

Available online at: <https://journal.lppmunindra.ac.id/index.php/JOTI>

Jurnal Optimasi Teknik Industri

| ISSN (Print) 2656-3789 | ISSN (Online) 2657-0181 |



Penerapan Algoritma *Simulated Annealing* dan *Large Neighborhood Search* pada *Vehicle Routing Problem with Simultaneous Pickup and Delivery* Di PT Pos Indonesia Yogyakarta

Nazhifa Rahmi Susilo¹, Lathiiyah Thawafani², Muhammad Achirudin Buchari³, Bella Renata Valencia⁴, Achmad Pratama Rifai^{5*}

^{1, 2, 3, 4, 5}Departemen Teknik Mesin dan Industri, Universitas Gadjah Mada, Yogyakarta, Indonesia

*Corresponding author: achmad.p.rifai@ugm.ac.id

ARTICLE INFORMATION

Received :
Revised :
Accepted :
Available online :

KATA KUNCI

Large Neighborhood Search
Simultaneous pickup and delivery
Simulated Annealing
Vehicle routing problem

ABSTRAK

PT. Pos Indonesia adalah perusahaan milik negara yang bergerak di bidang penyediaan jasa logistik dan dikelola oleh pemerintah Indonesia. PT. Pos Indonesia mengatur pengirimannya dengan menempatkan Sentral Pengolahan Pos (SPP) di beberapa kota besar seperti ibukota provinsi. SPP bertanggung jawab atas pengelolaan kiriman, termasuk *collecting*, *processing*, *transporting*, hingga *delivery*. SPP Yogyakarta adalah salah satu kantor yang mengatur pengiriman barang yang masuk dan keluar dari Kota Yogyakarta, dan juga melakukan proses *pickup and delivery* dalam menjalankan tugas operasionalnya. Saat ini kegiatan *pickup and delivery* dilakukan secara terpisah yang berdampak pada total jarak transportasi yang panjang, sehingga mengakibatkan tingginya penggunaan bahan bakar dan juga waktu transportasi yang lama. Penelitian ini bertujuan untuk menentukan rute kunjungan dari SPP Yogyakarta ke KCP/agen dengan mempertimbangkan pengambilan dan pengiriman secara bersamaan atau *Vehicle Routing Problem with Simultaneous Pickup and Delivery* (VRSPD). Penentuan rute dilakukan dengan membandingkan dua algoritma metaheuristik, yaitu *Simulated Annealing* (SA) dan *Large Neighborhood Search* (LNS). Hasil penelitian menunjukkan dari 73 KCP/agen yang harus dikunjungi oleh 8 kendaraan, dengan *fixed cost* berupa gaji pengemudi (karyawan) per hari untuk tiap kendaraan sebesar Rp 106.294,9 dan *variable cost* berupa *fuel cost* dengan biaya bahan bakar tiap kilomernya sebesar Rp 869,57, menunjukkan penyelesaian VRSPD pada PT. Pos Indonesia menggunakan algoritma SA menghasilkan total biaya yang lebih rendah dibandingkan dengan algoritma LNS. Dimana total biaya yang dihasilkan oleh algoritma SA adalah sebesar sebesar Rp 1.377.000, sedangkan total biaya dari algoritma LNS sebesar Rp 1.546.400.

I. PENDAHULUAN

Perkembangan internet dan dunia digital telah berkontribusi banyak pada perubahan gaya hidup manusia sekarang, salah satunya pada sektor jual-beli. Interaksi antara penjual dan pembeli secara *online* semakin marak digunakan tanpa mengenal batasan waktu dan tempat. Hal tersebut didukung dengan pertumbuhan pasar *business to consumer e-commerce* di Indonesia yang diramalkan akan mencapai 13,5% pada tahun 2024 ini [1].

Peningkatan penggunaan *online shopping* ini tidak terlepas dari pelayanan ekspedisi yang turut berkontribusi besar dalam pengiriman barang dari *retailer* ke konsumen. Perusahaan ekspedisi dihadapkan pada tuntutan kecepatan waktu pengiriman dan kemurahan biaya pengantaran dari konsumen *online shopping* [2].

Kecepatan dan kemurahan biaya pengiriman akan sangat bergantung pada efisiensi kegiatan distribusi yang dilakukan oleh suatu penyedia jasa ekspedisi. Efisiensi kegiatan distribusi dapat

dilakukan dengan pengoptimalan rute pengiriman, sumber daya, teknologi, volume pengiriman, serta mekanisme pengiriman. Kegiatan distribusi dari penyedia jasa logistik tidak lepas dari proses *pickup and delivery* atau proses mengambil barang dari pelanggan dan mengirimkan barang ke pelanggan. Proses *pickup and delivery* ini juga berpengaruh pada kecepatan waktu pengiriman karena penjadwalan proses *pickup and delivery* yang kurang tepat dapat menambah waktu transit barang, sehingga meningkatkan potensi keterlambatan pengantaran serta risiko kerusakan dan kehilangan barang. Selain penjadwalan, pemilihan rute kunjungan dalam melakukan proses *pickup and delivery* juga harus dikelola dengan baik untuk mengefisienkan alokasi sumber daya serta menghemat biaya dan waktu pengiriman.

