
Optimasi Strategi Repeat Buyer pada E-commerce Indonesia Melalui Pendekatan Dynamic Programming untuk Bundling Product Multi-Kategori

Siti Fatimah Az Zahrah¹⁾ *, Yeremia Agung Chandra²⁾ & Yohannes³⁾

^{1,2,3)}Prodi Informatika, Fakultas Ilmu Komputer dan Rekayasa,
Universitas Multi Data Palembang, Indonesia

*Corresponding Email: ¹⁾sitifatimahazzahrah_2327250055@mhs.mdp.ac.id,
²⁾yeremiaagungchandra_2327250032@mhs.mdp.ac.id, ³⁾yohannesmasterous@mdp.ac.id

Abstrak

Penelitian ini bertujuan untuk mengoptimalkan strategi peningkatan *repeat buyer* pada *e-commerce* di Indonesia melalui penyusunan rekomendasi *bundling product* multi-kategori berbasis pendekatan komputasional. Pendekatan yang digunakan adalah Dynamic Programming melalui model optimasi Knapsack yang dikombinasikan dengan analisis Threshold Standard Deviation untuk menyaring kategori produk berdasarkan kedekatan demografis pelanggan. Proses penelitian meliputi tahap *preprocessing* data, pemodelan parameter bobot dan profit, optimasi kombinatorial, serta penentuan prioritas rekomendasi berbasis *customer profiling*. Hasil penelitian menunjukkan bahwa sistem mampu menghasilkan rekomendasi *bundling* yang relevan dan terpersonalisasi berdasarkan usia dan riwayat transaksi pelanggan. Dynamic Programming menunjukkan performa yang lebih stabil dan efisien pada kompleksitas data yang lebih tinggi, meskipun pada dataset kecil Brute Force memiliki waktu eksekusi lebih cepat. Secara keseluruhan, pendekatan yang diusulkan dinilai mampu meningkatkan akurasi rekomendasi serta mendukung strategi pemasaran untuk mendorong loyalitas pelanggan.

Kata Kunci: Rekomendasi *Bundling*, Dynamic Programming, Knapsack Bottom-Up, *E-commerce*, *Repeat Buyer*.

Abstract

The study aims to optimize strategies for increasing repeat buyers in Indonesian e-commerce by generating multi-category product bundling recommendations using a computational approach. The method applied is Dynamic Programming through a Knapsack optimization model combined with Threshold Standard Deviation analysis to filter product categories based on customer demographic proximity. The research process includes data preprocessing, parameter modeling of weight and profit, combinatorial optimization, and recommendation prioritization based on customer profiling. The results indicate that the system successfully produces relevant and personalized bundling recommendations based on customer age and transaction history. Furthermore, performance evaluation shows that DP achieves efficient and stable execution time, although Brute Force performs faster on small-scale Repeat Buyer E-Commerce datasets. Overall, the proposed approach proves effective in improving recommendation accuracy and supporting marketing strategies to enhance customer loyalty.

Keywords: Bundling Recommendation, Dynamic Programming, Knapsack Bottom-Up, *E-commerce*, *Repeat Buyer*.

PENDAHULUAN

Perkembangan teknologi digital mendorong pertumbuhan *e-commerce* di Indonesia semakin pesat dan kompetitif. Persaingan antar platform *e-commerce* tidak lagi hanya berfokus pada jumlah pengunjung, tetapi juga pada kemampuan platform mempertahankan pelanggan agar melakukan pembelian ulang (*repeat buyer*). Kepuasan pelanggan memiliki pengaruh besar terhadap *repeat buyer* yang menjadi salah satu faktor penting dalam menjaga pertumbuhan jangka panjang platform *e-commerce* [1]. Namun, penelitian [2] menunjukkan bahwa menurunnya jumlah kunjungan pengguna pada platform *e-commerce* terbesar di Indonesia seperti Tokopedia, Shopee, Bukalapak, Lazada dan Blibli mengindikasikan bahwa tingkat kepuasan pelanggan masih rendah sehingga berdampak pada rendahnya minat *repeat buyer* [2].

Salah satu pendekatan yang dapat digunakan untuk meningkatkan *repeat buyer* adalah dengan melakukan penerapan sistem rekomendasi dan strategi pengelompokan produk berbasis perilaku pelanggan (*behavior-based recommender system*). Sistem rekomendasi memanfaatkan data pribadi dan historis pengguna seperti riwayat pembelian untuk memberikan rekomendasi produk yang lebih personal dan relevan. Penelitian [3] menjelaskan bahwa sistem rekomendasi membantu pelanggan menemukan produk sesuai minatnya sehingga mampu meningkatkan loyalitas pelanggan. Namun, sebagian besar penelitian terkait sistem rekomendasi masih berfokus pada rekomendasi produk tunggal dan belum banyak yang mengembangkan rekomendasi dalam bentuk *bundling product* multi-kategori.

