Knapsack Problem Dengan Algoritma Greedy Dalam Efisisensi Kapasitas Pengangkutan Barang

Adam Taufiqurrahman 118140065 (Author)
Program Studi Teknik Informatika
Jurusan Teknik Elektro dan Informatika
Institut Teknologi Sumatera, Jl. Terusan Ryacudu Lampung
E-mail (gmail): adam.118140065@student.itera.ac.id

***Abstract*— Di masa pandemi, masyarakat lebih menyukai transaksi online, karena cenderung lebih mudah dan menyenangkan. Distribusi barang yang didapatkan oleh pembeli dari penjual, tentunya terdapat pihak ketiga yang berperan mengangkut barang yang akan dipesan, yang biasa kita kenal dengan sebutan jasa pengiriman. Pihak jasa pengiriman berperan untuk menyortir barang yang akan dikirim, mulai dari kota tujuan, bentuk barang yang mudah pecah atau tidak serta berat barang. Hal ini patut diperhatikan karena dngan strategi pengiriman kapasitas yang tepat. Pihak jasa pengiriman akan lebih efisien dalam mengatur barang yang akan didistribusikan. Dengan menggunakan algoritma greedy, kita dapat menyelesaikan persoalan mengenai jasa pengiriman barang, sehingga jika ingin menentukan keuntungan dan optimasi ruang kepada transportasi miliknya, hal ini dapat di implementasikan pada pengiriman barang bedasarkan pengurutan berat , profit dan kapasistas yang tersedia didalamnya sehingga pengiriman barang akan menjadi lebih efisien.**

Keywords—Algoritma greedy, pengiriman Barang Knapsack

I. PENDAHULUAN

Di masa pandemi, masyarakat lebih menyukai transaksi online, karena cenderung lebih mudah dan menyenangkan. Distribusi barang yang didapatkan oleh pembeli dari penjual, tentunya terdapat pihak ketiga yang berperan mengangkut barang yang akan dipesan, yang biasa kita kenal dengan sebutan jasa pengiriman. Pihak jasa pengiriman berperan untuk menyortir barang yang akan dikirim, mulai dari kota tujuan, bentuk barang yang mudah pecah atau tidak serta berat barang. Hal ini patut diperhatikan karena dngan strategi pengiriman kapasitas yang tepat. Pihak jasa pengiriman akan lebih efisien dalam mengatur barang yang akan didistribusikan.

Masalah *knapsack* (KP) dikenal sebagai masalah optimasi kombinatorial yang dipelajari dengan baik dan telah dipelajari secara menyeluruh dalam beberapa dekade terakhir. Secara umum, KP diklasifikasikan menjadi SKP yang dapat dipisahkan dimana barang-barangnya dapat dipisahkan secara sewenang-wenang dan 0-1 KP (KP01) yang itemnya tidak dapat dipisah. Di antara mereka, KP01 adalah jenis KP yang penting karena kekerasan NP-nya (Merkle, 2008).

Oleh karena itu, semakin banyak peneliti telah memperhatikan masalah optimasi KP01. Terutama, Martello memberikan ulasan yang komprehensif dengan pembahasan lebih lanjut mengenai teknik-teknik yang biasa digunakan dalam penyelesaian KP01. Karena KP01 telah terbukti sebagai NP-hard, metode yang digunakan untuk menyelesaikan KP01 telah dibagi menjadi tiga: kategori, yaitu, metode eksak dengan solusi eksak, metode metaheuristik, dan metode heuristik dengan perkiraan solusi.

Tentang metode eksak, beberapa penelitian terkait telah dilakukan. Pengembangkan algoritma pemrograman dinamis sesuai dengan pengurangan negara dan menggunakan banyak hubungan dominasi komplementer untuk meningkatkan algoritma pemrograman dinamis (Kolesar, 2008).

