J. Ris. & Ap. Mat. Vol. 6, No. 1 (2022), pp. 13–22.
Jurnal Riset dan Aplikasi Matematika
e-ISSN 2581-0154
URL: journal.unesa.ac.id/index.php/jram

PERBANDINGAN SOLUSI *CVRP* PADA DISTRIBUSI BUKU AQILA DI SURAKARTA MENGGUNAKAN ALGORITME *TABU SEARCH* DAN ALGORITME *ACO*

RAMA ADITYA PRATAMA[1*], PUTRANTO HADI UTOMO[2], SUPRIYADI WIBOWO[3]

[1,2,3]**Program Studi Matematika, Fakultas MIPA, Universitas Sebelas Maret**

***ramaadipra@student.uns.ac.id**

ABSTRAK

Capacitated Vehicle Routing Problem (*CVRP*) merupakan masalah permintaan transportasi yang mempunyai kendala berupa kapasitas kendaraan. PT. Tiga Serangkai Pustaka Mandiri merupakan sebuah unit bisnis yang bergerak di bidang penerbitan, percetakan, dan distribusi buku sekolah serta buku umum. Perusahaan ini memiliki jaringan distribusi yang luas. Agar proses ini efisien diperlukan optimasi agar total biaya distribusi minimum yaitu dengan menentukan rute optimal. Pada artikel ini, digunakan algoritme *Tabu Search* dan algoritme *Ant Colony Optimization* (*ACO*) untuk mengoptimalkan rute. Solusi CVRP pada distribusi buku Aqila di Surakarta ditentukan dengan membuat program dalam bahasa pemrograman Python berdasarkan algoritme tersebut. Berdasarkan hasil penelitian diperoleh solusi *CVRP* pada distribusi buku Aqila di Surakarta menggunakan algoritme *Tabu Search* menghasilkan rute dengan jarak total sebesar 259.7 km dalam waktu 81.73 detik dan menggunakan algoritme *ACO* menghasilkan rute dengan jarak total sebesar 261.8 km dalam waktu 72.46 detik. Algoritme *Tabu Search* menghasilkan jarak yang lebih baik dibandingkan algoritme *ACO* dengan selisih sebesar 0.8%. Algoritme *ACO* memiliki waktu eksekusi yang lebih cepat dengan selisih sebesar 112%.

Kata kunci: Buku Aqila, *CVRP*, *Tabu Search*, *Ant Colony Optimization*.

ABSTRACT

Capacitated Vehicle Routing Problem (*CVRP*) is a transportation demand problem with vehicle capacity constraint. PT. Tiga Serangkai Pustaka Mandiri is a business unit engaged in the publishing, printing and distribution of school and general books. This company has an extensive distribution network. In order for this process to be efficient, optimization is required to find the optimal route in order to minimize the distribution cost. The CVRP solution for the distribution of the "Aqila" book in Surakarta is determined by implementing several algorithm in Python. The result of this research obtained CVRP solution in the distribution of the "Aqila" book in Surakarta using the Tabu Search algorithm obtains a route with a total distance of 259.7 km in 81.73 seconds and using the ACO algorithm obtains a route with a total distance of 261.8 km in 72.46 seconds. The Tabu Search algorithm produces a better distance compared to the ACO algorithm with a gap of 0.8%. The ACO algorithm has a faster execution time with a gap of 112%.

Keywords: "Aqila" book, CVRP, Tabu Search, Ant Colony Optimization.

2010 Mathematics Subject Classification: 90C27, 90C11.
Tanggal masuk: 24-08-21, direvisi: 25-03-22, dimuat: 30-04-22.

1 Pendahuluan

Alat transportasi merupakan salah satu bagian penting dari aktivitas yang dilakukan manusia dalam kehidupan sehari-hari. Salah satunya adalah kegiatan pendistribusian barang. Dalam hal ini, aspek penting yang perlu diperhatikan adalah bagaimana cara mendistribusikan barang dengan tujuan mengoptimalkan waktu dan jarak tempuh sehingga dapat meminimumkan biaya distribusi. Menurut Sesa [1], membentuk rute optimal merupakan salah satu cara untuk meminimumkan total biaya pendistribusian. Masalah yang sering muncul adalah bagaimana menentukan serangkaian rute kendaraan untuk memenuhi permintaan pelanggan menggunakan kendaraan yang tersedia dengan biaya minimum. Khususnya memilih kendaraan mana yang menangani permintaan sesuai dalam rute yang ditentukan sehingga setiap permintaan dapat terpenuhi. Masalah ini dikenal sebagai *Vehicle Routing Problem* (*VRP*). Variasi dari *VRP* adalah *Capacitated Vehicle Routing Problem* (*CVRP*) yang mempunyai kendala berupa kapasitas kendaraan.