Strategi *product bundling* merupakan metode pemasaran yang dapat meningkatkan penjualan dan mendorong pembelian dalam jumlah lebih besar. Penelitian [4] mendefinisikan *product bundling* sebagai permasalahan *Multiple Constraint Knapsack Problem* (MCKP), yaitu pemilihan kombinasi produk terbaik dengan mempertimbangkan berbagai batasan. Kemudian diimplementasikan pada *e-commerce*, pembentukan *bundling product* menjadi lebih kompleks karena

banyaknya kemungkinan kombinasi produk yang dihasilkan. Pada penelitian [5] menyatakan bahwa *Multiple Knapsack Problem* (MKP) merupakan permasalahan optimasi bersifat *NP-hard* yang kompleksitasnya meningkat secara eksponensial seiring dengan jumlah item dan knapsack yang bertambah, sehingga membutuhkan metode yang efektif untuk memperoleh solusi optimal.

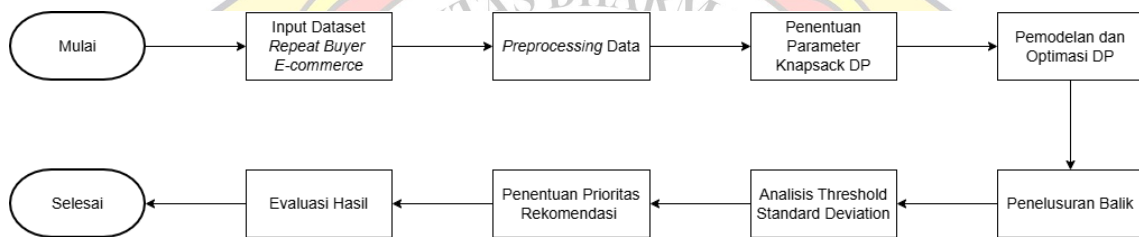
Dalam mengatasi kompleksitas tersebut, pendekatan Dynamic Programming (DP) dapat digunakan sebagai metode optimasi yang efektif. Penelitian [5] menjelaskan bahwa mampu memecah permasalahan optimasi yang kompleks seperti MKP menjadi sub-masalah yang lebih sederhana sehingga proses pencarian solusi dapat dilakukan secara sistematis. Varian DP terbukti mampu mengeksplorasi kombinasi item dengan hasil utilisasi sumber daya di atas 99,66%.

Selain aspek optimasi, pengembangan rekomendasi *bundling* yang terpersonalisasi juga menjadi perhatian dalam penelitian *e-commerce*. Penelitian [6] mengembangkan sistem rekomendasi *bundling* berbasis preferensi pengguna menggunakan integral Choquet untuk menangkap hubungan antar atribut produk dalam suatu *bundling*. Sementara itu, penelitian [7] mengembangkan kerangka optimasi dua tahap yang mengintegrasikan model prediktif dan untuk alokasi pemasaran multi-produk dan sukses meningkatkan volume pesanan hingga 14,48%. Meskipun demikian, penelitian-penelitian tersebut belum secara khusus mengintegrasikan data pribadi dan historis pelanggan *repeat buyer* sebagai bobot nilai produk dalam pembentukan *bundling product* multi-kategori menggunakan DP.

Penelitian ini bertujuan mengembangkan model optimasi berbasis Dynamic Programming dalam perancangan *bundling product* multi-kategori untuk menghasilkan rekomendasi penawaran yang optimal. Sebagai implikasi praktis, platform *e-commerce* dapat memanfaatkan strategi ini untuk merancang program diskon yang efektif dalam meningkatkan loyalitas *repeat buyer e-commerce* Indonesia.

METODE PENELITIAN

Penelitian ini menggunakan pendekatan algoritma Knapsack 0/1 berbasis DP yang diintegrasikan dengan analisis statistik Threshold Standard Deviation untuk mengoptimalkan sistem rekomendasi diskon *bundling product e-commerce*. Pendekatan DP ini merujuk pada formulasi Bellman Recursion (1957) yang divalidasi oleh penelitian [8] sebagai algoritma standar dengan kompleksitas waktu $O(nt)$ yang terbukti optimal secara kondisional. Tahapan penelitian mencakup pengumpulan dataset, *preprocessing* data, penentuan parameter, optimasi Knapsack, perhitungan *threshold* statistik, dan evaluasi rekomendasi *bundling product* yang dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

Uraian mengenai tahapan penelitian pada Gambar 1 adalah sebagai berikut:

1. Dataset

Penelitian menggunakan dataset publik "*Repeat Buyer E-commerce Dataset*" dari repositori Mendeley Data [9]. Dataset ini dipilih karena menyimpan informasi perilaku pembelian pelanggan multi-kategori yang representatif terhadap lanskap *e-commerce* di Indonesia. Validitas dataset ini dalam mempresentasikan perilaku pelanggan *e-commerce* di Indonesia telah diuji dan dikonfirmasi pada penelitian [10].

