Available *online* at : http://jurnal.poltekapp.ac.id/

Jurnal Manajemen Industri dan Logistik

| ISSN (Print) 2622-528X | ISSN (Online) 2598-5795 |

article category : Logistic Management

ANALISA PERBANDINGAN METODE *SIMULATED ANNEALING* DAN *LARGE NEIGHBORHOOD SEARCH* UNTUK MEMECAHKAN MASALAH LOKASI DAN RUTE KENDARAAN DUA ESELON

COMPARISON ANALYSIS BETWEEN SIMULATED ANNEALING AND LARGE NEIGHBORHOOD SEARCH HEURISTICS FOR SOLVING TWO-ECHELON VEHICLE ROUTING PROBLEM

Winarno[1,*)], A. A. N. Perwira Redi[2)]

[1)] Program Studi Teknik Industri, Fakultas Teknik, Universitas Singaperbangsa Karawang, Indonesia
[2)] Program Studi Teknik Logistik, Fakultas Teknologi Industri, Universitas Pertamina, Jakarta, Indonesia

ARTICLE INFORMATION

Article history:

Received: January 17, 2020
Revised 1: February 10, 2020
Accepted: April 09, 2020

Keywords:
Two-Echelon Location Routing Problem
Simulated Annealing
Large Neighborhood Search

A B S T R A C T

Two-echelon location routing problem (2E-LRP) is a problem that considers distribution problem in a two-level / echelon transport system. The first echelon considers trips from a main depot to a set of selected satellite. The second echelon considers routes to serve customers from the selected satellite. This study proposes two metaheuristics algorithms to solve 2E-LRP: Simulated Annealing (SA) and Large Neighborhood Search (LNS) heuristics. The neighborhood / operator moves of both algorithms are modified specifically to solve 2E-LRP. The proposed SA uses swap, insert, and reverse operators. Meanwhile the proposed LNS uses four destructive operator (random route removal, worst removal, route removal, related node removal, not related node removal) and two constructive operator (greedy insertion and modived greedy insertion). Previously known dataset is used to test the performance of the both algorithms. Numerical experiment results show that SA performs better than LNS. The objective function value for SA and LNS are 176.125 and 181.478, respectively. Besides, the average computational time of SA and LNS are 119.02s and 352.17s, respectively.

Kata kunci:
Masalah Lokasi dan Rute Kendaraan Dua Eselon
Simulated Annealing
Large Neighborhood Search

A B S T R A K

Permasalahan penentuan lokasi fasilitas sekaligus rute kendaraan dengan mempertimbangkan sistem transportasi dua eselon juga dikenal dengan *two-echelon location routing problem* (2E-LRP) atau masalah lokasi dan rute kendaraan dua eselon (MLRKDE). Pada eselon pertama keputusan yang perlu diambil adalah penentuan lokasi fasilitas (diistilahkan satelit) dan rute kendaraan dari depo ke lokasi satelit terpilih. Pada eselon kedua dilakukan penentuan rute kendaraan dari satelit ke masing-masing pelanggan mempertimbangan jumlah permintaan dan kapasitas kendaraan. Dalam penelitian ini dikembangkan dua algoritma metaheuristik yaitu *Simulated Annealing* (SA) dan *Large Neighborhood Search* (LNS). *Operator* yang digunakan kedua algoritma tersebut didesain khusus untuk permasalahan MLRKDE. Algoritma SA menggunakan *operator swap, insert,* dan *reverse*. Algoritma LNS menggunakan *operator* perusakan (*random route removal, worst removal, route removal, related node removal,* dan *not related node removal*) dan perbaikan *(greedy insertion* dan *modified greedy insertion)*. *Benchmark* data dari penelitian sebelumnya digunakan untuk menguji performa kedua algoritma tersebut. Hasil eksperimen menunjukkan bahwa performa algoritma SA lebih baik daripada LNS. Rata-rata nilai fungsi objektif dari SA dan LNS adalah 176.125 dan 181.478. Waktu rata-rata komputasi algoritma SA and LNS pada permasalahan ini adalah 119.02 dan 352.17 detik.

Corresponding Author

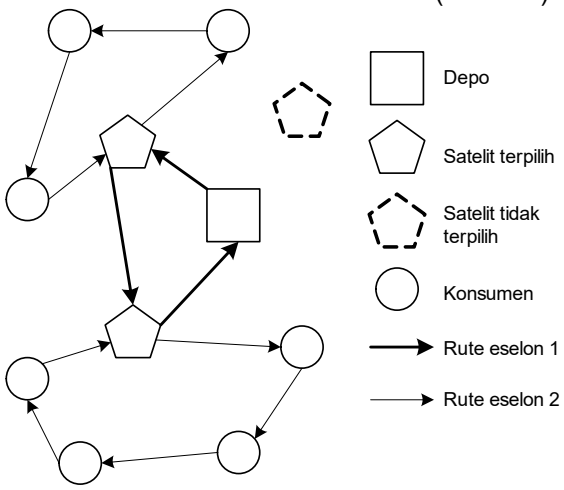
Name : winarno
E-mail: winarno@staff.unsika.ac.id

1. PENDAHULUAN

Masalah lokasi dan rute kendaraan dua eselon (MLRKDE) merupakan suatu model untuk memecahkan masalah sistem distribusi barang pada logistik perkotaan. Barang dikirim dari lokasi sumber ke lokasi tujuan dimana konsumen akhir berada. Untuk mencapai skala ekonomi, barang yang dikirim dari lokasi sumber menggunakan kendaraan dengan kapasitas besar. Namun penggunaan kendaraan besar di lokasi konsumen dilarang karena adanya keterbatasan lebar jalan dan kepadatan lalu lintas. Oleh karena itu, barang-barang tersebut dikirim menggunakan kendaraan yang lebih kecil. Fasilitas antara (satelit) diperlukan untuk memindahkan barang dari kendaraan besar ke kendaraan kecil.

Fungsi tujuan MLRKDE adalah untuk meminimasi total biaya distribusi barang yang terdiri dari biaya kendaraan di kedua eselon, biaya satelit, dan biaya rute kendaraan di kedua eselon. Kendaraan di eselon pertama akan menghubungkan sumber dengan satelit, sedangkan kendaraan di eselon kedua menghubungkan satelit dengan konsumen. Pemilihan lokasi satelit dan penentuan rute kendaraan yang tepat dapat mempercepat respon terhadap kebutuhan konsumen, meningkatkan kualitas pelayanan, dan meningkatkan kepuasan konsumen di dalam sistem logistik perkotaan. Ilustrasi MLRKDE disajikan dalam gambar 1. Dari gambar tersebut mengilustrasikan jaringan pengiriman barang dua eselon yang terdiri dari satu depo (segiempat), tiga satelit potensial (segilima), dan tujuh konsumen (lingkaran). Kendaraan berkapasitas besar berangkat dari depo menuju satelit-satelit terpilih (segilima garis penuh) dan kembali lagi ke depo, sehingga terbentuk rute kendaraan eselon 1 (garis panah tebal). Selanjutnya kendaraan berkapasitas lebih kecil mengangkut barang di satelit menuju konsumen-konsumen yang akan dilayani dan kembali lagi ke satelit, sehingga terbentuk rute kendaraan eselon 2 (garis panah tipis).

