



JAIER

Journal of Agro-industry Engineering Research



Algoritma *Simulated Annealing* untuk Penentuan Lokasi Fasilitas dan Rute Kendaraan dengan Mempertimbangkan Jendela Waktu dan Kendaraan Heterogen Terbatas pada Sistem Pengelolaan Limbah

Andi Muhammad Fiqri Achmad¹, Asadian Puja Enggita², Widya Hastuti Afris³
^{1,2,3} Politeknik ATI Makassar

Received: May 2022 Accepted: June 2022 Published: July 2022

Doi:

Abstract. Location – Routing Problem can be classified as a NP-hard problem, as it consists of two problems that are NP-hard. Nearest Neighbor algorithm is chosen to generate the initial solution, and then improved the solution using Simulated Annealing Algorithm. Comparison or gap of the average objective function between analytical method and the algorithms is 2,92%, and hap of the average computational time is 98586,06%. The result of the analysis of variance for the solution quality is $0.000 < 0.05$, which means that there is a difference in the solution quality between the analytical method and the SA algorithm. The results of the analysis of variance for computational time is $0.023 < 0.05$, which means the computational time by the analytical method is different from the computational time of the algorithm.

Keyword: Location-Routing Problem, Waste Management, Nearest Neighbor, Simulated Annealing

Abstrak. Masalah penentuan lokasi dan rute kendaraan dapat diklasifikasikan sebagai NP-hard, karena mencakup dua masalah yang merupakan NP-hard. Algoritma *Nearest Neighbor* dipilih untuk menghasilkan solusi awal, kemudian dilakukan perbaikan solusi menggunakan algoritma *Simulated Annealing*. Perbandingan atau gap rata – rata fungsi objektif antara metode analitik dan algoritma adalah sebesar 2,92%, dan gap rata – rata waktu komputasi adalah sebesar 98586,06%. Hasil analisis variansi kualitas solusi bernilai $0,000 < 0,05$, yang berarti terdapat perbedaan kualitas solusi yang dihasilkan antara metode analitik dan algoritma SA. Hasil analisis variansi waktu komputasi bernilai $0,023 < 0,05$, yang berarti waktu komputasi oleh metode analitik berbeda dengan waktu komputasi algoritma.

Kata Kunci: Penentuan Lokasi Dan Rute Kendaraan, Pengelolaan Limbah, Nearest Neighbor, Simulated Annealing

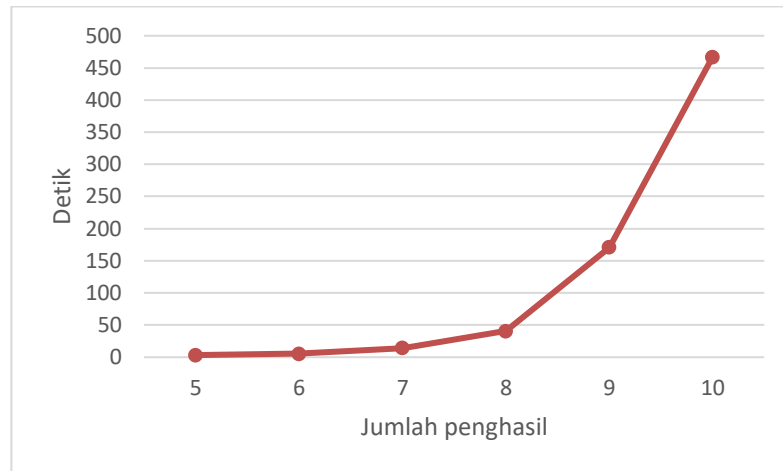
1. Pendahuluan

[1], [2], dan [3] menjelaskan bahwa masalah penentuan lokasi dan rute kendaraan dapat diklasifikasikan sebagai NP-hard, karena mencakup dua masalah yang merupakan NP-hard. Artinya adalah apabila ukuran masalahnya bertambah (jumlah depot dan konsumen semakin banyak) maka semakin sulit untuk menemukan solusi optimal globalnya dengan waktu komputasi yang wajar. Sehingga cara untuk dapat menghasilkan solusi dengan waktu yang relatif singkat adalah dengan menggunakan algoritma metaheuristik [4].