Selanjutnya, proses iterasi yang kompleks tidak disesuaikan dengan optimasi masalah rekayasa. Algoritma *Greedy* memiliki perbedaan dengan algoritma lainnya diantaranya yaitu pertama dari segi kecepatan. Perhitungan algoritma *greedy* menggunakan komputasi yang lebih dikarenakan algoritma *greedy* menggunakan prinsip pemilihan keputusan disetiap langkahnya.

Yang kedua yaitu dari segi ketepatan dikarenakan algoritma *greedy* tidak beroperasi secara menyeluruh

terhadap semua alternatif solusi yang ada sehingga algoritma *greedy* tidak selalu memberikan hasil yang optimal, akan tetapi memberikan hasil optimal ketika terdapat banyak alternatif yang diberikan. Adapun rumusan masalah dalam penelitian ini adalah bagaimana implementasi algoritma *greedy* dalam menyelesaikan kasus *knapsack problem.*

Adapun tujuan penelitian ini berdasarkan rumusan masalah adalah mengetahui implementasi algoritma *greedy* dalam menyelesaikan kasus *knapsack problem* pada jasa pengiriman barang. Hasil penelitian ini dapat menambah pengetahuan dan wawasan peneliti tentang mata kuliah strategi algoritma, algoritma *greedy* dan knapsack problem, serta kaitan antara kedua mata kuliah tersebut.Selain itu, penelitian ini menjadi pengalaman berharga bagi peneliti dalam menerapkan disiplin ilmu yang diperoleh di bangku perkuliahan

II. TEORI DASAR

A. Pengertian Algoritma

Menurut catatan sejarah Abu ja'far Muhammad Ibnu Musa Al- khawarizmi, penulis buku "Aljabar wal muqabala" beberapa abad yang lalu (pada abad IX), dianggap sebagai pencetus perama algoritma karena didalam buku tersebut Abu ja'far menjelaskan langkah-langkah

dalam menyelesaikan berbagai persoalan aritmatika (aljabar), kemungkinan besar kata Algoritma diambil dari kata "Al-khawarizmi" yang kemudian berubah menjadi "Algorism' selanjutnya menjadi "Algorithm".

Algoritma adalah setiap prosedur komputasi yang terdefinisi dengan baik yang mengambil beberapa nilai atau seperangkat nilai-nilai, sebagai masukan dan menghasilkan beberapa nilai, atau seperangkat nilai-nilai, sebagai output.

Sebuah algoritma merupakan langkah komputasi yang mengubah input ke output. atau dapat pula melihat sebuah algoritma sebagai alat bantu untuk memecahkan masalah (Cormen, 2001)

B. Knapsack Problem

Knapsack problem merupakan masalah optimasi kombinatorial. Sebagai contoh adalah suatu kumpulan barang masing masing memiliki berat dan nilai, kemudian akan ditentukan jumlah tiap barang untuk dimasukkan dalam koleksi sehingga total berat kurang dari batas yang diberikan dan nilai total seluas mungkin (Gao, 2014). *Knapsack problem* merupakan masalah optimasi klasik berupa pengapakan barang yang didefinisikan sebagai berikut:

1. Di berikan sebuah *knapsack* (wadah) dan *n* objek (setiap objek bisa terdiri
daribanyak barang);
2. Objek *i* memiliki berat
> 0 dan nilai > 0;
3. *Knapsack* berkapasitas W;
4. Tujuan: tentukan jumlah barang dari setiap objek yang harus dimasukkan ke
dalam *knapsack* sehingga total nilainya semaksimal mungkin.

C. Algoritma Greedy

Algoritma untuk masalah optimasi biasanya dilakukan melalui urutan langkah-langkah, dengan satu set pilihan pada setiap langkah. Bagi banyak masalah optimasi, menggunakan pemrograman dinamisuntuk menentukan pilihan terbaik terlalu berlebihan, Sebuah algoritma *greedy* selalu membuat pilihan yang terlihat terbaik padasaat ini.

