JSAI : Journal Scientific and Applied Informatics
Vol. 09, No. 3, *October* 2026, *pages*. 479~486
E-ISSN: 2614-3054; P-ISSN: 2614-3062, *accredited by Kemenristekdikti*, Sinta 4
DOI: 10.36085

Cheapest Insertion Heuristic dan *Time Window Batching* untuk Rute dan Harga Dinamis Aplikasi Kurir *Online Web-Mobile*

[1]Dimas Prabowo Wahyudi, [2]Sutarman
[1,2]Universitas Teknologi Yogyakarta, Indonesia
[1]*prabowodimas715@gmail.com*; [2]*sutarman@uty.ac.id*

Article Info

Article history:

Received, 2026-10-19
Revised, 2026-11-16
Accepted, 2026-11-22

Kata Kunci:

cheapest_insertion_heuristic
harga_dinamis
time_window_batching
kurir_online
web_mobile

ABSTRAK

Aplikasi kurir *online* umumnya menghitung ongkos kirim dari tarif zona atau jarak titik ke titik, sehingga rute yang lebih pendek saat kurir membawa beberapa paket tidak menurunkan tarif yang dibayar pelanggan. Pembagian tarif antara kurir dan platform juga jarang ditampilkan. Penelitian ini membangun aplikasi kurir online berbasis *web-mobile* yang menerapkan algoritma *Cheapest Insertion Heuristic* untuk menyusun urutan kunjungan, menghitung tarif dinamis dari jarak rute, dan menggabungkan pesanan searah melalui time *window batching*. Pesanan yang masuk dalam jendela waktu 15 menit dan titik tujuannya berjarak paling jauh 2 km dari pesanan pertama kelompok digabung ke dalam satu rute dengan tarif bersama. Sistem dibangun dengan Laravel 11, Flutter, dan MySQL, sedangkan jarak dihitung dengan formula *Haversine* dari koordinat hasil *geocoding OpenStreetMap*. Simulasi pada lima skenario pengiriman di Surabaya menunjukkan bahwa rute yang dihasilkan rata-rata 29,37% lebih pendek dibanding urutan input dan lebih pendek dari metode *Nearest Neighbor* pada kelima skenario. Pada tiga skenario penggabungan, tarif bersama rata-rata 36,41% lebih rendah dibanding total tarif individual, sementara pendapatan kurir per kilometer naik 4,9% sampai 47,8%. Kesepuluh skenario pengujian *black-box* memberikan hasil sesuai harapan.

Keywords:

cheapest_insertion_heuristic
dynamic_pricing
time_window_batching
online_courier
web_mobile

ABSTRACT

Online courier applications usually price deliveries by zone or by point-to-point distance, so the shorter route a courier takes when carrying several parcels did not lower the customer's fare. The split of the fare between courier and platform was also rarely shown. This study built a web-mobile online courier application that applied the Cheapest Insertion Heuristic algorithm to order delivery stops, computed a dynamic fare from the route distance, and merged orders heading in the same direction through time window batching. Orders received within a 15-minute window whose destinations lay within 2 km of the first order in a group were merged into one route with a shared fare. The system was built with Laravel 11, Flutter, and MySQL, and distances were computed with the Haversine formula from coordinates obtained through OpenStreetMap geocoding. A simulation on five delivery scenarios in Surabaya showed that the resulting routes were on average 29.37% shorter than the input order and shorter than the Nearest Neighbor method in all five scenarios. In three batching scenarios, shared fares were on average 36.41% lower than the sum of individual fares, while courier earnings per kilometre rose by 4.9% to 47.8%. All ten black-box test scenarios produced the expected results.

This is an open access article under the CC BY-SAlicense.

Penulis Korespondensi:

Dimas Prabowo Wahyudi,
Program Studi Informatika,
Universitas Teknologi Yogyakarta,
Email: prabowodimas715@gmail.com

1. PENDAHULUAN

Layanan pengiriman berbasis aplikasi berkembang seiring meningkatnya kebutuhan distribusi barang secara cepat dan fleksibel. Dalam model layanan ini, kurir menjadi bagian penting dari proses pemenuhan pesanan, sementara struktur tarif dan pembagian pendapatan memengaruhi insentif kerja kurir. Studi mengenai pengemudi transportasi daring menunjukkan bahwa potongan komisi platform dapat berkaitan dengan pendapatan bersih pengemudi, sedangkan target pendapatan harian dan tekanan penumpang berhubungan dengan perilaku berkendara berisiko [1], [2]. Pada *platform ride-hailing*, mekanisme surge pricing juga memiliki konsekuensi distribusional: perubahan harga dapat memengaruhi kesejahteraan pengguna dan surplus pengemudi, sehingga rancangan harga perlu mempertimbangkan insentif para pihak [3], [4]. Temuan tersebut menunjukkan pentingnya transparansi perhitungan tarif dan pembagian pendapatan, khususnya ketika beberapa pesanan dilayani dalam satu perjalanan.

