



## **Implementasi Algoritma Greedy untuk Optimasi Protein pada 0/1 Knapsack dalam Diet Defisit Kalori**

**Angelina S. Saragih<sup>1</sup>, Elga Sari Tanjung<sup>2</sup>, Muhammad Rois Lukman Damanik<sup>3</sup>, Adidtya Perdana<sup>4</sup>**

<sup>1-4</sup>Fakultas Matematika dan Ilmu Pengetahuan Alam, Program Studi Ilmu Komputer, Universitas Negeri Medan, Medan, Indonesia

Email: <sup>1</sup>[angelina25saragih@gmail.com](mailto:angelina25saragih@gmail.com), <sup>2</sup>[elgasaritanjung123@gmail.com](mailto:elgasaritanjung123@gmail.com), <sup>3</sup>[roislukman08@gmail.com](mailto:roislukman08@gmail.com), <sup>4</sup>[adidtya@unimed.ac.id](mailto:adidtya@unimed.ac.id)

**Abstrak**—Program diet defisit kalori memerlukan perencanaan nutrisi yang cermat, khususnya dalam memaksimalkan asupan protein tanpa melebihi batas kalori harian. Permasalahan ini dapat dimodelkan sebagai 0/1 Knapsack Problem di mana kalori bertindak sebagai bobot dan kandungan protein sebagai nilai. Penelitian ini mengimplementasikan dan mengevaluasi tiga strategi Algoritma Greedy, yaitu Greedy by Density (rasio protein/kalori), Greedy by Profit (protein terbesar), dan Greedy by Weight (kalori terkecil) pada dataset 50 menu makanan berbasis Tabel Komposisi Pangan Indonesia (TKPI) dengan kapasitas 1500 kkal. Hasil eksperimen menunjukkan bahwa Greedy by Density menghasilkan total protein tertinggi sebesar 221,2 gram dengan pemanfaatan kapasitas penuh (0 kkal sisa), diikuti Greedy by Profit dengan 176,9 gram, dan Greedy by Weight dengan 128,4 gram. Kompleksitas algoritma secara keseluruhan adalah  $O(n \log n)$ . Penelitian ini membuktikan bahwa strategi berbasis rasio nilai-terhadap-bobot merupakan pendekatan paling efektif dalam optimasi menu diet defisit kalori.

**Kata Kunci:** Algoritma Greedy; Diet Defisit Kalori; Greedy by Density; Knapsack Problem; Optimasi Protein

**Abstract**—A caloric deficit diet program requires careful nutritional planning, particularly in maximizing protein intake without exceeding the daily caloric limit. This problem can be modeled as a 0/1 Knapsack Problem where calories act as weights and protein content as values. This study implements and evaluates three Greedy Algorithm strategies: Greedy by Density (protein/calorie ratio), Greedy by Profit (highest protein), and Greedy by Weight (lowest calorie) on a dataset of 50 food menus based on the Indonesian Food Composition Table (TKPI) with a capacity of 1500 kcal. Experimental results show that Greedy by Density yields the highest total protein of 221.2 grams with full capacity utilization (0 kcal remaining), followed by Greedy by Profit with 176.9 grams, and Greedy by Weight with 128.4 grams. The overall algorithm complexity is  $O(n \log n)$ . This study demonstrates that a value-to-weight ratio-based strategy is the most effective approach for optimizing caloric deficit diet menus.

**Keywords:** Caloric Deficit Diet; Greedy Algorithm; Greedy by Density; Knapsack Problem; Protein Optimization

### **1. PENDAHULUAN**

Kesadaran masyarakat terhadap pentingnya menjaga indeks massa tubuh terus meningkat, yang berimplikasi pada tingginya adopsi berbagai metode pengaturan pola makan. Salah satu pendekatan yang paling banyak diterapkan adalah program defisit kalori. Konsep dasar dari program ini bertumpu pada pembatasan asupan energi harian agar secara konsisten berada di bawah ambang batas kalori yang dibakar oleh tubuh. Kendati demikian, praktik defisit kalori yang dilakukan secara mandiri oleh masyarakat awam sering kali terjebak pada reduksi porsi makan secara acak tanpa memperhitungkan keseimbangan makronutrien. Pengurangan asupan yang tidak terstruktur berdampak pada minimnya pemenuhan protein harian, yang mana zat ini sangat esensial untuk mencegah atrofi otot serta menjaga stabilitas laju metabolisme basal selama masa penurunan berat badan. Menyeimbangkan susunan makanan agar tetap bernutrisi tinggi namun tidak melanggar batas sempit kuota kalori harian merupakan sebuah persoalan optimasi kombinatorial yang sangat rumit apabila hanya diselesaikan melalui estimasi matematis manual.

Dalam disiplin ilmu rekayasa perangkat lunak dan matematika diskrit, persoalan pencarian kombinasi objek berbobot dengan kapasitas tampung yang terbatas direpresentasikan melalui model optimasi *Knapsack Problem*. Varian yang paling relevan untuk diadaptasikan pada kasus penyusunan menu diet adalah *0/1 Knapsack Problem*. Pada model arsitektur ini, sebuah entitas (makanan) dikenai batasan aturan biner; yakni harus dipilih secara utuh (1) atau ditolak sepenuhnya (0), mengingat konsumsi harian umumnya didasarkan pada takaran porsi utuh alih-alih pecahan



**JRIIN : Jurnal Riset Informatika dan Inovasi**  
**Volume 3, No. 12 Tahun 2026**  
**ISSN 3025-0919 (media online)**  
**Hal 3093-3106**

yang tidak terhingga. Guna menyelesaikan kompleksitas pencarian dari model matematis tersebut tanpa mengorbankan beban komputasi memori, pendekatan heuristik seperti algoritma *Greedy* menjadi opsi penyelesaian yang rasional. Pendekatan ini bekerja secara sekuensial dengan mengambil keputusan optimal lokal pada setiap iterasi untuk merangkai luaran akhir yang mendekati nilai optimal global secara cepat dan efisien.