2. *Preprocessing* Data

Tahap *preprocessing* bertujuan mentransformasi data transaksi mentah menjadi metrik agregat per kategori yang terstruktur dan siap diproses oleh algoritma optimasi. Proses dimulai dengan filtrasi *missing*

values untuk menjaga integritas data, dilanjutkan dengan identifikasi pelanggan *repeat buyer* berdasarkan frekuensi transaksi lebih dari satu kali. Data kemudian diagregasi berdasarkan kategori produk untuk memperoleh total *amount*, jumlah pelanggan unik serta total usia seluruh pelanggan (W_1) dan total usia pelanggan unik (W_2). Selanjutnya dilakukan kalkulasi dua metrik demografis utama yang dikonversi ke nilai diskrit melalui pembulatan ke bawah, yaitu PW_1 sebagai rata-rata usia seluruh pelanggan pada tiap kategori dan PW_2 sebagai rata-rata usia pelanggan *repeat buyer*. Kategori yang tidak mempunyai pelanggan *repeat buyer* diberikan nilai $PW_2 = 0$ dan dieksklusi dari optimasi.

3. Penentuan Parameter Knapsack DP

Pada tahap ini, hasil *preprocessing* dialokasikan ke dalam parameter operasional model Knapsack. Penelitian menetapkan kapasitas maksimum Knapsack sebesar $W = 36$, yang diperoleh dari nilai tengah antara rata-rata PW_1 dan PW_2 seluruh populasi data. Berbeda dengan Knapsack konvensional yang menggunakan bobot statis, penelitian ini mengadopsi konsep *parametric weight knapsack problem* di mana bobot item dapat dimodelkan sebagai fungsi afin $w_i(\lambda) = a_i + \lambda \cdot b_i$ [11]. Pendekatan ini memungkinkan bobot bersifat lebih adaptif terhadap konteks data.

Berdasarkan konsep tersebut, bobot efektif item ke- i diformulasikan menggunakan fungsi non-negatif agar item yang melampaui kapasitas dihitung sebagai beban sistem sebagaimana pada Persamaan 1.

$$w_i = \max(0, PW_{1_i} - W) \quad (1)$$

w_i adalah bobot efektif item ke- i , PW_{1_i} adalah rata-rata usia pelanggan umum pada kategori ke- i , dan W adalah kapasitas maksimum knapsack. Formulasi ini mengacu pada model parametrik oleh penelitian [11] dengan menggunakan nilai parameter $a_i = PW_{1_i}$, $\lambda = W$, dan $b_i = -1$. Penggunaan operator $\max(0, \dots)$ juga sejalan dengan konsep *stochastic knapsack* pada penelitian [12] melalui ekspresi $E[\max(w'x - C, 0)]$,

sehingga item dengan $PW_{1_i} \leq W$ tidak diberi penalti bobot. Selanjutnya, profit item dimodelkan menggunakan nilai profil demografis pelanggan unik sebagaimana pada Persamaan 2.

$$v_i = PW_{2_i} \quad (2)$$

v_i adalah parameter profit efektif item ke- i dan PW_{2_i} adalah rata-rata usia pelanggan unik pada kategori ke- i . Pendekatan relatif terhadap kapasitas ini mengikuti prinsip dari penelitian [13] untuk mengurangi bias nilai absolut dan mendukung optimasi loyalitas pelanggan pada DP.

4. Pemodelan dan Optimasi DP

Proses optimasi DP dilakukan menggunakan formulasi rekursif Bellman Recursion (1957) sebagai metode standar penyelesaian masalah Knapsack [14]. Sistem membangun matriks pencarian M berukuran $(n + 1) \times (W + 1)$ dengan pendekatan Bottom-Up. Matriks diinisialisasi menggunakan kondisi basis $M(0, w) = 0$ untuk seluruh kapasitas w dan $M(i, 0) = 0$ untuk seluruh item i , dengan n sebagai jumlah kategori valid dan W sebagai kapasitas maksimum knapsack. Pengisian matriks dilakukan secara iteratif menggunakan formulasi pada Persamaan 3 dan 4.

$$M(i, w) = \begin{cases} M(i - 1, w), & \text{if } w_i > w \quad (3) \\ \max(M(i - 1, w), M(i - 1, w - w_i) + v_i), & \text{if } w_i \leq w \quad (4) \end{cases}$$

Pada Persamaan 3, item ke- i tidak dipilih apabila bobot w_i melebihi kapasitas tersedia w , sehingga nilai sel mengikuti hasil sebelumnya. Sebaliknya, Persamaan 4 digunakan ketika $w_i \leq w$, di mana sistem memilih nilai maksimum antara mengabaikan item atau memasukkan item untuk memperoleh tambahan profit sebesar v_i .

5. Penelusuran Balik

Setelah matriks pencarian M terisi penuh, proses penelusuran balik dilakukan untuk mengidentifikasi kombinasi item yang membentuk solusi optimal. Penelusuran dimulai dari posisi $M(n, W)$ pada baris terakhir $i = n$

dengan kapasitas referensi $k = W$, kemudian bergerak menuju baris ke-0. Pada setiap posisi (i, k) , sistem membandingkan nilai sel (i) dengan nilai pada baris sebelumnya $(i - 1)$ untuk menentukan apakah item ke- i dipilih atau tidak. Aturan penelusuran dapat dilihat pada Persamaan 5 dan 6.

$$i > 0; k > 0 \begin{cases} i = i - 1; k = k - w_i, & \text{if } M(i, k) > M(i - 1, k) \\ i = i - 1, & \text{if } M(i, k) \leq M(i - 1, k) \end{cases} \quad (5)$$

$$i = i - 1, \quad \text{if } M(i, k) \leq M(i - 1, k) \quad (6)$$

Pada Persamaan 5, item ke- i dinyatakan terpilih apabila nilai profitabilitas meningkat sehingga indeks baris diturunkan ($i = i - 1$) dan kapasitas dikurangi sebesar ($k = k - w_i$). Sedangkan pada Persamaan 6 menunjukkan bahwa item ke- i diabaikan apabila tidak terjadi peningkatan profitabilitas. Proses iteratif berhenti ketika indeks baris pada $i = 0$.

