

Memaksimalkan Pemanfaatan Ruang Kontainer dengan Pemrograman Dinamis: Pendekatan Berbasis Knapsack

by Muhammad Arifin Ilham

Submission date: 16-Jul-2024 10:40PM (UTC+0700)

Submission ID: 2417793144

File name: 1547_IGNITE_VOL2_NO3_JULY_HAL01.pdf (254.85K)

Word count: 1609

Character count: 9687

Memaksimalkan Pemanfaatan Ruang Kontainer dengan Pemrograman Dinamis: Pendekatan Berbasis Knapsack

Muhammad Arifin Ilham¹, Tety Citra Natha², Nur Haniatin Jannah³, Efrans Christian⁴

¹⁻⁴ Fakultas Teknik Universitas Palangka Raya

Kampus UPR Tunjung Nyaho, Jl. Yos Sudarso, Palangka Raya 73112, Indonesia

Korespondensi penulis: imarifin593@gmail.com

Abstract. *This study examines the use of dynamic programming in the context of container filling optimization, known as the knapsack problem. This problem requires selecting a number of goods with a certain volume and value to be loaded into a container with a certain capacity. We develop a knapsack-based algorithm using dynamic programming techniques to maximize container space utilization. By considering the volume and value of goods, our algorithm is able to achieve optimal results. Through a case study involving 25 items with predetermined volumes and values, we demonstrate the effectiveness of our algorithm in improving container space utilization. Our experimental results show significant improvements in container space utilization compared to naive filling methods. This research shows that the knapsack approach with dynamic programming can be an effective solution to the container filling problem in the context of logistics and optimization.*

Keywords: *Dynamic Programming, Knapsack Problem, Container Packing Optimization, Container Space Utilization, Knapsack-Based Algorithm*

Abstrak. Studi ini meneliti penggunaan pemrograman dinamis dalam konteks optimasi pengisian kontainer, yang dikenal sebagai masalah knapsack. Masalah ini mengharuskan pemilihan sejumlah barang dengan volume dan nilai tertentu untuk dimuat ke dalam kontainer dengan kapasitas tertentu. Kami mengembangkan algoritma berbasis knapsack menggunakan teknik pemrograman dinamis untuk memaksimalkan pemanfaatan ruang kontainer. Dengan mempertimbangkan volume dan nilai barang, algoritma kami mampu mencapai hasil optimal. Melalui studi kasus yang melibatkan 25 barang dengan volume dan nilai yang telah ditentukan, kami menunjukkan efektivitas algoritma kami dalam meningkatkan pemanfaatan ruang kontainer. Hasil eksperimen kami menunjukkan peningkatan signifikan dalam pemanfaatan ruang kontainer dibandingkan dengan metode pengisian yang naif. Penelitian ini menunjukkan bahwa pendekatan knapsack dengan pemrograman dinamis dapat menjadi solusi yang efektif untuk masalah pengisian kontainer dalam konteks logistik dan optimasi.

Kata kunci: Pemrograman Dinamis, Masalah Knapsack, Optimasi Pengisian Kontainer, Pemanfaatan Ruang Kontainer, Algoritma Berbasis Knapsack.

PENDAHULUAN

Dalam industri logistik dan pengiriman, efisiensi dalam penggunaan ruang kontainer sangat penting untuk mengoptimalkan biaya dan meningkatkan kinerja operasional. Salah satu tantangan utama dalam mengelola pengisian kontainer adalah masalah knapsack, di mana kita harus memilih sejumlah barang dengan volume dan nilai tertentu untuk dimuat ke dalam kontainer dengan kapasitas yang ditentukan. Masalah ini merupakan masalah optimasi yang kompleks dan memerlukan pendekatan yang cermat untuk memaksimalkan pemanfaatan ruang kontainer.

Dalam konteks ini, pendekatan menggunakan pemrograman dinamis menjadi solusi yang menarik dan efektif. Pemrograman dinamis adalah teknik algoritmik yang memungkinkan kita untuk menyelesaikan masalah optimasi dengan memecahnya menjadi submasalah yang lebih kecil dan menyelesaikan sub masalah tersebut secara iteratif. Dengan menerapkan pemrograman dinamis pada masalah knapsack, kita dapat mengembangkan algoritma yang efisien untuk memaksimalkan pemanfaatan ruang kontainer.