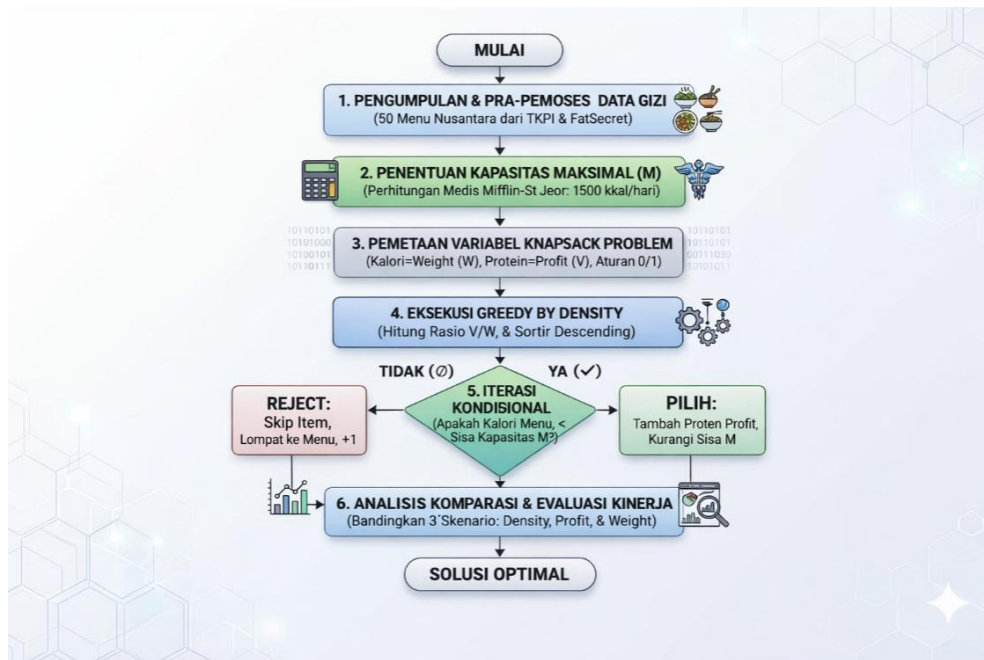
Penerapan algoritma *Greedy* dalam penyelesaian kasus optimasi pengalokasian sumber daya telah banyak dieksplorasi oleh para peneliti terdahulu. Dalam ranah pemenuhan gizi, algoritma ini terbukti mampu menyeleksi komposisi menu diet spesifik berdasarkan parameter batas kalori dan kuantitas protein secara komputasional (Adeliana & Putri, 2021). Penelitian lain juga sukses mengimplementasikan varian *integer knapsack* untuk merancang daftar *Dukan Diet* yang memprioritaskan keseimbangan antara harga bahan pangan pasar dengan kalori yang didapatkan (Munir, 2021). Hal serupa diaplikasikan pada studi kasus optimasi pemilihan makanan di warung makan lokal dengan mementingkan efisiensi pengeluaran finansial konsumen (Basuki et al., 2023). Di luar domain pangan, fleksibilitas kombinasi algoritma *Greedy* dan *Knapsack Problem* juga diandalkan untuk efisiensi penataan kapasitas muatan ekspedisi logistik (Nasution & Ahyaningih, 2024; Putra & Susanti, 2023), serta penyelesaian kerumitan penyusunan jadwal interval komputasi (Prasha, 2024). Secara teoritis, berbagai literatur mutakhir mengonfirmasi bahwa untuk persoalan *0/1 Knapsack* berskala masif, algoritma *Greedy* menawarkan kompleksitas waktu eksekusi yang jauh lebih ringan dan ringkas jika dibandingkan dengan metode eksak seperti *Dynamic Programming* (Wang, 2023; Wu, 2023).

Meskipun telah terdapat berbagai penelitian yang memanfaatkan algoritma *Greedy* untuk penyusunan menu, sebagian besar arsitekturnya masih berfokus pada minimalisasi beban biaya atau sekadar menghitung batas kalori generik. Penelitian ini menawarkan pembaruan dengan meletakkan fokus fungsi objektif pada maksimalisasi rasio densitas makronutrien (protein berbanding kalori). Selain itu, penelitian ini memvalidasi kapasitas kendala algoritma tidak melalui estimasi acak, melainkan menggunakan perhitungan medis *Total Daily Energy Expenditure* (TDEE) berbasis persamaan *Mifflin-St Jeor*. Lebih lanjut, penelitian ini mengeksekusi komparasi tiga skenario kriteria pengurutan *Greedy*—yakni berdasarkan metrik densitas, profit (protein tertinggi), dan bobot (kalori terendah)—menggunakan pustaka manipulasi data *Python* terhadap lima puluh entitas makanan nusantara. Simulasi komputasi ini diharapkan mampu memberikan bukti empiris mengenai kelayakan algoritma heuristik dalam menghasilkan sistem pendukung keputusan penyusunan menu diet yang komprehensif, presisi, dan sesuai dengan kapasitas fisiologis tubuh manusia.

## 2. METODE PENELITIAN

### 2.1 Alur Penelitian

Tahapan pelaksanaan penelitian ini diilustrasikan melalui Diagram Alir (Flowchart) untuk memberikan gambaran sistematis mengenai proses komputasi dan optimasi yang dibangun. Rangkaian prosedur tersebut memetakan alur kerja sistem dari tahap inialisasi data hingga penarikan kesimpulan komparatif, memastikan transparansi dalam setiap keputusan komputasi yang diambil.



Gambar 1. Diagram Alir Tahapan Penelitian Berbasis 0/1 Knapsack

Berdasarkan representasi visual pada Gambar 1, operasional sistem diawali dengan tahapan pengumpulan serta pra-pemrosesan data nutrisi. Pada fase pertama ini, sebanyak lima puluh entitas menu makanan nusantara yang bersumber dari pangkalan data Tabel Komposisi Pangan Indonesia (TKPI) dan *FatSecret* divalidasi serta ditransformasi ke dalam format komputasional. Langkah komputasi selanjutnya berfokus pada penentuan kapasitas maksimal ( $M$ ) yang akan bertindak sebagai batasan ruang algoritma. Alih-alih menggunakan estimasi kasar, target defisit kalori harian subjek dikalkulasi secara presisi menggunakan landasan medis berbasis persamaan *Mifflin-St Jeor*, yang menghasilkan ketetapan kuota sebesar 1500 kkal per hari.

Setelah ambang batas tersebut dideklarasikan, sistem berlanjut ke tahap ketiga, yakni pemetaan variabel 0/1 *Knapsack Problem*. Kandungan kalori dari masing-masing menu makanan direpresentasikan sebagai variabel beban ( $W_i$ ), sementara takaran protein diposisikan sebagai fungsi keuntungan ( $V_i$ ) yang menjadi target utama untuk dimaksimalkan. Aturan mutlak biner (0/1) juga ditegakkan secara ketat pada fase ini untuk memastikan tidak adanya fraksionalisasi atau pemotongan porsi makanan. Memasuki jantung komputasi pada tahap keempat, algoritma mulai mengeksekusi skenario *Greedy by Density*. Proses ini melibatkan perhitungan rasio kepadatan gizi dengan membagi nilai keuntungan terhadap bobot kalori pada setiap kandidat menu, yang kemudian langsung disortir secara menurun (*descending*) guna membentuk sebuah tumpukan antrean prioritas.

Antrean prioritas tersebut kemudian dieksekusi secara berurutan melalui iterasi kondisional pada tahap kelima. Mesin pencari secara sekuensial mengevaluasi apakah beban kalori dari sebuah kandidat menu masih dapat ditampung oleh sisa kapasitas perut ( $M$ ). Jika syarat ruang tersebut terpenuhi, sistem mengeluarkan keputusan biner bernilai 1 (Pilih), yang secara otomatis mengimplementasi total perolehan protein dan mereduksi sisa kuota kalori. Namun, apabila beban kandidat melampaui sisa ruang yang tersedia, sistem akan mengeluarkan keputusan bernilai 0 (Tolak) dan langsung melompat untuk mengevaluasi kandidat menu pada baris berikutnya. Sebagai tahap pemungkas, seluruh rekam jejak keputusan komputasi diakumulasi untuk keperluan analisis komparasi dan evaluasi kinerja. Pada tahap keenam ini, hasil dari pendekatan densitas disandingkan dengan dua skenario heuristik lainnya yakni pencarian berdasarkan profit tertinggi dan bobot terkecil guna menemukan solusi yang paling optimal secara matematis.