6. Analisis Threshold Standard Deviation

Dalam tahap *customer profiling*, digunakan parameter *threshold* (T) untuk membentuk toleransi kedekatan usia pelanggan. Pendekatan ini mengacu pada penelitian [15] yang mengevaluasi kedekatan data berdasarkan rata-rata dan standar deviasi distribusi empiris. Nilai standar deviasi (σ) dihitung menggunakan Persamaan 7.

$$\sigma = \sqrt{\frac{\sum f_j (X_j - \lambda)^2}{N - 1}} \quad (7)$$

f_j adalah jumlah pelanggan pada usia tertentu, X_j adalah usia pelanggan, λ adalah rata-rata usia keseluruhan populasi pelanggan unik, dan N adalah total pelanggan unik. Nilai standar deviasi tersebut kemudian digunakan untuk menentukan batas toleransi pada Persamaan 8.

$$T = \frac{\sigma}{2} \quad (8)$$

Penggunaan setengah standar deviasi sebagai *proximity caliper* melegitimasi konsep pada penelitian [16]. Nilai T selanjutnya digunakan

untuk mengevaluasi kesesuaian usia pelanggan terhadap rata-rata usia pelanggan umum (PW_{1_i}) melalui Persamaan 9.

$$|X - PW_{1_i}| \leq T \text{ dan } PW_{2_i} > 0 \quad (9)$$

Pada Persamaan 9, usia pelanggan X dinyatakan valid apabila berada dalam batas T dan kategori memiliki riwayat *repeat buyer*. Proses *bundling* hanya dilanjutkan jika terdapat minimal dua kategori yang memenuhi kriteria tersebut.

7. Penentuan Prioritas Rekomendasi

Setelah daftar kategori produk lolos verifikasi melalui Persamaan 9, sistem melakukan *sorting prioritization* untuk menentukan kombinasi rekomendasi terbaik. Prioritas utama diberikan pada kategori dengan nilai selisih absolut terkecil antara usia pelanggan target dan rata-rata usia pelanggan umum ($|X - PW_{1_i}|$). Pendekatan ini mengikuti prinsip *smallest distance* oleh penelitian [16] untuk menjaga presisi personalisasi demografis. Jika terdapat nilai selisih yang sama, prioritas berikutnya ditentukan berdasarkan nilai rata-rata usia pelanggan *repeat buyer* tertinggi (PW_{2_i}). Apabila kondisi masih seimbang, maka dilihat dari akumulasi volume penjualan (*Total_Amount*) dan jumlah pelanggan unik (*Total_Cust_Unique*) sebagai penentu akhir agar rekomendasi *bundling* tetap optimal secara demografis dan performa pasar.

8. Evaluasi Hasil

Tahap evaluasi dilakukan dengan membandingkan kompleksitas waktu antara algoritma Brute Force dan DP menggunakan notasi Big-O. Analisis komparatif ini mengevaluasi skenario *best case*, *worst case* dan *average case* untuk membuktikan efisiensi komputasi kedua metode pada optimasi Knapsack. Hasil evaluasi ini akan disajikan dalam bentuk tabel perbandingan yang mensubstitusikan nilai jumlah kategori ($n = 11$) dan kapasitas knapsack ($W = 36$) untuk melihat jumlah operasi konkret dari masing-masing algoritma.

HASIL DAN PEMBAHASAN

Tahapan pengujian dan implementasi algoritma DP dilakukan langsung pada dataset. Bagian ini menguraikan hasil komputasi sistem mulai dari *preprocessing* data, penetapan parameter matematis, optimasi kombinatorial, hingga penyusunan prioritas *bundling* menggunakan ambang batas demografis.

1. Dataset

Repeat Buyer E-commerce Dataset yang digunakan terdiri dari 1.000 transaksi penjualan dari 178 pelanggan pada 13 kategori produk. Seluruh data historis diimpor ke basis data sistem sebagai input pada tahap *preprocessing*. Representasi data mentah ditampilkan pada Tabel 1.

Tabel 1. Data Mentah *Repeat Buyer E-commerce*

Trx_Date	Cust_ID	Age	Gender	AREA	Product_ID	Product_Name	Category	Amount
6/1/2022	CTR0200437	49	Laki-laki	Aceh	HP-PV15	Laptop HP Pavilion 15	Computer	1
6/1/2022	CTR0200460	18	Perempuan	Bandung	192021232	Hair Mask CDE	Hair Treatment	1
...
7/31/2022	CTR0200555	36	Perempuan	Yogyakarta	112233445	Lip Balm DEF	Cosmetics	1
7/31/2022	CTR0200412	38	Perempuan	Purwokerto	232425275	Perfume OPQ	Cosmetics	1

2. *Preprocessing* Data

Tahap *preprocessing* menghasilkan metrik agregat terstruktur melalui proses filtrasi data, identifikasi pelanggan *repeat buyer*, dan perhitungan metrik demografis setiap kategori. Kategori tanpa riwayat *repeat buyer* ($PW_{2i} = 0$) otomatis dieliminasi dari proses lanjutan. Hasil ekstraksi yang digunakan sebagai parameter optimasi ditampilkan pada Tabel 2.