*Corresponding author at: Politeknik ATI Makassar, Makassar, 90211, Indonesia

E-mail address: am_fiqri@atim.ac.id

Pemecahan masalah untuk model matematis yang telah dikembangkan oleh [5] menggunakan metode analitik. Pada tabel dan grafik hubungan waktu komputasi dan jumlah titik, dapat dilihat bahwa waktu komputasi akan bertambah secara eksponensial seiring bertambahnya jumlah pelanggan atau penghasil. Selain itu, semakin bertambahnya jumlah kandidat fasilitas gudang, waktu komputasi juga akan semakin meningkat. Hal ini dapat terjadi karena semakin banyaknya kombinasi kemungkinan solusi yang harus diperhitungkan. Oleh karena itu dibutuhkan adanya algoritma pemecahan masalah yang dapat menyelesaikan masalah tersebut dengan waktu komputasi yang lebih cepat.



Gambar 1. Perbandingan jumlah penghasil dan waktu komputasi

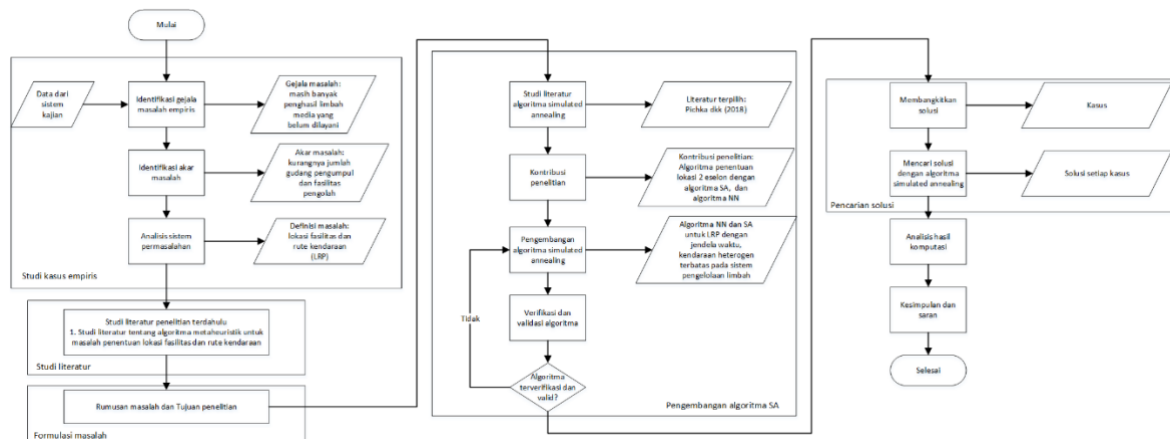
Survei yang dilakukan oleh [1] menerangkan bahwa terdapat empat algoritma yang menghasilkan solusi mendekati optimal untuk masalah penentuan lokasi dan rute kendaraan. Keempat algoritma tersebut adalah *greedy randomized adaptive search procedure-integer linear programming* (GRASP-ILP) oleh [6], *multiple ant colony optimization* (MACO) oleh [7], *adaptive large neighborhood search* (ALNS) oleh [8], dan *simulated annealing* (SALRP) oleh [9]. Pada survei tersebut, mereka lebih lanjut menjelaskan dua jenis algoritma metaheuristik yang digunakan untuk masalah pemilihan lokasi dan rute berkapasitas (CLRP), yaitu *neighborhood-based* dan *population and multi-agent metaheuristics*.

Algoritma metaheuristik yang berdasarkan populasi dan multiagen cenderung masih kurang dikembangkan pada masalah pemilihan lokasi dan rute, walaupun algoritma jenis ini sudah sering dijumpai pada masalah rute kendaraan. Algoritma berbasis multiagen, seperti *ant colony* dan *particle swarm optimization*, bekerja berdasarkan pada perpindahan semut atau agen sehingga membentuk suatu rute. Metode ini menyebabkan algoritma kurang sesuai apabila diimplementasikan pada masalah penentuan lokasi. Sedangkan pada *genetic algorithm*, perancangan pengkodean kromosom yang mencakup dimensi lokasi dan rute menggunakan operator *crossover* menyebabkan algoritma ini kurang sesuai diterapkan pada masalah pemilihan lokasi dan rute. Dengan kata lain, jenis algoritma yang paling tepat untuk menyelesaikan masalah pada penelitian ini adalah *neighborhood-based metaheuristics*. Salah satu algoritma *neighborhood-based* tersebut adalah *simulated annealing* [1].