Salah satu komponen yang dapat memengaruhi biaya pengiriman adalah urutan kunjungan kurir ke sejumlah titik tujuan. Permasalahan penentuan urutan tersebut berkaitan dengan *Traveling Salesman Problem* (TSP), yang bertujuan membentuk tur dengan total jarak atau biaya perjalanan yang rendah. *Cheapest* Insertion *Heuristic* (CIH) merupakan metode konstruksi rute yang menyisipkan titik tujuan pada sisi rute dengan tambahan jarak terkecil. Rosenkrantz et al. membahas karakteristik dan batas kinerja sejumlah heuristik TSP, termasuk metode insertion [5]. Pada penerapan distribusi, CIH telah dibandingkan dengan *Nearest Neighbor* dan digunakan dalam berbagai konteks, seperti distribusi barang, pengangkutan sampah, dan penentuan rute distribusi [6], [7], [8], [9], [10]. Penelitian-penelitian tersebut memperlihatkan pemanfaatan heuristik untuk membentuk urutan kunjungan, tetapi fokus utamanya masih pada pembentukan rute atau evaluasi jarak. Dengan demikian, hasil optimasi rute belum selalu dihubungkan secara langsung dengan mekanisme penetapan tarif yang dibayar pelanggan dan perhitungan bagian pendapatan kurir.

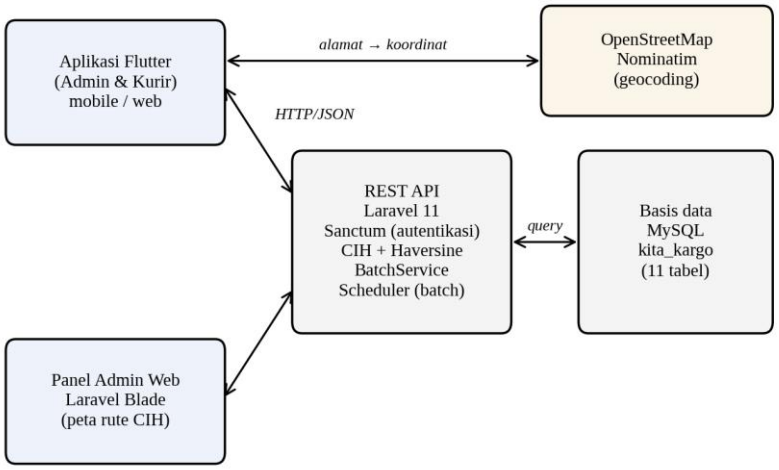
Selain optimasi rute, penggabungan pesanan atau order batching merupakan pendekatan untuk melayani beberapa permintaan dalam satu kelompok perjalanan. Keputusan mengenai kapan pesanan mulai digabungkan menjadi penting karena jendela waktu yang lebih panjang dapat memberi kesempatan terbentuknya kelompok yang lebih efisien, tetapi juga menambah waktu tunggu pelanggan. Gil-Borrás et al. menunjukkan bahwa strategi time window memengaruhi kinerja online order batching [11]. Penelitian lain mengkaji batching dan matching untuk pengantaran makanan pada jaringan jalan dinamis [12], serta integrasi penjadwalan *batching* dan pengiriman pada perdagangan elektronik [13]. Kajian taksonomi order *batching* juga memperlihatkan bahwa keputusan pembentukan kelompok, penentuan waktu, dan pengaturan pengiriman merupakan bagian-bagian yang saling berkaitan [14]. Di sisi lain, penelitian mengenai pemilihan dan penetapan harga pengiriman dinamis menunjukkan bahwa keputusan layanan dan harga dapat dirancang secara adaptif terhadap karakteristik permintaan [15], sedangkan skema kompensasi dinamis bagi pengemudi sesekali menjadi salah satu pendekatan untuk mengelola pasokan kapasitas pengiriman [16]. Meskipun demikian, kajian-kajian tersebut belum secara khusus menjadi satu mekanisme aplikasi kurir yang menghubungkan hasil CIH, penggabungan pesanan berbasis *time window*, tarif berdasarkan jarak rute gabungan, dan pembagian tarif antara kurir dengan platform.

Berdasarkan kondisi tersebut, penelitian ini mengembangkan aplikasi kurir online Kita Kargo berbasis *web-mobile* dengan mengintegrasikan tiga mekanisme, yaitu CIH untuk menyusun urutan kunjungan, perhitungan tarif dinamis berdasarkan jarak rute, dan *time window batching* untuk menggabungkan pesanan dengan tujuan yang berdekatan. Kontribusi penelitian bukan berupa pengembangan algoritma CIH atau konsep *batching* yang baru, melainkan integrasi operasional ketiga mekanisme tersebut ke dalam satu aplikasi yang menampilkan rincian tarif pelanggan serta bagian pendapatan kurir dan platform. Sistem menerapkan jendela waktu 15 menit dan radius 2 km dari titik pesanan pertama sebagai jangkar kelompok. Kinerja mekanisme dievaluasi melalui lima skenario simulasi optimasi rute, tiga skenario simulasi batching, dan sepuluh skenario pengujian *black-box*. Oleh karena itu, tujuan penelitian adalah menganalisis perubahan jarak rute dibandingkan urutan input dan *Nearest Neighbor*, membandingkan tarif batch dengan total tarif individual, serta mengamati pendapatan kurir per kilometer pada skenario simulasi. Hasil penelitian dibatasi pada data lokasi simulasi dan belum dimaksudkan untuk merepresentasikan kinerja operasional pada data pesanan riil.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan *Design and Development Research* (DDR) yang meliputi empat tahap, yaitu analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Analisis kebutuhan dilakukan melalui studi literatur, kemudian kebutuhan tersebut diterjemahkan ke dalam rancangan aplikasi dan aturan pengolahan pengiriman. Sistem dibatasi pada satu depot dan satu jenis kendaraan dengan dua kelompok pengguna internal. Admin atau operator mengelola pelanggan, pengiriman, *delivery*, *batch*, dan karyawan, sedangkan kurir menerima tugas pengantaran harian melalui aplikasi mobile. Pemesanan melalui WhatsApp dan agen fisik tidak termasuk dalam cakupan penelitian.

