

Penyelesaian *Capacitated Vehicle Routing Problem* Menggunakan Pengembangan Algoritma *Clarke and Wright Savings*

Nur Muhammad Yusuf*, Sukoyo*

Program Studi Magister Teknik dan Manajemen Industri, Fakultas Teknologi Industri, Institut Teknologi Bandung; email: nmyusuf@itb.ac.id, sukoyo@itb.ac.id

* *Corresponding author*

Abstrak

Permasalahan penentuan rute distribusi merupakan suatu permasalahan yang harus diperhatikan agar biaya distribusi yang dihasilkan lebih rendah. Permasalahan ini dapat disebut sebagai *Vehicle Routing Problem (VRP)*. Salah satu varian VRP yang populer digunakan adalah *Capacitated Vehicle Routing Problem (CVRP)* yaitu VRP yang berkaitan dengan kendala kapasitas. CVRP ini dapat diselesaikan menggunakan algoritma *Clarke and Wright Savings*. Penyelesaian menggunakan algoritma ini didasari pada nilai penghematan dari dua titik pelanggan yang akan dilayani jika keduanya dilayani sekaligus dalam rute yang sama. Namun, algoritma ini hanya memperhatikan nilai penghematan dari kedua titik saja sehingga hasil algoritma hanya berupa kelompok rute dan perlu diperbaiki ulang urutannya menggunakan algoritma lain. Pengembangan dilakukan dengan mengubah cara penentuan rute berdasarkan nilai penghematan dengan menggunakan konsep *head and tail* serta menambahkan batasan waktu operasional distribusi. Pengembangan bertujuan agar hasil dari algoritma merupakan sebuah urutan rute yang tidak perlu diperbaiki ulang menggunakan algoritma lain. Algoritma yang telah dikembangkan menghasilkan solusi dengan perbaikan jarak tempuh sebesar 1,8% dan biaya sebesar 0,12% dari algoritma *Clarke and Wright Savings*. Selain itu, pengembangan algoritma mampu menghasilkan solusi dalam waktu 1 menit dan solusi yang dihasilkan lebih baik dibandingkan solver LINGO dengan waktu komputasi 74 jam.

Kata Kunci: *Vehicle Routing Problem, Capacitated Vehicle Routing Problem, algoritma Clarke and Wright Savings*

Abstract

[Solving the Capacitated Vehicle Routing Problem Using the Development of Clarke and Wright Savings Algorithm] The problem of determining distribution routes is a problem that must be considered so that the resulting distribution costs are lower. This problem can be referred to as a *Vehicle Routing Problem (VRP)*. One variant of VRP that is popularly used is the *Capacitated Vehicle Routing Problem (CVRP)*, which is VRP related to capacity constraints. This CVRP can be solved using the *Clarke and Wright Savings* algorithm. Settlement using this algorithm is based on the value of savings from two customer points that will be served if both are served at the same time on the same route. However, this algorithm only pays attention to the savings value of both points so that the algorithm results are only groups of routes and need to be re-corrected using another algorithm. Development is carried out by changing the way of routing based on savings value using the *head and tail* concept and adding distribution operational time limits. Development aims to make the result of the algorithm a sequence of routes that do not need to be corrected using other algorithms. The algorithm that has been developed resulted in a solution with mileage improvements of 1.8% and costs of 0.12% from the *Clarke and Wright Savings* algorithm. In addition, algorithm development is able to produce solutions within 1 minute and the resulting solution is better than the LINGO solver with a computational time of 74 hours.

Keywords: *Vehicle Routing Problem, Capacitated Vehicle Routing Problem, Clarke and Wright Savings algorithm*

Kelompok BoK yang bersesuaian dengan artikel: *Supply Chain Management*

Saran format untuk mensitasi artikel ini:

Yusuf, N.M. dan Sukoyo (2023). Penyelesaian *Capacitated Vehicle Routing Problem* Menggunakan Pengembangan Algoritma *Clarke and Wright Savings*. *Prosiding Seminar Nasional Teknik Industri (SENASTI) 2023*, 628-638.

1. Pendahuluan

Permasalahan distribusi suatu produk dari depot ke sejumlah pelanggan merupakan salah satu permasalahan yang kompleks. Semakin banyaknya pelanggan menyebabkan jumlah jaringan yang terbentuk menjadi semakin meningkat secara polinomial. Produk didistribusikan kepada pelanggan sesuai dengan permintaan kebutuhan masing-masing pelanggan. Distribusi produk kepada pelanggan harus ditentukan sebaik mungkin sehingga biaya yang dikeluarkan akan lebih murah. Biaya yang lebih murah dapat direpresentasikan dengan jumlah kendaraan yang sedikit dan jarak yang ditempuh merupakan jarak yang pendek. Masalah penentuan rute disebut sebagai *Vehicle Routing Problem (VRP)*.