Simulated annealing (SA) merupakan salah satu algoritma yang paling banyak digunakan untuk menyelesaikan masalah penentuan lokasi dan rute kendaraan, dan menjadi salah satu algoritma terbaik yang dapat menghasilkan solusi mendekati optimal dengan waktu komputasi yang singkat. Oleh karena itu, penelitian ini akan menggunakan algoritma *simulated annealing* untuk menyelesaikan masalah penentuan lokasi dan rute kendaraan pada konteks pengelolaan limbah B3 medis dengan mempertimbangkan jendela waktu pelanggan, kendaraan heterogen terbatas. Algoritma *nearest neighbor* dipilih untuk menghasilkan solusi awal karena mudah diimplementasikan dan dijalankan.

2. Metodologi

Metodologi penelitian yang dilakukan secara garis besar dapat dibagi kedalam empat tahapan, yaitu studi kasus empiris, studi literatur, formulasi masalah, dan pengembangan algoritma. Tahapan tersebut dapat dilihat pada Gambar 2.



Gambar 2. Flowchart penelitian

3. Hasil dan Pembahasan

Pembangkitan solusi awal dilakukan menggunakan algoritma *Nearest Neighbor* (NN) untuk memperoleh solusi awal yang layak. Pada penelitian ini, algoritma NN diadaptasi dan dikembangkan dari penelitian Pop dkk. (2011). Penelitian tersebut adalah penelitian untuk menentukan rute kendaraan saja, sehingga pada penelitian ini dikembangkan dengan menambahkan algoritma untuk penentuan lokasi.

Langkah – langkah atau *pseudocode* algoritma NN yang dikembangkan adalah sebagai berikut:

- 1: Input: data penghasil, data kandidat gudang, data pangkalan, data pengolah, data kendaraan kecil, data kendaraan besar
- 2: Bangkitkan bilangan random untuk kandidat gudang dan pengolah
- 3: **while** penghasil terlayani < jumlah penghasil
- 4: Pilih pasangan rute pangkalan – beberapa penghasil - gudang berdasarkan jarak terdekat
- 5: **if** jendela waktu dan kapasitas kendaraan terpenuhi
- 6: Pasangkan hingga membentuk rute menuju gudang pengumpul
- 7: **else**
- 8: Kembali ke langkah 3
- 9: Bentuk rute dari gudang ke pengolah berdasarkan jarak terdekat
- 10: Bentuk solusi awal S_{α} dan hitung nilai objektif awal obj_{α}

Algoritma SA yang digunakan pada penelitian ini dikembangkan dari penelitian Pichka dkk. (2018). Pada penelitian ini, algoritma dikembangkan agar dapat memilih dua jenis fasilitas pada lokasi yang berbeda, sedangkan algoritma yang dikembangkan oleh Pichka dkk. (2018) hanya digunakan untuk memilih satu jenis depot saja. Output dari algoritma NN akan menjadi input pada algoritma SA ini. Input yang digunakan adalah gudang dan fasilitas pengolah yang dibuka, kendaraan yang digunakan, serta rute kendaraan yang terbentuk.

Langkah – langkah atau *pseudocode* algoritma SA yang dikembangkan adalah sebagai berikut:

- 1: Input: solusi awal S_{α} dan nilai objektif awal obj_{α} , parameter SA (temperatur awal T_{α} dan akhir T_{β} , iterasi max, *cooling rate* Cr)
- 2: Atur parameter SA
- 3: **while** $T_{\alpha} > T_{\beta}$
- 4: **for** $i = 1$: iterasi max
- 5: Bangkitkan bilangan random $r = \text{random}(1:6)$
- 6: **if** $r = 1$: tutup gudang terbuka dan buka 1 gudang tertutup secara random
- 7: **if** $r = 2$: buka 1 gudang yang tertutup secara random
- 8: **if** $r = 3$: tutup 1 gudang yang terbuka secara random
- 9: **if** $r = 4$: tutup pengolah terbuka dan buka 1 pengolah tertutup secara random
- 10: **if** $r = 5$: *swap* rute kendaraan untuk 2 penghasil secara random
- 11: **if** $r = 6$: *relocation* rute kendaraan secara random
- 12: Bentuk solusi sementara S_{β} dan hitung nilai objektif sementara obj_{β}
- 13: $\Delta = obj_{\alpha} - obj_{\beta}$

