



Program Dinamis Pada Penentuan Rute Kendaraan Dengan *Time Windows*

Mirta Fera^{1*}, Irwan Endrayanto²

¹Universitas Maritim Raja Ali Haji, Kota Tanjungpinang, Kepulauan Riau 29115, Indonesia

²Universitas Gadjah Mada, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55281, Indonesia

Pengiriman: 5 Agustus 2018; Diterima: 29 September 2018; Publikasi: 30 September 2018

DOI: <https://doi.org/10.31629/jg.v3i2.511>

Abstrak

Penentuan rute armada merupakan salah satu permasalahan optimisasi kombinatorik yang memiliki pengaruh pada distribusi barang. Pengiriman barang cepat busuk (*perishable good*) seperti produk darah, dengan karakteristik jarak tempuh yang pendek memungkinkan untuk dilakukan dengan satu kendaraan. Terdapat kendala *time windows* pada pelanggan dan depot yang membatasi pengiriman. Masalah penentuan rute dalam penelitian ini dipandang sebagai *single vehicle routing problem* dengan *time windows*. Penelitian ini bertujuan untuk mendeskripsikan algoritma yang ditulis berdasarkan program dinamis untuk masalah penentuan rute kendaraan dengan *time windows*. Pada algoritma diterapkan tes yang bertujuan meningkatkan performa algoritma. Pada bagian akhir diberikan contoh penyelesaian masalah penentuan rute kendaraan dengan *time windows* menggunakan algoritma.

Kata kunci: penentuan rute kendaraan; program dinamis; algoritma eksak

Abstract

Routing problem is kind of combinatoric optimization problem that has an influence on the distribution of goods. Delivery of perishable good such as blood products with short travel characteristics makes it possible to do with one vehicle. There are time-windows constraints on customer and depots that limit delivery. This research aims to describe algorithms written based on dynamic programs for the problem of determining vehicle routes with time windows. In the algorithm applied a test that aims to improve the performance of the algorithm. In the end, given an example of solving the problem of determining a vehicle route with time windows using an algorithm.

Keywords: vehicle routing problem; dynamic programming; exact algorithm

I. Pendahuluan

Masalah optimisasi kombinatorik yang sering ditemui diantaranya yaitu masalah penentuan rute dan penjadwalan kendaraan yang berpengaruh pada sistem logistik. Penentuan rute pada distribusi barang oleh kendaraan yang berangkat dari depot ke sejumlah pelanggan yang

tersebar secara geografis untuk meminimumkan biaya perjalanan dimana memenuhi kendala masalah yang diberikan dikenal dengan *Vehicle Routing Problem* (VRP) atau masalah penentuan rute kendaraan.

Topik VRP dan variasinya sangat berkembang dalam beberapa tahun terakhir.

*Penulis Korespondensi

Email Address : mirtafera0901@umrah.ac.id

Handphone : +62 852 6330 2756

Penambahan variasi dari kendala yang dihadapi diantaranya waktu pelayanan yang dikenal dengan *time windows*. VRP dengan jumlah permintaan diketahui, kendala kapasitas serta kendala *time windows* dikenal dengan VRPTW (Lee, Tan, Ou, & Chew, 2003). Kendala *time windows* terdapat pada depot dan pelanggan. *Time windows* depot didefinisikan sebagai batas waktu kendaraan berangkat dan kembali ke depot selanjutnya. Pada pelanggan *time windows* didefinisikan sebagai interval waktu yang ditentukan untuk menerima barang.

Beberapa peneliti yang membahas VRP dan *time windows* dijelaskan oleh Batista. Pembahasan berhubungan dengan bentuk umum formulasi masalah, tujuan yang ingin dicapai seperti meminimumkan jumlah kendaraan, total jarak tempuh, biaya operasional dan motivasi diterapkannya *multi-objective* serta metode menemukan solusi. Dari pembahasan ini dapat dipahami bahwa dalam penentuan rute dimungkinkan untuk menerapkan satu tujuan atau banyak tujuan berdasarkan pertimbangan tertentu (Melián-batista, Santiago, Angelbello, & Alvarez, 2014).

