

Perbandingan Branch and Bound dan Nearest Neighbour Heuristic Pada CVRP Distribusi MBG SPPG Mangunsari Semarang

Nurizal Dwi Priandani^{1*}, Fajar Rohman Hariri¹, Nadia Roosmalita Sari²

¹Teknik Informatika, Fakultas Sains dan Teknologi, UIN Maulana Malik Ibrahim, Malang, Indonesia

²Tadris Matematika, Fakultas Tarbiyah dan Ilmu Keguruan, UIN Sayyid Ali Rahmatullah, Tulungagung, Indonesia

Email: ^{1*}nd.priandani@uin-malang.ac.id, ²fajar@ti.uin-malang.ac.id, ³nadiaroosmalitasari2@uinsatu.ac.id

Email Penulis Korespondensi: nd.priandani@uin-malang.ac.id

Submitted: 28/04/2026; Accepted: 20/07/2026; Published: 30/09/2026

Abstrak—Program Makan Bergizi Gratis (MBG) merupakan kebijakan strategis pemerintah Indonesia untuk memperkuat pembangunan sumber daya manusia menuju Visi Indonesia Emas 2045. Sistem distribusi MBG menuntut perencanaan logistik yang presisi agar makanan dapat diterima oleh setiap sekolah secara tepat sasaran, efektif, dan tepat waktu. Pada penelitian ini, permasalahan distribusi dimodelkan sebagai Capacitated Vehicle Routing Problem (CVRP). Studi kasus dilakukan pada distribusi MBG di SPPG Mangunsari Kota Semarang yang melibatkan 1 depot, 17 sekolah penerima, 2 kendaraan distribusi, dengan kapasitas maksimum 1.000 paket makanan pada setiap kendaraan. Penelitian ini membandingkan kinerja algoritma *Branch and Bound* (BnB) dan *Nearest Neighbor Heuristic* (NNH) menggunakan lima metrik evaluasi, yaitu total jarak tempuh, total operasi komputasi (jumlah operasi yang dihitung), rerata waktu komputasi, panjang rute maksimum, dan keseimbangan jarak rute distribusi. Hasil pengujian menunjukkan bahwa BnB menghasilkan total jarak yang lebih efisien 14% yaitu sebesar 26,31 km, sedangkan NNH menghasilkan total jarak sebesar 30,59 km. Dari sisi total komputasi dan rerata waktu komputasi, NNH 99,99% lebih sedikit langkah komputasi dan 96,78% lebih cepat daripada BnB. Selain itu, NNH juga menunjukkan keunggulan pada keseimbangan jarak rute distribusi antar kendaraan dimana NNH menghasilkan selisih jarak antarkendaraan 98,91% lebih rendah dengan rute terpanjang yang dihasilkan NNH 15,75% lebih pendek dibandingkan dengan BnB. Dengan demikian, BnB lebih sesuai digunakan ketika efisiensi total jarak menjadi prioritas, sedangkan NNH lebih sesuai digunakan ketika efisiensi proses komputasi dan pemerataan jarak rute distribusi menjadi pertimbangan utama. Adapun hasil penelitian ini hanya terbatas pada kasus distribusi MBG di SPPG Mangunsari Kecamatan Gunungsari – Kota Semarang dan belum dapat digeneralisasi untuk seluruh distribusi MBG pada Kota Semarang.

Kata Kunci: MBG; Optimasi; Rute; Distribusi; CVRP; Branch-and-Bound; Nearest Neighbor Heuristic

Abstract— Program Makan Bergizi Gratis (MBG) is a strategic policy of the Indonesian government to strengthen human resource development towards the Golden Indonesia Vision 2045. The MBG distribution system requires precise logistics planning so that food can be received by each school in a targeted, effective, and timely manner. In this study, the distribution problem is modeled as a Capacitated Vehicle Routing Problem (CVRP). The case study was conducted on MBG distribution at SPPG Mangunsari, Semarang City, involving one depot, 17 recipient schools, two distribution vehicles, and a maximum capacity of 1,000 food packages per vehicle. This study compares the performance of the *Branch and Bound* (BnB) and *Nearest Neighbor Heuristic* (NNH) algorithms using five evaluation metrics: total travel distance, total computational operations (the number of fundamental algorithmic operations performed), average computation time, maximum route length, and distribution route distance balance. Experimental results show that BnB produces a total travel distance of 26.31 km, which was 13.99% shorter than the 30.59 km obtained by NNH. However, NNH required 99.99% fewer counted computational operations and achieved a 96.78% shorter average computation time than BnB. NNH also provided a more balanced allocation of travel distances between vehicles, reducing the intervehicle route-distance difference by 98.91% and producing a maximum route length that was 15.75% shorter than that generated by BnB. These findings indicate that BnB is more appropriate when minimizing total distribution distance is the primary objective, whereas NNH is preferable when computational efficiency and balanced route allocation are the main considerations. Nevertheless, the findings are limited to the MBG distribution case at SPPG Mangunsari, Gunungpati District, Semarang City, and therefore cannot yet be generalized to the overall MBG distribution system in Semarang City.

Keywords: MBG; Optimization; Route; Distribution; CVRP; Branch-and-Bound; Nearest Neighbor Heuristic

1. PENDAHULUAN

Sejalan dengan Visi Indonesia Emas 2045, Pemerintah Indonesia menetapkan Program Makan Bergizi Gratis (MBG) sebagai salah satu kebijakan strategis dalam memperkuat pembangunan sumber daya manusia melalui intervensi pemenuhan gizi bagi kelompok sasaran [1]. Secara kelembagaan, penguatan program ini dilakukan dengan pembentukan Badan Gizi Nasional (BGN) sebagai lembaga pemerintah di bawah dan bertanggung jawab kepada Presiden dimana mempunyai tugas melaksanakan pemenuhan gizi nasional [2]. Melalui Peraturan Presiden Nomor 115 Tahun 2025, Program MBG dilaksanakan melalui suatu alur layanan yang menempatkan SPPG sebagai unit pelaksana utama, sekolah sebagai titik penerimaan sekaligus pengawasan penyaluran, dan siswa sebagai penerima manfaat [3]. Dalam implementasinya, tiap SPPG diwajibkan menyiapkan makanan sesuai standar program, dikemas dalam food tray tertutup, dan selanjutnya didistribusikan ke satuan pendidikan dengan menggunakan kendaraan box. Setiap SPPG diwajibkan untuk memiliki sedikitnya dua kendaraan box untuk menunjang kelancaran proses distribusi [4]. Berdasarkan Pedoman Umum Sistem dan Tata Kelola BGN untuk program MBG, satu unit mobil box diharuskan mempunyai kapasitas angkut sekitar 1.000 paket makanan dalam satu kali perjalanan [5].

Kondisi lapangan penyelenggaraan Program MBG di Kota Semarang menunjukkan skala operasional yang besar dimana terdapat 152 SPPG, 1.507 sekolah penerima, dan 281.259 penerima manfaat [6]. Merujuk pada data tersebut, karakteristik permintaan tiap sekolah bersifat beragam dan tidak seragam. Jika melihat terdapat petunjuk teknis dengan adanya keterbatasan kapasitas kendaraan distribusi, maka menyebabkan setiap perjalanan distribusi harus dioptimalkan. Pada skala tersebut, proses distribusi tidak lagi dapat dipandang sebagai aktivitas pengantaran biasa, melainkan sebagai sistem logistik layanan publik yang harus direncanakan secara presisi agar makanan dapat diterima oleh setiap sekolah secara tepat sasaran, efektif, dan dalam waktu distribusi yang seefisien mungkin.

Kondisi lapangan SPPG Mangunsari Kecamatan Gunungsari – Kota Semarang mempunyai sejumlah 1629 penerima MBG dengan 2 buah kendaraan roda 4 dimana masing-masing mempunyai kapasitas 1000 paket makanan sebagai kendaraan distribusi. Jika dilihat dari karakteristik operasionalnya, permasalahan distribusi MBG pada SPPG Mangunsari secara konseptual memiliki kesesuaian yang kuat dengan *Capacitated Vehicle Routing Problem* (CVRP), yaitu model penentuan rute kendaraan yang dirancang untuk melayani sekumpulan pelanggan dengan permintaan tertentu di bawah batas kapasitas kendaraan [7]. Dalam konteks ini, setiap sekolah dapat direpresentasikan sebagai *customer node* dengan besaran permintaan tertentu, sedangkan SPPG berperan sebagai depot tempat kendaraan memulai dan mengakhiri perjalanan. Tujuan utama distribusi tidak hanya berkaitan dengan pemenuhan seluruh permintaan sekolah, tetapi juga dengan penentuan rute yang mampu meminimalkan jarak tempuh distribusi secara keseluruhan tanpa melebihi kapasitas kendaraan yang tersedia. Untuk menyelesaikan permasalahan *Capacitated Vehicle Routing Problem* (CVRP), pendekatan yang dapat digunakan adalah menggunakan algoritma *Branch-and-Bound* (BnB) dan *Nearest Neighbor Heuristic* (NNH).