MLRKDE merupakan salah satu masalah optimasi kombinatorial yang *NP-hard*. Model tersebut sangat sulit diselesaikan dengan metode pemecahan eksak, oleh karena itu kebanyakan metode heuristik dipakai untuk memperoleh solusi mendekati optimal. Literatur menunjukkan bahwa para peneliti telah memecahkan MLRKDE menggunakan berbagai metode heuristik yang berbeda-beda. Berdasarkan pengetahuan penulis, *simulated annealing* (SA) merupakan salah satu metode heuristik dan telah banyak dipakai para peneliti untuk memecahkan masalah optimasi kombinatorial yang NP-hard [1-4]. Sedangkan hampir dua dekade banyak peneliti juga menggunakan metode heuristik *large neighborhood search* (LNS) untuk memecahkan masalah tersebut khususnya pada MLRKDE dan masalah rute kendaraan dua eselon (MRKDE).



Gambar 1. Ilustrasi Masalah lokasi dan rute kendaraan dua eselon

Suatu masalah optimasi dapat dipecahkan dengan berbagai metode pemecahan yang berbeda. Beberapa peneliti membandingkan kinerja dua metode atau lebih untuk memecahkan masalah tersebut. Misalnya, [5] membandingkan kinerja heuristik *descent*, *simulated annealing* dan *tabu search* untuk menyelesaikan masalah rute kendaraan, sedangkan [6] mengajukan metode *variable neighborhood search* (VNS) dan metode hibrid VNS dan SA untuk memecahkan masalah lokasi dan rute kendaraan pada penanganan limbah padat. Dalam artikel ini, akan dilakukan analisis perbandingan antara metode SA dan LNS untuk memecahkan MLRKDE.

Cuda, et al. [7] mendefinisikan bahwa MRKDE adalah suatu model sistem distribusi barang yang melibatkan keputusan perencanaan tingkat taktis dan strategis, dan penentuan rute kendaraan yang ada di kedua eselon. Di dalam MRKDE, barang terletak di lokasi sumber yang biasa disebut depo akan dikirim ke lokasi tujuan melalui fasilitas antara yang biasa disebut satelit. Biaya pembukaan akan dibebankan ke setiap depo dan satelit. Depo dan satelit yang akan dibuka perlu dipilih dari sekumpulan lokasi depo atau satelit.

MRKDE pertama kali dikaji oleh Jacobsen and Madsen [8]. Peneliti tersebut mengkaji sistem distribusi surat kabar yang melibatkan titik-titik transfer untuk menghubungkan kantor percetakan dan para pengecer. Dewasa ini, setelah hampir empat dekade, beberapa metode heuristik telah diterapkan oleh para peneliti untuk menyelesaikan MRKDE. Metode heuristik seperti tabu search

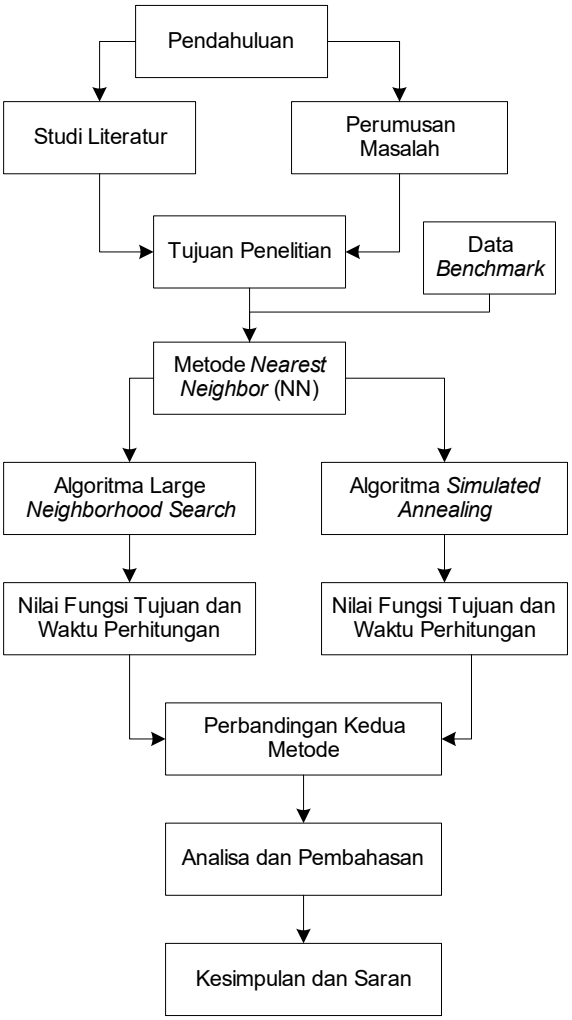
(TS)[9], *variable neighborhood search* (VNS) [10], *greedy randomized adaptive search procedure* (GRASP) dengan learning process dan *path relinking* [11], *multi-start iterated local search* (ILS) dengan *path relinking* [12], dan LNS [13] telah saling melengkapi satu sama lain untuk memecahkan MRKDE secara efisien. Hemmelmayr, et al. [14] juga menerapkan LNS untuk memecahkan MRKDE.

Sementara itu, ada metode heuristik (atau metaheuristik) yang sederhana, tetapi mampu menghasilkan solusi yang cukup bagus dengan waktu perhitungan yang relatif cepat. Di dalam literatur menunjukkan bahwa simulated annealing (SA) adalah metode metaheuristik yang sederhana tetapi mampu memecahkan masalah secara cepat. SA telah banyak diterapkan secara sukses di berbagai masalah optimasi kombinatorial yang cukup kompleks. Misalnya pada masalah penyusunan jadwal sekolah [1], tata letak fasilitas [4, 15], perancangan jaringan distribusi [2], turnamen perjalanan [16], penjadwalan mesin [3] dan rute kendaraan [17]. Lebih lanjut, di dalam Lin, et al. [18], SA digunakan untuk memecahkan masalah rute truk dan *trailer* (MRTT) secara efisien dengan hasil solusi yang bagus. Sebagai catatan bahwa MRTT ini berbeda dengan MLRKDE, namun demikian MRTT merupakan salah satu model masalah rute kendaraan dua eselon [7].

2. METODE PENELITIAN

Dalam artikel ini, penulis mengajukan metode heuristik dua tahap. Tahap awal merupakan pembentukan solusi awal untuk mendapatkan solusi secara cepat. Tahap berikutnya adalah memperbaiki solusi awal menggunakan metode heuristik. Selanjutnya, representasi solusi terdiri dari dua bagian, yaitu solusi eselon satu dan dua. Solusi eselon satu terdiri dari satu depo, sekumpulan satelit dan sejumlah angka nol imitasi (*dummy zeros*). Solusi eselon dua terdiri dari himpunan satelit, himpunan konsumen dan sejumlah *dummy zeros*.

Pada solusi eselon satu, setiap rute mulai dari depo. Selanjutnya satelit ditambahkan satu per satu tanpa melampaui kapasitas kendaraan eselon satu, dari kiri ke kanan ke dalam rute saat ini dan berakhir di depo. Pada solusi eselon dua, setiap rute berawal dari satelit. Konsumen ditambahkan satu per satu tanpa melanggar kapasitas kendaraan eselon kedua, dari kiri ke kanan ke dalam rute dari satelit saat ini, dan berakhir pada satelit. *Dummy zeros* di dalam kedua bagian solusi digunakan untuk menghentikan suatu rute dan memulai rute baru meskipun jumlah muatan kendaraan masih lebih kecil dari kapasitas kendaraan. Oleh karena itu, suatu rute dapat berakhir secara acak menggunakan *dummy zero* di dalam representasi solusi dan algoritma dapat mencari kesempatan lebih besar untuk mendapatkan solusi yang lebih baik.