Menurut Toth dan Vigo (2002), VRP merupakan suatu istilah umum yang digunakan pada proses penentuan sekumpulan rute kendaraan untuk melayani permintaan pelanggan dengan biaya operasional yang optimal. VRP pertama kali dipelajari oleh Dantzig dan Ramser pada tahun 1959 dengan sebutan *Vehicle Dispatching* untuk rute penjadwalan truk. Kemudian VRP dilanjutkan oleh Clarke dan Wright pada tahun 1964 dengan sebutan *Vehicle Scheduling Problem (VSP)*. VRP berkembang menjadi varian-varian baru dan dijabarkan oleh Lin dkk. (2014). Salah satu varian yang sangat populer digunakan adalah *Capacitated Vehicle Routing Problem (CVRP)*. CVRP merupakan varian VRP yang mempertimbangkan kapasitas kendaraan. CVRP ini merupakan salah satu bagian dari masalah optimasi sehingga dalam penyelesaiannya dapat digunakan beberapa pendekatan seperti pendekatan eksak, heuristik, dan metaheuristik.

Pendekatan heuristik lebih cocok digunakan ketika masalah yang dihadapi merupakan masalah yang sangat kompleks dan pendekatan eksak tidak mampu untuk digunakan untuk menyelesaikan secara cepat. Pendekatan heuristik menghasilkan solusi yang cukup baik dalam waktu yang lebih singkat namun tidak ada jaminan akan seoptimal hasil solusi dari pendekatan eksak. Salah satu algoritma heuristik yang dapat digunakan dalam penyelesaian CVRP adalah algoritma *Clarke and Wright Savings* yang ditemukan oleh Clarke dan Wright pada tahun 1964. Dasar penentuan rute yang digunakan pada algoritma *Clarke and Wright Savings* adalah nilai penghematan (*saving value*) jika dua pelanggan dilayani sekaligus.

Yuliza dkk. (2020) menggunakan algoritma *Clarke and Wright Savings* dan *solver LINGO* untuk menyelesaikan permasalahan distribusi LPG kemudian membandingkan hasil kedua metode tersebut. Meilani dan Iswara (2018) juga menyelesaikan penentuan rute distribusi produk LPG menggunakan algoritma *Clarke and Wright Savings* dan memperbaikinya dengan algoritma *Branch and Bound*. Marpaung dkk. (2022) juga menggunakan algoritma *Clarke and Wright Savings* untuk mengelompokkan rute distribusi produk kaca, kemudian memperbaiki

urutan rute dengan algoritma *Nearest Insert*, *Nearest Neighbor*, dan *Farthest Insert*, serta membandingkan hasil ketiga metode tersebut. Irman dkk. (2017) dan Arifudin dkk. (2017) menggunakan algoritma *Clarke and Wright Savings* untuk menyelesaikan penentuan rute distribusi produk makanan dan membandingkan dengan metode *saving matrix*. Algoritma *Clarke and Wright Savings* menghasilkan solusi yang lebih baik dibandingkan dengan *saving matrix*.

Rohandi dkk. (2014), Octora dkk. (2014), dan Rupiah dkk. (2017) menggunakan algoritma *Clarke and Wright Savings* dan *Sequential Insertion* untuk menyelesaikan penentuan rute distribusi masing-masing produk dan membandingkan hasil kedua metode. Hasilnya menunjukkan bahwa algoritma *Sequential Insertion* lebih baik dibandingkan dengan algoritma *Clarke and Wright Savings*. Algoritma *Clarke and Wright Savings* juga digunakan oleh Kurniawan dkk. (2014) untuk menyelesaikan penentuan rute distribusi produk air mineral dalam kemasan. Purnomo (2010); Fuadi dan Pujotomo (2018) menggunakan algoritma *Clarke and Wright Savings* untuk menyelesaikan permasalahan penentuan rute distribusi produk minuman dan menghasilkan solusi yang lebih baik dibandingkan kondisi eksisting.

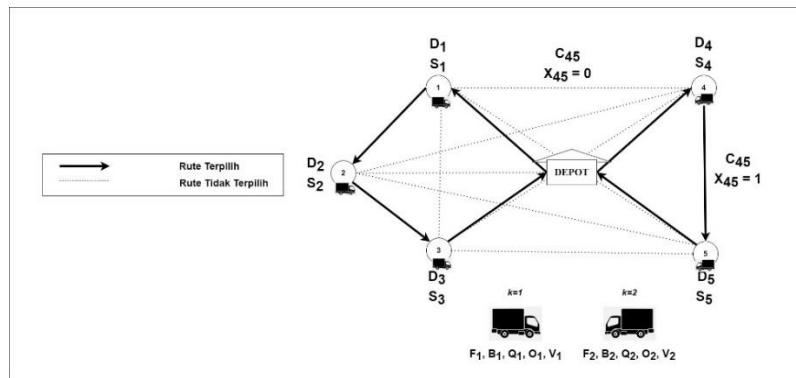
Algoritma *Clarke and Wright Savings* yang ditemukan oleh Clarke dan Wright dapat digunakan untuk menyelesaikan penentuan rute distribusi dalam VRP. Namun, solusi yang didapatkan merupakan sebuah kelompok rute yang perlu diperbaiki ulang dengan algoritma lain. Selain itu, algoritma *Clarke and Wright Savings* hanya mempertimbangkan kendala kapasitas, sedangkan dalam penerapannya perlu dipertimbangkan juga kendala waktu operasionalnya. Maka dari itu, tujuan dari penelitian ini adalah mengembangkan algoritma *Clarke and Wright Savings* dengan mengubah cara pemilihan rutenya menggunakan konsep *head* dan *tail* sehingga solusi yang dihasilkan merupakan urutan rute serta menambahkan kendala waktu operasional distribusi yang terdiri dari waktu perjalanan dan waktu pelayanan. Hasil akan dibandingkan dengan solusi algoritma acuan *Clarke and Wright Savings* dan solusi hasil pendekatan eksak menggunakan *solver* LINGO.