BnB merupakan algoritma pencarian eksak yang banyak digunakan untuk menyelesaikan masalah optimasi kombinatorial. Tinjauan komprehensif tentang Prinsip kerja BnB yang mencakup strategi pencarian (*search strategy*), strategi pencabangan (*branching strategy*), dan aturan pemangkasan (*pruning rules*), menunjukkan bahwa BnB efisien dalam menyelesaikan kasus CVRP [8]. BnB telah terbukti efektif untuk berbagai variasi masalah routing salah satunya untuk *Multiple Knapsack Problem* (MKP) dengan memperkenalkan kriteria *path-symmetry* dan *path-dominance* untuk memangkas *node* pencarian [9]. Penelitian lain menunjukkan bahwa algoritma *iterative deepening* BnB untuk *Container Pre-Marshalling Problem* (CPMP) dengan memperkuat penggunaan aturan *dominance* dan batas bawah, berhasil menyelesaikan instance berukuran industri secara optimal [10]. Permasalahan penjadwalan dan perutean AGV (*Automated Guided Vehicle*) pada terminal kontainer dengan teknik percepatan seperti *infeasibility test* dan *reoptimization primal-dual*, BnB dapat mengurangi waktu komputasi secara signifikan [11]. Selain itu, rancangan BnB untuk *Shortest Path Problem With Conflicts* (SPC) dengan memanfaatkan solusi relaksasi *shortest path tree* dan prosedur reoptimization efisien dapat menunjukkan performa yang lebih baik daripada *solver* komersial [12]. BnB juga digunakan untuk menangani fungsi tujuan CVRP kompleks untuk *Vehicle Routing Problem With Customer Costs* (VRP-CC) dengan dua strategi pencabangan berbeda, serta batas bawah untuk biaya perjalanan dan biaya pelanggan [13]. Permasalahan *Travelling Salesman Problem* (TSP) dengan parameter tidak pasti menggunakan bilangan fuzzy intuitionistik juga telah terbukti berhasil diselesaikan menggunakan BnB [14]. Beberapa penelitian lain menunjukkan penggunaan BnB pada optimalisasi produksi di berbagai industri seperti spring bed, kain tenun, roti, dan mebel yang diformulasikan sebagai Integer Linear Programming (ILP) [15], [16]. Namun, penerapan algoritma Branch and Bound pada kasus skala besar menghadapi kendala komputasi yang signifikan, yaitu kompleksitas waktu eksponensial. Fenomena ini dikenal sebagai "ledakan ruang solusi," di mana jumlah *node* yang harus dieksplorasi meningkat secara drastis seiring bertambahnya jumlah titik tujuan [11].

NNH didefinisikan sebagai algoritma heuristik konstruktif yang membangun solusi rute secara bertahap dari depot dengan memilih pelanggan *feasible* terdekat yang belum dikunjungi, lalu membuka rute baru ketika tidak ada pelanggan lain yang masih dapat ditambahkan ke rute aktif [17], [18], [19], [20]. Algoritma ini digunakan secara luas masalah NP-Hard salah satunya adalah CVRP dengan menghasilkan solusi yang baik dalam waktu komputasi yang wajar [21], [20], [22], [23]. NN telah digunakan untuk membentuk rute pada sepuluh Russian CVRP Instances dan berhasil menegaskan bahwa NN layak digunakan sebagai pendekatan dasar yang cepat dan praktis [17]. Pada aplikasi distribusi produk dan pengangkutan sampah, NN juga telah menghasilkan penghematan jarak, waktu, biaya, dan jumlah armada dibandingkan kondisi awal atau metode pembandingan yang digunakan [24] [25]. Selain berdiri sendiri, NN juga sering digunakan sebagai komponen inisialisasi pada algoritma yang lebih kuat. Hal ini dikarenakan solusi awal dapat dibentuk dengan cepat sebelum dilakukan perbaikan lebih lanjut oleh prosedur optimasi lain. NN dapat digabungkan dengan Sweep dan perhitungan Haversine distance dimana telah menghasilkan rute yang lebih pendek daripada pendekatan tunggal [19]. Pada penelitian lain, Hasil Algoritma NN digunakan sebagai solusi awal yang kemudian diperbaiki lebih lanjut oleh tahap Tabu Search [18]. Studi lain menunjukkan pola yang sama, yaitu NN memberikan waktu komputasi yang lebih cepat tetapi kualitas rutenya tidak lebih baik dibanding heuristik lain [17].

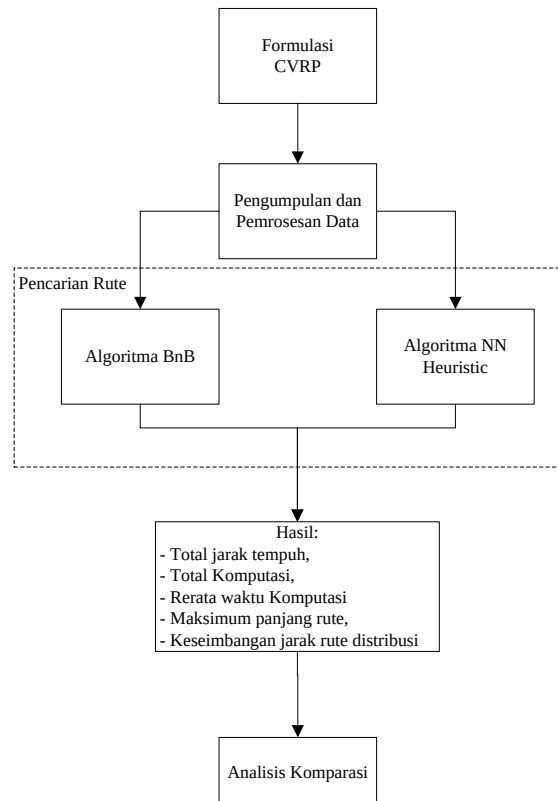
Dalam penelitian ini akan dilakukan perbandingan BnB dan NNH dalam pencarian rute optimal pada permasalahan distribusi MBG SPPG Mangunsari Kecamatan Gunungsari – Kota Semarang. Penelitian ini memberikan kontribusi ilmiah berupa analisis komparatif antara BnB dan NNH pada kasus distribusi nyata dengan menggunakan metrik evaluasi komprehensif, yaitu total jarak tempuh, total komputasi, rerata waktu komputasi, maksimum panjang rute, dan keseimbangan jarak rute antar kendaraan. Melalui pendekatan tersebut, penelitian ini diharapkan dapat memperkaya kajian optimasi rute distribusi dengan menunjukkan *trade-off* antara kualitas

solusi, efisiensi komputasi, dan pemerataan jarak rute distribusi, sehingga dapat menjadi dasar pemilihan algoritma yang lebih sesuai untuk sistem distribusi berskala nyata. Sebagai catatan, penelitian ini hanya menggunakan satu SPPG sehingga hasilnya tidak mewakili seluruh SPPG di Kota Semarang.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini menggunakan pendekatan eksperimental komparatif pada studi kasus nyata. Gambar 1 menampilkan alur tahapan penelitian. Penelitian ini dimulai dari Formulasi CVRP, Pengumpulan dan pengolahan data, kemudian pencarian rute optimal menggunakan BnB dan NNH. Hasil yang didapatkan dievaluasi berdasarkan empat metrik dan terakhir dilakukan analisis komparatif.



Gambar 1. Diagram Alur Tahapan Penelitian

2.2 Capacitated Vehicle Routing Problem (CVRP)

Capacitated Vehicle Routing Problem (CVRP) merupakan permasalahan optimasi distribusi yang bertujuan menentukan rute kendaraan dari suatu depot menuju sejumlah *customer* dan kembali lagi ke depot, dengan tetap memperhatikan batas kapasitas kendaraan [7]. Model ini menuntut agar setiap *customer* dilayani tepat satu kali, seluruh permintaan *customer* terpenuhi, dan total jarak tempuh distribusi minimum. CVRP banyak digunakan pada sistem logistik dan distribusi karena model ini mampu merepresentasikan proses pengiriman barang secara terstruktur, terukur, dan efisien tanpa melanggar batas daya angkut kendaraan.

Permasalahan distribusi MBG pada penelitian ini dimodelkan sebagai *Capacitated Vehicle Routing Problem (CVRP)*. Model ini digunakan karena proses distribusi melibatkan depot, pelanggan, kendaraan, dan batas kapasitas muatan. Setiap sekolah penerima dapat direpresentasikan sebagai *customer node* yang memiliki permintaan tertentu dalam bentuk jumlah paket makanan. Sementara itu, Satuan Pelayanan Pemenuhan Gizi (SPPG) berperan sebagai depot, yaitu lokasi tempat kendaraan memulai distribusi dan kembali setelah menyelesaikan rute pengantaran. Setiap depot diasumsikan memiliki dua kendaraan, dan setiap kendaraan memiliki kapasitas maksimum sebesar 1000 paket makanan. Dengan kata lain, total kapasitas kendaraan yang tersedia pada satu depot adalah 2000 paket makanan dalam satu kali proses distribusi.

