Jurnal Teknik Industri, Vol. 18, No.02, Agustus 2017, pp. 191-206
ISSN 1978-1431 print / ISSN 2527-4112 online
https://doi.org/10.22219/JTIUMM.Vol18.No2.191-206

Pengembangan Algoritma *Hybrid* Metaheuristik Untuk Penentuan Rute Pengiriman Produk *Perishable*

Luki Trihardani, Oki Anita Candra Dewi
Teknik Logistik, Fakultas Teknologi Industri dan Agroindustri, Universitas Internasional Semen Indonesia
Kompleks PT. Semen Indonesia (Persero) Tbk. Jl. Veteran, Gresik Jawa Timur, 61122, (031) 3985482
*Surel: luki.trihardani@uisi.ac.id

Abstract

The decision to dispatch consumers demand has become a strategic and tactical consideration to be solved in an integrated manner. In this study, the problem of determining routing problem take case study of delivery of perishable product. The routes determination should take into account the unique characteristics of perishable products possess. Perishable products continuously decreases quality over their lifetime. The challenge for distributors is how to minimize the cost of delivering perishable products by taking into account the temperature so that it can serve a number of customers within the specified time frame. The problem of determining the route on delivery is included in the combinatorial optimization problem, thus causing this problem to be complex to be solved by the exact method. On the other hand, metaheuristic methods are increasingly being developed to be applied in the completion of combinatorial optimizations. This research started from mathematical model of perishable product delivery which pay attention to perishability (quality, temperature, quality loss) and time windows. Based on this model, this research develops the route settlement algorithm of delivery of perishable product using metaheuristic, particle swarm optimization. The algorithm development is required because route determination included in discrete issues. In addition, the development of algorithms to improve performance by combining (hybrid) algorithms, nearest neighbor and particle swarm optimization. Experiments were performed on 2 sets of Solomon data. From the experimental results with the metaheuristic hybrid algorithm is able to provide better performance than pure metaheuristik. Although the solution gap produced by these two algorithms is not very significant, but when viewed from the computation time and the number of iterations required to find the best solution, this metaheuristic hybrid algorithm can save an average time of 17 times from pure metaheuristic algorithm.

Keywords: *Perishable products, delivering, route, hybrid metaheuristics, particle swarm optimization*

Abstrak

Keputusan pengiriman permintaan sekumpulan konsumen telah menjadi pertimbangan strategis maupun taktis untuk diselesaikan secara terintegrasi. Pada penelitian ini, permasalahan penentuan rute kendaraan (routing problem) mengambil studi kasus pengiriman produk perishable. Penentuan rute tersebut harus memerhatikan karakteristik unik yan dimiliki produk perishable. Produk perishable (tidak tahan lama) memiliki karakteristik mudah rusak dan secara kontinu mengalami penurunan kualitas selama umur hidupnya. Tantangan bagi distributor adalah bagaimana meminimumkan biaya pengiriman produk perishable dengan memperhatikan temperatur sehingga mampu melayani sejumlah pelanggan dalam rentang waktu yang telah ditentukan. Permasalahan penentuan rute pada pengiriman termasuk kedalam permasalahan optimasi kombinatorial, sehingga menyebabkan permasalahan ini menjadi kompleks untuk diselesaikan dengan metode eksak. Di sisi lain, metode metaheuristik semakin berkembang untuk diaplikasikan dalam penyelesaian optimasi kombinatorial. Penelitian ini berawal dari model matematis pengiriman produk perishable yang memerhatikan perishability

Diterima 26 Juni, 2017; Direvisi 27 Juni, 2017; Disetujui 30 Juli, 2017

(penurunan kualitas, temperatur, loss kualitas) serta time windows (batas waktu pengiriman). Berdasarkan model tersebut, penelitian ini mengembangkan algoritma penyelesaian penentuan rute pengiriman produk perishable menggunakan metaheuristik, particle swarm optimization. Pengembangan algoritma ini dilakukan karena penentuan rute termasuk permasalahan diskrit. Selain itu dilakukan pengembangan algoritma untuk meningkatkan performansi dengan melakukan kombinasi (hybrid) algoritma, nearest neighbor dan particle swarm optimization. Eksperimen dilakukan pada 2 set data Solomon. Dari hasil eksperimen dengan tersebut, algoritma hybrid metaheuristik mampu memberikan performa yang lebih baik daripada metaheuristik murni. Walaupun gap solusi yang dihasilkan kedua algoritma ini tidak terlalu signifikan, namun jika dilihat dari waktu komputasi dan jumlah iterasi yang diperlukan untuk menemukan solusi terbaik, algoritma hybrid metaheuristik ini mampu menghemat waktu rata-rata 17 kali dari algoritma metaheuristik murni.

Kata kunci: *Produk perishable, pengiriman, rute, hybrid metaheuristik, particle swarm optimization*

1. Pendahuluan

Sistem pengiriman dalam suatu rantai pasok, selain memegang peran krusial dalam pendapatan perusahaan, juga berpotensi terhadap capaian kepuasan pelanggan. Fakta empiris menunjukkan kenaikan biaya pengiriman sebesar 10%, berkontribusi menurunkan volume perdagangan lebih dari 20% [1]. Bisa disimpulkan, sistem pengiriman yang efisien adalah salah satu faktor penunjang keberhasilan rantai pasok. Suatu sistem pengiriman yang baik harus mampu menyusun strategi rute perjalanan kendaraan yang *cost effective* dan meminimalisasi berbagai *handling* perjalanan yang tidak perlu.