CVRP dapat diselesaikan dengan menggunakan algoritme heuristik maupun algoritme metaheuristik. Menurut Respati *et al*. [2] algoritme heuristik merupakan algoritme yang dirancang untuk memecahkan suatu permasalahan dalam pencarian solusi dan digunakan untuk menemukan suatu solusi yang dapat dibuktikan kebenarannya. Algoritme heuristik bersifat *problem dependent*, artinya algoritme ini hanya bisa diterapkan untuk jenis permasalahan tertentu. Contoh dari algoritme heuristik untuk *VRP* adalah *Clarke and Wright Saving*, *Nearest Neighbor*, dan algoritme Petal. Algoritme metaheuristik merupakan algoritme yang didasarkan pada algoritme heuristik. Algoritme ini digunakan untuk melakukan eksplorasi yang lebih dalam pada daerah yang menghasilkan ruang solusi yang ada. Algoritme metaheuristik bersifat *problem independent*, artinya algoritme ini tidak bergantung pada jenis permasalahan dan dapat digunakan untuk berbagai permasalahan.Contoh dari algoritme metaheuristik adalah *Simulated Annealing*, *Neural Networks*, *Deterministic Annealing*, *Ant Colony Optimization*, *Tabu Search*, dan algoritme Genetik.

Vehicle Routing Problem pertama kali diteliti oleh Dantzig dan Ramser [3] pada tahun 1959 dalam penelitiannya untuk memodelkan armada truk homogen dalam melayani permintaan minyak dengan jarak minimum. Penelitian ini kemudian menjadi dasar dari *CVRP*. Pada tahun 1964, Clarke dan Wright [4] memperbaiki model ini dengan pendekatan *greedy* yang lebih efektif yaitu algoritme *Saving*. Dalam perkembangannya yang sejalan dengan kompleksitas kehidupan nyata, muncul beberapa variasi dari *VRP*. Pada tahun 2012, Pradhana [5] menggunakan algoritme *Tabu Search* untuk penyelesaian *VRP* di perusahaan IT COMM dan memperoleh hasil rute yang lebih optimal dibandingkan dengan rute awal. Di tahun yang sama, Gunawan *et al*. [6] menggunakan algoritme *Nearest Neighbor*, *Brute Force*, dan *ACO* untuk menentukan rute optimum. Pada penelitian tersebut diperoleh algoritme *ACO* dapat menghasilkan solusi yang lebih mendekati solusi optimal. Pada tahun 2019, Silalahi [7] membandingkan algoritme eksak dan algoritme *ACO* untuk menentukan rute *Traveling Salesman Problem* dan diperoleh waktu eksekusi yang dihasilkan algoritme *ACO* lebih cepat daripada algoritme eksak.

PT. Tiga Serangkai Pustaka Mandiri merupakan sebuah unit bisnis dari Tiga Serangkai Group yang bergerak di bidang penerbitan, percetakan, dan distribusi buku sekolah serta buku umum. Salah satu jenis buku yang diproduksi adalah buku Aqila yang merupakan buku teks sekolah mata pelajaran berbasis Islam untuk siswa madrasah (MI, MTS, MA). Perusahaan ini memiliki jaringan distribusi yang luas. Agar proses ini efisien diperlukan optimasi untuk meminimumkan biaya yaitu dengan menentukan rute optimal. Berbeda dengan penelitian yang dilakukan oleh Pradhana [5] dan Silalahi [7], penelitian dalam artikel ini merupakan bentuk pengembangan penelitian dengan variasi *VRP* yaitu *CVRP* dan mengganti objek pengamatan yaitu pada pendistribusian buku Aqila terbitan PT Tiga Serangkai Pustaka Mandiri di Surakarta. Penelitian ini juga berbeda penelitian yang dilakukan oleh Gunawan *et al*. [6] dan Silalahi [7]. Kedua peneliti tersebut menggunakan data hipotetik yang diambil secara acak untuk menyelesaikan permasalahannya sedangkan pada penelitian ini digunakan data empiris. Pada artikel ini, solusi *CVRP* ditentukan menggunakan dua algoritme metaheuristik yaitu

algoritme *Tabu Search* dan *ACO* dalam bahasa pemrograman Python.

