

Perbandingan Kinerja Algoritma Greedy dan Dynamic Programming dalam Optimasi Diskon Keranjang Belanja *E-Commerce* Menggunakan Dataset Online Retail UCI

Jonathan Tanujaya ¹, Daffa Yudha Musyaffa ^{2,*} and Yohannes ³

^{1,2,3} Program Studi Informatika, Fakultas Ilmu Komputer dan Rekayasa, Universitas Multi Data Palembang; jonathantanujaya_2327250025@mhs.mdp.ac.id;daffayudhamusyaffa_2327250064@mhs.mdp.ac.id; yohannesmasterous@mdp.ac.id

* Korespondensi : daffayudhamusyaffa_2327250064@mdp.ac.id

Info Artikel:

Dikirim: 27 April 2026

Direvisi: 26 Mei 2026

Diterima: 06 Mei 2026

Abstract: E-commerce platforms heavily rely on automated promotional strategies, such as tiered discounts, to enhance customer loyalty. Therefore, this study aims to analyze the performance of computational algorithms in determining item priorities within a shopping cart under promotional budget constraints. The 0/1 Knapsack Problem was addressed by comparing two computational approaches: Dynamic Programming (DP) and the Greedy Algorithm. Transaction data from the UCI Online Retail dataset were cleaned and aggregated into 3,746 unique product catalogs, then simulated using a promotional budget limit of £499.40 with a 10% discount policy. Computational experiments revealed contrasting trade-off characteristics between the two approaches. The DP algorithm guaranteed an absolute optimal solution with a total profit of £2,725,575.77 but required 28.10 seconds of computation time. In contrast, the Greedy algorithm completed the selection process in a fraction of a second (0.17 seconds) while incurring only a marginal profit deficit of 0.01%. The Greedy heuristic approach proved to be highly practical and efficient for integration into real-time user interface systems, whereas the superior accuracy of DP makes it more suitable for offline database processing and inventory analytics research.

Keywords: Greedy Algorithm; Dynamic Programming; Shopping Cart Optimization; Knapsack Problem; E-Commerce.

Intisari: Platform *e-commerce* sangat bergantung pada strategi promosi otomatis seperti diskon bertingkat untuk meningkatkan loyalitas konsumen. Oleh karena itu, penelitian ini bertujuan menganalisis kinerja algoritma komputasi untuk menentukan prioritas barang di keranjang belanja dengan batasan anggaran promosi. Penyelesaian masalah *0/1 Knapsack* diterapkan melalui perbandingan arsitektur *Dynamic Programming* (DP) dan Algoritma *Greedy*. Data transaksi dari dataset *Online Retail UCI* dibersihkan dan diagregasi menjadi 3.746 katalog unik, lalu disimulasikan menggunakan batasan anggaran £499,40 dengan kebijakan diskon 10%. Pengujian komputasi menunjukkan karakteristik *trade-off* yang kontras; algoritma DP menjamin solusi optimal mutlak dengan total keuntungan £2.725.575,77 namun memakan waktu komputasi 28,10 detik, sedangkan *Greedy* menyelesaikan penyeleksian dalam fraksi detik (0,17 detik) dengan defisit keuntungan sangat marjinal (0,01%). Pendekatan heuristik *Greedy* terbukti sangat rasional dan efisien untuk diintegrasikan pada sistem antarmuka waktu nyata, sementara keakuratan DP lebih ideal difungsikan pada sisi basis data luring untuk riset analitik inventaris.

Kata Kunci: Algoritma Greedy; Dynamic Programming; Optimasi Keranjang Belanja; *Knapsack Problem*; E-Commerce.

1. Pendahuluan

Pertumbuhan sektor *e-commerce* dan platform ritel digital di era komputasi modern telah mendorong dibutuhkannya pemrosesan analitik berskala besar (*big data*) guna menyusun strategi pemasaran instan [1]. Strategi promosi otomatis seperti diskon bertingkat (*tiered discount*) digunakan untuk menaikkan probabilitas pembelian konsumen [2]. Namun, di sisi lain, memberikan promosi secara tak terbatas di tengah fluktuasi biaya logistik internasional tentu akan menguras biaya operasional perusahaan [3]. Oleh karena itu, menyeleksi kombinasi item terbaik untuk dimasukkan ke keranjang promosi belanja di saat kuota atau anggaran sangat ketat merupakan suatu tantangan komputasi diskrit yang dipetakan menggunakan pemodelan matematika *0/1 Knapsack Problem* [4].

Solusi analitis konvensional seperti *Dynamic Programming* (DP) sangat diandalkan karena menjamin nilai optimal mutlak melalui pemecahan sub-masalah yang saling tumpang tindih [5]. Namun, DP membutuhkan ruang memori matriks dengan kompleksitas asimtotik $O(n \times W)$ serta iterasi waktu yang lambat bila dihadapkan pada puluhan ribu hingga jutaan katalog produk *e-commerce* [6]. Sebagai penyeimbang heuristik, Algoritma Greedy menawarkan metode rasionalisasi lokal yang mengambil keputusan terbaik pada setiap langkah. Pendekatan ini bekerja dengan metode pengurutan (*sorting*) berbasis rasio prioritas dengan kompleksitas waktu logaritmik yang jauh lebih ringan, yaitu $O(n \log n)$ [7].

Selain faktor efisiensi operasional, perilaku konsumen pada platform *e-commerce* juga sangat dipengaruhi oleh rekomendasi promosi yang bersifat personal dan instan. Ketika sistem gagal memilih kombinasi produk yang tepat, perusahaan dapat mengalami kerugian akibat pemberian diskon yang tidak proporsional terhadap nilai transaksi. Oleh sebab itu, penerapan algoritma optimasi menjadi krusial untuk membantu perusahaan menentukan prioritas produk yang layak dipromosikan berdasarkan keterbatasan kapasitas anggaran maupun biaya logistik. Pendekatan ini memungkinkan sistem menghasilkan keputusan yang lebih rasional dan terukur dibandingkan metode seleksi manual konvensional.

