

MODEL VECHICLE ROUTING PROBLEM WITH BACKHAULS TIME WINDOWS PADA PRODUSEN X DENGAN PERBAIKAN 2-OPT

Granita Hajar¹, Muhammad Dzulfikar Fauzi², Nicko Nur Rakhmaddian³, Nisa Isrofi⁴

^{1,3,4} Teknik Logistik, Fakultas Teknologi Elektro dan Industri Cerdas, Institut Teknologi Telkom Surabaya,

² Informatika, Fakultas Teknologi Informasi dan Bisnis, Institut Teknologi Telkom Surabaya

Email: granita@ittelkom-sby.ac.id¹, muhhammad.dzulfikar.f@ittelkom-sby.ac.id², nickonurrakhmaddian@ittelkom-sby.ac.id³, nisa.isrofi@ittelkom-sby.ac.id⁴

ABSTRACT

Pastries are one of the snacks that are in demand by various ages, from children to adults. Demand for pastries will increase during the holidays. Cookies can be used as gifts to friends, family or work partners. The proliferation of pastry entrepreneurs makes producers have to improve the quality of their services. One way to improve service can be done in terms of timely delivery. Timeliness in the delivery of pastries is an important factor for an entrepreneur, this is because it is related to costs, so by determining the right route it is expected to minimize costs. In shipping pastries from one customer to another, they must pay attention to the time windows of each customer, but also pay attention to vehicle capacity. In this case, after delivering the cookies to the customer, the vehicle will pick up the raw material for the cake at several shops. This problem can be solved by Vehicle Routing Problem with Backhauls Time Windows (VRPBTW). This study describes the process of determining the route of cake delivery and retrieval of raw materials using the Nearest Neighbor and made improvements with 2-Opt. The results of this study required three vehicles to be able to serve all customers using VRPBTW with a total travel time of 701.1 minutes and a distance of 134.4 km.

Keywords: *Distribution, Vehicle Routing Problem Backhauls, Time Windows, 2-Opt*

ABSTRAK

Kue kering merupakan salah satu makanan ringan yang diminati oleh berbagai macam usia, mulai anak-anak hingga dewasa. Demand kue kering akan meningkat pada saat hari raya. Kue kering dapat dijadikan hadiah kepada teman, keluarga ataupun partner kerja. Menjamurnya pengusaha kue kering membuat para produsen harus meningkatkan kualitas pelayanan mereka. Salah satu cara peningkatan pelayanan dapat dilakukan dari sisi pengiriman yang tepat waktu. Ketepatan waktu dalam pengiriman kue kering menjadi faktor yang penting untuk sebuah pengusaha, hal ini dikarenakan terkait dengan biaya, maka dengan menentukan rute yang tepat diharapkan dapat meminimalkan biaya. Dalam pengiriman kue kering dari satu pelanggan ke pelanggan lainnya harus memperhatikan time windows dari masing-masing pelanggan, selain itu juga memperhatikan kapasitas kendaraan. Dalam permasalahan ini, setelah mengantar kue kering ke pelanggan, kendaraan akan mengambil bahan baku kue di beberapa toko. Permasalahan ini dapat diselesaikan dengan Vehicle Routing Problem with Backhauls Time Windows (VRPBTW). Penelitian ini menjelaskan proses penentuan rute pengiriman kue dan pengambilan bahan baku dengan menggunakan Nearest Neighbour dan dilakukan perbaikan dengan 2-Opt. Hasil dari penelitian ini diperlukan tiga kendaraan untuk dapat melayani semua pelanggan menggunakan VRPBTW dengan total waktu tempuh 701.1 menit dan jarak tempuh 134.4 km.

Kata kunci: *Distribusi, Vehicle Routing Problem Backhauls, Time Windows, 2-Opt*

Pendahuluan

Kue kering salah satu suguhan khas yang ada pada setiap rumah. Diminati banyak kalangan mulai dari anak-anak remaja hingga dewasa. Menjamurnya produsen kue kering membuat persiapan semakin terasa sengit, maka dari itu produsen berlomba-lomba meningkatkan pelayanan mereka. Salah satu pelayanan yang dapat ditingkatkan yaitu dalam kecepatan dan ketepatan dalam pengiriman. Selain itu bahan baku yang fresh dan higienis juga menjadi salah satu cara peningkatan kualitas produsen kue kering. Untuk mendapatkan bahan baku yang selalu *fresh*, proses pengambilan bahan baku dapat dilakukan setiap setelah pengiriman kue kering ke pelanggan. Produsen kue kering X memiliki pelanggan tetap pada 14 titik dan pengambilan bahan baku 5 titik di Surabaya. Penentuan rute untuk pengiriman kue kering ke 14 pelanggan menjadi kegiatan yang sangat penting bagi produsen X. Dalam setiap pengiriman, setiap pelanggan produsen kue kering X memiliki batasan waktu tertentu (time windows). Selain pelanggan yang memiliki batasan waktu, toko bahan baku juga memiliki batasan waktu tersendiri. Produsen melakukan pembelian bahan baku setelah menyelesaikan pengiriman kue kering ke pelanggan.

Produsen X ingin mengetahui berapa kendaraan yang dibutuhkan untuk dapat melakukan pengiriman kue kering dan pembelian bahan baku. Sehingga permasalahan dari produsen X ini dapat di modelkan sebagai *Vehicle Routing Problem with Backhauls Time Windows* (VRPBTW), sebagaimana model VRPBTW akan menghasilkan rute kendaraan untuk mengunjungi setiap pelanggan untuk mengirimkan kue kering dan akan melakukan pengambilan/pembelian bahan baku setelah selesai pengiriman. Setiap rute diawali dan diakhir pada toko produsen X (depo). Dalam satu rute, produsen harus memastikan total kue kering yang dikirim dan bahan baku yang akan di ambil tidak melebihi kapasitas kendaraan.