2 Tinjauan Pustaka

2.1 *Capacitated Vehicle Routing Problem*

Masalah penentuan rute kendaraan pertama kali diperkenalkan oleh Dantzig dan Ramser [3] pada tahun 1959. *VRP* merupakan generalisasi dari *m-TSP*, terdapat sebanyak *m salesman* yang akan mengunjungi sejumlah kota tepat satu kali. Menurut Purwanti [8], konsep *TSP* dikembangkan melalui *VRP* dengan penambahan berupa penugasan pada kendaraan. Salah satu variasi dari *VRP* adalah *CVRP* yaitu masalah permintaan transportasi yang terdiri dari pendistribusian barang dari satu depot dinotasikan dengan *vertex* v_0, ke pelanggan yang dinotasikan sebagai himpunan dari *n vertex* lain, $N = \{v_1, v_2, \ldots, v_n\}$. Banyaknya permintaan yang harus dikirimkan ke pelanggan $v_i \in N$ dinotasikan dengan $q_i \geq 0$. Semua kendaraan diasumsikan homogen dengan kapasitas yaitu $Q > 0$. Kendaraan yang melayani pelanggan $S \subseteq N$ dimulai dari depot, melangkah sekali ke setiap pelanggan di S, lalu kembali lagi ke depot. Perjalanan yang dilakukan dari i ke j dikenakan biaya perjalanan C_{ij}.

Permasalahan ini dapat dimodelkan dalam graf berarah maupun graf tidak berarah. Misalkan $V = \{v_0, v_1, v_2, \ldots, v_n\}$ adalah himpunan *vertex*. Didefinisikan $q_0 := 0$ untuk depot. Pada kasus simetris, biaya perjalanan dari i ke j tidak bergantung pada arah dengan graf $G = (V,E)$ *complete* dan tak berarah, E merupakan himpunan *edge* $E = \{e = (i,j) = (j,i); i,j \in V, i \neq j\}$, dan biaya perjalanan C_{ij}, $i,j \in E$. Pada kasus asimetris, biaya perjalanan $C_{ij} \neq C_{ji}$ dengan graf $G = (V,A)$, A merupakan himpunan *arc* $A = \{(i,j) \in V \times V;\ i \neq j\}$. *CVRP* dapat direpresentasikan sebagai *weighted complete graph* $G = (V,E,C_{ij},q_i)$ atau digraf $G = (V,A,C_{ij},q_i)$. Rute untuk suatu kendaraan direpresentasikan oleh barisan *vertex*, sebagai contoh $r = (v_0, v_1, v_2, \ldots, v_s, v_{s+1})$ dengan $v_0 = v_{s+1}$ dan $S = \{v_1, v_2, \ldots, v_s\} \subseteq N$ pelanggan yang terkunjungi. $C(r) = \sum_{p=0}^{s} C_{v_p, v_{p+1}}$ adalah biaya perjalanan untuk rute kendaraan tersebut. Misal $S \subseteq V$ adalah subset dari himpunan *vertex*. Untuk graf tak berarah, didefinisikan *cut set* $\delta(S) = \{\{i,j\} \in E : i \in S, j \notin S\}$, dengan $E(S) = \{\{i,j\} \in E : i,j \in S\}$ adalah himpunan dengan tepat satu atau kedua titik akhirnya di S. Pada graf berarah $G = (V,A)$, *arc* yang masuk ke S dilambangkan dengan $\delta^-(S) = \{(i,j) \in A : i \notin S, j \in S\}$ dan *arc* yang keluar dari S dilambangkan dengan $\delta^+(S) = \{(i,j) \in A : i \in S, j \notin S\}$. $A(S) = \{(i,j) \in A : i,j \in S\}$ adalah himpunan semua *arc* yang menghubungkan *vertex-vertex* di S. Untuk pelanggan $S \subseteq N$, misalkan $r(S)$ adalah jumlah kendaraan yang diperlukan untuk melayani S. $r(S)$ didefinisikan sebagai banyaknya permintaan dibagi dengan kapasitas kendaraan yaitu $\lceil q(s)/Q \rceil$. Model *CVRP* dalam bentuk *Mixed Integer Programming* yaitu:

$$\min \sum_{(i,j) \in A} C_{ij} X_{ij} \tag{1}$$

dengan kendala

$$\begin{aligned}
&\textstyle\sum_{j \in \delta^+(i)} X_{ij} = 1 && i \in N,\\
&\textstyle\sum_{i \in \delta^-(j)} X_{ij} = 1 && j \in N,\\
&\sum_{j \in \delta^+(0)} X_{0j} = |K|, &&\\
&\sum_{ij \in \delta^+(S)} X_{ij}(S) \geq r(S) && \forall S \in N, S \neq \emptyset,\\
&X_{ij} \in 0,1. &&
\end{aligned} \tag{2}$$