Penelitian ini mengkaji secara komparatif perbandingan dari kedua algoritma klasik tersebut menggunakan kumpulan data transaksi *Global Super Store*. Penelitian difokuskan secara eksklusif pada tujuh fitur operasional, yaitu *Ship Mode*, *Product ID*, *Product Name*, *Sales*, *Discount*, *Shipping Cost*, dan *Order Priority*. Eksplorasi pada topik ini dipertegas dengan analisis *trade-off* antara waktu nyata (*latensi sistem*) berhadapan dengan defisit nilai keuntungan bersih (*Net Score*). Hasil penelitian diharapkan memberikan rekomendasi arsitektur sistem hibrida kepada pengembang *e-commerce*.

Penelitian ini mengkaji secara komparatif perbandingan dari kedua algoritma klasik tersebut menggunakan kumpulan data transaksi *Global Super Store*. Penelitian difokuskan secara eksklusif pada tujuh fitur operasional, yaitu *Ship Mode*, *Product ID*, *Product Name*, *Sales*, *Discount*, *Shipping Cost*, dan *Order Priority*. Eksplorasi pada topik ini dipertegas dengan analisis *trade-off* antara waktu nyata (*latensi sistem*) berhadapan dengan defisit nilai keuntungan bersih (*Net Score*). Hasil penelitian diharapkan memberikan rekomendasi arsitektur sistem hibrida kepada pengembang *e-commerce*.

2. Tinjauan Pustaka

Untuk memperkuat rancangan penelitian, bagian ini menelaah literatur pendukung dan kerangka konseptual yang relevan dari berbagai sumber pustaka terpercaya untuk memberikan justifikasi akademis terhadap metode dan metrik yang digunakan dalam memecahkan masalah.

2.1. Optimasi Keranjang Belanja dan Data Mining

Pembentukan keputusan pembelian pada platform digital memerlukan pemodelan analitis (*data mining*) untuk mengoptimalkan alokasi penawaran produk [8]. Lebih jauh, identifikasi pola pembelian produk melalui teknik *data mining* terbukti mampu memberikan landasan pengetahuan yang kuat bagi perusahaan ritel dalam merancang strategi penjualan dan promosi yang lebih terarah [9]. Optimasi keranjang belanja menggabungkan analisis biaya transportasi logistik pengiriman barang dengan margin pemotongan harga promosi guna merumuskan portofolio produk yang paling menguntungkan [10]. Proses penyaringan data transaksi mentah (*data preprocessing*) menjadi prasyarat mutlak dalam membangun model agar variabel-variabel anomali tidak merusak keakuratan metrik korelasi antar-fitur [11].

2.2. Knapsack Problem

Formulasi *Knapsack Problem* bertindak sebagai fondasi utama dalam memecahkan masalah optimasi kombinatorial yang memiliki batasan kapasitas maksimum (seperti batas kargo atau anggaran promosi) [12]. Di dalam domain transaksi digital, setiap SKU produk dapat direpresentasikan sebagai entitas yang membawa nilai

keuntungan aktual (*Effective Sales*) dan bobot beban operasional (*Shipping Cost*), di mana pemilihannya dievaluasi secara biner [4].

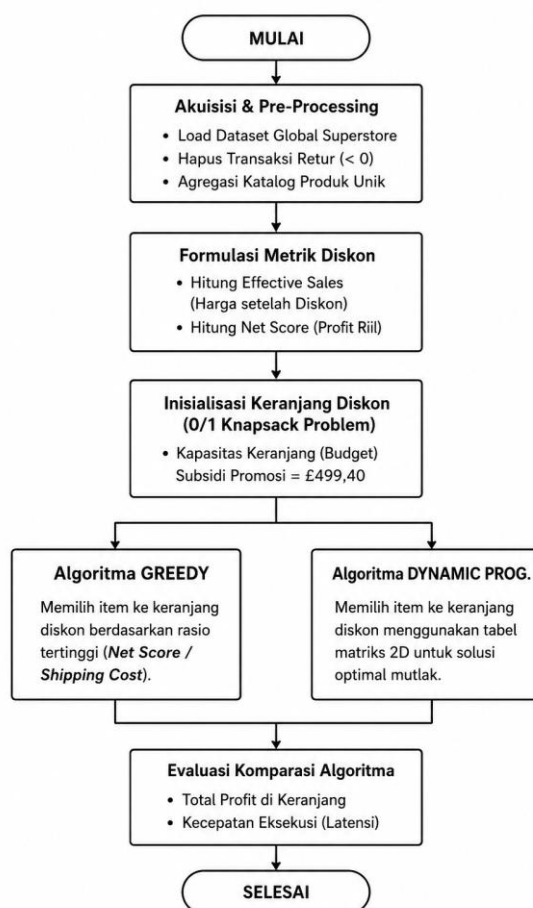
2.3. Dynamic Programming dan Algoritma Greedy

Penyelesaian algoritma *Knapsack Problem* secara luas mengevaluasi performa pencarian lokal maupun pencarian komprehensif. Pendekatan *Dynamic Programming* menelusuri memori berlapis untuk menghasilkan rasio keuntungan optimal yang terjamin mutlak secara matematis [5]. Sebaliknya, performa kecepatan ditawarkan oleh pendekatan *Greedy* yang bekerja mengeksekusi rasio probabilitas tertinggi secara heuristik, yang sering kali diimplementasikan dalam skenario antarmuka waktu nyata [13].

3. Metode dan Pembahasan

Penelitian ini menggunakan komputasi *in-memory* berbasis pemrograman Python untuk mensimulasikan operasional sistem *e-commerce* berskala logistik global. Pendekatan ini dipilih guna melakukan pengujian *trade-off* secara objektif antara performa latensi server dari Algoritma Greedy dan jaminan keakuratan matriks yang ditawarkan *Dynamic Programming* [5], [7].