Permasalahan CVRP pada distribusi MBG direpresentasikan dalam bentuk graf berarah $G = (V, E)$, dengan V sebagai himpunan *node* dan E sebagai himpunan lintasan antar *node*. *Node* 0 menyatakan depot, sedangkan *node* i , dengan $i = 1, 2, \dots, n$, menyatakan sekolah penerima MBG. Setiap sekolah memiliki permintaan q_i , dan setiap lintasan dari *node* i ke *node* j memiliki jarak d_{ij} . Kendaraan yang tersedia dinyatakan dalam

himpunan K , dengan jumlah kendaraan sebanyak dua unit pada setiap depot. Pada Tabel 1 dan Tabel 2. Masing-masing ditampilkan notasi himpunan, parameter, dan variabel keputusan yang digunakan.

Tabel 1. Himpunan dan Parameter Model CVRP

Simbol	Keterangan
$C = \{1, 2, \dots, n\}$	Himpunan sekolah penerima MBG
$V = \{0\} \cup C$	Himpunan seluruh node
0	Depot/SPPG
$K = \{1, 2\}$	Himpunan kendaraan pada satu depot
d_{ij}	Jarak dari node i ke node j
q_i	Permintaan sekolah i dalam paket makanan
$Q = 1000$	Kapasitas tiap kendaraan adalah 1000

Tabel 2. Variabel Keputusan Model CVRP

Simbol	Keterangan
x_{ijk}	Bernilai 1 jika kendaraan k bergerak dari node i ke node j , dan 0 jika tidak
y_{ik}	Bernilai 1 jika sekolah i dilayani oleh kendaraan k , dan 0 jika tidak

2.2.1 Fungsi Objektif

Model CVRP pada distribusi MBG bertujuan meminimalkan total jarak tempuh seluruh kendaraan yang digunakan untuk mengantarkan makanan dari depot ke sekolah-sekolah tujuan dan kembali lagi ke depot. Fungsi objektif dituliskan dalam Persamaan (1). Persamaan tersebut menyatakan bahwa model akan memilih lintasan antar *node* yang menghasilkan total jarak tempuh minimum.

$$\min Z = \sum_{k \in K} \sum_{i \in V} \sum_{\substack{j \in V \\ j \neq i}} d_{ij} x_{ijk} \quad (1)$$

2.2.2 Constraint Kunjungan Sekolah

Constraint kunjungan sekolah menyatakan bahwa setiap sekolah sebagai *customer node* harus dilayani tepat satu kali oleh satu kendaraan selama proses distribusi. Tujuan *constraint* ini adalah memastikan bahwa seluruh permintaan sekolah terpenuhi tanpa ada sekolah yang terlewat maupun dilayani lebih dari satu kali. *Constraint* ini diformulasikan pada persamaan (2).

$$\sum_{k \in K} \sum_{\substack{j \in V \\ j \neq i}} x_{ijk} = 1, \forall i \in C \quad (2)$$

2.2.3 Flow Conservation Constraint

Flow conservation constraint menyatakan bahwa setiap kendaraan yang masuk ke suatu sekolah harus keluar kembali dari sekolah tersebut. Tujuan *constraint* ini adalah menjaga kesinambungan rute distribusi agar lintasan yang terbentuk tidak terputus di tengah perjalanan dan setiap pergerakan kendaraan tetap konsisten dari satu *node* ke *node* berikutnya. *Constraint* ini diformulasikan pada persamaan (3).

$$\sum_{\substack{j \in V \\ j \neq i}} x_{ijk} - \sum_{\substack{j \in V \\ j \neq i}} x_{jik} = 0, \forall i \in C, \forall k \in K \quad (3)$$

2.2.4 Constraint Keberangkatan dan Kepulangan Kendaraan

Constraint keberangkatan dan kepulangan kendaraan menyatakan bahwa setiap kendaraan yang digunakan harus berangkat dari depot dan kembali lagi ke depot yang sama setelah menyelesaikan distribusi. Tujuan *constraint* ini adalah memastikan bahwa setiap rute distribusi membentuk lintasan yang lengkap dan valid, serta membatasi agar satu kendaraan hanya menjalankan paling banyak satu rute pada satu putaran distribusi. *Constraint* ini diformulasikan pada persamaan (4)

$$\sum_{j \in C} x_{0jk} = \sum_{i \in C} x_{i0k}, \forall k \in K \quad (4)$$

2.2.5 Allocation Constraint Sekolah - Kendaraan

Allocation Constraint sekolah - kendaraan menyatakan hubungan antara sekolah yang dilayani dan kendaraan yang melayani sekolah tersebut melalui variabel keputusan y_{ik} . Tujuan *constraint* ini adalah memastikan bahwa suatu sekolah dinyatakan dilayani oleh kendaraan k hanya jika kendaraan tersebut benar-benar menempuh lintasan menuju atau dari sekolah tersebut. *Constraint* ini diformulasikan pada persamaan (5).

$$y_{ik} = \sum_{\substack{j \in V \\ j \neq i}} x_{ijk}, \forall i \in C, \forall k \in K \quad (5)$$

2.2.6 Constraint Kapasitas Kendaraan

Constraint kapasitas kendaraan menyatakan bahwa total permintaan sekolah yang dilayani oleh setiap kendaraan tidak boleh melebihi kapasitas maksimum kendaraan yang tersedia. Tujuan *constraint* ini adalah memastikan bahwa setiap rute distribusi tetap layak secara operasional, sehingga kendaraan tidak membawa muatan melampaui batas kapasitas angkut selama proses pengiriman paket makanan. *Constraint* ini diformulasikan pada persamaan (6).

$$\sum_{i \in C} q_i y_{ik} \leq Q, \forall k \in K \quad (6)$$

2.2.6 Constraint Eliminasi Subtour

Constraint eliminasi subtour menyatakan bahwa rute distribusi tidak boleh membentuk siklus kecil antar sekolah yang terpisah dari depot. Tujuan kendala ini adalah memastikan bahwa setiap lintasan yang terbentuk merupakan rute distribusi yang valid dimana berangkat dari depot, mengunjungi sekolah-sekolah sesuai alokasi, dan kembali lagi ke depot tanpa membentuk putaran terpisah. Kendala ini diformulasikan pada persamaan (7) dan persamaan (8).

$$u_i - u_j + Q \sum_{k \in K} x_{ijk} \leq Q - q_j, \forall i, j \in C, i \neq j \quad (7)$$

$$q_i \leq u_i \leq Q, \forall i \in C \quad (8)$$

2.3 Pengumpulan Data

Penelitian ini menggunakan data Makanan Bergizi Gratis (MBG) kota Semarang pada SPPG Mangunsari Kecamatan Gunungsari yang diperoleh dari aplikasi SIMANGGIS [6]. Pada Tabel 3 ditampilkan data penerima MBG pada SPPG Mangunsari Kecamatan Gunungpati, Kota Semarang. Data tersebut berisi informasi alamat SPPG depot serta alamat sekolah penerima MBG sebagai titik pelanggan. Selain itu, data tersebut juga memuat jumlah permintaan makanan pada setiap sekolah penerima. Data ini kemudian digunakan sebagai dasar untuk membentuk *node* depot dan *node* pelanggan pada model *Capacitated Vehicle Routing Problem* (CVRP). Penggunaan kendaraan distribusi pada SPPG Mangunsari disesuaikan dengan aturan yang berlaku yaitu 2 kendaraan roda 4 dengan masing-masing kapasitas kendaraan adalah 1000 paket makanan.

Tabel 3. Data Penerima MBG Pada SPPG Mangunsari Gunungpati

Kode Sekolah	Nama Sekolah	Jumlah Siswa	Jenjang
S1	KB PELANGI	77	KB
S2	RA AL IMAN	136	RA
S3	RA ROUDLOTUL HUDA	50	RA
S4	TK SEKAR MEKAR	27	TK
S5	TK BAHRUL KHAIR ZEETGEE	49	TK
S6	TK PERTIWI 45 KALISEGORO	76	TK
S7	SD NEGERI KALISEGORO	144	SD
S8	RA SINAR PELANGI PATEMON	118	RA
S9	RA AL ISLAM MANGUNSARI 1	18	RA
S10	RA AL ISLAM MANGUNSARI 2	107	RA
S11	MIS AL ISLAM MANGUNSARI 1	23	MI
S12	MIS AL ISLAM MANGUNSARI 2	288	MI
S13	TK PERTIWI 29	34	TK
S14	TK PERTIWI 32	50	TK
S15	SD NEGERI PAKINTELAN 01	161	SD
S16	SD NEGERI PAKINTELAN 02	124	SD
S17	SD NEGERI PAKINTELAN 03	147	SD

Setelah data alamat depot dan sekolah penerima diperoleh, setiap alamat diverifikasi dan dikonversi menjadi koordinat geografis berupa garis lintang dan garis bujur. Penghitungan jarak antarlokasi dilakukan pada 30 April 2026, menggunakan layanan Google Routes API dengan *method* Compute Route Matrix. Sebanyak 18 lokasi, dimana terdiri atas 1 depot dan 17 sekolah penerima, digunakan secara bergantian sebagai titik asal (*origin*) dan tujuan (*destination*), sehingga dihasilkan matriks berukuran 18 x 18. *Method* Compute Route Matrix menghitung informasi perjalanan untuk setiap pasangan titik asal dan tujuan serta dapat mengembalikan indeks asal, indeks tujuan, jarak, durasi, status, dan kondisi rute.