Tabel 2. Hasil Ekstraksi Data *Repeat Buyer E-commerce*

Kategori	Total_Amount	Total_Cust_Unique	W_1	W_2	PW_1	PW_2	Status Knapsack
Body Treatment	260	39	37.6	38.51	37	38	Terpilih
Books	7	0	46	0	46	0	Dilewati
Computer	57	5	41.21	43.8	41	43	Terpilih

Kategori	Total_Amount	Total_Cust_Unique	W_1	W_2	PW_1	PW_2	Status Knapsack
Convenience Goods	47	5	37.96	39	37	39	Terpilih
Cosmetics	660	77	38.64	38.49	38	38	Terpilih
Electronics and Gadget	306	38	39.31	40.32	39	40	Terpilih
Equipments	47	0	44.75	0	44	0	Dilewati
Fashion	46	6	38.38	38.83	38	38	Terpilih
Furniture	9	1	35.12	37	35	37	Terpilih
Hair Treatment	213	27	38.02	38.85	38	38	Terpilih
Home Appliances	59	7	40.05	39	40	39	Terpilih
Kitchen Appliances	71	5	39.48	37	39	37	Terpilih
Sport	38	2	41.58	42.5	41	42	Terpilih

3. Penentuan Parameter Knapsack DP

Sebanyak 11 kategori produk dengan status "Terpilih" pada tahap *preprocessing* dialokasikan menjadi variabel operasional Knapsack dengan kapasitas maksimum $W = 36$. Nilai bobot w_i dihitung dari selisih antara atribut PW_1 terhadap batas kapasitas W , sedangkan profit v_i dipetakan secara langsung dari PW_2 . Kategori dengan nilai $PW_{1i} \leq 36$ memiliki bobot $w_i = 0$ sehingga tidak membebani kapasitas Knapsack. Hasil konversi parameter menjadi 11 item kandidat ditampilkan pada Tabel 3.

Tabel 3. Hasil Parameter Knapsack DP

i	Kategori	w_i	v_i
1	Body Treatment	1	38
2	Computer	5	43
3	Convenience Goods	1	39
4	Cosmetics	2	38
5	Electronics and Gadget	3	40
6	Fashion	2	38
7	Furniture	0	37
8	Hair Treatment	2	38
9	Home Appliances	4	39
10	Kitchen Appliances	3	37
11	Sport	5	42

4. Pemodelan dan Optimasi DP

Berdasarkan parameter bobot dan profit yang telah ditetapkan, sistem melakukan optimasi kombinatorial menggunakan algoritma DP secara Bottom-Up melalui matriks evaluasi berukuran 12×37 . Dimensi matriks ini terbentuk dari 11 kategori kandidat serta satu baris dan kolom inisialisasi awal. Setiap sel diisi secara iteratif dengan membandingkan profit maksimal antara kondisi memilih atau mengabaikan item pada setiap kapasitas. Hasil matriks optimasi ditampilkan pada Gambar 2.

i\w	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38
2	0	38	38	38	38	43	81	81	81	81	81	81	81	81	81	81	81	81	81
3	0	39	77	77	77	77	82	120	120	120	120	120	120	120	120	120	120	120	120
4	0	39	77	77	115	115	115	120	120	158	158	158	158	158	158	158	158	158	158
5	0	39	77	77	115	117	117	155	155	158	160	160	198	198	198	198	198	198	198
6	0	39	77	77	115	117	153	155	155	193	193	196	198	198	236	236	236	236	236
7	0	76	114	114	152	154	190	192	192	230	230	233	235	235	273	273	273	273	273
8	0	76	114	114	152	154	190	192	228	230	230	268	268	271	273	273	311	311	311
9	0	76	114	114	152	154	190	192	228	230	230	268	268	271	273	307	311	311	312
10	0	76	114	114	152	154	190	192	228	230	230	268	268	271	305	307	311	311	344
11	0	76	114	114	152	154	190	192	228	230	230	268	268	271	305	307	311	311	344

Gambar 2(a). Hasil Matriks DP Knapsack

i\w	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38
2	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81
3	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120
4	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158
5	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198
6	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236
7	273	273	273	273	273	273	273	273	273	273	273	273	273	273	273	273	273	273
8	311	311	311	311	311	311	311	311	311	311	311	311	311	311	311	311	311	311
9	312	350	350	350	350	350	350	350	350	350	350	350	350	350	350	350	350	350
10	348	350	350	350	387	387	387	387	387	387	387	387	387	387	387	387	387	387
11	348	350	353	353	387	390	392	392	392	429	429	429	429	429	429	429	429	429

