan sebagai berikut :

Table 3. Greedy by density

Property objek					
i	Wi	P _i	P _i /Wi	Simbol	Status
5	10	10	1,0	0	Tidak Diambil
2	50	35	0,7	1	Diambil
3	45	18	0,4	1	Diambil
1	100	40	0,4	0	Tidak Diambil
6	5	2	0,4	1	Diambil
4	20	4	0,2	0	Tidak Diambil
Total	95	53	-	-	-

Kemudian dibandingkan dengan proses mendapatkan solusi yang pertama yakni :

$$P1 = w = 85 \text{ dan } p = 51$$

$$P2 = w = 100 \text{ dan } p = 55$$

Maka $P2 > P1$

Sehingga solusi yang diambil ialah solusi pada tahap kedua. Untuk memastikan lagi solusi kedua merupakan solusi terbaik maka proses dapat dilanjutkan untuk memastikan hal ini :

Table 3. Greedy by density

Property objek					
i	Wi	P _i	P _i /Wi	Simbol	Status
5	10	10	1,0	0	Tidak Diambil
2	50	35	0,7	0	Tidak Diambil
3	45	18	0,4	1	Diambil
1	100	40	0,4	0	Tidak Diambil
6	5	2	0,4	1	Diambil
4	20	4	0,2	1	Diambil
Total	70	24	-	-	-

Kemudian dibandingkan dengan proses mendapatkan solusi yang kedua yakni :

$$P2 = w = 100 \text{ dan } p = 55$$

$$P3 = w = 70 \text{ dan } p = 24$$

Maka $P2 > P3$

Sehingga solusi yang diambil ialah solusi pada tahap kedua, karena P3 menunjukkan nilai yang lebih rendah daripada P2. Dapat disimpulkan bahwa P2 menghasilkan solusi terbaik atau solusi yang optimal.

Ilustrasi greedy by density

Dianggap sudah dilakukan proses pembagian sehingga telah diketahui nilai density tiap barang:

N = variable penampung nilai weight, p = nilai profit, n = nilai densitas terbesar, dan K adalah bobot maksimum kantong dengan nilai K= 100, baris pertama menyatakan nilai densitas, baris kedua menampilkan bobot dan ketiga profit

0.4	0.7	0.4	0.2	1	0.4
100	50	45	20	10	5
40	35	18	4	10	2

Dilakukan pengurutan data dengan metode atau fungsi sort dengan patokan ialah nilai densitasnya diurutkan dari terbesar hingga terkecil.

$$N = 0, K = 100 \text{ dan } n = 1$$

1.0	0.7	0.4	0.4	0.4	0.2
10	50	45	100	5	20
10	35	18	40	2	4

Jika $N < K$ maka proses berlanjut

$N = 10$, $K = 100$ dan $n = 1$, $p = 10$

1.0	0.7	0.4	0.4	0.4	0.2
10	50	45	100	5	20
10	35	18	40	2	4

$N < K$ (valid) $N = 10+50$, $K = 100$ dan $n = 1$, $p = 10+35$

1.0	0.7	0.4	0.4	0.4	0.2
10	50	45	100	5	20
10	35	18	40	2	4

$N < K$ (valid) $N = 10+50+45$, $K = 100$ dan $n = 1$, $p = 10+35+18$

1.0	0.7	0.4	0.4	0.4	0.2
10	50	45	100	5	20
10	35	18	40	2	4

$N > K$ (tidak valid) $N = 10+50$, $K = 100$ dan $n = 1$, $p = 10+35$

1.0	0.7	0.4	0.4	0.4	0.2
10	50	45	100	5	20
10	35	18	40	2	4

$N < K$ (valid) $N = 10+50+100$, $K = 100$ dan $n = 1$, $p = 10+35+40$

1.0	0.7	0.4	0.4	0.4	0.2
10	50	45	100	5	20
10	35	18	40	2	4

$N > K$ (tidak valid) $N = 10+50$, $K = 100$ dan $n = 1$, $p = 10+35$

1.0	0.7	0.4	0.4	0.4	0.2
10	50	45	100	5	20
10	35	18	40	2	4

$N > K$ (tidak valid) $N = 10+50+5$, $K = 100$ dan $n = 1$, $p = 10+35+2$

1.0	0.7	0.4	0.4	0.4	0.2
10	50	45	100	5	20
10	35	18	40	2	4

$N < K$ (valid) $N = 10+50+5$, $K = 100$ dan $n = 1$, $p = 10+35+2$

1.0	0.7	0.4	0.4	0.4	0.2
10	50	45	100	5	20
10	35	18	40	2	4

$N < K$ (valid) $N = 10+50+5+20$, $K = 100$ dan $n = 1$, $p = 10+35+2+4$

1.0	0.7	0.4	0.4	0.4	0.2
10	50	45	100	5	20

10	35	18	40	2	4
----	----	----	----	---	---

$N < K$ (valid) $N = 10+50+5+20$, $K = 100$ dan $n = 1$, $p = 10+35+2+4$

1.0	0.7	0.4	0.4	0.4	0.2
10	50	45	100	5	20
10	35	18	40	2	4

$N = 85$, $K = 100$, $n = 1$ $p = 51$

Lanjut proses kedua dengan ilustrasi yang sama dengan proses satu, perbedaan hanya ada dibagian $n = 1$ menjadi $n = 0.7$ lalu proses penambahan berlangsung hingga diakhir akan mendapatkan hasil sebagai berikut :

$(N + 1) < K$ (valid) $N = 50+45+5$, $K = 100$ dan $n = 0.7$, $(p + 1) = 35+18+2$

1.0	0.7	0.4	0.4	0.4	0.2
10	50	45	100	5	20
10	35	18	40	2	4

$(N+1) = 100$, $K = 100$, $n = 0.7$ $(p+1) = 55$

Dibandingkan

$(N+1) > N$ & $(p+1) > p$ maka $N+1$ merupakan solusi optimal.