Waktu pengiriman kue kering hanya di perbolehkan pukul 08.30 hingga pukul 15.30, sedangkan toko bahan baku memiliki batasan waktu yang beragam. Dengan menggunakan VRPBTW diharapkan waktu dan jarak pengiriman kue kering dan pengambilan bahan baku akan lebih optimal. Selain memperhitungkan waktu, penentuan rute juga memperhitungkan kapasitas kendaraan. Pengiriman kue kering menggunakan mobil box dengan kapasitas 120 box, dan pengambilan bahan baku dengan menggunakan mobil box yang sama namun dengan kapasitas 40 box. Tujuan penelitian ini untuk melakukan penentuan rute baru yang dilakukan dengan pendekatan heuristic dengan metode *Nearest Neighbour* dan dilakukan perbaikan dengan 2-opt. Jarak rute yang optimal akan membuat biaya transportasi juga optimal atau dapat diminimumkan [1]. Penelitian ini menggunakan VRPBTW yang diharapkan dapat minimal kendaraan yang digunakan, rute terpendek dan biaya yang minimum.

Metode Penelitian

Metode penelitian pada penelitian ini dimulai dari survei,

1. Survei

Studi kasus pada penelitian ini yaitu pengiriman kue kering dan pengambilan bahan baku pada produsen x dimana sebelumnya peneliti memahami dan mengenai permasalahan yang terjadi dalam pengiriman dan pembelian bahan baku.

2. Studi Pustaka

Setelah dilakukan survei terhadap permasalahan yang terjadi dilakukan studi Pustaka yang digunakan untuk mencari referensi guna untuk mendukung penelitian ini. Beberapa referensi yang digunakan pada penelitian ini yaitu sebagai berikut:

a. Penelitian Operasional

Penelitian Operasional atau Operational Research merupakan sebuah penelitian yang memiliki bertujuan untuk memberikan solusi pada permasalahan dibidang operasional dengan tetap memperhatikan kaidah ilmiah [2]. Pada penelitian operasional untuk menyelesaikan permasalahan hal yang paling penting adalah pada tahap pembuatan model dari permasalahan tersebut. Model merupakan sebuah representasi, atau deskripsi yang menjelaskan suatu sistem, yang telah disederhanakan sehingga mudah di rekayasa [3].

b. Travelling Salesman Problem

Traveling Salesman Problem (TSP) adalah salah satu permasalahan optimasi kombinatorial. TSP ditemukan pada tahun 1800-an oleh William Rowan Hamilton dan Thomas Penyngton. Permasalah TSP adalah adanya sebuah sekumpulan node dan biaya perjalanan (atau jarak) antar node untuk menemukan jalan terbaik untuk melakukan kunjungan ke semua node dan kembali ke titik awal dengan tujuan meminimalkan biaya atau jarak perjalanan [4]. Pentingnya TSP yaitu bisa mewakili kelas masalah yang lebih besar (masalah optimasi kombinatorial). Masalah TSP termasuk dalam kelas masalah seperti yang dikenal sebagai NP-complete. Secara khusus, jika seseorang dapat menemukan efisien (yaitu, polinomial-waktu) algoritma untuk masalah penjual keliling, maka algoritma yang efisien dapat ditemukan untuk semua masalah lain di kelas NP-complete [5].

c. Vehicle Routing Problem

Biaya Transportasi merupakan bagian dari sistem logistik yang mendominasi total biaya keseluruhan. Biaya transportasi ini harus efisien memberikan kontribusi dalam mengurangi total biaya dan meningkatkan keunggulan kompetitif perusahaan. Sehingga rute kendaraan yang efisien dan jadwalnya adalah masalah logistik yang representative [6]. *Vehicle Routing Problem* (VRP) adalah pengembangan dari Traveling Salesman Problem (TSP) termasuk dalam *NP-Hard Optimization problem*. Penyelesaian kasus VRP adalah penentuan rute optimal, konsep dasar dari VRP melibatkan node yang dapat dikatakan sebuah *demand* atau pelanggan. Setiap kendaraan yang dipakai melakukan perjalanan harus dimulai dan berakhir di depot yang sama. Total *demand* dalam satu rute tidak diperbolehkan melebihi kapasitas kendaraan yang melewati rute tersebut. VRP memiliki tujuan untuk meminimalkan biaya berdasarkan total jarak atau waktu tempuh dengan batasan kapasitas kendaraan, dan juga meminimalkan jumlah kendaraan yang akan digunakan [7]. VRP yang konsisten memiliki rute yang biayanya minimal tetapi juga memenuhi tingkat kepuasan pelanggan tertentu. Rute yang konsisten bisa berupa waktu kunjungan atau pengemudi yang melayani pelanggan [8].

d. Vehicle Routing Problem with Backhauls Time Windows

Vehicle Routing Problem with Backhauls (VRPB) merupakan pengembangan dari VRP dimana ada penambahan satu kendala yaitu permintaan pengambilan barang (*backhaul*) dari pelanggan. Permintaan *backhaul* hanya dapat dilayani jika semua permintaan/pengiriman dari pelanggan *linehauls* telah terselesaikan. Minimal harus ada satu pelanggan *linehauls* dalam satu rute [9].