Gambar 2(b). Hasil Matriks DP Knapsack

5. Penelusuran Balik

Berdasarkan matriks yang terbentuk pada proses optimasi DP, dilakukan algoritma penelusuran balik dimulai dari titik elemen terakhir

untuk mengidentifikasi rute solusi optimal. Eksekusi tahap ini secara definitif menyaring dan menetapkan kategori produk mana saja yang divalidasi masuk ke dalam kombinasi *bundling* terbaik. Rute penelusuran balik dalam menentukan kategori terpilih dapat dilihat pada Gambar 3.

i\w	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38
2	0	38	38	38	38	43	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81	81
3	0	39	77	77	77	77	82	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120
4	0	39	77	77	115	115	115	120	120	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158	158
5	0	39	77	77	115	117	117	155	155	158	160	160	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198	198
6	0	39	77	77	115	117	153	155	155	193	193	196	198	198	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236	236
7	0	76	114	114	152	154	190	192	192	230	230	233	235	235	273	273	273	273	273	273	273	273	273	273	273	273	273	273	273	273	273	273	273	273	273	273	273	273	273
8	0	76	114	114	152	154	190	192	228	230	230	268	268	271	273	273	311	311	311	311	311	311	311	311	311	311	311	311	311	311	311	311	311	311	311	311	311	311	311
9	0	76	114	114	152	154	190	192	228	230	230	268	268	271	273	307	311	311	312	312	350	350	350	350	350	350	350	350	350	350	350	350	350	350	350	350	350	350	350
10	0	76	114	114	152	154	190	192	228	230	230	268	268	271	305	307	311	311	344	348	350	350	350	387	387	387	387	387	387	387	387	387	387	387	387	387	387	387	387
11	0	76	114	114	152	154	190	192	228	230	230	268	268	271	305	307	311	311	344	348	350	353	353	387	390	392	392	392	429	429	429	429	429	429	429	429	429	429	

Gambar 3. Hasil Matriks Penelusuran Balik

6. Analisis Threshold Standard Deviation

Sistem menentukan nilai *threshold* (T) dengan menganalisis distribusi empiris usia pelanggan melalui frekuensi pelanggan (f_j) pada setiap usia pelanggan unik (X_j) beserta akumulasi kuadrat selisih terhadap rata-rata populasi ($\sum f_j (X_j - \lambda)^2$). Hasil distribusi usia pelanggan yang digunakan untuk menghitung standar deviasi dijabarkan pada Tabel 4.

Tabel 4. Hasil Perhitungan Distribusi Usia Pelanggan

X_j	f_j	$\sum f_j (X_j - \lambda)^2$
18	5	2.205
19	2	800
20	1	361
...
55	2	512
56	2	578
60	1	441
Total	178	17.102

Berdasarkan hasil perhitungan pada Tabel 4, diperoleh nilai standar deviasi sebesar $\sigma \approx 96,67$. Nilai tersebut kemudian disubstitusikan ke

dalam Persamaan 8 sehingga diperoleh batas sebesar $T \approx 5$. Nilai T ini akan digunakan untuk mengevaluasi kelayakan kategori berdasarkan selisih absolut usia ($|X - PW_1|$) dan profit (PW_2). Hasil kelayakan kategori setiap usia pelanggan unik dapat dilihat pada Tabel 5.

Tabel 5. Hasil Kelayakan Kategori

X	Kategori Eligible	$ X - PW_1 $	PW_2	Status
18-29				Tidak ada <i>repeat buyer</i>
30	Furniture	5	37	Tidak memenuhi syarat
35	Furniture	0	37	Kategori terpilih layak
	Convenience Goods	2	39	
	Body Treatment	2	38	
	Cosmetics	3	38	
	Hair Treatment	3	38	
	Fashion	3	38	
	Electronics and Gadget	4	40	
	Kitchen Appliances	4	37	
	Home Appliances	5	39	
...
45	Computer	4	43	Kategori terpilih layak
	Sport	4	42	
	Home Appliances	5	39	
46	Computer	5	43	Kategori terpilih layak
	Sport	5	42	
47-60				Tidak ada <i>repeat buyer</i>

Hasil analisis menunjukkan bahwa beberapa kelompok usia tidak memperoleh rekomendasi karena tidak memiliki riwayat *repeat buyer* $PW_2 = 0$ atau jumlah kategori valid kurang dari dua. Sebaliknya, kelompok usia dengan intensitas transaksi tinggi menghasilkan lebih banyak kategori yang lolos validasi untuk proses *bundling*.

7. Penentuan Prioritas Rekomendasi

Hasil mekanisme prioritas menghasilkan hierarki rekomendasi kategori produk untuk setiap kelompok usia berdasarkan kedekatan demografis dan performa transaksi, mulai dari prioritas tertinggi hingga kategori yang tereliminasi. Sebagai bentuk representasi, dipilih salah satu hasil prioritas untuk kelompok usia 43 tahun dijabarkan pada Tabel 6.

