

Perbandingan Algoritma Brute Force, Dynamic Programming, Branch and Bound, dan Greedy untuk Optimasi Perencanaan Menu Makan Bergizi Gratis (MBG) Tujuh Hari

Angelina Andra Alanna - 13524079

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail: angelinaandra93@gmail.com , 13524079@std.stei.itb.ac.id

Abstract—Program Makan Bergizi Gratis (MBG) merupakan intervensi gizi nasional berskala besar yang memerlukan penyusunan menu memenuhi standar gizi dengan anggaran terbatas. Penyusunan menu yang memenuhi kecukupan gizi, batas anggaran, dan keragaman antar-hari secara bersamaan merupakan masalah optimasi kombinatorial yang tergolong NP-hard. Makalah ini memformulasikan perencanaan menu MBG tujuh hari sebagai masalah knapsack multi-dimensi multi-periode, dengan menu harian dirakit dari bahan pangan, lalu membandingkan empat strategi algoritma yaitu brute force, dynamic programming, branch and bound, dan greedy ke dalam dua mode objektif: minimasi biaya dan maksimasi skor gizi. Eksperimen dengan data komposisi gizi dan harga pangan Indonesia menunjukkan keempat algoritma menghasilkan solusi optimal yang identik pada instans layak, namun dengan kinerja yang sangat berbeda. Branch and bound paling seimbang, tetap praktis hingga $n = 50$ (di bawah 5 ms), sedangkan dynamic programming terkena ledakan ruang state hingga mencapai 39 detik pada $n = 30$. Greedy tercepat dan umumnya optimal, namun menyimpang dari optimum (gap 0,40%) pada anggaran ketat. Hasil ini semakin menegaskan bahwa pemilihan algoritma bergantung pada konteks kebutuhan optimalitas, ukuran masalah, dan ketatnya kendala.

Kata Kunci—Makanan Bergizi Gratis, knapsack multi-demensi, optimasi kombinatorial, branch and bound, NP-hard.

I. PENDAHULUAN

Permasalahan gizi masih menjadi permasalahan besar di Indonesia. Hasil Survei Status Gizi Indonesia (SSGI) 2024 yang dirilis Kementerian Kesehatan mencatat tingkat *stunting* nasional 19,8 persen, atau setara dengan 4.482.340 balita [1]. Meskipun angka ini menurun 1,7 persen dibandingkan tahun 2023 (21,5 persen), capaian tersebut masih jauh dari target Rencana Pembangunan Jangka Menengah Nasional (RPJMN) sebesar 14,2 persen pada tahun 2029 [2]. Selain itu, SSGI 2024 juga mengungkap kesenjangan antar kelompok ekonomi: prevalensi *stunting* pada kuintil termiskin mencapai 29,8 persen, lebih dari dua kali lipat kuintil terkaya yang berada di kisaran 12 persen [2]. Data ini menegaskan bahwa akses terhadap pangan bergizi belum merata dan intervensi negara dalam pemenuhan

gizi, khususnya bagi anak-anak dari keluarga kurang mampu, masih sangat dibutuhkan.

Sebagai respons atas permasalahan tersebut, pemerintah menjalankan program Makan Bergizi Gratis (MBG) pada Januari 2025 sebagai program prioritas nasional yang dikelola oleh Badan Gizi Nasional (BGN). Program ini memiliki tujuan untuk memperbaiki status gizi masyarakat, terutama anak sekolah, ibu hamil, dan balita, sekaligus menggerakkan ekonomi lokal melalui rantai pasok pangan daerah [3]. Dari sisi skala, MBG tergolong intervensi pangan terbesar dalam sejarah kebijakan publik Indonesia: pada tahun 2026 alokasi anggarannya mencapai Rp268 triliun dengan target 82,9 juta penerima manfaat [4]. Hingga akhir Maret 2026, program ini telah menjangkau lebih dari 61,6 juta penerima di 38 provinsi melalui lebih dari 26.000 dapur Satuan Pelayanan Pemenuhan Gizi (SPPG) [5]. Dengan skala sebesar ini, peningkatan efisiensi sekecil apa pun pada level porsi makanan akan berdampak besar secara keseluruhan terhadap keuangan negara.

Program pemberian makan di sekolah bukan hal yang baru. Jepang telah melaksanakan program *kyushoku* sejak 1954, dengan ahli gizi tersertifikasi di setiap sekolah yang merancang menu bulanan berbasis standar gizi nasional [6]. India menjalankan PM POSHAN (sebelumnya *Mid-Day Meal Scheme*), program makan sekolah terbesar di dunia yang melayani lebih dari 100 juta anak [7]. *Programa Nacional de Alimentação Escolar* (PNAE) bahkan menetapkan secara eksplisit persentase kebutuhan gizi harian yang wajib dipenuhi oleh setiap porsi makanan sekolah [8]. Pengalaman panjang negara-negara tersebut memperlihatkan hal yang serupa: keberhasilan program makan sekolah tidak hanya ditentukan oleh besarnya pendanaan, melainkan oleh perencanaan menu yang sistematis, terstandar, dan berbasis ilmu gizi.

Pada tahap implementasi, Program MBG masih menghadapi beberapa tantangan dalam mewujudkan penyediaan makanan yang aman, terstandar, dan sesuai dengan prinsip ilmu gizi. Pemerintah telah menyesuaikan anggaran MBG tahun 2026 dari rencana awal Rp335 triliun menjadi Rp268 triliun sebagai bagian dari upaya evaluasi dan peningkatan efisiensi tata kelola [9]. Program ini sempat menghadapi sejumlah kasus keracunan

pangan yang menjadi sorotan publik. Kajian independen juga mengidentifikasi berbagai permasalahan tata kelola dalam pengelolaan dapur SPPG [10], meskipun tren kejadian keracunan pangan menurun pada akhir 2025 seiring pengetatan standar operasional dapur [11]. Salah satu tantangan teknis yang menonjol adalah keterbatasan jumlah tenaga ahli gizi. Ribuan dapur SPPG dituntut untuk menyusun menu harian yang memenuhi kebutuhan gizi anak sekolah dengan batas anggaran yang relatif ketat, yaitu sekitar Rp10.000 per porsi, sementara ketersediaan ahli gizi pada setiap dapur masih menjadi perdebatan [12]. Kondisi ini berpotensi menimbulkan perbedaan kualitas, kecukupan gizi, serta keragaman menu antar dapur penyelenggara, sehingga tujuan pemerataan layanan gizi belum sepenuhnya dapat tercapai.

Apabila ditinjau dari perspektif komputasi, penyusunan menu MBG pada dasarnya merupakan permasalahan optimasi kombinatorial yang kompleks. Setiap dapur SPPG harus menentukan kombinasi bahan pangan dan hidangan dari ratusan alternatif yang tersedia, seperti yang tercantum dalam Tabel Komposisi Pangan Indonesia (TKPI) [13], dengan mempertimbangkan berbagai kendala secara bersamaan. Kendala tersebut meliputi pemenuhan kebutuhan energi, protein, serta zat gizi mikro sesuai Angka Kecukupan Gizi (AKG) anak usia sekolah [14], batas anggaran per porsi yang telah ditetapkan pemerintah, dan kebutuhan menjaga variasi menu agar tidak menimbulkan kejenuhan konsumsi dalam jangka waktu tertentu.

Kompleksitas permasalahan meningkat seiring bertambahnya jumlah kandidat bahan pangan dan periode perencanaan menu. Jumlah kemungkinan kombinasi yang dapat dibentuk tumbuh secara eksponensial sehingga penyusunan menu secara manual menjadi tidak efisien dan berisiko menghasilkan solusi yang kurang optimal, baik dari sisi biaya maupun pemenuhan kebutuhan gizi. Secara matematis, permasalahan ini dapat dimodelkan sebagai Multi-Dimensional Knapsack Problem (MKP) dengan perluasan pada beberapa periode perencanaan (multi-period), suatu kelas permasalahan optimasi yang telah diketahui bersifat NP-hard [15].