Arsitektur sistem Kita Kargo dipertahankan seperti ditunjukkan pada Gambar 1. Aplikasi Flutter digunakan oleh admin/operator dan kurir serta berkomunikasi dengan REST API Laravel 11 melalui HTTP/JSON dengan autentikasi *Laravel Sanctum.* Panel admin web dibangun menggunakan *Laravel Blade* dan mengakses *backend* yang sama. *Backend* mengelola proses autentikasi, perhitungan Haversine dan CIH, pembentukan batch, perhitungan tarif, serta pencatatan aktivitas. Data disimpan pada basis data MySQL yang terdiri atas 11 tabel, yaitu *users, customers, vehicles, shipments, deliveries, delivery_routes, route_optimizations, tracking_logs, activity*_logs, route_access_logs, dan *batch_pools.* Alamat tujuan dikonversi menjadi koordinat lintang dan bujur menggunakan layanan *geocoding Nominatim* berbasis data *OpenStreetMap* (© kontributor *OpenStreetMap*, lisensi ODbL).



Gambar 1. Arsitektur sistem Kita Kargo

Jarak geografis antara dua titik dihitung dari koordinat lintang (φ) dan bujur (λ) menggunakan formula Haversine pada Persamaan (1) dan (2). R menyatakan jari-jari bumi sebesar 6.371 km, sedangkan Δφ dan Δλ merupakan selisih lintang dan bujur kedua titik dalam radian. Nilai d yang dihasilkan merupakan jarak garis lengkung terpendek di permukaan bumi (*great-circle distance*), bukan jarak perjalanan kendaraan pada jaringan jalan. Dengan demikian, seluruh jarak dalam simulasi optimasi rute merupakan estimasi geografis berdasarkan koordinat hasil *geocoding*, dan angka penghematan tidak dapat langsung ditafsirkan sebagai penghematan jarak operasional di jalan.

$$a = \sin^2(\Delta\varphi/2) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2(\Delta\lambda/2) \quad (1)$$

$$d = 2R \cdot \arcsin(\sqrt{a}) \quad (2)$$

Cheapest Insertion Heuristic (CIH) merupakan metode konstruksi rute untuk menyelesaikan TSP dengan menyisipkan satu titik tujuan pada sisi rute yang menghasilkan tambahan jarak paling kecil [5]. Dalam penelitian ini, rute awal dibentuk dari depot (D), menuju titik tujuan yang memiliki jarak *Haversine* terdekat dari depot, kemudian kembali ke depot. Untuk setiap titik k yang belum dimasukkan dan setiap sisi berurutan (i, j) pada rute, tambahan jarak penyisipan dihitung menggunakan Persamaan (3). Pasangan titik dan sisi dengan nilai tambahan jarak terkecil dipilih, kemudian titik tersebut disisipkan di antara i dan j. Proses diulang hingga seluruh titik tujuan masuk ke dalam rute. Karena rute selalu diawali dan diakhiri di depot, keluaran CIH pada tahap optimasi ini merupakan rute tertutup. Pseudocode pada Tabel 1 menunjukkan urutan proses tersebut. Pada modul delivery, rute awal dan rute hasil CIH beserta jaraknya disimpan pada tabel *route_optimizations*, sedangkan urutan kunjungan paket disimpan pada *delivery_routes*. Pada proses batch, hasil rute setiap kelompok disimpan pada tabel *batch_pools*.

$$c(i, k, j) = d(i, k) + d(k, j) - d(i, j) \quad (3)$$

Algoritma 1 : *Pseudecode Cheapest insertion heuristic*

```
============================================================
  ALGORITMA: CHEAPEST INSERTION HEURISTIC
============================================================
Input  : depot D, himpunan titik tujuan T
Output : rute optimal dan total jarak

 1. k₀ ← titik di T dengan d(D, k) terkecil
 2. rute ← [D, k₀, D];  T ← T – {k₀}
```

```
 3. selama T tidak kosong:
 4.     untuk setiap k di T dan setiap sisi (i, j) pada rute:
 5.         hitung c(i, k, j) = d(i, k) + d(k, j) − d(i, j)
 6.     pilih (i', k', j') dengan c terkecil
 7.     sisipkan k' di antara i' dan j';  T ← T − {k'}
 8. kembalikan rute dan total jarak
============================================================
```

