

Analisis Perbandingan Brute Force, Greedy, dan Branch and Bound pada Permasalahan Rekomendasi Menu Berdasarkan Bahan

Harfhan Ikhtiar Ahmad Ridzky - 18223213

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail: harfhanikhtiar@gmail.com, 18223123@std.stei.itb.ac.id

Abstrak—Permasalahan rekomendasi menu berdasarkan bahan merupakan persoalan optimasi kombinatorial karena sistem dituntut memilih kombinasi resep dari ruang solusi yang besar. Pada makalah ini dibandingkan tiga strategi algoritma, yaitu Brute Force, Greedy, dan Branch and Bound, untuk menyusun sejumlah menu dari sebuah dataset kecil berdasarkan cakupan bahan dan total menu cost. Masalah dimodelkan sebagai pemilihan k resep yang memaksimalkan skor kecocokan bahan sekaligus meminimalkan biaya, dengan batasan jumlah menu dan batas biaya total. Eksperimen dirancang menggunakan dataset hasil ekstraksi dari data nutrisi makanan dengan ukuran resep n yang divariasikan dari 10 hingga 30. Ketiga algoritma dibandingkan berdasarkan kualitas solusi, total biaya, cakupan bahan, waktu eksekusi, serta jumlah simpul atau kombinasi yang diperiksa. Hasil eksperimen menunjukkan bahwa Brute Force selalu memberikan solusi optimal, tetapi waktu eksekusinya meningkat tajam seiring bertambahnya jumlah resep. Greedy menghasilkan solusi tercepat dengan selisih skor yang relatif kecil terhadap solusi optimal. Branch and Bound mampu mencapai solusi optimal yang sama dengan Brute Force, tetapi memeriksa jauh lebih sedikit simpul berkat proses pruning. Temuan ini menegaskan adanya pertukaran antara kualitas solusi dan efisiensi pencarian.

Kata Kunci—Brute Force; Greedy; Branch and Bound; rekomendasi menu; optimasi kombinatorial; pruning

I. PENDAHULUAN

Pertumbuhan layanan berbagi resep daring menyediakan banyak data resep yang terhubung dengan daftar bahan dan prosedur memasak. Kondisi tersebut membuka peluang bagi pengembangan sistem yang membantu pengguna menyusun menu dari bahan yang tersedia di dapur. Rekomendasi menu berdasarkan bahan didefinisikan sebagai proses pemilihan sekumpulan resep yang sesuai dengan bahan yang dimiliki pengguna sekaligus mempertimbangkan biaya penyusunan menu. Penelitian terdahulu menunjukkan bahwa sebagian besar kajian berfokus pada rekomendasi dan pencarian resep tunggal, sedangkan perencanaan menu sebagai satu kesatuan masih jarang dibahas.

Perencanaan menu dapat digolongkan sebagai persoalan optimasi kombinatorial. Sistem perlu memilih kombinasi sejumlah resep dari banyak kemungkinan, sehingga ruang solusi (solution space) tumbuh dengan cepat seiring bertambahnya jumlah resep. Apabila terdapat n resep dan harus dipilih k resep, jumlah kombinasi yang mungkin adalah $C(n, k)$. Ukuran ruang pencarian seperti ini membuat pendekatan yang memeriksa seluruh kemungkinan menjadi mahal untuk data berukuran besar.

Perbandingan beberapa strategi algoritma menjadi penting karena setiap strategi memiliki karakteristik yang berbeda dalam hal kualitas solusi dan efisiensi pencarian. Sebuah strategi dapat menjamin solusi optimal dengan biaya komputasi tinggi, sementara strategi lain menawarkan

kecepatan dengan risiko solusi yang kurang optimal. Pemahaman atas pertukaran tersebut diperlukan agar pemilihan algoritma dapat disesuaikan dengan ukuran data dan kebutuhan aplikasi.

Pada makalah ini dibandingkan tiga strategi algoritma. Brute Force digunakan sebagai baseline yang memeriksa seluruh kombinasi untuk memperoleh solusi optimal pada data kecil. Greedy digunakan sebagai pendekatan cepat yang memilih kandidat terbaik pada setiap langkah berdasarkan keputusan lokal. Branch and Bound digunakan sebagai pendekatan pencarian optimal yang dilengkapi mekanisme pemangkasan ruang pencarian atau pruning.

