

Penerapan *Traveling Salesman Problem with Time Windows* dalam Pendistribusian Produk

Amar Sumarsa¹, Maya Widyastiti¹

¹Program Studi Matematika, Universitas Pakuan

Abstract. Penelitian ini bertujuan untuk menentukan rute optimal dalam pendistribusian produk sehingga diperoleh jarak terpendek. Metode yang digunakan adalah metode *Branch and Bound* dengan menggunakan model *Traveling Salesman Problem with Time Windows*. Pada model ini, sebanyak 1 kendaraan yang digunakan untuk mendistribusikan produk dari suatu depot ke toko lainnya, dan akan kembali lagi ke depot. Selain itu, terdapat *Time Windows* yang menjadi batasan waktu pelayanan di toko. Hasil yang diperoleh berupa rute pendistribusian dengan jarak sebesar 20904 meter, dengan rute depot 1-toko 10- toko 18- toko 7- toko 2- toko 11- toko 22- toko 27- toko 23- toko 31- toko 24- toko 29- toko 12- toko 28- toko 16- toko 8- toko 25- toko 26- toko 6- toko 5- toko 15- toko 14- toko 3- toko 32- toko 4- toko 9- toko 30- toko 17- toko 21- toko 20- toko 13- toko 19-depot 1.

Keyword. Distribusi, *Time Windows*, *Traveling Salesman Problem*

1. Pendahuluan

Proses distribusi adalah suatu kegiatan pemasaran yang diharapkan dapat memperlancar dan mempermudah pengiriman barang dari produsen ke konsumen (Oentoro, 2012). Dalam proses pelaksanaannya, proses distribusi sering kali terdapat kendala, yaitu bagaimana menentukan rute yang harus ditempuh supaya biaya pengiriman dapat seminimal mungkin dan mendapatkan keuntungan yang maksimal. Oleh karena itu, dibutuhkan strategi distribusi yang tepat. Salah satu strategi yang dapat dilakukan adalah dengan menentukan rute pengiriman yang terpendek sehingga dapat meminimasi biaya distribusi.

Khalifah Sejahtera Mandiri (KSM) merupakan salah satu Usaha Mikro Kecil dan Menengah (UMKM) yang bergerak di bidang produksi makanan ringan yang terletak di Kota Bogor, Jawa Barat. Dalam menjalankan usahanya, KSM perlu memasarkan produk dengan cara melakukan pendistribusian produk ke berbagai toko dan warung setiap harinya. KSM mengharapkan biaya distribusi produk yang harus dikeluarkan seminimum mungkin sehingga dapat memberikan keuntungan yang maksimal. Secara umum masalah yang dihadapi oleh KSM dalam melakukan pendistribusian produknya antara lain banyaknya permintaan yang berbeda-beda untuk setiap toko, dan adanya batasan waktu pengiriman untuk setiap toko. Proses distribusi yang selama ini dilakukan oleh KSM belum optimal karena pendistribusian dilakukan hanya untuk memenuhi permintaan setiap toko tanpa mempertimbangkan jarak, atau biaya dan waktu tempuh pendistribusian. Oleh karena itu perlu dilakukan pencarian solusi sehingga KSM dapat melakukan proses distribusi secara optimal. Salah satu cara yang dapat digunakan untuk memecahkan masalah ini adalah *Traveling Salesman Problem* (TSP).

Permasalahan TSP (*Traveling Salesman Problem*) merupakan suatu model optimasi dalam penentuan rute yang optimal dimana terdapat seorang pengirim barang (*salesman*) harus mengunjungi semua