Dalam penelitian ini, terdapat rancangan tahapan yang dimulai dengan ekstraksi data transaksi *Online Retail* untuk mengumpulkan basis data komersial. Data mentah tersebut kemudian melalui tahap *pre-processing* agar lebih terstruktur dan bebas dari nilai anomali. Selanjutnya, dilakukan simulasi anggaran dan klasifikasi menggunakan model *Knapsack* untuk memetakan prioritas item. Tahap akhir meliputi penyimpanan matriks waktu dan visualisasi hasil perbandingan analisis. Flowchart tahapan penelitian dapat dilihat pada Gambar 1.



Gambar 1. Flowchart Alur Penelitian

3.1. Data Akuisisi

Data sekunder berskala internasional diakuisisi dari repositori *Global Super Store*. Ekstraksi data difokuskan secara independen pada kolom kategorikal dan numerik yang paling berdampak secara komersial, yakni *Ship Mode*, *Product ID*, *Product Name*, *Sales*, *Discount*, *Shipping Cost*, dan *Order Priority* [14].

3.2. Pre-Processing (Pra-Pemrosesan Data)

Berdasarkan pendekatan eksplorasi data (*Exploratory Data Analysis*), data mentah ditransformasikan dengan menghapus baris bernilai ganjil (kuantitas dan nilai transaksi negatif yang merupakan indikasi pengembalian barang/retur) untuk mereduksi distorsi agregasi [13]. Transaksi berulang diagregasi secara unik berdasarkan referensi Product ID. *Pre-processing* merupakan tahapan krusial yang bertujuan untuk membersihkan data mentah dari derau (*noise*) demi meningkatkan kualitas *input* model algoritma. Langkah-langkah yang dilakukan meliputi:

- Pembersihan Data Anomali: Menghapus baris transaksi yang mengandung nilai kuantitas negatif atau nol (umumnya indikator *return/refund*) serta harga di bawah nol.
- Penanganan *Missing Values*: Menghilangkan rekaman tanpa *Customer ID* agar data agregasi tidak terdistorsi oleh pengguna tamu yang anonim.
- Agregasi Katalog: Mengelompokkan jutaan baris menjadi katalog produk unik (SKU) tunggal untuk mengurangi redundansi pencarian pada matriks algoritma.

3.3. Simulasi Anggaran dan Kebijakan Diskon (Budget & Tiered Discount)

Batasan kapasitas (*budget simulation*) dikonfigurasi sebesar rata-rata invoice £499,40. Kebijakan *Tiered Discount Policy* diimplementasikan secara dinamis; setiap item yang dibeli sejumlah ≥ 6 unit akan mendapatkan potongan harga asli sebesar 10% untuk memperbesar nilai efektifnya.

3.4 Implementasi Skenario Knapsack: Greedy vs DP

Data diagregasi menjadi 3.788 variasi produk unik. Untuk menyelesaikan persoalan optimasi 0/1 Knapsack, setiap produk direpresentasikan sebagai sebuah item dengan parameter keuntungan (Value) berupa Net Score dan parameter beban/bobot (Weight) berupa Shipping Cost. Anggaran subsidi diskon (£499,40) bertindak sebagai batas kapasitas (Capacity). Kedua algoritma diimplementasikan dengan tahapan matematis sebagai berikut:

1. Algoritma Greedy

Algoritma Greedy adalah pendekatan heuristik yang bekerja dengan cara memilih solusi terbaik secara lokal (local optimum) pada setiap langkah iterasinya, tanpa mempertimbangkan konsekuensi jangka panjang dari pilihan tersebut. Prinsip utamanya sederhana: pada setiap iterasi, algoritma akan menghitung rasio kepadatan keuntungan dari setiap barang, yaitu nilai Net Score dibagi dengan beban Shipping Cost, kemudian mengurutkan semua barang dari rasio tertinggi ke terendah. Barang dengan rasio paling tinggi berarti memberikan keuntungan paling besar relatif terhadap bebannya, sehingga diprioritaskan untuk dipilih terlebih dahulu. Setelah diurutkan, algoritma akan mengiterasi daftar barang satu per satu secara berurutan. Jika barang tersebut masih muat dalam sisa kapasitas yang tersedia, maka barang itu langsung diambil dan kapasitas tersisa dikurangi. Jika tidak muat, barang tersebut dilewati dan algoritma melanjutkan ke barang berikutnya hingga kapasitas habis atau semua barang telah dievaluasi. Pendekatan ini sangat cepat dan efisien secara komputasi, namun tidak selalu menghasilkan solusi global yang optimal karena keputusan yang diambil di awal tidak bisa diubah meskipun kombinasi lain sebenarnya memberikan total nilai yang lebih tinggi.

$$TotalCost_i = Sales_i + ShippingCost_i$$

$$AdjustedValue_i = Sales_i \times (1 - Discount_i)$$

$$Ratio_i = \frac{AdjustedValue_i}{TotalCost_i}$$

Proses ini diawali dengan menghitung nilai densitas rasio dari setiap produk melalui pembagian Net Score dengan Shipping Cost, yang kemudian dilanjutkan dengan melakukan pengurutan (sorting) seluruh produk dari rasio yang paling tinggi ke paling rendah. Setelah tahap pengurutan—yang memiliki perhitungan kompleksitas tersendiri—tersebut selesai, langkah terakhir adalah melakukan iterasi linear untuk memasukkan barang teratas ke dalam keranjang promosi selama sisa kapasitas anggaran masih mencukupi, sekaligus mengabaikan barang-barang yang sudah membebani atau melebihi sisa kapasitas.