Dalam artikel ini, kami menyajikan pendekatan berbasis knapsack yang menggunakan pemrograman dinamis untuk memaksimalkan pemanfaatan ruang kontainer. Kami menguraikan langkah-langkah pengembangan algoritma, menerapkannya pada studi kasus yang melibatkan berbagai barang dengan volume dan nilai yang telah ditentukan, dan mengevaluasi kinerja algoritma melalui serangkaian eksperimen. Diharapkan hasil dari penelitian ini dapat memberikan wawasan yang berharga dalam meningkatkan efisiensi pengisian kontainer dalam konteks industri logistik dan pengiriman.

METODE PENELITIAN

Algoritma Dynamic Programming

Dynamic Programming adalah metode pemecahan masalah yang sering digunakan dalam ilmu komputer dan matematika untuk menyelesaikan masalah optimasi atau pencarian solusi terbaik dari sekumpulan pilihan yang mungkin. Prinsip dasarnya adalah memecah masalah besar menjadi serangkaian sub masalah yang lebih kecil dan lebih mudah dipecahkan. Solusi untuk setiap submasalah disimpan dan digunakan kembali untuk menghindari pengulangan perhitungan yang tidak perlu. Ada beberapa ciri khas yang membedakan masalah yang dapat diselesaikan menggunakan metode Dynamic Programming:

1. Sub Masalah Bersifat Optimal: Solusi optimal dari masalah keseluruhan dapat ditemukan dengan menggabungkan solusi optimal dari sub masalah yang lebih kecil. Ini disebut sebagai properti optimal substructure.
2. Overlapping Subproblems: Dalam memecahkan masalah yang lebih besar, kita mungkin harus memecahkan sub masalah yang sama beberapa kali. Dynamic Programming menghindari perhitungan berulang ini dengan menyimpan hasil dari setiap sub masalah yang telah dipecahkan, sehingga mengurangi waktu komputasi secara signifikan.
3. Keselarasan (Principle of Optimality): Solusi optimal dari masalah global harus berasal dari keputusan lokal yang optimal pada setiap tahap. Artinya, kita memastikan bahwa memilih solusi terbaik pada tahap sebelumnya akan membantu kita mencapai solusi terbaik secara keseluruhan.

Knapsack 0/1

Knapsack problem merupakan masalah optimasi kombinasi dengan tujuan memaksimalkan total nilai dari barang-barang yang dimasukkan ke dalam knapsack atau suatu wadah tanpa melewati kapasitasnya. Dalam varian ini, setiap barang hanya dapat dipilih satu kali atau tidak dipilih sama sekali. Artinya, jika sebuah barang telah dimasukkan ke dalam kontainer(knapsack), maka tidak dapat dimasukkan kembali. Dalam konteks ini, setiap barang biasanya diidentifikasi dengan indeks atau nomor tertentu.

HASIL DAN PEMBAHASAN

Pada penelitian kali ini, kami menggunakan algoritma Dynamic Programming dengan metode Knapsack 0/1 untuk memaksimalkan dan mengoptimalkan kapasitas dari sebuah kontainer yang memiliki volume sebesar 2.5 m^3 dengan beberapa daftar barang yang ada pada tabel berikut ini:

Tabel Daftar Barang

NO	NAMA BARANG	VOLUME (m³)	HARGA (\$)
1	Panci	0.2	50
2	Wajan	0.5	70
3	Teflon	0.7	100
4	Cangkir	0.4	80
5	Beras	0.6	90

6	Meja	0.3	60
7	Minyak Goreng	0.9	150
8	Kursi	0.5	80
9	Rokok	0.8	120
10	Kasur	0.3	40
11	Vacuum Cleaner	0.6	110
12	Televisi	0.5	90
13	Lemari	0.2	30
14	Kaca	0.7	40
15	Sunscreen	0.6	100
16	Blender	0.4	70
17	Kipas Angin	0.4	90
18	Sofa	0.9	150
19	Serum	0.2	100
20	Karpet	0.8	130
21	Vas Bunga	0.5	80
22	Lampu Gantung	0.3	50
23	Printer	0.7	120

Dari tabel diatas akan diseleksi beberapa barang yang akan masuk ke dalam kontainer, dengan melakukan pemilihan barang dengan volume yang sesuai agar dapat dimasukkan ke dalam kontainer secara efektif.