pelanggan yang ada, dengan aturan setiap pelanggan hanya dikunjungi satu kali, dan proses pengiriman harus dimulai dari dan kembali ke kota awal (Yumalia, 2017). Tujuan dari permasalahan ini adalah menentukan rute pendistribusian dengan jarak total atau biaya atau waktu tempuh yang paling minimum. Penelitian sebelumnya tentang *Traveling Salesman Problem* sudah banyak dilakukan dengan berbagai metode, antara lain TSP dengan Jaringan Saraf *Self Organizing* telah diteliti oleh Puspitorini (2008). Pada penelitian ini, rute perjalanan terpendek yang diperoleh sangat dipengaruhi oleh parameter pelatihan, seperti ϕ , θ , momentum dan η , dan keadaan lainnya sehingga apabila proses ini dilakukan beberapa kali dengan data yang sama akan memungkinkan memperoleh jalur yang berbeda. Selain itu, Kusri, dkk (2007) juga pernah melakukan penelitian mengenai TSP dengan *algoritma Cheapest Insertion Heuristics* dan *Basis Data*. Pada penelitian ini dapat disimpulkan bahwa *basis data* dapat digunakan sebagai *tools* dalam mengimplementasikan algoritma *Cheapest Insertion Heuristics*. Namun, implementasi algoritma ini belum dapat menyelesaikan masalah pencarian rute dengan 2 kota yang memiliki bobot yang berbeda dengan melihat arah dan jika ada 2 kota yang tidak terhubung. Penelitian lain mengenai TSP juga pernah dilakukan oleh Andri, dkk (2013) dengan metode *Artificial Bee Colony*. Hasil penelitian ini menunjukkan semakin besarnya parameter yang diinput, berupa nilai koloni, iterasi dan alpha akan berdampak semakin optimal panjang jalur yang diperoleh, sedangkan nilai K (faktor skalar), λ , dan β yang diinput terlalu besar akan diperoleh panjang jalur yang kurang optimal. Waktu proses menggunakan algoritma ini lebih cepat dibandingkan dengan proses manual. TSP dengan algoritma Genetika telah dilakukan oleh Permanasari, dkk, (2006). Pada penelitian ini menggunakan program MATLAB dalam menyelesaikan masalah TSP. Hasil yang diperoleh cukup baik dan mendekati optimal dengan panjang jalur terbaik 25.9901. Pada penelitian ini, penentuan rute pendistribusian dengan menerapkan *Traveling Salesman Problem with Time Windows* karena adanya batasan waktu toko dapat dilayani dan proses penyelesaiannya menggunakan metode *Branch and Bound*. Tujuan dari penelitian ini untuk menentukan rute optimal dalam pendistribusian produk sehingga diperoleh jarak terpendek dan proses penyelesaiannya menggunakan bantuan program LINGO 11.0.

2. Metode Penelitian

Traveling Salesman Problem

Permasalahan TSP (*Traveling Salesman Problem*) merupakan masalah penentuan rute optimal dimana terdapat seorang pengirim barang (*salesman*) harus mengunjungi semua pelanggan yang ada, dengan aturan setiap pelanggan hanya dikunjungi satu kali, dan proses pengiriman harus dimulai dari dan kembali ke kota awal (Yumalia, 2017). Tujuan dari permasalahan ini adalah menentukan rute pendistribusian dengan jarak total atau biaya atau waktu tempuh yang paling minimum.

Menurut Smith (1996), misalkan permasalahan TSP melibatkan kota $1, 2, 3, \dots, n$ dan c_{ij} adalah jarak atau biaya dari kota i ke kota j dan ditentukan variabel keputusan:

$$x_{ij} = \begin{cases} 1 & \text{jika terdapat perjalanan dari kota } i \text{ ke kota } j \\ 0 & \text{jika selainnya} \end{cases}$$

Fungsi tujuan:

$$\min \sum_{i=1}^N \sum_{j=1}^N c_{ij} x_{ij}$$

Kendala:

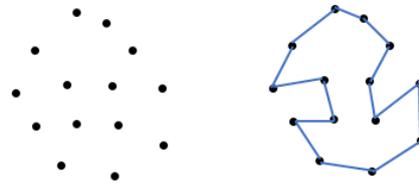
$$\sum_{i=1}^N x_{ij} = 1 \quad (j = 1, 2, 3, \dots, n)$$

$$\sum_{j=1}^N x_{ij} = 1 \quad (i = 1, 2, 3, \dots, n)$$

$$u_i - u_j + nx_{ij} \leq n - 1 \quad (i \neq j = 2, 3, \dots, n)$$

Kendala (1) dan (2) memastikan bahwa rute yang dihasilkan mendatangi setiap kota satu kali dan meninggalkan kota tersebut satu kali. Kendala (3) memastikan bahwa rute yang dibentuk meliputi semua

kota dan tidak terjadinya *subtour* (rute yang melibatkan hanya sebagian kota). Gambaran input dan output dari TSP diberikan pada Gambar 1.