2. Metode

Metode yang digunakan pada penelitian ini adalah (1) studi literatur, (2) perumusan model matematis, (3) penyelesaian masalah menggunakan algoritma acuan *Clarke and Wright Savings* (*heuristic*), (4) penyelesaian masalah menggunakan *solver* LINGO (*exact*), (5) penyelesaian masalah menggunakan hasil pengembangan algoritma (*heuristic*), (6) penarikan kesimpulan.

2.1. Karakteristik sistem

Ilustrasi sistem yang digunakan dalam penelitian ini dapat dilihat pada Gambar 1. Sejumlah permintaan pelanggan dilayani oleh beberapa kendaraan dari depot. Setiap kendaraan akan melayani satu rute pelanggan dan akan kembali ke depot. Setiap kali kendaraan digunakan akan ada biaya tetap yang harus dikeluarkan dan biaya variabel dipengaruhi oleh variabel jarak. Setiap kendaraan memiliki kapasitas, kecepatan rata-rata, dan batas waktu operasional distribusi masing-masing. Waktu distribusi ditentukan dari waktu perjalanan dan waktu pelayanan masing-masing pelanggan.



Gambar 1. Ilustrasi permasalahan distribusi

2.2. Model matematis

Model matematika dikembangkan berdasarkan permasalahan yang diselesaikan. Notasi model matematika ditunjukkan pada Tabel 1.

Tabel 1. Formulasi model matematis

Notasi	Keterangan	Satuan
Himpunan		
N	Himpunan <i>node</i> . $N_d = \{0\}$ untuk <i>node</i> depot. $N_c = \{1,2,3, \dots, n\}$ untuk himpunan <i>node</i> pelanggan.	-
K	Himpunan kendaraan $K = \{1,2, \dots, k\}$	-
Indeks		
i, j	Indeks <i>node</i>	-
k	Indeks kendaraan	-
Parameter		
F_k	Biaya tetap kendaraan k	Rp
B_k	Biaya variabel kendaraan k	Rp/km
C_{ij}	Jarak tempuh dari <i>node</i> i ke <i>node</i> j	km
D_i	Permintaan <i>node</i> i	kg
Q_k	Kapasitas kendaraan k	kg
O_k	Waktu operasional distribusi kendaraan k	jam
V_k	Kecepatan kendaraan kendaraan k	km/jam
S_i	Waktu pelayanan di <i>node</i> i	jam
M	Big M	-
Variabel Keputusan		
X_{ij}^k	$\begin{cases} 1, & \text{jika kendaraan } k \text{ melakukan perjalanan dari node } i \text{ ke } j \\ 0, & \text{jika kendaraan } k \text{ tidak melakukan perjalanan dari node } i \text{ ke } j \end{cases}$	-
T_i	Waktu kedatangan di <i>node</i> i	jam ke-

Fungsi objektif dari penelitian ini adalah meminimasi biaya distribusi yang terdiri dari biaya tetap dan biaya variabel kendaraan distribusi. Biaya tetap kendaraan dikeluarkan jika suatu kendaraan digunakan dan akan bernilai 0 jika tidak digunakan. Biaya variabel kendaraan dikeluarkan berdasarkan variabel yang berkaitan dengan kendaraan. Pada permasalahan ini, biaya tetap diestimasi dari upah sopir dan kenek kendaraan, sedangkan biaya variabel kendaraan diestimasi dari biaya bahan bakar yang digunakan kendaraan per satuan jarak. Berdasarkan kedua komponen biaya tersebut maka didapatkan fungsi objektif pada Persamaan (1).

$$\text{Min } Z = \sum_{k \in K} \sum_{j \in N} \sum_{i \in N_d} F_k \times X_{ij}^k + \sum_{k \in K} \sum_{j \in N} \sum_{i \in N} B_k \times C_{ij} \times X_{ij}^k \quad (1)$$

dengan pembatas:

$$\sum_{k \in K} \sum_{j \in N, j \neq i} X_{ij}^k = 1 \quad \forall i \in N_c \quad (2)$$

$$\sum_{i \in N, i \neq h} X_{ih}^k = \sum_{j \in N, j \neq h} X_{hj}^k \quad \forall h \in N, \forall k \in K \quad (3)$$

$$\sum_{j \in N_c} \sum_{i \in N_d} X_{ij}^k = 1 \quad \forall k \in K \quad (4)$$

$$T_j \geq \frac{C_{ij}}{V_k} \times X_{ij}^k \quad \forall i \in N_d, \forall j \in N_c, \forall k \in K, i \neq j \quad (5)$$