Algoritma heuristik, metaheuristik dan eksak merupakan algoritma yang banyak digunakan dalam menyelesaikan masalah penentuan rute. Heuristik dan metaheuristik cenderung sulit karena banyaknya langkah yang harus dilakukan. Karakteristik algoritma ini bekerja dengan sifat *trial and error* dan biasanya telah memiliki intuisi untuk dugaan solusi. Dipahami bahwa heuristik dan metaheuristik cepat dalam menemukan solusi namun belum tentu optimal. Selanjutnya pada algoritma eksak digunakan salah satu prinsip yaitu relaksasi Lagrange, program dinamis, dan generasi kolom.

Metode eksak untuk VRPTW sudah menjadi perhatian para peneliti sejak lama. Kolen et al pada tahun 1987 menerapkan program dinamis dan menggunakan *branch and bound* untuk solusi optimal. Desrochers, Desrosier dan Salomon pada tahun 1992 menggunakan model partisi himpunan dengan generasi kolom dalam menyelesaikan contoh.

Kohl, Desrosiers, Medsen, Solomon dan Soumis pada tahun 1997 menerapkan algoritma yang sama serta menggunakan *branch and cut* dengan bidang potong baru berdasarkan formula yang diperkuat dari ketaksamaan eliminasi subtur yang disebut *k-path cuts* (Rich, 1999).

Pertimbangan terkait karakteristik tertentu dari masalah penentuan rute juga dapat menjadi alasan dalam membuat formulasi masalah. Seperti yang terdapat dalam (Azi, Gendreau, & Potvin, 2007) menjelaskan terkait pengiriman barang dengan karakter rute yang cukup pendek dan kelompok barang cepat busuk (*perishable good*) misalnya makanan atau produk darah maka dalam prosesnya dapat dilakukan dengan satu kendaraan (*single vehicle*).

Masalah yang diberikan dapat dinyatakan sebagai berikut. Terdapat satu kendaraan (*single vehicle*) dengan kapasitas Q dan akan mendistribusikan barang dari node depot menuju himpunan pelanggan yang direpresentasikan oleh sebuah jaringan $G = (V, A)$, $V = \{0, 1, \dots, n, n+1\}$ adalah himpunan node dan A himpunan busur $A = \{(i, j) \mid i, j \in V\}$. Untuk $i \in V$ terdapat *time window* $[a_i, b_i]$ dan waktu pelayanan s_i , misalkan $V' = V - \{n+1\}$. Untuk busur $(i, j) \in A$ terdapat waktu tempuh t_{ij} dan biaya (jarak) c_{ij} . Depot dinotasikan oleh 0 atau $n+1$ sesuai dengan posisi yang menunjukkan node asal atau node akhir dengan $s_0 = s_{n+1} = 0$, $q_0 = q_{n+1} = 0$, E dan L menunjukkan waktu keberangkatan paling awal dari depot dan waktu kedatangan terakhir saat mencapai depot. Tujuan dari masalah ini adalah meminimumkan jarak tempuh dalam melayani semua pelanggan dengan memenuhi kendala *time windows*.

Berikut diberikan formulasi masalah:

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (1)$$

dengan kendala

$$\sum_{i \in V} x_{ij} = 1 \quad \forall j \in N, \quad (2)$$

$$\sum_{j \in V} x_{ij} = 1 \quad \forall i \in N, \quad (3)$$

$$\sum_{j \in N} x_{0j} = 1, \quad (4)$$

$$\sum_{i \in N} x_{i(n+1)} = 1, \quad (5)$$

$$\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{ji} = 0, \quad \forall i \in N \quad (6)$$

$$q_i \leq u_i \leq Q, \quad (7)$$

$$u_i - u_j + Qx_{ij} \leq Q - q_j, \quad \forall i, j \in V \setminus \{0\}, i \neq j \quad (8)$$

dengan $q_i + q_j \leq Q$.