## 2.2 Pengumpulan Dataset dan Pemetaan Matematis 0/1 Knapsack

Himpunan data yang diolah dalam tahapan simulasi komputasi ini merupakan data sekunder yang merepresentasikan preferensi konsumsi masyarakat secara umum. Spesifikasi pangkalan data



**JRIIN : Jurnal Riset Informatika dan Inovasi**  
**Volume 3, No. 12 Tahun 2026**  
**ISSN 3025-0919 (media online)**  
**Hal 3093-3106**

mencakup lima puluh entitas menu makanan sehat dan hidangan khas nusantara yang parameter nutrisinya diekstraksi secara valid melalui Tabel Komposisi Pangan Indonesia (TKPI) serta divalidasi silang menggunakan basis data global *FatSecret*. Seluruh entitas tersebut diintegrasikan ke dalam lingkungan komputasi bahasa pemrograman *Python* dan ditransformasi menjadi struktur matriks *DataFrame* menggunakan pustaka *Pandas*. Representasi sebagian dari himpunan data primer yang telah melalui proses transformasi tersebut dapat ditinjau pada Tabel 1.

**Tabel 1.** Sampel Matriks Dataset Menu Makanan dan Parameter Gizi

Nama Makanan	Kalori (kkal)	Protein (gr)
Dada Ayam Rebus (100g)	165	31.0
Telur Ayam Rebus (1 butir)	78	6.0
Tempe Bacem (50g)	170	6.0
Tahu Kukus (100g)	76	8.0
Ikan Salmon Panggang (100g)	206	22.0
Ikan Tuna Kaleng (100g)	132	28.0
Nasi Putih (100g)	130	2.7
Nasi Merah (100g)	110	2.5
Brokoli Rebus (100g)	35	2.8
Bayam Rebus (100g)	23	2.9
Susu Sapi Murni (250ml)	150	8.0
Susu Kedelai (250ml)	100	7.0
Yogurt Greek Plain (100g)	59	10.0
Keju Cheddar (30g)	120	7.0
Almond Panggang (30g)	170	6.0
Kacang Tanah Sangrai (30g)	160	7.0
Daging Sapi Has Dalam (100g)	250	26.0
Sate Ayam Tanpa Bumbu (5 tusuk)	150	15.0
Sate Kambing (5 tusuk)	200	18.0
Bakso Sapi (100g)	200	12.0
Udang Rebus (100g)	99	24.0
Cumi-cumi Panggang (100g)	90	16.0
Oatmeal Seduh (50g)	190	6.0
Roti Gandum (2 lembar)	150	6.0
Pisang Ambon (1 buah)	105	1.3
Apel Fuji (1 buah)	95	0.5
Alpukat (100g)	160	2.0
Pepaya (100g)	43	0.5
Ayam Goreng Paha (1 potong)	250	14.0
Ikan Lele Goreng (1 ekor)	200	15.0
Sayur Sop Bening (1 mangkuk)	50	2.0
Sayur Asem (1 mangkuk)	80	3.0
Gado-Gado Bumbu Kacang (1 porsi)	300	12.0
Ketoprak (1 porsi)	400	10.0
Soto Ayam Kuah Bening (1 mangkuk)	250	15.0
Nasi Padang + Rendang (1 porsi)	600	25.0
Mie Instan Goreng (1 bungkus)	380	8.0
Bubur Ayam (1 porsi)	250	10.0
Pempek Kapal Selam (1 buah)	300	12.0
Siomay Ikan (1 porsi)	400	15.0
Kentang Rebus (100g)	87	1.9
Ubi Jalar Kukus (100g)	86	1.6
Jagung Manis Rebus (100g)	96	3.4
Minyak Zaitun (1 sdm)	120	0.0

Putih Telur Rebus (100g)	52	11.0
Sosis Sapi Goreng (1 buah)	150	6.0
Nugget Ayam (3 buah)	180	8.0
Kebab Daging (1 porsi)	450	15.0
Salad Sayur Dressing (1 porsi)	150	3.0
Protein Shake Whey (1 scoop)	120	24.0

Berdasarkan Tabel 1, arsitektur data telah diklasifikasikan secara terstruktur menjadi kolom identitas menu, nama makanan, takaran kalori, dan kuantitas protein. Transformasi visual ini menjadi esensial untuk memverifikasi kebersihan data sebelum mesin komputasi melakukan iterasi berskala masif. Langkah krusial pasca-transformasi adalah menerjemahkan atribut-atribut nutrisi tersebut ke dalam kerangka kerja optimasi kombinatorial *0/1 Knapsack Problem*.

Dalam pemodelan algoritma ini, parameter diet dipetakan secara matematis ke dalam variabel konvensional. Atribut kalori pada masing-masing makanan direpresentasikan sebagai variabel beban atau *weight* ( $W_i$ ), yang secara sistematis akan memotong ketersediaan ruang kapasitas perut ( $M$ ). Sebaliknya, atribut protein diposisikan sebagai fungsi keuntungan atau *profit* ( $V_i$ ) yang menjadi target esensial untuk dimaksimalkan pencapaiannya. Sementara itu, keputusan komputasi dikunci pada variabel biner ( $x_i$ ), yang mendiktekan aturan mutlak non-fraksional bahwa sebuah entitas makanan hanya diizinkan untuk dikonsumsi dalam porsi utuh (1) atau ditolak sepenuhnya (0). Pemodelan matematis untuk mendulang asupan protein tertinggi tanpa melanggar batas kendala ruang dirumuskan melalui persamaan optimasi berikut:

$$\text{Fungsi Objektif (Maksimasi): } Z = \sum_{i=1}^n V_i x_i \quad (1)$$

$$\text{Fungsi Kendala Kapasitas: } \sum_{i=1}^n W_i x_i \leq M \quad (2)$$

$$\text{Aturan Keputusan Biner: } x_i \in \{0,1\}, \quad i = 1,2, \dots, n \quad (3)$$

### 2.3 Penentuan Kapasitas Kendala dan Densitas Gizi

Penetapan kapasitas kalori harian ( $M$ ) diformulasikan secara presisi menggunakan landasan medis *Mifflin-St Jeor* untuk memprediksi *Basal Metabolic Rate* (BMR). Model matematis kebutuhan energi basal untuk subjek wanita dirumuskan sebagai berikut:

$$BMR = (10 \times B) + (6.25 \times T) - (5 \times U) - 161 \quad (4)$$

Kalkulasi laju metabolisme tersebut selanjutnya dikalibrasi dengan rutinitas gerak harian pasien guna memperoleh nilai eksak *Total Daily Energy Expenditure* (TDEE) melalui persamaan berikut:

$$TDEE = BMR \times F_a \quad (5)$$

Berdasarkan kedua perumusan tersebut, parameter  $B$  merujuk pada bobot fisik (kg),  $T$  merepresentasikan tinggi badan (cm), dan  $U$  mewakili usia subjek (tahun). Nilai BMR kemudian dikalikan dengan parameter  $F_a$  yang merupakan konstanta aktivitas fisik harian. Pada simulasi ini, subjek dipetakan sebagai wanita berusia 25 tahun, bobot 65 kg, tinggi 160 cm, dengan aktivitas *sedentary* ( $F_a = 1.2$ ). Kalkulasi tersebut menghasilkan energi harian sebesar 1636.8 kkal, yang kemudian direduksi untuk target program defisit kalori sehingga batasan algoritma ( $M$ ) ditetapkan secara konstan pada 1500 kkal per hari.