Gambar 2. *Flow chart* penelitian

Sebelum metode heuristik SA atau LNS diterapkan, inisial solusi dibentuk. Dalam artikel ini, pembentukan solusi awal di kedua eselon menggunakan metode *nearest neighbor* (NN). Di dalam eselon pertama, rute besar berawal dari depo, kemudian semua satelit ditambahkan satu per satu mengikuti aturan metode NN dengan tetap memperhatikan kapasitas kendaraannya dan berakhir di depo. Berikutnya, di eselon kedua, setiap konsumen dimasukkan ke dalam rute mengikuti aturan metode NN tanpa melanggar

kapasitas kendaraannya. Jumlah muatan rute-rute yang dilayani satelit di eselon kedua tidak diijinkan melewati kapasitas satelit. Kemudian, rute besar di eselon pertama diperbaharui rutenya tanpa melanggar kapasitas kendaraan eselon pertama.

Solusi hasil NN selanjutnya diperbaiki dengan metode SA dan LNS. Solusi dari kedua metode tersebut kemudian dibandingkan kinerjanya baik nilai fungsi tujuan maupun waktu perhitungannya. Dalam analisa dan pembahasan, selain perbandingan kinerja kedua metode tersebut, metode SA dan LNS juga dibandingkan dengan metode lain dari literatur. Tahap akhir dari penelitian ini berupa kesimpulan dan saran. Secara visual tahapan penelitian ini disajikan dalam Gambar 2.

2.1. Simulated Annealing untuk MLRKDE

Metode SA telah diperkenalkan oleh Metropolis, et al. [19]. Kirkpatrick, et al. [20] dan Eglese [21] mempopulerkan metode ini dengan melakukan tinjauan SA dalam masalah optimasi kombinatorial. SA telah diterapkan pada masalah-masalah optimasi kombinatorial termasuk masalah lokasi dan rute kendaraan (MLRK) Yu, et al. [22] dan masalah lokasi dan rute kendaraan terbuka (MLRKT) Yu and Lin [23]. Berikutnya, metode heuristik SA dengan representasi solusi khusus untuk MLRKDE juga akan diusulkan dalam penelitian ini.

Metode SA ini memerlukan lima parameter: T_0, T_{max}, I_{iter}, $N_{non\text{-}improving}$, dan α. T_0 merupakan temperatur awal; T_{max} adalah temperatur akhir; I_{iter} menyatakan jumlah iterasi dalam pencarian solusi pada nilai temperatur yang sama; $N_{non\text{-}improving}$ menunjukkan frekuensi penurunan suhu yang diijinkan tanpa perbaikan solusi; dan α adalah koefisien laju penurunan temperatur.

Di dalam metode SA ini, temperatur awal (T_0) ditetapkan sebagai temperatur saat ini (T), nilai fungsi tujuan X dinotasikan $f(X)$, solusi terbaik (X_{best}) saat ini ditetapkan sebagai X. F_{best} adalah nilai fungsi tujuan terbaik yang dicapai oleh SA saat ini dan diinisiasi sebagai $f(X)$. Setiap iterasi di temperatur tertentu melakukan mekanisme pencarian terdekat yang akan menghasilkan solusi baru Y dari solusi X. Ditetapkan Δ sebagai selisih nilai fungsi tujuan antara solusi hasil pencarian terdekat dengan solusi saat ini, yaitu $\Delta=f(Y)-f(X)$. Jika $\Delta<0$, maka solusi baru lebih baik dari solusi saat ini dan Y menggantikan X. Sebaliknya, jika $\Delta\geq0$, solusi baru dapat diterima dengan probabilitas $\exp(-\Delta/T)$. Proses penerimaan tersebut dilakukan dengan pembangkitan bilangan acak r~U(0,1) dan penggantian X dengan Y jika $r < \exp(-\Delta/T)$.

SA (T_0, T_{max}, I_{iter}, $N_{non\text{-}improving}$, α)
Tahap 1: Bangkitkan solusi awal X dengan metode NN
Tahap 2: $T = T_0$; I=0; N=0; $F_{best}=f(X)$; $X_{best}=X$
Tahap 3: $I=I+1$;
Tahap 4: (Bangkitkan solusi baru Y berdasarkan solusi X)
Tahap 4.1: Bangkitkan bilangan acak $r(0,1)$
Tahap 4.2: Bangkitkan solusi Y dari solusi X sesuai r
  Jika $r \leq 1/6$: Lakukan perpindahan *swap* pada eselon pertama.
  Jika $1/6<r \leq 2/6$: Lakukan perpindahan *swap* pada eselon kedua.
  Jika $2/6<r \leq 3/6$: Lakukan perpindahan *insert* pada eselon pertama.
  Jika $3/6<r \leq 4/6$: Lakukan perpindahan *insert* pada eselon kedua.
  Jika $4/6<r \leq 5/6$: Lakukan perpindahan *reverse* pada eselon pertama.
  Jika $r > 5/6$: Lakukan perpindahan *reverse* pada eselon kedua.
Tahap 5: Jika $f(Y)-f(X)<0$, maka $X = Y$, jika tidak maka bangkitkan bilangan acak r (0,1). Jika $r < \exp(-\Delta/T)$, maka $X = Y$.
Tahap 6: jika $f(X)<F_{best}$, maka $X_{best}=X$, N=0
Tahap 7: jika $I = I_{iter}$, maka $T=\alpha T$; I=0; $N=N+1$;
Lakukan perpindahan *swap* pada X_{best};
Lakukan perpindahan *insert* pada X_{best};
Jika tidak kembali ke Tahap 3.
Tahap 8: jika $T \leq T_{max}$ atau $N = N_{non\text{-}improving}$, maka algoritma SA stop. Jika tidak maka kembali ke Tahap 3.

Gambar 3. *Pseudo code* SA usulan

Temperatur saat ini turun menjadi αT, $0<\alpha<1$, setelah beroperasi sebanyak I_{iter} iterasi pada temperatur saat ini. Algoritma akan berhenti jika terjadi dua kondisi: (i) temperatur saat ini (T) di bawah atau sama dengan temperatur akhir (T_{max}), atau (ii) solusi terbaik sudah tidak dapat diperbaiki lagi sampai $N_{non\text{-}improving}$. Solusi terbaik (X_{best}) dan nilai fungsi tujuannya (F_{best}) diperbaharui setiap memperoleh solusi terbaik yang baru ditemukan. Solusi MLRKDE terbaik diperoleh dari X_{best} ketika algoritma berhenti. Metode heuristik SA yang diusulkan disajikan dalam *pseudo code* pada Gambar 3.
Perubahan solusi di dalam SA terjadi karena adanya mekanisme pencarian alternatif solusi

terdekat. Yu, et al. [22] dan Yu and Lin [23] menggunakan tiga *operator* perpindahan terdekat: *swap*, *insert*, dan *reverse*. Probabilitas pemilihan setiap *operator* perpindahan adalah 1/3. Di dalam penelitian ini, penulis melakukan pemilihan *operator* perpindahan dengan cara yang berbeda. Setiap *operator* perpindahan dilakukan secara terpisah di antara bagian solusi eselon pertama dengan eselon kedua. Oleh karena itu, di sini dilakukan enam perpindahan, yaitu: perpindahan *swap* eselon pertama, perpindahan *swap* eselon kedua, perpindahan *insert* eselon pertama, perpindahan *insert* eselon kedua, perpindahan *reverse* eselon pertama dan perpindahan *reverse* eselon kedua. Oleh karena itu, probabilitas dipilihnya setiap *operator* perpindahan adalah 1/6. Ilustrasi penerapan perpindahan dapat ditelusuri di dalam Yu and Lin [23]

2.2. *Large Neighborhood Search* untuk MLRKDE

Large neighborhood search (LNS) dalam artikel ini diadaptasi dari LNS yang disampaikan oleh Breunig, et al. [13]. Solusi awal secara iteratif dirusak dan diperbaiki agar diperoleh perbaikan solusi secara bertahap. Di dalam LNS ada dua jenis *operator* perubahan solusi, yaitu *operator* perusak dan *operator* perbaikan. *Operator* pertama akan mengambil sejumlah anggota solusi, sedangkan *operator* kedua akan menyisipkan kembali anggota yang diambil tersebut ke dalam solusi yang telah diambil anggotanya tersebut. LNS bekerja secara acak, dimana setiap *operator* dipilih berdasarkan bilangan acak (0,1) yang telah dibangkitkan.