$$s_i + t_{ij} - K(1 - x_{ij}) \leq s_j \quad \forall i, j \in N, \quad (9)$$

$$a_i \leq D_i \leq b_i, \quad \forall i \in V \setminus \{0\} \quad (10)$$

$$x_{ij} \in \{0,1\} \quad \forall (i, j) \in A \quad (11)$$

Pada formulasi ini, persamaan (2) dan persamaan (3) menyatakan setiap pelanggan dikunjungi dan ditinggalkan tepat satu kali. Persamaan (4) dan (5) menyatakan terdapat tepat satu busur yang menuju dan meninggalkan depot. Persamaan (6) menyatakan aliran kendaraan pada suatu node. Persamaan (7) menyatakan permintaan pelanggan tidak melebihi kapasitas kendaraan. Persamaan (8) menyatakan kendala eliminasi subtur. Serta persamaan (9) dan (10) menjamin pelayanan sesuai dengan *time windows*. Secara umum diperhatikan bahwa terdapat empat kendala utama pada formulasi masalah sesuai dengan model dasar *Vehicle Routing Problem* yang dijelaskan Toth (2002).

Azi menjelaskan algoritma *label correcting* yang dapat digunakan untuk menyelesaikan masalah lintasan dasar terpendek dengan kendala sumber daya pada graf. Sebuah *path p* dari suatu node asal *o* menuju suatu node *j* dan diberilabel $R_p = (d_p, t_p^1, \dots, t_p^l, s_p, V_p^1, \dots, V_p^n)$ dengan $L = \{1, \dots, l\}$ himpunan sumber daya, d_p

panjang *path p*, pemakaian sumber daya $k = 1, \dots, l$, s_p banyaknya node yang tidak tercapai dan $V_p^i = 1$ jika node *i* tidak tercapai, 0 untuk keadaan lain. Jika *p* dan *p'* dua path dari node asal menuju node *j* dengan label R_p dan $R_{p'}$, maka path *p* mendominasi *p'* jika dan hanya jika $d_p \leq d_{p'}$, $s_p \leq s_{p'}$, $t_p^k \leq t_{p'}^k$, $k = 1, \dots, l$, $V_p^i \leq V_{p'}^i$, $i = 1, \dots, n$ (Azi et al., 2007)

II. Metode Penelitian

Studi ini merupakan studi literatur yang dilakukan dalam penulisan algoritma. Masalah SVRPTW dapat dipandang sebagai kasus khusus dari *vehicle routing problem* dengan *time windows*. Berkaitan dengan karakteristik barang *perishable* yang didistribusikan, pada penelitian ini diperhatikan bahwa kapasitas kendaraan mampu menampung semua permintaan pelanggan dengan kata lain $Q \geq d(q)$.

Algoritma eksak menggunakan program dinamis dimanfaatkan untuk menyelesaikan SVRPTW. Prinsip optimalitas sebagai konsep dasar program dinamis merujuk pada pernyataan Belman yaitu “*An optimal policy has the property that whatever the initial state and initial decision are, the remaining decision must constitute an optimal policy with regard to state resulting from the first decisions*” (Belman, 1954). Adapun karakteristik serta pengembangan yang dilakukan pada program dinamis merujuk pada Cormen (2001) serta Horowitz dan Sahani (1988) dengan pendekatan perhitungan setiap tahap algoritma berupa pendekatan mundur (*backward*). Terkait program dinamis yang diterapkan pada TSP dan VRP merujuk pada Desrochers (1988), Held dan Karp (1962).

Algoritma Optimal untuk SVRPTW

Diberikan jaringan $G = (V, A)$ dengan $V = \{1, \dots, n\}$ himpunan node dan A himpunan busur. Untuk $i \in V$ terdapat *time windows* $[a_i, b_i]$ dan waktu pelayanan s_i , misalkan

$V' = V - \{n\}$. Untuk busur $(i, j) \in A$ terdapat waktu tempuh t_{ij} dan biaya (jarak) c_{ij} . Sebuah busur $(i, j) \in A$ dikatakan fisibel jika total penjumlahan waktu awal suatu node i ditambah waktu pelayanan dan waktu tempuh dari i menuju j kurang dari sama dengan batas waktu terakhir *time windows* node j .

Setiap *state* memuat pemakaian (*consumtion*) sumber daya (*resources*), dalam kasus ini yaitu *time windows* pada lintasan yang bersesuaian. Penambahan parameter waktu pada *state* dilakukan bertujuan untuk menemukan bagian rute yang memenuhi kendala waktu dan berpeluang sebagai pembentuk solusi. *State* (S, i, t) menunjukkan lintasan fisibel yang berawal dari node asal, mengunjungi semua node anggota S , berakhir di $i \in S$ dan mulai melayani i pada waktu t . S merupakan himpunan node yang tidak berurutan dan dikunjungi masing-masing satu kali.