Strategi penyusunan rute kendaraan secara sederhana diawali dari *loading* muatan kendaraan di sentral depot. Pada saat *loading* tersebut, muatan yang diangkut tidak boleh melebihi kapasitas kendaraan. Muatan ini kemudian dikirimkan ke pelanggan sesuai permintaan. Selanjutnya setelah menyelesaikan rute perjalanan, kendaraan kembali ke sentral depot. Berdasarkan gambaran tersebut, diperlukan strategi penyusunan rute yang secara efektif mampu meminimalkan biaya perjalanan pada saat pengiriman. Perlu diingat bahwa permasalahan penentuan rute tidak hanya menyangkut perencanaan operasional, tetapi juga menyangkut perencanaan baik strategis maupun taktis dalam suatu sistem distribusi.

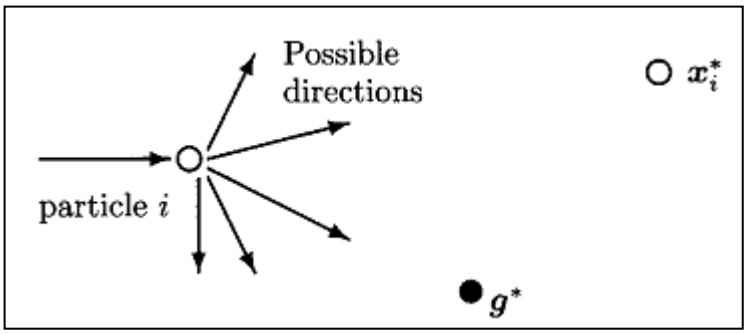
Berdasarkan cabang ilmu terkait manajemen logistik dan rantai pasok, permasalahan penyusunan rute kendaraan dalam pengiriman termasuk dalam *Vehicle Routing Problem* (VRP). *Vehicle Routing Problem* (VRP) adalah permasalahan optimasi klasik yang menjadi kunci pengelolaan distribusi dan logistik [2]. Penelitian VRP sendiri sudah berkembang lebih luas dalam berbagai komponen. Salah satu pengembangan penentuan rute kendaraan adalah VRP dengan batasan *time windows* atau interval waktu konsumen dapat dilayani, atau disebut dengan *Vehicle Routing Problem with Time Windows* (VRPTW). VRPTW sendiri adalah perluasan VRP di mana pelayanan setiap pelanggan dimulai pada rentang waktu tertentu (*time windows*) dan kendaraan harus berada pada rentang waktu tersebut untuk mendapatkan pelayanan [3].

Munculnya pertimbangan *time windows* karena disebabkan berbagai alasan, diantaranya: meminimalkan persediaan, mengurangi waktu siklus, mengimplementasikan *just-in-time*, dan lain-lain. VRPTW dianggap lebih aplikatif dan

representatif terhadap real sistem dalam suatu sistem pengiriman. Terdapat dua jenis *time window* yang dapat digunakan dalam VRPTW yaitu *hard time window* dan *soft time window*. Adanya batasan *hard time window* menyebabkan kenaikan biaya distribusi karena untuk melayani semua konsumen dibutuhkan banyak kendaraan. Batasan *soft time window* dirasa lebih aplikatif jika diterapkan dalam kasus nyata. Figliozzi [4] menyatakan bahwa VRPSTW (VRP *Soft Time Window*) dapat diaplikasikan untuk membuat VRPHTW (VRP *Hard Time Window*) lebih fleksibel dan dapat menurunkan biaya distribusi tanpa mengurangi kepuasan konsumen. Selain itu, sebagian besar permasalahan logistik tidak terlalu mensyaratkan *hard time window* dan waktu perjalanan juga tidak dapat diketahui secara akurat. Secara spesifik, penelitian ini membahas mengenai penyusunan rute kendaraan menggunakan batasan *soft time windows*, di mana pengiriman masih bisa dilakukan meskipun kedatangan terjadi di luar *time windows* dengan membayarkan penalti.

VRPTW merupakan persoalan optimasi kombinatorial kategori *NP-hard* (*non polynomial*) sama seperti VRP. Pada persoalan ini peningkatan kesulitan dan kekompleksan akan bertambah secara faktorial seiring bertambah besarnya ukuran problem. Sangatlah diperlukan metode optimisasi yang kompetitif untuk menyelesaikan problem penyusunan rute. Teknik penyelesaian VRPTW sendiri terbagi menjadi dua metode, yaitu metode eksak serta heuristik. Metode eksak perhitungannya membutuhkan waktu lama sehingga tidak sesuai bila diaplikasikan di dunia nyata (Müller [5]; Calvete, et al. [6]; Yücenur and Demirel [7]). Para pakar kemudian mengembangkan algoritma heuristik atau *approximation method* untuk memecahkan VRP.