Tarif pengiriman dihitung berdasarkan jarak yang digunakan pada jenis layanan, dengan tarif per kilometer (r) sebesar Rp3.000 dan jarak (D) dalam kilometer sebagaimana Persamaan (4). Untuk pesanan individual, D merupakan jarak satu arah dari depot ke titik tujuan. Untuk pesanan batch, D merupakan panjang rute terbuka hasil urutan kunjungan CIH, yaitu jarak dari depot melalui seluruh titik tujuan hingga titik terakhir tanpa menambahkan perjalanan kembali ke depot. Dengan demikian, rute tertutup digunakan untuk evaluasi optimasi CIH, sedangkan rute terbuka digunakan sebagai dasar tarif *batch.* Kedua ukuran jarak ini dibedakan agar evaluasi algoritma tidak tercampur dengan perhitungan tarif. Perlu dicatat bahwa urutan CIH yang terbaik untuk rute tertutup belum tentu menghasilkan rute terbuka terpendek setelah ruas kembali ke depot dihilangkan; kondisi tersebut diperlakukan sebagai keterbatasan pendekatan pada penelitian ini.

$$H = r \cdot D \tag{4}$$

$$K = p \cdot H, \quad F = (1 - p) \cdot H \tag{5}$$

Time Window Batching diterapkan pada pesanan yang masuk melalui menu *Batch Order*. Pesanan pertama membuka satu pool dengan jendela waktu tetap selama 15 menit; pesanan berikutnya dapat masuk ke pool tersebut selama jendela masih aktif, tanpa memperpanjang waktu penutupannya. Laravel Scheduler memeriksa pool setiap menit dan memproses pool yang jendela waktunya telah berakhir. Pembentukan kelompok dilakukan secara *greedy* berdasarkan urutan kedatangan. Pesanan pertama yang belum dikelompokkan dipilih sebagai jangkar, kemudian pesanan lain dimasukkan ke kelompok yang sama apabila jarak *Haversine* dari titik tujuan pesanan tersebut ke titik tujuan jangkar tidak lebih dari 2 km. Batas 2 km ini berlaku terhadap jangkar, bukan terhadap setiap pasangan anggota; karena itu, dua anggota dalam satu kelompok dapat berjarak hingga sekitar 4 km satu sama lain. Setelah kelompok terbentuk, setiap kelompok diproses menggunakan CIH. Tarif rute batch dibagi rata kepada seluruh pesanan dalam kelompok (tarif per pesanan = tarif kelompok dibagi jumlah pesanan). Pesanan yang tidak memiliki pasangan tetap diproses sebagai kelompok satu pesanan dengan tarif individual. Mekanisme ini memberi peluang pengurangan tarif, dengan konsekuensi pelanggan menunggu hingga jendela waktu 15 menit selesai sebelum kelompok diproses.

Evaluasi dilakukan melalui simulasi algoritma dan batching menggunakan program Python dengan parameter dan aturan yang sama seperti implementasi aplikasi. Data uji terdiri atas 21 lokasi landmark di Surabaya yang koordinatnya diperoleh melalui Nominatim, dengan depot pada koordinat (−7,2575; 112,7521). Lima skenario optimasi menggunakan 5, 7, 10, 12, dan 15 titik tujuan yang dipilih secara acak dari 21 lokasi dengan seed 42. Urutan hasil pengambilan acak diperlakukan sebagai urutan input pesanan. Kinerja CIH dibandingkan dengan urutan input dan *Nearest Neighbor* (NN). Pada NN, rute dimulai dari depot dan setiap langkah memilih titik tujuan yang belum dikunjungi dengan jarak *Haversine* terkecil dari posisi saat ini, hingga seluruh titik dikunjungi, kemudian rute ditutup dengan kembali ke depot. Kedua algoritma menggunakan depot, himpunan titik tujuan, dan matriks jarak yang sama; seluruh jarak perbandingan dihitung sebagai rute tertutup. *Persentase* penghematan jarak dihitung menggunakan Persamaan (6). Pengujian batching menggunakan tiga skenario kelompok yang terdiri atas 2, 3, dan 4 pesanan, dengan membandingkan jumlah tarif individual terhadap tarif rute batch. Pendapatan kurir per kilometer dihitung sebagai bagian tarif kurir dibagi jarak perjalanan yang menjadi dasar evaluasi: untuk pesanan individual digunakan jarak pulang-pergi depot–tujuan (2d), sedangkan untuk batch digunakan panjang rute terbuka kelompok. Oleh karena itu, indikator tersebut menunjukkan pendapatan relatif terhadap jarak tempuh, bukan perubahan pendapatan total kurir. Pengujian fungsional sistem dilakukan dengan metode *black-box* pada level API menggunakan Laravel Feature Test dan pada level antarmuka secara manual. Pengujian mencakup sepuluh skenario; hasilnya hanya digunakan untuk menyatakan kesesuaian fungsi pada skenario yang diuji, bukan untuk menyimpulkan seluruh aspek kualitas, keamanan, atau kinerja sistem.