- 14: **if** $\Delta \leq 0$
- 15: $S_\alpha = S_\beta, obj_\alpha = obj_\beta$
- 16: **else**
- 17: Bangkitkan bilangan random $r = \text{random}(0:1)$
- 18: **if** $r < \exp(-\Delta/T_\alpha)$
- 19: $S_\alpha = S_\beta, obj_\alpha = obj_\beta$
- 20: $T_\alpha = T_\alpha \times Cr$
- 21: Bentuk solusi akhir $S_\gamma = S_\alpha$ dan hitung nilai objektif akhir $obj_\gamma = obj_\alpha$

Analisis penentuan parameter algoritma dilakukan untuk menemukan kombinasi parameter yang tepat sehingga akan menghasilkan solusi yang mendekati global optimal. Dalam algoritma SA, parameter yang mempengaruhi solusi adalah temperature awal dan akhir, *cooling rate*, dan jumlah iterasi. Metode yang digunakan untuk penentuan parameter algoritma SA adalah metode Taguchi dengan rasio *signal to noise*. Persamaan *signal to noise* adalah sebagai berikut:

$$\frac{s}{N} = -10x \log \frac{S(Y)^2}{n} \quad 1$$

Keterangan:

Y = respon

n = jumlah susunan orthogonal

Pada Gambar 3 diperoleh level parameter SA terbaik yang memiliki rasio paling rendah. Rasio paling rendah dipilih karena fungsi tujuan algoritma adalah mencari nilai minimum. Parameter temperature awal yang terpilih adalah 4000, parameter temperature akhir yang terpilih adalah 0,001, parameter *cooling rate* yang terpilih adalah 0,8, dan parameter jumlah iterasi yang terpilih adalah 100.

Analisis performansi algoritma dilakukan untuk mengetahui seberapa besar perbandingan antara hasil dari model MIQP menggunakan metode analitik dengan hasil yang diperoleh algoritma SA.

Analisis dilakukan dengan menerapkan tiga buah skenario, yaitu menggunakan jendela waktu dan kendaraan heterogen pada skenario pertama, menggunakan kendaraan heterogen tanpa jendela waktu pada skenario kedua, dan tanpa jendela waktu serta kendaraan homogen pada skenario ketiga. Analisis dilakukan menggunakan lima – dua belas jumlah penghasil, serta dua hingga empat jumlah kandidat gudang. Solusi dan waktu komputasi yang dihasilkan oleh metode analitik selanjutnya dibandingkan dengan solusi dan waktu komputasi oleh algoritma.

Tabel 1. Hasil perbandingan kualitas solusi dan algoritma dengan (a) 2 kandidat gudang, (b) 3 kandidat gudang, (c) 4 kandidat gudang

Jumlah penghasil	MILP		SA		Perbandingan (gap)	
	Objektif	Waktu (s)	Objektif	Waktu (s)	Objektif	Waktu
5	1004002	9.14	1064608	2.54	5.69%	359.84%
	991199	5.53	1009548	2.27	1.82%	243.61%
	991199	8.17	1009548	2.26	1.82%	361.50%
6	1005681	30.67	1005681	3.17	0.00%	967.51%
	991298	17.93	1005681	2.32	1.43%	772.84%
	1005681	21.4	1005681	2.16	0.00%	990.74%
7	1022137	147.57	1076932	2.65	5.09%	5568.68%
	1003304	34.69	1034326	1.87	3.00%	1855.08%
	1011837	33.42	1013542	2.14	0.17%	1561.68%
8	1023348	287.98	1078911	2.36	5.15%	12202.54%
	1004515	88.43	1032618	1.76	2.72%	5024.43%
	1013048	72.77	1134520	1.6	10.71%	4548.13%
9	1046071	2105.05	1050986	2.74	0.47%	76826.64%
	1015223	165.3	1038279	2.41	2.22%	6858.92%
	1022856	395.23	1040938	1.86	1.74%	21248.92%
10	1058538	44128.81	1107534	2.78	4.42%	1587367.27%
	1016265	546.1	1037462	2.16	2.04%	25282.41%
	1051661	34734.6	1108486	1.51	5.13%	2300304.64%
Rata - rata					2.98%	225130.30%

(a)