Metode heuristik saat ini telah banyak dikembangkan untuk menyelesaikan permasalahan kombinatorial seperti VRPTW. Menurut Figliozzi [4], heuristik untuk permasalahan VRPTW bisa diklasifikasikan sebagai berikut (diurut berdasar kualitas penyelesaiannya): *construction*, *local search*, serta metaheuristik (banyak titik-titik calon solusi).



Gambar 1 Representasi Pergerakan Kawanan dalam PSO [8]

Algoritma *Particle Swarm Optimization* (PSO) adalah salah satu metode metaheuristik yang diilhami perilaku sosial kawanan (*swarm*) makhluk hidup, atau teknik *swarm intelligence* [9]. Algoritma pencarian berdasar populasi ini diperkenalkan oleh Kennedy dan Ebenhart tahun 1995 diilhami perilaku pencarian kawanan makhluk hidup (kelompok lebah, ikan, burung). Mekanisme ini kemudian diadaptasi dalam permasalahan optimasi, di mana penyelesaian terbaik berada pada posisi tertentu pada ruang pencarian. Pada iterasi PSO, tiap partikel yang tersebar pada posisi dan kecepatan acak, terbang menuju posisi baru dengan menyesuaikan kecepatan terbang. Kedua karakteristik tersebut, posisi dan kecepatan, akan selalu dimutakhirkan di setiap iterasi. Penyesuaian kecepatan mencari posisi terbaik didasarkan dua perilaku, yaitu, perilaku kognitif individu serta perilaku sosial partikel. Posisi terbaik partikel,

meniru perilaku kognitif, adalah posisi terbaik yang dikunjungi suatu partikel berdasar fungsi tujuan yang dicapai. Sedangkan posisi terbaik global, meniru perilaku sosial partikel, adalah posisi terbaik yang dikunjungi semua partikel dalam kawanan. Representasi pergerakan kawanan dalam PSO ditunjukkan pada Gambar 1.

Algoritma PSO bisa dengan mudah dan efisien diaplikasikan pada permasalahan klasik optimasi, seperti VRP, tetapi relatif masih jarang digunakan [10]. Kemudahan *coding* serta kemampuan mengeksplorasi baik lokal maupun global, memicu keefisienan yang tidak meninggalkan kualitas penyelesaian. Berbeda dengan meteheuristik lainnya, PSO hanya menghitung dua variabel setiap anggota populasi, yaitu kecepatan serta posisi pada setiap iterasi. Pemutakhiran variabel kecepatan, bobot inersia serta posisi ditunjukkan oleh persamaan 13, 14 dan 16.

Pada penelitian ini, algoritma penyelesaian yang dikembangkan adalah berdasarkan model pengiriman penyusunan rute kendaraan produk *perishable* oleh Trihardani [11]. Produk *perishable* memiliki karakteristik unik di mana umur ketahanan (*shelf life*) terbatas dan sepanjang *shelf life* tersebut, *value* produk kontinu menurun sehingga berpotensi menyebabkan *loss*. Makanan, obat, vaksin, susu, *yogurt, ice cream*, kosmetik, adalah berbagai contoh produk *perishable* (tidak tahan lama).

Perlu diperhatikan bahwa konsumsi produk *perishable*, terutama produk makanan mengalami peningkatan signifikan pada beberapa dekade terakhir. Permintaan produk makanan *perishable* berkontribusi terhadap sepertiga dari total penjualan ritel dunia, yang nilainya melebihi $1000 miliar [12]. Peningkatan permintaan ini juga pasti akan berimbas pada tuntutan kebutuhan layanan pengiriman produk *perishable* yang *cost effective*. Ketidakoptimalan sistem pengiriman menyebakan *loss* kualitas karena barang tertahan (*idle*) saat pendistribusian, baik dalam sentral depot maupun dalam kendaraan.

Sepanjang pengetahuan, masih sedikit penelitian yang fokus pada permasalahan pengiriman produk *perishable*. Padahal Berbagai penelitian yang membahas pengiriman produk *perishable* adalah: Hwang [13];[Tarantilis and Kiranoudis [14], 15]; Faulin [16]; Belenguer, et al. [17]; Hsu, et al. [18]; Osvald and Stirn [19]; Chen, et al. [20]; Rong, et al. [21]; Trihardani [11]. Trihardani [11]mengembangkan suatu model penyusunan rute kendaraan yang mengangkut produk perishable dengan mempertimbangkan kemungkinan kepadatan lalu lintas, *soft time windows*, *perishability*, serta *setting* temperatur kendaraan sesuai standar. Dalam analisisnya, Trihardani [11] menggunakan dua strategi penyelesaian yaitu *distance-dependent* dan *temperature-dependent* yang dibandingkan performanya melalui suatu analisis sensitivitas.