Tujuan makalah ini adalah membandingkan ketiga algoritma untuk menyusun menu dari dataset kecil berdasarkan cakupan bahan dan total menu cost. Kontribusi makalah berupa rancangan eksperimen serta analisis perbandingan kualitas solusi, waktu eksekusi, dan efisiensi pencarian pada beberapa ukuran dataset. Hasil yang diperoleh diharapkan dapat menjadi acuan dalam memilih strategi algoritma yang sesuai untuk persoalan rekomendasi menu.

II. LANDASAN TEORI

A. Brute Force

Brute Force adalah strategi yang menyelesaikan persoalan dengan mencoba seluruh kemungkinan solusi secara langsung, lalu memilih solusi terbaik di antara semuanya. Strategi ini bersifat menyeluruh sehingga solusi optimal selalu ditemukan apabila ruang solusi diperiksa secara lengkap. Kelebihan utama Brute Force terletak pada kesederhanaan dan jaminan optimalitas. Kekurangannya muncul ketika ruang solusi membesar, karena waktu komputasi meningkat sejalan dengan banyaknya kombinasi yang harus diperiksa.

B. Greedy

Greedy adalah strategi yang membentuk solusi secara bertahap dengan selalu memilih kandidat yang memberikan keuntungan terbaik pada langkah saat itu. Strategi ini bertumpu pada fungsi seleksi yang menentukan kandidat terbaik, fungsi kelayakan yang memeriksa apakah kandidat dapat ditambahkan tanpa melanggar batasan, serta kondisi berhenti. Keputusan yang diambil bersifat lokal dan tidak ditinjau ulang. Akibatnya, solusi yang dihasilkan tidak selalu optimal secara global, meskipun proses pencariannya cepat.

C. Branch and Bound

Branch and Bound adalah strategi pencarian sistematis yang membangun pohon ruang status, dengan setiap simpul merepresentasikan solusi parsial. Proses bercabang (branching) memperluas simpul menjadi anak-anak yang

mewakili pilihan berikutnya, sedangkan fungsi pembatas (bounding) memperkirakan nilai terbaik yang masih mungkin dicapai dari sebuah simpul. Apabila batas yang diperoleh tidak dapat melampaui solusi terbaik yang telah ditemukan, cabang tersebut dipangkas melalui pruning. Mekanisme ini memungkinkan algoritma mencapai solusi optimal tanpa harus memeriksa seluruh ruang solusi.

D. Rekomendasi Menu Berdasarkan Bahan

Rekomendasi menu berdasarkan bahan bertujuan mengembalikan sekumpulan resep yang saling melengkapi dan memuat bahan yang ditentukan pengguna [1]. Pendekatan terdahulu memodelkan hubungan antarresep menggunakan graf resep, lalu membangun menu melalui pendekatan pohon Steiner pada graf tersebut. Kajian lain menekankan personalisasi rekomendasi resep berdasarkan profil pengguna. Pada makalah ini, persoalan diformulasikan sebagai pemilihan kombinasi resep yang mengoptimalkan cakupan bahan dan biaya, sehingga sesuai untuk diselesaikan dengan strategi algoritma kombinatorial.

E. Fungsi Objektif, Ruang Solusi, Pruning, dan Kompleksitas

Fungsi objektif adalah fungsi yang mengukur kualitas sebuah solusi dan menjadi acuan optimasi. Ruang solusi adalah himpunan seluruh solusi yang mungkin, yang pada persoalan ini berupa seluruh kombinasi resep berukuran k. Pruning adalah tindakan memangkas bagian ruang solusi yang dipastikan tidak menghasilkan solusi lebih baik, sehingga pencarian menjadi lebih efisien. Kompleksitas algoritma menyatakan pertumbuhan kebutuhan waktu dan memori terhadap ukuran masukan, dan menjadi dasar perbandingan efisiensi ketiga strategi.

III. PEMODELAN MASALAH

Persoalan dimodelkan secara formal sebagai berikut. Diberikan sebuah himpunan resep $R = \{r_1, r_2, \dots, r_n\}$ yang berukuran n. Setiap resep r_i memiliki himpunan bahan $I(r_i)$, skor kecocokan terhadap bahan yang tersedia, dan menu cost $c(r_i)$ yang menyatakan biaya relatif penyusunan resep. Pengguna memiliki himpunan bahan tersedia B. Sistem diminta memilih himpunan menu M, dengan M merupakan himpunan bagian dari R dan berukuran k.