Di Indonesia sendiri terdapat beberapa penyedia jasa logistik, seperti JNE, J&T, Sicepat, Tiki, DHL, dan bahkan perusahaan BUMN, PT. Pos Indonesia. Nama-nama penyedia jasa logistik swasta seperti JNE, J&T, dan Sicepat telah banyak dikenal dan sering digunakan dalam pengiriman *online shopping* dari berbagai macam *marketplace*, seperti Shopee, Tokopedia, Lazada, Zalora, dan lainnya. Berbeda dengan PT. Pos Indonesia yang masih dikenal dan dilabeli sebagai penyedia jasa pengiriman surat, meskipun PT. Pos Indonesia juga melakukan pengiriman barang di berbagai kota di Indonesia [3].

PT. Pos Indonesia merupakan BUMN di bidang penyedia jasa logistik yang dikelola oleh pemerintah Indonesia. Jangkauan pelayanan PT. Pos Indonesia juga cukup luas dengan 58.700 kantor pelayanan yang menjangkau 100% kota/kabupaten, hampir 100% kecamatan, 42% kelurahan/desa, dan 940 lokasi terpencil di Indonesia [4]. PT. Pos Indonesia sendiri mengelola pengirimannya dengan menempatkan Sentral Pengolahan Pos (SPP) di beberapa wilayah ibukota provinsi, seperti di Medan, Palembang, Jakarta, Bandung, Semarang, Yogyakarta, Surabaya, Makassar. SPP ini bertugas untuk melakukan pengelolaan kiriman, mulai dari *collecting, processing, transporting*, hingga *delivery*. SPP Yogyakarta merupakan salah satu kantor yang mengelola pengiriman barang yang masuk maupun keluar dari Kota Yogyakarta. Lokasi SPP Yogyakarta berada di Jalan Plemburan, Ngaglik, Sleman dan satu area dengan Kantor Cabang Pembantu (KCP) Plemburan. SPP Yogyakarta ini juga melakukan proses *pickup and delivery* dalam menjalankan tugas operasionalnya.

Proses *pickup and delivery* yang dilakukan SPP Yogyakarta ini bertujuan untuk mengirimkan barang dari SPP ke Kantor Cabang Pembantu (KCP) dan mengumpulkan barang dari KCP ke SPP untuk selanjutnya dikirim ke kantor tujuan. Mekanisme proses *pickup and delivery* di SPP Yogyakarta masih

dilakukan secara terpisah, yaitu dengan mendahulukan proses *delivery* ke KCP pada waktu pagi hingga siang hari. Selanjutnya, proses *pickup* ke seluruh KCP pada sore hingga malam hari. Ruang lingkup pelayanan *pickup and delivery* dari SPP Yogyakarta mencakup wilayah Kota Yogyakarta dan Kabupaten Sleman.

Praktik *pickup and delivery* secara terpisah yang dilakukan oleh SPP Yogyakarta saat ini berdampak pada total jarak transportasi yang panjang, sehingga bahan bakar yang digunakan juga semakin besar. Dampak lain dari proses *pickup and delivery* yang terpisah ini juga menyebabkan pengiriman barang akan memakan waktu yang lebih lama lagi, sebagaimana penelitian dari Wibowo (2023) yang menyatakan bahwa proses pengiriman barang dari kantor pos sering mengalami keterlambatan [5]. Selain itu, durasi pengiriman yang panjang ini juga akan menyebabkan biaya pengiriman juga meningkat.

Penelitian ini bertujuan untuk menentukan rute kunjungan dari SPP Yogyakarta menuju KCP dengan mempertimbangkan *pickup and delivery* secara bersamaan atau *Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD)*. Proses penentuan rute dilakukan dengan bantuan algoritma metaheuristik, yaitu Simulated Annealing (SA) dan Large Neighborhood Search (LNS) mengingat skala *problem* yang relatif besar sehingga memerlukan waktu komputasi yang lebih lama jika dilakukan menggunakan metode *exact*. Dalam kasus ini, kapasitas kendaraan distribusi diasumsikan sama (homogen) dan kemacetan serta kondisi lalu lintas tidak dipertimbangkan. Dengan adanya optimasi rute *pick up and delivery* secara bersamaan ini diharapkan dapat mempersingkat waktu, jarak, dan juga menurunkan biaya transportasi dari SPP Yogyakarta.

II. METODE

1. Objek Penelitian

Objek penelitian berfokus pada penentuan rute optimal untuk meminimasi total cost pada proses pengiriman dan pengangkutan barang dari dan kembali ke SPP Yogyakarta atau Kantor Pos Cabang Pembantu (KCP) Plemburan menuju ke 73 mitra SPP Yogyakarta yakni agen dan Kantor Cabang Pembantu (KCP) yang tersebar di wilayah Kota Yogyakarta dan Kabupaten Sleman. Untuk kelancaran distribusi, KCU mengoperasikan sebanyak 8 unit kendaraan GrandMax dengan kapasitas 1000 kg.