Path yang fisibel dapat disyaratkan dengan menetapkan perjalanan yang dilakukan, dengan kata lain aturan perpanjangan (*extention rule*) untuk sumber daya *time window* fisibel yaitu jika j dikunjungi langsung setelah i , maka waktu mulai pelayanan pada j memenuhi $t_j = \max\{a_j, t_i + s_i + t_{ij}\}$ Selanjutnya *state* (S, i, t) dapat diperpanjang menuju j dalam arti *state* $(S \cup \{j\}, j, \max\{t + s_i + t_{ij}\})$ dapat dibentuk jika $t + s_i + t_{ij} \leq b_j$.

Solusi fisibel untuk SVRPTW direpresentasikan oleh sebuah himpunan terurut $H = (1, i_1, i_2, \dots, i_n, 1)$ dengan node 1 menempati dua posisi yaitu 1 dan $n+1$ dan $i_k \in H$ merupakan node pada posisi $k (2 \leq k \leq n)$.

Formulasi Program Dinamis

Didefinisikan $F(S, i, t)$ jarak minimum sebuah lintasan yang berangkat mulai dari node 1, melalui setiap node $S \subseteq V'$ tepat satu kali dan berakhir pada suatu node $i \in S$ dan siap melayani node i pada waktu t . Dumas (1995)

menjelaskan $F(S, i, t)$ untuk kasus penentuan rute kendaraan dengan satu kendaraan dan *time windows* dihitung menggunakan persamaan rekuren berikut.

$$F(S, j, t) = \min_{(i,j) \in A} \{F(S - \{j\}, i, t') + c_{ij} \mid t \geq t' + s_i + t_{ij}, a_i \leq t' \leq b_i\},$$

untuk semua $S \subseteq V', j \in S$, dan $a_j \leq t \leq b_j$

.Inisialisasi dari persamaan diatas yaitu:

$$F(\{1, j\}, j, t) = c_{1j} \text{ jika } (i, j) \in A \text{ dan}$$

$$F(\{1, j\}, j, t) = \infty \text{ jika lainnya,}$$

$$\text{dengan } a_j \leq t \leq b_j, t = \max\{a_1 + s_1 + t_{1j}, a_j\}.$$

Selanjutnya solusi optimal diberikan oleh

$$\min_{i, n \in A} \{F(V', i, t) + c_{im} \mid t \leq b_n - t_m - s_i\}.$$

(Dumas, Desrosiers, Gelinas, & Solomon, 1995).

Formulasi persamaan rekuren program dinamis di atas mendefinisikan masalah lintasan terpendek pada *state* graf dengan *state* (S, i, t) sebagai node dan transisi dari suatu node ke node berikutnya sebagai busur. Perhitungan yang dilakukan secara *forward* dengan *stage* yang dikerjakan $s, s = 1, \dots, n-1$, yang menunjukkan path dengan panjang s .

Langkah-Langkah Menyelesaikan Masalah

Penyelesaian *vehicle routing problem with time window* menggunakan program dinamis dengan perhitungan *forward* dengan mengikuti langkah yang dikembangkan Dumas seperti pada penjelasan di atas sebagai berikut:

Langkah 1. Inisialisasi

Bentuk *state* (S, j, t) , dan $j \in S$

$$F(\{1, j\}, j, t) = c_{1j} \text{ jika } (i, j) \in A,$$

$$F(\{1, j\}, j, t) = \infty \text{ jika } (i, j) \notin A,$$

$$\text{dengan } a_j \leq t \leq b_j, t = \max\{a_1 + s_1 + t_{1j}, a_j\}.$$

Langkah 2. Ekspansi setiap *state* anggota himpunan *state* $k-1$.

Untuk setiap (S, i, t) pada himpunan *state* $k-1$, ulangi Langkah 3.