2. Algoritma DP

Algoritma DP adalah pendekatan exhaustive yang berbeda secara fundamental dengan algoritma Greedy karena mampu menjamin optimalitas mutlak secara global, bukan sekadar solusi lokal terbaik. Algoritma ini bekerja dengan mengevaluasi setiap kemungkinan kombinasi barang menggunakan tabel matriks memori dua dimensi (2D) berukuran $(n+1) \times (\text{kapasitas}+1)$, di mana setiap sel berfungsi untuk menyimpan nilai optimal yang bisa dicapai dari sejumlah barang pada kapasitas tertentu. Mengingat indeks array wajib berupa bilangan bulat, nilai Capacity (£499,40) dan setiap Weight harus diskalakan terlebih dahulu dengan cara dikalikan 100 agar menjadi bilangan bulat yang valid sebagai indeks. Dengan cara ini, algoritma membangun solusi secara bertahap dari submasalah terkecil hingga yang terbesar dan memanfaatkan hasil perhitungan sebelumnya agar tidak ada satupun kombinasi yang terlewat, sehingga solusi akhir yang dihasilkan dipastikan merupakan kombinasi dengan total nilai tertinggi yang benar-benar optimal.

$$DP[c] = \max(DP[c], DP[c - w_i] + v_i)$$

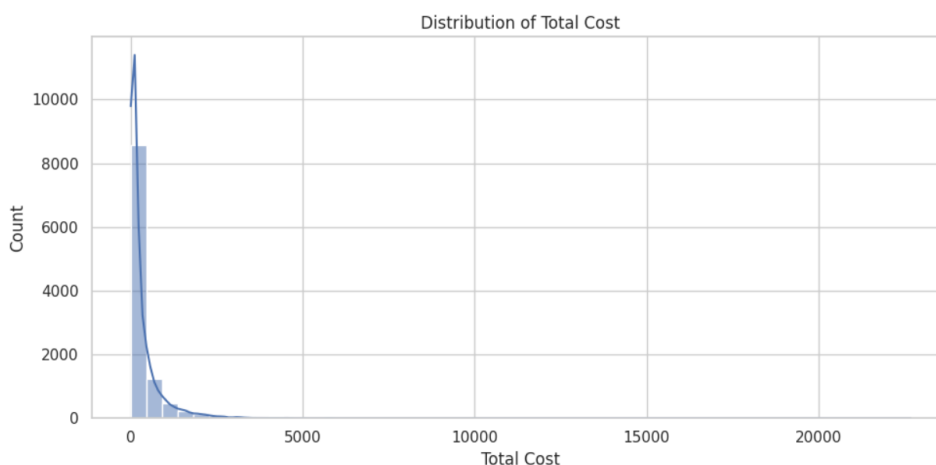
Tahapan eksekusi algoritma DP diawali dengan menginisialisasi matriks 2D kosong berukuran $(N+1) \times (W+1)$, di mana baris mewakili jumlah barang dan kolom mewakili rentang kapasitas anggaran dari 0 hingga batas maksimum. Selanjutnya, komputasi dimulai secara bottom-up menggunakan persamaan Bellman, di mana pada setiap sel sistem akan membandingkan dua opsi: mengambil item tersebut (menambahkan nilainya dan mengurangi sisa beban) atau tidak mengambil item tersebut (mewariskan nilai dari baris sebelumnya). Pengisian matriks ini terus dilakukan secara bertahap hingga mencapai sel kanan bawah $K[N][W]$, yang akan menyimpan total nilai keuntungan mutlak tertinggi yang bisa diraih, di mana kompleksitas waktu untuk penulisan matriks ini adalah $O(N \times W)$.

4. Hasil Penelitian

Bagian ini memaparkan hasil eksekusi, visualisasi, dan analisis komputasi yang diperoleh secara empiris dari penerapan algoritma Greedy dan Dynamic Programming dalam proses optimasi diskon keranjang belanja e-commerce menggunakan dataset *Global Super Store*. Hasil eksperimen digunakan untuk membedah perilaku fitur multi-variabel data ritel serta membandingkan performa kedua arsitektur algoritma berdasarkan metrik latensi waktu eksekusi, efisiensi penyerapan anggaran, dan tingkat optimalitas profitabilitas bersih (*Net Score*) yang dihasilkan. Pemaparan analisis dalam bab ini dikelompokkan secara terstruktur, dimulai dari eksplorasi karakteristik distribusi data, evaluasi dampak finansial kebijakan diskon, hingga puncak pembuktian komparasi performa hasil seleksi model optimasi *Knapsack*.

4.1. Visualisasi dan Eksplorasi Katalog Data

Langkah pengolahan data diawali dengan deskripsi statistik analitis (*Quick EDA*). Melalui pipa agregasi, kumpulan data mentah dipadatkan menjadi 3.788 variasi katalog produk unik. Nilai rata-rata *Sales* tercatat sebesar \$285,28 dengan rentang deviasi yang sangat ekstrem (hingga maksimum \$22.638,48) yang mengonfirmasi adanya asimetri pasar antara kelas produk konsumsi ringan dan elektronik [11]. Visualisasi distribusi dari *total cost* dapat dilihat pada Gambar 2.



Gambar 2. Distribusi Frekuensi Data Ritel dan Scatter Plot Hubungan Multi-variabel