e. Vehicle Routing Problem with Backhauls Time Window

Vehicle Routing Problem with Backhauls Time Window (VRPBTW) juga merupakan pengembangan dari VRPB dimana pelayanan pada setiap pelanggan *linehauls* dan *backhauls* harus dimulai dalam sebuah interval waktu yang disebut dengan *time windows*. VRPBTW memiliki kesamaan dengan VRPB namun dalam VRPBTW ditambah satu kendala yaitu, jangka waktu [10]. Jangka waktu ini berkaitan dengan setiap pelanggan, dimana dapat pelanggan *linehauls* harus dikirim dalam interval waktu tertentu dan pelanggan *backhauls* harus diambil dalam interval waktu tertentu juga. Tujuannya sama dengan VRP yaitu meminimalkan jumlah kendaraan dan total waktu perjalanan. Komponen dari VRPBTW yaitu adanya sejumlah kendaraan dengan kapasitas tertentu, beberapa pelanggan *linehauls* dan pelanggan *backhauls*, lokasi pelanggan (jarak) dan juga interval waktu [11]. Pelanggan dinotasikan dengan angka berupa 1,2,3,...i sedangkan depo tempat dimulainya pengiriman untuk pelanggan *linehauls* dan tempat kembalinya kendaraan setelah pengiriman *backhauls* dinotasikan angka 0. Pengambilan dan pengiriman barang dari dan ke pelanggan hanya dapat dilayani dengan batas waktu (*time windows*). *Time windows* yang ditentukan oleh pelanggan dinotasikan sebagai $[a_i, b_i]$, dimana a_i adalah dimulainya pelayanan sedangkan b_i adalah akhir dari waktu pelayanan. Berikut ini adalah formulasi *vehicle routing problem with backhauls* dengan menggunakan fungsi pembatas kapasitas dan *time windows*: variabel keputusan:

$$X_{ijk} = \begin{cases} 1, & \text{jika kendaraan } k \text{ segera melayani } j \text{ setelah mengunjungi } i \\ 0, & \text{untuk lainnya} \end{cases}$$

Formulasi :

Fungsi tujuan

Minimasi

$$\sum_{k=1}^K \sum_{i=0}^n \sum_{j=0}^n C_{ijk} X_{ijk} \quad (1)$$

Pembatas :

$$\sum_{i=0}^n \sum_{k=1}^K X_{ijk} = 1 \quad j = 1, \dots, n \quad (2)$$

$$\sum_{j=0}^n \sum_{k=1}^K X_{ijk} = 1 \quad i = 1, \dots, n \quad (3)$$

$$\sum_{j=0}^n \sum_{k=1}^K X_{0jk} = K \quad (4)$$

$$\sum_{i=0}^n \sum_{k=1}^K X_{i0k} = K \quad (5)$$

$$\sum_{i=1}^n \sum_{j=1}^n d_i X_{jk} \leq Q_k \quad k = 1, \dots, K; j = 0, 1, \dots, n \quad (6)$$

$$\sum_{i=1}^n \sum_{j=1}^n g_i X_{jk} \leq Q_k \quad k = 1, \dots, K; j = 0, 1, \dots, n \quad (7)$$

$$\sum_{i=0}^n d_i \sum_{i=0}^n X_{ijk} \leq 1 \quad k = 1, \dots, K \quad (8)$$

$$X_{ijk}(T_{ik} + S_i + t_{jk} - T_{jk}) \leq 0 \quad (9)$$

$$a_i \leq T_{ik} \leq b_i \quad k = 1, \dots, K; (i, j) = 1, \dots, n \quad (10)$$

$$X_{ijk} \in \{1, 0\} \quad (11)$$

Notasi

L = Jumlah dari pelanggan *linehaul*, \dots , $L = 0$ (*depo*)

B = Jumlah dari pelanggan *backhaul*, \dots , $L + 1, \dots, n$

n = Total jumlah dari pelanggan ($L + B$)

K = Jumlah kendaraan $k = 1 \dots \dots \dots , K$

Q_k = Kapasitas dari kendaraan k

d_i = *demand* pelanggan linehaul ke i

g_i = *demand* pelanggan backhaul ke i

C_{ij} = *travelling cost* (jarak) dari node i ke node j

T_{ik} = waktu memulai pelayanan di pelanggan i dengan menggunakan kendaraan k s_i = waktu pelayanan di pelanggan i

t_{ij} = waktu perjalanan yang dibutuhkan dari i ke j

a_i = waktu awal pelanggan i

b_i = waktu terakhir pelanggan i

Persamaan 1 fungsi tujuan yaitu meminimalkan biaya. Persamaan 2 yaitu fungsi pembatas yang menjamin bahwa dalam satu rute terdapat kendaraan k melayani pelanggan j dan tepat langsung ke pelanggan i . Persamaan 3 yaitu fungsi pembatas yang menjamin bahwa dalam satu rute terdapat kendaraan k melayani pelanggan i dan langsung ke pelanggan j . Persamaan 4 dan Persamaan 5 fungsi pembatas yang menjamin bahwa jumlah kendaraan yang meninggalkan (keluar) depo harus sama dengan jumlah kendaraan yang kembali (masuk) ke depo. Persamaan 6 menjamin bahwa *demand* dari pelanggan *linehauls* tidak diperbolehkan melebihi kapasitas kendaraan. Persamaan 7 menjamin bahwa *demand* dari pelanggan *backhauls* tidak diperbolehkan melebihi kapasitas kendaraan. Persamaan 8 menjamin bahwa pelanggan *backhauls* hanya dapat dilayani setelah pelanggan *linehauls* dilayani. Persamaan 9 dan Persamaan 10 menjamin bahwa jadwal yang feasible memperhatikan time windows. Pembatas pada Persamaan 9 akan bernilai 0/Minus (-) apabila $T_{jk} \geq T_{ik} + s_i + t_{ij}$. Persamaan 11 akan bernilai 1 apabila ada kendaraan k dari pelanggan i langsung segera mengunjungi pelanggan j .

f. Nearest Neighbour

Nearest Neighbour metode heuristic dalam penyelesaian VRP dimana penentuan rute diarahkan kepada pelanggan terdekat yang belum dikunjungi [12]. Algoritma metode *Nearest Neighbour* dapat dijelaskan sebagai berikut [13]:

- Berawal dari depo (node 0) dan mencari lokasi yang dikunjungi dengan jarak terdekat dengan syarat semua pelanggan hanya dikunjungi 1 kali.
- Jumlah *demand* tidak melebihi kapasitas kendaraan.
- Apabila kapasitas kendaraan tidak memenuhi *demand*, kendaraan harus kembali ke depo untuk mengambil muatan dan kembali melakukan pengiriman ke pelanggan yang belum dikunjungi. Jika tidak kembali dapat membentuk rute baru dengan kendaraan yang berbeda.
- Kemudian proses yang sama juga dilakukan untuk kendaraan-kendaraan berikutnya, sampai semua kendaraan telah penuh atau semua customer telah dikunjungi [14]

g. Tabu search

Tabu Search merupakan metode heuristik yang digunakan untuk menemukan solusi mendekati optimal dari sebuah masalah dengan dilakukannya move. Move yang adalah proses pencarian bergerak dari satu solusi ke solusi berikutnya. Tabu Search memperbaiki solusi awal yang ditemukan dengan pencarian lokal. Tabu List digunakan untuk menyimpan solusi-solusi optimal yang telah ditemukan pada iterasi sebelumnya. Tabu List juga digunakan untuk menuntun proses pencarian agar menelusuri solusi-solusi yang belum pernah dikunjungi sehingga tidak terjadinya perulangan [15].

3. Pengumpulan Data

Terdapat beberapa data yang dibutuhkan pada penelitian ini yaitu: jumlah pelanggan, jumlah *demand*, jumlah *backhauls*, kapasitas kendaraan yang digunakan, jarak untuk setiap pelanggan, dan jarak *backhauls*. Untuk data jarak diperoleh dari google maps dan untuk data permintaan serta waktu menggunakan data dummy.

4. Model Matematis

Berdasarkan permasalahan yang telah dijelaskan pada latar belakang maka pada penelitian ini akan menggunakan metode *Vehicle Routing Problem With Backhauls Time Windows: Simultaneous Pick Up and Delivery with Split Load*. Model matematis akan memuat fungsi tujuan serta batasan. Fungsi tujuan dari *Vehicle Routing Problem With Backhauls Time Windows* : meminimasi waktu pengiriman dan pengambilan bahan baku, dan juga meminimasi jumlah kendaraan yang digunakan.

5. Hasil dan Pembahasan

Dengan menggunakan model matematis maka akan diperoleh hasil dengan menggunakan *Microsoft Excel* dengan perhitungan heuristic dengan algoritma *nearest neighbor*. Kemudian hasil yang diperoleh akan dilakukan analisa dan pembahasan.

6. Simpulan

Penarikan kesimpulan dari hasil penelitian yang telah dilakukan serta memberikan saran untuk produsen x dalam melakukan pengiriman berupa rute pengiriman.

Pada panduan ini akan dijelaskan tentang penulisan *heading*. Jika *heading* anda melebihi satu, gunakan level kedua heading seperti di bawah ini.

Hasil dan Pembahasan

Tahap ini akan di bahas langkah-langkah perhitungan dengan menggunakan *Vehicle Routing Problem Backhauls with Time Windows* agar dapat menemukan rute yang optimal total jarak tempuh dalam pengiriman dan pengambilan bahan baku kue kering. Penelitian ini berfokus pada pembahasan rute yang saja. Data-data dan asumsi yang digunakan dalam penelitian ini sebagai berikut:

- Jumlah pelanggan Linehaul 14 pelanggan.
- Jumlah pelanggan Backhaul 5 pelanggan.
- Pelanggan *Backhauls* hanya dilayani setelah pelanggan *linehauls* selesai dilayani.
- Menggunakan kendaraan mobil box dengan ukuran 130 x 200 x 130 cm
- Kapasitas kendaraan 120 Box untuk pengiriman *linehauls* dan 40 Box untuk *backhauls*
- Kendaraan harus keluar depo seawal-awalnya sehingga datang di pelanggan tepat waktu pelanggan buka.
- Jumlah depo hanya 1
- Data jarak depot dan antar pelanggan diperoleh dari google maps
- Kecepatan kendaraan diasumsikan 20 km/jam.

Waktu tempuh kendaraan diperoleh dari rumus berikut :

$$t = \frac{s}{v} \tag{12}$$

dimana: t= waktu; v=kecepatan; s=jarak

Tabel 1. Data *linehauls*, *backhauls*, *demand* dan *time windows*

	No	Keterangan	Demand (box)	Minimal (Menit)	Maksimum (Menit)	Service Time (Menit)
	0	Adityawarman				
<i>Linehaul</i>	1	Embong Malang	35	510	930	15
	2	Arif rahman hakim	20	510	930	15
	3	Dharmahusada	30	510	930	15
	4	Lidah	22	510	930	15
	5	Siwalankerto	30	510	930	15
	6	Tenggilis	15	510	930	15
	7	Graha pena	14	510	930	15
	8	Basuki Rahmad	25	510	930	15
	9	Indrapura	36	510	930	15
	10	Dipenogoro	30	510	930	15
	11	Pahlawan	25	510	930	15
	12	Perak	15	510	930	15
	13	Jemursari	20	510	930	15
	14	Panglima sudirman	20	510	930	15
<i>Backhaul</i>	15	Indogrosir jemur	20	600	1260	30
	16	Pasar Atom	16	540	1020	30
	17	Pusat Grosir Surabaya	15	540	1020	30
	18	Pasar Kapasan	15	540	960	30
	19	Pasar Pabean	20	480	840	30

Tabel 1 merupakan data *demand* yang ada pada *linehauls* dan *backhauls*, nantinya akan digunakan untuk menghitung kapasitas kendaraan dalam penentuan rute. Selain data *demand*, pada tabel 1 juga disertakan waktu buka toko dalam hitungan menit dan juga waktu *service time*, yaitu dimana *service time* merupakan waktu yang digunakan dalam penurunan atau Data yang diperoleh selanjutnya diolah dengan menggunakan metode *Nearest Neighbour* untuk menemukan rute yang optimal. Berikut langkah-langkah menggunakan metode *Nearest Neighbour*:

1. Menghitung Matriks jarak untuk setiap node dari depo (node 0) ke *Linehaul* atau *Backhaul* dan sebaliknya.

Tabel 2. Matrix jarak tiap node

Jarak (KM)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0	9.3	11.05	8.7	8.6	8.4	10	7.7	5.6	9.2	4.2	9	14	7.9	6.8	7.3	8	8.8	12	11.9
1	7	0	12	7.1	9.7	13.15	13	12.1	5.7	4.2	6.9	4.1	7.7	11.1	3.9	10	4	2.6	5.3	5.1
2	8.5	11	0	4.6	17	12.9	7.1	12	6.2	10	7.9	10.1	15.1	6.2	6	7.4	9.8	10	8.5	11.1
3	6.6	7.3	4.5	0	15	10	7.7	10.9	3.4	6.3	6	5.6	11	10	3.2	6.05	4.9	6	6.1	6.2
4	8.9	10	17	15.1	0	15	16	11.9	12.1	16.1	10	12.85	17	14.1	13	13.9	14	12.95	15.9	18
5	8.8	15.1	13	12	13.9	0	6.7	5	10	14	9.4	14.1	19	4.1	12	6.1	15	14.05	17	16
6	10	14.9	7.8	7.4	16	6.1	0	6.4	9	13	11	12.9	18	5.9	8.8	4	13	13.1	14	14.1
7	4.3	10.1	10.9	7.5	9.8	4.8	8	0	5.8	9.9	4.9	9.8	14.9	4.5	7	6.2	11.1	9.4	11	12
8	4.9	5.2	7.5	4.4	13	9.8	10	9.1	0	5.6	4.4	5.8	9.4	9.3	1.6	8.7	5.1	4.2	6.7	6.5
9	8.7	4.7	11	6.9	14.05	14	14.1	13	4.5	0	8.2	1.3	4.9	13.1	5.4	12	3	2.7	3.7	3.1
10	2.2	5.1	8	5.25	10	7.8	8.5	7.1	2.6	4.9	0	4.8	9.9	7.3	4.2	6.7	5.7	4.5	7.3	7.1
11	7.1	3.5	9.8	5.2	12.95	12	12.8	11	3.9	1.7	6.5	0	6.2	11	3.7	11	1.5	1	3	2.9
12	13	7.3	14	11	16.95	18	19	16.95	9.9	6.3	13	5.9	0	18	10	16	6.3	6.9	7	4.8
13	7.4	9.9	7.55	7.1	12	4	6.6	3.4	8.9	11.5	7.2	12.05	17	0	8.5	2.6	12	12.1	13	14
14	3.3	1.6	6.5	4.25	11	8.2	8.9	7.5	0.5	4.3	3.3	4.2	9.3	7.7	0	7	4.9	4	4.5	6.3
15	6.2	7.6	6.4	4.6	13.1	7.2	3.9	6.5	6.2	9.95	6.8	10	15	6	5.95	0	9.7	10	10.05	11
16	9.9	3.5	8.8	5.15	14	13.2	12	12.05	4.8	2.5	7.3	1.2	6.5	11.9	4.6	12	0	2.2	1.6	1.4
17	8.1	3.6	10.95	6.6	11.9	14	13.9	13	5.3	2.3	8	2.1	7.2	12.9	5.2	13.05	3	0	4.6	4.4
18	7.7	5	8.6	4.2	16	13	11.9	12.1	4.6	3	7.2	1.9	8	12	6.5	9.6	1.3	2.75	0	2.7
19	8.6	3.8	9.3	6.2	17.05	13.1	15	12.9	5.4	1.9	8.3	2	6.8	13	5.5	11.9	1.5	2.5	2.1	0

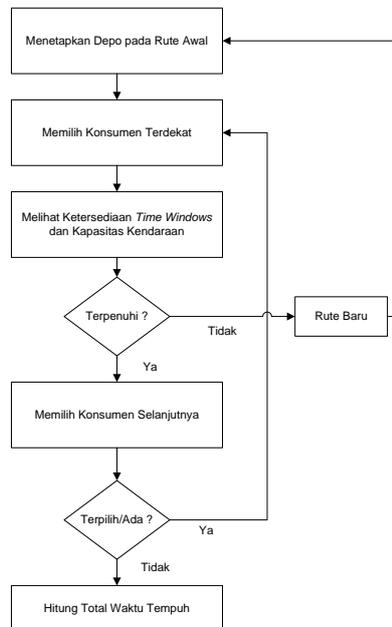
2. Menghitung matrik waktu tempuh menggunakan rumus (3.1)

Tabel 3. Matrix waktu tempuh tiap node

Waktu Tempuh (Menit)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0	27.9	33.15	26.1	25.8	25.2	30	23.1	16.8	27.6	12.6	27	42	23.7	20.4	21.9	24	26.4	36	35.7
1	21	0	36	21.3	29.1	39.45	39	36.3	17.1	12.6	20.7	12.3	23.1	33.3	11.7	30	12	7.8	15.9	15.3
2	25.5	33	0	13.8	51	38.7	21.3	36	18.6	30	23.7	30.3	45.3	18.6	18	22.2	29.4	30	25.5	33.3
3	19.8	21.9	13.5	0	45	30	23.1	32.7	10.2	18.9	18	16.8	33	30	9.6	18.15	14.7	18	18.3	18.6
4	26.7	30	51	45.3	0	45	48	35.7	36.3	48.3	30	38.55	51	42.3	39	41.7	42	38.85	47.7	54
5	26.4	45.3	39	36	41.7	0	20.1	15	30	42	28.2	42.3	57	12.3	36	18.3	45	42.15	51	48
6	30	44.7	23.4	22.2	48	18.3	0	19.2	27	39	33	38.7	54	17.7	26.4	12	39	39.3	42	42.3
7	12.9	30.3	32.7	22.5	29.4	14.4	24	0	17.4	29.7	14.7	29.4	44.7	13.5	21	18.6	33.3	28.2	33	36
8	14.7	15.6	22.5	13.2	39	29.4	30	27.3	0	16.8	13.2	17.4	28.2	27.9	4.8	26.1	15.3	12.6	20.1	19.5
9	26.1	14.1	33	20.7	42.15	42	42.3	39	13.5	0	24.6	3.9	14.7	39.3	16.2	36	9	8.1	11.1	9.3
10	6.6	15.3	24	15.75	30	23.4	25.5	21.3	7.8	14.7	0	14.4	29.7	21.9	12.6	20.1	17.1	13.5	21.9	21.3
11	21.3	10.5	29.4	15.6	38.85	36	38.4	33	11.7	5.1	19.5	0	18.6	33	11.1	33	4.5	3	9	8.7
12	39	21.9	42	33	50.85	54	57	50.85	29.7	18.9	39	17.7	0	54	30	48	18.9	20.7	21	14.4