Gambar 4 menyajikan *pseudo-code* dari LNS usulan. Sesuai dengan [13], mekanisme perusakan dan perbaikan dilakukan pada setiap iterasi hanya pada solusi eselon kedua. Sedangkan solusi eselon pertama disusun menggunakan metode heuristik sederhana. Hal ini dilakukan karena ukuran solusi eselon pertama relatif kecil sehingga lebih mudah memperoleh solusi optimal atau mendekati optimal.

Setiap tahap perusakan menggunakan satu *operator* dari enam *operator* yang tersedia yang dipilih secara acak. Selanjutnya, mekanisme perbaikan dilakukan dengan merekonstruksi solusi. Dengan demikian mekanisme perusakan-perbaikan dilakukan terus-menerus untuk memperbaiki solusi (Tahap 3–10). Jika solusi yang lebih baik diperoleh, maka solusi tersebut akan diterima sebagai solusi saat ini (Tahap 4-5). Selanjutnya jika solusi saat ini lebih baik dari solusi terbaik, maka solusi tersebut diterima sebagai solusi terbaik (Tahap 6–7). Sementara itu, jika perbaikan solusi tidak dapat diperoleh, algoritma akan merubah mekanisme pencarian alternatif solusi menggunakan mekanisme pencarian lokal (terdekat). Perubahan mekanisme perusakan-perbaikan ke mekanisme pencarian lokal dikontrol oleh parameter N_{MaxLNS} (Tahap 10). N_{MaxLNS} menyatakan jumlah maksimum iterasi tanpa perbaikan solusi yang dilakukan oleh mekanisme perusakan-perbaikan solusi. Ketika algoritma menjalankan mekanisme pencarian lokal, prosedurnya mirip dengan mekanisme perusakan-perbaikan. Secara konsep, Tahap 11–18 sama dengan Tahap 3–10. Perubahan mekanisme pencarian lokal ke mekanisme perusakan-perbaikan solusi dikontrol oleh parameter N_{MaxLS} (Tahap 18). N_{MaxLS} menyatakan jumlah maksimum iterasi tanpa perbaikan solusi yang dilakukan oleh mekanisme pencarian lokal.

Metode heuristik LNS menggunakan enam *operator* perusakan dan dua *operator* perbaikan. Keenam *operator* perusakan tersebut yaitu *random route removal*, *worst removal*, *related node removal*, *not related node removal*, *smallest route removal*, dan *satellite removal*. Sedangkan dua *operator* perbaikan terdiri dari *greedy insertion* dan *modified greedy insertion*.

Penjelasan keenam *operator* perusakan sebagai berikut. Pertama, *random route removal* akan memilih rute secara acak, kemudian semua konsumen dalam rute tersebut diambil (dirusak) dan dimasukkan ke penyimpanan konsumen. Kedua, *worst removal* didasarkan pada kemiripan *operator* seperti yang digunakan oleh [24] and [14]. *Operator* ini mengambil sejumlah *n* konsumen dengan tingkat biaya terburuk. Biaya terburuk ini didefinisikan sebagai selisih biaya antara biaya ketika konsumen masuk bagian rute yang dirusak dengan biaya ketika konsumen tersebut tidak masuk ke rute tersebut. *Operator* ini dikontrol oleh parameter W_α. Ketiga, *related node removal* didasarkan pada kemiripan *operator* seperti yang digunakan oleh [14]. Satu konsumen dipilih sembarang (n_x) dan sejumlah n-1 konsumen yang lokasinya terdekat dengan n_x diidentifikasi. Semua konsumen ini kemudian diambil dari rute yang akan dirusak dan dimasukkan ke dalam penyimpanan konsumen. Jumlah konsumen yang diambil (dihapus) dikontrol oleh parameter R_α. Keempat, *not related node removal* mirip dengan *related node removal*, perbedaannya terletak pada penentuan jumlah konsumen yang akan dihapus. Dalam *operator*

keempat ini, sejumlah n–1 konsumen yang diidentifikasi adalah konsumen yang lokasinya terjauh dari n_x. Jumlah konsumen yang akan dihapus dikontrol oleh parameter NR_α. *Operator* ini diterapkan dengan tujuan untuk mendiversifikasi pencarian solusi. Kelima, *smallest route removal* membuang rute dengan jumlah konsumen paling kecil. Keenam, *satellite removal*, *operator* ini diusulkan oleh [14]. *Operator* ini memilih secara acak salah satu satelit yang dibuka dan menutupnya, semua konsumen yag dilayani satelit tersebut dimasukkan ke dalam penyimpanan konsumen. *Operator* ini juga bermanfaat untuk diversifikasi dalam pencarian alternatif solusi.

LNS (W_α, R_α, NR_α, N_{maxLNS}, N_{MaxLS})
Tahap 1: Bangkitkan solusi awal X yang terdiri dari eselon 1 dan 2 dengan metode NN
Tahap 2: N_{LNS} =0; N_{LS} =0; N=0; F_{best}=$f(X)$; X_{best}=X
Tahap 3: Y=perbaikan(perusakan(X))
Tahap 4: Jika $f(Y) < f(X)$
Tahap 5: $X = Y$
Tahap 6: Jika $f(X) < F_{best}$
Tahap 7: $X_{best} = X$; $F_{best} = f(X)$
Tahap 8: N_{LNS}=0
Tahap 9: Jika tidak, maka $N_{LNS} = N_{LNS} + 1$
Tahap 10: Jika $N_{LNS} = N_{maxLNS}$, maka lanjut ke Tahap 11, jika tidak lanjut ke Tahap 3.
Tahap 11: Y = pencarian lokal (X)
Tahap 12: Jika $f(Y) < f(X)$
Tahap 13: $X = Y$
Tahap 14: Jika $f(X) < F_{best}$
Tahap 15: $X_{best} = X$; $F_{best} = f(X)$
Tahap 16: N_{LS}=0
Tahap 17: Jika tidak, maka $N_{LS} = N_{LS} + 1$
Tahap 18: Jika $N_{LS} = N_{maxLS}$, maka lanjut ke Tahap 3, jika tidak lanjut ke Tahap 11
Tahap 19: Algoritma berhenti setelah mencapai kriteria pemberhentian.
Tahap 20: Solusi akhir = X_{best}

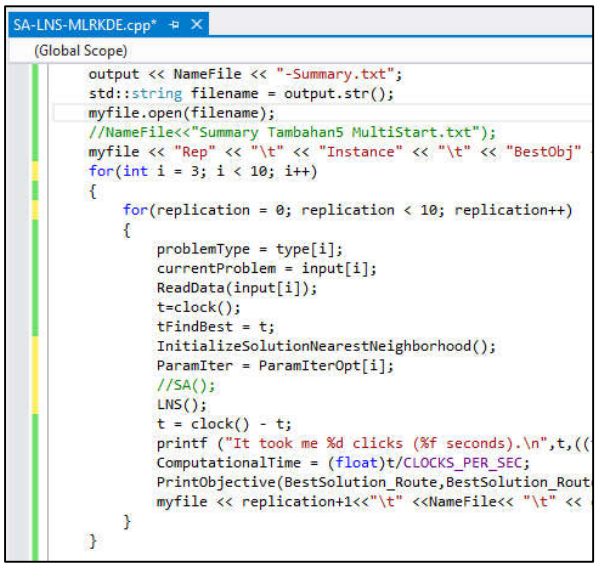
Gambar 4. *Pseudo code* LNS usulan

Penjelasan berikutnya adalah dua *operator* perbaikan. Pertama, *greedy insertion* yang diadopsi dari *operator* yang digunakan oleh [14]. *Operator* ini menyisipkan konsumen secara acak satu per satu ke dalam posisi representasi solusi yang meminimalkan biaya distribusi pada setiap satelit yang dibuka dan rutenya. Kedua, *modified greedy insertion* yang diusulkan untuk diversifikasi pencarian solusi. Letak perbedaan dengan *operator* pertama adalah pada urutan konsumen di tempat penyimpanan konsumen sebelum disisipkan ke solusi yang telah dirusak. Metode pertama, sebelum disisipkan urutan konsumen di tempat penyimpanan tanpa mengalami perubahan, sedangkan pada metode kedua, urutan konsumen dibalik (*reverse*) terlebih dahulu sebelum disisipkan.