Konfigurasi *route request* menggunakan parameter *travelMode* = DRIVE karena proses distribusi dilakukan menggunakan kendaraan roda empat. Preferensi lalu lintas yang digunakan adalah *routingPreference* = TRAFFIC_UNAWARE dengan satuan pengukuran *units* = METRIC. Data respons yang diambil melalui *field mask* meliputi *originIndex*, *destinationIndex*, *distanceMeters*, *duration*. Adapun Nilai *distanceMeters* pada setiap pasangan lokasi dikonversi dari meter ke kilometer.

Matriks jarak dalam penelitian ini berbentuk matriks asimetris, sehingga jarak dari lokasi *i* menuju lokasi *j* tidak selalu sama dengan jarak dari lokasi *j* menuju lokasi *i*. Perbedaan tersebut terjadi karena arah jalan satu arah, pembatasan putar balik, posisi akses masuk dan keluar lokasi, median jalan, larangan belok, serta pemilihan ruas jalan yang berbeda untuk masing-masing arah perjalanan. Matriks jarak tersebut selanjutnya digunakan sebagai input yang sama pada algoritma BnB maupun NNH agar perbandingan kualitas rute dan beban komputasi dilakukan menggunakan data perjalanan yang sama.

2.4 Algoritma Branch-and-Bound

Secara prinsip, Algoritma Branch-and-Bound (BnB) membagi permasalahan utama ke dalam sejumlah subpermasalahan melalui proses *branching*. *Lower bound* atau *upper bound* pada setiap subpermasalahan dihitung setelah proses *branching* dilakukan. Nilai *lower bound* atau *upper bound* digunakan untuk menyeleksi cabang yang masih berpotensi menghasilkan solusi terbaik. Selain itu nilai *lower bound* atau *upper bound* juga digunakan untuk memangkas (*pruning*) cabang yang dipastikan tidak akan menghasilkan solusi yang lebih baik dari solusi sementara yang telah diperoleh [8], [24]. Keandalan metode BnB dalam menjamin solusi optimal untuk model distribusi kompleks telah dikonfirmasi oleh [25], [26], [27], [28].

Pada penelitian ini, nilai *lower bound* setiap *state* dihitung dari akumulasi biaya perjalanan yang telah terbentuk, biaya *Minimum Spanning Tree* (MST) pada himpunan sekolah yang belum dilayani, biaya koneksi minimum dari posisi kendaraan saat ini menuju sekolah yang tersisa, serta biaya koneksi minimum dari sekolah yang tersisa untuk kembali ke depot. Penggunaan relaksasi berbasis MST sebagai batas bawah biaya perjalanan telah diterapkan dalam pendekatan Branch-and-Bound untuk permasalahan TSP dan VRP [13].

Theurich et al. menjelaskan bahwa relaksasi kendala derajat menghasilkan masalah MST yang dapat digunakan sebagai *lower bound* dalam proses pencarian eksak. Dengan sifat tersebut, *lower bound* yang digunakan bersifat *admissible*, yaitu tidak memberikan estimasi yang lebih besar daripada biaya minimum yang masih diperlukan untuk menyelesaikan rute [13]. Oleh karena itu, suatu *state* hanya dipangkas apabila nilai *lower bound* sama dengan atau lebih besar daripada nilai solusi *feasible* terbaik yang telah ditemukan (*incumbent*). Dalam kondisi tersebut, seluruh solusi yang mungkin dihasilkan dari *state* tersebut tidak dapat memberikan nilai objektif yang lebih baik, sehingga proses *pruning* pasti tidak membuang solusi optimal. Prinsip ini sejalan dengan mekanisme BnB yang mengeliminasi suatu cabang ketika batas bawahnya tidak lebih baik daripada *upper bound* dari solusi *feasible* yang tersedia.

Untuk mempercepat penemuan solusi *feasible* yang baik, *child state* tidak dieksplorasi secara acak, tetapi diurutkan berdasarkan rerata jarak kandidat sekolah terhadap sekolah yang masih tersisa dan depot. Nilai rerata tersebut hanya digunakan sebagai strategi pengurutan pencarian dan tidak digunakan untuk menghapus kandidat atau menentukan *pruning*. Seluruh *child state* yang *feasible* tetap dieksplorasi selama tidak memenuhi kondisi pemangkas berdasarkan *lower bound*. Dengan demikian, strategi pengurutan tersebut dapat mempercepat diperolehnya *incumbent* yang baik tanpa mengurangi kelengkapan ruang pencarian maupun menghilangkan sifat eksak algoritma Branch-and-Bound. Pseudocode algoritma BnB ditunjukkan sebagai berikut:

Algorithm: Branch and Bound for CVRP

Input: *dist*[0..n][0..n], *demand*[1..n], *capacity* *Q*, *vehicles* *K*

Output: Optimal routes *R**, minimal distance *z**

```

z* ← ∞
root ← initial state (all vehicles at depot, load=0, visited=∅, cost=0)
root.LB ← ComputeLowerBound(root)
PQ ← priority queue ordered by LB; push root

while PQ ≠ ∅ do
    node ← PQ.pop()
    if node.LB ≥ z* then continue

```

```

if node.visited = all customers then
    total  $\leftarrow$  node.cost +  $\sum_k \text{dist}[\text{node.pos}_k][0]$ 
    if total <  $z^*$  then ( $z^*$ ,  $R^*$ )  $\leftarrow$  (total, node.routes)
    continue
end if
for each  $i \notin \text{node.visited}$  do
    for each vehicle  $k$  with  $\text{node.load}_k + \text{demand}_i \leq Q$  do
        child  $\leftarrow$  node after appending  $i$  to route  $k$ 
        child.LB  $\leftarrow$  ComputeLowerBound(child)
        if child.LB <  $z^*$  then PQ.push(child)
    end for
end for
end while

return  $R^*$ ,  $z^*$ 

```

Setiap *node* pada pohon pencarian menyimpan informasi rute setiap kendaraan, beban kendaraan saat ini, posisi terakhir kendaraan, himpunan pelanggan yang telah dikunjungi, dan total jarak tempuh. Informasi tersebut digunakan untuk merepresentasikan kondisi solusi parsial pada setiap tahap pencarian (*State Representation*). Selanjutnya, fungsi *ComputeLowerBound* digunakan untuk memperkirakan biaya minimum tambahan yang masih diperlukan untuk melayani pelanggan yang tersisa. Nilai perkiraan dihitung dengan menjumlahkan jarak yang telah ditempuh, menambahkan estimasi biaya perjalanan antar pelanggan yang belum dilayani, dan menambahkan estimasi biaya perjalanan kembali ke depot apabila kapasitas kendaraan tidak mencukupi total permintaan yang tersisa. Dengan mekanisme tersebut, potensi kualitas setiap solusi parsial dapat dinilai sejak awal.

Proses *branching* dilakukan dengan menugaskan setiap pelanggan yang belum dikunjungi ke kendaraan yang masih memiliki sisa kapasitas yang memadai. Proses tersebut menghasilkan *child node* baru sebagai representasi solusi lanjutan. Kemudian proses *pruning* dilakukan untuk membatasi perluasan pohon pencarian. Suatu *node* tidak dieksplorasi lebih lanjut apabila nilai *lower bound*-nya melebihi atau sama dengan solusi terbaik yang telah ditemukan sebelumnya (z^*). Dengan demikian, ruang pencarian dapat dikurangi, sehingga proses komputasi dapat dilakukan secara lebih efisien tanpa menghilangkan peluang untuk memperoleh solusi terbaik.