Proses dilakukan hingga 0.4 saja karena program berhenti jika melebihi kapasitas. Dan dapat dilihat pada tahap kedua bobot sudah mencapai bobot maksimum dan keuntungan maksimum sehingga program berhenti dan menampilkan solusi optimal yakni mengambil indeks ke 2, 3 dan 6. Jika ingin dilanjutkan bisa saja dengan nilai $n = 0.4$ tetapi jika sudah mendapatkan solusi yang memenuhi bobot hingga sama dengan bobot maksimum kantong maka proses dapat diberhentikan atau break.

III. KESIMPULAN

Penyelesaian knapsack 0-1 menggunakan greedy by density dapat dibuat dengan diawali penentuan atau perhitungan nilai densitas tiap benda atau barang kemudian dilanjutkan dengan pengurutan dari densitas terbesar lalu dipilih tiap densitas terbesar dengan memperhatikan bobot yang ditambahkan dalam kantong tidak melebihi kapasitas maksimum kantong. jika sudah melebihi maka penambahan stop dan penghitungan profit dilakukan serta total bobot juga dikalkulasikan karena bisa saja mendekati bobot maksimum atau sama dengan bobot maksimum.

Pada kasus ini greedy by density tidak mendapatkan solusi yang optimal. Kemudian disempurnakan dengan mencoba menambahkan beberapa perubahan proses yakni adanya proses lanjutan dengan menjadikan nilai densitas terbesar dapat berubah jika solusi optimum tidak didapatkan pada nilai densitas terbesar yang sesungguhnya. Nilai densitas dapat berubah perujukannya atau pointer dari nilai densitas ke- 1 menjadi ke-2 dan ke-2 menjadi ke-3 hingga ke-n.

Dengan catatan ditiap dua proses yang sudah dijalankan hasilnya dibandingkan jika sudah mendapat solusi optimum maka program berhenti jika belum maka program lanjut dengan pergantian indeks nilai densitas terbesar.

Cara ini belum tentu benar dikarenakan ini berdasarkan pemikiran sang penulis, mungkin saja ada kesalahan dalam penjabaran maupun pemahaman materi sehingga ide ini menjadi salah.

VIDEO LINK AT YOUTUBE (*Heading 5*)

<https://youtu.be/VV5v8auDjVM>

Ucapan Terimakasih

Alhamdulillah puji syukur penulis panjatkan kehadiran Allah swt karena atas berkah dan rahmatnya tugas strategi algoritma tentang pembuatan technical report dapat terselesaikan. Dengan judul “Optimisasi Masalah Knapsack (0-1) Menggunakan Algoritma Greedy by Density” ini merupakan salah satu bentuk pemenuhan tugas mandiri strategi algoritma pada semester 4 prodi teknik informatika. Oleh karena itu, tidak lupa saya sampaikan ucapan terimakasih kepada Bapak Imam Wicaksono S.Si.,M.Si selaku dosen pengampu mata kuliah strategi algoritma yang sudah membimbing saya dalam mempelajari strategi algoritma.

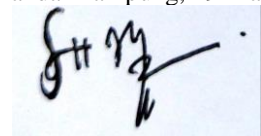
REFERENCES

- [1] Munir, Rinaldi. Diktat Kuliah IF2211 Strategi Algoritma. Program Studi Teknik Informatika ITB. 2020
- [2] Paryati, “Optimasi Strategi Algoritma Greedy Untuk Menyelesaikan Permasalahan Knapsack 0-1”, UPN “Veteran : Yogyakarta, 23 Mei 2009. Link : <https://media.neliti.com/media/publications/173398-ID-optimasi-strategi-algoritma-greedy-untuk.pdf>
- [3] Hapidah, Muhtarulloh, “Penyelesaian Masalah Knapsack (0-1) Menggunakan Algoritma Greedy”. Senter : Bandung, 25 November 2020. Link : <file:///C:/Users/Windows/Downloads/mufid.+39.pdf>
- [4] Ammar, “Implementasi Algoritma Greedy dalam Menyelesaikan Kasus Knapsack Problem Pada Jasa Pengiriman PT CITRA VAN TITIPAN KILAT (TIKI) Kota Makassar”. Jurnal Matematika dan Aplikasinya Vol 1 No2. September 2019. Link : <file:///C:/Users/Windows/Downloads/304-Article%20Text-1597-1-10-20191002.pdf>

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.

Bandar Lampung, 29 Maret 2022



Fujita Rahmah 120140070