3. HASIL PENELITIAN

3.1. Data Numerik untuk Pengujian Metode

Metode heuristik yang diusulkan diterapkan dalam Microsoft Visual C++ 2012 dan dijalankan pada komputer Intel Core i7 6700 CPU pada 3.40 GHz dan 8 GB RAM Windows 10. Cetak layar salah satu bagian program tersebut disajikan dalam Gambar 5. Untuk memahami kinerja dari metode heuristik SA dan LNS, penulis menguji kedua metode tersebut pada himpunan data numerik MLRDE dari Nguyen. Himpunan data numerik ini tersedia di Prodhon [25]. Himpunan data numerik tersebut terdiri dari satu depo dan dua grup satelit. Grup pertama terdiri dari lima satelit yang berisi 25, 50, dan 100 konsumen. Grup kedua terdiri dari 10 satelit yang berisi 50, 100, dan 200 konsumen. Dengan demikian himpunan data numerik tersebut terdiri dari 6 grup dengan berbagai kombinasi jumlah satelit dan konsumen. Di dalam penelitian ini, penulis memilih 12 data numerik yang terdiri dari dua data dari setiap grup.



Gambar 5. Cetak layar salah satu bagian program dari metode yang diusulkan dalam C++

3.2. Nilai Parameter

Pemilihan nilai parameter dapat mempengaruhi kualitas hasil perhitungan. Untuk percobaan awal, kombinasi nilai berbagai parameter SA usulan sebagai berikut:

T_0 = 5, 10, 15;

T_{max} = 0.01, 0.001, 0.0001;
I_{iter} = 600L, 800L, 1000L, dimana L adalah panjang (jumlah anggota) di dalam representasi solusi;
$N_{non\text{-}improving}$ = 200, 400, 600;
α = 0.9, 0.99, 0.999.

Setiap kombinasi nilai parameter dijalankan 20 kali. Hasil percobaan awal untuk menentukan nilai-nilai parameter dalam SA disajikan berbentuk grafik dalam Lampiran A. Penggunaan T_0 = 10, T_{max} = 0.0001, I_{iter} = 1000L, $N_{non\text{-}improving}$ = 200, dan α = 0.9 nampak menghasilkan hasil terbaik di antara semua kombinasi yang mungkin. Selanjutnya, nilai-nilai parameter tersebut digunakan dalam perhitungan berikutnya.

Sedangkan, percobaan awal untuk LNS usulan menggunakan kombinasi nilai parameter sebagai berikut:

W_α = 8, 10, 12;
R_α = 10, 11, 12;
NR_α = 6, 8, 10;
N_{MaxLNS} = 5, 10, 15;
N_{MaxLS} = 50, 75, 100.

Setiap kombinasi nilai parameter dijalankan 20 kali. Hasil percobaan awal untuk menentukan nilai-nilai parameter dalam LNS disajikan berbentuk grafik dalam Lampiran B. Penggunaan W_α = 10, R_α = 10, NR_α = 6, N_{MaxLNS} = 5, dan N_{MaxLS} = 75 nampak menghasilkan hasil terbaik di antara semua kombinasi yang mungkin. Selanjutnya, nilai-nilai parameter tersebut digunakan dalam perhitungan berikutnya.

3.3. Hasil Percobaan

Tabel 1 dan 2 menunjukkan kinerja kedua metode heuristik. Tabel 1 menyajikan nilai fungsi tujuan antara SA dan LNS. Setiap sampel dijalankan 10 kali. Kode sampel dikaitkan dengan jumlah satelit dan konsumen, sebaran lokasi konsumen dan kapasitas kendaraan di kedua eselon. Misalnya, sampel 25-5N menunjukkan bahwa sampel tersebut terdiri dari 25 konsumen dan 5 potensial satelit. Kode N menunjukkan lokasi konsumen menyebar mengikuti distribusi normal. Sedangkan kode b pada sampel 25-5Nb dan sampel-sampel lainnya menunjukkan bahwa kapasitas kendaraan yang dipakai di kedua eselon lebih besar dari kapasitas kendaraan dari sampel-sampel tanpa kode b. Kolom “Min(f)” menunjukkan nilai fungsi tujuan terkecil dari 10 kali pengulangan dan kolom “Rerata(f)” menunjukkan rata-rata nilai fungsi tujuan dari 10 kali pengulangan. Selanjutnya Tabel 2 menunjukkan waktu perhitungan dari kedua metode. Kolom “Min(t)” menunjukkan waktu yang diperlukan algoritma untuk menyelesaikan perhitungan. Waktu ini merupakan waktu tercepat dari 10 pengulangan. Sedangkan kolom “Rerata(t) menunjukkan rata-rata waktu perhitungan algoritma dari 10 kali pengulangan.

Tabel 1. Hasil nilai fungsi tujuan dari SA dan LNS

Sampel	SA		LNS	
	Min(f)	Rerata(f)	Min(f)	Rerata(f)
25-5N	80.370	81.954	80.370	83.473
25-5Nb	64.562	66.115	64.562	65.786
50-5N	142.484	145.995	145.071	149.052
50-5Nb	117.067	119.260	114.413	119.750
50-10N	120.572	123.021	119.476	123.502
50-10Nb	90.057	95.826	94.869	100.981
100-5N	206.121	213.013	212.740	217.127
100-5Nb	161.899	171.521	165.079	167.538
100-10N	226.232	231.235	239.357	249.625
100-10Nb	171.304	181.511	184.451	189.471
200-10N	376.156	388.005	387.954	403.579
200-10Nb	283.910	296.044	300.452	307.858
Rerata	170.061	176.125	175.733	181.478

Berdasarkan Tabel 1, menunjukkan bahwa pada sampel kecil sampai 50 konsumen dan 10 satelit, kedua metode belum menunjukkan perbedaan kinerja secara signifikan. SA lebih bagus dari LNS pada sampel 50-5N dan 50-10Nb, sedangkan LNS lebih bagus dari SA pada sampel 50-5Nb dan 50-10N. Hal ini dapat diinterpretasikan bahwa penggunaan bilangan acak dalam pemilihan *operator* untuk menjalankan iterasi berikutnya belum signifikan perbedaannya di kedua metode ini untuk ukuran sampel relatif kecil. Sedangkan untuk ukuran sampel yang lebih besar solusi yang dihasilkan metode SA lebih baik daripada solusi LNS. Hal ini dimungkinkan karena pada metode SA adanya strategi untuk menghindari terjebaknya solusi optimal lokal dengan menerima solusi buruk dari solusi sebelumnya pada tingkat probabilitas tertentu, sedangkan pada metode LNS kriteria penerimaan solusi baru hanya didasarkan dari solusi yang lebih baik dari solusi sebelumnya.