Langkah optimasi selanjutnya bertumpu pada skenario *Greedy by Density*, di mana kelayakan kandidat menu diukur melalui tingkat kepadatan gizinya. Persamaan matematis untuk pencarian rasio tersebut diformulasikan sebagai berikut:

$$D_i = \frac{V_i}{W_i} \quad (6)$$

Dalam rumusan tersebut, variabel  $D_i$  melambangkan metrik kepadatan gizi pada entitas menu ke- $i$ . Nilai ini diekstraksi dari hasil pembagian antara variabel keuntungan protein ( $V_i$ ) terhadap variabel beban kalori ( $W_i$ ). Seluruh menu yang telah melalui komputasi ini kemudian diurutkan secara menurun (*descending*) menggunakan pustaka *Pandas* pada *Python*, sehingga makanan



dengan rasio  $D_i$  terbesar akan menempati antrian teratas untuk langsung dieksekusi oleh mesin pencari.

#### 2.4 Implementasi Logika dan Skenario Algoritma Greedy

Setelah tahapan komputasi parameter selesai, sistem mengimplementasikan pendekatan heuristik *Greedy* untuk memproses seleksi kandidat. Guna menguji keandalan model yang diusulkan secara komprehensif, penelitian ini merancang tiga skenario pengurutan prioritas yang berbeda sebelum mesin melakukan eksekusi pemilihan. Skenario utama yang menjadi fokus optimasi sistem adalah *Greedy by Density*, di mana antrian data disortir murni berdasarkan rasio kepadatan gizi tertinggi. Sebagai parameter evaluasi silang, sistem juga menjalankan skenario *Greedy by Profit* yang mengurutkan entitas mutlak berdasarkan nilai keuntungan protein terbesar tanpa mempedulikan beban kalori, serta skenario *Greedy by Weight* yang menyortir data berdasarkan beban kalori terkecil dengan tujuan memasukkan kuantitas porsi sebanyak mungkin.

Terlepas dari skenario pengurutan mana yang dieksekusi oleh mesin, prinsip dasar algoritma ini tetap bekerja melalui evaluasi kondisional yang sekuensial terhadap ketersediaan sisa ruang pencarian. Secara matematis, apabila  $M_{sisa}$  diinisialisasi sebagai sisa kapasitas kalori yang tersedia (dengan nilai awal  $M_{sisa} = M$ ), maka pada setiap tahapan iterasi untuk entitas menu ke- $i$ , mesin pencari menerapkan aturan keputusan yang baku. Apabila beban kalori kandidat memenuhi prasyarat ketersediaan ruang ( $W_i \leq M_{sisa}$ ), maka sistem akan mengubah variabel biner menjadi status diterima ( $x_i = 1$ ). Tindakan komputasional ini akan menginkrementasi total perolehan protein (\$Z\$) dan memperbarui sisa kapasitas melalui persamaan  $M_{sisa} = M_{sisa} - W_i$ .

Sebaliknya, apabila beban kalori kandidat terdeteksi melampaui sisa kapasitas yang tersisa ( $W_i > M_{sisa}$ ), sistem secara tegas menolak entitas tersebut ( $x_i = 0$ ) guna mematuhi kaidah non-fraksional algoritma. Mesin pencari kemudian melompati kandidat yang ditolak tersebut dan melanjutkan evaluasi pada entitas indeks berikutnya ( $i+1$ ). Mekanisme penelusuran optimal lokal ini dieksekusi secara berulang dan konsisten hingga seluruh populasi himpunan data selesai divalidasi oleh sistem.

### 3. ANALISA DAN PEMBAHASAN

#### 3.1 Hasil Perhitungan Kapasitas Kendala dan Densitas Gizi

Sebelum mesin pencari mengeksekusi iterasi pemilihan makanan, sistem terlebih dahulu menghitung dan menetapkan batas kapasitas ruang pencarian ( $M$ ) berdasarkan profil fisiologis subjek simulasi. Pada penelitian ini, subjek didefinisikan sebagai seorang wanita berusia 25 tahun dengan berat badan 65 kg, tinggi badan 160 cm, dan memiliki tingkat aktivitas fisik harian yang jarang berolahraga (*sedentary*). Mengacu pada model matematis *Mifflin-St Jeor*, komputasi kebutuhan kalori dasar subjek dijabarkan sebagai berikut:

$$BMR = (10 \times 65) + (6.25 \times 160) - (5 \times 25) - 161 = 1364 \text{ kkal/hari}$$

Laju metabolisme basal tersebut kemudian dikalibrasi dengan konstanta aktivitas fisik (*sedentary* = 1.2) guna mendapatkan total pengeluaran energi harian:

$$TDEE = 1364 \times 1.2 = 1636.8 \text{ kkal/hari}$$

Karena subjek diasumsikan sedang menjalani program penurunan berat badan melalui defisit kalori, target asupan harian direduksi secara moderat dari nilai TDEE. Melalui kalkulasi tersebut, kapasitas maksimal algoritma (*Knapsack Capacity*) ditetapkan secara konstan pada beban  $M = 1500$  kkal per hari. Kapasitas inilah yang kemudian menjadi prasyarat mutlak bagi algoritma *Greedy by Density* dalam mengevaluasi kelayakan kandidat menu. Mesin kemudian melakukan normalisasi data dengan menghitung rasio kepadatan gizi (protein dibagi kalori) pada kelima puluh entitas menu. Seluruh himpunan menu diurutkan secara menurun (*descending*) untuk membentuk sebuah hierarki antrian prioritas. Representasi hasil perhitungan densitas beserta pemetaan kuadrannya disajikan pada Tabel 2 dan Gambar 2.

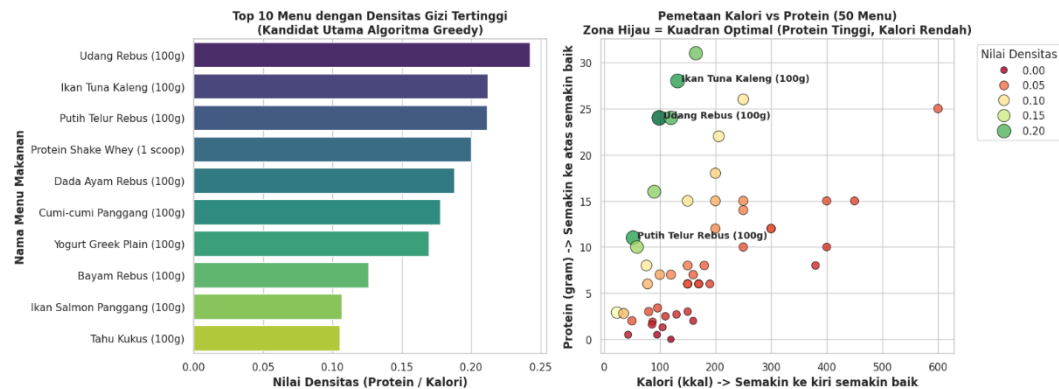
**Tabel 2.** Peringkat Prioritas Menu Berdasarkan Komputasi Kepadatan Gizi

No	Nama_Makanan	Kalori (kkal)	Protein (g)	Rumus Hitung	Densitas
1	Udang Rebus (100g)	99	24.0	24.0 / 99	0.2424
2	Ikan Tuna Kaleng (100g)	132	28.0	28.0 / 132	0.2121