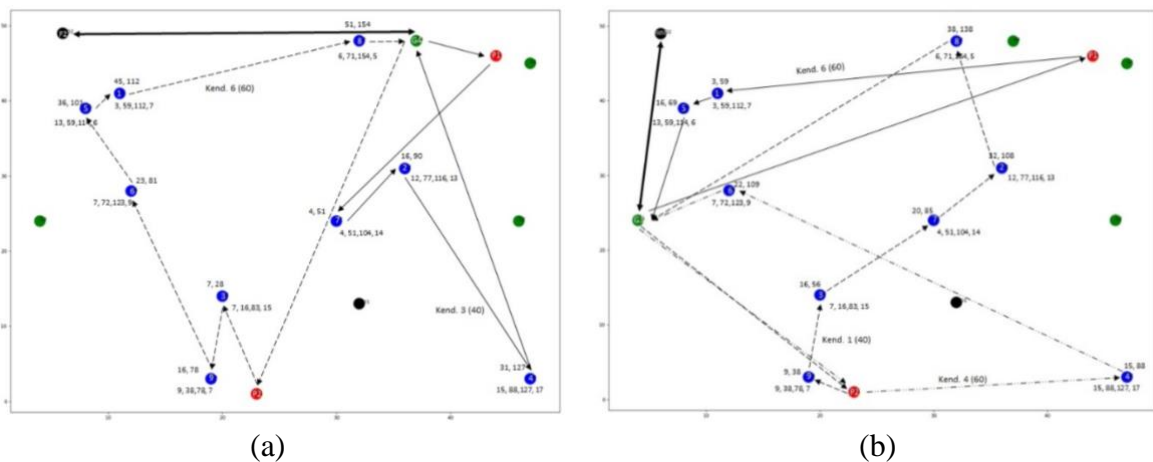
Jumlah penghasil	MILP		SA		Perbandingan (gap)	
	Objektif	Waktu (s)	Objektif	Waktu (s)	Objektif	Waktu
5	970118	28.97	1015722	2.33	4.49%	1243.35%
	949411	20.11	970246	2.18	2.15%	922.48%
	949411	14.51	970246	2.52	2.15%	575.79%
6	970821	39.33	972349	2.85	0.16%	1380.00%
	954536	18.63	972206	2.01	1.82%	926.87%
	954536	34.11	972206	2.08	1.82%	1639.90%
7	986420	153.77	1038379	2.82	5.00%	5452.84%
	962325	39.76	974416	2.34	1.24%	1699.15%
	962325	34.3	974416	2.44	1.24%	1405.74%
8	1023348	740.56	1115496	2.67	8.26%	27736.33%
	969202	97.5	974564	2.13	0.55%	4577.46%
	974031	93.96	1021930	2.11	4.69%	4453.08%
9	1046071	3586.2	1065849	2.79	1.86%	128537.63%
	977591	635.55	989699	2.16	1.22%	29423.61%
	984420	480.05	1040938	1.94	5.43%	24744.85%
10	1058538*	72645.44*	1065801	2.97	NA	NA
	984661	1372.15	1012002	2.24	2.70%	61256.70%
	1034249*	72164.64*	1034249	1.71	NA	NA
Rata - rata					2.80%	18498.49%

(b)

Jumlah penghasil	MILP		SA		Perbandingan (gap)	
	Objektif	Waktu (s)	Objektif	Waktu (s)	Objektif	Waktu
5	970118	30.74	1015722	2.49	4.49%	1234.54%
	949411	23.74	970246	2.18	2.15%	1088.99%
	949411	16.74	970246	2.1	2.15%	797.14%
6	970821	90.34	972349	2.97	0.16%	3041.75%
	954536	83.78	972206	2.34	1.82%	3580.34%
	954536	42.34	972206	2.16	1.82%	1960.19%
7	986420	173.53	1038379	2.75	5.00%	6310.18%
	962325	107.39	970812	1.85	0.87%	5804.86%
	962325	90.61	968277	2.21	0.61%	4100.00%
8	1013826	1919.2	1086808	2.86	6.72%	67104.90%
	969202	119.68	983008	1.99	1.40%	6014.07%
	969202	121.91	1021930	2.08	5.16%	5861.06%
9	1036549	8907.74	1065849	2.8	2.75%	318133.57%
	977591	967.84	1006655	2.03	2.89%	47676.85%
	984420	573.36	1040938	1.89	5.43%	30336.51%
10	1051333*	72708.73*	1065801	2.51	NA	NA
	984661	1628.88	1028202	2.09	4.23%	77936.84%
	1034249*	72180.08*	1034249	1.64	NA	NA
Rata - rata					2.98%	36311.36%

(c)

Berdasarkan Tabel 1(a), diperoleh perbandingan atau gap rata – rata fungsi objektif sebesar 2,98% dan waktu komputasi sebesar 225130,30%. Pada Tabel 1(b), diperoleh perbandingan atau gap rata – rata fungsi objektif sebesar 2,80% dan waktu komputasi sebesar 18498,49%. Sedangkan pada Tabel 1(c), diperoleh perbandingan atau gap rata – rata fungsi objektif sebesar 2,98% dan waktu komputasi sebesar 36311,36%. Rata – rata dari seluruh nilai fungsi objektif yang dihasilkan adalah sebesar 2,92%, dan rata – rata dari seluruh waktu komputasi yang digunakan adalah sebesar 98586,06%.