Penelitian ini mengembangkan algoritma penyelesaian penentuan rute pengiriman produk perishable menggunakan metaheuristik, *Particle Swarm Optimization* (PSO). Pengembangan algoritma ini dilakukan karena penentuan rute termasuk permasalahan diskrit. Selain itu dilakukan pengembangan algoritma untuk meningkatkan performa dengan melakukan kombinasi (*hybrid*) algoritma, Pengembangan algoritma metaheuristik melalui proses kombinasi muncul sebagai upaya untuk meningkatkan performansi algoritma induk yang dirasa memiliki kelemahan dan dapat diperbaiki menggunakan mekanisme kombinasi algoritma lain. Salah satu penerapan *hybrid* metaheuristik dikembangkan oleh Yan, et al. [22] dan Das, et al. [23] yang sukses digunakan dalam permasalahan kontinu. *Hybrid* metaheuristik memiliki potensi yang besar untuk dikembangkan dan diterapkan untuk menyelesaikan permasalahan optimasi kombinatorial seperti VRPTW.

Oleh karena itu pada penelitian ini juga akan mengembangkan algoritma *hybrid* metaheuristik yang menggabungkan algoritma *Nearest Neighbor* (NN) dengan

PSO untuk menyelesaikan kasus VRPSTW. NN sendiri termasuk algoritma *route construction* di mana iterasi algoritma dilakukan dengan menyisipkan pelanggan terdekat dalam rute parsial sehingga mendapatkan solusi *feasible*. Algoritma NN digunakan untuk menetapkan rute awal yang selanjutnya akan diperbaiki performasnya menggunakan *swarm intellegence* pada algoritma PSO secara metaheuristik dengan berbagai calon titik solusi. Penyusunan rute tersebut menggunakan batasan *soft time window* dan meminimumkan waktu tunggu karena kedatangan kendaraan yang lebih awal. Penyelesaian model ini menggunakan data Solomon yang mewakili karakteristik pelanggan dilihat dari penyebaran pelanggan (*random* atau terklaster), Masing-masing teknik penyelesaian akan dianalisis kualitas solusinya, mana yang lebih efektif meminimumkan biaya.

2. Metode Penelitian

Tujuan permodelan ini adalah meminimalkan total biaya pengiriman yang terdiri atas: biaya tetap penggunaan kendaraan, biaya transportasi, biaya persediaan, biaya energi, serta biaya penalti keterlambatan. Elemen-elemen biaya pengiriman dalam fungsi tujuan ini adalah sebagai berikut Trihardani [11]:

1) Biaya tetap penggunaan kendaraan (*dispatching*)
   Biaya tetap penggunaan kendaraan berhubungan dengan banyaknya kendaraan yang dipakai serta biaya tetap penggunaan kendaraan per unit.
2) Biaya transportasi
   Biaya transportasi adalah biaya variabel yang berhubungan dengan jarak antar pelanggan serta biaya transportasi per satuan jarak.
3) Biaya persediaan
   Biaya persediaan adalah biaya risiko *loss* membawa persediaan selama perjalanan yang ditanggung distributor.
4) Biaya energi
   Biaya energi berhubungan dengan total biaya yang diperlukan kompresor mendinginkan ruangan
5) Biaya penalti keterlambatan
   Biaya penalti berhubungan dengan selisih antara waktu kedatangan, y_i, dengan waktu akhir *time windows*, s_i, (di mana $s_i \le y_i \le S_i$) serta biaya penalti keterlambatan per unit waktu

Model matematis untuk meminimalkan total biaya distribusi penelitian ini adalah sebagai berikut:

$$\begin{aligned} \boldsymbol{min} \sum_{\forall l \in L} \boldsymbol{f} + \sum_{\substack{\forall i \in I_0 \\ i \neq j}} \sum_{\substack{\forall j \in I_0 \\ j \neq i}} \sum_{\forall m \in M} \sum_{\forall l \in L} \boldsymbol{Rc_{ij} q_{ijlm}} + \; + \sum_{\forall i \in I_0} \sum_{\forall j \in I} \sum_{\forall m \in M} \sum_{\forall l \in L} \boldsymbol{Pz_{jlm} d_i} \\ + \sum_{\forall i \in I_0} \sum_{\forall j \in I} \sum_{\forall m \in M} \sum_{\forall l \in L} \boldsymbol{E}\left(\boldsymbol{y_{ilm} - y_{(i-1)lm}}\right)\left(\frac{\boldsymbol{1}}{\boldsymbol{s}} \boldsymbol{kA}\right) + \left(\left(\frac{\boldsymbol{1}}{\boldsymbol{24}} \boldsymbol{w_{lm} C(e - T_0)}\right) + \left(\boldsymbol{r_{lm}\, Cl}\right)\right. \\ \left. + \left(\boldsymbol{r_{lm} C(T_0 - t_{lm})}\right)\right) + \sum_{\forall i \in I_0} \boldsymbol{Uq_i} \end{aligned} \quad (1)$$

Sementara kendala yang digunakan dalam model ini adalah sebagai berikut:

1) Kendala yang menyatakan, satu kendaraan mengunjungi pelanggan i satu kali.

$$\sum_{\forall i \in I_0} \sum_{\forall l \in L} \sum_{\forall m \in M} z_{ilm} = \begin{cases} l; & \forall i \in I_0 \\ 1; & \forall i \in I \end{cases} \quad (2)$$

2) Kendala yang menyatakan, rute (i ,j) hanya dilewati satu kendaraan.
$$\sum_{\forall i \in I_0} \sum_{\forall j \in I_0} \sum_{\forall l \in L} \sum_{\forall m \in M} q_{ijlm} = z_{jlm} \; ; \forall i \in I_0, \forall j \in I_0, \forall l \in L, \forall m \in M \quad (3)$$