Tabel 6. Hasil Daftar Prioritas Rekomendasi Usia 43 Tahun

Kategori	PW1	PW2	$ X - PW_1 $	Total_Amount	Total_Cust_Unique	Prioritas
Computer	41	43	2	57	5	Prioritas 1
Sport	41	42	2	38	2	Prioritas 2
Home Appliances	40	39	3	59	7	Prioritas 3
Electronics and Gadget	39	40	4	306	38	Prioritas 4
Kitchen Appliances	39	37	4	71	5	Prioritas 5
Cosmetics	38	38	5	660	77	Prioritas 6
Hair Treatment	38	38	5	213	27	Prioritas 7
Fashion	38	38	5	46	6	Prioritas 8
Body Treatment	37	38	6	260	39	Tidak prioritas
Convenience Goods	37	39	6	47	5	Tidak prioritas
Furniture	35	37	8	9	1	Tidak prioritas

Berdasarkan hasil Tabel 6, kategori dengan nilai selisih absolut terkecil menempati prioritas tertinggi, sedangkan kategori dengan jarak usia melebihi batas toleransi tidak diprioritaskan. Ketika terdapat nilai kedekatan yang sama, sistem menggunakan volume penjualan dan jumlah pelanggan unik sebagai atribut pemecah seri untuk menentukan urutan prioritas rekomendasi. Daftar prioritas tersebut kemudian dicocokkan dengan riwayat transaksi pelanggan pada Tabel 7.

Tabel 7. Riwayat Transaksi Salah Satu Pelanggan Usia 43 Tahun

Cust_ID	Usia	Kategori	Amount
CTR0200557	43	Hair Treatment	2
CTR0200557	43	Cosmetics	2
CTR0200557	43	Body Treatment	2

Riwayat transaksi pada Tabel 7 menunjukkan hasil pencocokan kategori 'Body Treatment' dieliminasi karena melebihi batas T , sedangkan kategori 'Cosmetics' dan 'Hair Treatment' tetap valid. Berdasarkan urutan prioritas pada Tabel 6, sistem menetapkan 'Cosmetics' sebagai *Bundling* Utama dan 'Hair Treatment' sebagai *Bundling* Pasangan. Rekomendasi yang dihasilkan tetap mempertimbangkan kedekatan demografis dan riwayat pembelian pelanggan.

8. Evaluasi Hasil

Evaluasi hasil dilakukan dengan membandingkan kompleksitas waktu antara algoritma DP dan Brute Force dalam menyusun kombinasi *bundling*. Perbandingan hasil kompleksitas waktu kedua algoritma dapat dilihat pada Tabel 8.

Tabel 8. Perbandingan Kompleksitas Waktu

Skenario	Dynamic Programming		Brute Force	
	Notasi Big-O	Total Operasi	Notasi Big-O	Total Operasi
Best Case	$O(nW)$	396	$O(1)$	1
Worst Case	$O(nW)$	396	$O(2^{n-1})$	1.024
Average Case	$O(nW)$	396	$O(2^n)$	2.048

Berdasarkan Tabel 8, dapat dilihat bahwa hasil algoritma DP memiliki kompleksitas deterministik sebesar $O(nW)$, sehingga jumlah operasinya tetap sama pada seluruh skenario yaitu $11 \times 36 = 396$. Hal ini terjadi dikarenakan DP selalu mengisi seluruh matriks solusi tanpa mekanisme *early stopping*. Oleh karena itu, beban komputasi yang dieksekusi akan selalu konstan pada semua skenario pencarian, mulai dari *worst case*, *average case*, dan *best case* setelah disubstitusi ke parameter dataset dalam notasi Big-O. Hasilnya algoritma DP hanya membutuhkan 396 operasi komputasi.

Sebaliknya, efisiensi algoritma Brute Force memiliki variansi tinggi karena sangat bergantung pada letak atau urutan ditemukannya solusi optimal. Pada *best case*, solusi ditemukan pada subset pertama sehingga kompleksitasnya $O(1)$. Pada *average case*, diestimasikan memeriksa sekurang-kurangnya separuh dari total kemungkinan subset dengan tingkat kompleksitas $O(2^{n-1})$ atau membutuhkan sebanyak 1.024 operasi. Sementara pada *worst case*, solusi optimal berada pada pencarian akhir dan menghasilkan kompleksitas eksponensial penuh sebesar $O(2^n)$ atau membutuhkan sebanyak 2.048 operasi.

Secara keseluruhan, hasil evaluasi menunjukkan bahwa algoritma DP lebih unggul dan stabil dibandingkan Brute Force dalam optimasi Knapsack.

Pada pengujian ini, DP terbukti $\pm 5,2$ kali lebih efisien dibandingkan skenario terburuk Brute Force, yang memiliki pertumbuhan kompleksitas eksponensial saat jumlah kategori produk bertambah.

SIMPULAN

Penelitian ini menyimpulkan bahwa optimasi strategi pelanggan *repeat buyer* pada e-commerce di Indonesia berhasil diwujudkan melalui pendekatan Dynamic Programming untuk rekomendasi *bundling product* multi-kategori. Integrasi algoritma Knapsack dan analisis statistik Threshold Standard Deviation melalui pemodelan bobot adaptif mampu menyaring serta menyusun prioritas produk berdasarkan usia dan riwayat transaksi pelanggan. Penerapan ambang batas demografis juga berhasil mengeliminasi kategori yang tidak relevan sehingga menghasilkan rekomendasi *bundling* secara akurat untuk mendorong loyalitas pelanggan. Dari sisi performa komputasi, algoritma yang diusulkan menunjukkan kompleksitas waktu yang efisien dan tetap stabil untuk menghadapi peningkatan volume data transaksi di masa mendatang.