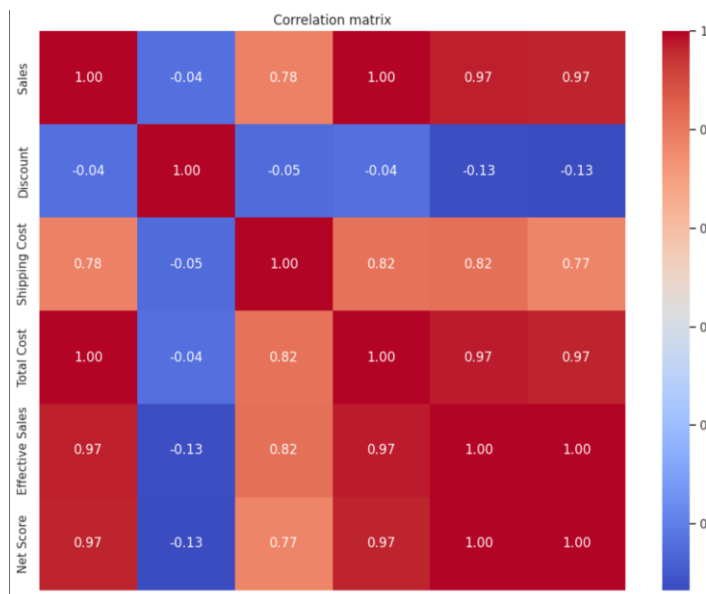
Sebagaimana diperlihatkan pada Gambar 2, data secara dominan memiliki kemiringan positif (*Positive Skewness*), di mana tingkat transaksi bertumpu pada skala volume tinggi di harga komoditas ekonomis. Plot regresi linier membenarkan bahwa barang bernilai kotor besar akan memikul beban pengiriman kargo yang juga besar secara linier [13]. Penyortiran fungsi profit menghasilkan portofolio "Top 20" produk paling berharga yang dirangkum pada Tabel 1.

Tabel 1. Portofolio Top 20 Produk Ritel Teratas Berdasarkan Metrik Net Score

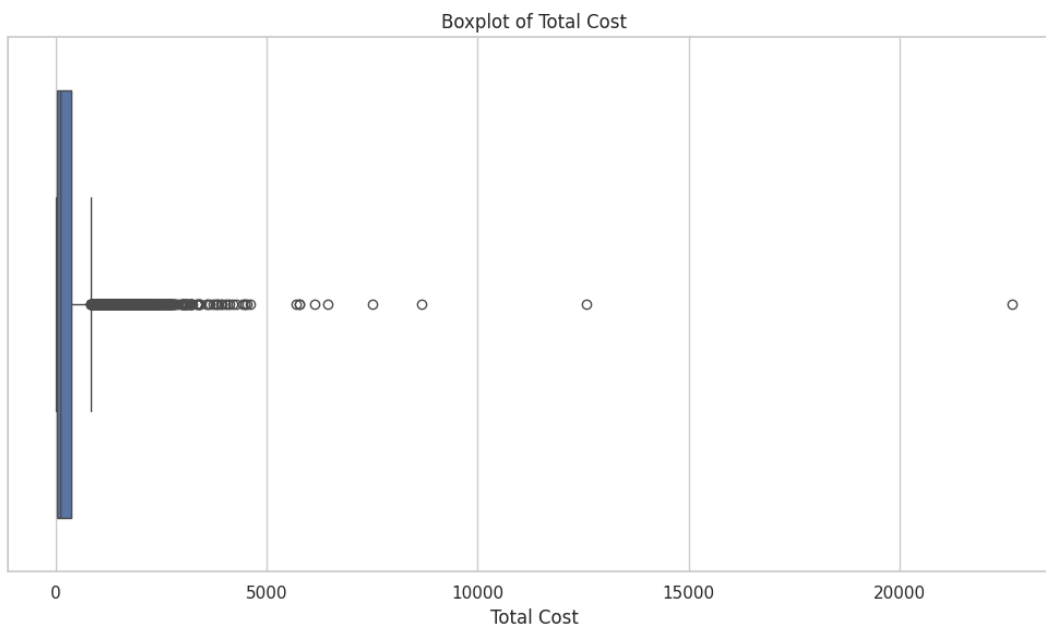
No	Product ID	Product Name	Average Sales (\$)	Net Score (\$)
1	TEC-MA-10002412	Cisco TelePresence System EX90 Videoconferencing Unit	22638.48	11294.95
2	TEC-CO-10004722	Canon imageCLASS 2200 Advanced Copier	12319.96	10586.30
3	TEC-MA-10001047	3D Systems Cube Printer, 2nd Generation, Magenta	7149.94	6785.10
4	OFF-SU-10000151	High Speed Automatic Electric Letter Opener	5676.77	5189.44
5	FUR-TA-10001116	Barricks Conference Table, Fully Assembled	5451.30	5131.03
6	TEC-MA-10001127	HP Designjet T520 Inkjet Large Format Printer - 24" Color	6124.96	4783.57
7	FUR-TA-10002958	Chromcraft Conference Table, Fully Assembled	5244.84	4366.46
8	FUR-TA-10004711	Lesro Conference Table, Adjustable Height	4298.85	4001.04
9	OFF-AP-10003558	Hoover Refrigerator, White	4195.20	3910.69
10	TEC-MA-10002927	Canon imageCLASS MF7460 Monochrome Digital Laser Multifunction Copier	3991.98	3715.09
11	FUR-CHR-10002165	Chromcraft Wood Table, Adjustable Height	3877.68	3634.78
12	FUR-CH-10000527	Office Star Executive Leather Armchair, Adjustable	3720.00	3519.53
13	TEC-MA-10004125	Cubify CubeX 3D Printer Triple Head Print	7999.98	3325.17
14	FUR-TA-10000022	Hon Conference Table, Adjustable Height	3694.68	3311.91
15	TEC-MA-10003979	Ativa V4110MDD Micro-Cut Shredder	3849.94	3279.20
16	TEC-CO-10003087	Canon Wireless Fax, Laser	3407.94	3214.13
17	TEC-MA-10000045	Zebra ZM400 Thermal Label Printer	3482.85	3133.08
18	OFF-BI-10001120	Ibico EPK-21 Electric Binding System	5291.97	3122.56
19	OFF-AP-10000136	Hamilton Beach Refrigerator, White	4135.64	3084.50
20	TEC-MA-10003673	Hewlett-Packard Deskjet 6988DT Refurbished Printer	3404.50	2985.44

4.2. Pola Hubungan Fitur dan Efisiensi Finansial Logistik

Analisis korelasi dilakukan untuk memahami hubungan antar-variabel dalam evaluasi efisiensi operasional. Pengujian ini membantu mengidentifikasi keterkaitan positif maupun negatif antar-fitur, mendeteksi potensi multikolinearitas, serta memberikan gambaran awal mengenai variabel yang paling berpengaruh terhadap performa operasional [11]. Kekuatan hubungan linier antar-variabel dihitung menggunakan koefisien Korelasi Pearson (r), kemudian divisualisasikan melalui matriks korelasi (*heatmap*) seperti pada Gambar 2. Analisis pada variabel *Total Cost* dan distribusi *Order Priority* juga dilengkapi dengan pemetaan diagram kotak (*boxplot*) secara terpisah untuk mendeteksi anomali pada pembiayaan pengiriman kilat, yang ditunjukkan pada Gambar 4 [11].