$$T_j \geq T_i + S_i + \frac{C_{ij}}{V_k} - M(1 - X_{ij}^k) \quad \forall i \in N_c, \forall j \in N_c, \forall k \in K, i \neq j \quad (6)$$

$$\sum_{j \in N_c} \sum_{i \in N, i \neq j} d_j \times X_{ij}^k \leq Q_k \quad \forall k \in K \quad (7)$$

$$\sum_{j \in N} \sum_{i \in N} \left(\frac{C_{ij}}{V_k} + S_j \right) \times X_{ij}^k \leq O_k \quad \forall k \in K \quad (8)$$

$$X_{ij}^k \in \{0,1\} \quad \forall i, j \in N, \forall k \in K \quad (9)$$

Fungsi objektif pada Persamaan (1) bertujuan untuk meminimasi biaya tetap kendaraan dan biaya variabel kendaraan. Pembatas pada Persamaan (2) memastikan bahwa satu pelanggan hanya dikunjungi satu kali oleh kendaraan. Pembatas pada Persamaan (3) memastikan bahwa pelanggan dikunjungi dan ditinggalkan oleh kendaraan yang sama. Pembatas pada Persamaan (4) memastikan bahwa seluruh kendaraan melakukan perjalanan dari depot. Pembatas pada Persamaan (5) dan (6) memastikan waktu tiba kendaraan di pelanggan pertama dan waktu tiba di pelanggan berikutnya sekaligus memastikan bahwa tidak ada sub-rute dalam kendaraan. Pembatas pada Persamaan (7) memastikan bahwa jumlah permintaan yang diangkut oleh kendaraan tidak melebihi kapasitas kendaraan. Pembatas pada Persamaan (8) memastikan bahwa waktu distribusi total tidak melebihi waktu operasional distribusi kendaraan. Pembatas pada Persamaan (9) memastikan bahwa variabel keputusan bersifat biner 0 dan 1.

2.3. Pengembangan algoritma acuan *Clarke and Wright Savings*

Algoritma acuan *Clarke and Wright Savings* merupakan salah satu algoritma penyelesaian CVRP. Namun, algoritma ini hanya mempertimbangkan kendala kapasitas kendaraan saja, sedangkan pada kondisi nyata terdapat batasan waktu distribusi kendaraan yang dibatasi oleh jam kerja. Selain itu, dalam memilih urutan pelanggan yang dilayani, algoritma *Clarke and Wright Savings* hanya mempertimbangkan nilai *savings* dari dua titik saja dan hasilnya merupakan pengelompokan pelanggan yang dilayani sehingga hasil tersebut perlu diperbaiki kembali dengan algoritma-algoritma lain. Maka dari itu, pengembangan algoritma dilakukan dengan menambah kendala waktu distribusi yang mempertimbangkan waktu perjalanan dan waktu layanan serta mengubah cara pemilihan urutan pelanggan yang akan dilayani menggunakan konsep *head* dan *tail*.

Penyelesaian menggunakan pengembangan algoritma acuan *Clarke and Wright Savings* terdapat 5 tahapan umum yaitu menentukan matriks jarak, menentukan *saving matrix*, menentukan *head* dan *tail* awal, menentukan *head* dan *tail* baru, dan menentukan rute. Secara detail, 5 tahapan tersebut akan dijelaskan beserta contohnya sebagai berikut.

1. Menentukan matriks jarak

Pada tahap ini, jarak antara depot dan pelanggan maupun jarak antar pelanggan ditentukan menggunakan bantuan *Google Maps* sehingga jarak yang dihasilkan akan lebih sesuai dengan kondisi riil yang sudah mempertimbangkan ketersediaan akses jalan. Contoh matriks jarak yang dapat dilihat pada Tabel 2.

Tabel 2. Matriks jarak

Node	Depot	1	2	3	4
Depot	0	25	21	13	5
1	25	0	14	10	12
2	21	14	0	7	1
3	13	10	7	0	5
4	5	12	1	5	0

2. Menentukan *saving matrix*

Pada tahap ini, *saving matrix* ditentukan berdasarkan matriks jarak yang telah didapatkan pada tahapan sebelumnya. *Saving matrix* berisi *saving value* untuk setiap pasangan titik yang dihitung dengan Persamaan (10). Berdasarkan matriks jarak pada Tabel 2, maka didapatkan *saving matrix* yang dapat dilihat pada Tabel 3.

$$S_{ij} = C_{0i} + C_{0j} - C_{ij} \tag{10}$$

Tabel 3. *Saving matrix*

Node	1	2	3	4
1	0	32	28	18
2	32	0	27	25
3	28	27	0	13
4	18	25	13	0

3. Menentukan *head* dan *tail* awal

Setelah didapatkan *saving matrix* pada tahap sebelumnya, pasangan titik diurutkan dari *saving value* terbesar ke terkecil. *Saving value* bersifat simetris sehingga $S_{ij} = S_{ji}$. Selanjutnya dibuat rute pertama menggunakan kendaraan k_1 dan ditentukan *head* dan *tail* awal. *Head* dan *tail* awal merupakan pasangan titik yang memiliki *saving value* terbesar. Jika terdapat lebih dari satu pasangan titik dengan *saving value* terbesar maka diambil pasangan titik yang memiliki jarak terdekat.