Gambar 1. Input dan Output dari TSP

Metode *Branch and Bound*

Metode *Branch and Bound* pertama kali diperkenalkan tahun 1960 oleh A.H. Land dan A.G. Doig. Metode ini merupakan salah satu metode untuk menyelesaikan masalah optimisasi pemrograman linier. Metode ini akan menghasilkan variable-variabel keputusan yang bernilai *integer* (bilangan bulat) (Siswanto, 2007). Menurut Winston (2004), prinsip dasar metode ini adalah memecah daerah fisibel suatu masalah pemrograman linier relaksasi dengan membuat *subproblem* baru. Ada dua konsep yang digunakan pada metode ini, yaitu:

1. *Branching*: membatasi penyelesaian optimisasi yang menghasilkan bilangan pecahan dengan membuat cabang (*branch*) atas dan cabang bawah untuk setiap variabel keputusan yang memiliki nilai pecahan agar bernilai berupa bilangan bulat sehingga setiap pembatasan akan menghasilkan cabang baru (*subproblem*) yang mungkin mengarah ke solusi
2. *Bounding*: proses untuk mencari batas atas (masalah minimisasi) dan batas bawah (masalah maksimisasi) untuk solusi optimal pada *subproblem* yang mengarah ke solusi.

3. Hasil Penelitian

Data

Data yang digunakan pada penelitian ini merupakan data yang diperoleh dari UMKM Khalifah Sejahtera Mandiri/depot (diberi indeks 1) dan toko-toko tujuan pengiriman produk sebanyak 31 toko (diberi indeks 2-32). Data jarak antar toko diambil menggunakan google maps dan diasumsikan membentuk matriks simetrik, sehingga jarak dari toko i ke toko j sama dengan jarak dari toko j ke toko i $d_{ij} = d_{ji}$. Diasumsikan juga kecepatan dari kendaraan adalah 30 km/jam atau 500 meter/menit, sehingga waktu tempuh dapat dihitung dengan rumus $t_{i,j} = \frac{d_{ij}}{500}$. Selain itu, karena toko memiliki batasan waktu dalam melakukan pelayanan, maka dibutuhkan data waktu pelayanan di setiap toko dan waktu tercepat dan terlama toko dapat dilayani. Data waktu pelayanan, waktu tercepat dan terlama toko dapat dilayani dapat dilihat pada Tabel 1.

Tabel 1. Data waktu pelayanan, waktu tercepat dan terlama toko dapat dilayani

| Toko ke- | $c(i)$ | $l(i)$ | $y(i)$ | Toko ke- | $c(i)$ | $l(i)$ | $y(i)$ | Toko Ke- | $c(i)$ | $l(i)$ | $y(i)$ |
|----------|--------|--------|--------|----------|--------|--------|--------|----------|--------|--------|--------|
| 1 | 0 | 0 | 0 | 12 | 90 | 300 | 5 | 23 | 120 | 540 | 5 |
| 2 | 0 | 300 | 5 | 13 | 90 | 300 | 5 | 24 | 120 | 540 | 5 |
| 3 | 0 | 300 | 5 | 14 | 90 | 300 | 5 | 25 | 120 | 540 | 5 |
| 4 | 0 | 300 | 5 | 15 | 90 | 300 | 5 | 26 | 120 | 540 | 5 |
| 5 | 60 | 300 | 5 | 16 | 90 | 300 | 5 | 27 | 120 | 540 | 5 |
| 6 | 60 | 300 | 5 | 17 | 90 | 300 | 5 | 28 | 120 | 540 | 5 |
| 7 | 60 | 300 | 5 | 18 | 90 | 300 | 5 | 29 | 120 | 540 | 5 |
| 8 | 60 | 300 | 5 | 19 | 90 | 540 | 5 | 30 | 120 | 540 | 5 |
| 9 | 60 | 300 | 5 | 20 | 90 | 540 | 5 | 31 | 120 | 540 | 5 |
| 10 | 60 | 300 | 5 | 21 | 90 | 540 | 5 | 32 | 120 | 540 | 5 |
| 11 | 60 | 300 | 5 | 22 | 120 | 540 | 5 | | | | |

Formulasi Matematika

Parameter

d_{ij} = jarak dari toko i ke toko j (satuan: meter)

t_{ij} = waktu tempuh dari toko i ke toko j ; $t_{i,j} = \frac{d_{ij}}{500}$ (satuan: menit)

b_i = waktu toko i dilayani

y_i = lama pelayanan di toko i

c_i = waktu tercepat toko i dapat dilayani

l_i = waktu terlama toko i dapat dilayani

M = konstanta riil yang nilainya relatif besar

Variabel Keputusan

$$x_{ij} = \begin{cases} 1, & \text{jika toko } i \text{ dilayani setelah toko } j \\ 0, & \text{selainnya} \end{cases}$$

Fungsi Tujuan

$$\min \sum_{i=1}^{32} \sum_{j=1}^{32} d_{ij} x_{ij}$$

Fungsi Objektif

1. Setiap toko hanya dilayani satu kali

$$\sum_{\substack{i=1 \\ i \neq j}}^{32} x_{ij} = 1 \quad (j = 1, 2, 3, \dots, 32)$$
$$\sum_{\substack{j=1 \\ i \neq j}}^{32} x_{ij} = 1 \quad (i = 1, 2, 3, \dots, 32)$$