Berdasarkan latar belakang tersebut, penelitian ini memformulasikan perencanaan menu MBG selama tujuh hari sebagai permasalahan *multi-dimensional multi-period knapsack*. Selanjutnya, empat pendekatan algoritmik dibandingkan untuk menyelesaikan permasalahan tersebut, yaitu brute force sebagai metode eksak dasar, dynamic programming sebagai metode eksak pseudo-polinomial, branch and bound sebagai metode eksak yang memanfaatkan pemangkasan ruang pencarian, serta algoritma greedy sebagai pendekatan heuristik berbiaya komputasi rendah. Evaluasi dilakukan menggunakan data komposisi gizi dan harga pangan Indonesia untuk menilai kinerja masing-masing algoritma berdasarkan tiga aspek utama, yaitu optimalitas solusi, waktu eksekusi, dan skalabilitas terhadap ukuran masalah. Sistematika penulisan makalah ini adalah sebagai berikut. BAB II membahas landasan teori yang mencakup Program MBG, standar kebutuhan gizi, dan karakteristik keempat algoritma yang digunakan. BAB III menjelaskan formulasi masalah, rancangan eksperimen, serta implementasi sistem. BAB IV menyajikan hasil eksperimen dan analisis kinerja algoritma, sedangkan BAB V memuat kesimpulan dan arah pengembangan penelitian selanjutnya.

II. LANDASAN TEORI

A. Program Makan Bergizi Gratis dan Standar Gizi Indonesia

Program Makan Bergizi Gratis (MBG) merupakan program intervensi gizi nasional yang diselenggarakan oleh Badan Gizi Nasional (BGN) sejak Januari 2025. Dalam pelaksanaannya, program ini didukung oleh dapur Satuan Pelayanan Pemenuhan Gizi (SPPG) yang berperan sebagai unit operasional utama. Setiap SPPG melayani sekitar 3.000 porsi makanan per hari bagi sekolah-sekolah di wilayah sekitarnya [3]. Selain memastikan ketersediaan makanan, setiap SPPG juga bertanggung jawab menyusun siklus menu mingguan dengan tetap memperhatikan keterbatasan anggaran bahan pangan, yang ditetapkan sekitar Rp10.000 per porsi [3].

Standar kebutuhan gizi masyarakat Indonesia mengacu pada Peraturan Menteri Kesehatan Nomor 28 Tahun 2019 tentang Angka Kecukupan Gizi (AKG) yang Dianjurkan untuk Masyarakat Indonesia [14]. Berdasarkan regulasi tersebut, anak usia sekolah dasar kelompok umur 7-9 tahun memerlukan asupan energi sebesar 1.650 kkal dan protein 40 gram per hari. Sementara itu, kelompok umur 10-12 tahun membutuhkan sekitar 1.900-2.000 kkal energi dan 50-55 gram protein per hari [14]. Karena MBG hanya menyediakan satu kali makan dalam sehari, umumnya berupa makan siang, porsi yang disajikan diharapkan mampu memenuhi sekitar 30-35% kebutuhan gizi harian peserta didik [16]. Dengan demikian, satu porsi MBG idealnya menyediakan sekitar 500-580 kkal energi dan 12-14 gram protein untuk kelompok usia 7-9 tahun, serta turut berkontribusi terhadap pemenuhan zat gizi mikro, seperti zat besi dan vitamin A.

Data komposisi gizi bahan pangan Indonesia tersedia dalam Tabel Komposisi Pangan Indonesia (TKPI) yang diterbitkan Kementerian Kesehatan [13]. TKPI memuat kandungan energi, protein, lemak, karbohidrat, serta zat gizi mikro untuk lebih dari 1.000 jenis bahan pangan per 100 gram berat dapat dimakan. Dalam makalah ini, TKPI menjadi sumber data utama untuk membangun himpunan kandidat menu, sedangkan AKG menjadi dasar penentuan nilai batas kendala gizi.

B. Knapsack Problem dan Variannya

1) 0/1 Knapsack Problem

0/1 Knapsack Problem adalah masalah optimasi kombinatorial klasik: diberikan n objek, masing-masing dengan nilai v_j dan bobot w_j , serta sebuah knapsack berkapasitas W , tujuannya adalah memilih subhimpunan objek yang memaksimalkan total nilai tanpa melampaui kapasitas [15]. Secara formal:

$$\begin{aligned} &\text{maksimasi } \sum_j v_j \cdot x_j \\ &\text{dengan kendala } \sum_j v_j \cdot x_j \leq W, \quad x_j \in \{0, 1\} \end{aligned}$$

Istilah “0/1” menyimbolkan bahwa setiap objek hanya dapat dipilih utuh atau tidak sama sekali, berbeda dengan varian fractional knapsack yang memperbolehkan pengambilan sebagian objek dan dapat diselesaikan optimal oleh algoritma greedy [17].

2) Multidimensional Knapsack Problem

Multidimensional Knapsack Problem (MKP) menggeneralisasi knapsack klasik ke m dimensi kendala: setiap objek j memiliki vektor bobot $(w_{1j}, w_{2j}, \dots, w_{mj})$ dan terdapat m kapasitas $(c_1, \dots, c_m)$ yang seluruhnya harus dipenuhi [15]:

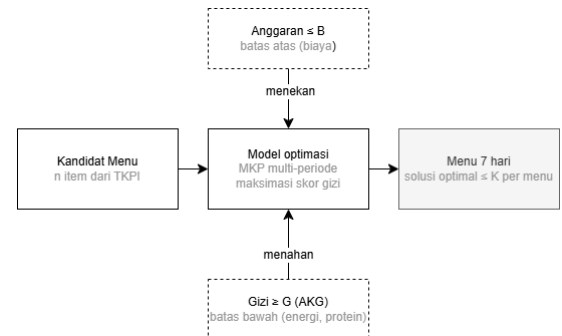
$$\text{maksimasi } \sum_j v_j \cdot x_j \text{ dengan kendala } \sum_j w_{ij} \cdot x_j \leq c_i \\ \text{untuk setiap } i = 1, \dots, m; \quad x_j \in \{0, 1\}$$

Dalam konteks perencanaan menu, setiap “dimensi” merepresentasikan satu sumber daya atau atribut yang dibatasi. Misalnya biaya, energi, protein, sehingga MKP merupakan kerangka alami untuk masalah penyusunan menu bergizi dengan anggaran terbatas.

3) Varian campuran Batas Atas-Batas Bawah dan Perluasan Multi-Periode

Knapsack Problem dan Multidimensional Knapsack Problem (MKP) standar umumnya hanya melibatkan kendala batas atas (*packing constraints*), yaitu pembatasan agar total penggunaan sumber daya tidak lebih dari kapasitas yang tersedia. Namun, permasalahan menyusun menu pada Program Makan Bergizi Gratis (MBG) memiliki karakteristik yang berbeda. Selain kendala anggaran yang bersifat batas atas (total biaya $\leq B$), terdapat pula kendala gizi yang bersifat batas bawah, seperti kebutuhan minimum energi, protein, dan zat gizi lainnya (total energi $\geq G_1$, total protein $\geq G_2$, dan seterusnya). Keberadaan kendala atas bawah ini mencerminkan tujuan utama program, yaitu menjamin terpenuhinya kebutuhan gizi minimum penerima manfaat. Dalam literatur optimasi, kendala semacam ini dikenal sebagai *demand constraints* atau *covering constraint*. Kombinasi antara *packing constraints* dan *covering constraints* menghasilkan suatu varian masalah yang dikenal sebagai *mixed packing-covering knapsack* [15], [18].

Selain itu, proses perencanaan menu MBG tidak dilakukan untuk satu periode saja, melainkan untuk satu siklus menu mingguan yang terdiri dari tujuh hari. Kondisi ini menimbulkan kendala tambahan berupa keterkaitan antar-periode, khususnya dalam aspek keberagaman menu. Untuk menjaga variasi, suatu menu yang sama sebaiknya tidak disajikan lebih dari K kali selama satu minggu. Adanya kendala variasi antar-hari tersebut menjadikan model yang digunakan lebih kompleks dibandingkan MKP standar yang hanya mempertimbangkan satu periode pengambilan keputusan. Oleh karena itu, penelitian ini menggunakan formulasi multi-periode dengan kendala keterkaitan antar-periode, yang akan dijelaskan secara rinci pada bagian III.