Fungsi objektif (1) bertujuan untuk meminimumkan total biaya perjalanan. Variabel keputusan X_{ij} pada kendala (2) didefinisikan sebagai berikut:

$$X_{ij} = \begin{cases} 1, & \text{jika kendaraan melakukan perjalanan dari } i \text{ ke } j \\ 0, & \text{lainnya.} \end{cases}$$

2.2 Algoritme *Tabu Search*

Tabu Search pertama kali diperkenalkan oleh Glover pada tahun 1986. Konsep dasar *Tabu Search* adalah menentukan solusi pada setiap iterasi agar dapat menemukan solusi terbaik tanpa terjebak kedalam solusi optimum lokal selama iterasi berlangsung. Menurut Ribeiro [9], algoritme *Tabu Search* dapat menuntun prosedur pencarian lokal heuristik untuk menjelajahi daerah solusi di luar titik optimum lokal. Ketika ditemui solusi optimum lokal, maka akan diarahkan untuk bergerak (*move*) ke solusi terbaik pada setiap *neighbour* walaupun mungkin akan terjadi penurunan nilai pada fungsi objektif. Pengarahan ini dilakukan dengan memanfaatkan suatu struktur memori yang mencatat sebagian jejak proses pencarian yang telah dilakukan.

Terdapat dua struktur memori pada algoritme ini yaitu *short term memory* dan *long term memory*. *Short term memory* terdiri dari *tabu list*, *tabu tenure* dan kriteria aspirasi, sedangkan *long term memory* digunakan untuk fase intensifikasi (*move* untuk menentukan solusi alternatif) dan diversifikasi (pengarahan *move* menuju solusi terbaik). Menurut Pradhana [5], terdapat dua teknik utama dalam diversifikasi. Teknik yang pertama adalah *restart diversification*, yaitu mengulang proses pencarian pada solusi alternatif untuk menemukan solusi yang berbeda dengan membangun solusi secara acak. Teknik kedua adalah *continuous diversification*, yaitu menggunakan informasi yang diperoleh dari proses pencarian pada solusi alternatif sebelumnya untuk menemukan *move* dengan jarak yang lebar. Langkah-langkah pada algoritme *Tabu Search* sebagai berikut:

1. Menentukan solusi awal dan menetapkannya sebagai solusi optimum.
2. Menentukan solusi alternatif yaitu dengan melakukan *move*.
3. Memilih solusi terbaik diantara solusi alternatif pada langkah 2. Jika solusi alternatif terdapat dalam *tabu list* maka solusi alternatif tidak akan dievaluasi kecuali memenuhi kriteria aspirasi.
4. Memilih nilai solusi optimum. Jika nilai solusi terbaik pada langkah 3 lebih kecil dari nilai solusi optimum sebelumnya maka solusi terbaik menjadi solusi optimum baru. Jika tidak, dilakukan proses diversifikasi.
5. Memperbarui *tabu list* dengan memasukkan solusi optimum baru.
6. Jika kriteria penghentian dipenuhi maka berhenti dan diperoleh solusi optimum. Jika tidak, kembali ke langkah 2.

2.3 Algoritme *Ant Colony Optimization*

Algoritme *Ant Colony Optimization* merupakan algoritme yang mengadopsi perilaku koloni semut dalam mencari makanan. Semut mampu mencari makanan dengan cara menemukan jarak terpendek antara sumber makanan dengan sarangnya tanpa menggunakan indra penglihatan. Semut berkomunikasi dengan mengeluarkan zat kimia disebut feromon. Jika semut menemukan sumber makanan, semut akan meninggalkan feromon pada rute saat kembali ke koloninya. Rute yang pendek akan mengandung feromon yang cukup padat dan mempengaruhi probabilitas semut lain mengikuti rute tersebut. Probabilitas ini dapat dihitung dengan rumus sebagai berikut:

$$P_{ij} = \frac{\tau_{ij}^{\alpha} \eta_{ij}^{\beta}}{\Sigma(\tau_{ij}^{\alpha})(\eta_{ij}^{\beta})} \tag{3}$$

P_{ij} : probabilitas semut melalui *edge* ij,
τ_{ij} : banyaknya feromon yang terdapat dalam *edge* ij,
dengan η_{ij} : invers jarak antara *edge* ij,
α : parameter pengendali feromon,
β : parameter pengendali jarak.