Tabel 2. Hasil waktu perhitungan dari SA dan LNS (dalam detik)

Sampel	SA		LNS	
	Min(t)	Rerata(t)	Min(t)	Rerata(t)
25-5N	0,05	6,93	0,17	44,03
25-5Nb	0,04	6,66	0,13	44,39
50-5N	0,17	17,95	11,42	81,41
50-5Nb	2,41	19,36	72,54	85,45
50-10N	15,13	25,50	53,04	76,20
50-10Nb	4,22	14,44	17,92	76,96
100-5N	1,03	51,48	262,78	333,08
100-5Nb	1,08	50,19	59,15	332,78
100-10N	33,11	89,12	195,11	326,28
100-10Nb	7,98	62,72	235,44	248,12
200-10N	338,37	608,85	1.273,57	1.315,43
200-10Nb	219,79	475,05	886,06	1.261,88
Rerata	51,95	119,02	255,61	352,17

Selanjutnya, rerata fungsi tujuan di Tabel 1 menunjukkan bahwa 10 dari 12 sampel yang diselesaikan dengan metode SA lebih bagus daripada LNS. Namun demikian yang perlu diperhatikan pada sampel 100-5Nb, meskipun nilai fungsi tujuan terkecil dari 10 kali pengulangan pembangkitan solusi, tetapi rerata fungsi tujuan SA lebih tinggi daripada LNS. Hal tersebut dapat diintepretasikan bahwa pada sampel tersebut metode LNS lebih stabil dalam menghasilkan solusi.

Selanjutnya berdasarkan Tabel 2, secara keseluruhan metode SA juga lebih unggul dari metode LNS dari sisi waktu perhitungan. Konsep metode SA yang mengandalkan mekanisme pencarian alternatif solusi terdekat (lokal) membutuhkan waktu perhitungan yang lebih kecil dibandingkan dengan metode LNS. Konsep perusakan dan perbaikan solusi membutuhkan usaha yang lebih lama untuk menghasilkan solusi baru. Hal tersebutlah yang menyebabkan metode LNS memerlukan waktu perhitungan yang lebih lama. Namun demikian, kedua metode mempunyai kecenderungan yang sama dalam hal waktu perhitungan, yaitu semakin besar ukuran sampel maka akan semakin lama waktu perhitungan. Hal ini disebabkan ukuran iterasi dikaitkan dengan jumlah satelit dan konsumen, semakin besar jumlah entitas tersebut maka akan membutuhkan waktu yang lebih lama dalam menyelesaikan satu iterasi.

Metode SA dan LNS yang telah dikembangkan dalam artikel ini juga dibandingkan dengan metode *greedy randomized adaptive search* dengan *learning process* dan *path relinking* (GRASP-LP-PR) oleh [11] dan hibrid SA oleh [26] untuk menyelesaikan MLRKDE, yang masing-masing disajikan dalam Tabel 3 dan 4. Dalam Tabel 3, kolom SA_S dan LNS_S masing-masing merupakan persentase selisih antara nilai fungsi tujuan metode SA dan LNS dari Tabel 1 dengan nilai fungsi tujuan dari metode GRASP-LP-PR. Jika persentasenya positif, maka metode tersebut lebih baik dengan metode SA atau LNS, dan sebaliknya. Sebagai contoh nilai 3,8% dalam Tabel 3 kolom SA_S dan sampel 50-5Nb menunjukkan bahwa pada sampel tersebut metode GRASP-LP-PR lebih bagus dari metode SA. Nilai tersebut diperoleh dari perhitungan $(117{,}067-112{,}764)/\ 112{,}764\times100\%$. Berdasarkan Tabel 3 tersebut, hasil perhitungan metode SA-LNS sama dengan GRASP-LP-PR pada sampel 25-5N dan 25-5Nb, sedangkan pada sampel 50-5N, metode SA unggul 0,6%. Sedangkan dalam Tabel 4, penjelasan kolom SA_s dan LNS_S mirip seperti yang telah dijelaskan dalam Tabel 3. Dari tabel tersebut menunjukkan bahwa hasil perhitungan SA-LNS sama dengan hibrid SA pada sampel 25-5N dan 25-5Nb. Sementara pada sampel 100-5Nb, SA lebih unggul 0.6%. Dari kedua tabel tersebut dapat disampaikan bahwa metode SA-LNS masih ada potensi untuk diperbaiki karena secara umum hasilnya masih di bawah GRASP-LP-PR dan hibrid SA terutama untuk sampel ukuran besar.

Tabel 3. Perbandingan hasil perhitungan SA-LNS dengan GRASP-PR

Sampel	Min(f)	Min(t)	SA_S	LNS_S
25-5N	80,370	2.2	**0.0%**	**0.0%**
25-5Nb	64,562	1.8	**0.0%**	**0.0%**
50-5N	143,328	3.9	**-0.6%**	1.2%
50-5Nb	112,764	3.6	3.8%	1.5%
50-10N	116,132	7.1	3.8%	2.9%
50-10Nb	87,315	10	3.1%	8.7%
100-5N	196,999	8	4.6%	8.0%
100-5Nb	159,714	8.5	1.4%	3.4%
100-10N	215,792	32.4	4.8%	10.9%
100-10Nb	160,322	29.5	6.8%	15.1%
200-10N	357,286	35.9	5.3%	8.6%
200-10Nb	264,241	77.6	7.4%	13.7%
Rerata	163,235	18.4	3.4%	6.2%

Tabel 4. Perbandingan hasil perhitungan SA-LNS dengan hibrid SA

Sampel	Min(f)	Min(t)	SA_S	LNS_S
25-5N	80,370	2	**0.0%**	**0.0%**
25-5Nb	64,562	2	**0.0%**	**0.0%**
50-5N	138,444	43	2.9%	4.8%
50-5Nb	111,840	31	4.7%	2.3%
50-10N	116,132	42	3.8%	2.9%
50-10Nb	89,744	32	0.3%	5.7%
100-5N	198,444	81	3.9%	7.2%
100-5Nb	162,813	60	**-0.6%**	1.4%
100-10N	217,056	79	4.2%	10.3%
100-10Nb	158,269	58	8.2%	16.5%
200-10N	356,391	483	5.5%	8.9%
200-10Nb	269,577	430	5.3%	11.5%
Rerata	163,637	111.9	3.2%	5.9%

4. KESIMPULAN

Dari hasil percobaan yang telah dilakukan dapat ditarik kesimpulan bahwa metode heuristik SA lebih unggul dibandingkan LNS ditinjau baik dari kualitas solusi maupun waktu perhitungan. MLRKDE dapat diterapkan di berbagai perusahaan distribusi barang dari pusat produksi ke area konsumen melalui fasilitas antara seperti distribusi media cetak, benda pos dan barang kebutuhan pokok lainnya.

Untuk kelanjutan penelitian berikutnya dapat disarankan hal-hal sebagai berikut: Menambahkan karakteristik lain pada model MLRKDE yang lebih mendekati masalah nyata seperti adanya rentang waktu pelayanan konsumen, rentang waktu tertentu di jalan-jalan penghubung antar lokasi sebagai efek program ganjil genap kendaraan yang bisa lewat pada jam tertentu. Mengkaji lebih lanjut penerapan metode heuristik LNS pada ukuran masalah yang lebih besar seperti melayani konsumen yang jumlahnya 1000 lokasi ke atas.