$$P = (D_{input} - D_{CIH}) / D_{input} \times 100\% \tag{6}$$

3. HASIL DAN ANALISIS

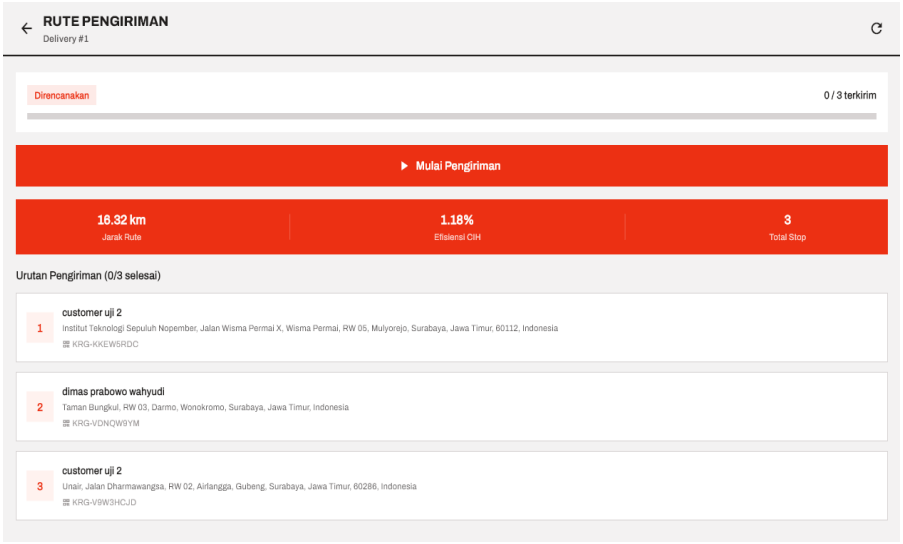
Fitur yang dihasilkan pada setiap bagian sistem dirangkum pada Tabel 1. Pada aplikasi *Flutter*, alamat pada form tambah pelanggan, tambah pengiriman, dan *Batch Order* dicari melalui Nominatim dan koordinatnya

terisi otomatis. Setelah *delivery* dibuat, aplikasi menjalankan CIH dan menampilkan urutan titik beserta ringkasan jarak pada *RouteScreen*. Tampilan pada Gambar 2 merupakan *prototype* antarmuka *RouteScreen* yang memperlihatkan urutan rute hasil CIH, bukan bukti navigasi GPS secara *real-time* atau pengukuran lintasan kendaraan di jalan. Kurir dapat melihat *delivery* hari ini dan mengonfirmasi pengantaran setiap paket dengan *barcode* dan foto. Panel admin web menampilkan peta rute hasil CIH serta rincian harga dasar, diskon, ongkos kirim final, bagian kurir, dan bagian platform.

Tabel 1. Fitur sistem Kita Kargo

Bagian sistem	Pengguna	Fitur utama
Aplikasi Flutter	Admin / operator	Kelola pelanggan, pengiriman, *delivery*, dan *batch order*; pencarian alamat dengan Nominatim; menjalankan CIH; melihat urutan rute
Aplikasi Flutter	Kurir	Daftar *delivery* hari ini; ringkasan tugas; memulai *delivery*; konfirmasi pengantaran dengan *barcode* dan foto
Panel admin *web*	Admin / operator	*Dashboard* statistik; manajemen paket dengan rincian tarif; manajemen karyawan; Rute CIH dengan peta; *Batch Order*
REST API Laravel 11	Semua klien	Autentikasi Sanctum; CIH dan Haversine; pengelompokan *batch* dan perhitungan tarif; pelacakan resi

Gambar 2 memperlihatkan *prototype* tampilan urutan rute hasil CIH pada *RouteScreen.* Pada tahap awal pengembangan, koordinat tujuan dimasukkan secara manual sehingga urutan rute tampak tidak beraturan ketika koordinat tidak tepat mewakili lokasi alamat.



Gambar 2. Tampilan urutan rute hasil CIH pada RouteScreen

Setelah *geocoding Nominatim* diterapkan, pengguna memilih alamat dari daftar saran dan koordinat terisi otomatis. Perubahan tersebut memperbaiki konsistensi input koordinat yang digunakan algoritma. Hasil simulasi perbandingan rute menggunakan koordinat hasil geocoding ditunjukkan pada Tabel 2.

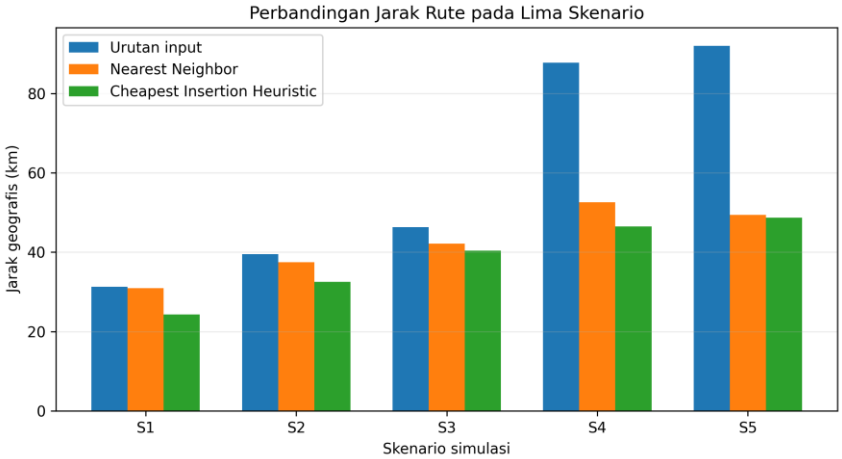
Tabel 2. Perbandingan jarak urutan input, NN, dan rute CIH