13	22.2	29.7	22.65	21.3	36	12	19.8	10.2	26.7	34.5	21.6	36.15	51	0	25.5	7.8	36	36.3	39	42
14	9.9	4.8	19.5	12.75	33	24.6	26.7	22.5	1.5	12.9	9.9	12.6	27.9	23.1	0	21	14.7	12	13.5	18.9
15	18.6	22.8	19.2	13.8	39.3	21.6	11.7	19.5	18.6	29.85	20.4	30	45	18	17.85	0	29.1	30	30.15	33
16	29.7	10.5	26.4	15.45	42	39.6	36	36.15	14.4	7.5	21.9	3.6	19.5	35.7	13.8	36	0	6.6	4.8	4.2
17	24.3	10.8	32.85	19.8	35.7	42	41.7	39	15.9	6.9	24	6.3	21.6	38.7	15.6	39.15	9	0	13.8	13.2
18	23.1	15	25.8	12.6	48	39	35.7	36.3	13.8	9	21.6	5.7	24	36	19.5	28.8	3.9	8.25	0	8.1
19	25.8	11.4	27.9	18.6	51.15	39.3	45	38.7	16.2	5.7	24.9	6	20.4	39	16.5	35.7	4.5	7.5	6.3	0

1. Rute dimulai dari depo (node 0). Cari jarak terdekat dari depo ke pelanggan *linehauls* dengan memastikan time windows dan juga kapasitas kendaraan.
2. Kendaraan tidak diperbolehkan datang sebelum pelanggan buka agar tidak terdapat waiting time.
3. Hitung waktu tempuh yang dibutuhkan untuk sampai ke pelanggan tersebut.
4. Selanjutnya ulangi langkah ke 3,4 dan 5.
5. Jika kapasitas *linehauls* terpenuhi, tentukan rute *backhauls* dengan cara yang sama mulai langkah 3,4 dan 5.
6. Cek kapasitas kendaraan.
7. Hitung total waktu tempuh yang dibutuhkan untuk sampai ke pelanggan tersebut.
8. Ulangi langkah ke 3, 4 dan 5
9. Berhenti hingga kapasitas kendaraan sudah terpenuhi dan juga time windows habis.



Gambar 1. Flowchat Nearest Neighbour

Iterasi I dimulai dari depo (node 0) pada menit ke 497.4 kemudian menuju pelanggan *linehauls* dengan jarak terdekat yaitu node 10 dengan jarak tempuh 12.6 menit. Kendaraan tiba di node 10 pada menit ke 510 dan dilayani dengan service time 15 menit, sehingga kendaraan meninggalkan node 10 pada menit ke 525. Pada node 10 terdapat *demand* 30 box maka kapasitas kendaraan menjadi 90 box. Iterasi kedua dilakukan hal yang sama dengan mencari rute terdekat dan melihat ketersediaan kapasitas kendaraan untuk pelanggan *linehauls*. Pada iterasi ke 4 kapasitas untuk pelanggan *linehauls* hanya sisa 10 box sedangkan pelanggan lain *demand* melebihi 10 maka selanjutnya dilakukan pencarian untuk pengambilan bahan baku (*backhauls*) dengan cara yang sama, yaitu dengan mencari jarak terdekat. Iterasi ke 5 dari node 1 ditemukan jarak terdekat ke pelanggan *backhaul* pada node 17 dengan jarak tempuh 7.8 menit. Kendaraan tiba di node 17 pada menit 595.2 dan langsung di layani dengan waktu service 30 menit, kendaraan keluar dari node 17 pada menit 625.2. Dengan cara yang sama iterasi terus dilanjutkan hingga iterasi 6 yang berakhir pada node 16. Pada node 16 kapasitas kendaraan *backhaul* habis sehingga kendaraan harus kembali ke depo sehingga perlu ditentukan rute baru untuk melayani sisa pelanggan yang belum terlayani. Untuk perhitungan rute 1 dapat dilihat pada tabel 4.

Tabel 4. Rute 1

Rute 1	Iterasi	Dari	Ke	Berangkat	Jarak	Tiba	Mulai	Service	Keluar	Demand	Kapasitas	Sisa Kapasitas
<i>Linehauls</i>	Iterasi 1	0	10	497.4	12.6	510	510	15	525	30	120	90
	Iterasi 2	10	8	525	7.8	532.8	532.8	15	547.8	25	90	65
	Iterasi 3	8	14	547.8	4.8	552.6	552.6	15	567.6	20	65	45
	Iterasi 4	14	1	567.6	4.8	572.4	572.4	15	587.4	35	45	10
<i>Backhauls</i>	Iterasi 5	1	17	587.4	7.8	595.2	595.2	30	625.2	15	40	25
	Iterasi 6	17	16	625.2	9	634.2	634.2	30	664.2	16	25	9
		16	0	664.2	29.7	693.9						