2.5 Algoritma Nearest Neighbor Heuristic

Algoritma Nearest Neighbor Heuristic (NNH) membangun rute secara sekuensial untuk setiap kendaraan. Kendaraan secara berulang memilih pelanggan terdekat dimana dimulai dari depot ke sekolah yang belum dikunjungi dan permintaannya tidak melebihi kapasitas tersisa (*load*) [18]. Pencarian dilakukan dengan mencari nilai minimum dari matriks jarak. Ketika tidak ada lagi pelanggan yang memenuhi syarat, kendaraan kembali ke depot dan rute disimpan. Proses diulang hingga semua pelanggan terlayani atau seluruh kendaraan digunakan. Strategi ini bersifat *greedy* karena keputusan hanya didasarkan pada keuntungan lokal. Pseudocode algoritma NNH ditunjukkan sebagai berikut:

Algorithm: Nearest Neighbor Heuristic for CVRP

Input: depot (index 0), $\text{customers} = \{1, \dots, n\}$, $\text{demand}[i]$, $\text{dist}[i][j]$, capacity, numVehicles

Output: routes, totalDist

unvisited $\leftarrow \text{customers}$

routes $\leftarrow \emptyset$

totalDist $\leftarrow 0$

for $v = 1$ to numVehicles do

 if unvisited = $\emptyset$ then break

 route $\leftarrow [\text{depot}]$

 current $\leftarrow \text{depot}$

 load $\leftarrow \text{capacity}$

 while true do

 nearest $\leftarrow \text{null}$

 minDist $\leftarrow \infty$

 for each $i \in \text{unvisited}$ do

 if $\text{demand}[i] \leq \text{load}$ then

 if $\text{dist}[\text{current}][i] < \text{minDist}$ then

 minDist $\leftarrow \text{dist}[\text{current}][i]$

 nearest $\leftarrow i$

 end if

 end if

```
end for

if nearest = null then break

route ← route + [nearest]
totalDist ← totalDist + minDist
load ← load - demand[nearest]
current ← nearest
unvisited ← unvisited \ {nearest}
end while

route ← route + [depot]
totalDist ← totalDist + dist[current][depot]
routes ← routes ∪ {route}
end for

return routes, totalDist
```

2.6 Evaluasi

Evaluasi Kinerja kedua algoritma dilakukan dengan empat metrik evaluasi. Pemilihan empat metrik berfungsi untuk mengukur kualitas solusi dan efisiensi komputasi secara menyeluruh. Adapun metrik yang digunakan antara lain:

a. Total Jarak tempuh

Metrik ini digunakan untuk mengukur jumlah seluruh jarak tempuh kendaraan. Nilai ini dihitung dari penjumlahan panjang seluruh rute. Semakin kecil nilai total jarak tempuh maka semakin efisien rute yang dihasilkan.

b. Total Komputasi

Evaluasi kinerja total komputasi dilakukan untuk membandingkan efisiensi algoritma BnB dan NNH menggunakan 2 komponen yaitu jumlah akses matriks jarak dan jumlah pengecekan kelayakan. Kedua metrik tersebut dipilih karena dapat diamati pada kedua algoritma dengan definisi operasional yang sama agar lebih adil meskipun keduanya memiliki mekanisme pencarian yang berbeda.

Jumlah akses matriks jarak dihitung setiap kali algoritma membaca satu elemen d_{ij} dari matriks jarak untuk mengevaluasi perpindahan antara dua lokasi. Penghitungan mencakup seluruh akses yang terjadi selama proses pencarian solusi, baik pada evaluasi kandidat, perhitungan biaya rute, maupun perhitungan *lower bound* pada BnB. Komponen ini digunakan untuk menunjukkan intensitas pemanfaatan data jarak oleh masing-masing algoritma.

Jumlah pengecekan kelayakan dihitung setiap kali algoritma mengevaluasi apakah suatu sekolah dapat dimasukkan ke dalam rute kendaraan tanpa melanggar *constraint* CVRP. Agar perbandingan adil, satu pengecekan kelayakan didefinisikan sebagai satu pemanggilan prosedur yang memeriksa status kunjungan pelanggan dan kecukupan kapasitas kendaraan terhadap permintaan pelanggan yang sedang dievaluasi. Dengan kata lain, jumlah pengecekan kelayakan merepresentasikan banyaknya kandidat penugasan yang harus diperiksa oleh masing-masing algoritma selama pembentukan solusi.

c. Rerata Waktu Komputasi

Waktu eksekusi aktual diukur sejak proses pencarian solusi dimulai hingga solusi akhir diperoleh. Pengukuran tidak mencakup proses pengambilan data melalui Google Routes API, pembacaan berkas, pembentukan matriks jarak, pencetakan hasil, dan visualisasi rute. Kedua algoritma dijalankan menggunakan dataset, matriks jarak, perangkat keras, sistem operasi, bahasa pemrograman, serta konfigurasi perangkat lunak yang sama. Untuk mengurangi pengaruh aktivitas sistem di luar program, setiap algoritma dijalankan berulang sebanyak 30 kali, kemudian waktu eksekusi dihitung menggunakan reratanya.

d. Maksimum Panjang Rute

Metrik ini digunakan untuk mengukur panjang rute terpanjang di antara seluruh kendaraan. Nilai ini digunakan untuk melihat rute terburuk yang dihasilkan oleh algoritma.

e. Keseimbangan Jarak Rute Distribusi

Metrik ini digunakan untuk mengukur keseimbangan jarak rute distribusi antar kendaraan. Nilai metrik ini didapatkan dari selisih absolut panjang rute antar kendaraan. Nilai ini digunakan karena jarak distribusi yang seimbang akan lebih mudah dioperasikan dan lebih adil untuk tiap kendaraan.

3. HASIL DAN PEMBAHASAN

Berdasarkan data yang diperoleh dari SIMANGGIS, telah dibentuk matriks jarak berukuran 18x18 yang memuat jarak antar semua *node* termasuk SPPG dan Sekolah penerima dimana 0 adalah SPPG atau depot dan 1 hingga 17 adalah sekolah 1 sampai 17. Selain itu jumlah permintaan tiap sekolah diubah menjadi sebuah *array* demand[]

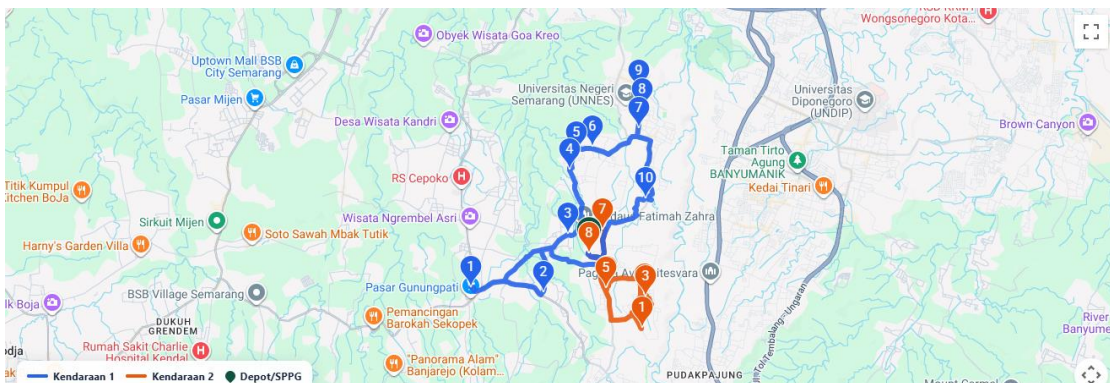
dimana representasi demand[0] merupakan permintaan sekolah 1, begitu pula seterusnya. Adapun cuplikan matriks jarak ditampilkan pada Tabel 4.

Tabel 4. Matriks Jarak

	0	1	2		17
0	0	1.82	4.91	...	1.07
1	1.82	0	4.34	...	1.87
2	4.9	4.32	0	...	5.26
...	...	...	..	...	...
17	1.07	1.87	5.27	...	0

3.1 Hasil BnB

Hasil penerapan BnB menunjukkan bahwa distribusi dapat diselesaikan total panjang rute 26,31 km. Kendaraan 1 melayani 10 sekolah dimana total paket makanan yang dibawa adalah 668 paket dengan rute Depot – S9 – S11 – S1 – S6 – S7 – S5 – S3 – S4 – S2 – S8 – Depot dan menghasilkan panjang rute 18,22 km. Kendaraan 2 melayani 7 sekolah dimana total paket makanan yang dibawa adalah 961 paket dengan rute 0 – S16 – S14 – S15 – S17 – S13 – S10 – S12 – 0 dan menghasilkan panjang rute 8,09 km. Visualisasi rute kendaraan ditampilkan pada Gambar 2. Berdasarkan hasil tersebut menunjukkan bahwa solusi algoritma BnB mampu membentuk rute dengan jarak total yang relatif efisien dan muatan kendaraan tidak melanggar kapasitas yang telah ditentukan. Namun, keseimbangan jarak rute distribusi antarkendaraan belum sepenuhnya seimbang, karena rute kendaraan pertama jauh lebih panjang daripada rute kendaraan kedua. Kondisi ini dapat dilihat dari maksimum panjang rute sebesar 18,22 km, dan selisih jarak rute distribusi sebesar 10,13 km. Tabel 5 menampilkan hasil penerapan BnB.