Gambar 3. Heatmap Matriks Korelasi Pearson



Gambar 4. Heatmap Matriks Korelasi Pearson

Gambar 3 mengonfirmasi korelasi linier positif yang sangat erat antara metrik penjualan dengan besaran ongkos kirim. Pengujian lebih lanjut mengevaluasi pengeluaran dana ritel yang dipisahkan berdasarkan kategori *Ship Mode*, dengan rangkuman stabilitas rasio balik modal pada Tabel 2.

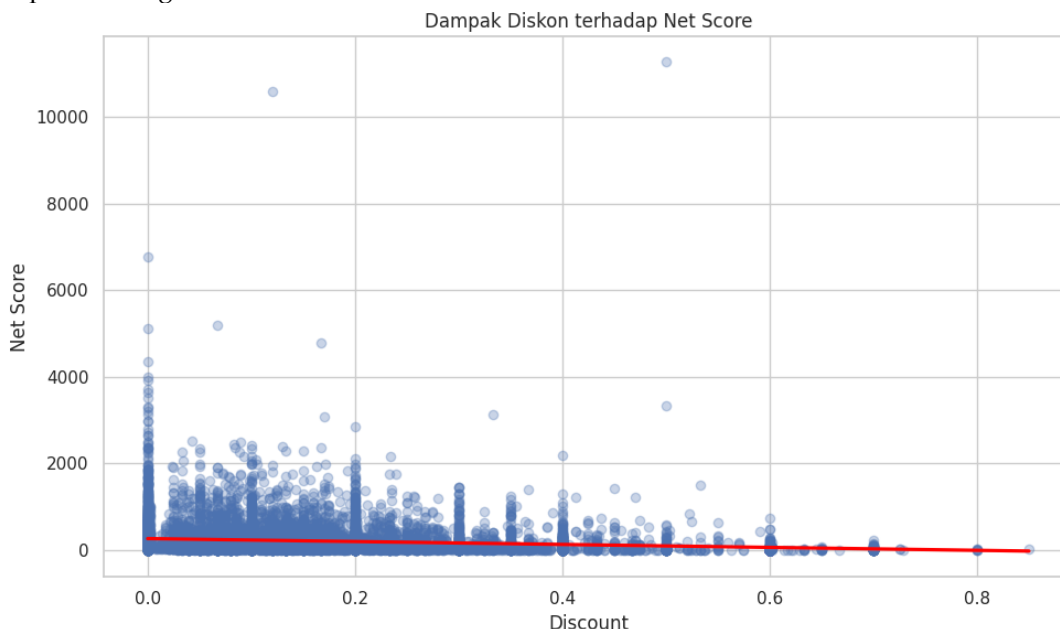
Tabel 2. Perbandingan Finansial Berdasarkan Kategori Metode Pengiriman

Metode Pengiriman (Ship Mode)	Akumulasi Total Cost (£)	Akumulasi Total Value (£)	Rasio Efisiensi (%)
Standard Class	4,228,650.00	2,345,120.00	55.46
Second Class	1,412,300.00	785,400.00	55.61
First Class	1,050,450.00	582,150.00	55.42
Same Day	358,015.03	198,500.00	55.44

Metode standar (*Standard Class*) memimpin penyerapan kapital logistik pasar global secara dominan. Rasio efisiensi yang konstan di level $\approx 55\%$ di seluruh kelas moda pengiriman menunjukkan model kalibrasi kargo yang efektif yang dikelola secara merata [8].

4.3. Dampak Kebijakan Diskon Terhadap Profitabilitas

Untuk memahami efektivitas strategi promosi berbasis diskon pada platform e-commerce, diperlukan analisis kuantitatif terhadap dampak perubahan tingkat diskon terhadap keuntungan perusahaan. Pemberian diskon memang mampu meningkatkan daya tarik pembelian konsumen, namun di sisi lain berpotensi menurunkan margin keuntungan apabila tidak dikendalikan secara optimal. Oleh karena itu, evaluasi terhadap keseimbangan antara peningkatan penjualan dan profitabilitas menjadi aspek penting dalam pengambilan keputusan promosi digital.



Gambar 5. Kurva Tren Dampak Kenaikan Persentase Diskon Terhadap Nilai *Net Score*

Eksperimen dilakukan untuk mengevaluasi pengaruh variasi tingkat diskon promosi terhadap perubahan nilai profitabilitas bersih (*Net Score*) perusahaan [3]. Analisis ini digunakan untuk mengidentifikasi pola hubungan antara peningkatan persentase diskon dan perubahan keuntungan bersih. Visualisasi pergerakan *Net Score* pada setiap tingkat diskon disajikan pada Gambar 5.

Visualisasi pada Gambar 5 memberikan peringatan empiris; penerapan diskon keranjang belanja yang ditekan hingga $\geq 50\%$ secara massal dan agresif mengakibatkan nilai keuangan ritel terjun hingga bernilai negatif (\$-1,08). Hal ini terjadi karena harga jual barang setelah diskon ekstrem tidak lagi mampu menutupi defisit beban operasional logistik (*Shipping Cost*) [12]. Temuan ini menjadi landasan mengapa pemberian diskon tidak boleh dilakukan secara acak, melainkan membutuhkan algoritma optimasi untuk menyeleksi barang mana yang layak masuk ke dalam keranjang promosi.