Selain feromon, faktor yang mempengaruhi semut dalam memilih rute adalah *visibility*. *Visibility* merupakan naluri semut untuk menemukan rute terdekat menuju sumber makanan ataupun kembali ke koloninya. Setiap semut memiliki *tabu list* yang berisi semua titik yang telah dikunjunginya pada setiap rute. Setelah semua semut menyelesaikan rutenya, maka dilakukan pembaruan feromon. Pembaruan ini dapat dihitung dengan rumus sebagai berikut:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \Delta\tau_{ij} \tag{4}$$

dengan
ρ : laju penguapan feromon,
$\Delta\tau_{ij}$: banyaknya feromon yang ditinggalkan pada *edge* ij yaitu

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k, & \text{jika semut ke-}k\text{ melakukan perjalanan dari } i \text{ ke } j \\ 0, & \text{lainnya.} \end{cases}$$

dengan Q adalah banyaknya rute yang terbentuk dan L_k adalah panjang rute semut ke-k. Langkah-langkah dalam algoritme *ACO* adalah sebagai berikut:

1. Inisialisasi parameter α, β, ρ, τ, jumlah semut, banyak iterasi, dan Q.
2. Meletakkan semut ke setiap pelanggan secara acak.
3. Untuk setiap semut,
   (a) Menghitung P_{ij} berdasarkan persamaan (3) dan memilih probabilitas terbesar untuk bergerak ke pelanggan berikutnya.
   (b) Menambahkan pelanggan yang dilalui ke dalam *tabu list* hingga mendapat solusi.
4. Memperbarui feromon berdasarkan persamaan (4) untuk setiap semut yang telah menyelesaikan rute.
5. Jika solusi lokal terbaik lebih baik daripada solusi global maka simpan solusi lokal terbaik sebagai solusi global.
6. Jika kriteria penghentian dipenuhi maka berhenti dan diperoleh solusi global sebagai solusi optimum. Jika tidak, kembali ke langkah 3.

3 Metode

Metode penelitian yang digunakan dalam penelitian ini adalah studi literatur dan studi lapangan. Studi literatur dilakukan dengan mempelajari referensi berupa buku, jurnal, skripsi, dan tesis mengenai algoritme metaheuristik dan *VRP*. Studi lapangan dilakukan pada proses pengambilan data berupa data permintaan buku Aqila di Surakarta dan data lokasi pelanggan. Algoritme yang dipilih untuk menyelesaikan *CVRP* dieksekusi menggunakan bantuan laptop dengan spesifikasi Intel Celeron N3160 110-14IBR CPU @ 1.6 GHz, RAM 2GB. Langkah-langkah penelitian yang dilakukan dalam artikel ini sebagai berikut:

1. Melakukan pengambilan data permintaan buku Aqila di Surakarta dan data lokasi pelanggan.
2. Menentukan asumsi yang digunakan.

3. Menentukan solusi eksak *CVRP* menggunakan *Mixed Integer Programming*, solusi pendekatan menggunakan algoritme *Tabu Search* dan algoritme *ACO* dengan program dalam bahasa pemrograman Python.
4. Membandingkan solusi pendekatan yang diperoleh menggunakan algoritme tersebut terhadap solusi eksak, waktu eksekusi, dan jarak total yang dihasilkan.

4 Hasil dan Pembahasan

4.1 Profil Sistem Distribusi Buku Aqila di Surakarta dan Asumsi Model

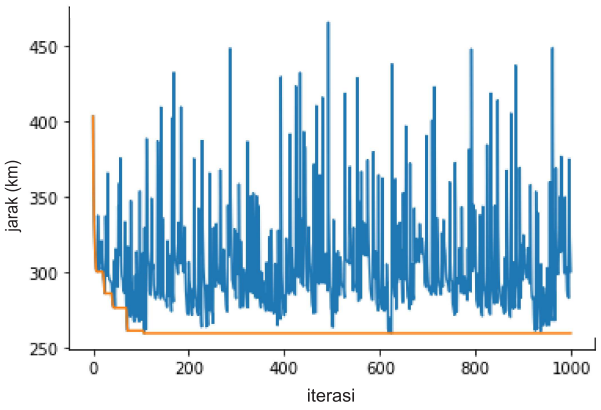
Proses pendistribusian buku Aqila dari gudang percetakan ke setiap daerah melalui kantor cabang di tiap kota. Mobil boks yang digunakan dapat memuat hingga 580 eksemplar buku. Data pelanggan yang digunakan pada penelitian ini adalah data pelanggan SMP/MTS. Terdapat sebanyak 19 SMP/MTS di Surakarta yang menjadi pelanggan buku Aqila. Untuk membatasi permasalahan pada artikel ini, digunakan asumsi sebagai berikut:

1. Seluruh permintaaan pelanggan dapat dilayani oleh perusahaan.
2. Setiap buku yang didistribusikan memiliki tebal yang sama.
3. Kendaraan selalu dalam keadaan baik.
4. Kondisi jalan tidak rusak dan tidak ada kemacetan.
5. Jalan yang dipilih untuk melayani pelanggan adalah jalan terdekat dan dapat dilalui oleh kendaraan pengangkut buku.
6. Setiap pelanggan dan depot saling *connected*, terdapat jarak antar pelanggan yang tidak simetris.