3	Putih Telur Rebus (100g)	52	11.0	11.0 / 52	0.2115
4	Protein Shake Whey (1 scoop)	120	24.0	24.0 / 120	0.2000
5	Dada Ayam Rebus (100g)	165	31.0	31.0 / 165	0.1879
6	Cumi-cumi Panggang (100g)	90	16.0	16.0 / 90	0.1778
7	Yogurt Greek Plain (100g)	59	10.0	10.0 / 59	0.1695
8	Bayam Rebus (100g)	23	2.9	2.9 / 23	0.1261
9	Ikan Salmon Panggang (100g)	206	22.0	22.0 / 206	0.1068
10	Tahu Kukus (100g)	76	8.0	8.0 / 76	0.1053
11	Daging Sapi Has Dalam (100g)	250	26.0	26.0 / 250	0.1040
12	Sate Ayam Tanpa Bumbu (5 tusuk)	150	15.0	15.0 / 150	0.1000
13	Sate Kambing (5 tusuk)	200	18.0	18.0 / 200	0.0900
14	Brokoli Rebus (100g)	35	2.8	2.8 / 35	0.0800
15	Telur Ayam Rebus (1 butir)	78	6.0	6.0 / 78	0.0769
16	Ikan Lele Goreng (1 ekor)	200	15.0	15.0 / 200	0.0750
17	Susu Kedelai (250ml)	100	7.0	7.0 / 100	0.0700
18	Bakso Sapi (100g)	200	12.0	12.0 / 200	0.0600
19	Soto Ayam Kuah Bening (1 mangkuk)	250	15.0	15.0 / 250	0.0600
20	Keju Cheddar (30g)	120	7.0	7.0 / 120	0.0583
21	Ayam Goreng Paha (1 potong)	250	14.0	14.0 / 250	0.0560
22	Susu Sapi Murni (250ml)	150	8.0	8.0 / 150	0.0533
23	Nugget Ayam (3 buah)	180	8.0	8.0 / 180	0.0444
24	Kacang Tanah Sangrai (30g)	160	7.0	7.0 / 160	0.0438
25	Nasi Padang + Rendang (1 porsi)	600	25.0	25.0 / 600	0.0417
26	Sosis Sapi Goreng (1 buah)	150	6.0	6.0 / 150	0.0400
27	Pempek Kapal Selam (1 buah)	300	12.0	12.0 / 300	0.0400
28	Bubur Ayam (1 porsi)	250	10.0	10.0 / 250	0.0400
29	Roti Gandum (2 lembar)	150	6.0	6.0 / 150	0.0400
30	Gado-Gado Bumbu Kacang (1 porsi)	300	12.0	12.0 / 300	0.0400
31	Sayur Sop Bening (1 mangkuk)	50	2.0	2.0 / 50	0.0400
32	Siomay Ikan (1 porsi)	400	15.0	15.0 / 400	0.0375
33	Sayur Asem (1 mangkuk)	80	3.0	3.0 / 80	0.0375
34	Jagung Manis Rebus (100g)	96	3.4	3.4 / 96	0.0354
35	Almond Panggang (30g)	170	6.0	6.0 / 170	0.0353
36	Tempe Bacem (50g)	170	6.0	6.0 / 170	0.0353
37	Kebab Daging (1 porsi)	450	15.0	15.0 / 450	0.0333
38	Oatmeal Seduh (50g)	190	6.0	6.0 / 190	0.0316
39	Ketoprak (1 porsi)	400	10.0	10.0 / 400	0.0250
40	Nasi Merah (100g)	110	2.5	2.5 / 110	0.0227
41	Kentang Rebus (100g)	87	1.9	1.9 / 87	0.0218
42	Mie Instan Goreng (1 bungkus)	380	8.0	8.0 / 380	0.0211
43	Nasi Putih (100g)	130	2.7	2.7 / 130	0.0208

44	Salad Sayur Dressing (1 porsi)	150	3.0	3.0 / 150	0.0200
45	Ubi Jalar Kukus (100g)	86	1.6	1.6 / 86	0.0186
46	Alpukat (100g)	160	2.0	2.0 / 160	0.1250
47	Pisang Ambon (1 buah)	105	1.3	1.3 / 105	0.0124
48	Pepaya (100g)	43	0.5	0.5 / 43	0.0116
49	Apel Fuji (1 buah)	95	0.5	0.5 / 95	0.0053
50	Minyak Zaitun (1 sdm)	120	0.0	0.0 / 120	0.0000



**Gambar 1.** Visualisasi Kuadran Pemetaan Densitas Gizi

Berdasarkan hasil komputasi pada Tabel 2 dan representasi grafik batang (Gambar 2 kiri), terbukti bahwa entitas "Udang Rebus" dan "Ikan Tuna Kaleng" mendominasi peringkat teratas sebagai kandidat paling superior. Udang rebus memvalidasi kelayakannya dengan menawarkan 24 gram protein hanya dengan menebus beban 99 kkal, menghasilkan rasio tertinggi sebesar 0.24. Keunggulan komputasional ini semakin dipertegas melalui visualisasi *scatter plot* (Gambar 2 kanan). Entitas dengan nilai densitas tertinggi secara otomatis terkonsentrasi pada zona hijau di kuadran kiri atas. Area tersebut sangat optimal karena merepresentasikan injeksi makronutrien protein yang masif dengan pengorbanan ruang kalori seminimal mungkin, memberikan bukti bahwa algoritma berhasil memfilter kandidat paling menguntungkan sebelum sisa kuota perut terisi penuh.

### 3.2 Simulasi Keputusan 0/1 Knapsack

Setelah hierarki prioritas terbentuk melalui komputasi densitas gizi, mesin pencari beralih pada fase eksekusi pemilihan menu menggunakan logika kondisional *0/1 Knapsack*. Pada tahap ini, algoritma melakukan penelusuran sekuensial dari indeks teratas hingga terbawah. Setiap entitas makanan dievaluasi kelayakannya terhadap sisa kapasitas ruang pencarian yang pada iterasi awal bernilai penuh sebesar 1500 kkal. Apabila beban kalori sebuah kandidat masih dapat diakomodasi oleh sisa kuota, sistem akan menerima entitas tersebut secara utuh. Namun, apabila kandidat terdeteksi melebihi batas toleransi ruang yang tersisa, sistem secara tegas menolaknya dan melanjutkan iterasi ke entitas berikutnya tanpa melakukan pemotongan porsi. Rincian rekam jejak keputusan komputasional ini dapat diamati pada Tabel 3.