Gambar 1. Skema Formulasi Masalah
(Solusi terjepit: anggaran membatasi dari atas, kecukupan gizi dituntut dari bawah)

C. Teori P, NP, dan NP-Hardness

Kelas P memuat masalah keputusan yang dapat diselesaikan dalam waktu polynomial oleh mesin deterministik, sedangkan kelas NP adalah masalah keputusan yang solusinya dapat diverifikasi dalam waktu polinomial. Sementara itu, masalah disebut NP-hard apabila ia setidaknya sesulit masalah NP-complete, tanpa harus berada di NP [19].

Versi keputusan dari 0/1 Knapsack Problem telah terbukti NP-complete melalui reduksi dari Subset-Sum Problem [20]. Meskipun demikian, permasalahan ini tergolong weakly NP-complete karena masih memiliki algoritma dynamic programming dengan kompleksitas waktu $O(n \cdot W)$, yang bersifat pseudo-polinomial. Artinya, kompleksitas tersebut polinomial terhadap nilai numerik kapasitas W , tetapi tidak terhadap ukuran representasi biner dari W [15]. Selain itu, knapsack satu dimensi juga memiliki *Fully Polynomial-Time Approximation Scheme* (FPTAS), yaitu skema aproksimasi yang mampu menghasilkan solusi dengan nilai minimal $(1-\epsilon)$ kali solusi optimal dalam waktu polinomial terhadap n dan $1/\epsilon$ [15].

Karakteristik tersebut tidak berlaku pada generalisasi multidimensi. Multidimensional Knapsack Problem (MKP) dengan dua atau lebih kendala ($m \geq 2$) diketahui bersifat *strongly NP-hard* dan tidak memiliki FPTAS kecuali jika $P = NP$ [15], [18]. Konsekuensinya, tidak tersedia metode aproksimasi yang sekaligus efisien dan memiliki jaminan kualitas solusi sebagaimana pada knapsack satu dimensi. Dalam konteks perencanaan menu MBG yang melibatkan berbagai kendala gizi dan biaya secara bersamaan, kondisi ini menjadikan pencarian solusi optimal sebagai permasalahan yang secara komputasional menantang. Oleh karena itu, pendekatan yang dapat digunakan terbagi menjadi dua kategori, yaitu metode eksak yang menjamin optimalitas, namun memerlukan biaya komputasi tinggi, seperti brute force, dynamic programming, dan branch and bound, serta metode heuristik seperti greedy yang lebih cepat tapi tidak menjamin solusi optimal. Perbandingan karakteristik kedua kelompok metode tersebut menjadi fokus analisis pada bagian IV.

D. Algoritma Brute force

Algoritma brute force menyelesaikan masalah dengan mengenumerasi seluruh kandidat solusi secara eksplisit dan memilih yang terbaik di antara yang layak [17]. Untuk knapsack dengan n objek, terdapat 2^n subhimpunan yang harus diperiksa. Pada setiap subhimpunan, verifikasi m kendala membutuhkan waktu $O(n \cdot m)$, sehingga kompleksitas totalnya $O(2^n \cdot n \cdot m)$. Pada masalah multi-periode 7 hari, ruang pencarian membengkak menjadi hasil kali ruang solusi tiap hari, sehingga brute force hanya layak untuk instans sangat kecil.

Meskipun tidak praktis, brute force memiliki peran metodologis penting dalam makalah ini, yaitu sebagai penghasil ground truth (solusi optimal pasti) untuk memvalidasi kebenaran implementasi DP dan B&B pada instans kecil.

E. Algoritma Greedy

Algoritma greedy membangun solusi secara bertahap dengan memilih, pada setiap langkah, opsi yang tampak terbaik secara lokal tanpa mempertimbangkan konsekuensi global [17]. Untuk knapsack klasik, heuristik greedy yang umum digunakan adalah dengan mengurutkan objek berdasarkan rasio nilai per bobot (v_j / w_j) dan memasukkan objek satu per satu selama kapasitas masih tersedia. Strategi ini optimal untuk fractional knapsack, tetapi tidak menjamin optimal pada 0/1 knapsack. Solusinya dapat terjebak pada pilihan lokal yang menutup kombinasi global yang lebih baik [17].

Pada MKP, tidak ada satu rasio yang alami karena bobot bersifat multi-dimensi. Pendekatan yang umum adalah rasio agregat, misalnya efisiensi $v_j / \sum_i (w_{ij} / c_i)$, yang menormalkan konsumsi tiap kendala terhadap kapasitasnya [15]. Pada formulasi dengan kendala batas bawah gizi, heuristik greedy perlu dimodifikasi lebih jauh, misalnya memprioritaskan item dengan kepadatan gizi per rupiah tertinggi hingga seluruh batas bawah terpenuhi. Kompleksitas greedy didominasi pengurutan, yaitu $O(n \log n)$, menjadikannya pendekatan tercepat, dan pertanyaan utamanya adalah seberapa jauh kualitas solusinya dari optimal, yang diukur sebagai optimality gap pada eksperimen.

F. Dynamic Programming

Dynamic programming (DP) menyelesaikan masalah optimasi dengan memecahnya menjadi submasalah yang saling tumpang tindih (*overlapping subproblems*) dan memanfaatkan prinsip optimasi Bellman [21]. Pada 0/1 Knapsack Problem, relasi rekurens yang digunakan adalah

$$f(j, w) = \max(f(j-1, w), f(j-1, w - w_j) + v_j)$$

dengan $f(j, w)$ merupakan nilai optimal menggunakan j objek pertama dan kapasitas w . Tabel berukuran $n \times w$ diisi secara bottom-up menghasilkan kompleksitas waktu dan memori ($O(n \cdot W)$), yang tergolong pseudo-polinomial [15], [17].

Pada MKP, setiap kendala tambahan menambah dimensi tabel menjadi: $f(j, w_1, w_2, \dots, w_m)$, sehingga

kompleksitasnya menjadi $O(n \cdot W_1 \cdot W_2 \cdot \dots \cdot W_m)$. Pertumbuhan ini dikenal sebagai *curse of dimensionality* karena kebutuhan memori dan waktu meningkat sangat cepat seiring bertambahnya jumlah kendala [21]. Oleh karena itu, implementasi DP untuk perencanaan menu umumnya memerlukan diskretisasi nilai biaya dan gizi, yang menciptakan trade-off antara efisiensi komputasi dan akurasi solusi.

G. Branch and Bound

Branch and Bound (B&B) merupakan metode eksak yang mengeksplorasi ruang solusi dalam bentuk pohon pencarian dan memangkas cabang yang tidak berpotensi menghasilkan solusi lebih baik daripada solusi terbaik yang telah ditemukan [15], [17]. Pada knapsack, setiap simpul merepresentasikan keputusan memilih atau tidak memilih suatu objek, sedangkan fungsi bounding digunakan untuk menghitung batas atas nilai yang masih mungkin dicapai dari simpul tersebut.

Batas atas umumnya diperoleh dari relaksasi *fractional knapsack*, yaitu dengan melonggarkan kendala integralitas sehingga dapat dihitung secara cepat menggunakan pendekatan greedy [15], [17]. Selain itu, pada masalah perencanaan menu, simpul juga dapat dipangkas apabila kebutuhan gizi minimum tidak mungkin lagi dipenuhi oleh item yang tersisa. Meskipun kompleksitas terburuknya tetap ($O(2^n)$), branch and bound sering kali jauh lebih efisien daripada brute force dan lebih hemat memori dibandingkan DP multidimensi pada banyak instans praktis [17].

III. FORMULASI MASALAH DAN RANCANGAN EKSPERIMEN

A. Formulasi Masalah

Perancangan menu MBG selama satu siklus mingguan dimodelkan sebagai perakitan menu harian dari bahan pangan, dengan setiap hari dipilih satu sumber karbohidrat, satu hingga dua sumber protein, satu hingga dua sayur, dan satu buah, dengan tujuan total biaya bahan minimum sementara seluruh kebutuhan gizi tetap terpenuhi. Pendekatan “paket menu utuh” dipilih karena mencerminkan praktik nyata di dapur SPPG, setiap hari disajikan satu paket lengkap (karbohidrat, lauk protein, sayur, dan buah), sekaligus menyederhanakan ruang masalah dibandingkan perakitan dari bahan mentah.