Gambar 2. Visualisasi Rute Distribusi Menggunakan BnB

Tabel 5. Hasil BnB

Kendaraan	Rute	Banyak Kunjungan	Panjang rute	Total Panjang Rute	Total Komputasi	Rerata Waktu Komputasi
1	0 (depot) – S9 – S11 – S1 – S6 – S7 – S5 – S3 – S4 – S2 – S8 – 0 (depot)	10	18,22 km	26,31 km	2.028.201	1.956 milidetik
2	0 (depot) – S16 – S14 – S15 – S17 – S13 – S10 – S12 – 0 (depot)	7	8,09 km			

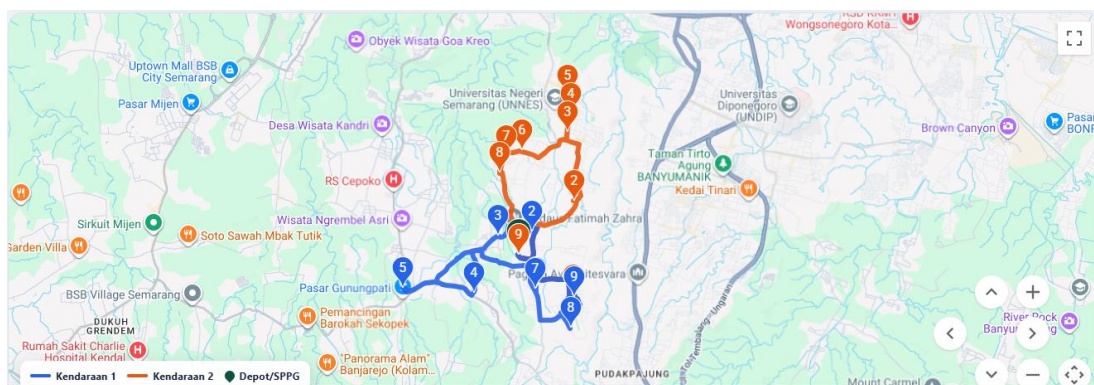
Dari sisi proses komputasi, BnB menunjukkan bahwa algoritma melakukan 2.028.201 kali mengakses array matriks jarak dan 269.306 kali pengecekan kelayakan (isFeasibleAssignment) dengan nilai total komputasi yang dibandingkan adalah 2.297.507 proses. Selain 2 komponen tersebut, BnB melakukan 269.307 kali bound evaluations, 774.740 kali iterasi dalam perhitungan MST dan 269.306 kali anak yang dibuat (sebelum *pruning*). Dari sisi rerata waktu komputasi, BnB membutuhkan rerata waktu 1.956 milidetik untuk menyelesaikan semua langkah algoritma dalam penyelesaian permasalahan ini.

Dilihat dari struktur bebannya, komponen komputasi terbesar ditopang oleh akses array matriks jarak, yang mencapai sekitar 54,20% dari total operasi inti. Setelah itu, beban terbesar berikutnya ditopang oleh iterasi dalam perhitungan MST, yang mencapai sekitar 20,70%. Sementara itu, bound evaluations, pengecekan kelayakan, dan

pembuatan anak masing-masing berkontribusi sekitar 7,20%. Pola ini menunjukkan bahwa beban utama algoritma tidak terletak pada proses percabangan semata, tetapi terutama pada evaluasi jarak antar *node* dan pembentukan *lower bound* berbasis MST.

3.2 Hasil NNH

Hasil penerapan NNH menunjukkan bahwa distribusi dapat diselesaikan dengan total panjang rute 30,59 km. Kendaraan 1 melayani 9 sekolah dimana total paket makanan yang dibawa adalah 979 paket dengan melalui rute Depot – S10 – S12 – S1 – S11 – S9 – S17 – S13 – S16 – S15 – Depot dengan panjang rute 15,24 km, sedangkan Kendaraan 2 melayani 8 sekolah dimana total paket makanan yang dibawa adalah 650 paket melalui rute Depot – S14 – S8 – S3 – S4 – S2 – S5 – S7 – S6 – Depot dengan panjang rute 15,35 km. Visualisasi rute kendaraan ditampilkan pada Gambar 3. Berdasarkan hasil tersebut, solusi NNH mampu membentuk rute yang layak, terstruktur dan muatan tiap kendaraan tidak melebihi kapasitas, meskipun efisiensi jarak globalnya tidak sebaik hasil algoritma exact. Namun jika dilihat dari keseimbangan jarak rute distribusi antar kendaraan, NNH menghasilkan solusi yang seimbang. Hal ini ditunjukkan oleh maksimum panjang rute sebesar 15,35 km, dan selisih panjang rute antarkendaraan sebesar 0,11 km. Tabel 6 menampilkan hasil penerapan NNH.



Gambar 3. Visualisasi Rute Distribusi Menggunakan NNH

Tabel 6. Hasil NNH

Kendaraan	Rute	Banyak Kunjungan	Panjang rute	Total Panjang Rute	Total Komputasi	Rerata Waktu Komputasi
1	Depot - S10 - S12 - S1 - S11 - S9 - S17 - S13 - S16 - S15 - 0 (Depot)	9	15,24	30,59	314	63 milidetik
2	0 (Depot) - S14 - S8 - S3 - S4 - S2 - S5 - S7 - S6 - 0 (Depot)	8	15,35			

Dari sisi proses komputasi, NNH menunjukkan mekanisme yang jauh lebih sederhana dibandingkan pendekatan *exact*. NNH melakukan 153 kali evaluasi jarak melalui akses matriks jarak dan 161 kali pengecekan kandidat sehingga nilai total komputasi yang dibandingkan adalah 314 proses. Selain 2 komponen tersebut, NNH melakukan 19 kali pencarian tetangga terdekat dan 153 kali perbandingan jarak pada proses pemilihan pelanggan. Hasil ini menunjukkan bahwa seluruh biaya komputasi NN sepenuhnya dibentuk oleh proses pencarian pelanggan *feasible* dan proses perbandingan jarak lokal. Pada aspek rerata waktu komputasi, NNH membutuhkan 63 milidetik untuk menyelesaikan semua langkah algoritma dalam penyelesaian permasalahan ini.

3.3 Evaluasi

Berdasarkan hasil pengujian, perbedaan kinerja yang jelas antara BnB dan NNH telah ditunjukkan. Pada penelitian ini, tidak ditemukan satu algoritma yang unggul pada seluruh metrik evaluasi. Sebaliknya, terdapat *trade-off* antara kualitas solusi rute dan efisiensi operasional pada kedua algoritma. Pada Tabel 7 ditampilkan perbandingan BnB dan NNH berdasarkan empat metrik evaluasi yang digunakan.

Tabel 7. Perbandingan Hasil BnB dan NNH

Metrik Evaluasi	BnB	NNH	Algoritma yang Lebih Unggul	Keterangan
Total Jarak	26,31 km	30,59 km	BnB	BnB menghasilkan jarak 14% lebih pendek.
Total Komputasi	2.297.507	314	NNH	NNH membutuhkan 99,99% lebih sedikit proses.

Rerata Waktu Komputasi	1956	63	NNH	NNH 96,78% lebih cepat.
Panjang Rute Maksimum	18,22 km	15,35 km	NNH	Rute maksimum NNH 15,75% lebih pendek.
Keseimbangan Jarak Rute	10,13 km	0,11 km	NNH	NNH 98,91% lebih seimbang.

Pada metrik total jarak, BnB menunjukkan hasil yang lebih baik dengan total jarak sebesar 26,31 km, sedangkan NNH menghasilkan total jarak 26,31 km, atau 14% lebih pendek dibandingkan NNH yang menghasilkan jarak 30,59 km. Namun, keunggulan kualitas solusi tersebut disertai kebutuhan komputasi yang jauh lebih besar. NNH membutuhkan 99,99% lebih sedikit operasi komputasi dan memiliki rerata waktu eksekusi 96,78% lebih singkat daripada BnB. NNH juga menghasilkan rute maksimum 15,75% lebih pendek serta selisih jarak antarkendaraan 98,91% lebih rendah. Hasil tersebut menunjukkan bahwa BnB lebih unggul dari sisi optimalitas total jarak, sedangkan NNH lebih efisien dari sisi komputasi dan menghasilkan distribusi jarak antarkendaraan yang lebih seimbang.

3.4 Pembahasan

Perbedaan kinerja antara BnB dan NNH tidak hanya telah ditentukan oleh hasil akhir yang diperoleh, tetapi juga telah dibentuk oleh mekanisme internal masing-masing algoritma. Pada BnB, ruang solusi dieksplorasi secara sistematis melalui proses *branching*, *bounding*, dan *pruning*. Melalui mekanisme tersebut, banyak solusi parsial dibangkitkan, dievaluasi, lalu dieliminasi apabila tidak lagi berpotensi menghasilkan solusi yang lebih baik. Dengan mekanisme tersebut, solusi dengan total jarak yang lebih kecil dapat ditemukan karena pencarian tidak hanya diarahkan pada keputusan lokal, tetapi pada kombinasi rute yang paling efisien secara global. Akan tetapi, konsekuensi dari proses tersebut terlihat pada tingginya kebutuhan komputasi. Dikarenakan fungsi tujuan utama hanya diarahkan pada minimisasi total jarak, keseimbangan panjang rute antar kendaraan tidak dapat dioptimalkan secara langsung. Hal ini mengakibatkan solusi yang dihasilkan merupakan solusi yang efisien secara global namun terdapat keseimbangan jarak rute distribusi masih kurang merata. Temuan pada penelitian ini sejalan dengan perkembangan penelitian terbaru pada bidang *Vehicle Routing Problem* yang menunjukkan bahwa algoritma exact berbasis pencarian sistematis masih memiliki keunggulan dalam menghasilkan solusi global terbaik. Penelitian yang dilakukan oleh Zhang et al. (2022) menunjukkan bahwa pendekatan exact pada CVRP melalui mekanisme *branch-and-price-and-cut* mampu menghasilkan solusi dengan kualitas optimal yang lebih baik dibandingkan pendekatan heuristik, khususnya ketika fungsi objektif utama difokuskan pada minimisasi total biaya distribusi [29]. Hal tersebut memperkuat hasil penelitian ini bahwa mekanisme eksplorasi ruang solusi secara menyeluruh pada algoritma BnB memungkinkan diperolehnya total jarak distribusi yang lebih kecil dibandingkan pendekatan heuristik.