Skenario	Jumlah titik	Urutan input (km)	NN (km)	CIH (km)	Penghematan (%)
S1	5	31,32	30,98	24,29	22,43
S2	7	39,49	37,48	32,57	17,52
S3	10	46,32	42,13	40,37	12,85
S4	12	87,74	52,62	46,49	47,02
S5	15	91,99	49,37	48,72	47,04
Rata-rata		59,37	42,52	38,49	29,37

Berdasarkan lima skenario pada Tabel 3, total jarak rute CIH memiliki rata-rata 38,49 km, lebih rendah dibandingkan rata-rata urutan input sebesar 59,37 km dan *Nearest Neighbor* (NN) sebesar 42,52 km. Persentase penghematan CIH terhadap urutan input dihitung pada setiap skenario dengan Persamaan (6), kemudian menghasilkan rata-rata 29,37%. Penghematan terbesar terjadi pada S5 (47,04%) dan S4 (47,02%), sedangkan penghematan terkecil terjadi pada S3 (12,85%). CIH juga menghasilkan jarak lebih pendek daripada NN pada kelima skenario (5 dari 5 atau 100% skenario simulasi).

Persentase 100% ini merupakan proporsi skenario ketika jarak CIH lebih pendek daripada NN, bukan akurasi prediksi atau akurasi model. Rata-rata pengurangan jarak CIH terhadap NN, dihitung relatif terhadap jarak NN pada tiap skenario, adalah sekitar 10,37%. Hasil ini sejalan dengan Rahmah et al. [6] yang memperoleh rute CIH lebih pendek daripada NN. Meskipun demikian, CIH merupakan heuristik dan tidak menjamin

diperolehnya rute global terpendek. Seluruh jarak pada Tabel 3 dihitung menggunakan Haversine, sehingga merupakan jarak geografis dan bukan jarak jalan aktual. Hasil 29,37% karena itu hanya berlaku pada lima skenario simulasi dan dataset lokasi yang digunakan.



Gambar 3. Perbandingan jarak rute pada lima skenario simulasi
Sumber: Hasil simulasi lima skenario menggunakan data lokasi landmark di Surabaya.

Gambar 3 memperlihatkan bahwa jarak CIH lebih rendah daripada urutan input pada seluruh skenario dan lebih rendah daripada NN pada lima skenario yang diuji. Perbedaan paling kecil terhadap NN terlihat pada S5, sehingga keunggulan relatif CIH bergantung pada sebaran dan urutan titik tujuan. Visualisasi ini mendukung hasil simulasi, tetapi tidak mengubah batasan bahwa jarak yang dibandingkan merupakan jarak geografis Haversine.

Tabel 3. Perbandingan tarif individual dan tarif *batch*

Skenario	Jumlah pesanan	Total tarif individual (Rp)	Tarif *batch* (Rp)	Selisih (%)
B1	2	54.049	29.374	45,65
B2	3	44.020	34.092	22,55
B3	4	28.745	16.956	41,01
Rata-rata		42.271	26.807	36,41

Tabel 3 menunjukkan hasil pengujian batching pada tiga skenario. Tarif individual rata-rata mencapai Rp42.271, sedangkan tarif batch rata-rata Rp26.807. Selisih rata-rata sebesar 36,41% menunjukkan potensi pengurangan biaya pelanggan pada skenario simulasi karena satu rute gabungan melayani beberapa tujuan, sehingga jarak depot ke setiap tujuan tidak lagi dihitung sebagai perjalanan individual yang terpisah. Persentase tersebut merupakan perbandingan total tarif pada tiga kelompok simulasi, bukan estimasi penghematan untuk seluruh transaksi operasional.

Pada B2, Royal Plaza berjarak 2,08 km dari Taman Bungkul sebagai jangkar sehingga tidak masuk kelompok, dan diproses sendiri dengan tarif individual Rp18.206. Inilah sebabnya selisih B2 paling kecil. Aturan jangkar juga membuat dua anggota kelompok dapat berjarak hingga sekitar 4 km satu sama lain, karena yang dibatasi hanya jarak setiap anggota ke jangkar. Pembagian rata membuat setiap pesanan dalam satu kelompok membayar tarif yang sama; pada B3 keempat pesanan masing-masing membayar Rp4.239, padahal tarif individualnya berkisar antara Rp5.986 dan Rp9.051. Dalam simulasi ini semua pesanan tetap mendapat potongan, tetapi pada kelompok yang jarak anggotanya sangat berbeda, pembagian proporsional terhadap jarak dapat dipertimbangkan.