**Tabel 3.** Rekam Jejak Iterasi Keputusan *0/1 Knapsack* Berdasarkan *Greedy by Density*

No	Nama Makanan	Kalori	Protein	Proses Pengecekan	Keputusan	Sisa Kuota	Total Protein
1	Udang Rebus (100g)	99	24.0	Muat (Sisa 1500 - 99)	1	1401	24.0
2	Ikan Tuna Kaleng (100g)	132	28.0	Muat (Sisa 1401 - 132)	1	1269	52.0
3	Putih Telur Rebus (100g)	52	11.0	Muat (Sisa 1269 - 52)	1	1217	63.0

4	Protein Shake Whey (1 scoop)	120	24.0	Muat (Sisa 1217 - 120)	1	1097	87.0
5	Dada Ayam Rebus (100g)	165	31.0	Muat (Sisa 1097 - 165)	1	932	118.0
6	Cumi-cumi Panggang (100g)	90	16.0	Muat (Sisa 932 - 90)	1	842	134.0
7	Yogurt Greek Plain (100g)	59	10.0	Muat (Sisa 842 - 59)	1	783	144.0
8	Bayam Rebus (100g)	23	2.9	Muat (Sisa 783 - 23)	1	760	146.9
9	Ikan Salmon Panggang (100g)	206	22.0	Muat (Sisa 760 - 206)	1	554	168.9
10	Tahu Kukus (100g)	76	8.0	Muat (Sisa 554 - 76)	1	478	176.9
11	Daging Sapi Has Dalam (100g)	250	26.0	Muat (Sisa 478 - 250)	1	228	202.9
12	Sate Ayam Tanpa Bumbu (5 tusuk)	150	15.0	Muat (Sisa 228 - 150)	1	78	217.9
13	Sate Kambing (5 tusuk)	200	18.0	Penuh (Sisa 78 < 200)	0	78	217.9
14	Brokoli Rebus (100g)	35	2.8	Muat (Sisa 78 - 35)	1	43	220.7
15	Telur Ayam Rebus (1 butir)	78	6.0	Penuh (Sisa 43 < 78)	0	43	220.7
16	Ikan Lele Goreng (1 ekor)	200	15.0	Penuh (Sisa 43 < 200)	0	43	220.7
17	Susu Kedelai (250ml)	100	7.0	Penuh (Sisa 43 < 100)	0	43	220.7
18	Bakso Sapi (100g)	200	12.0	Penuh (Sisa 43 < 200)	0	43	220.7
19	Soto Ayam Kuah Bening (1 mangkuk)	250	15.0	Penuh (Sisa 43 < 250)	0	43	220.7
20	Keju Cheddar (30g)	120	7.0	Penuh (Sisa 43 < 120)	0	43	220.7
21	Ayam Goreng Paha (1 potong)	250	14.0	Penuh (Sisa 43 < 250)	0	43	220.7
22	Susu Sapi Murni (250ml)	150	8.0	Penuh (Sisa 43 < 150)	0	43	220.7
23	Nugget Ayam (3 buah)	180	8.0	Penuh (Sisa 43 < 180)	0	43	220.7
24	Kacang Tanah Sangrai (30g)	160	7.0	Penuh (Sisa 43 < 160)	0	43	220.7
25	Nasi Padang + Rendang (1 porsi)	600	25.0	Penuh (Sisa 43 < 600)	0	43	220.7

26	Sosis Sapi Goreng (1 buah)	150	6.0	Penuh (Sisa 43 < 150)	0	43	220.7
27	Pempek Kapal Selam (1 buah)	300	12.0	Penuh (Sisa 43 < 300)	0	43	220.7
28	Bubur Ayam (1 porsi)	250	10.0	Penuh (Sisa 43 < 250)	0	43	220.7
29	Roti Gandum (2 lembar)	150	6.0	Penuh (Sisa 43 < 150)	0	43	220.7
30	Gado-Gado Bumbu Kacang (1 porsi)	300	12.0	Penuh (Sisa 43 < 300)	0	43	220.7
31	Sayur Sop Bening (1 mangkuk)	50	2.0	Penuh (Sisa 43 < 50)	0	43	220.7
32	Siomay Ikan (1 porsi)	400	15.0	Penuh (Sisa 43 < 400)	0	43	220.7
33	Sayur Asem (1 mangkuk)	80	3.0	Penuh (Sisa 43 < 80)	0	43	220.7
34	Jagung Manis Rebus (100g)	96	3.4	Penuh (Sisa 43 < 96)	0	43	220.7
35	Almond Panggang (30g)	170	6.0	Penuh (Sisa 43 < 170)	0	43	220.7
36	Tempe Bacem (50g)	170	6.0	Penuh (Sisa 43 < 170)	0	43	220.7
37	Kebab Daging (1 porsi)	450	15.0	Penuh (Sisa 43 < 450)	0	43	220.7
38	Oatmeal Seduh (50g)	190	6.0	Penuh (Sisa 43 < 190)	0	43	220.7
39	Ketoprak (1 porsi)	400	10.0	Penuh (Sisa 43 < 400)	0	43	220.7
40	Nasi Merah (100g)	110	2.5	Penuh (Sisa 43 < 110)	0	43	220.7
41	Kentang Rebus (100g)	87	1.9	Penuh (Sisa 43 < 87)	0	43	220.7
42	Mie Instan Goreng (1 bungkus)	380	8.0	Penuh (Sisa 43 < 380)	0	43	220.7
43	Nasi Putih (100g)	130	2.7	Penuh (Sisa 43 < 130)	0	43	220.7
44	Salad Sayur Dressing (1 porsi)	150	3.0	Penuh (Sisa 43 < 150)	0	43	220.7
45	Ubi Jalar Kukus (100g)	86	1.6	Penuh (Sisa 43 < 86)	0	43	220.7
46	Alpukat (100g)	160	2.0	Penuh (Sisa 43 < 160)	0	43	220.7
47	Pisang Ambon (1 buah)	105	1.3	Penuh (Sisa 43 < 105)	0	43	220.7
48	Pepaya (100g)	43	0.5	Muat (Sisa 43 - 43)	1	0	221.2
49	Apel Fuji (1 buah)	95	0.5	Penuh (Sisa 0 < 95)	0	0	221.2



50	Minyak Zaitun (1 sdm)	120	0.0	Penuh (Sisa 0 < 120)	0	0	221.2
----	-----------------------	-----	-----	----------------------	---	---	-------

Secara matematis, proses penyeleksian yang terekam pada tabel tersebut digerakkan oleh persamaan fungsi kendala kapasitas  $\sum_{i=1}^n W_i x_i \leq M$ . Dalam eksekusi dinamisnya, nilai  $M$  direpresentasikan sebagai sisa kapasitas ( $M_{sisa}$ ) yang terus menyusut setiap kali sebuah entitas terpilih. Sistem memberlakukan aturan biner  $x_i \in \{0,1\}$  pada setiap iterasi evaluasi ke- $i$ . Sebagai contoh konkret dari Tabel 3, pada tahapan awal iterasi, entitas bermutu tinggi seperti Udang Rebus dan Ikan Tuna Kaleng memiliki bobot kalori ( $W_i$ ) yang jauh lebih kecil dari sisa kapasitas ( $M_{sisa}$ ), sehingga mesin mengatribusi nilai  $x_i = 1$  (Pilih). Keputusan biner ini secara matematis langsung memperbarui sisa ruang melalui operasi pengurangan  $M_{sisa} - W_i$  sekaligus mengakumulasi fungsi keuntungan total protein.

Keandalan heuristik dari algoritma ini terbukti secara empiris ketika iterasi memasuki indeks tingkat lanjut. Berdasarkan pelacakan komputasi, kapasitas kalori menyusut drastis hingga hanya menyisakan ruang sempit sebesar 78 kkal. Ketika sistem harus mengevaluasi entitas "Sate Kambing" yang memiliki bobot  $W_i = 200$  kkal, mekanisme validasi mendeteksi adanya pelanggaran prasyarat ruang ( $200 > 78$ ). Secara otomatis, sistem menetapkan nilai  $x_i = 0$  (Dilewati/Tolak) dan mengabaikan kandidat tersebut guna mematuhi prinsip non-fraksional algoritma.