2. Simulated Annealing

Metode SA adalah metode metaheuristik optimasi global acak diusulkan oleh [6] atas dasar kesamaan antara proses *annealing* materi padat dan proses

umum untuk menyelesaikan masalah optimasi kombinasional. Kekhasan dari algoritma SA ini adalah memodifikasi solusi yang dibangkitkan di awal dengan 3 (tiga) cara yakni menukar (*swap*), memasukkan (*insert*) dan membalik (*flip*) node terpilih serta penerapan kriteria metropolis untuk probabilitas penerimaan solusi modifikasi meskipun solusi tersebut memberikan nilai fungsi tujuan yang buruk. Secara umum urutan logis algoritma SA sebagai berikut:

1. Input: T_0, T_{final}, α
 T_0 = temperatur awal, T_{final} = temperatur akhir, α = cooling rate
2. Membangkitkan solusi awal, X
3. Menghitung nilai fungsi tujuan dari solusi awal $f(X)$
4. Inisialisasi temperatur saat ini $T = T_0$
5. *While* $T > T_{final}$
6. Menampilkan solusi yang telah dimodifikasi $X_{new} \leftarrow X$
7. Menghitung nilai dari fungsi tujuan dari solusi baru $f(X_{new})$
8. Menghitung $\Delta = f(X_{new}) - f(X)$
9. Menerapkan Kriteria Metropolis untuk penerimaan solusi baru
 If $\Delta < 0$
 $f(X) = f(X_{new})$
 $X = X_{new}$
 Else Generate a random variable $R = U(0,1)$
 If $R < \exp(-\Delta/T)$
 $f(X) = f(X_{new})$
 $X = X_{new}$
 Else
 $f(X) = f(X)$
 $X = X$
10. Menyesuaikan temperatur saat ini $T = \alpha T$
11. End *while*
12. Output: Solusi terbaik (X) dan nilai tujuan minimum $f(X)$

3. Large Neighborhood Search

Metode LNS merupakan metode metaheuristik berbasis pelacakan yang diusulkan oleh [7]. LNS sendiri adalah kelas dari metode heuristik *Very Large Neighborhood Search (VLNS)*. Dalam LNS, solusi yang dibangkitkan di awal secara bertahap akan ditingkatkan dengan dihancurkan oleh *destroy operator* dan diperbaiki oleh *repair operator* kemudian dilakukan penyesuaian dengan pencarian lokal secara bergantian. Secara umum algoritma LNS dapat diurutkan sebagai berikut:

1. Input: T, D
 T = Jumlah iterasi, D = Tingkat penghancuran

2. Membangkitkan solusi awal, X
3. Menghitung nilai fungsi tujuan dari solusi awal $f(X)$
4. *While* $t > T$
5. Menampilkan operasi *destroy* $DX \leftarrow d(X)$
6. Menampilkan operasi *repair* $X_{new} \leftarrow r(DX)$
7. Menampilkan penyesuaian terhadap solusi baru $f(X_{new})$
8. Menghitung nilai fungsi tujuan dari solusi yang baru $f(X_{new})$
 jika $f(X_{new}) < f(X)$ sehingga $f(X) = f(X_{new}), X = X_{new}$
9. End *while*
10. Output: Solusi terbaik (X) dan nilai tujuan minimum $f(X)$

III. HASIL DAN PEMBAHASAN

Data yang digunakan dalam penelitian ini adalah matriks jarak dan data *pickup load* dan *delivery load* dari Kantor Pos Cabang Pembantu (KCP) Plemburan yang mana juga merupakan lokasi dari SPP Yogyakarta menuju ke 73 titik lokasi Agen atau Kantor Cabang Pembantu (KCP). Penelitian ini merupakan permasalahan rute dari *Vehicle Routing Problem with Simultaneous Delivery and Pickup (VRPSPD)*, dimana merupakan kasus permintaan gabungan antara operasi pengiriman dan penjemputan barang yang dilakukan secara bersamaan pada satu kali pemberhentian kendaraan [8]. Model matematika untuk kasus VRPSPD adalah sebagai berikut:

Set:

- N = set *customers*, $\{1, 2, 3, \dots, n\}$
- N_0 = set *customers* dan depot, $N_0 = N \cup \{0\}$
- V = set kendaraan, $\{1, 2, \dots, m\}$

Parameter:

- Q_k = kapasitas kendaraan $k \in V$
- F_k = biaya tetap kendaraan $k \in V$
- v_k = biaya variabel kendaraan $k \in V$
- c_{ij} = jarak antara titik $i \in N_0$ dan $j \in N_0$
- d_j = *delivery load customer* $j \in N$
- p_j = *pickup load customer* $j \in N$

Variabel keputusan:

- $x_{ijk} \begin{cases} 1, & \text{jika busur } (i, j) \text{ dilintasi kendaraan } k \\ 0, & \text{lainnya} \end{cases}$
- y_{ijk} = total *pickup load* dari kendaraan k
- z_{ijk} = total *delivery load* dari kendaraan k

Fungsi Tujuan:

$$\text{Min } Z = \sum_{k \in V} \sum_{i \in N_0} \sum_{j \in N_0} F_k x_{ijk} + \sum_{k \in V} \sum_{i \in N_0} \sum_{j \in N_0} v_k x_{ijk} c_{ij} \quad (1)$$