3) Kendala *time windows*, menyatakan bahwa waktu kedatangan pelanggan (i+1) kendaraan l item p temperatur g adalah penjumlahan dari waktu kedatangan di pelanggan i, lama kunjungan di pelanggan i, dan waktu perjalanan yang diharapkan dari i menuju (i+1).
$$y_{(i+1)} \geq y_i + u_i + \bar{t}_{i(i+1)} - \left(1 - x_{i(i+1)}\right); \forall i \in I, \forall l \in L, \quad \forall m \in M \quad (4)$$

4) Kendala *time windows*, menyatakan bahwa waktu kedatangan pelanggan i adalah penjumlahan dari waktu keberangkatan kendaraan l item p temperatur g dari sentral depot dan waktu perjalanan yang diharapkan dari sentral depot menuju pelanggan i.
$$y_i \geq y_{0lm} + \bar{t}_{0ilm} - (1 - x_{0ilm})M; \forall i \in I, \forall l \in L, \quad \forall m \in M \quad (5)$$

5) Kendala *time windows*, menyatakan bahwa waktu kembali kendaraan l item p temperatur g ke sentral depot adalah penjumlahan dari waktu kedatangan pelanggan (i + 1), lamanya pelayanan pelanggan (i + 1), serta waktu perjalanan yang diharapkan pelanggan (i + 1).
$$y_{0lm} \geq y_{(i+1)} + u_{(i+1)} + \bar{t}_{(i+1)0lm} - \left(1 - x_{(i+1)0lm}\right)M \, , \forall i \in I \quad (6)$$

6) Kendala yang menyatakan batasan waktu mengirimkan barang ke pelanggan i. Rentang waktu kedatangan pelanggan i berada di antara waktu pelayanan terawal dan waktu pelayanan terakhir dengan penalti pelanggan i.
$$r_i \leq y_i \leq S_i, \quad \forall i \in I \quad (7)$$

7) Kendala yang menyatakan bahwa jumlah muatan yang dibawa kendaraan l temperatur g tidak melebihi kapasitas kendaraan l
$$w_{lm} = \sum_{\forall i \in I} \sum_{\forall l \in L} \sum_{\forall m \in M} z_{ilm} \leq K; \quad \forall i \in I; \; \forall l \in L, \forall m \in M \quad (8)$$

8) Kendala yang menyatakan jumlah *loss* ketika melayani pelanggan i dengan kendaraan l item p temperatur g. Kuantitas *loss* ketika berada di pelanggan i adalah penjumlahan fungsi kumulatif probabilitas *loss* (berhubungan dengan *shelf life* serta durasi perjalanan dari sentral depot ke pelanggan i) serta probabilitas *loss* membuka *cold storage* (berhubungan dengan *shelf life* dan durasi pelayanan pelanggan i)
$$x_{0ilm} = x_{0ilm} r_{lm} \, x\left[F\left(y_{ilm} - y_{jlm}\right) + G(d_i)\right], \forall i \in I; \; \forall l \in L, \forall m \in M \quad (9)$$

9) Kendala yang menyatakan temperatur *cold storage* kendaraan l untuk pelanggan i adalah temperatur minimum kumpulan item p kendaraan l untuk pelanggan i.
$$t_{ilm} = min\{0, (t_{ilm})\}; \forall i \in I; \; \forall l \in L, \forall m \in M \quad (10)$$

10) Kendala yang menyatakan temperatur kendaraan l yang memuat item p untuk pelanggan i adalah sama sepanjang rute perjalanan.
$$t_{ilm} = t_{(i+1)lm}; \; \forall i \in I; \; \forall l \in L, \forall m \in M \quad (11)$$

11) Kendala yang menyatakan waktu keterlambatan penalti adalah waktu kedatangan dikurangi dengan waktu akhir *time windows*.
$$q_i = \max\{0, (y_i - s_i)\}; \; \forall i \in I \quad (12)$$

12) Kendala biner untuk variabel q_{ijlm} dan z_{ilm}
$$q_{ijlm} \in \{0,1\}, z_{ilm} \in \{0,1\} \quad (13)$$

Sedangkan persamaan terkait algoritma PSO adalah sebagai berikut:

13) Pemutakhiran kecepatan

$$V_j(i) = \theta V_j(i\text{-}1) + c_1 r_1 [P_{best,j} - x_j(i\text{-}1)] + c_2 r_2 [G_{best} - x_j(i\text{-}1)] \quad (14)$$