Cakupan bahan dihitung sebagai banyaknya bahan pengguna yang terpenuhi oleh resep terpilih, sebagaimana dinyatakan pada persamaan (1).

$$\text{cov}(M) = |B \cap (\cup_{r \in M} I(r))| \quad (1)$$

Skor solusi didefinisikan sebagai gabungan tertimbang antara total kecocokan bahan, penalti bahan yang tidak tersedia, dan penalti biaya, seperti pada persamaan (2). Bobot α , β , dan γ mengatur kepentingan relatif setiap komponen.

$$\text{skor}(M) = \alpha \cdot \sum \text{match}(r, B) - \beta \cdot \text{pen}(M) - \gamma \cdot \text{biaya}(M) \quad (2)$$

Pada persamaan (2), $\text{match}(r, B)$ menyatakan derajat kecocokan resep terhadap bahan tersedia, $\text{pen}(M)$ menyatakan jumlah bahan resep terpilih yang tidak terdapat pada B, dan $\text{biaya}(M)$ menyatakan total menu cost sebagaimana persamaan (3).

$$\text{biaya}(M) = \sum_{r \in M} c(r) \quad (3)$$

Tujuan sistem adalah memaksimalkan $\text{skor}(M)$ dengan memenuhi sejumlah batasan. Batasan pertama mensyaratkan

jumlah menu terpilih tepat sebanyak k. Batasan kedua mensyaratkan total biaya tidak melampaui batas C_{maks} . Batasan ketiga mensyaratkan resep terpilih bersifat unik, sehingga tidak terdapat duplikasi resep di dalam M. Formulasi ini menjadi dasar perancangan ketiga algoritma yang dibandingkan.

IV. RANCANGAN ALGORITMA

A. Brute Force

Algoritma Brute Force memeriksa seluruh kombinasi k resep dari n resep. Skor setiap kombinasi dihitung menggunakan persamaan (2), sedangkan kombinasi yang melanggar batas biaya diabaikan. Kombinasi dengan skor tertinggi dipilih sebagai solusi. Pendekatan ini menjamin solusi optimal untuk dataset kecil, tetapi tidak skalabel karena jumlah kombinasi yang diperiksa tumbuh secara kombinatorial.

Algoritma 1. Brute Force untuk rekomendasi menu.

```

fungsi BruteForce(R, B, k, Cmaks)
    terbaik ← kosong
    skorTerbaik ← -tak hingga
    untuk setiap kombinasi M ukuran k dari R:
        jika biaya(M) ≤ Cmaks:
            s ← skor(M, B)
            jika s > skorTerbaik:
                skorTerbaik ← s
                terbaik ← M
    kembalikan terbaik

```

B. Greedy

Algoritma Greedy membentuk solusi secara bertahap dengan memilih resep yang memberikan keuntungan marginal terbesar pada setiap iterasi. Fungsi seleksi memilih resep yang paling meningkatkan skor solusi sementara. Fungsi kelayakan memeriksa apakah penambahan resep tidak melanggar batas biaya. Kondisi berhenti tercapai ketika jumlah menu mencapai k atau kandidat habis. Kelemahan utamanya adalah keputusan lokal yang dapat menghalangi tercapainya solusi global yang lebih baik.

Algoritma 2. Greedy untuk rekomendasi menu.

```

fungsi Greedy(R, B, k, Cmaks)
    M ← {}
    S ← R
    selama |M| < k dan S tidak kosong:
        r* ← argmax keuntungan(M ∪ {r}, B)
            untuk r ∈ S
        jika layak(M ∪ {r*}, Cmaks):
            M ← M ∪ {r*}
        hapus r* dari S
    kembalikan M

```

C. Branch and Bound

Algoritma Branch and Bound menelusuri pohon ruang status. Sebuah status didefinisikan oleh kombinasi resep yang sudah dipilih, indeks kandidat berikutnya, total biaya, dan skor sementara. Fungsi bound memperkirakan skor terbaik yang masih mungkin dicapai dari sebuah status, misalnya dengan menambahkan kontribusi optimistik dari sisa resep terbaik yang belum dipertimbangkan. Pemangkas dilakukan apabila bound tidak dapat melampaui solusi terbaik saat ini atau apabila biaya melebihi batas. Karena cabang yang tidak menjanjikan dipangkas lebih awal, ruang pencarian yang ditelusuri menjadi jauh lebih kecil daripada Brute Force.