Setelah dipilih pasangan titik dengan *saving value* terbesar, maka pasangan titik tersebut dicek terlebih dahulu apakah memenuhi kendala kapasitas kendaraan dan waktu distribusi kendaraan. Jika pasangan titik tersebut memenuhi maka titik tersebut ditentukan sebagai *head* dan *tail* awal. Jika tidak memenuhi maka dilanjutkan ke pasangan titik dengan *saving value* terbesar selanjutnya. Pasangan titik yang tidak memenuhi kendala waktu dan kapasitas dihapus dari *list saving value* dan dimasukkan kembali ke *list saving value* untuk kendaraan selanjutnya. Pasangan titik yang terpilih

dihapus dari *list saving value* dan ditetapkan sebagai pelanggan yang sudah terpenuhi permintaannya.

Berdasarkan *saving matrix* pada Tabel 3, didapatkan *list saving value* yang dapat dilihat pada Tabel 4. Berdasarkan Tabel 4, maka titik 1 dan 2 ditentukan sebagai *head* dan *tail* awal .

Tabel 4. Iterasi 0

<i>Saving value</i> (S_{ij})	(i, j)	Rute sementara
32	(1,2)	
28	(1,3)	
27	(2,3)	1 - 2
25	(2,4)	
18	(1,4)	
13	(3,4)	

4. Menentukan *head* dan *tail* baru

Setelah ditentukan *head* dan *tail* awal yaitu titik 1 dan 2, maka tahap selanjutnya adalah menentukan titik-titik yang akan digabungkan dalam rute sementara. *List saving value* pasangan titik yang belum terpenuhi dengan *head* dan pasangan titik yang belum terpenuhi dengan *tail* dibandingkan untuk menentukan titik baru yang akan ditambahkan ke rute sementara. Titik tersebut akan menggantikan *head* atau *tail* awal jika memenuhi kendala kapasitas kendaraan dan waktu distribusi kendaraan sehingga didapatkan *head* dan *tail* yang baru. Jika terdapat *saving value* yang sama maka dipilih pasangan titik dengan jarak terdekat.

Tahap ini dilakukan sampai *list saving value* sudah tidak memenuhi kendala dan iterasi dilanjutkan untuk kendaraan selanjutnya ($k = k + 1$) dengan memulai kembali pemilihan *head* dan *tail* awal. Berikut merupakan tahapan penentuan *head* dan *tail* baru sampai seluruh titik terpenuhi.

Iterasi 1

Tabel 5. Iterasi 1

<i>Saving value</i> (S_{ij})	($i, head$)	<i>Saving value</i> (S_{ij})	($tail, j$)	Rute sementara
28	(3,1)	27	(2,3)	3 - 1 - 2
18	(4,1)	25	(2,4)	

Saving value terbesar terdapat pada pasangan titik 3 dan 1 sehingga titik 3 ditentukan sebagai *head* yang baru.

Iterasi 2

Tabel 6. Iterasi 2

<i>Saving value</i> (S_{ij})	($i, head$)	<i>Saving value</i> (S_{ij})	($tail, j$)	Rute sementara
13	(4,3)	25	(2,4)	3 - 1 - 2 - 4

Saving value terbesar terdapat pada pasangan titik 2 dan 4 sehingga titik 4 ditentukan sebagai *tail* yang baru.

5. Menentukan rute

Tahapan sebelumnya dihentikan jika seluruh titik pelanggan telah terpenuhi permintaannya. Pada tahap ini ditentukan rute terbentuk dari rute-rute sementara yang telah terbentuk sebelumnya dengan menambahkan Depot pada kedua ujungnya. Jarak antar titik bersifat simetris sehingga rute terbentuk dapat ditentukan dengan

urutan sebaliknya. Rute terbentuk: Depot – 3 – 1 – 2 – 4 – Depot atau Depot – 4 – 2 – 1 – 3 – Depot.

3. Hasil dan Pembahasan

3.1. Data

Pada penelitian ini, data yang digunakan merupakan data referensi dari penelitian sebelumnya (Yuliza, 2020). Parameter-parameter permasalahan disesuaikan sehingga hasil yang didapatkan sesuai dengan model yang dibuat. Data yang digunakan terdiri dari 24 titik pelanggan dengan masing-masing permintaan pelanggan ditunjukkan pada Tabel 8, kapasitas kendaraan $Q_k = 4000 \text{ kg}$, biaya tetap $F_k = Rp150.000$, dan $B_k = Rp1000/km$, dan data matriks jarak dapat dilihat pada Tabel 7.