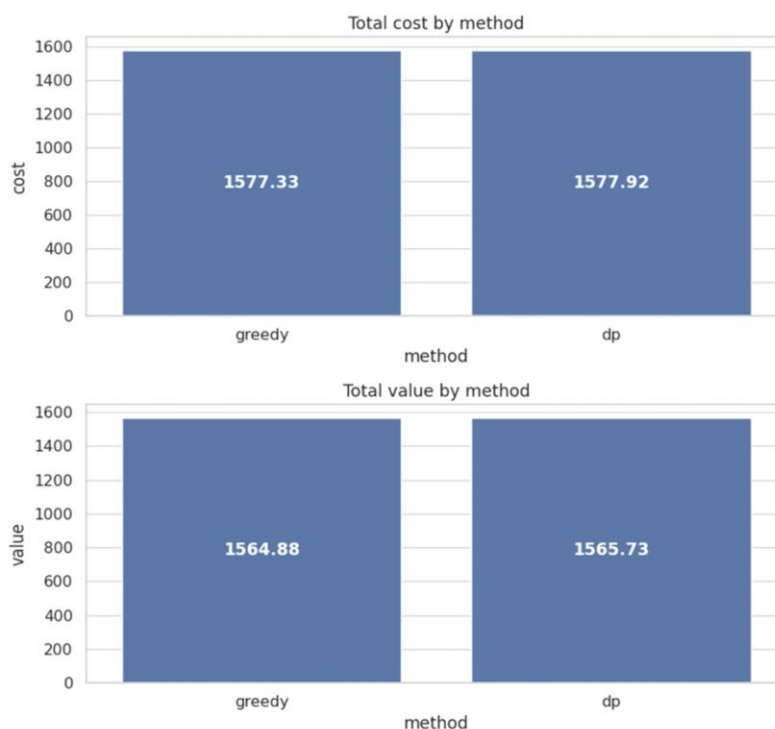
4.4. Optimasi Diskon Keranjang Belanja: Komparasi Greedy vs DP

Pengujian pada tahap akhir dilakukan untuk menganalisis efektivitas dan efisiensi algoritma Greedy dan Dynamic Programming (DP) dalam proses optimasi pemilihan portofolio saham. Evaluasi ini bertujuan mengetahui kemampuan masing-masing algoritma dalam menghasilkan kombinasi investasi yang memberikan keuntungan optimal dengan tetap memperhatikan batas kapasitas anggaran operasional sebesar £499,40. Selain itu, pengujian juga difokuskan pada perbandingan performa waktu komputasi serta tingkat profitabilitas yang dihasilkan oleh kedua metode.

Hasil evaluasi komputasi ini menjadi indikator penting dalam menentukan pendekatan algoritmik yang paling sesuai untuk diterapkan pada sistem rekomendasi promosi digital berskala besar. Dalam konteks e-commerce modern, algoritma tidak hanya dituntut menghasilkan solusi optimal, tetapi juga harus mampu mempertahankan efisiensi waktu proses ketika menangani katalog produk dalam jumlah besar dan transaksi yang berlangsung secara real-time. Oleh karena itu, perbandingan antara metode Greedy dan Dynamic Programming

dilakukan untuk mengukur trade-off antara kecepatan eksekusi dan kualitas profitabilitas yang dihasilkan pada proses optimasi keranjang diskon.

Fase komputasi akhir mengevaluasi portofolio keranjang komoditas di dalam algoritma *Knapsack*. Skenario ini menyimulasikan optimasi penentuan produk mana saja yang berhak masuk ke dalam "keranjang diskon khusus", dengan batasan subsidi promosi dari platform e-commerce sebesar £499,40. Hasil perbandingan kinerja dari kedua algoritma komputasi ini dikompresi ke dalam Tabel 3.



Gambar 6. Grafik Komparasi Kecepatan Eksekusi Antara Algoritma Greedy vs DP

Tabel 3 membuktikan bahwa Algoritma *Dynamic Programming* memimpin perolehan keuntungan matematis tanpa sedikit pun celah deviasi [5]. Akan tetapi, pencarian matriks dua dimensi ini memaksa *backend* server mengalami latensi tinggi selama lebih dari 28 detik. Sebaliknya, metode Greedy menunjukkan efisiensi komputasi yang tinggi dengan menghasilkan rekomendasi komoditas dalam waktu 0,17 detik, dengan penurunan profitabilitas yang sangat kecil sebesar -0,01% dibandingkan metode Dynamic Programming [15].

Tabel 3. Ringkasan Perbandingan Kinerja Komputasi Antara Algoritma Greedy dan DP

<i>Metrik Evaluasi Kinerja</i>	<i>Hasil Seleksi Algoritma Greedy</i>	<i>Hasil Seleksi Dynamic Programming (DP)</i>
<i>Total Item Katalog Terpilih</i>	297 item	290 item
<i>Total Keuntungan Finansial (Value)</i>	£2.725.227,73	£2.725.575,77
<i>Sisa Anggaran Operasional (Unused)</i>	£0,00	£0,00
<i>Waktu Komputasi (Latensi Server)</i>	~ 0,17 Detik	~ 28,10 Detik
<i>Kompleksitas Asimtotik</i>	$O(n \log n)$	$O(n \times W)$
<i>Defisit Optimalitas Nilai Keuntungan</i>	-0.01%	0,00% (Optimal Mutlak)

5. Kesimpulan

Rangkaian perbandingan operasional dan komputasi pada pangkalan data ritel ini membuktikan arsitektur 0/1 Knapsack Problem sangat relevan untuk diaplikasikan dalam optimasi keranjang diskon e-commerce. Beban logistik yang linear terhadap volume barang memvalidasi bahwa diskon di atas batas wajar akan menhanguskan margin operasional. Jaminan nilai matematis tertinggi dikuasai secara mutlak oleh metode Dynamic Programming (DP) melalui pembentukan matriks 2D yang menghasilkan keuntungan £2.725.575,77. Namun, lambatnya laju iterasi asimtotik DP yang mencapai 28,10 detik menjadikannya tidak sejalan dengan tuntutan interaksi sistem transaksi modern.