Gambar 3. Ilustrasi solusi metode analitik (a) dan algoritma SA (b) dengan 9 penghasil 4 gudang

Selanjutnya dilakukan analisis kestabilan solusi untuk melihat apakah algoritma yang dikembangkan dapat menghasilkan solusi yang stabil. Kestabilan solusi dapat dinyatakan dengan nilai koefisien variansi. Analisis kestabilan solusi dilakukan menggunakan kombinasi dari jumlah penghasil dan jumlah kandidat gudang. Jumlah penghasil yang digunakan adalah sebanyak 7, 8, 9, dan 10 penghasil, dan jumlah kandidat gudang yang digunakan adalah sebanyak 2, 3, dan 4 kandidat gudang. Analisis akan dilakukan sebanyak 10 replikasi.

Tabel 2. Hasil percobaan uji kestabilan solusi

Replikasi	Jumlah Penghasil: 7			Jumlah Penghasil: 8			Jumlah Penghasil: 9			Jumlah Penghasil: 10		
	Jumlah Kandidat Gudang			Jumlah Kandidat Gudang			Jumlah Kandidat Gudang			Jumlah Kandidat Gudang		
	2	3	4	2	3	4	2	3	4	2	3	4
1	1098910	1038379	1023310	1079049	1115496	1115496	1055730	1065561	1065561	1107534	1065801	1133488
2	1076932	1035088	1023310	1072652	1115496	1130056	1050986	1065561	1065849	1137848	1132911	1127029
3	1076932	1039839	1038379	1081264	1195335	1104077	1050986	1065849	1065561	1107534	1065801	1066090
4	1076932	1050886	1038379	1114317	1225917	1130056	1058961	1087557	1065849	1111361	1065801	1101092
5	1106760	1071730	1027003	1079049	1115496	1115332	1050986	1065849	1065849	1107534	1065801	1066090
6	1083950	1035088	1023310	1079049	1129857	1229097	1058961	1071314	1065849	1113248	1143129	1066090
7	1108672	1038379	1039389	1072652	1201029	1115496	1055730	1065849	1065849	1112577	1066090	1066090
8	1076932	1023310	1037994	1081264	1225917	1115496	1084938	1065849	1071314	1111361	1065801	1066090
9	1083950	1035088	1035088	1079049	1208988	1130056	1055730	1065849	1065849	1113673	1127029	1127029
10	1089928	1053429	1032247	1067162	1192714	1115496	1058961	1065561	1079408	1133488	1127029	1066090
Rata - rata	1087990	1042122	1031841	1080551	1172625	1130066	1058197	1068480	1067694	1115616	1092519	1088518
Standar deviasi	12582.3	13403.3	6934.38	12711.5	47549.6	35821.8	9950.35	6929.29	4470.22	10875.3	34709.1	30119.6
Koefisien variansi	1.16%	1.29%	0.67%	1.18%	4.05%	3.17%	0.94%	0.65%	0.42%	0.97%	3.18%	2.77%

Pada Tabel 2 diatas dapat dilihat bahwa objektif yang dihasilkan oleh algoritma telah cukup stabil. Nilai objektif yang paling stabil diperoleh dari kombinasi 9 penghasil dan 4 kandidat gudang dengan koefisien variansi 0,42%. Sedangkan kombinasi dengan nilai koefisien variansi yang paling besar diperoleh dari kombinasi 8 penghasil dan 3 kandidat gudang sebesar 4,05%. Selain itu, beberapa kombinasi cenderung menghasilkan nilai yang berubah – ubah pada tiap replikasi, tetapi rentang nilainya cenderung kecil sehingga hanya menghasilkan koefisien variansi yang kecil juga.