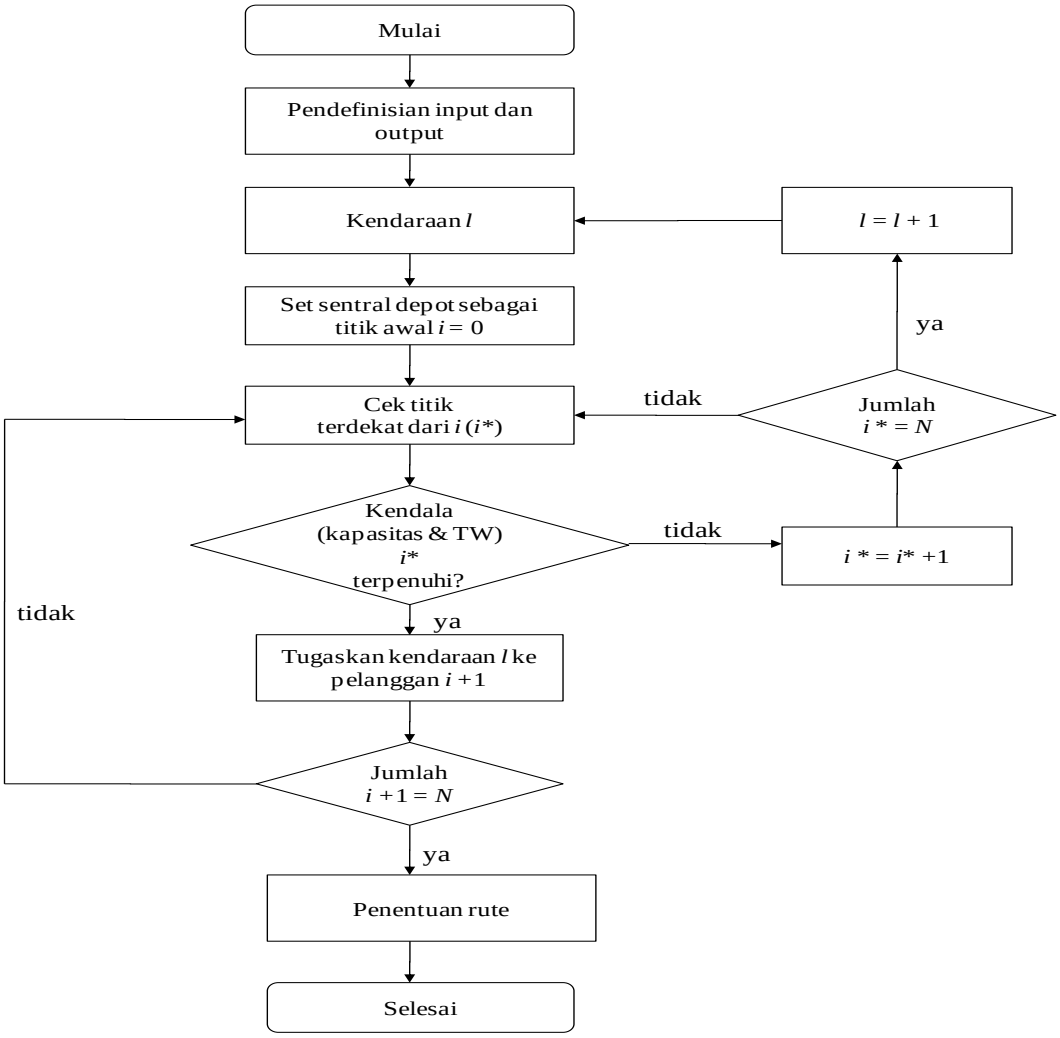
14) Pemutakhiran bobot inersia

$$\theta(i) = \theta_{max} - \frac{\theta_{max} - \theta_{min}}{i_{max}} i \quad (15)$$

15) Pemutakhiran posisi

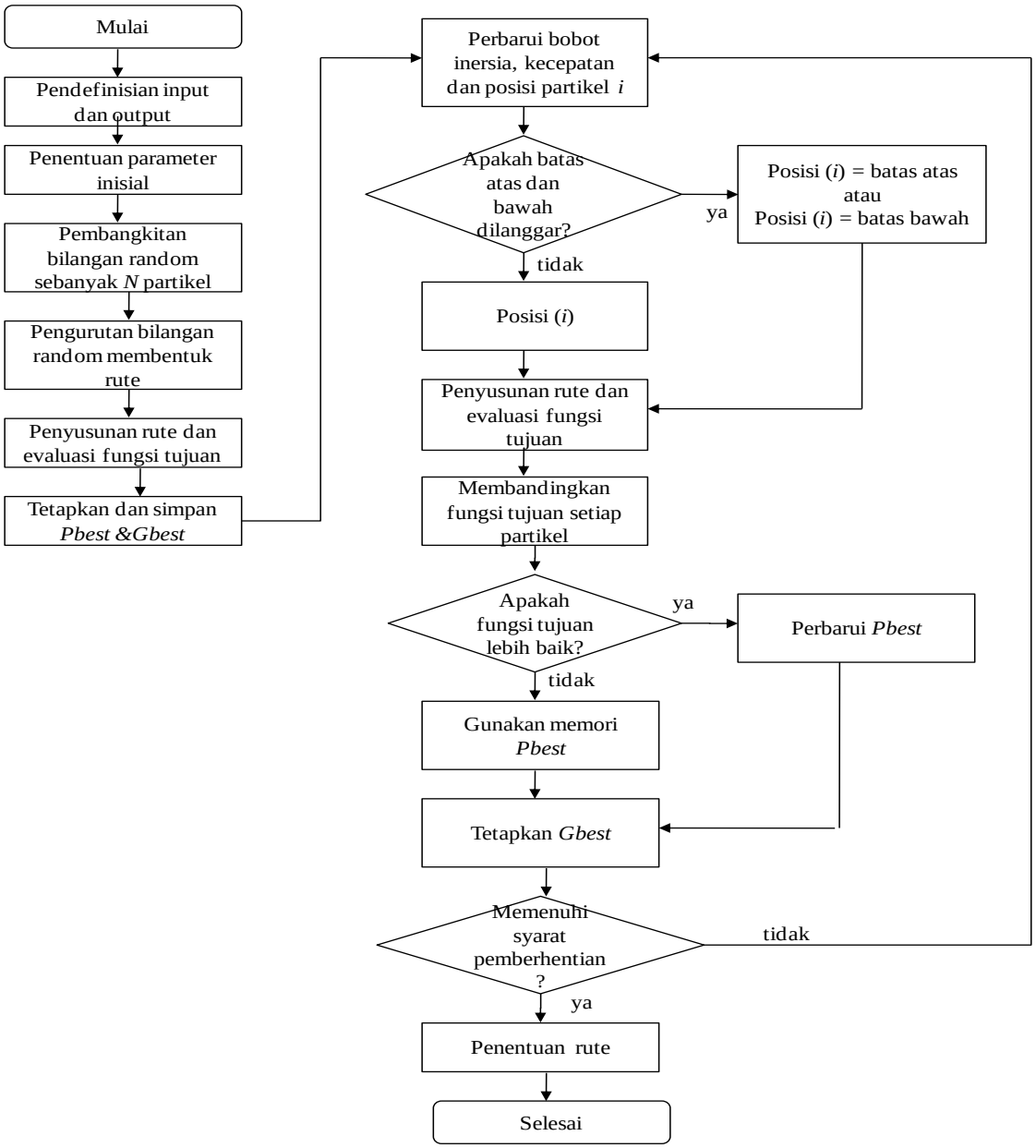
$$X_j(i) = X_j(i\text{-}1) + V_j(i) \quad (16)$$