1) Notasi

Bahan pangan dikelompokkan ke dalam empat kategori: karbohidrat (C), protein (P), sayur (S), dan buah (F). Misalkan himpunan hari dalam satu siklus dinyatakan sebagai $T = \{1, 2, \dots, d\}$, dengan $d = 7$ untuk siklus mingguan. Setiap bahan j memiliki atribut biaya b_j (rupiah, sesuai berat porsi) dan vektor kandungan gizi ($g_{1j}, g_{2j}, \dots, g_{mj}$) untuk m jenis nutrisi (energi, protein, lemak, karbohidrat).

Setiap hari, satu menu harian dirakit dengan memilih tepat satu bahan dari kategori karbohidrat, satu hingga dua bahan dari kategori protein, satu hingga dua bahan dari

kategori sayur, dan satu bahan dari kategori buah. Variabel keputusan didefinisikan sebagai:

$$x_{jt} = 1 \text{ jika bahan } j \text{ dipilih untuk menu pada hari } t, \text{ dan } 0 \text{ jika tidak.}$$

2) Fungsi Objektif

Penelitian ini menguji dua mode objektif yang merepresentasikan dua tujuan kebijakan berbeda dalam penyelenggaraan MBG.

Mode pertama: minimasi biaya. Tujuannya meminimalkan total biaya bahan sepanjang siklus, dengan kecukupan gizi sebagai batas bawah yang wajib dipenuhi.

$$\text{minimasi } Z = \sum_t \sum_j b_j \cdot x_{jt}$$

Mode kedua: maksimasi gizi. Tujuannya memaksimalkan total skor pemenuhan gizi, dengan anggaran sebagai batas atas yang tidak boleh dilampaui.

$$\text{maksimasi } Z = \sum_t \sum_j s_j \cdot x_{jt}$$

dengan s_j adalah skor pemenuhan AKG menu, yang dibatasi maksimal 100% per nutrisi sehingga kelebihan gizi tidak memberi nilai tambah. Mode ini mencerminkan tujuan maksimalisasi mutu gizi dalam batas anggaran yang tersedia.

3) Kendala

- (1) Satu paket per hari, setiap hari tepat satu paket menu disajikan.

$$x_{jt} = 1 \text{ jika paket } j \text{ dipilih pada hari } t, \text{ dan } 0 \text{ jika tidak.}$$

- (2) Kecukupan gizi (batas bawah, demand/covering constraint), paket pada setiap hari harus memenuhi minimal target gizi makan siang G_i untuk setiap nutrisi i :

$$\sum_j g_{ij} \cdot x_{jt} \geq G_i, \text{ untuk setiap } i = 1, \dots, m \text{ dan setiap } t \in T$$

- (3) Anggaran (batas atas, packing constraint), biaya paket pada setiap hari tidak melebihi anggaran per porsi B :

$$\sum_j b_j \cdot x_{jt} \leq B, \text{ untuk setiap } t \in T$$

- (4) Keragaman antar-hari (kendala multi-periode), Satu paket yang sama tidak boleh disajikan lebih dari K kali dalam satu siklus:

$$\sum_t x_{jt} \leq K, \text{ untuk setiap } j \in M$$

- (5) Integralitas:

$$x_{jt} \in \{0, 1\}$$

Kendala (2) yang bersifat batas bawah menjadikan masalah ini varian *covering* dari keluarga knapsack, sedangkan kendala (3) tetap bersifat packing, kombinasi keduanya membentuk masalah *mixed packing-covering*. Kendala (4), masalah akan terurai menjadi d submasalah harian yang identik dan dapat diselesaikan secara terpisah. Formulasi gabungan multi-dimensi multi-periode dengan struktur packing-covering inilah yang menjadi kontribusi pemodelan dalam makalah ini.

B. Konstruksi Dataset

Data menu MBG pada tingkat per-sekolah tidak tersedia sebagai dataset terbuka yang terstandar. Karena itu, dataset kandidat paket menu dikonstruksi secara realistis dari sumber resmi dengan menjaga keterlacakan setiap nilai: kandungan gizi diturunkan dari Daftar Bahan Makanan

Penukar dan Tabel Komposisi Pangan Indonesia [13], [22]. Biaya bahan dihitung dari harga komoditas yang dipublikasikan Panel Harga Badan Pangan Nasional [23]. Nilai batas gizi G_i ditetapkan 30% dari AKG harian sesuai Permenkes No. 28 Tahun 2019 [14], [16] (energi ± 650 kkal, protein ± 19 g untuk anak usia sekolah), dan anggaran B ditetapkan Rp10.000 per porsi sesuai ketentuan BGN [3].

Setiap paket merupakan kombinasi empat komponen yaitu, karbohidrat, protein, sayur, dan buah yang disusun menyerupai menu nyata di lapangan. Himpunan paket layak dibentuk melalui enumerasi seluruh kombinasi satu karbohidrat, satu hingga dua protein, satu hingga dua sayur, dan satu buah yang memenuhi target gizi dan anggaran. Dari himpunan ini, n kandidat dipilih secara round-robin per pasangan (karbohidrat, protein) agar keragaman kombinasi tetap terjaga meski n kecil. Jumlah paket kandidat n divariasikan pada eksperimen. Tabel I menampilkan contoh sebagian paket kandidat beserta atributnya.

TABEL I. CONTOH DATASET BAHAN PANGAN

Kode	Bahan	Kategori	Energi (kkal)	Protein (g)	Lemak (g)	Karbo (g)	Harga (Rp/kg)
B01	Beras/Nasi putih	Karbohidrat	360	6,8	0,7	78,9	13.000
B03	Nasi jagung	Karbohidrat	345	8,0	1,5	73,0	11.000
B11	Ayam tanpa kulit	Protein hewani	170	18,2	10,0	0,0	32.000

C. Rancangan Algoritma

Implementasi Brute force pada Algoritma 1 merekursi per hari, mencoba setiap paket kandidat sebagai pilihan hari tersebut sebelum melangkah ke hari berikutnya (backtracking). Kendala keragaman ditegakkan melalui larik `usage[j]` yang menghitung pemakaian setiap paket, sedangkan kendala protein hewani dicek hanya pada simpul daun ($\text{hari} = 7$) sehingga seluruh kombinasi yang melanggar kuota otomatis tidak terhitung sebagai solusi.

```
function BruteForce(menu, mode):
    best ← 0; bestBobot ← ∞
    function rekursi(hari, akum, hewani):
        // Basis: satu siklus 7 hari sudah lengkap
        if hari = 7:
            // Terima hanya jika kuota protein hewani terpenuhi
            if hewani ≥ MIN_HEWANI and akum < bestBobot:
                bestBobot ← akum; best ← pilihan
            return
        // Coba SEMUA kemungkinan menu untuk hari ini (enumerasi penuh)
        for tiap menu j:
            // Lewati jika langgar batas variasi atau aturan berurutan
            if usage[j] < K and not langgarBerurutan(j, hari):
                pilih j; usage[j]++
                rekursi(hari+1, akum + bobot(j, mode), hewani + hewani(j))
                batalkan j; usage[j]-- // backtrack
        rekursi(0, 0, 0)
    return best
```