Tabel 7. Data matriks jarak

Node	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	0,0	8,7	1,5	12,0	9,0	1,1	2,5	2,0	1,6	0,3	3,2	1,5	7,9	2,9	0,9	16,0	5,3	2,5	4,2	3,0	3,4	0,2	0,5	0,7	3,3
1	8,7	0,0	12,0	21,0	4,7	15,0	16,0	10,0	11,0	12,0	8,9	12,0	6,6	15,0	12,0	24,0	8,0	15,0	9,4	12,0	18,0	12,0	13,0	13,0	18,0
2	1,5	12,0	0,0	13,0	10,0	3,3	3,2	3,1	2,6	1,0	4,5	0,1	9,2	3,6	1,9	16,0	6,6	3,3	5,5	1,9	4,2	0,9	1,1	0,5	4,1
3	12,0	21,0	13,0	0,0	15,0	8,1	8,3	11,0	10,0	9,6	12,0	10,0	13,0	7,0	9,6	9,5	14,0	6,6	13,0	9,5	10,0	9,6	9,6	10,0	9,8
4	9,0	4,7	10,0	15,0	0,0	11,0	13,0	8,2	8,5	10,0	6,9	10,0	3,7	13,0	10,0	22,0	3,5	13,0	7,5	10,0	14,0	10,0	11,0	11,0	13,0
5	1,1	15,0	3,3	8,1	11,0	0,0	1,4	2,7	2,2	1,5	3,9	2,2	8,6	1,8	1,5	14,0	6,0	1,4	4,9	0,7	2,4	1,5	1,6	1,9	2,2
6	2,5	16,0	3,2	8,3	13,0	1,4	0,0	4,5	4,1	3,4	5,7	4,1	11,0	0,8	3,4	13,0	7,9	0,4	6,7	3,3	3,8	3,3	3,4	3,8	3,6
7	2,0	10,0	3,1	11,0	8,2	2,7	4,5	0,0	0,5	3,5	1,6	4,7	7,5	6,0	1,3	19,0	5,0	5,7	2,4	2,7	6,6	3,4	3,7	3,8	6,4
8	1,6	11,0	2,6	10,0	8,5	2,2	4,1	0,5	0,0	3,0	2,4	4,2	7,1	5,6	1,0	18,0	4,5	5,2	3,4	2,9	6,1	3,0	3,2	3,4	6,0
9	0,3	12,0	1,0	9,6	10,0	1,5	3,4	3,5	3,0	0,0	3,3	1,0	8,0	2,9	1,0	16,0	5,4	2,6	4,2	0,9	3,5	0,2	0,2	0,6	3,4
10	3,2	8,9	4,5	12,0	6,9	3,9	5,7	1,6	2,4	3,3	0,0	4,3	6,3	5,6	3,0	18,0	3,7	5,2	1,2	3,0	6,2	3,0	3,2	3,4	6,0
11	1,5	12,0	0,1	10,0	10,0	2,2	4,1	4,7	4,2	1,0	4,3	0,0	9,2	3,7	1,9	16,0	6,6	3,3	5,4	1,9	4,3	0,9	0,9	0,5	4,1
12	7,9	6,6	9,2	13,0	3,7	8,6	11,0	7,5	7,1	8,0	6,3	9,2	0,0	10,0	13,0	19,0	3,0	10,0	9,3	13,0	13,0	13,0	13,0	13,0	13,0
13	2,9	15,0	3,6	7,0	13,0	1,8	0,8	6,0	5,6	2,9	5,6	3,7	10,0	0,0	3,4	13,0	7,8	0,4	6,7	3,3	1,7	3,3	3,4	3,8	1,8
14	0,9	12,0	1,9	9,6	10,0	1,5	3,4	1,3	1,0	1,0	3,0	1,9	13,0	3,4	0,0	16,0	4,9	3,1	2,2	0,4	4,0	0,8	1,1	1,2	3,8
15	16,0	24,0	16,0	9,5	22,0	14,0	13,0	19,0	18,0	16,0	18,0	16,0	19,0	13,0	16,0	0,0	20,0	13,0	19,0	16,0	16,0	16,0	16,0	16,0	16,0
16	5,3	8,0	6,6	14,0	3,5	6,0	7,9	5,0	4,5	5,4	3,7	6,6	3,0	7,8	4,9	20,0	0,0	13,0	6,8	9,3	13,0	9,3	9,6	9,8	12,0
17	2,5	15,0	3,3	6,6	13,0	1,4	0,4	5,7	5,2	2,6	5,2	3,3	10,0	0,4	3,1	13,0	13,0	0,0	6,3	2,9	3,3	2,9	3,0	3,0	32,0
18	4,2	9,4	5,5	13,0	7,5	4,9	6,7	2,4	3,4	4,2	1,2	5,4	9,3	6,7	2,2	19,0	6,8	6,3	0,0	3,8	7,0	3,8	4,1	4,2	6,8
19	3,0	12,0	1,9	9,5	10,0	0,7	3,3	2,7	2,9	0,9	3,0	1,9	13,0	3,3	0,4	16,0	9,3	2,9	3,8	0,0	4,1	1,0	1,2	1,4	4,0
20	3,4	18,0	4,2	10,0	14,0	2,4	3,8	6,6	6,1	3,5	6,2	4,3	13,0	1,7	4,0	16,0	13,0	3,3	7,0	4,1	0,0	3,9	3,9	4,3	1,0
21	0,2	12,0	0,9	9,6	10,0	1,5	3,3	3,4	3,0	0,2	3,0	0,9	13,0	3,3	0,8	16,0	9,3	2,9	3,8	1,0	3,9	0,0	0,5	0,5	3,5
22	0,5	13,0	1,1	9,6	11,0	1,6	3,4	3,7	3,2	0,2	3,2	0,9	13,0	3,4	1,1	16,0	9,6	3,0	4,1	1,2	3,9	0,5	0,0	0,5	3,4
23	0,7	13,0	0,5	10,0	11,0	1,9	3,8	3,8	3,4	0,6	3,4	0,5	13,0	3,8	1,2	16,0	9,8	3,0	4,2	1,4	4,3	0,5	0,5	0,0	3,9
24	3,3	18,0	4,1	9,8	13,0	2,2	3,6	6,4	6,0	3,4	6,0	4,1	13,0	1,8	3,8	16,0	12,0	32,0	6,8	4,0	1,0	3,5	3,4	3,9	0,0

