SKRIPSI

PENCARIAN K SOLUSI TERBAIK UNTUK PERMASALAHAN 0/1 *KNAPSACK* DENGAN ALGORITMA *BRANCH AND BOUND*

Luzman Irshadi

NPM: 2012730077

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2018

UNDERGRADUATE THESIS

FINDING K BEST SOLUTION FOR 0/1 KNAPSACK PROBLEM USING BRANCH AND BOUND ALGORITHM

Luzman Irshadi

NPM: 2012730077

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2018

LEMBAR PENGESAHAN

PENCARIAN K SOLUSI TERBAIK UNTUK PERMASALAHAN 0/1 *KNAPSACK* DENGAN ALGORITMA *BRANCH AND BOUND*

Luzman Irshadi

NPM: 2012730077

Bandung, 5 Desember 2018

Menyetujui,

Pembimbing

Luciana Abednego, M.T.

Ketua Tim Penguji

Vania Natali, M.T.

Anggota Tim Penguji

Mariskha Tri Adithia, P.D.Eng

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PENCARIAN K SOLUSI TERBAIK UNTUK PERMASALAHAN 0/1 *KNAPSACK* DENGAN ALGORITMA *BRANCH AND BOUND*

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 5 Desember 2018

Luzman Irshadi
NPM: 2012730077

ABSTRAK

Permasalahan *knapsack* (*knapsack problem*) merupakan permasalahan untuk mencari kombinasi *item-item* yang memiliki berat $(w_1, w_2, ...w_n)$ dan nilai $(v_1, v_2, ...v_n)$ sehingga dapat dimasukkan ke dalam sebuah tas dengan kapasitas C. Kombinasi yang dipilih haruslah kombinasi yang paling berharga yang cukup untuk masuk ke dalam *knapsack*. Dengan kata lain tujuan penyelesaian permasalahan *knapsack* adalah untuk mencari kombinasi *item-item* yang total beratnya tidak melebihi kapasitas *knapsack* dan memiliki nilai tertinggi.
Selain itu solusi yang dicari tidak hanya berupa 1 solusi saja tetapi beberapa solusi terbaik (k *best solution*). Untuk menyelesaikan permasalahan ini pendekatan yang paling mudah dengan menggunakan algoritma *exhaustive search*. Meskipun algoritma ini dapat memberikan solusi yang terbaik, kompleksitas waktunya terlalu besar $(O(2^n))$ sehingga tidak cocok digunakan untuk menyelesaikan masalah dengan jumlah objek yang sangat banyak, karena itu dalam skripsi ini akan digunakan pendekatan metode *branch and bound*yang diharapkan dapat mempercepat proses pencarian.
Untuk algoritma *branch and bound* diimplementasikan secara langsung dari langkah algoritma tersebut dengan sedikit modifikasi dan digunakan algoritma *exhaustive search* menggunakan *bit-string* sebagai algoritma pembanding.
Berdasarkan hasil pengujian algoritma *exhaustive search* dan *branch and bound*, algoritma *branch and bound* lebih baik daripada algoritma *exhaustive search* untuk kasus dengan jumlah *item* yang cukup banyak, sedangkan algoritma *exhaustive search* lebih baik untuk kasus dengan jumlah *item* yang terbilang sedikit.

Kata-kata kunci: *Exhaustive Search*, *Branch and Bound*, *Bit-String*, K *Best Solution*, *Knapsack Problem*, *Upper Bound*, *Lower Bound*, *Breadth First Search*

ABSTRACT

Knapsack problem is a case to find the combinations of some items, each with their own weights $((w_1, w_2, ...w_n))$ and values $(v_1, v_2, ...v_n)$ so that they can be stored into a knapsack with capacity of C. The selected combination must be the most valuable combinations of all and can be stored into a knapsack. In other words, the goal to solve this problem is to find the combinations of items with a total of their weight not exceeding the capacity of knapsack and with highest values. Also the best solution can be more than 1 solutions or k best solutions. To solve this problem the easiest method is using exhaustive search algorithm. Even if this algorithm could give the best solution, the time complexity is too high $(O(2^n))$ and it's not suitable for a case with large amount of items hence the using of branch and bound algorithm. The using of branch and bound are expected to accelerate the search process.
For branch and bound algorithm it will use it's own algorithm to be implemented into program with a little modification and exhaustive search algorithm implemented using bit-string for comparison.
Based on the results of testing exhaustive search and branch and bound algorithm, the branch and bound algorithm is better than for a cases that involve large number of items, while exhaustive search algorithm is better for a cases with the number of items is relatively small.