Pengujian juga memperlihatkan keterbatasan ketika urutan CIH untuk rute tertutup digunakan sebagai dasar rute terbuka. CIH membentuk rute dari depot, mengunjungi seluruh tujuan, lalu kembali ke depot, sedangkan tarif batch menggunakan rute terbuka tanpa perjalanan kembali. Pada rute tertutup, pembalikan arah putaran tidak mengubah total jarak; tetapi setelah ruas kembali ke depot dihilangkan, arah urutan dapat memengaruhi panjang rute terbuka. Pada kelompok Taman Bungkul dan Kebun Binatang Surabaya, rute terbuka hasil CIH tercatat 5,30 km, lebih panjang daripada urutan input 4,66 km. Temuan ini menunjukkan bahwa optimasi rute tertutup tidak selalu menghasilkan rute terbuka terpendek. Salah satu penyempurnaan yang dapat diuji pada pengembangan berikutnya adalah membandingkan panjang rute terbuka dengan urutan yang dibalik, kemudian memilih yang lebih pendek; langkah tersebut belum diterapkan pada hasil simulasi yang dilaporkan.

Pendapatan kurir per perjalanan dapat lebih rendah pada batch karena beberapa pesanan dilayani dengan satu tarif kelompok, tetapi jarak perjalanan yang menjadi dasar pembanding juga berkurang 27,0% sampai 60,1%. Untuk pesanan individual, pendapatan kurir per kilometer adalah Rp1.275, yaitu 85% dari tarif Rp3.000 dibagi jarak pulang-pergi depot–tujuan. Pada batch, pendapatan kurir per kilometer menjadi Rp1.337 pada B1, Rp1.353 pada B2, dan Rp1.885 pada B3. Dengan demikian, kenaikan 4,9% sampai 47,8% merupakan peningkatan rasio pendapatan terhadap jarak tempuh, bukan bukti bahwa pendapatan total kurir selalu meningkat. Proporsi 85% merupakan konstanta simulasi yang dapat disesuaikan dan tidak dimaksudkan sebagai pernyataan kepatuhan terhadap ketentuan operasional atau regulasi tertentu. Pelanggan memperoleh potensi penghematan tarif dengan konsekuensi waktu tunggu hingga 15 menit. Penelitian ini belum membandingkan beberapa nilai time window, sehingga belum dapat menentukan apakah 15 menit merupakan konfigurasi yang paling optimal.

Hasil pengujian *black-box* ditunjukkan pada Tabel 4. Seluruh 10 skenario menghasilkan keluaran sesuai harapan, sehingga tingkat keberhasilan fungsional pada skenario yang diuji adalah 10/10 × 100% = 100%. Delapan pengujian otomatis dijalankan melalui Laravel Feature Test pada basis data SQLite in-memory, sedangkan skenario 3, 6, dan 10 serta tampilan skenario 9 diperiksa secara manual sesuai pembagian pengujian pada tabel. Angka 100% ini menunjukkan keberhasilan pada skenario fungsional yang diuji, bukan akurasi algoritma, pengujian keamanan menyeluruh, atau validasi seluruh kondisi operasional.

Tabel 4. Hasil pengujian *black-box*

No	Fitur	Skenario	Hasil yang diharapkan	Hasil
1	*Login*	Email dan *password* benar	Masuk ke halaman sesuai peran	Sesuai (API dan *web*)
2	*Login*	*Password* salah	Muncul pesan galat	Sesuai (API)
3	Tambah pelanggan	Memilih alamat dari saran Nominatim	Koordinat terisi otomatis	Sesuai (manual)
4	Tambah pengiriman	Mengisi data lengkap	Data tersimpan	Sesuai (API)
5	Tambah *delivery*	Memilih beberapa pengiriman	CIH berjalan dan rute tersimpan	Sesuai (*web*)
6	Visualisasi rute	Membuka RouteScreen	Urutan titik dan ringkasan jarak tampil	Sesuai (manual)
7	*Batch order*	Dua pesanan berjarak ≤ 2 km dalam 15 menit	Tergabung dalam satu *batch*	Sesuai (API)
8	*Batch order*	Dua pesanan berjarak > 2 km	Tidak digabung	Sesuai (API)
9	Rincian tarif	Memproses *batch* lalu membuka riwayat *batch*	Bagian kurir dan platform tampil	Sesuai (API dan manual)
10	*Dashboard* admin *web*	Membuka *dashboard*	Kartu statistik, paket terbaru, dan daftar kurir tampil	Sesuai (manual)

4. KESIMPULAN

Penelitian ini menghasilkan aplikasi kurir online Kita Kargo berbasis web-mobile yang mengintegrasikan *Cheapest Insertion Heuristic* (CIH), perhitungan tarif berdasarkan jarak rute, dan *Time Window Batching* untuk mendukung pengiriman beberapa pesanan dalam satu perjalanan. Pada lima skenario simulasi, jarak rute CIH rata-rata 29,37% lebih pendek daripada urutan input dan lebih pendek daripada *Nearest Neighbor* pada seluruh skenario yang diuji (5 dari 5 skenario), sedangkan pada tiga skenario batching tarif *batch* rata-rata 36,41% lebih rendah daripada total tarif individual. Pendapatan kurir per kilometer pada skenario *batch* meningkat 4,9%–47,8% dibandingkan pembanding individual, tetapi peningkatan ini merupakan rasio pendapatan terhadap jarak, bukan peningkatan pendapatan total per perjalanan. Seluruh 10 skenario *black-box* menghasilkan keluaran sesuai harapan. Temuan tersebut menunjukkan potensi integrasi optimasi rute dan batching dalam mendukung perhitungan tarif yang lebih efisien dan transparan, tetapi masih terbatas pada simulasi dengan jarak geografis *Haversine*, lima skenario rute, dan tiga skenario *batching*; karena itu, hasilnya belum dapat digeneralisasikan sebagai kinerja operasional pada pesanan riil.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Program Studi Informatika Universitas Teknologi Yogyakarta atas dukungan selama penelitian ini.

