

ABSTRAK

Knapsack problem adalah salah satu permasalahan optimasi kombinatorial yang penting dalam bidang ilmu komputer dan matematika diskret. Tujuan dari permasalahan ini adalah untuk memilih sejumlah item dari kumpulan item yang diberikan sehingga nilai total dari item-item yang dipilih maksimum, dengan mempertimbangkan batasan kapasitas tertentu. Dalam penelitian ini, Penulis membandingkan dua pendekatan yang umum digunakan untuk menyelesaikan *integer knapsack problem*, yaitu pendekatan *dynamic programming* dan *brute force*. Pendekatan *dynamic programming* memecahkan permasalahan dengan memecahkannya menjadi submasalah yang lebih kecil dan kemudian membangun solusi optimal secara bertahap. Di sisi lain, pendekatan *brute force* memeriksa semua kemungkinan kombinasi item yang mungkin untuk mencari solusi optimal. Penulis menerapkan kedua pendekatan ini pada beberapa contoh kasus *knapsack* dengan ukuran yang berbeda-beda dan mengukur kinerja keduanya dalam hal waktu eksekusi dan penggunaan sumber daya komputasi. Matriks unjuk kerja yang digunakan pada penelitian ini adalah Nilai Utilitas Maksimum, *time complexity*, dan *space complexity*.

Kata Kunci: *knapsack problem*, *integer knapsack problem*, *dynamic programming*, *brute force*



ABSTRACT

The Knapsack problem is one of the important combinatorial optimization problems in the fields of computer science and discrete mathematics. The goal of this problem is to select a subset of items from a given set of items such that the total value of the selected items is maximized, while considering a certain capacity constraint. In this study, the author compares two commonly used approaches to solving the integer Knapsack problem, namely dynamic programming and brute force. The dynamic programming approach solves the problem by breaking it down into smaller subproblems and gradually building an optimal solution. On the other hand, the brute force approach examines all possible combinations of items to find the optimal solution. The author applies both of these approaches to several Knapsack problem instances with varying sizes and measures their performance in terms of execution time and computational resource usage. The performance metrics used in this study are Maximum Utility Value, Time Complexity, and Space Complexity.

Keywords: knapsack problem, integer knapsack problem, dynamic programming, brute force