Implementasi Dynamic Programming pada Algoritma 2 merepresentasikan state sebagai tupel (vektor pemakaian tiap paket, karbohidrat hari sebelumnya, protein hari sebelumnya, jumlah hari hewani sejauh ini), karena keempat informasi tersebut diperlukan untuk menegakkan kendala keragaman K , aturan tidak berurutan, dan kuota protein hewani secara simultan. Tabel DP diperbarui maju

per hari, hanya menyimpan bobot minimum untuk setiap state yang sama (memanfaatkan struktur overlapping subproblems). Akibatnya, ukuran ruang state tumbuh kombinatorial terhadap K , menjadikan implementasi ini rawan (*curse of dimensionality*).

```
function DynamicProgramming(menu, mode):
    // State menyimpan riwayat agar semua kendala bisa
    ditegakkan
    state ← (vektor_pemakaian, karbo_terakhir,
    protein_terakhir, jumlah_hewani)
    dp[state_awal] ← bobot 0
    // Bangun solusi maju, hari demi hari
    for hari = 0 to 6:
        dp_baru ← 0
        // Perluas setiap state yang sudah ada dengan setiap
        menu yang mungkin
        for tiap (state, akum) di dp:
            for tiap menu j:
                // Saring: batas variasi, aturan berurutan, dan
                kelayakan kuota hewani
                if usage[j] < K
                and karbo(j) ≠ karbo_terakhir
                and protein(j) ≠ protein_terakhir
                and (jumlah_hewani + hewani(j) + sisa_hari)
                ≥ MIN_HEWANI:
                    state_baru ← perbarui(state, j)
                    // Simpan hanya bobot minimum untuk tiap
                    state (overlapping subproblem)
                    simpan bobot minimum ke dp_baru[state_baru]
                    dp ← dp_baru
        // Ambil state akhir terbaik yang memenuhi kuota hewani
        return state dengan bobot minimum dan jumlah_hewani ≥
        MIN_HEWANI
```

Implementasi Branch and Bound pada Algoritma 3 mempertahankan dua kriteria pemangkasan spesifik untuk perencanaan menu MBG: pertama, batas bawah optimistik dihitung dari bobot satuan terbaik di seluruh kandidat dikalikan sisa hari, sehingga cabang yang pasti tidak dapat mengalahkan solusi terbaik saat ini langsung dibuang. Kedua, cabang dibuang lebih awal jika kuota protein hewani mustahil tercapai dengan sisa hari yang ada. Kedua pemangkasan ini diterapkan sebelum percabangan dilakukan, sehingga ruang pencarian yang benar-benar dieksplorasi jauh lebih kecil dari 2^n .

```
function BranchAndBound(menu, mode):
    best ← 0; bestBobot ← ∞
    terbaikSatuan ← bobot menu terbaik // untuk hitung batas
    bawah optimistik
    function rekursi(hari, akum, hewani):
        // PRUNING 1 (bound): jika estimasi terbaik pun tak
        bisa kalahkan solusi terbaik, potong
        batasBawah ← akum + (7 - hari) × terbaikSatuan
        if batasBawah ≥ bestBobot: return
        // PRUNING 2 (kelayakan): jika kuota hewani mustahil
        tercapai, potong
        if hewani + (7 - hari) < MIN_HEWANI: return
        // Basis: siklus lengkap
        if hari = 7:
            if hewani ≥ MIN_HEWANI: bestBobot ← akum; best ←
            pilihan
            return
        // Cabangkan: pilih satu menu untuk hari ini
        for tiap menu j:
            if usage[j] < K and not langgarBerurutan(j, hari):
                pilih j; usage[j]++
                rekursi(hari+1, akum + bobot(j, mode), hewani
                + hewani(j))
                batalkan j; usage[j]-- // backtrack
        rekursi(0, 0, 0)
    return best
```

Implementasi Greedy pada Algoritma 4 memilih, untuk setiap hari, paket berbobot terbaik pertama dari daftar yang sudah diurutkan, dengan satu modifikasi penting: ketika sisa hari sama dengan sisa kuota protein hewani yang belum terpenuhi, algoritma memaksa pemilihan paket

hewani agar kuota minimum tidak terlewat di akhir siklus. Mekanisme paksa-hewani ini diperlukan karena pengurutan murni berdasarkan bobot tidak menjamin kuota gizi tercapai pada formulasi mixed packing-covering.

```
function Greedy(menu, mode):
    // Urutkan menu dari "paling menguntungkan" lebih dulu
    urutkan menu berdasarkan bobot(mode) menaik
    for hari = 0 to 6:
        // Cek apakah kuota hewani memaksa hari ini harus menu
        hewani
        sudahHewani ← jumlah menu hewani yang sudah dipilih
        wajibHewani ← (MIN_HEWANI - sudahHewani) ≥ (7 - hari)
        // Ambil pilihan lokal terbaik pertama yang valid
        (tanpa menengok ke depan)
        pilih j pertama (terurut) dengan:
            usage[j] < K, not langgarBerurutan(j, hari),
            dan jika wajibHewani maka j harus mengandung
        hewani
        if tidak ada j: return INFEASIBLE
        tetapkan j untuk hari ini; usage[j]++
        // Validasi akhir: pastikan kuota hewani benar-benar
        tercapai
        if total menu hewani < MIN_HEWANI: return INFEASIBLE
        return pilihan
```

D. Rancangan Eksperimen

1) Metrik Evaluasi

Lima besaran dicatat untuk setiap kombinasi algoritma dan item uji:

1. Waktu eksekusi: durasi murni komputasi algoritma (tidak termasuk waktu baca dataset), diukur dengan pewaktu resolusi tinggi dan dilaporkan dalam milidetik. Setiap konfigurasi dijalankan sebanyak r kali (mis. $r = 10$) dan diambil reratanya untuk mengurangi noise dari penjadwalan sistem operasi.
2. Biaya solusi: total biaya bahan rencana menu 7 hari (rupiah) yang dihasilkan tiap algoritma. Karena objektifnya minimasi biaya, nilai lebih rendah menandakan solusi lebih baik.
3. Optimality gap: selisih relatif biaya solusi greedy terhadap biaya optimal (dari algoritma eksak), dihitung sebagai $(\text{biaya_greedy} - \text{biaya_optimal}) / \text{biaya_optimal} \times 100\%$. Metrik ini mengukur "harga" dari kecepatan greedy.
4. Penggunaan memori puncak: memori maksimum yang dipakai selama eksekusi, dipantau untuk menyoroti ledakan ruang state pada dynamic programming.
5. Status kelayakan: apakah suatu instans memiliki solusi layak (feasible) atau tidak (infeasible), penting untuk skenario anggaran ketat.

2) Variabel Bebas dan Skenario Uji

Tiga variabel divariasikan secara terkontrol, saat satu variabel diuji, lainnya ditahan pada nilai default (ditandai *):

TABEL II. RANCANGAN VARIABEL DAN SKENARIO EKSPERIMEN

Variabel	Nilai yang diuji	Tujuan
Jumlah paket kandidat (n)	10, 15, 20*, 25, 30	Mengukur skalabilitas dan pertumbuhan waktu/memori

Batas keragaman (K)	1, 2*, 3, 4	Melihat pengaruh kekuatan kendala multi-periode
Anggaran (B)	Rp5.000, Rp7.000, Rp8.000, Rp10.000*	Mengamati titik transisi feasible → infeasible

Penjelasan tiap variasi:

1. Skalabilitas (n): ukuran masalah dinaikkan bertahap. Brute force diperkirakan tidak terukur di atas ambang tertentu (waktu melampaui batas wajar), dan titik putus ini sendiri merupakan temuan. Algoritma lain diamati pertumbuhan waktunya.
2. Variasi anggaran (B): nilai anggaran per porsi diturunkan bertahap (Rp5.000 hingga Rp10.000) untuk menguji titik kelayakan. Pada anggaran sangat ketat, masalah menjadi infeasible karena tidak ada kombinasi menu yang memenuhi gizi minimum dalam batas biaya tersebut.
3. Kekuatan kendala (K): pada $K = 1$ setiap paket hanya boleh muncul sekali (paling ketat, butuh ≥ 7 paket layak berbeda), sedangkan K besar melonggarkan kendala. Variasi ini mengukur seberapa besar kendala multi-periode mempersulit masalah.
4. Titik infeasibility (B): pada anggaran ketat, sebagian instans menjadi tak punya solusi. Melaporkan pada anggaran berapa AKG tak lagi terpenuhi adalah hasil yang punya makna kebijakan.

3) Pengendalian dan Validasi

Untuk menjamin perbandingan adil, keempat algoritma menerima dataset, nilai constraint, dan instans yang identik pada setiap pengujian. Kebenaran solusi algoritma dynamic programming, branch and bound, dan greedy divalidasi dengan membandingkan keluarannya terhadap brute force pada instans kecil ($n \leq 15$), di mana brute force dijamin menghasilkan solusi optimal. Algoritma eksak (brute force, DP, B&B) harus menghasilkan biaya optimal yang sama. Bila berbeda, terdapat kesalahan implementasi yang diperbaiki sebelum eksperimen skala penuh.