Pada NNH, solusi dibentuk melalui aturan greedy yang sederhana dimana algoritma akan selalu memilih pelanggan feasible yang paling dekat dari posisi kendaraan saat ini. Proses tersebut menyebabkan pencarian dilakukan secara langsung, tanpa eksplorasi cabang solusi yang luas. Dengan prosedur tersebut, NN Heuristic menghasilkan kebutuhan komputasi yang sangat rendah. Namun dikarenakan keputusan hanya didasarkan pada informasi lokal pada setiap langkah, jaminan untuk memperoleh solusi global terbaik tidak dapat diberikan. Konsekuensinya, total jarak yang dihasilkan tidak lebih baik daripada BnB. Meskipun demikian, pola pembentukan rute secara bertahap dari tetangga terdekat cenderung menghasilkan rute yang lebih kompak dan lebih seragam antar kendaraan. Dengan demikian, panjang rute maksimum yang lebih kecil dan pemerataan jarak rute antar kendaraan yang lebih baik dapat dihasilkan oleh NN, walaupun efisiensi globalnya lebih rendah. Hasil penelitian ini juga konsisten dengan penelitian Cavaliere et al. (2022) yang menunjukkan bahwa pendekatan heuristik pada CVRP cenderung memiliki kebutuhan komputasi yang jauh lebih rendah karena proses pencarian solusi dilakukan secara lokal tanpa eksplorasi seluruh ruang solusi [30]. Penelitian tersebut menegaskan bahwa algoritma heuristik memiliki keunggulan utama pada efisiensi proses komputasi sehingga lebih realistis diterapkan pada sistem distribusi berskala besar yang membutuhkan pengambilan keputusan secara cepat. Kondisi tersebut sejalan dengan hasil penelitian ini dimana algoritma NNH hanya membutuhkan 486 operasi komputasi, jauh lebih kecil dibandingkan BnB yang membutuhkan 3.610.860 operasi komputasi.

Dalam sistem distribusi nyata, rute dengan jarak minimum memang dibutuhkan, tetapi kecepatan perhitungan dan pemerataan jarak antar kendaraan juga perlu dipertimbangkan. Rute yang lebih efisien secara global mempunyai potensi penghematan perjalanan kendaraan dan penggunaan sumber daya distribusi yang lebih baik. Namun Kecepatan perhitungan yang realistis perlu dipertimbangkan karena sistem distribusi nyata menuntut solusi yang dapat dihasilkan dalam waktu yang sesuai dengan kebutuhan operasional. Selain itu, pemerataan jarak rute antar kendaraan juga perlu diperhatikan karena sistem distribusi yang baik tidak hanya mengejar jarak total minimum, tetapi juga menjaga agar tidak ada satu kendaraan yang menerima pembagian jarak perjalanan secara berlebihan. Keseimbangan jarak rute distribusi yang lebih merata akan mendukung kestabilan jadwal,

mempermudah pengelolaan armada, dan menurunkan risiko keterlambatan layanan. Oleh sebab itu, penerapan algoritma pada distribusi MBG tidak hanya perlu didasarkan pada satu metrik, tetapi perlu ditentukan melalui pertimbangan beberapa metrik evaluasi secara bersamaan. Jika dibandingkan dengan penelitian-penelitian terkini, fokus evaluasi algoritma *routing* saat ini tidak lagi hanya terbatas pada pencarian rute dengan jarak minimum, tetapi juga mulai mempertimbangkan aspek keseimbangan distribusi, efisiensi komputasi, dan fleksibilitas operasional. Simensen et al. (2022) menunjukkan bahwa pada implementasi distribusi nyata, solusi yang paling optimal secara matematis belum tentu menjadi solusi terbaik secara operasional apabila menghasilkan keseimbangan jarak rute distribusikendaraan yang tidak seimbang [31]. Perspektif tersebut mendukung hasil penelitian ini dimana algoritma BnB menghasilkan total jarak minimum, tetapi keseimbangan jarak rute distribusi kendaraan menjadi kurang merata, sedangkan algoritma NNH menghasilkan keseimbangan distribusi yang jauh lebih baik.

Selain itu, penelitian Hamdi (2024) menegaskan bahwa perkembangan model CVRP modern semakin mengarah pada pendekatan *multi-objective optimization*, yaitu optimasi yang mempertimbangkan beberapa indikator performa secara simultan, seperti efisiensi rute, kapasitas kendaraan, waktu distribusi, serta keseimbangan operasional [32]. Dalam konteks distribusi Program Makan Bergizi Gratis (MBG), hasil penelitian ini menunjukkan bahwa pemilihan algoritma tidak dapat hanya berorientasi pada satu indikator optimasi, melainkan perlu mempertimbangkan kebutuhan operasional sistem distribusi layanan publik secara lebih komprehensif.

Selain aspek metodologis, penelitian ini juga memberikan kontribusi empiris yang berbeda dibandingkan sebagian besar penelitian terdahulu pada CVRP. Sebagian besar studi sebelumnya menggunakan *benchmark dataset* atau simulasi distribusi pada sektor komersial seperti logistik barang, distribusi produk, maupun pengangkutan sampah. Sementara itu, penelitian ini menggunakan kasus nyata distribusi Program Makan Bergizi Gratis (MBG) yang merupakan sistem distribusi layanan publik dengan karakteristik permintaan yang heterogen, keterbatasan kapasitas kendaraan, serta tuntutan ketepatan distribusi dalam skala operasional harian. Dengan demikian, hasil penelitian ini memperluas penerapan studi optimasi CVRP pada konteks distribusi program pemerintah, sekaligus menunjukkan bahwa evaluasi algoritma pada sistem distribusi nyata memerlukan pertimbangan yang lebih luas dibandingkan sekadar pencapaian solusi optimum secara matematis.

4. KESIMPULAN

Pada penelitian ini, perbandingan BnB dan NNH pada kasus distribusi MBG di SPPG Mangunsari Kecamatan Gunungsari – Kota Semarang telah dilakukan dengan menggunakan empat metrik evaluasi, yaitu total jarak, total komputasi, panjang rute maximum, dan pemerataan jarak rute antar kendaraan. Berdasarkan hasil pengujian, BnB menunjukkan keunggulan pada aspek efisiensi jarak dengan total jarak sebesar 26,31 km. sedangkan NNH menghasilkan total distance sebesar 30,59 km. Hasil tersebut menunjukkan bahwa BnB menghasilkan rute 14% lebih efisien secara global. Sebaliknya, NNH menunjukkan keunggulan pada aspek keseimbangan jarak rute dan efisiensi komputasi, baik proses komputasi maupun rerata waktu komputasi. NNH menghasilkan panjang rute maximum sebesar 15,35 km dan selisih panjang rute antar kendaraan sebesar 0,11 km. Sedangkan BnB menghasilkan panjang rute maximum sebesar 18,22 km dan panjang rute antar kendaraan sebesar 10,13 km. dengan kata lain, NNH mempunyai panjang rute maximum 15,75% lebih pendek dan selisih panjang rute antar kendaraan 98,91% lebih seimbang daripada BnB. Selain itu, NNH menghasilkan total komputasi sebesar 314 langkah atau 99,99% lebih sedikit proses daripada BnB dimana BnB membutuhkan 2.297.607 langkah komputasi. Pada segi rerata waktu komputasi, NNH 96,78% lebih cepat dimana membutuhkan waktu proses 63 milidetik sedangkan BnB membutuhkan waktu proses 1956 milidetik. Dengan demikian, NNH menghasilkan keseimbangan jarak rute distribusi yang lebih merata meskipun efisiensi jarak global yang dihasilkan masih lebih rendah serta proses komputasi yang lebih ringan dan cepat. Secara keseluruhan, tidak ada satu algoritma yang dapat dinyatakan unggul secara mutlak pada seluruh metrik evaluasi dalam kasus distribusi MBG di SPPG Mangunsari Kecamatan Gunungsari – Kota Semarang. BnB lebih tepat digunakan ketika pencarian rute dengan jarak total minimum menjadi prioritas utama, sedangkan NNH lebih tepat digunakan ketika prioritas utama diarahkan pada kecepatan perhitungan dan keseimbangan jarak rute distribusi antar kendaraan. Oleh karena itu, pemilihan algoritma pada distribusi MBG perlu dilakukan secara kontekstual, sesuai dengan tujuan optimasi yang ingin dicapai. Sebagai catatan, hasil penelitian ini hanya terbatas pada kasus distribusi MBG di SPPG Mangunsari Kecamatan Gunungsari – Kota Semarang dan belum dapat digeneralisasi untuk seluruh distribusi MBG pada Kota Semarang. Pada penelitian selanjutnya, pengembangan dapat diarahkan pada penggunaan algoritma yang mampu memberikan solusi dengan efisiensi jarak, dan keseimbangan jarak rute distribusi namun mempunyai komputasi yang relatif lebih rendah. Selain itu diperlukan penelitian lanjutan dimana permasalahan distribusi MBG dengan menambahkan konstrain jendela waktu tiap sekolah.