DAFTAR PUSTAKA

- [1] S. Afinia and E. Tjahjaningsih, "Customer Satisfaction's Influence on Repurchase Intention in Indonesia's E-commerce Sector," *Jurnal Informatika Ekonomi Bisnis*, vol. 6, no. 3, pp. 634–639, Aug. 2024, doi: <https://doi.org/10.37034/infec.v6i3.948>.
- [2] Y. M. Ginting, T. Chandra, I. Miran, and Y. Yusriadi, "Repurchase intention of e-commerce customers in Indonesia: An overview of the effect of e-service quality, e-word of mouth, customer trust, and customer satisfaction mediation," *International Journal of Data and Network Science*, vol. 7, no. 1, pp. 329–340, Dec. 2023, doi: 10.5267/J.IJDNS.2022.10.001.
- [3] R. B. Nozari, M. Divsalar, S. A. Abkenar, M. F. Amiri, and A. Divsalar, "A Novel Behavior-Based Recommendation System for E-commerce," Mar. 2024, doi: <https://doi.org/10.37034/infec.v6i3.948>.
- [4] A. N. Carloman, U. V. W. Bermudo, E. M. Estilloso, and O. E. Llantos, "Bundle AI: An Application of Multiple Constraint Knapsack Problem (MCKP) Through Genetic Algorithm (GA)," *Procedia Comput. Sci.*, vol. 231, pp. 24–31, Jan. 2024, doi: 10.1016/J.PROCS.2023.12.153.
- [5] D. A. Santoso, I. Rizqa, D. Aqmala, F. Alzami, N. Rijati, and A. Marjuni, "Performance Analysis of Multiple Knapsack Problem Optimization

- Algorithms: A Comparative Study for Retail and SME Applications," *Ingenierie des Systemes d'Information*, vol. 30, no. 2, pp. 533–550, Feb. 2025, doi: 10.18280/ISI.300224.
- [6] E. Robbi, M. Bronzini, P. Viappiani, and A. Passerini, "Personalized bundle recommendation using preference elicitation and the Choquet integral," *Front. Artif. Intell.*, vol. 7, p. 1346684, Feb. 2024, doi: 10.3389/FRAI.2024.1346684/TEXT.
- [7] Y. Xie, H. Q. Ye, and W. Zhu, "Prediction and Optimization for Multi-Product Marketing Resource Allocation in Cross-Border E-Commerce," *Journal of Theoretical and Applied Electronic Commerce Research 2025*, Vol. 20, Page 124, vol. 20, no. 2, p. 124, Jun. 2025, doi: 10.3390/JTAER20020124.
- [8] C. Jin, "Solving Knapsack with Small Items via L0-Proximity," Jul. 2023, Accessed: May 24, 2026. [Online]. Available: <https://arxiv.org/pdf/2307.09454>
- [9] S. F. Mauludiah, "Repeat Buyer E-commerce Dataset," vol. 1, 2025, doi: 10.17632/HWVZ4KV232.1.
- [10] S. F. Mauludiah, C. Crysdiyan, and Y. M. Arif, "Enhancing Repeat Buyer Classification with Multi Feature Engineering in Logistic Regression," *Applied Information System and Management (AISM)*, vol. 8, no. 1, pp. 103–110, May 2025, doi: 10.15408/AISM.V8I1.45025.
- [11] N. Halman, M. Holzhauser, and S. O. Krumke, "An FPTAS for the Knapsack Problem with Parametric Weights," *Operations Research Letters*, vol. 46, no. 5, pp. 487–491, Mar. 2017, doi: 10.1016/j.orl.2018.07.005.
- [12] R. Rossi, S. D. Prestwich, and S. A. Tarim, "Mixed-Integer Linear Programming Approximations for the Stochastic Knapsack," Dec. 2025, Accessed: May 25, 2026. [Online]. Available: <https://arxiv.org/pdf/2512.14912>
- [13] Z. Zhang, H. Yin, L. Zuo, and P. Lai, "Reinforcement Learning for Solving the Knapsack Problem," *Computers, Materials and Continua*, vol. 84, no. 1, pp. 919–936, 2025, doi: 10.32604/CMC.2025.062980.
- [14] H. Kellerer, U. Pferschy, and D. Pisinger, "Knapsack Problems," *Knapsack Problems*, 2004, doi: 10.1007/978-3-540-24777-7.
- [15] P. Zhao, X. Su, T. Ge, and J. Fan, "Propensity Score and Proximity Matching Using Random Forest," *Contemp. Clin. Trials*, vol. 47, p. 85, Mar. 2015, doi: 10.1016/J.CCT.2015.12.012.
- [16] M. S. E. Kasem, M. Hamada, and I. Taj-Eddin, "Customer profiling, segmentation, and sales prediction using AI in direct marketing," *Neural Computing and Applications 2023 36:9*, vol. 36, no. 9, pp. 4995–5005, Dec. 2023, doi: 10.1007/S00521-023-09339-6.