2. Waktu toko dapat mulai dilayani

$$b_i + y_i + t_{i,j} - M(1 - x_{i,j}) \leq b_j$$
$$\forall i = 1, 2, 3, \dots, 32$$
$$\forall j = 2, 3, \dots, 32$$
$$i \neq j$$

3. Memastikan toko akan dilayani pada waktu yang telah ditentukan

$$c_i \leq b_i, \forall i = 2, 3, \dots, 32$$
$$b_i + y_i \leq l_i, \forall i = 2, 3, \dots, 32$$

Hasil

Formulasi matematika yang telah dibentuk kemudian diselesaikan dengan bantuan program LINGO 11.0. Adapun hasil rute yang diperoleh dengan menggunakan metode *Branch and Bound* dapat dilihat pada Tabel 1. Berdasarkan Tabel 1, jarak yang ditempuh dalam mendistribusikan produk dari KSM ke pelanggan adalah sebesar 20904 meter atau 20,904 kilometer setiap harinya. Adapun rute yang ditempuh adalah depot 1 (KSM) – toko 10 – toko 18 – toko 7 – toko 2 – toko 11 – toko 22 – toko 27 – toko 23 – toko 31 – toko 24 – toko 29 – toko 12 – toko 28 – toko 16 – toko 8 – toko 25 – toko 26 – toko 6 – toko 5 – toko 15 – toko 14 – toko 3 – toko 32 – toko 4 – toko 9 – toko 30 – toko 17 – toko 21 – toko 20 – toko 13 – toko 19 – depot 1 (KSM). Rute dengan menggunakan metode *Branch and Bound* diperoleh hasil yang lebih baik (jarak tempuh yang lebih kecil) jika dibandingkan dengan rute yang biasa ditempuh oleh UMKM KSM yang tanpa menggunakan metode *Branch and Bound*. Rute tanpa menggunakan metode *Branch and Bound* dapat dilihat pada Tabel 3, sedangkan perbandingan rute yang diperoleh dari

penggunaan metode *Branch and Bound* dan tanpa menggunakan metode *Branch and Bound* dapat dilihat pada Tabel 4.

Tabel 2. Hasil Rute yang diperoleh dengan menggunakan Metode *Branch and Bound*

| Dilayani Ke- | Toko | Total Jarak | Dilayani Ke- | Toko | Total Jarak |
|--------------|------|-------------|--------------|------|-------------|
| 1 | 1 | 0 | 18 | 26 | 11425 |
| 2 | 10 | 780 | 19 | 6 | 12015 |
| 3 | 18 | 1624 | 20 | 5 | 12639 |
| 4 | 7 | 2329 | 21 | 15 | 13277 |
| 5 | 2 | 3084 | 22 | 14 | 13944 |
| 6 | 11 | 3866 | 23 | 3 | 14823 |
| 7 | 22 | 4397 | 24 | 32 | 15434 |
| 8 | 27 | 5153 | 25 | 4 | 16079 |
| 9 | 23 | 5977 | 26 | 9 | 16678 |
| 10 | 31 | 6486 | 27 | 30 | 17361 |
| 11 | 24 | 7240 | 28 | 17 | 18082 |
| 12 | 29 | 7843 | 29 | 21 | 18813 |
| 13 | 12 | 8830 | 30 | 20 | 19370 |
| 14 | 28 | 9186 | 31 | 13 | 19895 |
| 15 | 16 | 9719 | 32 | 19 | 20403 |
| 16 | 8 | 10337 | 33 | 1 | 20904 |
| 17 | 25 | 10881 | | | |

Tabel 3 Hasil Rute yang diperoleh tanpa menggunakan Metode *Branch and Bound*

| Dilayani Ke- | Toko | Total Jarak | Dilayani Ke- | Toko | Total Jarak |
|--------------|------|-------------|--------------|------|-------------|
| 1 | 1 | 0 | 18 | 18 | 10274 |
| 2 | 19 | 501 | 19 | 29 | 11081 |
| 3 | 13 | 1009 | 20 | 4 | 11796 |
| 4 | 20 | 1534 | 21 | 9 | 12395 |
| 5 | 21 | 2091 | 22 | 30 | 13078 |
| 6 | 17 | 2822 | 23 | 3 | 14094 |
| 7 | 16 | 3388 | 24 | 32 | 14705 |
| 8 | 28 | 3921 | 25 | 22 | 15634 |
| 9 | 12 | 4277 | 26 | 11 | 16165 |
| 10 | 15 | 5033 | 27 | 31 | 17395 |
| 11 | 5 | 5671 | 28 | 24 | 18149 |
| 12 | 6 | 6295 | 29 | 10 | 18918 |
| 13 | 26 | 6885 | 30 | 23 | 20577 |
| 14 | 25 | 7429 | 31 | 27 | 21401 |
| 15 | 8 | 7973 | 32 | 14 | 23924 |
| 16 | 2 | 8814 | 33 | 1 | 24854 |