REFERENCES

- [1] Anonim, “UU No. 59 Tahun 2024,” Undang-undang (UU) Nomor 59 Tahun 2024. Accessed: Apr. 27, 2026. [Online]. Available: <https://peraturan.bpk.go.id/Details/299728/uu-no-59-tahun-2024>
- [2] Badan Pemeriksa Keuangan, “Perpres No. 83 Tahun 2024,” Peraturan Presiden (Perpres) Nomor 83 Tahun 2024. Accessed: Apr. 27, 2026. [Online]. Available: <https://peraturan.bpk.go.id/Details/295857/perpres-no-83-tahun-2024>
- [3] Badan Pemeriksa Keuangan, “Perpres No. 115 Tahun 2025,” Peraturan Presiden (Perpres) Nomor 115 Tahun 2025. Accessed: Apr. 27, 2026. [Online]. Available: <https://peraturan.bpk.go.id/Details/343430/perpres-no-115-tahun-2025>
- [4] Badan Gizi Nasional, “Petunjuk Teknis Tata Kelola Penyelenggaraan Program Makan Bergizi Gratis,” Badan Gizi Nasional. Accessed: Apr. 27, 2026. [Online]. Available: <https://www.bgn.go.id/juknis/lb1PKg-petunjuk-teknis-tata-kelola-penyelenggaraan-program-makan-bergizi-gratis%5D>
- [5] Badan Gizi Nasional, “Pedoman Umum Sistem dan Tata Kelola badan Gizi Nasional Untuk Program Makan Bergizi Gratis” Badan Gizi Nasional. Accessed: May 12, 2026. [Online]. Available: <https://www.bgn.go.id/juknis/LEP9KP-pedoman-umum-sistem-dan-tata-kelola-badan-gizi-nasional-untuk-program-makan-bergizi-gratis>
- [6] MBG Kota Semarang, “MBG Kota Semarang — Dasbor.” Accessed: Apr. 27, 2026. [Online]. Available: https://si-manggis.vercel.app/sppg_aktif.html
- [7] P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications*. SIAM, 2014.
- [8] D. R. Morrison, S. H. Jacobson, J. J. Sauppe, and E. C. Sewell, “Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning,” *Discrete Optim.*, vol. 19, pp. 79–102, 2016.
- [9] A. S. Fukunaga, “A branch-and-bound algorithm for hard multiple knapsack problems,” *Ann. Oper. Res.*, vol. 184, no. 1, pp. 97–119, 2011.
- [10] S. Tanaka and K. Tierney, “A branch and bound approach for large pre-marshalling problems,” *Eur. J. Oper. Res.*, vol. 278, no. 1, pp. 211–225, 2019.
- [11] Z. Wang and Q. Zeng, “A branch-and-bound approach for AGV dispatching and routing problems in automated container terminals,” *Comput. Ind. Eng.*, vol. 166, p. 107968, 2022.
- [12] B. Pamuk, T. Oncan, and I. K. Altinel, “An efficient branch-and-bound algorithm for the one-to-many shortest path problem with additional disjunctive conflict constraints,” *Eur. J. Oper. Res.*, vol. 324, no. 2, pp. 398–413, 2025.
- [13] F. Theurich, A. Fischer, and G. Scheithauer, “A branch-and-bound approach for a Vehicle Routing Problem with Customer Costs,” *EURO J. Comput. Optim.*, vol. 8, no. 3, 2020.
- [14] S. Dhanasekar, S. K. Dash, and N. Uthaman, “A Branch and Bound Algorithm to Solve Travelling Salesman Problem (TSP) with Uncertain Parameters,” *Math. Stat.*, vol. 10, no. 2, pp. 358–365, 2022.
- [15] S. P. Revika, J. Nurhakiki, B. Naysabilla, and S. S. B. Ginting, “Systematic Literature Review: Penerapan Metode Branch and Bound dalam Optimalisasi Produksi,” *BILANGAN*, vol. 3, no. 3, pp. 141–155, 2025.
- [16] F. A. A. Situnggang and N. Napitupulu, “Implementation of Branch and Bound Algorithm to Solve the Travelling Salesman Problem at PT Jasa Harapan Barat,” *J. Math.*, vol. 1, no. 1, pp. 1–7, 2023.
- [17] E. Sanggala and M. A. Bisma, “Penyelesaian Capacitated Vehicle Routing Problem (CVRP) dengan Nearest Neighbour (Studi Kasus: Russian CVRP Instances),” *JUTIN J. Tek. Ind. Terintegrasi*, vol. 8, no. 3, pp. 2586–2599, 2025, doi: 10.31004/jutin.v8i3.46463.
- [18] I. Masudin, R. F. Sa’diyah, D. M. Utama, D. P. Restuputri, and F. Jie, “Capacitated Vehicle Routing Problems: Nearest Neighbour vs. Tabu Search,” *Int. J. Comput. Theory Eng.*, vol. 11, no. 4, pp. 76–79, 2019, doi: 10.7763/IJCTE.2019.V11.1246.
- [19] E. E. Rosyida, Sugianto, and I. B. Efendi, “Capacitated Vehicle Routing Problem (CVRP) with Sweep and Nearest Neighbor Algorithm,” *Sinergi Int. J. Logist.*, vol. 2, no. 1, pp. 17–29, 2024.
- [20] F. Liu, C. Lu, L. Gui, Q. Zhang, X. Tong, and M. Yuan, “Heuristics for Vehicle Routing Problem: A Survey and Recent Advances,” *ArXiv Prepr.*, 2023.
- [21] A. S. Hameed, H. M. B. Alrikabi, A. A. Abdul-Razaq, H. K. Nasser, M. L. Mutar, and H. H. Katea, “A Detailed Review of the Capacitated Vehicle Routing Problem: Model, Computational Complexity, Solutions, and Practical Applications,” *J. Internet Serv. Inf. Secur.*, vol. 15, no. 1, pp. 218–235, 2025, doi: 10.58346/JISIS.2025.I1.014.
- [22] K. Buyukozdemir, A. Bas, K. Yildiz, and B. C. Uslu, “A Review of Heuristic Approaches to Vehicle Routing Problems,” presented at the 2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), 2021. doi: 10.1109/CSDE53843.2021.9718378.
- [23] S. Z. Abidin, N. I. Jaini, and H. Daud, “Decision-Making Support in Vehicle Routing Problems: A Review of Recent Literature,” *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 44, no. 2, pp. 124–134, 2025, doi: 10.37934/araset.44.2.124134.
- [24] J. Bauß, S. N. Parragh, and M. Stiglmayr, “On improvements of multi-objective branch and bound,” *EURO J. Comput. Optim.*, vol. 12, p. 100099, 2024.

- [25] D. Pecin, A. Pessoa, M. Poggi, and E. Uchoa, “Improved branch-cut-and-price for capacitated vehicle routing,” *Math. Program. Comput.*, vol. 9, no. 1, pp. 61–100, Mar. 2017, doi: 10.1007/s12532-016-0108-8.
- [26] C. Archetti and M. G. Speranza, “A survey on matheuristics for routing problems,” *EURO J. Comput. Optim.*, vol. 2, no. 4, pp. 223–246, Nov. 2014, doi: 10.1007/s13675-014-0030-7.
- [27] D. Pecin, C. Contardo, G. Desaulniers, and E. Uchoa, “New Enhancements for the Exact Solution of the Vehicle Routing Problem with Time Windows,” *Inf. J. Comput.*, vol. 29, no. 3, pp. 489–502, Aug. 2017.
- [28] R. Fukasawa, J. Lysgaard, M. P. de Aragao, M. Reis, E. Uchoa, and R. F. Werneck, “Robust Branch-and-Cut-and-Price for the Capacitated Vehicle Routing Problem”.
- [29] X. Zhang, L. Chen, M. Gendreau, and A. Langevin, “A Branch-and-Price-and-Cut Algorithm for the Vehicle Routing Problem with Two-Dimensional Loading Constraints,” *Transp. Sci.*, vol. 56, no. 6, pp. 1618–1635, Nov. 2022, doi: 10.1287/trsc.2022.1135.
- [30] F. Cavaliere, E. Bendotti, and M. Fischetti, “An integrated local-search/set-partitioning refinement heuristic for the Capacitated Vehicle Routing Problem,” *Math. Program. Comput.*, vol. 14, no. 4, pp. 749–779, Dec. 2022, doi: 10.1007/s12532-022-00224-2.
- [31] M. Simensen, G. Hasle, and M. Stålhane, “Combining hybrid genetic search with ruin-and-recreate for solving the capacitated vehicle routing problem,” *J. Heuristics*, vol. 28, no. 5–6, pp. 653–697, Dec. 2022, doi: 10.1007/s10732-022-09500-9.
- [32] Imen Hamdi, “Solving the cumulative capacitated vehicle routing problem with drones,” *J. Ind. Prod. Eng.*, vol. 41, no. 4, pp. 344–361, 2024, doi: <https://doi.org/10.1080/21681015.2024.2304570>.