Langkah 3. Membangun *state* pada himpunan *state* k yang bisa dicapai dari *state* (S, i, t) pada himpunan *state* $k-1$.

$$F(S, j, t) = \min_{(i,j) \in A} \{F(S - \{j\}, i, t') + c_{ij} \mid t \geq t' + s_i + t_{ij}, a_i \leq t' \leq b_i\},$$

untuk $S \subseteq V', j \in S$, dan $a_j \leq t \leq b_j$. Diperoleh *state*

$(S', j, t^*) = (S \cup \{j\}, j, \max\{a_j, t + s_i + t_{ij}\})$
 sebagai perpanjangan yang fisibel dari dari state (S, i, t) . (S', j, t^*) ditolak jika salah satu dari kondisi di bawah ini terjadi :

(Dominance check)

Jika terdapat state (S, i, t^1) dan (S, i, t^2) , state (S, i, t^2) dapat dieliminasi jika $t^1 \leq t^2$ dan $F(S, i, t^1) \leq F(S, i, t^2)$.

(Feasibility check)

Jika $t + s_i + t_{ij} > b_j$.

Langkah 4. Pandang $k \leftarrow k + 1$, jika $k \leq n$ maka lanjut ke Langkah 2.

Langkah 5. Menghitung solusi optimal.

Solusi optimal jarak z diberikan oleh :

$$z = \min(i, n) \in A \min_{a_i \leq t \leq b_i} \{F(V', i, t) + c_{in} \mid t \leq b_n - t_{in} - s_1\}.$$

penentuan pembentuk rute yang memberikan solusi dilakukan dengan *backtracking* dari state yang memberikan nilai minimum di atas.

III. Pembahasan

Agar algoritma dapat lebih mudah dipahami, berikut diberikan contoh numerik penggunaan algoritma pada contoh kasus distrsibusi produk darah dari PMI sebagai unit donor dan transfusi darah ke beberapa Bank Darah Rumah Sakit (BDRS) yang tersebar di wilayah kerja PMI.

Armada ambulan harus melakukan pengiriman kantong darah ke dari PMI ke BDRS sesuai permintaan masing-masing yang dilakukan dalam interval waktu tertentu. Mengingat karakteristik barang yang dikirimkan dan kondisi wilayah, dipandang satu armada kendaraan cukup untuk melakukannya.

Dalam hal penerimaan barang BDRS selaku pelanggan dapat menetapkan waktu sesuai keadaan mereka dalam hal ini jam operasional. Waktu penerimaan ini dinyatakan dalam interval yang menunjukkan batasan awal dan akhir menerima pelayanan yang diizinkan.

Diberikan suatu jaringan $G = (V, A)$ yang merupakan graf lengkap berarah. Himpunan $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$ dengan

$V' = \{1, 2, 3, 4, 5, 6, 7\}$. Himpunan node yang dilalui dimana $S \subseteq V'$. Node $i = 2, 3, 4, 5, 6, 7$ merupakan node yang akan dikunjungi selanjutnya node depot yaitu 1 dan 8 yang merupakan node yang sama yaitu depot dan menunjukkan awal dan akhir perjalanan. Untuk kepentingan penulisan, ditetapkan 1 sebagai node awal dan 8 sebagai node akhir.

Selanjutnya jarak c_{ij} , waktu tempuh t_{ij} antara node i dan j untuk pasangan $i, j \in V$ dan *time window* $[a_i, b_i]$ untuk setiap node i . Jarak dan waktu tempuh memenuhi ketaksamaan segitiga, $c_{ij} \leq c_{ik} + c_{kj}$, $t_{ij} \leq t_{ik} + t_{kj}$ untuk $i, j, k \in V$.

Dengan menetapkan kecepatan kendaraan yaitu satu satuan waktu kecepatan maka $c_{ij} = t_{ij}$. Jam operasional ditetapkan 12 jam dengan pertimbangan pada kegiatan distribusi. Selanjutnya dilakukan pembobotan waktu dengan bilangan bulat dimana 1 untuk 30 menit pertama dan seterusnya dan waktu pelayanan s_i selama 30 menit dengan bobot 1.

Berikut diberikan tabel 1 yang memuat jarak antar node dan tabel 2 yang memuat *time windows*.