4) Lingkungan Eksperimen

Seluruh algoritma diimplementasikan dalam bahasa C++. Eksperimen dijalankan pada laptop dengan prosesor Intel Core i5-1135G7, RAM 8 GB, sistem operasi Windows 11, menggunakan kompiler g++ (MSYS2 UCRT64) dengan tingkat optimasi -O2 dan standar C++17. Untuk menjaga konsistensi, eksperimen dijalankan tanpa proses berat lain yang berjalan bersamaan. Sesuai ketentuan tugas, spesifikasi teknis dan struktur kode program tidak dirinci dalam makalah. Implementasi lengkap tersedia pada tautan repositori di bagian akhir.

IV. HASIL DAN ANALISIS

A. Validasi Kebenaran

Kebenaran logika seluruh algoritma divalidasi dengan membandingkan keluaran keempat algoritma pada instans yang sama. Pada seluruh instans layak, brute force, dynamic programming, branch and bound, dan greedy menghasilkan total biaya optimal yang identik. Menggunakan parameter default (anggaran $B = \text{Rp}10.000/\text{porsi}$, target energi makan siang ≥ 650 kkal, protein ≥ 19 g, batas variasi $K = 2$, dan minimal tiga hari mengandung protein hewani), program menyusun siklus menu pada Tabel III.

TABEL III. RENCANA MENU MINGGUAN HASIL OPTIMASI BRANCH & BOUND ($N=25$, $K=2$, $B=\text{Rp}10.000$)

Hari	Karbohidrat	Protein	Sayur	Buah	Biaya
Senin	Nasi putih	Tahu	Daun singkong	Semangka	Rp4.755
Selasa	Nasi merah	Telur ayam	Daun singkong	Semangka	Rp6.180
Rabu	Nasi putih	Tahu	Daun singkong	Semangka	Rp4.755
Kamis	Nasi merah	Ikan lele	Daun singkong	Pepaya	Rp6.120
Jumat	Nasi putih	Oncom	Kangkung	Pepaya	Rp4.680
Sabtu	Nasi merah	Ikan lele	Daun singkong	Pepaya	Rp6.120
Minggu	Nasi putih	Oncom	Kangkung	Pepaya	Rp4.680



Gambar II. Ilustrasi Menu Mingguan Hasil Optimasi Branch & Bound ($N=25$, $K=2$, $B=\text{Rp}10.000$)

- Total biaya seminggu: Rp37.290 (rata-rata $\approx \text{Rp}5.327/\text{porsi}$)
- Seluruh hari memenuhi target gizi: energi ≥ 650 kkal, protein ≥ 19 g (skor pemenuhan 100%)

Analisis Kelayakan (Feasibility): menu di atas memenuhi semua kendala (mixed-packing covering) karena:

1. Zat gizi harian berada di atas batas minimum AKG
2. Biaya komponen per hari tidak melampaui batas anggaran Rp10.000
3. Tidak ada paket menu yang muncul lebih dari $K = 2$ kali dalam satu siklus.

B. Kelayakan dan Pengaruh Ukuran Kandidat (n)

Pada jumlah kandidat kecil ($n \leq 20$), masalah berstatus tidak layak (infeasible). Hal ini terjadi karena paket menu murah yang tersedia cenderung menggunakan basis karbohidrat atau lauk yang sama, sehingga langsung melanggar batasan keragaman berurutan dan batas variasi harian ($K = 2$). Masalah baru memiliki solusi layak ketika variasi ruang kandidat dinaikkan ke $n \geq 25$.

Secara praktis, temuan ini menunjukkan bahwa keterbatasan ragam bahan pangan di lapangan secara matematis dapat membuat penyusunan menu yang layak menjadi mustahil.

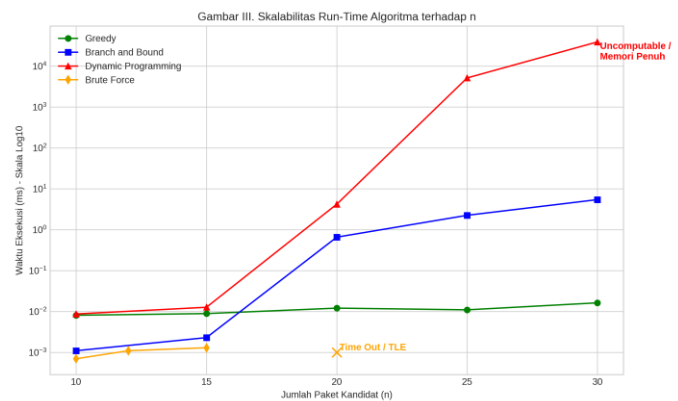
C. Skalabilitas dan Waktu Eksekusi (Run-Time)

Pengujian performa menunjukkan perbedaan waktu eksekusi yang drastis seiring bertambahnya ukuran kandidat (n).

TABEL IV. METRIK PERBANDINGAN RUN-TIME (MS) TERHADAP UKURAN MASALAH (N)
(Kondisi Pengujian : K = 2, Anggaran B = 10.000 per porsi, Mode: MIN BIAYA)

Jumlah Paket (n)	Status Kelayakan	Brute Force (ms)	Dynamic Programming (ms)	B&B (ms)	Greedy (ms)
n = 10	Infeasible	0,0007	0,0087	0,0011	0,0081
n = 12	Infeasible	0,0011	0,0123	0,0020	0,0085
n = 15	Infeasible	0,0013	0,0127	0,0023	0,0089
n = 20	Infeasible	Time Out (-1,0)	4,2075	0,6577	0,0121
n = 25	Feasible	Uncomputable	5.141,7319	2,2501	0,0110
n = 30	Feasible	Uncomputable	39.018,6666	5,4379	0,0163
n = 40	Feasible	Uncomputable	Memori Penuh (-1,0)	1,1101	0,0104
n = 50	Feasible	Uncomputable	Memori Penuh (-1,0)	1,3641	0,0081

Pada $n = 30$, DP membutuhkan waktu 39.018,67 ms (≈ 39 detik), yang berarti sekitar 7.175 kali lebih lambat dari Branch and Bound (5,44 ms) dan 2,39 juta kali lebih lambat dari Greedy (0,016 ms). Brute force mengalami *combinatorial explosion* dan langsung terhenti sejak $n \geq 20$.



Gambar III. Grafik perbandingan run-time komputasi (ms) terhadap jumlah paket kandidat (n) dengan sumbu-Y logaritmik.

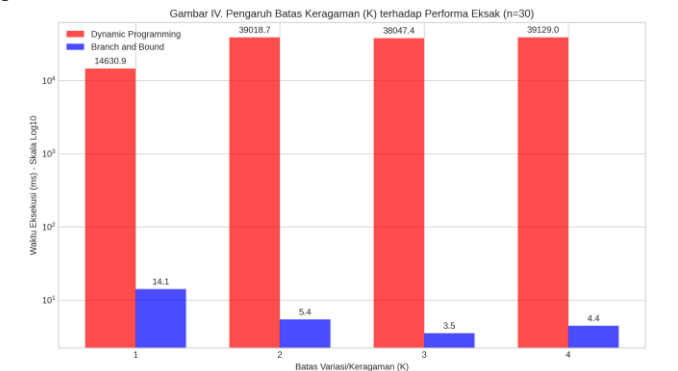
D. Ledakan Ruang State pada Dynamic Programming

Meskipun menjamin solusi eksak, DP tidak praktis untuk masalah multi-periode karena mengalami *Curse of Dimensionality*. Untuk menegakkan kendala keragaman, state DP harus mencatat riwayat pemakaian setiap paket menu, sehingga ukuran matriks memori membesar secara kombinatorial terhadap nilai K:

TABEL V. DAMPAK VARIASI KERAGAMAN (K) TERHADAP KOMPLEKSITAS EKSPLORASI (N=30)