Tabel 8. Data permintaan

Node	D (kg)	Node	D (kg)	Node	D (kg)	Node	D (kg)	Node	D (kg)	Node	D (kg)	Node	D (kg)	Node	D (kg)
1	4000	4	2520	7	3240	10	1560	13	450	16	1440	19	2700	22	1500
2	1350	5	1200	8	1440	11	1050	14	1950	17	2340	20	1500	23	600
3	2730	6	3900	9	750	12	2160	15	4000	18	2880	21	900	24	4000

3.2. Solusi model menggunakan algoritma Clarke and Wright Savings, solver LINGO, dan pengembangan algoritma Clarke and Wright Savings

Model yang dibuat digunakan untuk menyelesaikan permasalahan dengan data referensi. Pertama, model diselesaikan menggunakan algoritma Clarke and Wright Savings. Selanjutnya model diselesaikan menggunakan solver LINGO. Penyelesaian dengan solver LINGO menggunakan komputer dengan spesifikasi Intel(R) Core(TM) i5-8265U CPU

@1.60GHz 1.80 GHz dan RAM 12 GB. Namun, solusi yang dihasilkan dari *solver* LINGO bukan merupakan solusi global optimal melainkan solusi layak dengan waktu komputasi *solver* selama 74 jam. Terakhir, model diselesaikan menggunakan pengembangan algoritma *Clarke and Wright Savings*. Hasil penyelesaian model menggunakan 3 metode tersebut ditunjukkan pada Tabel 9.

Berdasarkan solusi penentuan rute menggunakan algoritma *Clarke and Wright Savings*, solusi model menggunakan *solver* LINGO, dan solusi menggunakan pengembangan algoritma *Clarke and Wright Savings* didapatkan perbandingan hasil yang dapat dilihat pada Tabel 10. Solusi dari pengembangan algoritma *Clarke and Wright Savings* menghasilkan perbaikan jarak tempuh sebesar 1,81% dan total biaya sebesar 0,12% dari solusi algoritma acuan *Clarke and Wright Savings*. Solusi dari pengembangan algoritma *Clarke and Wright Savings* juga menghasilkan perbaikan jarak tempuh sebesar 3,09% dan total biaya sebesar 0,21% dari solusi menggunakan *solver* LINGO selama 74 jam. Dengan waktu komputasi yang lebih singkat hanya dengan selama 1 menit, solusi yang dihasilkan dari pengembangan algoritma lebih baik dibandingkan dengan solusi *solver* LINGO selama 74 jam.

Tabel 9. Solusi algoritma *Clarke and Wright Savings*

Kendaraan	Algoritma <i>Clarke and Wright Savings</i>	<i>Solver</i> LINGO (74 jam)	Pengembangan algoritma <i>Clarke and Wright Savings</i>
1	0-4-16-0	0-3-13-0	0-4-16-0
2	0-3-13-9-0	0-15-0	0-9-3-13-0
3	0-10-12-0	0-18-0	0-10-12-0
4	0-5-19-0	0-22-11-2-0	0-5-19-0
5	0-2-11-23-21-0	0-5-23-21-0	0-23-2-11-21-0
6	0-8-14-0	0-19-9-0	0-17-20-0
7	0-17-20-0	0-14-8-0	0-8-14-0
8	0-1-0	0-24-0	0-1-0
9	0-6-0	0-6-0	0-6-0
10	0-7-0	0-1-0	0-7-0
11	0-15-0	0-7-0	0-15-0
12	0-18-0	0-10-12-0	0-18-0
13	0-22-0	0-17-20-0	0-22-0
14	0-24-0	0-4-16-0	0-24-0
Total jarak	151,94	153,94	149,25

Tabel 10. Perbandingan solusi

Indikator	Hasil pengembangan (a)	Algoritma <i>Clarke and Wright Savings</i> (b)	Perbaikan (c)=((b)-(a))/(b)	<i>Solver</i> LINGO (d)	Gap (e)=((d)-(a))/(d)
Jarak	149,19	151,94	1,81%	153,94	3,09%
Total biaya	Rp2.249.190	Rp2.251.940	0,12%	Rp2.253.940	0,21%

3.3. Analisis Implikasi Manajerial

Dari segi ekonomi, hasil pengembangan algoritma *Clarke and Wrights Saving* dapat menurunkan total biaya distribusi dari jumlah kendaraan yang digunakan dan jarak tempuh masing-masing kendaraan. Pada penelitian ini, penurunan total biaya sebesar 0,12% hanya dihasilkan dari penurunan jarak tempuh sebesar 1,81% dari solusi algoritma acuan. Namun, tidak menutup kemungkinan pada kasus lain akan terdapat penurunan jumlah kendaraan yang digunakan. Penurunan total biaya distribusi ini akan meningkatkan keuntungan yang diperoleh perusahaan.