Batasan:

$$\sum_{k \in V} \sum_{i \in N_0} x_{ijk} = 1, \forall j \in N \quad (2)$$

$$\sum_{i \in N_0} x_{pjk} - \sum_{j \in N_0} x_{pjk} = 0, \forall p \in N_0, \forall k \in N \quad (3)$$

$$\sum_{j \in N_0} x_{ojk} \leq 1, \forall k \in V \quad (4)$$

$$y_{oik} = 0, \forall i \in N, \forall k \in V \quad (5)$$

$$z_{oik} = 0, \forall i \in N, \forall k \in V \quad (6)$$

$$\sum_{i \in N_0} \sum_{k \in V} y_{jik} + \sum_{i \in N_0} \sum_{k \in V} y_{ijk} = p_j, \forall j \in N \quad (7)$$

$$\sum_{i \in N_0} \sum_{k \in V} z_{ijk} + \sum_{i \in N_0} \sum_{k \in V} z_{jik} = d_j, \forall j \in N \quad (8)$$

$$\sum_{i \in N} \sum_{k \in V} y_{iok} = \sum_{i \in N} p_i \quad (9)$$

$$\sum_{i \in N} \sum_{k \in V} z_{iok} = \sum_{i \in N} d_i \quad (10)$$

$$y_{ijk} + z_{ijk} \leq Q_k x_{ijk}, \forall i \in N_0, \forall j \in N_0, \forall k \in V \quad (11)$$

$$x_{ijk} \in \{0,1\}, \forall i \in N_0, \forall j \in N_0, \forall k \in V \quad (12)$$

$$y_{ijk} \geq 0, \forall i \in N_0, \forall j \in N_0, \forall k \in V \quad (13)$$

$$z_{ijk} \geq 0, \forall i \in N_0, \forall j \in N_0, \forall k \in V \quad (14)$$

Fungsi tujuan dari model ini ditunjukkan pada persamaan (1) yakni meminimalkan total biaya tetap dan variabel kendaraan. Batasan pada persamaan (2) menjamin bahwa setiap node hanya dikunjungi tepat satu kali. Batasan pada persamaan (3) menjamin bahwa setiap node akan dikunjungi dan ditinggalkan oleh kendaraan yang sama. Batasan pada persamaan (4) memastikan bahwa setiap kendaraan dapat digunakan untuk maksimal satu rute. Muatan yang diangkut saat memulai rute dan jumlah muatan yang harus diantarkan saat kembali ke titik asal masing-masing diset ke nol pada persamaan batasan (5) dan (6). Persamaan pada batasan (7) dan (8) adalah persamaan aliran untuk pengambilan dan muatan pengiriman. Batasan yang diberikan pada persamaan (9) dan (10) masing-masing menjamin bahwa jumlah arus masuk ke titik asal adalah sama dengan total permintaan penjemputan, dan jumlah arus keluar dari titik asal sama dengan total permintaan pengiriman. Batasan kapasitas ditunjukkan pada persamaan (11) dan batasan pada persamaan (12)-(14) mendefinisikan sifat dari variabel keputusan.

1. Penyelesaian Menggunakan Algoritma Simulated Annealing

Metode *metaheuristics* pertama yang digunakan yaitu SA merupakan salah satu metode yang efektif untuk menemukan solusi mendekati optimal untuk masalah rute kendaraan. Algoritma ini menggunakan prinsip penurunan suhu yang terinspirasi dari proses *annealing* untuk eksplorasi ruang solusi secara ekstensif dan menghindari terjebak pada solusi lokal. Algoritma SA menggunakan berbagai parameter yang digunakan untuk memastikan proses optimasi dapat berjalan efisien.

a. Inisialisasi Parameter

Parameter utama yang digunakan dalam Algoritma SA dapat dilihat pada Tabel 1. berikut.

Tabel 1. Parameter Algoritma SA

Parameter	Nilai
Initial Temperature (T0)	5000
Final Temperature (TF)	0,001
Cooling rate (Alpha)	0,99
Jumlah Nodes	73
Jumlah Kendaraan	8
Fixed cost/vehicle (Rp)	106294.9
Fuel cost/km (Rp)	869,57

Berdasarkan Tabel 1. diketahui bahwa *fixed cost/vehicle* adalah Rp Rp 106.294,9 yang diperoleh dari Upah Minimum Kabupaten dan Kota (UMK) Provinsi DIY sebesar Rp 2.125.898 yang dibagi 20 hari kerja dalam sebulan [9]. Sedangkan nilai *variable cost* adalah harga bahan bakar minyak yang dihitung tiap kilometer dengan mempertimbangkan standar konsumsi bahan bakar minyak untuk kendaraan GrandMax, yang akhirnya diperoleh nilai sebesar Rp 869,57/km.

b. Solution Modification

Solution Modification pada SA kali ini menggunakan tiga *operator*. Penentuan operator berdasarkan nilai *random* yang dihasilkan oleh *software* Matlab R2024a. Berikut adalah *operator* yang digunakan di penelitian ini yang menggunakan algoritma SA:

1) Swap Operator

Langkah ini secara acak memilih dua posisi dan kemudian menukar node di kedua posisi tersebut untuk mendapatkan solusi baru [10]. Pada penelitian ini, keputusan penggunaan *swap operator* terjadi apabila bilangan *random* yang dibangkitkan ≤ 0.33 . Ilustrasi *swap operator* ditunjukkan pada Gambar 1.