Keywords: Exhaustive Search, Branch and Bound, Bit-String, K Best Solution, Knapsack Problem, Upper Bound, Lower Bound, Breadth First Search

Skripsi ini dipersembahkan untuk Teknik Informatika UNPAR dan diri sendiri

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas seluruh berkat dan rahmat-Nya yang diberikan kepada penulis sehingga dapat menyelesaikan skripsi dengan judul **Pencarian K Solusi Terbaik Untuk Permasalahan 0/1 *Knapsack* dengan Algoritma *Branch and Bound*** dengan baik dan tepat waktu. Penulis juga berterima kasih kepada pihak-pihak yang telah memberikan dukungan serta bantuan kepada penulis dalam menyelesaikan skripsi ini:

1. Orang tua dan keluarga yang selalu memberikan dukungan kepada penulis.
2. Bapak Husnul Hakim selaku dosen pembimbing yang telah membimbing penulis dalam menyelesaikan skripsi ini.
3. Ibu Vania Natalia dan Ibu Mariskha Tri Adithia sebagai penguji dalam sidang akhir skripsi ini.
4. Rekan-rekan Teknik Informatika UNPAR angkatan 2012 yang saling membantu dan menyemangati dalam penyelesaian skripsi ini.
5. Pihak lain yang belum disebutkan yang berperan dalam penyelesaian skripsi ini.

Akhir kata, penulis berharap agar skripsi ini dapat bermanfaat bagi pembaca yang hendak melakukan penelitian dan pengembangan terkait dengan skripsi ini.

Bandung, Desember 2018

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Permasalahan *knapsack* (*knapsack problem*) merupakan permasalahan untuk mencari kombinasi *item-item* yang memiliki berat $(w_1, w_2, ...w_n)$ dan *value* $(v_1, v_2, ...v_n)$ dapat dimasukkan ke dalam sebuah tas dengan kapasitas C, kombinasi yang dipilih haruslah kombinasi yang paling berharga yang cukup untuk masuk ke dalam *knapsack*.[1] Dengan kata lain tujuan penyelesaian permasalahan *knapsack* adalah untuk mencari kombinasi *item-item* yang total beratnya tidak melebihi kapasitas *knapsack* dan memiliki total *value* tertinggi. Permasalahan *knapsack* dapat dikelompokkan sebagai berikut:

- 0/1 *Knapsack problem*
  Permasalahan dengan kondisi *item* harus dibawa secara keseluruhan atau tidak dibawa sama sekali. Contoh *item* seperti televisi, laptop, kursi, dsb.
- *Fractional knapsack problem*
  Permasalahan dengan kondisi *item* dapat dibawa hanya sebagian saja. Contoh *item* adalah beras, gula, garam, dsb.
- *Bounded knapscak problem*
  Permasalahan dengan kondisi setiap *item* memiliki jumlah yang terbatas.
- *Unbounded knapsack problem*
  Permasalahan dengan kondisi setiap *item* memiliki jumlah yang tidak terbatas.