Tabel 4. Perbandingan Rute Dengan dan Tanpa menggunakan Metode *Branch and Bound*

| Jarak dengan Metode <i>Branch and Bound</i> | Jarak tanpa Metode <i>Branch and Bound</i> | Selisih Jarak | Persentase |
|--|---|----------------------|-------------------|
| 20904 meter | 24854 meter | 3950 meter | 18,9% |

Berdasarkan Tabel 4, dapat dilihat bahwa jarak tempuh dengan menggunakan metode *Branch and Bound* menghasilkan solusi yang lebih baik (jarak terpendek) daripada jarak tempuh tanpa menggunakan metode *Branch and Bound*, dengan selisih jarak 3950 meter atau sebesar 18.9%. Oleh karena itu, rute pendistribusian produk dari KSM dapat menggunakan jarak yang menerapkan metode *Branch and Bound* sehingga waktu yang ditempuh semakin kecil dan keuntungan yang diperoleh akan semakin besar.

Kesimpulan

Model *Traveling Salesman Problem with Time Windows* dapat diselesaikan dengan menggunakan metode *Branch and Bound*. Jarak tempuh yang diperoleh sebesar 20904 m atau 20,904 km. Adapun rute pendistribusiannya yaitu depot 1-toko 10- toko 18- toko 7- toko 2- toko 11- toko 22- toko 27- toko 23- toko 31- toko 24- toko 29- toko 12- toko 28- toko 16- toko 8- toko 25- toko 26- toko 6- toko 5- toko 15- toko 14- toko 3- toko 32- toko 4- toko 9- toko 30- toko 17- toko 21- toko 20- toko 13- toko 19-depot 1. jarak tempuh dengan menggunakan metode *Branch and Bound* menghasilkan solusi yang lebih baik (jarak terpendek) daripada jarak tempuh tanpa menggunakan metode *Branch and Bound*, dengan selisih jarak 3950 meter.

Daftar Pustaka

- [1] Oentoro D 2012 *Manajemen Pemasaran Modern* (Yogyakarta: LaksBang PRESSindo)
- [2] Yumalia A 2017 *Minimasi Biaya Distribusi dengan Menggunakan Metode Traveling Salesman Problem (TSP)* (Indonesia: Seminar Nasional Sains dan Teknologi) p 1-8
- [3] Puspitorini S 2008 *Penyelesaian Masalah Traveling Salesman Problem dengan Jaringan Saraf Self Organizing* vol 6 (Indonesia: Media Informatika) p 39-55
- [4] Kusrini and Istiyanto JE 2007 *Penyelesaian Traveling Salesman Problem dengan Algoritma Cheapest Insertion Heuristics dan Basis Data* vol 8 (Indonesia: Jurnal Informatika) p 109-114
- [5] Andri, Suyandi, Winwin 2013 *Aplikasi Traveling Salesman Problem dengan Metode Artificial Bee Colony* vol 14 (Indonesia: JSM STMIK Mikroskil) p 59-68
- [6] Permanasari Y and Salin RA 2006 *Representasi Jalur (Path) pada Traveling Salesman Problem untuk Menentukan Jalur Terpendek Menggunakan Algoritma Genetika* vol 6 (Indonesia: Jurnal Matematika) p 55-62
- [7] Smith K 1996 *An argument for Abandoning the Traveling Salesman Problem as a Natural Network Benchmark* vol 7 (IEEE Transaction on Natural Network) p 1542-1544
- [8] Siswanto 2007 *Operations Research* (Jakarta: Erlangga)
- [9] Winston WL 2004 *Operations Research Applications and Algorithms Ed ke-4* (New York: Duxbury)

Ucapan terima kasih

Penulis mengucapkan terima kasih kepada Yayasan Pakuan Siliwangi atas pendanaan hibah internal Tahun 2022 yang diberikan kepada penulis. Penulis juga mengucapkan terima kasih kepada UMKM Khalifah Sejahtera Mandiri atas kesempatan dalam pengambilan data penelitian ini.