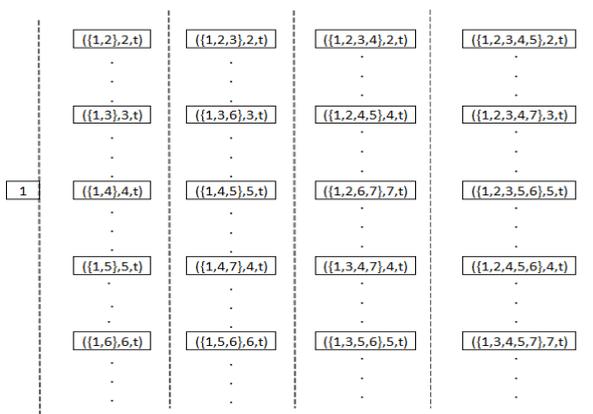
Tabel 1.
Jarak antar node

Node	1	2	3	4	5	6	7
1	0	2	3	1	2	2	2
2	2	0	2	2	3	1	2
3	3	2	0	3	1	2	2
4	1	2	3	0	2	2	2
5	2	2	3	2	0	2	2
6	2	1	2	2	2	0	1
7	2	2	2	2	2	2	0

Tabel 2.
Time windows

No	Node	Time Windows	Time Windows (bobot)
1	1	[06.00-18.00]	[1,24]
2	2	[(06.00-06.30)-(08.30-09.00)]	[1,6]
3	3	[(09.00-09.30)-(11.30-12.00)]	[7,12]
4	4	[(15.00-15.30)-(17.30-18.00)]	[19,24]
5	5	[(09.00-09.30)-(11.30-12.00)]	[7,12]
6	6	[(11.30-12.00)-(14.30-15.00)]	[11,16]
7	7	[(06.00-06.30)-(08.30-09.00)]	[1,6]

Hasil dari algoritma yang dijalankan untuk contoh di atas ditampilkan pada tabel 3. Diperhatikan bahwa persamaan rekuren yang terdapat dalam program dinamis mendefinisikan terbentuknya *state* graf (S, i, t) dengan *state* yang terbentuk sebagai node dan transisi *state* pada sebagai busur seperti pada gambar 1 di bawah. Waktu mulai pelayanan t berada dalam *time windows* untuk setiap i yaitu $a_i \leq t \leq b_i$, dengan lebar *time window* 6 dan syarat $t = \max\{a_i + s_i + t_{1j}, a_j\}$, $j \in S$ untuk *stage* 1 dan $t_j = \max\{a_j, t_i + s_i + t_{ij}\}$ untuk *stage* lainnya, lama pelayanan s_i dan waktu tempuh antar *state* t_{ij} .



Gambar 1. Pembentukan node baru berupa *state* yang membentuk *state* graf.

Sifat dominan atau *feasibility* mampu mereduksi jumlah *state* yang telah dibentuk. Sebuah *path* mendominasi *path* yang lain jika mengunjungi anggota himpunan yang sama dari node asal yang kemudian dikatakan sampai pada *state* sebagai node yang baru. *Path* yang didominasi dapat dieliminasi, sehingga *state* yang berkaitan dapat dieliminasi dan tidak digunakan dalam perhitungan berikutnya. *State* yang telah dieliminasi menyebabkan tidak ada busur dari *state* di *stage* sebelumnya ke *state* tersebut. *Backtracking* dilakukan untuk mendapatkan pembentuk rute fisibel yang menghasilkan solusi optimal. Node fisibel terakhir yang dikunjungi sebelum armada kembali ke depot adalah node 4 dengan *state* $(\{1,2,3,4,5,6,7\},4,19)$.

Berikut diberikan tabel yang memuat proses *backtracking* yang dilakukan.

Tabel 3.
Backtracking State

No	Stage	State (S, i, t)	F(S, i, t)
1	6	$(\{1,2,3,4,5,6,7\},4,19)$	11
2	5	$(\{1,2,3,5,6,7\},6,14)$	9
3	4	$(\{1,2,3,5,7\},5,11)$	7
4	3	$(\{1,2,3,7\},3,9)$	6
5	2	a. $(\{1,2,7\},7,6)$ b. $(\{1,2,7\},2,6)$	4 4
6	1	a* $(\{1,2\},2,3)$ b* $(\{1,7\},7,3)$	2 2

Terdapat dua pilihan rute yang memberikan hasil optimal berupa jarak minimum dalam masalah SVRPTW ini yaitu 1-2-7-3-5-6-4-8 atau 1-7-2-3-5-6-4-8 dengan jarak 12.