REFERENSI

[1] S. E. H. Nasution and M. Batubara, “Analisis dampak potongan komisi platform terhadap pendapatan pengemudi ojek online,” *J. Sci. Soc. Res.*, vol. 9, no. 2, pp. 1855–1860, 2026, doi: 10.54314/jssr.v9i2.6054.

[2] Z. Djunaidi, A. M. Khaliwa, A. Hafia, and N. Putri, “Daily income targets and passenger pressure on safety risky riding behavior among online motorcycle taxi riders in Jakarta, Indonesia,” *Kesmas*, vol. 19, no. 1, pp. 8–17, 2024, doi: 10.21109/kesmas.v19i1.7796.

[3] J. C. Castillo, “Who benefits from surge pricing?,” *Econometrica*, vol. 93, no. 5, pp. 1811–1854,

2025, doi: 10.3982/ECTA19106.

[4] N. Garg and H. Nazerzadeh, "Driver surge pricing," *Manage. Sci.*, vol. 68, no. 5, pp. 3219–3235, 2022, doi: 10.1287/mnsc.2021.4058.

[5] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, "An analysis of several heuristics for the traveling salesman problem," *SIAM J. Comput.*, vol. 6, no. 3, pp. 563–581, 1977, doi: 10.1137/0206041.

[6] S. Rahmah, Yulindawati, M. Jamil, and A. B. Fawait, "Perbandingan algoritma Nearest Neighbor dan Cheapest Insertion Heuristic dalam menyelesaikan pendistribusian barang SPX Express Shopee," *J. Rekayasa Teknol. Inf.*, vol. 10, no. 2, pp. 165–174, 2026, doi: 10.30872/jurti.v10i2.26545.

[7] D. E. Wijayanti, A. Thobirin, and P. W. Prasetyo, "Menentukan rute kendaraan pengangkut sampah Kota Yogyakarta dengan algoritma Cheapest Insertion Heuristic modifikasi route construction," *J. Fourier*, vol. 9, no. 2, pp. 85–95, 2020, doi: 10.14421/fourier.2020.92.85-95.

[8] S. Mas'ud, "Route determination for distribution by using a combination of Branch and Bound algorithm and Cheapest Insertion Heuristic," *ARRUS J. Math. Appl. Sci.*, vol. 4, no. 1, pp. 20–27, 2024, doi: 10.35877/mathscience2636.

[9] M. Y. Aswin, Wamiliana, Fitriani, M. Ansori, and Notiragayu, "Perbandingan Cheapest Insertion Heuristic dan algoritma Christofides untuk menentukan tour pasar tradisional di Kota Bandar Lampung," *J. Pepadun*, vol. 5, no. 2, pp. 182–194, 2024, doi: 10.23960/pepadun.v5i2.228.

[10] D. W. Saputra, "Optimalisasi rute distribusi kurir menggunakan metode Traveling Salesman Problem (studi kasus: JNE Balige)," *G-Tech J. Teknol. Terap.*, vol. 6, no. 2, pp. 159–165, 2022, doi: 10.33379/gtech.v6i2.1577.

[11] S. Gil-Borrás, E. G. Pardo, E. Jiménez, and K. Sörensen, "The time-window strategy in the online order batching problem," *Int. J. Prod. Res.*, vol. 62, no. 12, pp. 4446–4469, 2024, doi: 10.1080/00207543.2023.2263884.

[12] M. Joshi, A. Singh, S. Ranu, A. Bagchi, P. Karia, and P. Kala, "Batching and matching for food delivery in dynamic road networks," in *Proceedings of IEEE 37th International Conference on Data Engineering*, 2021, pp. 2099–2104. doi: 10.1109/ICDE51399.2021.00207.

[13] J. Zhang, X. Wang, and K. Huang, "Integrated on-line scheduling of order batching and delivery under B2C e-commerce," *Comput. Ind. Eng.*, vol. 94, pp. 280–289, 2016, doi: 10.1016/j.cie.2016.02.001.

[14] E. G. Pardo, S. Gil-Borrás, A. Alonso-Ayuso, and A. Duarte, "Order batching problems: Taxonomy and literature review," *Eur. J. Oper. Res.*, vol. 313, no. 1, pp. 1–24, 2024, doi: 10.1016/j.ejor.2023.02.019.

[15] F. Akkerman, P. Dieter, and M. Mes, "Learning dynamic selection and pricing of out-of-home deliveries," *Transp. Sci.*, vol. 59, no. 2, pp. 207–278, 2025, doi: 10.1287/trsc.2023.0434.

[16] R. Schur and K. Winheller, "Optimizing last-mile delivery: A dynamic compensation strategy for occasional drivers," *OR Spectr.*, vol. 47, pp. 1075–1132, 2025, doi: 10.1007/s00291-024-00796-6.