Gambar 1. Ilustrasi Swap Operator [10]

2) Insert Operator

Operator ini secara acak memilih dua posisi dan menyisipkan *node* dari posisi asal sebelum posisi destinasi untuk mendapatkan solusi baru [10]. *Insert operator* akan digunakan apabila nilai bilangan *andom* ≤ 0.67 . Ilustrasi *insert operator* ditunjukkan pada Gambar 2.



Gambar 2. Ilustrasi Insert Operator [10]

3) Reverse Operator

Reverse operator secara acak memilih dua posisi dan kemudian membalik urutan di antara keduanya untuk mendapatkan solusi baru [10]. Penelitian ini akan menggunakan *reverse operator* apabila nilai *random* > 0.67 . Ilustrasi *reverse operator* ditunjukkan pada Gambar 3.



Gambar 3. Ilustrasi Reverse Operator [10]

c. Hasil Running Algoritma SA

Algoritma SA dijalankan menggunakan software Matlab R2024a dengan parameter yang telah ditentukan diatas, hasil *running* menunjukkan *output* solusi terbaik rute *pickup* dan *delivery* dapat dilihat pada Tabel 2. berikut dan Tabel 3. menunjukkan lokasi-lokasi kunjungan dari *output* rute yang diperoleh menggunakan Matlab.

Tabel 2. Output Solusi Terbaik dengan Algoritma SA pada Matlab

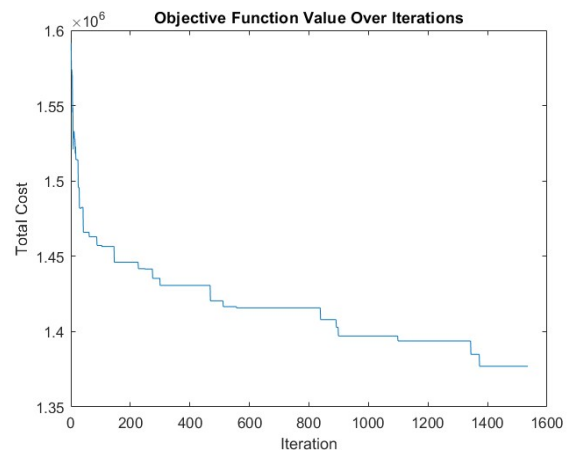
Kendaraan	Rute
Kendaraan 1	[73, 49, 39, 46, 11, 16, 9, 64, 21]
Kendaraan 2	[33, 71, 70, 67, 58, 8, 50, 6, 48]
Kendaraan 3	[52, 7, 51, 62, 57, 41, 10, 36, 40]
Kendaraan 4	[3, 32, 56, 27, 42, 65, 19, 68, 35]
Kendaraan 5	[63, 12, 2, 15, 59, 5, 66, 45, 4]
Kendaraan 6	[30, 69, 47, 44, 34, 29, 61, 28, 17]
Kendaraan 7	[22, 53, 43, 54, 72, 13, 31, 14, 1]
Kendaraan 8	[38, 55, 25, 20, 26, 37, 24, 23, 18, 60]

Tabel 3. Hasil Rute Terbaik dengan Algoritma SA

Kendaraan	Rute
Kendaraan 1	Agen Yudha – Agen Kuncen – Agen Concat – Agen Katamso – KCP Danurejan – KCP Kaliurang – KCP Cangkringan – Agen Tahunan – KCP Minggir
Kendaraan 2	KCP Wirobrajan – Agen Wirogunan – Agen Umbulmartani – Agen Tegaltirto – Agen Nusa Indah – KCP Berbah – Agen Makmur – KCP Babarsari – Agen Kronggahan
Kendaraan 3	Agen Medari – KCP Banyuraden – Agen Mancasan – Agen Sinduadi – Agen Ngasem – Agen Jambusari – KCP Condongcatur – Agen Besi Sedan – Agen Gito Gati
Kendaraan 4	KCP Prambanan – KCP Turi – Agen Mlangi – KCP Ngemplak – Agen Jlagran – Agen Tajem – KCP Mantrijeron – Agen Terban – Agen Besi Jangkang
Kendaraan 5	Agen Suryowijayan – KCP Gamping – KCP Sleman – KCP Kalasan – Agen Paingan – KCU Yogyakarta – Agen Tegal Gendu – Agen Karang Waru – KCP Pakem
Kendaraan 6	KCP Tegalrejo – Agen Ukrim – Agen KOPMA UGM – Agen Kadisoka – Agen Agro – KCP Seyegan – Agen

Kendaraan	Rute
	Rajawali – KCP Pingit – KCP Karangmalang
Kendaraan 7	KCP Minomartani – Agen Melfindo Seturan – Agen Jogoragan – Agen Menukan – Agen Wonorejo – KCP Gondolayu – KCP Tempel – KCP Gondomanan – KCP Godean
Kendaraan 8	Agen Cebongan – Agen Minggir – KCP Muja Muju – KCP Mergangsan – KCP Ngaglik – Agen Bulaksumur – KCP Moyudan – KCP Mlati – KCP Kotagede – Agen Pringwulung