Untuk menyelesaikan permasalahan ini pendekatan yang paling mudah dengan menggunakan algoritma *exhaustive search*. Algoritma ini mencari semua kombinasi dan permutasi dari objek-objek yang ada, kemudian dari kombinasi-kombinasi tersebut dipilih yang memenuhi seluruh *constraint* dan memiliki hasil yang diinginkan atau sesuai tujuan. Semakin banyak objek, semakin banyak kemungkinan kombinasi sehingga semakin banyak pula kemungkinan solusinya. Meskipun algoritma ini dapat memberikan solusi yang terbaik, kompleksitas waktunya terlalu besar $(O(2^n))$ sehingga tidak cocok digunakan untuk menyelesaikan masalah dengan jumlah objek yang sangat banyak.
Berdasarkan kekurangan algoritma *exhaustive search* tersebut, dalam skripsi ini akan digunakan pendekatan metode *branch and bound*. *Branch and bound* adalah strategi dari *divide and conquer*, idenya adalah untuk memecah wilayah yang layak (*feasible*) menjadi subdivisi yang lebih mudah dikelola, dan apabila diperlukan dilakukan pemecahan subdivisi tersebut lebih lanjut.[2] Tujuan utama alogritma ini ada untuk membuat *search tree* yang akan dihasilkan menjadi sekecil mungkin dengan *pruning*. Selain itu solusi terbaik yang dicari bisa lebih dari 1 solusi (k solusi). *Pruning* yang dilakukan pada algoritma *branch and bound* inilah yang diharapkan dapat mempercepat proses pencarian k solusi terbaik.

1.2 Rumusan Masalah

Rumusan masalah pada skripsi ini adalah sebagai berikut:

- Bagaimana menggunakan teknik *exhaustive search* untuk menyelesaikan permasalahan *knapsack* sehingga diperoleh k solusi terbaik?
- Bagaimana menggunakan teknik *branch and bound* untuk menyelesaikan permasalahan *knapsack* sehingga diperoleh k solusi terbaik?
- Bagaimana membangun perangkat lunak untuk memperoleh k solusi terbaik dari permasalahan *knapsack* dengan *exhaustive search* dan *branch and bound*?

1.3 Tujuan

Tujuan dari skripsi ini adalah:

- Mempelajari teknik *exhaustive search* untuk menyelesaikan permasalahan *knapsack* untuk mendapatkan k solusi terbaik.
- Mempelajari teknik *branch and bound* untuk menyelesaikan permasalahan *knapsack* untuk mendapatkan k solusi terbaik.
- Membangun perangkat lunak untuk menyelesaikan permasalahan *knapsack* dengan *exhaustive search* dan *branch and bound*.

1.4 Batasan Masalah

Dalam mengimplimentasi algoritma *branch and bound* untuk permasalahan *knapsack* batasan masalah yang digunakan adalah jumlah *item* (nilai N) yang digunakan tidak lebih dari 30 ($N \leq 30$). Karena nilai k merupakan *integer* dengan nilai maksimal $(2^{31}) - 1$ dan program tidak dapat menerima input nilai k melebihi 2^{30}.

1.5 Langkah-langkah Penyusunan Skripsi

Langkah-langkah penyusunan skripsi adalah sebagai berikut:

- Studi literatur *knapsack problem*, *exhaustive search*, dan *branch and bound*.
- Analisis kasus permasalahan 0/1 *knapsack*.
- Implementasi algoritma *exhaustive search* dan *branch and bound* untuk permasalahan 0/1 *knapasack*.
- Uji coba hasil implementasi algoritma *exhaustive search* dan *branch and bound*.
- Pengambilan kesimpulan dan saran berdasarkan hasil pengujian.
- Penyusunan dokumentasi hasil studi literatur, implementasi, uji coba, serta pengambilan kesimpulan dan saran.

1.6 Sistematika Pembahasan

Laporan skripsi ini dikelompokkan menjadi beberapa bab sebagai berikut:

- Bab I Pendahuluan
  Berisi latar belakang, rumusan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan.
- Bab II Landasan Teori
  Berisi teori-teori yang digunakan dalam skripsi ini beserta pengertian dan definisi yang didapatkan dari berbagai artikel, jurnal, buku, ataupun *e-book* yang membahas teori-teori tersebut.
- Bab III Analisis
  Mengandung contoh kasus mengenai 0/1 *knapsack problem* beserta langkah-langkah penyelesainnya.
- Bab IV Perancangan
  Berisi penjelasan mengenai kelas-kelas beserta atribut dan *method* yang ada di dalam-nya.
- Bab V Pengujian
  Berisi tentang hasil pengujian atas implementasi algoritma branch and bound untuk penyelesaian 0/1 *knapsack problem*.
- Bab VI Kesimpulan dan Saran
  Berisi kesimpulan dan saran berdasarkan hasil analisa pada implementasi algoritma *branch and bound* untuk penyelesaian 0/1 *knapsack problem*.