Analisis dilanjutkan dengan analisis variansi. Analisis variansi dilakukan untuk mengetahui apakah terdapat perbedaan antara solusi yang dihasilkan oleh metode analitik dan algoritma SA serta faktor – faktor yang berpengaruh terhadap perubahan itu. Analisis variansi yang dilakukan pada penelitian ini melibatkan tiga buah faktor. Ketiga faktor tersebut adalah metode pencarian solusi, jumlah penghasil, dan jumlah kandidat gudang pengumpul. Ketiga faktor ini akan dianalisis satu per satu maupun gabungan antar faktor. Selain melakukan analisis terhadap nilai objektif, penelitian ini juga akan menganalisis waktu komputasi yang dihasilkan dalam pencarian solusi.

Tabel 3. Hasil metode dan jumlah penghasil untuk kualitas solusi

Tests of Between-Subjects Effects

Dependent Variable: Y

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	115063230383.160 ^a	35	3287520868.090	3.740	.000
Intercept	89317299305400.830	1	89317299305400.830	101610.190	.000
Jumlahpenghasil	37706983766.310	5	7541396753.262	8.579	.000
Jumlahkandidatgudang	36707422983.084	2	18353711491.542	20.880	.000
Metode	22387474658.454	1	22387474658.454	25.469	.000
Jumlahpenghasil *	2155323698.504	10	215532369.850	.245	.990
Jumlahkandidatgudang					
Jumlahpenghasil * Metode	4744359217.281	5	948871843.456	1.079	.380
Jumlahkandidatgudang *	147602295.975	2	73801147.987	.084	.920
Metode					
Jumlahpenghasil *					
Jumlahkandidatgudang *	582185055.223	10	58218505.522	.066	1.000
Metode					
Error	56257223410.000	64	879019115.781		
Total	101298836894830.000	100			
Corrected Total	171320453793.160	99			

a. R Squared = .672 (Adjusted R Squared = .492)

Berdasarkan Tabel 3, terlihat bahwa nilai signifikansi jumlah penghasil bernilai $0,000 < 0,05$. Hal ini menjelaskan bahwa semakin banyak jumlah penghasil yang dilayani maka total biaya system juga akan semakin meningkat. Metode pencarian solusi yang digunakan berpengaruh signifikan terhadap solusi yang dihasilkan karena kedua nilai signifikansi metode bernilai $0,000 < 0,05$. Hal ini disebabkan karena rata – rata perbedaan kualitas solusi yang dihasilkan antara metode analitik dan SA mencapai 2,92%. pada tabel terlihat bahwa nilai signifikansi jumlah kandidat gudang bernilai $0,000 < 0,05$. Hal ini menjelaskan bahwa jumlah kandidat gudang berpengaruh signifikan terhadap solusi yang dihasilkan. Adanya perbedaan hasil untuk gudang mungkin disebabkan karena gudang yang terpilih pada tiap scenario berbeda – beda sehingga nilai kandidat gudang juga akan mempengaruhi solusi. Untuk jumlah penghasil dan jumlah kandidat gudang, metode pencarian dan jumlah penghasil, metode pencarian dan jumlah kandidat gudang, serta gabungan ketiga faktor semuanya bernilai $> 0,05$ yang berarti faktor - faktor tersebut tidak memiliki interaksi terhadap solusi yang dihasilkan.

Tabel 4. Hasil metode dan jumlah penghasil untuk waktu komputasi

Tests of Between-Subjects Effects

Dependent Variable: Y

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	2043917888.643 ^a	35	58397653.961	3.375	.000
Intercept	93756780.371	1	93756780.371	5.419	.023
Jumlahpenghasil	166901281.531	5	33380256.306	1.929	.102
Jumlahkandidatgudang	79303048.865	2	39651524.432	2.292	.109
Metode	93345954.614	1	93345954.614	5.395	.023
Jumlahpenghasil *	349175255.958	10	34917525.596	2.018	.046
Jumlahkandidatgudang	166911856.665	5	33382371.333	1.929	.102
Jumlahpenghasil * Metode	79301093.548	2	39650546.774	2.292	.109
Jumlahkandidatgudang *	349154643.104	10	34915464.310	2.018	.046
Metode	1107284746.667	64	17301324.167		
Error	3261950897.000	100			
Total	3151202635.310	99			
Corrected Total					

a. R Squared = .649 (Adjusted R Squared = .456)