Pemilihan solusi terbaik SA pada Matlab berdasarkan hasil melakukan *running coding* SA sebanyak 5 kali, kemudian memilih solusi dengan *total cost* paling kecil (gambar grafik hasil *running* SA sebanyak 5 kali dilampirkan). Hasil solusi terbaik dengan Algoritma SA diatas, menunjukkan pembagian lokasi *nodes* KCP atau agen ke dalam rute yang dilalui oleh masing-masing kendaraan. Setiap kendaraan memiliki rute yang dirancang untuk memenuhi kapasitas maksimal kendaraan tanpa melanggar batasan beban. Algoritma SA berhasil mengoptimalkan rute *pickup* dan *delivery* untuk 73 *nodes* dengan menggunakan 8 kendaraan, dengan menghasilkan total biaya per hari yang dikeluarkan sebesar Rp 1.377.000. Parameter dan implementasi algoritma SA dalam software Matlab menunjukkan efisiensi dalam pemecahan masalah penentuan rute optimal yang akan dilalui kendaraan, memberikan solusi yang memenuhi semua kendala dan menghasilkan total biaya minimal.



Gambar 4. Grafik Solusi Terbaik dengan Algoritma SA

Berdasarkan hasil *running* menggunakan Algoritma SA, seiring berjalannya waktu pada grafik terlihat adanya penurunan total biaya yang cukup signifikan. Hal ini menunjukkan bahwa Algoritma SA berhasil menemukan solusi yang lebih baik dengan mengurangi total biaya seiring waktu. Pada sekitar iterasi ke-1400, grafik menunjukkan bahwa

total biaya mulai *convergence*, dengan sedikit atau perubahan yang signifikan dalam nilai fungsi objektif. Ini menunjukkan bahwa algoritma telah menemukan solusi yang stabil dan mendekati optimal (*near optimal*). Pada iterasi awal, algoritma lebih eksploratif mencari solusi dalam ruang solusi yang lebih luas, seiring dengan penurunan suhu, algoritma menjadi lebih eksploitatif, berfokus pada perbaikan kecil untuk mendekati solusi optimal.

2. Penyelesaian Menggunakan Algoritma Large Neighborhood Search

Metode *metaheuristics* kedua yang dijadikan pembandingan solusi terbaik untuk pengoptimalan rute permasalahan VRPSPD, yaitu *Large Neighborhood Search* (LNS) merupakan algoritma menggunakan prinsip eksplorasi ruang solusi dalam skala besar dengan cara modifikasi bagian signifikan dari solusi yang ada untuk memperbaiki kualitas solusi. Pada penyelesaian menggunakan algoritma LNS.

a. Inisialisasi parameter

Parameter utama yang digunakan dalam Algoritma LNS ditunjukkan pada Tabel 4. berikut.

Tabel 4. Parameter Algoritma LNS

Parameter	Nilai
Total number of Iterations (T)	1000
Degree of Destruction range (DoD)	[0,1 0,5]
Jumlah Nodes	73
Jumlah Kendaraan	8
Fixed cost/vehicle (Rp)	106294,9
Fuel cost/km (Rp)	869,57

b. Destroy and Repair

Perbedaan SA dan LNS adalah ketika proses modifikasi solusi. Jika SA menggunakan satu tipe modifikasi saja, LNS menggunakan dua tipe modifikasi, yaitu *destroy* dan *repair*.

1) Destroy Operator

Destroy operator yang digunakan pada penelitian ini hanya menggunakan satu *operator*, yaitu *random removal*. Memilih komponen solusi secara *random* akan membantu proses pencarian menjadi lebih beragam, sehingga dapat keluar dari local optimum, dan dapat memperluas *neighborhood* [11]. Ilustrasi *random removal* untuk *destroy*, ditunjukkan oleh Gambar 5.



Gambar 5. Ilustrasi *Random Removal* pada *Destroy Operator* [12]

2) Repair Operation

Sama halnya dengan *destroy operator*, pada *repair operator* menggunakan satu *operator* saja, yaitu *random insertion*. *Operator random insertion* melakukan *repair* di mana satu atau lebih node yang

sebelumnya dilakukan *destroy*, dimasukkan kembali ke dalam rute yang ada secara acak [13]. Ilustrasi *random insertion* ditunjukkan pada Gambar 6.



Gambar 6. Ilustrasi *Random Insertion* pada *Repair Operator* [12]

c. Hasil Running Algoritma LNS

Algoritma LNS dijalankan menggunakan *software* Matlab R2024a dengan parameter yang telah ditentukan diatas, hasil *running* menunjukkan *output* solusi terbaik rute *pickup* dan *delivery* dapat dilihat pada Tabel 5. berikut dan lokasi kunjungan kendaraan ditunjukkan pada Tabel 6.