Setelah dilakukannya perhitungan hingga iterasi 19 didapatkan 3 rute dalam memenuhi semua layanan untuk pelanggan *linehauls* dan *backhauls* Total waktu yang dibutuhkan dalam menempuh setiap rute dapat dilihat pada tabel 5 dibawah ini:

Tabel 5. Total waktu tempuh dan jarak tempuh sebelum

Rute	Urutan	Waktu Tempuh (Menit)	Jarak (KM)
1	0-10-8-14-1-17-16-0	183.9	25.5
2	0-7-13-5-6-3-15-18-0	274.2	54.1
3	0-4-11-9-12-2-19-0	264.45	61.75
	Total	722.55	141.35

Setelah menemukan rute dengan *Nearest Neighbour* langkah berikutnya melakukan pencarian solusi baru dengan 2-opt:

1. Solusi awal untuk permasalahan VRPBTW yaitu menghitung rute awal perjalanan menggunakan *Nearest Neighbour*
2. Menentukan solusi baru. Tahap ini mencari rute baru dari rute awal yang dihasilkan pada tahap sebelumnya dengan cara melakukan *Neighbourhood search* menggunakan aturan kombinasi. Melakukan pertukaran 2 titik pada setiap rute dalam rute yang sama.

Rute awal yang ditemukan dengan *Nearest Neighbour* dimasuk dalam Tabu List pada iterasi 0 sekaligus sebagai solusi awal. Langkah kedua yaitu menentukan iterasi selanjutnya dan mencari solusi alternatif. Solusi alternatif diperoleh dengan *Neighbourhood search* menggunakan aturan kombinasi. Penyelesaian untuk mendapatkan waktu yang optimal digunakan dengan cara menukar 2 titik atau menukar posisi 2 waktu secara berurutan. Sebagai contoh pada rute 1 iterasi 1 dilakukan pertukaran node 8 dan node 10. Berikut ini adalah hasil dari pertukaran 2 titik pada rute 1 dapat dilihat pada tabel 6:

Tabel 6. Tabu list rute 1

Iterasi	Rute	Waktu Tempuh (Menit)
0	0-10-8-14-1-17-16-0	183.9
1	0-8-10-14-1-17-16-0	197.1

Rute 1 diperoleh nilai terbaik pada iterasi ke 7. Dengan waktu tempuh sebesar 180.3 menit. Pencarian solusi rute optimal lainnya juga pada rute ke 2 dan rute ke 3 dengan menggunakan cara sama yaitu melakukan pertukaran 2 titik. Pada rute 2 diperoleh nilai terbaik dengan waktu tempuh 267.9 menit dan pada rute 3 didapatkan waktu tempuh terbaik dengan 252.9 menit. Setelah dilakukan perbaikan urutan pada rute 1,2 dan 3 didapatkan total waktu tempuh yang lebih baik 21.45 menit dari sebelumnya. Perbandingan waktu tempuh dan jarak tempuh pada sebelum dan setelah perbaikan rute ditunjukkan pada tabel 7.

Tabel 7. Total waktu tempuh dan jarak tempuh setelah perbaikan

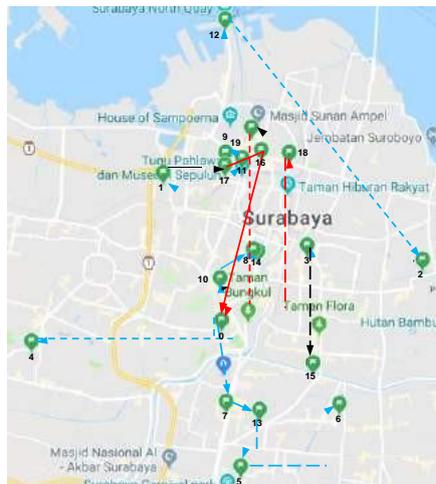
Rute	Urutan	Waktu Tempuh (Menit)	Jarak (KM)
1	0-10-8-14-1-16-17-0	180.3	24.3
2	0-7-13-5-6-3-18-15-0	267.9	52.2
3	0-4-2-9-12-11-19-0	252.9	57.9

Tabel 8 menunjukkan perbandingan antara total waktu tempuh dan jarak tempuh sebelum dan sesudah perbaikan, dari tabel tersebut setelah dilakukan perbaikan waktu tempuh rute yang baru berkurang 21.45 menit sedangkan jarak tempuh berkurang sebesar 6.95 KM.

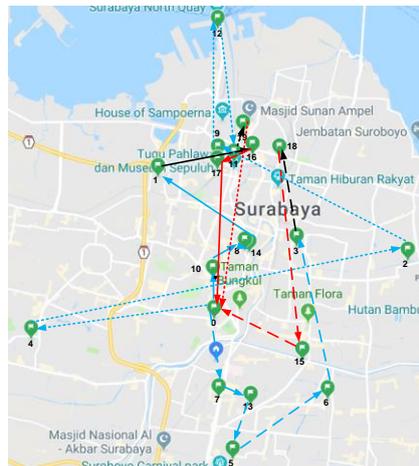
Tabel 8. Total waktu tempuh dan jarak tempuh setelah perbaikan

Rute	Sebelum		Sesudah		Effisiensi	
	Waktu Tempuh (Menit)	Jarak (KM)	Waktu Tempuh (Menit)	Jarak (KM)	Waktu Tempuh (Menit)	Jarak (KM)
1	183.9	25.5	180.3	24.3	3.6	1.2
2	274.2	54.1	267.9	52.2	6.3	1.9
3	264.45	61.75	252.9	57.9	11.55	3.85
Total Effisiensi					21.45	6.95

Rute sebelum perbaikan dapat dilihat pada gambar 2 sedangkan rute setelah perbaikan dapat dilihat pada gambar 3.