Penelitian ini akan mengembangkan algoritma *hybrid* metaheuristik yang menggabungkan algoritma NN dengan PSO untuk menyelesaikan kasus VRPSTW. Algoritma NN digunakan untuk menetapkan solusi awal berupa *initial route* dengan cara menyisipkan titik terakhir rute dengan titik terdekat. Apabila kandidat solusi tersebut tidak memenuhi kendala, maka akan disisipkan kandidat yang lebih dekat dan seterusnya. Aliran algoritma VRPTW menggunakan algoritma NN ditunjukkan Gambar 2 dengan penjelasan sebagai berikut:



Gambar 2 Aliran Algoritma *Nearest Neighbor* VRPTW

Setelah mendapatkan rute awal dengan algoritma NN, selanjutnya adalah melakukan perbaikan rute dengan berbagai titik solusi menggunakan algoritma PSO. VRPTW menggunakan variabel diskret biner [0, 1] sehingga PSO kontinu perlu dimodifikasi menjadi PSO diskret. Partikel bergerak pada ruang terbatas antara nol dan satu setiap dimensinya pada PSO diskrit. Prosedur VRPTW PSO diskret ini berdasar pengurutan bilangan *random*, diambil dari Santosa and Willy [24]. Adapun alur algoritma PSO ditunjukkan oleh Gambar 3.



Gambar 3 Aliran Algoritma PSO VRPTW

Langkah-langkah penyelesaian VRPTW menggunakan PSO adalah sebagai berikut:

1) Input data awal: Data awal ini meliputi informasi pelanggan serta sentral depot, yaitu: koordinat kartesian lokasi, permintaan, temperatur penyimpanan, *time windows*, waktu pelayanan, *shelf life,* kepadatan lalu lintas (1 jika ya, 0 sebaliknya), kapasitas tiap kendaraan, konstanta *time windows* penalti,

konstanta *service time*, probabilitas kepadatan, invers kecepatan *A* (β_0), invers kecepatan *A'* bila tidak mengalami kepadatan lalu lintas (β_1), invers kecepatan *A'* bila mengalami kepadatan lalu lintas (β_2), biaya tetap penggunaan kendaraan, biaya risiko, biaya transport serta biaya penalti. Pelanggan 0 merupakan sentral depot sebagai pusat distribusi. Input data awal ditunjukkan Gambar 4.

Pelanggan	Koordinat x_i	Koordinat y_i	Permintaan	Temperatur	*Time windows*			*Service time*	*Purchasing cost*	*Shelf life*	Kepadatan
					waktu awal	waktu akhir	waktu akhir penalti				
0	0	0	0		200						
1	16.47	64.44	30	5	240	750	870	150	5	24	0
2	80.09	12.54	40	-5	240	650	770	200	3	24	0
3	42.39	93.37	60	-10	270	700	820	300	4	36	0
4	25.23	44.24	50	3	270	850	970	250	7	48	1

Kapasitas tiap kendaraan	100	
Konstanta *time windows* penalti	120	+ waktu akhir *time windows*
Konstanta *service time*	0.05	* permintaan
Probabilitas kepadatan	0.7	
β0 β1	0.025	
β2	0.03	
Biaya *dispatch* per kendaraan	0.025	
Biaya transport per satuan jarak	5	
Biaya energi per °C per waktu	25	
Biaya penalti per waktu	10	
	3	

Gambar 4 Input Data Awal

2) Menentukan matriks jarak serta matriks waktu tempuh *link* (*i* ,*j*): Jarak dua titik x_i dan y_i dihitung menggunakan jarak Euclidean. Sementara waktu tempuh dihitung dengan memerhatikan kepadatan lalu lintas. Matriks jarak dan matriks jarak tempuh ditunjukkan pada Tabel 1.