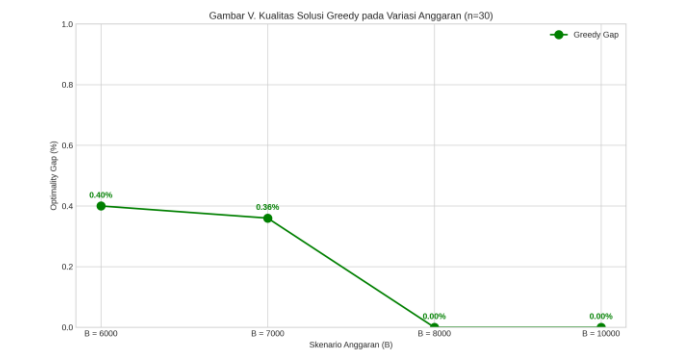
Batas Keragaman (K)	Run-Time DP (ms)	Run-Time B&B (ms)	Dampak pada Eksplorasi State
K = 1 (Sangat Ketat)	14.630,8930	14,1022	Ruang state dipersempit kendala ketat
K = 2 (Default)	37.443,1336	5,3356	Ruang state moderat
K = 3 (Longgar)	38.047,4282	3,5239	Ukuran state membesar
K = 4 (Sangat Longgar)	39.129,0201	4,4410	Ledakan kombinatorial tabel memo DP

Ketika K melonggar dari 1 ke 4, runtime DP melonjak dari 14.630,89 ms ke 39.129,02 ms karena sejumlah sub-masalah yang tumpang tindih bertambah masif di dalam tabel memo. Sebaliknya, B&B justru melandai hingga kisaran 3-4 ms karena pelonggaran kendala mempercepat fungsi bounding dalam memangkas cabang pohon keputusan yang tidak prospektif.



Gambar IV. Grafik pengaruh batas keragaman (K) terhadap run-time algoritma DP dan B&B.

E. Kualitas Solusi dan Optimality Gap Greedy



Gambar V. Grafik Kualitas Solusi Greedy pada Variasi Anggaran (n=30)

Pada pengujian mode minimasi biaya dengan anggaran longgar ($B = \text{Rp}8.000$ dan $\text{Rp}10.000$), algoritma greedy mencapai *optimality gap* 0,00%, solusinya sama dengan optimum dari metode eksak. Namun, ketika anggaran diperketat menjadi $\text{Rp}7.000$, greedy mulai menyimpang dari optimum dengan gap sebesar 0,36% (menghasilkan $\text{Rp}37.290$ dibanding solusi optimal $\text{Rp}37.155$). Penyimpangan ini semakin besar ketika anggaran diperketat lagi menjadi $\text{Rp}6.000$, dengan gap mencapai 0,40% (greedy menghasilkan $\text{Rp}33.840$, sedangkan solusi optimal hanya $\text{Rp}33.705$). Pada $\text{Rp}5.000$, ruang solusi sudah tidak lagi memungkinkan tercapainya kelayakan sama sekali bagi keempat algoritma. Pola ini menunjukkan bahwa keandalan strategi greedy bergantung pada keluasan ruang solusi. Ketika ruang menyempit akibat anggaran ketat, pendekatan greedy dapat terjebak pada pilihan lokal yang bukan optimum global.

F. Karakteristik Eksperimen Mode Maksimasi Gizi

Pengujian pada mode MAX_GIZI (anggaran sebagai batas atas) menghasilkan status *infeasible* pada $n \leq 20$, dan baru menemukan solusi layak pada $n \geq 25$. Berbeda dengan mode pertama, pada mode ini Greedy tetap mencapai skor gizi optimal yang identik dengan metode eksak (gap 0%) karena memilih menu berbasis skor gizi tertinggi harian tanpa penelusuran balik (*backtracking*). {ada konfigurasi yang diuji, beberapa kombinasi menu yang berbeda ternyata sama-sama mencapai skor gizi maksimum 100%, sehingga strategi greedy tidak terjebak pada optimum lokal dari sisi skor.

Pada $n = 30$, biaya greedy ($\text{Rp}54.480$) berbeda dari biaya B&B ($\text{Rp}58.815$) dan DP ($\text{Rp}62.120$), meskipun ketiganya sama-sama mencapai skor gizi optimal 700 (100%). Hal ini menunjukkan bahwa pada mode maksimasi gizi terdapat banyak solusi optimal (multiple optima): karena skor gizi dibatasi maksimal 100% per nutrisi, berbagai kombinasi menu dengan biaya berbeda dapat sama-sama mencapai skor gizi maksimum, sehingga biaya total bukan lagi pembeda optimalitas pada mode ini. Pemilihan mode objektif dengan demikian berdampak langsung pada struktur kelayakan dan karakteristik solusi yang dihasilkan, bukan hanya pada nilai fungsi objektifnya.

G. Rangkuman Komparasi Algoritma

Tabel VI merangkum *trade-off* performa keempat algoritma untuk masalah *multi-period multi-dimensional knapsack* Makan Bergizi Gratis (MBG).

TABEL VI. MATRIKS KARAKTERISTIK DAN PERFORMA PENDEKATAN ALGORITMA ($N = 30$)

Aspek Evaluasi	Brute Force	Dynamic Programming	Branch and Bound	Algoritma Greedy
Run-Time ($n=30$)	Time Out (TLE)	39.018,66 ms	5,44 ms	0,016 ms
Optimality Gap	0% (Ground Truth)	0%	0%	0,00% (naik hingga 0,40% saat anggaran ketat)
Kelebihan	Sederhana, jaminan eksak.	Optimalitas terjamin.	Hemat memori, pemangka	Eksekusi instan ($O(n \log n)$).

			san intensif.	
Kekurangan	Eksplisif (2^n).	Rentan <i>Curse of Dim.</i>	Tergantung kekuatan bound.	Risiko optimum lokal di mode kompleks.

V. KESIMPULAN DAN SARAN

Makalah ini memodelkan perencanaan menu Program Makan Bergizi Gratis (MBG) selama tujuh hari sebagai permasalahan optimasi kombinatorial multi-dimensi dan multi-periode. Empat pendekatan algoritmik, yaitu brute force, dynamic programming, branch and bound, dan greedy, dibandingkan dalam dua skenario optimasi: minimasi biaya dengan pemenuhan kebutuhan gizi sebagai kendala serta maksimasi kualitas gizi dengan batasan anggaran. Model ini juga mempertimbangkan keberagaman menu dan keberadaan protein hewani agar solusi yang dihasilkan tetap realistis untuk konsumsi anak usia sekolah.

Hasil eksperimen menunjukkan bahwa metode eksak mampu menghasilkan solusi optimal yang sama pada seluruh instans yang layak, tetapi dengan kebutuhan komputasi yang berbeda secara signifikan. Di antara metode tersebut, branch and bound memberikan keseimbangan terbaik antara optimalitas dan efisiensi, karena tetap mampu menyelesaikan instans berukuran besar dengan waktu eksekusi yang sangat rendah melalui mekanisme pemangkasan ruang pencarian.

Di sisi lain, brute force terbukti tidak cocok untuk permasalahan ini. Meskipun menjamin optimalitas dan berfungsi sebagai *ground truth* validator pada instans kecil ($n \leq 15$), algoritma ini mengalami ledakan kombinatorial yang menyebabkan *time out* sejak $n = 20$ dan sepenuhnya tidak dapat dikomputasi pada $n \geq 25$. Hal ini mengonfirmasi bahwa pendekatan enumerasi exhaustif, meski sederhana secara konsep, tidak layak digunakan secara praktis pada permasalahan optimasi berskala nyata yang bersifat NP-hard.

Sebaliknya, dynamic programming menunjukkan keterbatasan dalam penerapan praktis. Meskipun menjamin optimalitas secara teoritis, pertumbuhan ruang state yang sangat besar menyebabkan waktu eksekusi meningkat secara drastis seiring bertambahnya ukuran masalah. Temuan ini menunjukkan bahwa keunggulan teoritis suatu algoritma belum tentu sejalan dengan performanya pada permasalahan optimasi yang memiliki banyak dimensi dan kendala.