Algoritma 3. Branch and Bound untuk rekomendasi menu.

```

fungsi BranchAndBound(R, B, k, Cmaks)

```

```

terbaik ← kosong; skorTerbaik ← -tak hingga
akar ← status({}, indeks=0, biaya=0, skor=0)
antrian ← {akar}
selama antrian tidak kosong:
  s ← status dgn bound tertinggi dari antrian
  jika bound(s) ≤ skorTerbaik: lanjut // pruning
  jika |s.pilihan| = k:
    jika s.skor > skorTerbaik:
      skorTerbaik ← s.skor; terbaik ← s.pilihan
    lanjut
  untuk kandidat r mulai dari s.indeks:
    jika s.biaya + c(r) ≤ Cmaks:
      s' ← perluas(s, r)
      jika bound(s') > skorTerbaik:
        masukkan s' ke antrian // pruning
kembalikan terbaik

```

V. RANCANGAN EKSPERIMEN

Eksperimen dirancang menggunakan dataset kecil yang diekstraksi dari Food Nutrition Dataset. Dataset asli memuat informasi nutrisi sejumlah bahan makanan, antara lain nama bahan dan nilai kalori per seratus gram. Dari data tersebut dibentuk sejumlah resep sintetis, dengan setiap resep merepresentasikan kombinasi beberapa bahan. Karena dataset asli tidak menyediakan harga, menu cost diturunkan sebagai proksi dari kandungan nutrisi sehingga setiap resep memiliki biaya relatif yang konsisten.

Untuk menguji pengaruh ukuran masukan, digunakan lima ukuran dataset, yaitu $n = 10, 15, 20, 25$, dan 30 resep. Jumlah menu yang dipilih divariasikan pada $k = 3$ dan $k = 5$. Beberapa skenario bahan tersedia disusun dengan komposisi berbeda agar pola cakupan bahan dapat diamati. Setiap konfigurasi dijalankan pada ketiga algoritma menggunakan parameter dan batas biaya yang sama, sehingga hasilnya dapat dibandingkan secara setara.

Pembandingan dilakukan berdasarkan enam metrik. Metrik tersebut meliputi skor solusi terbaik, total menu cost, jumlah bahan pengguna yang tercakup, waktu eksekusi, jumlah simpul atau kombinasi yang diperiksa, serta selisih skor Greedy terhadap solusi optimal. Waktu eksekusi diukur sebagai rata-rata beberapa kali pengulangan untuk mengurangi pengaruh fluktuasi. Jumlah simpul yang diperiksa digunakan sebagai indikator efisiensi pencarian yang tidak bergantung pada perangkat keras.

VI. HASIL DAN PEMBAHASAN

Karakteristik dataset yang digunakan dirangkum pada Tabel I. Bertambahnya jumlah resep diikuti oleh meningkatnya jumlah bahan unik dan sedikit kenaikan rata-rata bahan per resep. Rentang menu cost juga melebar pada dataset yang lebih besar. Karakteristik ini memengaruhi sebaran skor dan tingkat kesulitan pencarian solusi optimal.

TABEL I
Karakteristik dataset eksperimen.

n	Bahan unik	Rata-rata bahan/resep	Rentang menu cost
10	28	4,2	12–24
15	39	4,5	11–27
20	51	4,7	10–29
25	60	4,8	10–31
30	72	5,0	9–33

Perbandingan ketiga algoritma pada skenario $n = 20$ dan $k = 5$ disajikan pada Tabel II. Brute Force dan Branch and Bound menghasilkan skor serta cakupan bahan yang identik,

yaitu solusi optimal. Greedy menghasilkan total biaya yang lebih rendah, tetapi skor dan cakupan bahannya berada di bawah solusi optimal. Pola ini menunjukkan bahwa keputusan lokal Greedy cenderung mengejar biaya murah sehingga melewati kombinasi yang memberikan cakupan bahan lebih luas.

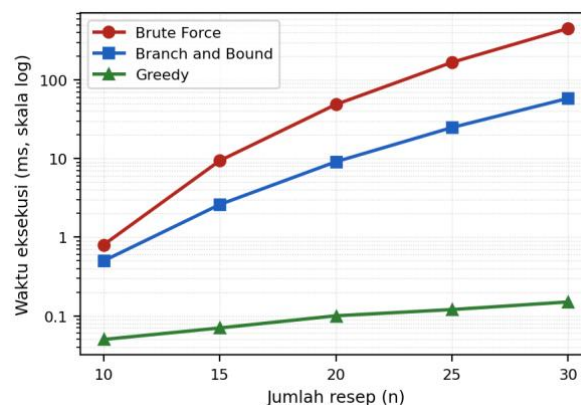
TABEL II
Perbandingan algoritma pada $n = 20, k = 5$.