Artinya itu membuat pilihan yang optimal secara lokal dengan harapan bahwa pilihan ini akan mengarah pada solusi optimal global. Algoritma *greedy* tidak selalu menghasilkan solusi optimal, tapi dapat dilakukan menyelesaikan berbagai masalah. Cukup kuat dan bekerja dengan baik untuk berbagai masalah termasuk merancang kode data (*Huffman*), struktur kombinasi *matroids*, algoritma minimum *spanningtree*, algoritma *Djikstra* untuk menentukan jalur terpendek, TSP, dan juga pada kasus *Knapsack problem*.

III. ANALISIS DAN PEMBAHASAN

Pada contoh permasalahan berikut sebuah jasa pengiriman barang ingin menentukan keuntungan dan optimasi ruang kepada transportasi miliknya hal ini dapat di implementasikan pada pengiriman barang bedasarkan pen gurutan berat , profit dan kapasistas yang tersedia didalamnya sehingga pengiriman barang

Pada gambar tabel 1 terdapat alat angkut dengan kapasitas 100 kg terdapat 4 buah barang dengan ukuran beragam :

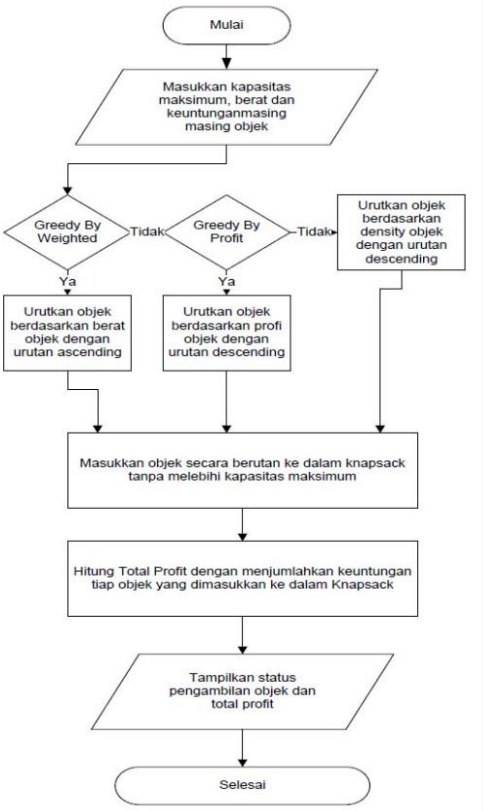
Barang	Weight	Profit	Density
1	23	67	2.913043478
2	45	86	1.911111111
3	67	139	2.074626866
4	39	42	1.076923077

Dengan menggunakan algoritma greedy dapat di urutkan sebagai berikut :

Keterangan :
GBW : Greedy By Weight
GBP : Greedy By Profit
GBD : Greedy By Density
0 : Barang tidak diangkut
1 : Barang diangkut

GBW	GBP	GBD
1	1	1
0	0	0
0	1	1
1	0	0
109	206	206

Berikut ialah flow chart pada program yang telah saya buat :



Pertama pengguna akan diperintahkan untuk memasukan kapasitas

maksimum angkutan , berat objek dan keuntungan dari masing masing objek , lalu program akan mengurutkan bedasrkan berat dan keuntungan yang di dapat. Lalu program secara otomatis akan memproses dan menampilkan status pengambilan objek dan total profit.