5. DAFTAR PUSTAKA

[1] D. Abramson, "Constructing school timetables using simulated annealing: sequential and parallel algorithms," *Management science,* vol. 37, pp. 98-113, 1991. https://doi.org/10.1287/mnsc.37.1.98

[2] V. Jayaraman and A. Ross, "A simulated annealing methodology to distribution network design and management," *European Journal of Operational Research,* vol. 144, pp. 629-645, 2003. https://doi.org/10.1016/S0377-2217(02)00153-4

[3] S.-W. Lin, S.-Y. Chou, and S.-C. Chen, "Meta-heuristic approaches for minimizing total earliness and tardiness penalties of single-machine scheduling with a common due date," *Journal of Heuristics,* vol. 13, pp. 151-165, 2007. https://doi.org/10.1007/s10732-006-9002-2

[4] A. R. McKendall, J. Shang, and S. Kuppusamy, "Simulated annealing heuristics for the dynamic facility layout problem," *Computers & operations research,* vol. 33, pp. 2431-2444, 2006. https://doi.org/10.1016/j.cor.2005.02.021

[5] A. Van Breedam, "Comparing descent heuristics and metaheuristics for the vehicle routing problem," *Computers & Operations Research,* vol. 28, pp. 289-315, 2001. https://doi.org/10.1016/S0305-0548(99)00101-X

[6] H. Asefi, S. Lim, M. Maghrebi, and S. Shahparvari, "Mathematical modelling and heuristic approaches to the location-routing problem of a cost-effective integrated solid waste management," *Annals of Operations Research,* vol. 273, pp. 75-110, 2019. https://doi.org/10.1007/s10479-018-2912-1

[7] R. Cuda, G. Guastaroba, and M. G. Speranza, "A survey on two-echelon routing problems," *Computers & Operations Research,* vol. 55, pp. 185-199, 2015. https://doi.org/10.1016/j.cor.2014.06.008

[8] S. K. Jacobsen and O. B. Madsen, "A comparative study of heuristics for a two-level routing-location problem," *European Journal of Operational Research,* vol. 5, pp. 378-387, 1980. https://doi.org/10.1016/0377-2217(80)90124-1

[9] M. Boccia, T. G. Crainic, A. Sforza, and C. Sterle, "A metaheuristic for a two echelon

location-routing problem," in *International Symposium on Experimental Algorithms*, 2010, pp. 288-301. https://doi.org/10.1007/978-3-642-13193-6_25

[10] M. Schwengerer, S. Pirkwieser, and G. R. Raidl, "A Variable Neighborhood Search Approach for the Two-Echelon Location-Routing Problem," in *EvoCOP*, 2012, pp. 13-24. https://doi.org/10.1007/978-3-642-29124-1_2

[11] V.-P. Nguyen, C. Prins, and C. Prodhon, "Solving the two-echelon location routing problem by a GRASP reinforced by a learning process and path relinking," *European Journal of Operational Research,* vol. 216, pp. 113-126, 2012 https://doi.org/10.1016/j.ejor.2011.07.030

[12] V.-P. Nguyen, C. Prins, and C. Prodhon, "A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem," *Engineering Applications of Artificial Intelligence,* vol. 25, pp. 56-71, 2012. https://doi.org/10.1016/j.engappai.2011.09.012

[13] U. Breunig, V. Schmid, R. Hartl, and T. Vidal, "A large neighbourhood based heuristic for two-echelon routing problems," *Computers & Operations Research,* vol. 76, pp. 208-225, 2016. https://doi.org/10.1016/j.cor.2016.06.014

[14] V. C. Hemmelmayr, J.-F. Cordeau, and T. G. Crainic, "An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics," *Computers & operations research,* vol. 39, pp. 3215-3228, 2012. https://doi.org/10.1016/j.cor.2012.04.007

[15] L. Chwif, M. R. P. Barretto, and L. A. Moscato, "A solution to the facility layout problem using simulated annealing," *Computers in industry,* vol. 36, pp. 125-132, 1998.

[16] A. Lim, B. Rodrigues, and X. Zhang, "A simulated annealing and hill-climbing algorithm for the traveling tournament problem," *European Journal of Operational Research,* vol. 174, pp. 1459-1478, 2006. https://doi.org/10.1016/S0166-3615(97)00106-1

[17] A. Van Breedam, "Improvement heuristics for the vehicle routing problem based on simulated annealing," *European Journal of Operational Research,* vol. 86, pp. 480-490, 1995. https://doi.org/10.1016/0377-2217(94)00064-J

[18] S.-W. Lin, F. Y. Vincent, and S.-Y. Chou, "Solving the truck and trailer routing problem based on a simulated annealing heuristic," *Computers & Operations Research,* vol. 36, pp. 1683-1692, 2009. https://doi.org/10.1016/j.cor.2008.04.005

[19] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics,* vol. 21, pp. 1087-1092, 1953. https://doi.org/10.1063/1.1699114

[20] S. Kirkpatrick, J. C. Gelatt, and M. P. Vecchi, "Optimization by simmulated annealing," *science,* vol. 220, pp. 671-680, 1983. https://doi.org/10.1126/science.220.4598.671

[21] R. Eglese, "Simulated annealing: a tool for operational research," *European journal of operational research,* vol. 46, pp. 271-281, 1990. https://doi.org/10.1016/0377-2217(90)90001-R

[22] V. Yu, F., S.-W. Lin, W. Lee, and C.-J. Ting, "A simulated annealing heuristic for the capacitated location routing problem," *Computers & Industrial Engineering,* vol. 58, pp. 288-299, 2010. https://doi.org/10.1016/j.cie.2009.10.007

[23] V. F. Yu and S.-Y. Lin, "A simulated annealing heuristic for the open location-routing problem," *Computers & Operations Research,* vol. 62, pp. 184-196, 2015. https://doi.org/10.1016/j.cor.2014.10.009

[24] S. Ropke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," *Transportation science,* vol. 40, pp. 455-472, 2006. https://doi.org/10.1287/trsc.1050.0135

[25] C. Prodhon. (2016, 11 January 2020). *LRP-2E instances*. Available: http://prodhonc.free.fr/Instances/instances0_us.html

[26] K. Pichka, A. H. Bajgiran, M. E. Petering, J. Jang, and X. Yue, "The two echelon open location routing problem: Mathematical model and hybrid heuristic," *Computers & Industrial Engineering,* vol. 121, pp. 97-112, 2018. https://doi.org/10.1016/j.cie.2018.05.010

Biografi Penulis

Winarno

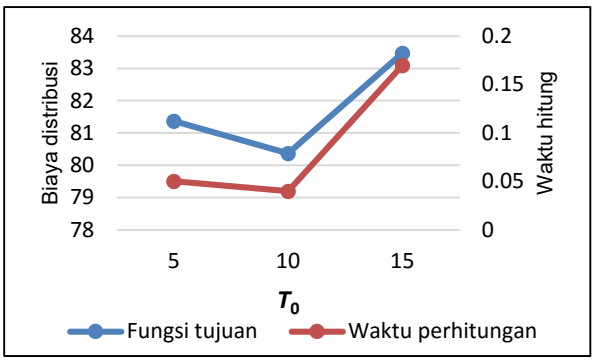
Winarno merupakan dosen tetap di Program studi Teknik Industri, Universitas Singaperbangsa Karawang. Memperoleh gelar S2 di jurusan Teknik dan Manajemen Industri – Institut Teknologi Bandung dan sedang menyelesaikan program S3 di National Taiwan University of Science and Technology (NTUST). Mempunyai pengalaman sebagai Ketua Program Studi Teknik Industri, Universitas Singaperbangsa Karawang pada masa jabatan 2011-2015.