Berdasarkan Tabel 4, terlihat bahwa nilai signifikansi jumlah penghasil bernilai $0,102 < 0,05$. Hal ini menjelaskan bahwa jumlah penghasil tidak memiliki pengaruh terhadap waktu komputasi yang digunakan. Hal ini mungkin terjadi karena masih kurangnya data untuk menjelaskan hubungan tersebut. metode pencarian solusi yang digunakan berpengaruh signifikan terhadap waktu komputasi yang digunakan karena nilai signifikansi metode bernilai $0,023 < 0,05$. Hal ini menjelaskan bahwa waktu komputasi oleh metode analitik berbeda dengan waktu komputasi oleh algoritma. nilai signifikansi jumlah kandidat gudang bernilai $0,109 > 0,05$. Hal ini menjelaskan bahwa jumlah kandidat gudang tidak berpengaruh signifikan terhadap solusi yang dihasilkan, yang berarti tidak ada perbedaan waktu komputasi antara jumlah kandidat gudang. Nilai signifikansi jumlah penghasil dan jumlah kandidat gudang bernilai $0,046 < 0,05$ yang berarti kedua factor tersebut memiliki interaksi terhadap waktu komputasi yang digunakan. Nilai signifikansi metode pencarian dan jumlah penghasil, serta metode pencarian dan jumlah kandidat gudang bernilai $> 0,05$ yang berarti semuanya tidak memiliki interaksi terhadap waktu komputasi. Sedangkan nilai signifikansi dari ketiga factor tersebut bernilai $0,046 < 0,05$ yang berarti ketiga factor tersebut memiliki interaksi atau pengaruh terhadap waktu komputasi.

4. Kesimpulan

Algoritma *Simulated Annealing* yang dikembangkan telah menghasilkan nilai fungsi objektif yang mendekati solusi global optimal yang dihasilkan oleh metode analitik dengan perbandingan sebesar 2,92%. Selain itu, waktu komputasi dari algoritma yang dikembangkan juga jauh lebih cepat dibanding waktu komputasi metode analitik, yaitu sebesar 98586,06%.

Daftar Pustaka

- [1] C. Prodhon dan C. Prins, "A survey of recent research on location-routing problems," *Eur. J. Oper. Res.*, vol. 238, no. 1, hal. 1–17, 2014.
- [2] G. Nagy, S. Salhi, G. Nagy, dan S. Salhi, "Location-routing : Issues , models and methods *," no. 1998, hal. 1–27.
- [3] H. Min, W. Patterson, dan A. F. B. Oh, "Combined location-routing problems : A synthesis

- and future research directions,” vol. 2217, no. 97, 1998.
- [4] R. B. Lopes, S. Barreto, C. Ferreira, dan B. S. Santos, “A decision-support tool for a capacitated location-routing problem,” *Decis. Support Syst.*, vol. 46, no. 1, hal. 366–375, 2008.
- [5] A. M. F. Achmad dan A. P. Enggita, “Model Matematis Penentuan Lokasi Fasilitas Dan Rute Kendaraan Dengan Mempertimbangkan Jendela Waktu Dan Kendaraan Heterogen Terbatas Pada Sistem Pengelolaan Limbah.” JAIER, Makassar, 2022.
- [6] C. Contardo, J. F. Cordeau, dan B. Gendron, *A GRASP + ILP-based metaheuristic for the capacitated location-routing problem*, vol. 20, no. 1. 2014.
- [7] C. J. Ting dan C. H. Chen, “A multiple ant colony optimization algorithm for the capacitated location routing problem,” *Int. J. Prod. Econ.*, vol. 141, no. 1, hal. 34–44, 2013.
- [8] V. C. Hemmelmayr, J. F. Cordeau, dan T. G. Crainic, “An adaptive large neighborhood search heuristic for Two-Echelon Vehicle Routing Problems arising in city logistics,” *Comput. Oper. Res.*, vol. 39, no. 12, hal. 3215–3228, 2012.
- [9] V. F. Yu, S. W. Lin, W. Lee, dan C. J. Ting, “A simulated annealing heuristic for the capacitated location routing problem,” *Comput. Ind. Eng.*, vol. 58, no. 2, hal. 288–299, 2010.
- [10] P. C. Pop, I. Zelina, V. Lupse, C. P. Sitar, dan C. Chira, “Heuristic algorithms for solving the generalized vehicle routing problem,” *Int. J. Comput. Commun. Control*, vol. 6, no. 1, hal. 158–165, 2011.
- [11] K. Pichka, A. H. Bajgiran, M. E. H. Petering, J. Jang, dan X. Yue, “The two echelon open location routing problem: Mathematical model and hybrid heuristic,” *Comput. Ind. Eng.*, vol. 121, no. August 2017, hal. 97–112, 2018.