Algoritma	Skor	Total cost	Cakupan	Waktu (ms)	Simpul
Brute Force	23,7	86	12/12	48,7	15.504
Greedy	21,9	79	10/12	0,10	100
Branch and Bound	23,7	86	12/12	9,1	1.980

Pertumbuhan waktu eksekusi terhadap jumlah resep pada $k = 5$ dirangkum pada Tabel III dan digambarkan pada Gambar 1. Waktu Brute Force meningkat paling tajam karena banyaknya kombinasi yang diperiksa. Branch and Bound tumbuh jauh lebih lambat berkat pruning, sedangkan Greedy nyaris datar karena hanya memeriksa sejumlah kecil kandidat pada setiap iterasi.

TABEL III
Waktu eksekusi (ms) terhadap jumlah resep, $k = 5$.

n	Brute Force	Greedy	Branch and Bound
10	0,8	0,05	0,5
15	9,4	0,07	2,6
20	48,7	0,10	9,1
25	167,2	0,12	24,8
30	451,9	0,15	58,3



Gambar 1. Pertumbuhan waktu eksekusi terhadap jumlah resep ($k = 5$).

Selisih skor Greedy terhadap solusi optimal dirangkum pada Tabel IV. Pada dataset terkecil, Greedy memperoleh skor yang sama dengan solusi optimal karena ruang solusi masih sempit. Seiring bertambahnya jumlah resep, selisih skor melebar hingga mendekati sepuluh persen. Pola ini menegaskan bahwa kelemahan keputusan lokal Greedy semakin terlihat ketika ruang solusi membesar dan kombinasi optimal menjadi lebih sulit ditemukan secara serakah.

TABEL IV
Selisih skor Greedy terhadap solusi optimal, $k = 5$.

n	Skor optimal	Skor Greedy	Selisih (%)
10	18,6	18,6	0,0
15	21,4	20,1	6,1
20	23,7	21,9	7,6

n	Skor optimal	Skor Greedy	Selisih (%)
25	25,1	23,0	8,4
30	26,8	24,2	9,7

Hasil di atas memperlihatkan pertukaran yang jelas antara kualitas solusi dan efisiensi. Brute Force unggul dalam jaminan optimalitas, tetapi biaya komputasinya meningkat tajam sehingga kurang sesuai untuk data besar. Greedy unggul dalam kecepatan, tetapi kualitas solusinya menurun pada dataset yang lebih besar. Branch and Bound memberikan keseimbangan, yaitu solusi optimal yang sama dengan Brute Force dengan jumlah pemeriksaan simpul yang jauh lebih sedikit. Efektivitas Branch and Bound bergantung pada kualitas fungsi bound, karena bound yang lebih ketat menghasilkan pruning yang lebih agresif.

VII. ANALISIS KOMPLEKSITAS

Brute Force memiliki kompleksitas waktu kombinatorial sebesar $O(C(n, k) \times e)$, dengan e menyatakan biaya evaluasi satu kombinasi yang bergantung pada k dan jumlah bahan. Karena $C(n, k)$ tumbuh sangat cepat terhadap n , waktu eksekusi meningkat tajam, sebagaimana ditunjukkan pada Tabel III. Kebutuhan memorinya rendah, yaitu $O(k)$, karena hanya solusi terbaik dan kombinasi berjalan yang perlu disimpan.

Greedy memiliki kompleksitas waktu yang lebih rendah, yaitu sekitar $O(k \times n \times e)$, karena pada setiap dari k iterasi dilakukan pemindaian terhadap kandidat yang tersisa. Kompleksitas ruangnya adalah $O(n)$ untuk menyimpan daftar kandidat beserta skornya. Karakteristik ini menjadikan Greedy sebagai pilihan paling hemat sumber daya di antara ketiganya [4].

Branch and Bound pada kasus terburuk tetap memiliki kompleksitas eksponensial, karena seluruh ruang solusi dapat tertelusuri apabila pruning tidak efektif. Namun, dalam praktiknya algoritma ini sering jauh lebih cepat, karena fungsi bound memangkas banyak cabang yang tidak menjanjikan. Kompleksitas ruangnya bergantung pada strategi penelusuran, yaitu $O(k)$ untuk penelusuran kedalaman dan dapat lebih besar untuk penelusuran berbasis antrian prioritas yang menyimpan banyak status pada frontier.