A. A. N. Perwira Redi

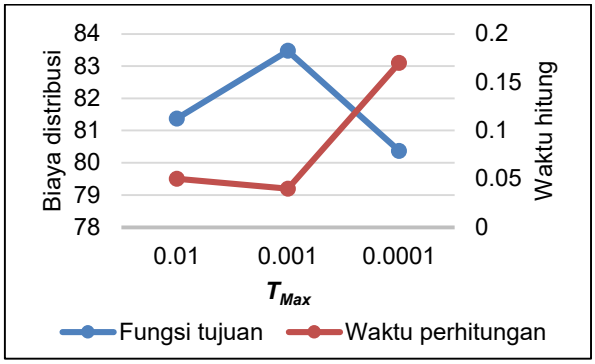
Dr. Redi adalah dosen tetap Jurusan Teknik Logistik Universitas Pertamina. Berpengalaman sebagai Research Fellow di Monash University, Australia. Memperoleh gelar S2 dan S3 di National Taiwan University of Science and Technology (NTUST).

Lampiran A

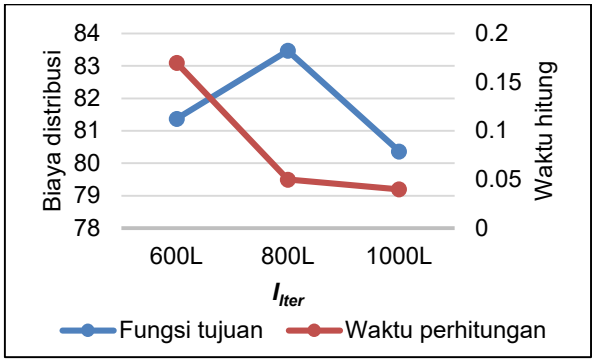
Hasil percobaan awal untuk menentukan nilai – nilai parameter SA



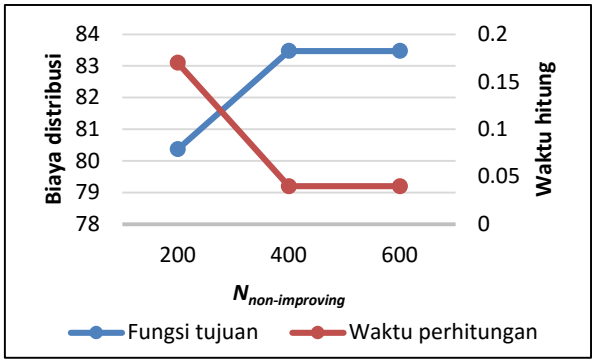
Gambar A1. *Setting* nilai parameter T_0



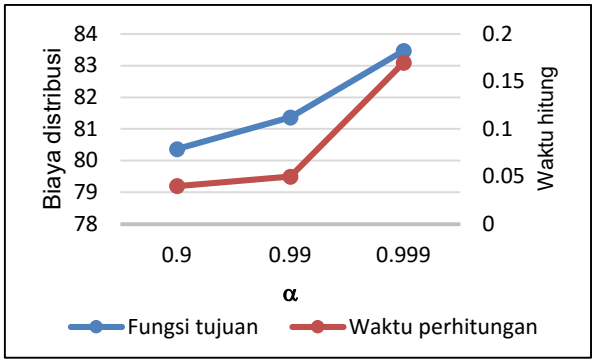
Gambar A2. *Setting* nilai parameter T_{Max}



Gambar A3. *Setting* nilai parameter I_{Iter}



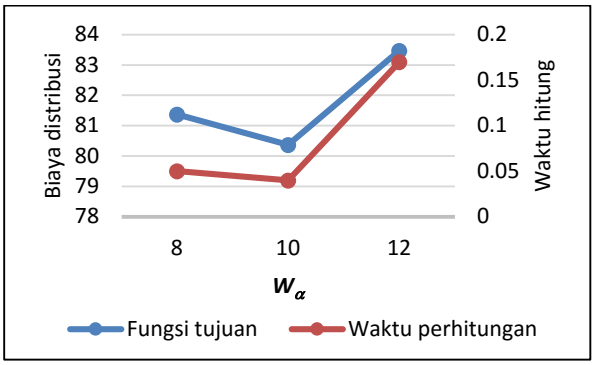
Gambar A4. *Setting* nilai parameter $N_{Non\text{-}improving}$



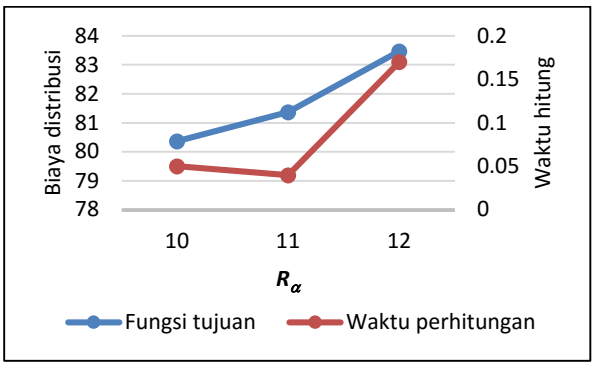
Gambar A5. *Setting* nilai parameter α

Lampiran B

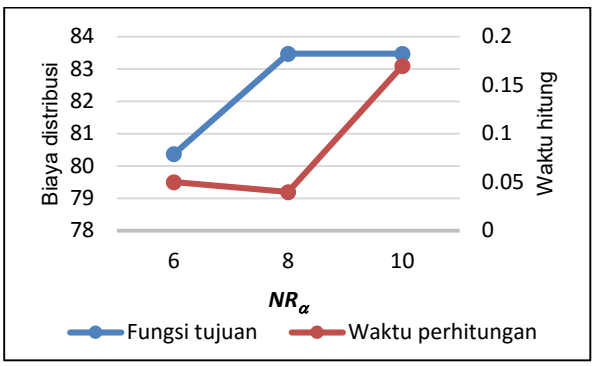
Hasil percobaan awal untuk menentukan nilai-nilai parameter LNS.



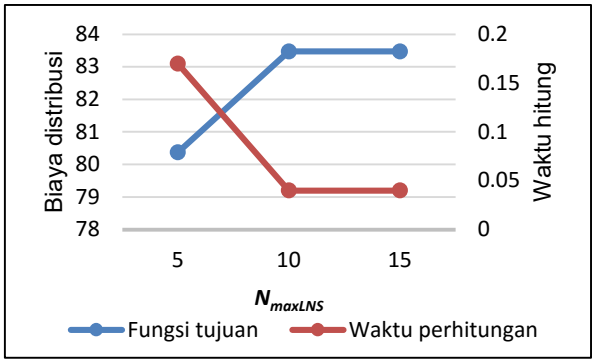
Gambar B1. *Setting* nilai parameter W_α



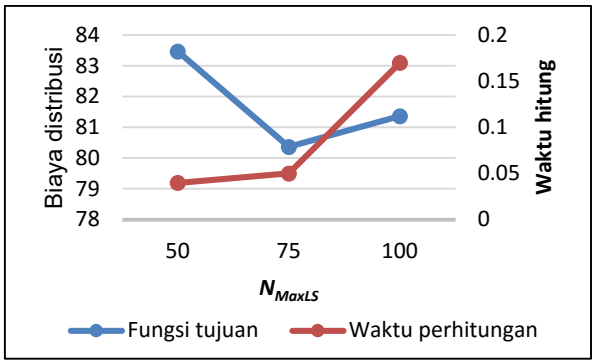
Gambar B2. *Setting* nilai parameter R_α



Gambar B3. *Setting* nilai parameter NR_α



Gambar B4. *Setting* nilai parameter N_{MaxLNS}



Gambar B4. *Setting* nilai parameter N_{MaxLS}