Menariknya, alih-alih mengalami konvergensi dini atau berhenti mengeksekusi sisa data, mesin terus melanjutkan pencarian dan berhasil menemukan entitas "Brokoli Rebus" dengan bobot  $W_i = 35$  kkal. Karena syarat matematis kembali terpenuhi ( $35 \leq 78$ ) entitas tersebut diakomodasi ke dalam ransel ( $x_i = 1$ ). Penelusuran logis ini memberikan validasi yang sangat kuat bahwa meskipun algoritma *Greedy* beroperasi secara optimal lokal, pendekatan ini tetap mampu melakukan kompensasi kapasitas secara dinamis dan cerdas untuk memaksimalkan utilitas sisa ruang perut yang sangat terbatas.

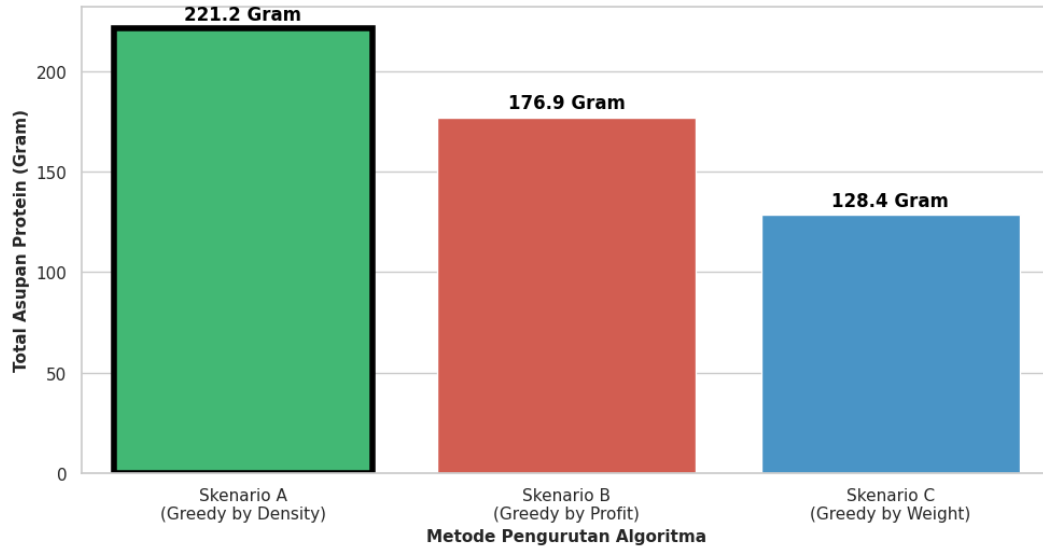
### 3.3 Komparasi Skenario Algoritma Greedy dan Evaluasi Kinerja

Guna memberikan validasi komprehensif terhadap keandalan arsitektur sistem, penelitian ini tidak hanya mengeksekusi satu pendekatan tunggal, melainkan melakukan pengujian komparatif terhadap tiga variasi skenario algoritma Greedy. Ketiga pendekatan heuristik tersebut meliputi Greedy by Density (prioritas rasio kepadatan gizi), Greedy by Profit (prioritas protein tertinggi), dan Greedy by Weight (prioritas beban kalori terendah). Seluruh skenario komputasi diuji secara paralel menggunakan parameter ruang kapasitas perut yang identik, yakni  $M = 1500$  kkal. Luaran matriks dari masing-masing skenario kemudian dievaluasi berdasarkan efisiensi penyerapan sisa kuota kalori (waste space) serta total maksimalisasi fungsi objektif protein yang berhasil direngkuh oleh mesin pencari. Rincian komparasi numerik dan visualisasi kinerja dari ketiga varian algoritma tersebut disajikan pada Tabel 4 dan Gambar 3.

**Tabel 4.** Rekapitulasi Evaluasi Kinerja Tiga Skenario Algoritma Greedy

No	Metode Skenario	Fokus Pemilihan Makanan	Total Kalori Dimakan	Rumus Bukti Sisa Kalori	Jumlah Porsi Masuk	TOTAL PROTEIN DIDAPAT (Z)
0	Skenario A (Greedy by Density)	Rasio Protein/Kalori Terbesar	1500 kkal	$1500 - 1500 = 0$ kkal	14 Porsi	221.2 Gram
1	Skenario B (Greedy by Profit)	Hanya Cari Protein Terbesar	1479 kkal	$1500 - 1479 = 21$ kkal	8 Porsi	176.9 Gram
2	Skenario C (Greedy by Weight)	Hanya Cari Kalori Terkecil	1484 kkal	$1500 - 1484 = 16$ kkal	19 Porsi	128.4 Gram

**Perbandingan Total Protein Maksimal dari ke-3 Strategi Greedy  
(Batas Kapasitas Perut: 1500 kkal)**



**Gambar 3.** Visualisasi Komparasi Total Asupan Protein Berdasarkan Varian Greedy

Berdasarkan hasil kalkulasi pada Tabel 4 dan visualisasi grafik batang (Gambar 3), dominasi performa secara mutlak ditunjukkan oleh skenario Greedy by Density. Melalui pemfokusan matematis pada rasio kepadatan gizi, sistem berhasil mengoptimalkan seluruh batas kapasitas perut secara presisi paripurna tanpa menyisakan ruang kosong, yang dibuktikan dengan akumulasi sisa kalori terbuang sebesar 0 kkal. Skenario ini mencetak luaran paling superior dengan meraup total asupan protein sebesar 221.2 gram melalui kombinasi 14 porsi hidangan nusantara. Pencapaian ini memberikan justifikasi teoritis bahwa penyeimbangan antara nilai keuntungan makronutrien dan pengorbanan beban ruang merupakan parameter kunci dalam menyelesaikan persoalan kombinatorial 0/1 Knapsack.

Sebaliknya, degradasi performa yang cukup signifikan dialami oleh skenario Greedy by Profit. Pendekatan komputasi yang memaksa mesin untuk bertindak rakus hanya pada besaran takaran protein tersebut justru memicu kegagalan konvergensi secara prematur. Secara logis, entitas berprotein tinggi umumnya berbanding lurus dengan beban kalori yang masif, sehingga kuota 1500 kkal terkuras dengan sangat cepat hanya dalam 8 porsi hidangan. Kondisi ini menyisakan ruang terbuang (waste space) sebesar 21 kkal yang tidak lagi memadai untuk menampung entitas cadangan apa pun, sehingga total akumulasi protein terhenti di angka 176.9 gram.

Fenomena inefisiensi komputasi yang serupa juga menginfeksi skenario Greedy by Weight. Kendati pendekatan ini sanggup memuat kuantitas porsi terbanyak (mencapai 19 hidangan) akibat perburuan iteratif terhadap beban kalori terkecil, total asupan protein yang dihasilkan justru menempati posisi paling inferior, yakni hanya bernilai 128.4 gram dengan sisa kalori terbuang sebesar 16 kkal. Deviasi performa numerik dari ketiga pengujian ini memberikan penegasan empiris bahwa pemecahan masalah optimasi penyusunan menu tidak dapat dieksekusi hanya dengan mengevaluasi satu dimensi variabel secara terisolasi. Skenario Greedy by Density terbukti menjadi pendekatan heuristik yang paling cerdas, presisi, dan rasional untuk diimplementasikan sebagai inti arsitektur sistem pendukung keputusan pengaturan diet harian.