Tabel 5. *Output* Solusi Terbaik dengan Algoritma LNS pada Matlab

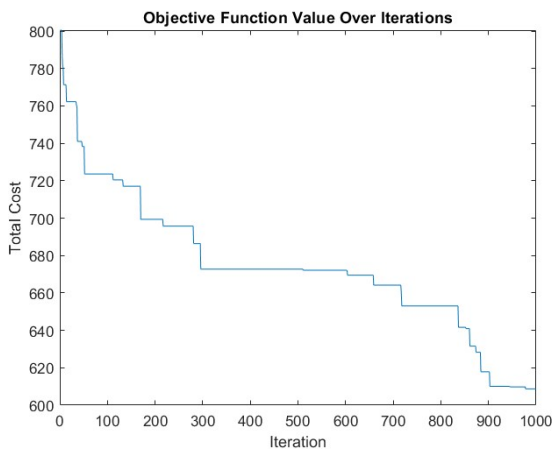
Kendaraan	Rute
Kendaraan 1	[37, 20, 71, 12, 47, 30, 29, 21, 13]
Kendaraan 2	[4, 31, 68, 65, 6, 34, 38, 17, 42]
Kendaraan 3	[16, 11, 25, 19, 46, 69, 52, 45, 49]
Kendaraan 4	[50, 67, 15, 9, 70, 58, 5, 61, 56]
Kendaraan 5	[28, 41, 22, 66, 18, 8, 64, 43, 33]
Kendaraan 6	[32, 27, 62, 51, 3, 59, 14, 72, 35]
Kendaraan 7	[36, 23, 60, 53, 26, 10, 40, 39, 44]
Kendaraan 8	[7, 73, 63, 54, 1, 24, 55, 2, 57, 48]

Tabel 6. Hasil Rute Terbaik dengan Algoritma SA

Kendaraan	Rute
Kendaraan 1	Agen Bulaksumur – KCP Mergangsan – Agen Wirogunan – KCP Gamping – Agen KOPMA UGM – KCP Tegalrejo – KCP Seyegan – KCP Minggir – KCP Gondolayu
Kendaraan 2	KCP Pakem – KCP Tempel – Agen Terban – Agen Tajem – KCP Babarsari – Agen Agro – Agen Cebongan – KCP Karangmalang – Agen Jlagran – KCP Kaliurang – KCP Danurejan – KCP Muja Muju – KCP Mantrijeron
Kendaraan 3	– Agen Katamso – Agen Ukrim – Agen Medari – Agen Karang Waru – Agen Kuncen
Kendaraan 4	– Agen Makmur – Agen Tegalirto – KCP Kalasan – KCP Cangkringan – Agen Umbulmartani – Agen Nusa Indah – KCU Yogyakarta – Agen Rajawali – Agen Mlangi – KCP Pingit – Agen Jambusari – KCP Minomartani – Agen Tegal Gendu – KCP Kotagede – KCP Berbah – Agen Tahunan – Agen Jogoragan – KCP Wirobrajan
Kendaraan 5	KCP Turi – KCP Ngemplak – Agen Sinduadi – Agen Mancasan – KCP Prambanan – Agen Paingan – KCP

Kendaraan	Rute
Kendaraan 7	Gondomanan – Agen Wonorejo – Agen Besi Jangkang
	Agen Besi Sedan – KCP Mlati – Agen Pringwulung – Agen Melfindo Seturan – KCP Ngaglik – KCP Condongcatur – Agen Gito Gati – Agen Concat – Agen Kadisoka
Kendaraan 8	KCP Banyuraden – Agen Yudha – Agen Suryowijayan – Agen Menukan – KCP Godean – KCP Moyudan – Agen Minggir – KCP Sleman – Agen Ngasem – Agen Kronggahan

Solusi Terbaik LNS dipilih melalui pencarian *total cost* paling sedikit dari hasil *running coding* LNS sebanyak 5 kali (gambar grafik hasil *running* LNS sebanyak 5 kali dilampirkan). Algoritma LNS berhasil mengoptimalkan rute *pickup* dan *delivery* untuk 73 *nodes* dengan menggunakan 8 kendaraan, dengan total biaya per hari yang dihitung untuk solusi terbaik tersebut adalah Rp 1.546.400. Solusi yang ditemukan menunjukkan pembagian beban *nodes* yang dioptimalkan sedemikian rupa sehingga setiap kendaraan melayani serangkaian rute menuju ke beberapa *nodes* diatas dengan efisien, dengan mengingat batasan kapasitas kendaraan.



Gambar 7. Grafik Solusi Terbaik dengan Algoritma LNS

Algoritma LNS secara efektif meningkatkan solusi dari 1 iterasi ke iterasi berikutnya, menghasilkan solusi terbaik dengan biaya yang semakin minimal dan optimal seiring berjalannya waktu. Penurunan biaya yang stabil menunjukkan bahwa algoritma LNS berhasil mengoptimalkan rute secara bertahap. Penurunan tajam pada beberapa titik, kemungkinan disebabkan oleh perubahan signifikan dalam solusi selama proses destruksi dan rekonstruksi (*repair*) dalam algoritma LNS

IV. SIMPULAN

Perusahaan logistik milik negara yang dikelola pemerintah, PT Pos Indonesia, mengatur pengiriman barang menggunakan Sentral Pengolahan Pos (SPP) yang ditempatkan di berbagai kota besar di Indonesia. SPP Yogyakarta, salah satunya, bertanggung jawab atas alur barang masuk dan keluar di Yogyakarta, termasuk *pick up and delivery* barang. Saat ini, *pick up and delivery* dilakukan secara terpisah, menyebabkan jarak tempuh yang jauh, penggunaan bahan bakar yang tinggi, dan waktu pengiriman yang lama. Penelitian ini bertujuan untuk mencari rute terbaik kunjungan SPP Yogyakarta ke Kantor Cabang Pembantu (KCP)/agen dengan mempertimbangkan *pick up and delivery* barang secara bersamaan (VRPSPD). Dua algoritma dibandingkan untuk menentukan rute optimal: *Simulated Annealing* (SA) dan *Large Neighborhood Search* (LNS).