Gambar 2. Rute 1 sebelum perbaikan



Gambar 3. Rute 1 setelah perbaikan

Note :
— Linehaul
— Connector
— Backhaul
- - - Rute 1
. Rute 2
. Rute 3

Simpulan

Setelah dilakukannya perhitungan ulang rute pengiriman maka didapatkan simpulan berupa, pengiriman kue kering dan pengambilan bahan baku membutuhkan tiga kendaraan. Rute pertama yaitu melakukan pengiriman dari depo ke pelanggan

linehauls Jl. Diponegoro-Jl.Basuki Rahmad-Jl.Panglima Sudirman-Jl.Embong Malang dan menuju ke pelanggan *backhauls* Pasar Atom-PGS kemudian kembali ke depo. Total waktu yang dibutuhkan 180.3 menit dengan jarak 24.3 km. Rute kedua yaitu melakukan pengiriman dari depo ke pelanggan *linehauls* Graha Pena-Jl.Jemursari-Jl.Siwalankerto-Jl.Tenggiling-Jl.Dharmahasada dan menuju ke pelanggan *backhauls* Pasar Kapasan-Indogrosir dan kembali ke depo. Rute ketiga yaitu melakukan pengiriman dari depo ke pelanggan *linehauls* Jl.Lidah-Jl.Arif Rahman-Jl.Indrapura-Jl.Perak-Jl.Pahlawan dan menuju pelanggan *backhauls* Pasar Pabean. Total waktu tempuh rute kedua dan rute ketiga berurutan sebesar 267.9 menit dan jarak tempuh 52.2 km, 252.9 menit dan 57.9 km. Total waktu tempuh sebelum perbaikan adalah 722.55 Menit sedangkan jarak tempuh sebesar 141.35 km. Setelah dilakukan perbaikan dengan menggunakan 2-opt total waktu tempuh menjadi 701.1 menit lebih baik 21.45 menit dari rute sebelumnya. Sedangkan total jarak tempuh menjadi 134.4 km, dengan selisih 6.95 km dari sebelum perbaikan. Penelitian ini hanya menentukan rute tanpa adanya batasan kendaraan yang dimiliki oleh perusahaan, diharapkan untuk penelitian selanjutnya, adanya batasan kendaraan yang dimiliki dan juga bentuk barang yang heterogen dapat menjadi pertimbangan.

Daftar Pustaka

- [1] G. Hajar and M. D. Fauzi, "Optimasi Penentuan Rute Pengiriman Dengan Vehicle Routing Problem Simultaneous Delivery And Pickup With Split Load," *J. Ind. Eng. Oper. Manag.*, vol. 5, no. 1, pp. 84–91, Jun. 2022, doi: 10.31602/JIEOM.V5I1.7154.
- [2] H. A. Eiselt and C.-L. Sandblom, "Operations Research," 2012, doi: 10.1007/978-3-642-31054-6.
- [3] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear programming and network flows*. Wiley-Interscience, 2009.
- [4] D. Davendra, "Traveling Salesman Problem, Theory and Applications," *Travel. Salesm. Probl. Theory Appl.*, Dec. 2010, doi: 10.5772/547.
- [5] G. Lancia and P. Serafini, "Traveling Salesman Problems," *EURO Adv. Tutorials Oper. Res.*, vol. 3, no. 1, pp. 155–164, 2018, doi: 10.1007/978-3-319-63976-5_12.
- [6] J. O. Ong and . S., "Vehicle Routing Problem with Backhaul, Multiple Trips and Time Window," *J. Tek. Ind.*, vol. 13, no. 1, Jun. 2011, doi: 10.9744/JTI.13.1.1-10.
- [7] P. Toth, D. Vigo, and Society for Industrial and Applied Mathematics, *Vehicle routing : problems, methods, and applications*, Second. Philadelphia: SIAM, 2014.
- [8] L. Crépin, Y. Demazeau, O. Boissier, and F. Jacquenet, *7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2009)*, vol. 55. Berlin ; Heidelberg: Springer, 2009. doi: 10.1007/978-3-642-00487-2_56.
- [9] Charlotte Jacobs-Blecha and Marc Goetschalckx, *The Vehicle Routing Problem With Backhauls*: Atlanta: Georgia Tech Research Corporation, 1992. Accessed: Jul. 12, 2022. [Online]. Available: <https://www2.isye.gatech.edu/~mgoetsch/cali/VEHICLE/VRPB/VRPB.HTM>
- [10] J. Brandão, "A new tabu search algorithm for the vehicle routing problem with *backhauls*," *Eur. J. Oper. Res.*, vol. 173, no. 2, pp. 540–555, Sep. 2006, doi: 10.1016/J.EJOR.2005.01.042.
- [11] J. J. Santa Chávez, J. W. Escobar, M. G. Echeverri, and C. A. P. Meneses, "A heuristic algorithm based on tabu search for vehicle routing problems with *backhauls*," *Decis. Sci. Lett.*, vol. 7, no. 2, pp. 171–180, 2018, doi: 10.5267/J.DSL.2017.6.001.
- [12] G. Gutin, R. Holloway, A. Punnen, B. / Dordrecht, and / London, "The Traveling Salesman Problem And Its Variations Edited by," in *Exponential Neighborhoods and Domination Analysis for The TSP, In The Traveling Salesman Problem and Its Variations*, USA: Springer, 2007.
- [13] P. C. Pop, I. Zelina, V. Lupse, C. P. Sitar, and C. Chira, "Heuristic algorithms for solving the generalized vehicle routing problem," *Int. J. Comput. Commun. Control*, vol. 6, no. 1, pp. 158–165, 2011, doi:

10.15837/IJCCC.2011.1.2210.

- [14] P. W. Gunawan, "Enhanced nearest neighbors algorithm for design of water network," *Chem. Eng. Sci.*, vol. 84, pp. 197–206, 2012.
- [15] Suyanto, *Algoritma Optimasi (Deterministik dan Probabilistik)*. Yogyakarta: Graha Ilmu, 2010.