Generalisasi proses struktur optimal masalah pada penelitian ini memiliki karakteristik sebagai berikut. Masalah dapat dibagi menjadi beberapa *stage* dan ada keputusan yang akan diambil pada tiap-tiap *stage* sebagai pembentuk solusi. Tiap *stage* terdiri dari sejumlah *state* yang berhubungan pada *stage* tersebut. Dalam hal ini diberikan *state* (S, i, t) dan x_k sebagai keputusan yang diambil. Diperhatikan bahwa *cost* (dalam penelitian ini berupa jarak) akan meningkat seiring

meningkatnya *stage*, sesuai pendefinisian fungsi rekuren. Prinsip optimalitas berlaku, berarti jika bekerja pada *stage* k ke $k+1$, dapat menggunakan hasil optimum dari k tanpa harus kembali ke *stage* awal. Algoritma ini terbatas pada masalah yang memenuhi asumsi tertentu dan karakteristik di atas.

Selanjutnya generalisasi solusi dari masalah pada penelitian. Solusi dari algoritma dijamin memberikan solusi optimal untuk masalah yang memenuhi karakteristik di atas. Dari penghitungan manual yang dilakukan, hasil yang diperoleh menunjukkan hal yang sama yaitu solusi lintasan dengan jarak terpendek yang memenuhi kendala sumber daya yang ditetapkan. Kompleksitas waktu untuk state graf pada program dinamis secara umum yaitu $O(|V|^{k-2})$.

IV. Kesimpulan

Berdasarkan pembahasan di atas, dapat disimpulkan bahwa program dinamis yang digunakan pada algoritma eksak untuk menyelesaikan SVRPTW yang diterapkan pada masalah distribusi barang cepat busuk (*perishable*), dapat memberikan solusi optimal yang dibangun oleh keputusan optimal di setiap *stage*. Kerja algoritma dengan program dinamis yang menerapkan tes sifat dominan dan *feasibility*, untuk mereduksi ruang dan transisi *state* cukup efektif untuk masalah nyata dengan ukuran data kecil, dibandingkan dengan pencarian mendalam yang membutuhkan waktu lebih lama. Karakteristik barang yang didistribusikan mempengaruhi ukuran data yang digunakan, yaitu barang cepat busuk (*perishable*) seperti produk darah yang jarak tempuhnya cukup pendek dan jumlah pelanggan yang dilayani tidak banyak.

Referensi

Azi, N., Gendreau, M., & Potvin, J. (2007). An exact algorithm for a single-vehicle routing problem with time windows and multiple routes, *178*, 755–766. <https://doi.org/10.1016/j.ejor.2006.02.019>

Belman, R. (1954). *The theory of dynamic programming*, P-550.

Cormen, Thomas H;Leiserson, Charles E;Rivest, R. L. (2001). *Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein Introduction to algorithms 2001*.

Desrochers, M;Lenstra, J.K;Savelsber, M.W.P; Soumis, F. (n.d.). 5-Apr-desrochers.pdf.

Dumas, Y., Desrosiers, J., Gelinat, E., & Solomon, M. M. (1995). *An optimal algorithm for the traveling salesman problem with time windows*, (September 2015).

Horowitz, Elis;Sahni, Sartaj;Rasekaran, S. (n.d.). horowitz-and-sahani-fundamentals-of-computer-algorithms-2nd-edition.pdf.

Lee, L. H., Tan, K. C., Ou, K., & Chew, Y. H. (2003). Vehicle capacity planning system: a case study on vehicle routing problem with time windows, *33*(2), 169–178.

Melián-batista, B., Santiago, A. De, Angelbello, F., & Alvarez, A. (2014). A bi-objective vehicle routing problem with time windows: A real case in Tenerife &. *Applied Soft Computing Journal*, *17*, 140–152. <https://doi.org/10.1016/j.asoc.2013.12.012>

Rich, J. L. (1999). *A Computational Study of Vehicle Routing Application*.

Toth, P. D. (2002). [Paolo_Toht;_Daniele_Vigo]_The_Vehicle_Routing_Pro(BookZZ.

JURNAL GANTANG. September 2018; III(2): 135 – 141

p-ISSN. 2503-0671

e-ISSN. 2548-5547