Berikut terdapat source code yang telah saya buat :

```
void main()
{
    int kapasitas, no_barang, cur_weight, barang; //membuat deklarasi variabel
   int used[10];
   float total_profit;
   int i;
   int berat[10];
   int harga[10];

   printf("masukan total kapasitas :\n");
    scanf("%d", &kapasitas);    // memasukan total kapasitas barang

    printf("masukan jumlah dari barang :\n");    //jumlah dari barang
   scanf("%d", &no_barang);   //jumlah barang

     printf("masukan berat dan harga dari %d barang:\n", no_barang);
   for (i = 0; i < no_barang; i++)   //menggunakan perulangan for untuk melakukan pemrosesan
   {
        printf("berat(kg)[%d]:\t", i);     //masukan berat barang sesuai dengan kapasitas yang telah di tentukan
      scanf("%d", &berat[i]);
        printf("harga(Rp)[%d]:\t", i);  //masukan jumlah harga barang
      scanf("%d", &harga[i]);
   }

   for (i = 0; i < no_barang; ++i)
      used[i] = 0;

   cur_weight = kapasitas;
   while (cur_weight > 0)
   {
      barang = -1;
      for (i = 0; i < no_barang; ++i)
         if ((used[i] == 0) &&
            ((barang == -1) || ((float) harga[i] / berat[i] > (float) harga[barang] / berat[barang])))
            barang = i;

      used[barang] = 1;
      cur_weight -=  berat[barang];
      total_profit += berat[barang];
      if (cur_weight >= 0)
            printf("sukses menambahkan barang %d (%d Rp., %dKg) pada keranjang. tempat tersisa: %d.\n", barang + 1, harga[barang], berat[barang], cur_weight);
      else
      {
            int item_percent = (int) ((1 + (float) cur_weight / berat[barang]) * 100);
            printf("tambahkan %d%% (%d Rb., %dKg) ke barang %d pada keranjang.\n", item_percent,  harga[barang], berat[barang], barang + 1);
         total_profit -= harga[barang];
            total_profit += (1 + (float)cur_weight / berat[barang]) * harga[barang];
      }
   }

   printf("Filled the bag with objects worth %.2f Rb.\n", total_profit);
```

berikut out program yang di hasilkan :

```
masukan total kapasitas :
10
masukan jumlah dari barang :
3
masukan berat dan harga dari 3 barang:
berat(kg)[0]:   5
harga(Rp)[0]:   10
berat(kg)[1]:   6
harga(Rp)[1]:   20
berat(kg)[2]:   7
harga(Rp)[2]:   30
sukses menambahkan barang 3 (30 Rp., 7Kg) pada keranjang. tempat
tambahkan 50% (20 Rb., 6Kg) ke barang 2 pada keranjang.
Filled the bag with objects worth 3.00 Rb.
```

IV. KESIMPULAN

Dengan menggunakan algoritma greedy, kita dapat menyelesaikan persoalan mengenai jasa pengiriman barang, sehingga jika ingin menentukan keuntungan dan optimasi ruang kepada transportasi miliknya, hal ini dapat di implementasikan pada pengiriman barang bedasarkan pengurutan berat , profit dan kapasistas yang tersedia didalamnya sehingga pengiriman barang akan menjadi lebih efisien.

Penulis berharap algoritma ini dpat dikembangkan lagi menjadi lebih baik dan menjadi lebih spesifik sehingga dapat berdampak baik bagi ekspedisi pengirimn jasa.

V. UCAPAN TERIMA KASIH

Puji dan syukur penulis sampaikan kepada Tuhan yang Maha Esa, berkat rahmat dan hidayah-Nya lah penulis dapat menyelesaikan makalah ini. Terimaksih kepada pihak Dosen dan teman-teman kelas yang telah senantiasa belajar bersam adalam mata kuliah Strategi Algoritma. Penulis sangat terbuka terhada kritik dan saran demi perbaikan penulisan ini selanjutnya. Demikian yang dapat penulis sampaikan, salah kata atau kurang kata mohon maaf. Sekian Terimakasih.

References

Cormen. (2001). An algorithm of 0-1 knapsack problem based on economic model. *Journal of Applied Mathematics and Physics 1*.

Gao. (2014). A quantum-inspired artificial immune system for the multiobjective 0-1 knapsack problem, Applied Mathematics.

Kolesar, P. (2008). A branch and bound algorithm for the knapsack problem. *Management Science*.

Merkle, R. (2008). *Hiding information and signatures in trapdoor knapsacks.* Eropa: IEEE Transactions on Information Theory.

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.

Lampung, 29 Maret 2022

Adam Taufiqurrahman

118140065