4.2 Solusi Pendekatan Menggunakan Algoritme *Tabu Search*

Dalam menentukan rute *CVRP* menggunakan algoritme *Tabu Search*, terlebih dahulu dilakukan inisialisasi rute yaitu dengan membentuk rute yang *feasible*. Rute awal yang diperoleh kemudian digabungkan ke dalam satu rute kendaraan untuk dilakukan *move*. Pada artikel ini, *move* yang digunakan adalah *move* 2-*opt*. Proses pada *move* ini yaitu menukar dua titik pada suatu rute. *Move* ini menghasilkan *value* yaitu berupa besarnya pengaruh *move* terhadap jarak dari rute awal. Selanjutnya menentukan *tabu list* dan *tabu contour* yang masih berupa himpunan kosong. Selain itu, ditentukan *tabu tenure* dengan nilai maksimal 3. Semua *move* yang telah dilakukan disimpan dalam *tabu contour*, kemudian dicari hasil terbaik dari setiap iterasi untuk disimpan dalam *tabu list*. Teknik diversifikasi yang digunakan adalah *restart diversification*. Kriteria penghentian yang digunakan adalah batasan iterasi yaitu sebanyak 1000 iterasi.

Berdasarkan algoritme *Tabu Search*, dibuat program dalam bahasa pemrograman Python dengan input jarak, permintaan, dan kapasitas kendaraan. Jarak yang dihasilkan pada setiap iterasi (warna biru) dan solusi terbaik pada setiap iterasi (warna oranye) terlihat pada Gambar 1. Berdasarkan Gambar 1, ditentukan solusi awal dengan membangun solusi secara acak. Solusi bergerak pada setiap iterasi dengan melakukan *move* membentuk solusi alternatif. Pada iterasi ke 108 didapatkan solusi optimum lokal yang bertahan hingga batas iterasi maksimal, sehingga solusi tersebut dipilih menjadi solusi optimum global. Diperoleh rute pendistribusian dengan nilai minimum pada iterasi ke 108 dengan total jarak sebesar 259.7 km. Rute pendistribusian buku Aqila di Surakarta menggunakan algoritme *Tabu Search* terlihat pada Tabel 1.



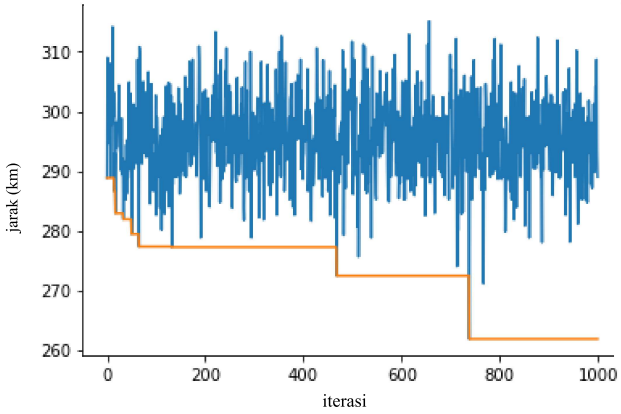
Gambar 1: Grafik jarak setiap iterasi pada algoritme *Tabu Search*

Tabel 1: Rute pendistribusian buku aqila dengan algoritme *Tabu Search*

1	Depot → MTS Assalam Pabelan → MTSN Ngemplak → MTSN Temon → MTS Muh 1 Gondangrejo → MTS Muh Surakarta → Depot	49 km
2	Depot → MTSN Karangmojo → SMPIT Abu Jafar → SMPIT MTA Karanganyar → Depot	57 km
3	Depot → SMPI Cendekia Al Mujtaba → SMPIT Mutiara Insan → SMPIT Mardhotilah → SMP Darussalam Serengan → Depot	45.2 km
4	Depot → SMP Al Islam Kartasura → SMP Muh 3 Ampel → SMP Muh 8 Karanggede → SMP MTA Gemolong → MTSN Gondangrejo → Depot	101.4 km
5	Depot → MTS Nahdlatul Muslimat → SMP Al Irsyad → Depot	7.1 km