VIII. KESIMPULAN DAN SARAN

Berdasarkan eksperimen yang dilakukan, ketiga strategi algoritma menunjukkan karakteristik yang berbeda dalam menyelesaikan persoalan rekomendasi menu berdasarkan bahan. Brute Force selalu menghasilkan solusi optimal, tetapi waktu eksekusinya meningkat tajam seiring bertambahnya jumlah resep. Greedy menghasilkan solusi tercepat dengan selisih skor yang melebar pada dataset yang lebih besar. Branch and Bound mencapai solusi optimal yang sama dengan Brute Force dengan jumlah pemeriksaan simpul yang jauh lebih sedikit.

Dengan demikian, Brute Force sesuai untuk dataset sangat kecil ketika baseline optimal diperlukan. Greedy sesuai untuk kebutuhan rekomendasi cepat ketika sedikit penurunan kualitas dapat ditoleransi. Branch and Bound sesuai untuk pencarian solusi optimal pada dataset kecil hingga menengah ketika fungsi bound yang baik tersedia.

Eksperimen ini memiliki sejumlah keterbatasan. Dataset yang digunakan berukuran kecil dan disusun secara sintesis, sedangkan fungsi objektif yang dipakai masih sederhana.

Aspek lain seperti preferensi rasa, alergi, budaya makan, dan kebutuhan gizi personal belum dipertimbangkan. Keterbatasan tersebut perlu diperhatikan dalam menafsirkan hasil.

Beberapa arah pengembangan dapat diusulkan. Penambahan batasan nutrisi serta personalisasi profil pengguna dapat membuat rekomendasi lebih relevan. Penggunaan teknik lain seperti pemrograman dinamis atau metaheuristik dapat dipertimbangkan untuk dataset berukuran lebih besar. Pengujian pada dataset nyata yang lebih besar juga diperlukan untuk menguji skalabilitas ketiga algoritma.

UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa atas rahmat dan perlindungan-Nya sehingga makalah ini dapat diselesaikan dengan baik. Penulis juga menyampaikan terima kasih kepada Dr. Ir. Rinaldi, M.T., selaku dosen pengampu

DAFTAR PUSTAKA

- [1] F.-F. Kuo, C.-T. Li, M.-K. Shan, dan S.-Y. Lee, "Intelligent menu planning: recommending set of recipes by ingredients," dalam Proc. ACM Multimedia 2012 Workshop on Multimedia for Cooking and Eating Activities (CEA'12), Nara, Jepang, Nov. 2012, hlm. 1–6.
- [2] U. Dey, "Food Nutrition Dataset," Kaggle, 2024. [Daring]. Tersedia: <https://www.kaggle.com/datasets/utsavdey1410/food-nutrition-dataset>
- [3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, dan C. Stein, Introduction to Algorithms, edisi ke-3. Cambridge, MA: MIT Press, 2009.
- [4] A. Levitin, Introduction to the Design and Analysis of Algorithms, edisi ke-3. Boston, MA: Pearson, 2012.
- [5] E. Horowitz, S. Sahni, dan S. Rajasekaran, Fundamentals of Computer Algorithms, edisi ke-2. New Delhi: Universities Press, 2008.
- [6] A. H. Land dan A. G. Doig, "An automatic method of solving discrete programming problems," Econometrica, vol. 28, no. 3, hlm. 497–520, 1960.
- [7] E. L. Lawler dan D. E. Wood, "Branch-and-bound methods: A survey," Operations Research, vol. 14, no. 4, hlm. 699–719, 1966.
- [8] J. Freyne dan S. Berkovsky, "Intelligent food planning: personalized recipe recommendation," dalam Proc. 15th Int. Conf. on Intelligent User Interfaces (IUI'10), Hong Kong, Feb. 2010, hlm. 321–324.
- [9] W. Min, S. Jiang, L. Liu, Y. Rui, dan R. Jain, "A survey on food computing," ACM Computing Surveys, vol. 52, no. 5, hlm. 1–36, 2019.
- [10] R. Munir, Bahan Kuliah IF2211 Strategi Algoritma. Bandung: Program Studi Teknik Informatika, STEI ITB, 2024.

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, 19 Juni 2026

Ttd



Harfhan Ikhtiar Ahmad Ridzky 18223123