Penelitian ini melibatkan 73 KCP/agen yang harus dikunjungi oleh 8 kendaraan. Biaya tetap (*fixed cost*) berupa gaji pengemudi per kendaraan dalam satu harinya sebesar Rp 106.294,9 dan biaya variabel (*variable cost*) berupa biaya bahan bakar per kilometer sebesar Rp 869,57. Dari kedua metode *metaheuristics*, Algoritma SA menghasilkan total biaya yang lebih rendah sebesar Rp 1.377.000 dibandingkan dengan Algoritma LNS sebesar Rp 1.546.400, menunjukkan bahwa metode SA lebih efisien dalam permasalahan VRPSPD dengan fungsi tujuan untuk meminimalkan biaya yang dikeluarkan. Kedua algoritma menunjukkan efisiensi dalam pemecahan masalah penentuan rute, namun metode SA lebih unggul dalam mencapai total biaya yang lebih rendah. Kedua metode berhasil mendistribusikan beban *nodes* secara efisien dengan memperhatikan batasan kapasitas kendaraan, meskipun menggunakan pendekatan dan mekanisme yang berbeda.

REFERENSI

- [1] C. J. Morgan, "Global E-commerce Trends Report," 2021. [Online]. Available: <https://www.jpmorgan.com/solutions/treasury-payments/global-ecommerce-trends-report>
- [2] R. Mangiaracina, A. Perego, A. Seghezzi, and A. Tumino, "Innovative solutions to increase last-mile delivery efficiency in B2C e-commerce: a literature review," *Int. J. Phys. Distrib. Logist. Manag.*, vol. 49, no. 9, pp. 901–920, 2019, doi: 10.1108/IJPDLM-02-2019-0048.
- [3] H. Warsono, D. R. Qurniawati, H. Sitorus, and

- S. M. Fajar S., "Inovasi Layanan Mpospay dan Free Pick Up Pos Indonesia Dalam Peningkatan Kualitas Pelayanan (Studi Pada Kantor Pos Semarang 50000)," *JPSI (Journal Public Sect. Innov.*, vol. 2, no. 2, p. 44, 2018, doi: 10.26740/jpsi.v2n2.p44-50.
- [4] Pos Indonesia, "Kantor Regional," Pos Indonesia. [Online]. Available: <https://www.posindonesia.co.id/id/content/kantor-regional>
- [5] W. A. R. M. Wibowo, "Kualitas Pelayanan Pt . Pos Indonesia KCU Semarang," 2023.
- [6] N. Metropolis, A. W. R. M. N. Rosenbluth, A. Teller, and E. Teller, "Simulated Annealing," *J. Chem. Phys.*, vol. 21, no. 161–162, pp. 1087–1092, 1953.
- [7] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1520, pp. 417–431, 1998, doi: 10.1007/3-540-49481-2_30.
- [8] S. K. Salsabila, H. Mayyani, and P. T. Supriyo, "Penyelesaian VRPSPD Menggunakan Firefly Algorithm (Studi Kasus Distribusi Aqua Galon)," *MILANG J. Math. Its Appl.*, vol. 19, no. 1, pp. 53–67, 2023, doi: 10.29244/milang.19.1.53-67.
- [9] Badan Pusat Statistik Provinsi D.I. Yogyakarta, "Upah Minimum Kabupaten/Upah Minimum Provinsi di DI Yogyakarta (Rupiah/Bulan), 2022-2024," Badan Pusat Statistik Provinsi D.I. Yogyakarta. [Online]. Available: <https://yogyakarta.bps.go.id/indicator/6/272/1/upah-minimum-kabupaten-upah-minimum-provinsi-di-di-yogyakarta.html>
- [10] V. F. Yu, G. Aloina, P. Jodiawan, A. Gunawan, and T. C. Huang, "The vehicle routing problem with simultaneous pickup and delivery and occasional drivers," *Expert Syst. Appl.*, vol. 214, no. October 2022, p. 119118, 2023, doi: 10.1016/j.eswa.2022.119118.
- [11] A. P. Rifai, H. T. Nguyen, and S. Z. M. Dawal, "Multi-objective adaptive large neighborhood search for distributed reentrant permutation flow shop scheduling," *Appl. Soft Comput. J.*, vol. 40, pp. 42–57, 2016, doi: 10.1016/j.asoc.2015.11.034.
- [12] A. P. Rifai, "Large Neighborhood Search (LNS)," *Bahan Ajar Mata Kuliah Metaheuristics*, 2024.
- [13] C. Thomas, R. Kameugne, and P. Schaus, "Insertion sequence variables for hybrid routing and scheduling problems," *Int. Conf. Integr. Constraint Program. Artif. Intell. Oper. Res.*, pp. 457–474, 2020.