4.3 Solusi Pendekatan Menggunakan Algoritme *ACO*

Dalam menentukan rute *CVRP* menggunakan algoritme *ACO*, terlebih dahulu dilakukan inisialisasi parameter α, β, ρ, τ, jumlah semut, iterasi, dan Q. Pemilihan parameter mengacu pada Gaertner dan Clark [10] serta Wang dan Han [11] yang telah melakukan simulasi parameter pada beberapa *instance TSP*, khususnya pada parameter α, β, ρ, τ dengan memilih nilai parameter yang relatif menghasilkan solusi yang baik, diperoleh nilai parameter $\alpha = 1$, $\beta = 6$, $\rho = 0.4$, $\tau = 0.2$. Untuk parameter jumlah semut, iterasi, dan Q diambil berdasarkan data yang digunakan. Diperoleh nilai parameter jumlah semut= 19, iterasi= 1000, dan $Q = 5$. Berdasarkan algoritme *ACO*, dibuat program dalam bahasa pemrograman Python dengan input jarak, permintaan, dan kapasitas kendaraan. Jarak yang dihasilkan pada setiap iterasi (warna biru) dan solusi terbaik pada setiap iterasi (warna oranye) terlihat pada Gambar 2.



Gambar 2: Grafik jarak setiap iterasi pada algoritme *ACO*

Berdasarkan Gambar 2, iterasi diawali dengan proses semut membangun rute awal secara acak, kemudian dengan pengaruh parameter yang digunakan diperoleh solusi optimum lokal. Selanjutnya iterasi terus berjalan hingga batas iterasi yang telah ditentukan, diperoleh solusi optimum global berupa rute pendistribusian dengan jarak minimum pada iterasi ke 738 dengan total jarak sebesar 261.8 km. Rute pendistribusian buku Aqila di Surakarta menggunakan algoritme *ACO* terlihat pada Tabel 2.

Tabel 2: Rute pendistribusian buku aqila dengan algoritme *ACO*

No.	Rute	Jarak
1	Depot → SMP Darussalam Serengan → MTS Nahdlatul Muslimat → SMP Al Irsyad → MTS Muh Surakarta → Depot	11.1 km
2	Depot → SMPI Cendekia Al Mujtaba → SMPIT Mutiara Insan → SMPIT Mardhotilah → Depot	45.3 km
3	Depot → MTSN Karangmojo → SMPIT Abu Jafar → SMPIT MTA Karanganyar → Depot	57 km
4	Depot → MTSN Gondangrejo → SMP MTA Gemolong → SMP Muh 8 Karanggede → SMP Muh 3 Ampel → MTSN Ngemplak → Depot	99.4 km
5	Depot → MTS Assalam Pabelan → SMP Al Islam Kartasura → MTSN Temon → MTS Muh 1 Gondangrejo → Depot	49km

4.4 Perbandingan Solusi

Solusi yang diperoleh menggunakan dua algoritme tersebut dibandingkan berdasarkan solusi eksak, waktu eksekusi, dan jarak total yang dihasilkan. Solusi eksak menggunakan model *CVRP* pada subbagian 2.1 ditentukan dengan membuat program dalam bahasa pemrograman Python. *Package* Gurobi [12] digunakan dalam program yang dibuat. Diperoleh solusi eksak terlihat pada Tabel 3.

Tabel 3: Rute pendistribusian buku aqila dengan *Mixed Integer Programming*

No.	Rute	Jarak
1	Depot → MTS Muh Surakarta → SMP Al Irsyad → MTS Nahdlatul Muslimat → Depot	9.6 km
2	Depot → MTS Muh 1 Gondangrejo → MTSN Temon → MTSN Ngemplak → MTS Assalam Pabelan → Depot	45 km
3	Depot → MTSN Karangmojo → SMPIT Abu Jafar → SMPIT MTA Karanganyar → Depot	57 km
4	Depot → SMP Al Islam Kartasura → SMP Muh 3 Ampel → SMP Muh 8 Karanggede → SMP MTA Gemolong → MTSN Gondangrejo → Depot	101 km
5	Depot → SMPIT Mardhotilah → SMPIT Mutiara Insan → SMPI Cendekia Al Mujtaba → SMP Darussalam Serengan → Depot	45 km

Perbandingan jarak dan waktu eksekusi terlihat pada Tabel 4.

Tabel 4: Perbandingan jarak dan waktu eksekusi.