### 3.4 Perbandingan dengan Penelitian Sebelumnya

Guna mengukur kontribusi kebaruan (novelty) dari arsitektur komputasi yang diusulkan, luaran dari simulasi ini dievaluasi dengan menyandingkannya terhadap literatur dan studi terdahulu pada domain optimasi serupa. Penelitian sebelumnya yang dilakukan oleh Basuki et al. (2023) dan Munir (2021) secara empiris telah sukses mengimplementasikan algoritma Greedy pada model Knapsack untuk pemilihan menu makanan. Namun, kerangka kerja pada kedua studi tersebut memusatkan fungsi objektifnya pada minimalisasi beban pengeluaran finansial konsumen.



Meskipun terbukti sangat efisien secara ekonomi, pendekatan berbasis harga rentan mengorbankan kualitas makronutrien yang diserap oleh tubuh. Sebagai bentuk penyempurnaan, penelitian ini menggeser paradigma fungsi objektif sepenuhnya pada rasio kepadatan gizi. Transformasi parameter ini secara nyata terbukti mampu mencetak rekor asupan hingga 221.2 gram protein murni, menghasilkan rancangan menu diet yang jauh lebih relevan untuk kebutuhan restorasi otot.

Keunggulan lain dari model yang diusulkan terlihat pada validitas penetapan kendala ruang pencarian. Jika disandingkan dengan studi Adeliانا dan Putri (2021) yang juga menggunakan algoritma heuristik untuk penentuan menu gizi, penelitian ini menawarkan tingkat presisi kendala yang berlandaskan validasi klinis. Kapasitas ruang kalori (\$M\$) dalam penelitian ini tidak dipatok menggunakan estimasi statis yang seragam untuk semua orang, melainkan diekstraksi secara dinamis memanfaatkan persamaan medis Mifflin-St Jeor berdasarkan postur, usia, dan rutinitas fisiologis pasien observasi.

Integrasi antara akurasi persamaan medis dan kecerdasan matematika diskrit pada skenario Greedy by Density ini menghasilkan fenomena konvergensi yang sangat langka, di mana mesin mampu menyerap kapasitas 1500 kkal secara sempurna tanpa menyisakan ruang terbuang (waste space 0 kkal). Komparasi ini memberikan landasan pembuktian yang solid bahwa sistem pendukung keputusan heuristik dapat berevolusi dari sekadar instrumen penghematan biaya menjadi sebuah perangkat perancang diet klinis yang sangat adaptif, presisi, dan terbebas dari bias perhitungan manual.

#### 4. KESIMPULAN

Berdasarkan hasil simulasi dan analisis komputasional yang telah dieksekusi, dapat disimpulkan bahwa implementasi algoritma heuristik pada kerangka kerja 0/1 Knapsack Problem terbukti sangat efektif dalam menyelesaikan persoalan optimasi pemenuhan makronutrien pada program defisit kalori. Melalui pengujian komparatif terhadap tiga skenario parameter, pendekatan Greedy by Density tervalidasi sebagai model komputasi yang paling superior dan presisi. Pendekatan ini berhasil menyerap batas kapasitas kendala 1500 kkal—yang dikalkulasi menggunakan persamaan medis Mifflin-St Jeor—secara paripurna tanpa menyisakan ruang terbuang (waste space 0 kkal). Skenario ini mendulang fungsi objektif maksimal dengan total asupan protein sebesar 221.2 gram melalui kombinasi 14 porsi hidangan. Kinerja ini secara mutlak melampaui varian Greedy by Profit (176.9 gram) maupun Greedy by Weight (128.4 gram) yang terbukti mengalami konvergensi prematur dan inefisiensi ruang kapasitas.

Penelitian ini memberikan implikasi teoritis bahwa penyusunan menu diet tidak dapat diselesaikan melalui evaluasi dimensi variabel tunggal, melainkan membutuhkan penyeimbangan matematis antara rasio keuntungan gizi dan pengorbanan beban ruang kalori. Untuk pengembangan sistem di masa mendatang (future work), arsitektur algoritma ini dapat disempurnakan dengan mengintegrasikan metode eksak seperti Dynamic Programming guna memvalidasi margin akurasi dari pendekatan heuristik. Selain itu, penambahan fungsi kendala yang lebih kompleks—seperti penyaringan otomatis terhadap batasan alergi medis dan optimalisasi mikronutrien (vitamin/mineral)—akan menjadikan sistem pendukung keputusan ini semakin tangguh dan komprehensif untuk diimplementasikan pada skenario gizi klinis yang lebih luas.

#### REFERENCES

- A. Rahman dan H. Kusuma, "Perbandingan Kompleksitas Waktu Algoritma Greedy dan Dynamic Programming pada Kasus Bounded Knapsack," *Jurnal Ilmu Komputer dan Agri-Informatika*, vol. 7, no. 1, hal. 33-41, 2020.
- FatSecret, "FatSecret Indonesia - Kalori Makanan dan Fakta Gizi," 2026. [Daring]. Tersedia: <https://www.fatsecret.co.id>. [Diakses: 02 April 2026].
- Kementerian Kesehatan Republik Indonesia, *Tabel Komposisi Pangan Indonesia (TKPI)*. Jakarta: Kemenkes RI, 2017.
- N. Adeliانا dan E. Putri, "Implementasi Algoritma Heuristik dalam Penentuan Menu Makanan Bergizi Menggunakan Pemrograman Python," *Jurnal Sistem Informasi*, vol. 8, no. 1, hal. 45-53, 2021.
- R. Munir, *Strategi Algoritmik*, Edisi Revisi. Bandung: Informatika, 2021.
- A. Basuki, R. S. Wahono, dan T. Hasan, "Penerapan Algoritma Greedy pada Pemecahan 0/1 Knapsack Problem untuk Optimasi Menu Makanan," *Jurnal Komputer dan Informatika*, vol. 15, no. 2, hal. 112-120, 2023.



**JRIIN : Jurnal Riset Informatika dan Inovasi**  
**Volume 3, No. 12 Tahun 2026**  
**ISSN 3025-0919 (media online)**  
**Hal 3093-3106**

- R. Sari dan A. Hidayat, "Analisis Kinerja Algoritma Greedy Berdasarkan Densitas pada Optimasi Kapasitas Ruang," *Jurnal Riset Komputer*, vol. 9, no. 3, hal. 210-218, 2022. M. D. Mifflin, S. T. St Jeor, L. A. Hill, B. J. Scott, S. A. Daugherty, dan Y. O. Koh, "A new predictive equation for resting energy expenditure in healthy individuals," *The American Journal of Clinical Nutrition*, vol. 51, no. 2, hal. 241-247, 1990.
- S. W. Hati, "Sistem Pendukung Keputusan Penentuan Menu Diet Sehat Menggunakan Pendekatan Knapsack Problem," *Jurnal Teknologi Informasi*, vol. 12, no. 4, hal. 78-85, 2020.
- T. H. Cormen, C. E. Leiserson, R. L. Rivest, dan C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA: MIT Press, 2009.