Dari segi waktu, hasil pengembangan algoritma dapat menghemat komputasi menjadi 1 menit. Sedangkan komputasi pada pendekatan eksak memerlukan waktu yang relatif lama,

pada penelitian ini selama 74 jam, dengan hasil yang tidak lebih baik dari solusi dari pengembangan algoritma sehingga jika solusi ini diterapkan pada perusahaan akan lebih cepat dalam mengambil keputusan penentuan rute distribusi.

4. Kesimpulan

Pada penelitian ini, hasil pengembangan algoritma *Clarke and Wright Savings* dengan mengubah cara pemilihan titik pelanggan menggunakan konsep *head* dan *tail* serta penambahan pembatas waktu menghasilkan solusi yang lebih baik dibandingkan dengan algoritma *Clarke and Wright Savings* dan penyelesaian menggunakan *solver* LINGO selama 74 jam.

Ucapan Terima Kasih: Penulis ingin menyampaikan penghargaan yang setulus-tulusnya kepada Kelompok Keahlian Sistem Manufaktur di FTI ITB yang telah bermurah hati mendanai publikasi penelitian ini.

Daftar Pustaka

- Arifudin, A., Wisnubroto, P., & Parwati, C. I. (2017). Optimalisasi vehicle routing problem dengan pendekatan metode saving matrix dan clarke & wright saving heuristic. *Jurnal REKAVASI*, 1-9.
- Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number. *12(4)*, 568-581.
- Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science: INFORMS*, 80-91.
- Fuadi, A. S., & Pujotomo, D. (2018). Penyelesaian vehicle routing problem menggunakan metode clarke and wright saving heuristic (Studi kasus: PT. Coca Cola Amatil Indonesia-Wilayah Banyuwangi). *IENACO (Industrial Engineering National Conference)*, (6), 164-172.
- Irman, A., Ekawati, R., & Febriana, N. (2017). Optimalisasi rute distribusi air minum quelle dengan algoritma clarke & wright saving dan model vehicle routing problem. *Prosiding SENIATI*, C1.1-C1.7.
- Kurniawan, I. S., Susanty, S., & Adiarto, H. (2014). Usulan rute pendistribusian air mineral dalam kemasan menggunakan metode nearest neighbour dan clarke & wright savings (Studi kasus di PT. X Bandung). *Jurnal Online Institut Teknologi Nasional*, 1, (1), 125-136.
- Lin, C., Choy, K. L., Ho, G. T., Chung, S. H., & Lam, H. Y. (2014). Survey of green vehicle routing problem: past and future trends. *Expert Systems with Applications*, 1118-1138.
- Marpaung, L. E., Arifin, J., & Winarno. (2022). Optimalisasi rute distribusi menggunakan algoritma clarke and wright savings. *Jurnal Media Teknik dan Sistem Industri*, (6), 76-83.
- Meilani, D., & Iswara, A. (2018). Aplikasi penentuan rute distribusi LPG 3 kg. *Jurnal Optimasi Sistem Industri*, 17(2), 208-219.
- Octora, L., Imran, A., & Susanty, S. (2014). Pembentukan rute distribusi menggunakan algoritma clarke & wright savings dan algoritma sequential insertion. *Jurnal Online Institut Teknologi Nasional*, 2, Vol. 02 (1-11).
- Purnomo, A. (2010). Penentuan rute pengiriman dan biaya transportasi dengan menggunakan metode clark and wright saving heuristic (Studi kasus di PT Teh Botol Sosro Bandung). *Jurnal Logistik Bisnis Politeknik Pos Indonesia*, Vol. 01 (97-117).

- Rohandi, S. M., Imran, A., & Prasetiyo, H. (2014). Penentuan rute distribusi produk obat menggunakan metode sequential insertion dan clarke & wright savings (Studi kasus di PT. X Bandung). *Jurnal Online Institut Teknologi Nasional*, Vol. 02 (34-45).
- Rupiah, S., Mulyono, & Sugiharti, E. (2017). Efektivitas algoritma clarke-wright dan sequential insertion dalam penentuan rute pendistribusian tabung gas LPG. *UNNES Journal of Mathematics*, 6(2), 198-210.
- Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. Philadelphia: Siam.
- Yuliza, E., Puspita, F. M., Yahdin, S., & Emiliya, R. (2020). Solving capacitated vehicle routing problem using of Clarke and Wright Savings algorithm and LINGO in LPG distribution. *Journal of Physics: Conference Series*, (Vol. 1663, No. 1, p. 012027).