Algoritme	Jarak (km)	Jarak solusi eksak (km)	Selisih jarak	Waktu eksekusi	Waktu eksekusi solusi eksak	Selisih waktu
Tabu Search	259.7	257.6	0.8%	81.73 detik	200.96 detik	245.86%
ACO	261.8		1.6%	72.46 detik		277.33%
Selisih	0.8%	-	-	112%	-	-

Berdasarkan tabel tersebut terlihat bahwa waktu eksekusi pada algoritme *Tabu Search* dan algoritme *ACO* lebih cepat dibandingkan waktu eksekusi menggunakan *MIP* dengan selisih berturut-turut sebesar 245.86% dan 277.33%. Total jarak yang dihasilkan dari algoritme *Tabu Search* dan algoritme *ACO* mendekati optimal dengan selisih berturut-turut sebesar 0.8% dan 1.6%. Hal ini menunjukkan bahwa algoritme *Tabu Search* dan *ACO* dapat menjadi alternatif untuk menentukan solusi *CVRP* karena menghasilkan solusi yang mendekati optimal dan dalam waktu yang lebih cepat. Pada kasus ini algoritme *Tabu Search* menghasilkan jarak yang lebih baik dibandingkan algoritme *ACO* dengan selisih sebesar 0.8% namun algoritme *ACO* memiliki waktu eksekusi yang lebih cepat dengan selisih sebesar 112%.

5 Kesimpulan

Berdasarkan hasil dan pembahasan diperoleh kesimpulan yaitu diperoleh solusi *CVRP* pada distribusi buku Aqila di Surakarta menggunakan algoritme *Tabu Search* menghasilkan rute yang terlihat pada Tabel 1 dengan jarak total sebesar 259.7 km dan menggunakan algoritme *ACO* menghasilkan rute yang terlihat pada Tabel 2 dengan jarak total sebesar 261.8 km. Perbandingan solusi antara dua algoritme tersebut terlihat pada Tabel 4. Pada kasus ini, algoritme *Tabu Search* menghasilkan jarak yang lebih baik dibandingkan dengan algoritme *ACO* dengan selisih sebesar 0.8% namun algoritme *ACO* memiliki waktu eksekusi yang lebih cepat dengan selisih sebesar 112%.

Referensi

[1] F. J. Sesa, H. Syarifudin, and Y. Rizal, "Optimasi Rute Pengiriman Produk dengan Meminimumkan Biaya Transportasi Menggunakan Metode *Saving Matrix* di PT. DEF," *UNP: Journal of Mathematics*, vol. 4, no. 2, 2019.

[2] Y. A. Respati, E. Zaini, and A. Imran, "Model Penjadwalan *No-Wait Job Shop* Menggunakan Algoritme Variabel *Neighbourhood Descent* dengan *Threshold* untuk Meminimisasi *Makespan*," *REKA INTEGRA*, vol. 3, no. 1, 2015.

[3] G. B. Dantzig and J. J. Ramser, "The Truck Dispatching Problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.

[4] G. Clarke and J. W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," *Operation Research*, vol. 12, pp. 568–581, 1964.

[5] F. E. Pradhana, E. Sugiharti, and M. Kharis, "Penerapan Algoritme *Tabu Search* untuk Menyelesaikan *Vehicle Routing Problem*," *Unnes Journal of Mathematics*, vol. 1, no. 1, 2012.

[6] G. Gunawan, I. Maryati, and H. K. Wibowo, "Optimasi Penentuan Rute Kendaraan pada Sistem Distribusi Barang dengan *Ant Colony Optimization*," *Semantik*, vol. 2, no. 1, 2012.

[7] B. P. Silalahi, N. Fathiah, and P. Spriyo, "Use of Ant Colony Optimization Algorithm for Determining Traveling Salesman Problem Routes," *Jurnal Matematika (MANTIK)*, vol. 5, pp. 100–111, 2019.

[8] R. Purwanti, "Pemodelan Persoalan *Vehicle Routing* periodik," *URL http://www.repositori.usu.ac.id/handle/123456789/14085*, 2014.

[9] C. C. Ribeiro and M. G. Resende, "Path-relinking Intensification Methods for Stochastic Local Search Algorithms," *Journal of Heuristics*, vol. 18, no. 2, pp. 193–214, 2012.

[10] D. Gaertner and K. L. Clark, "On Optimal Parameters for Ant Colony Optimization Algorithms," in *IC-AI*, pp. 83–89, 2005.

[11] Y. Wang and Z. Han, "Ant Colony Optimization for Traveling Salesman Problem Based on Parameters Optimization," *Applied Soft Computing*, vol. 107, p. 107439, 2021.

[12] I. Gurobi, "Gurobi Optimizer Reference Manual, 2018," *URL http://www.gurobi.com*, 2018.