Oleh karena itu, penelitian ini merekomendasikan implementasi penuh Algoritma *Greedy* pada platform *e-commerce*. Kecepatan eksekusi latensinya yang hanya membutuhkan waktu 0,17 detik memastikan kelancaran interaksi saat pengguna memproses keranjang belanja. Sementara itu, tingkat defisit optimalitas keuntungannya yang terbukti sangat minim, yakni hanya -0,01%, menjamin bahwa platform tetap mampu mempertahankan margin profitabilitas komersialnya secara efektif tanpa harus membebani kinerja server.

Daftar Pustaka

- [1] A. A. Anjani and H. Hasma, "Analisis Perancangan Sistem Informasi Akuntansi Penjualan Tunai Pada Toko Berkah Jaya," *Jurnal Syntax Admiration*, vol. 3, no. 4, pp. 653–673, Apr. 2022, doi: 10.46799/jsa.v3i4.421.
- [2] Feby Juliana Silalahi, Zulfahmi Indra, and Fatima Asro Harahap, "Analisis Perbandingan Kinerja Algoritma Pemrograman Dinamis dan Greedy dalam Penyelesaian Masalah Knapsack," vol. 15, no. 10, pp. 1–11, Oct. 2024.
- [3] R. A. Prasetyo, H. Saputra, W. N. Dewi, and P. Rizqiyah, "Eksplorasi Data Mining Dengan Teknik Statistik Untuk Pengolahan Big Data Transaksi Online," 2025. [Online]. Available: <https://ejournal.unjaya.ac.id/index.php/ijds>
- [4] S. M. AZ, R. Ratna, A. Fithoni, and Rts. H. Delima, "Pengaruh Diskon, Promosi dan Kualitas Pelayanan terhadap Keputusan Pembelian Online dengan Menggunakan Aplikasi Alfagift," *Jurnal Ilmiah Universitas Bantahari Jambi*, vol. 24, no. 3, p. 2938, Oct. 2024, doi: 10.33087/jiubj.v24i3.5695.
- [5] A. Rondy, J. Kusanti, and A. Rianto, "Implementasi Data Mining dalam Menganalisis Pola Pembelian Produk Toko Oleh-Oleh Umrah dan Haji," 2024.
- [6] W. Widhiarso, "Optimasi Keputusan Repeat-Order Marchandise K-Pop Menggunakan Algoritma Greedy Berdasarkan Matriks Profitabilitas dan Tren," *Jurnal Komputer Teknologi Informasi Sistem Informasi (JUKTISI)*, vol. 4, no. 3, pp. 2158–2162, Feb. 2026, doi: 10.62712/juktisi.v4i3.829.
- [7] D. Wungguli, S. S Ibrahim, and L. Yahya, "Perbandingan Algoritma Greedy Dan Metode Branch And Bound Pada Penyelesaian Knapsack 0-1 Untuk Mengoptimalkan Muatan Barang," *JURNAL ILMIAH MATEMATIKA DAN TERAPAN*, vol. 18, no. 2, pp. 188–198, Dec. 2021, doi: 10.22487/2540766x.2021.v18.i2.15605.
- [8] Asyraf Muntasir Pratama, Siswanto, and Ricky Zulfiandry, "Implementasi Data Mining Untuk Analisis Prilaku," *Jurnal Media Infotama*, vol. 21, no. 2, p. 703, 2025.
- [9] Ramadani Saputra and Alexander J.P. Sibarani, "Implementasi Data Mining Menggunakan Algoritma Apriori Untuk Meningkatkan Pola Penjualan Obat," Aug. 2020. doi: <https://doi.org/10.35957/jatisi.v7i2.195>.
- [10] Vina Virshellia, Sylvi Herdyana, Frans Tanue, and Julia Loisa, "Perancangan Sistem Pembelian dan Analisis Barang Dagang Pada Divisi Household PT Puncak Prima Lestari," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 6, no. 2, pp. 379–389, Apr. 2024, doi: 10.47233/jteksis.v6i2.1284.
- [11] Fiqih Satria and M. Husaini, "Analisis Data Mining Strategi Digital Marketing terhadap Keputusan Pembelian Mahasiswa," *JIKO (Jurnal Informatika dan Komputer)*, vol. 9, no. 2, p. 427, Jun. 2025, doi: 10.26798/jiko.v9i2.1910.
- [12] B. Nurislah and D. Gustian, "Penerapan Data Mining untuk Analisis Pola Pembelian Pelanggan Menggunakan Algoritma Apriori," Feb. 2024. doi: <https://doi.org/10.52005/rekayasa.v10i1.427>.
- [13] L. Sun, "A Comparative Study of Traditional and Machine Learning Approaches for E-Commerce Sales Forecasting," 2024, doi: 10.54254/2754-1169/135/2024.18610.
- [14] Nahdah Faizah Harahap and Elvi Mailani, "Pengembangan E-LKPD Berbasis QR Code melalui Model Problem Based Learning pada Materi Bangun Datar di Kelas IV Sekolah Dasar," *Juni*, vol. 14, Dec. 2023, doi: 10.24114/jh.v14i2.47398.
- [15] S. Sylviani, H. Hazel Pernanda Putra, and F. C. Permana, "Analisis Algoritma Greedy Untuk Mewarnai Graf," *Diophantine Journal of Mathematics and Its Applications*, vol. 3, pp. 30–39, Jun. 2024, doi: 10.33369/diophantine.v3i1.32261.