Algoritma greedy menjadi metode dengan waktu eksekusi tercepat. Pada sebagian besar konfigurasi, solusi yang dihasilkan identik atau sangat dekat dengan solusi optimal. Namun, ketika anggaran semakin ketat, kualitas solusi mulai menurun dan muncul selisih terhadap solusi optimum, sehingga menunjukkan keterbatasan pendekatan serakah dalam ruang solusi yang lebih restriktif.

Eksperimen juga mengungkap adanya batas anggaran minimum yang menentukan kelayakan solusi. Pada tingkat anggaran tertentu, tidak terdapat kombinasi menu mingguan yang mampu memenuhi kebutuhan gizi sekaligus menjaga keragaman menu. Hasil ini memberikan implikasi praktis bahwa terdapat ambang pendanaan minimum yang harus

dipenuhi agar target gizi program MBG dapat dicapai secara matematis.

Secara keseluruhan, keempat algoritma tersebut menempati posisi yang berbeda pada spektrum trade-off antara optimalitas dan efisiensi komputasi, sebagaimana lazim ditemukan pada masalah yang bersifat NP-hard. Pemilihan algoritma yang tepat bergantung pada konteks penggunaan: dapur SPPG dengan kebutuhan penyusunan menu rutin dan cepat dapat mengandalkan greedy, sedangkan keputusan kebijakan yang menuntut jaminan optimalitas, misalnya audit anggaran tahunan, lebih cocok diselesaikan dengan branch and bound.

Untuk penelitian selanjutnya, model dapat dikembangkan dengan memasukkan lebih banyak zat gizi mikro, seperti zat besi, vitamin A, dan kalsium, sehingga representasi kebutuhan gizi menjadi lebih komprehensif. Selain itu, penggunaan data harga pengadaan aktual dari pemasok MBG dapat meningkatkan akurasi estimasi biaya. Pengembangan lebih lanjut juga dapat mencakup aspek logistik, seperti optimasi distribusi dari dapur Satuan Pelayanan Pemenuhan Gizi (SPPG) ke sekolah, sehingga perencanaan yang dilakukan tidak hanya berfokus pada penyusunan menu, tetapi juga pada efisiensi rantai pasok program secara keseluruhan.

VIDEO LINK YOUTUBE

<https://youtu.be/4P4ZI6GXVtg>

LINK GITHUB

<https://github.com/andraalanna/Makalah-Mata-Kuliah-IF211-Strategi-Algoritma->

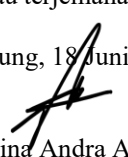
REFERENSI

- [1] Kementerian Kesehatan Republik Indonesia, "Hasil Survei Status Gizi Indonesia (SSGI) 2024," Jakarta, 2025. [Daring]. Tersedia: <https://kemkes.go.id/id/ssgi-2024-prevalensi-stunting-nasional-turun-menjadi-198>. [Diakses: 12 Juni 2026].
- [2] Badan Kebijakan Pembangunan Kesehatan, Kementerian Kesehatan RI, "Capaian SSGI 2024," Jakarta, 2025. [Daring]. Tersedia: <https://www.badankebijakan.kemkes.go.id/ssgi-2024-prevalensi-stunting-nasional-turun-menjadi-198/>. [Diakses: 12 Juni 2026].
- [3] Badan Gizi Nasional, "Tentang Program Makan Bergizi Gratis," 2025. [Daring]. Tersedia: <https://www.bgn.go.id>. [Diakses: 13 Juni 2026].
- [4] "Pemerintah pangkas anggaran MBG jadi Rp268 triliun pada 2026," ANTARA News, 2026. [Daring]. Tersedia: <https://www.antaranews.com/berita/5573624/pemerintah-pangkas-anggaran-mbg-jadi-rp268-triliun-pada-2026>. [Diakses: 13 Juni 2026].
- [5] Badan Gizi Nasional, "BGN Catat Lonjakan SPPG dan Penerima Manfaat MBG Hingga Puluhan Juta Jiwa," 2026. [Daring]. Tersedia: <https://www.bgn.go.id/news/berita/bgn-catat-lonjakan-sppg-dan-penerima-manfaat-mbg-hingga-puluhan-juta-jiwa>. [Diakses: 13 Juni 2026].
- [6] National Institute for Educational Policy Research (NIER), "School Lunch Program in Japan," Tokyo, Japan. [Daring]. Tersedia: <https://www.nier.go.jp/English/educationjapan/pdf/201303SLP.pdf>. [Diakses: 15 Juni 2026].
- [7] Ministry of Education, Government of India, "PM POSHAN Scheme," Department of School Education & Literacy. [Daring]. Tersedia: <https://dse.education.gov.in/en/scheme/pm-poshan-scheme>. [Diakses: 15 Juni 2026].
- [8] Food and Agriculture Organization, "Programa Nacional de Alimentação Escolar (PNAE), Brazil," 2023. [Daring]. Tersedia: <https://www.fao.org/brazil/programas/alimentacao-escolar-pnae>. [Diakses: 14 Juni 2026].
- [9] "Pemerintah Pangkas Anggaran MBG 2026, Menkeu Sebut Demi Efisiensi," Kompas, 2026. [Daring]. Tersedia: <https://money.kompas.com/read/2026/05/20/060719826/pemerintah-pangkas-anggaran-mbg-2026-menkeu-sebut-demi-efisiensi>. [Diakses: 12 Juni 2026].
- [10] "Program Makan Bergizi Gratis Dinilai Sarat Masalah Tata Kelola," Kompas.id, 2026. [Daring]. Tersedia: <https://www.kompas.id/artikel/en-program-makan-bergizi-gratis-dinilai-sarat-masalah-tata-kelola>. [Diakses: 14 Juni 2026].
- [11] "BGN Catat Tren Penurunan Kasus Keracunan MBG Sejak Akhir 2025," CNN Indonesia, 2026. [Daring]. Tersedia: <https://www.cnnindonesia.com/nasional/20260120122336-20-1318834/bgn-catat-tren-penurunan-kasus-keracunan-mbg-sejak-akhir-2025>. [Diakses: 13 Juni 2026].
- [12] "BGN Ungkap Kelangkaan Ahli Gizi, Minta Persagi Isi Kekosongan untuk SPPG," Detik News, 2025. [Daring]. Tersedia: <https://news.detik.com/berita/d-8223725/bgn-ungkap-kelangkaan-ahli-gizi-minta-persagi-isi-kekosongan-untuk-sppg>. [Diakses: 15 Juni 2026].
- [13] Kementerian Kesehatan Republik Indonesia, Tabel Komposisi Pangan Indonesia (TKPI). Jakarta: Kemenkes RI, 2020.
- [14] Peraturan Menteri Kesehatan Republik Indonesia Nomor 28 Tahun 2019 tentang Angka Kecukupan Gizi yang Dianjurkan untuk Masyarakat Indonesia. Jakarta: Kemenkes RI, 2019.
- [15] H. Kellerer, U. Pferschy, dan D. Pisinger, Knapsack Problems. Berlin, Heidelberg: Springer-Verlag, 2004.
- [16] Kementerian Kesehatan Republik Indonesia, Pedoman Gizi Seimbang. Jakarta: Kemenkes RI, 2014.
- [17] R. Munir, "Diktat Kuliah IF2211 Strategi Algoritma," Institut Teknologi Bandung, 2025.
- [18] A. Fréville, "The multidimensional 0–1 knapsack problem: An overview," European Journal of Operational Research, vol. 155, no. 1, hlm. 1–21, 2004.
- [19] M. R. Garey dan D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco, CA: W. H. Freeman, 1979.
- [20] R. M. Karp, "Reducibility among combinatorial problems," dalam Complexity of Computer Computations. New York: Plenum Press, 1972, hlm. 85–103.
- [21] R. Bellman, Dynamic Programming. Princeton, NJ: Princeton University Press, 1957.
- [22] Persatuan Ahli Gizi Indonesia (PERSAGI), Daftar Bahan Makanan Penukar. Jakarta: PERSAGI, 2021.
- [23] Badan Pangan Nasional, "Panel Harga Pangan," 2026. [Daring]. Tersedia: <https://panelharga.badanpangan.go.id>. [Diakses: 15 Juni 2026].

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 18 Juni 2026


Angelina Andra Alanna
13524079