ALGORITMA DYNAMIC PROGRAMING

```
def knapsack(container_volume, volumes,
            values):
    n = len(volumes)
    dp = [[0] * (container_volume + 1) for _ in range(n + 1)]
    for i in range(1, n + 1):
        for j in range(1, container_volume + 1):
            if volumes[i - 1] <= j:
```

```
        dp[i][j] = max(dp[i - 1][j], values[i - 1] + dp[i - 1][j - volumes[i
- 1]])else:
        dp[i][j] = dp[i - 1][j]
selected_items = []
total_value =
dp[n][container_volume]
remaining_volume =
container_volume
for i in range(n,
0, -1):if
total_value <=
0:
        break
if total_value != dp[i -
1][remaining_volume]:
        selected_items.append(i)
        total_value -= values[i - 1]
        remaining_volume -=
        volumes[i - 1]
return dp[n][container_volume], selected_items[::-1]

# Data barang dari tabel yang diberikan
volumes = [0.2, 0.5, 0.7, 0.4, 0.6, 0.3, 0.9, 0.5, 0.8, 0.3,
0.8, 0.6, 0.5, 0.2, 0.7, 0.4, 0.9, 0.6, 0.8, 0.5,
0.3, 0.7, 0.5, 0.2, 0.8]
values = [50, 70, 100, 80, 90, 60, 150, 80, 120, 50,
130, 110, 90, 30, 100, 70, 150, 100, 130, 80,
50, 120, 70, 40, 140]
```

```
container_volume = 2.5 # Kapasitas

kontainer# Menyelesaikan masalah

knapsack
max_value, selected_items = knapsack(int(container_volume * 10), [int(x * 10)
for x in volumes], values)
print("Total nilai barang yang bisa dimuat dalam kontainer:",
max_value)print("Barang yang dipilih:", selected_items)
```

Dari algoritma tersebut akan didapatkan daftar - daftar barang yang akan masuk kedalam kontainer dan akan dijabarkan di dalam tabel berikut.

Tabel Hasil Daftar Barang

NO	NAMA BARANG	VOLUME (m³)	HARGA (\$)
1	Panci	0.2	50
4	Cangkir	0.4	80
6	Meja	0.3	60
10	Kasur	0.3	40
12	Televisi	0.5	90
13	Lemari	0.2	30

KESIMPULAN DAN SARAN

Penelitian ini mengusulkan penggunaan pendekatan algoritma Dynamic Programming dalam mengatasi masalah optimasi pengisian kontainer, yang dikenal sebagai masalah knapsack. Masalah ini mengharuskan untuk memilih sejumlah barang dengan volume dan nilai tertentu untuk dimuat ke dalam kontainer dengan kapasitas yang telah ditetapkan. Dengan mengembangkan algoritma berbasis knapsack menggunakan teknik pemrograman dinamis, penelitian ini bertujuan untuk memaksimalkan pemanfaatan ruang kontainer dengan mempertimbangkan volume dan nilai barang yang tersedia.

Melalui studi kasus yang melibatkan 25 barang dengan karakteristik yang telah ditetapkan, memberikan kontribusi dalam mengatasi tantangan pengisian kontainer dalam

konteks industri logistik dan pengiriman. Sehingga hasil eksperimen menunjukkan bahwa algoritma yang diusulkan dapat meningkatkan pemanfaatan ruang kontainer secara signifikan dibandingkan dengan metode pengisian yang naif. Semoga penelitian ini dapat membantu menyelesaikan permasalahan knapsack dengan mudah dipahami.

DAFTAR PUSTAKA

- Kosasi, Sandi. "PENYELESAIAN BOUNDED KNAPSACK PROBLEM MENGGUNAKAN DYNAMIC PROGRAMMING." *Jurnal Informatika Mulawarman*, vol. 8, 2013.
- Rachmah, Jabal, et al. "Merancang Artificial Intelligence Karakter Pembeli Pada Wira Games Dengan Menggunakan Metode Knapsack Problem untuk Siswa Taman Kanak-Kanak." *e-Proceeding of Engineering*, vol. 10.
- Surjawan, Daniel Jahja, and Irene Susanto. "Aplikasi Optimalisasi Muat Barang Dengan Penerapan Algoritma Dynamic Programming Pada Persoalan Integer Knapsack." *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 1.

Memaksimalkan Pemanfaatan Ruang Kontainer dengan Pemrograman Dinamis: Pendekatan Berbasis Knapsack

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

/0

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7