Tabel 1 Matriks Jarak serta Waktu Tempuh

Matriks Jarak

	0	1	2	3	4
0	0.00	66.51	81.07	102.54	50.93
1	66.51	0.00	82.10	38.84	22.02
2	81.07	82.10	0.00	89.19	63.36
3	102.54	38.84	89.19	0.00	52.04
4	50.93	22.02	63.36	52.04	0.00

Matriks waktu tempuh

	0	1	2	3	4
0	0.00	1.66	2.03	2.56	1.27
1	1.66	0.00	2.05	0.97	0.55
2	2.03	2.05	0.00	2.23	1.58
3	2.56	0.97	2.23	0.00	1.30
4	1.57	0.68	1.95	1.60	0.00

3) Inisialisasi *random* partikel x_i dan kecepatan v_i dalam ruang pencarian problem *p*-dimensi: Menentukan inisialisasi kawanan, bobot inersia, atribut posisi serta kecepatan partikel (c_1, c_2, v_o). Misalkan jumlah partikel (N) = 6, bobot inersia θ =

0.3. Sementara r_1 dan r_2 adalah 0.4 dan 0.6, c_1 dan c_2 masing-masing bernilai 1, serta kecepatan awal v_o diasumsikan sama, $v_o = 0.7$ untuk semua partikel.

Tabel 2 Pembangkitan dan Pengurutan Bilangan Random

	0	1	2	3	4
1	0	0,207	0,645	0,042	0,171
2	0	0,709	0,172	0,171	0,447
3	0	0,223	0,83	0,299	0,547
4	0	0,694	0,888	0,602	0,713
5	0	0,31	0,466	0,431	0,51
6	0	0,046	0,921	0,178	0,016

(a) Pembangkitan bilangan random

	0	1	2	3	4
1	0	0,042	0,171	0,207	0,645
2	0	0,171	0,172	0,447	0,709
3	0	0,223	0,299	0,547	0,83
4	0	0,602	0,694	0,713	0,888
5	0	0,31	0,431	0,466	0,51
6	0	0,016	0,046	0,178	0,921

(b) Pengurutan bilangan random

4) Membangkitkan bilangan *random* dan mengurutkannya membentuk rute: Bilangan *random* yang dibangkitkan berada pada rentang [0, 1] sebanyak jumlah titik yang dikunjungi termasuk sentral depot dan pelanggan. Proses ini diulangi sebanyak jumlah partikel yang telah diinisialisasi di awal (N = 6). Bilangan random sentral depot selalu bernilai 0 karena titik awal harus mulai dari depot. Setelah pembangkitan bilangan random maka bilangan *random* setiap partikel akan diurutkan membentuk suatu rute. Pembangkitan dan pengurutan bilangan random dicontohkan pada Tabel 2.
5) Penyusunan rute serta evaluasi fungsi tujuan: Setelah masing-masing partikel diurutkan, maka akan didapat rute berdasar posisi bilangan *random* tersebut sebelum diurutkan. Misalkan partikel 2 urutan rutenya adalah 0 – 3 – 4 – 1 – 2 - 0. Masing-masing rute tersebut akan dievaluasi fungsi tujuannya berdasar persamaan 1 seperti terlihat pada Tabel 3, sedangkan rekapitulasi penyusunan rute beserta fungsi tujuannya ditunjukkan Tabel 4.

Tabel 3 Evaluasi Fungsi Tujuan (Partikel 2)

Kendaraan	Node awal	Node tujuan	Load kum	*Kapasitas Feasible ?*	Waktu Keberangkatan	Lama Perjalanan	Waktu Kedatangan	TW Start	Mulai pelayanan	Lama pelayanan	Selesai pelayanan	TW akhir pelayanan	TW dengan penalti	*TW feasible ?*	Penalti ?	Jarak *(ij)*	*F (-)*	*g(di)*	Biaya pembelian	Biaya persediaan	Opsi temperatur	Temperatur kendaraan	Durasi kendaraan	Biaya Energi
1	0	3	60	yes	200	2,56	203	270	270	300	570	700	820	yes	no	103	0,2	0,1	4,0	240	10000	-10,0	370	100
1	3	2	100	yes	570	2,23	572	240	572	200	772	650	770	yes	no	892	0,1	0,1	3,0	300	-5000	-10,0	202	100
1	2	4	150	no	772	1,58	774	270	774	250	1024	850	970	yes	no	0	0,0	0,0	7,0	0				
1	2	0	0	no	1024	2,03	1026	200	1026	0	1026	0	0	yes	no	0	0,0	0,0	0,0	0				
2	0	4	50	yes	200	1,27	201	270	270	250	520	850	970	yes	no	509	0,1	0,1	7,0	350	3000	3,0	320	30
2	4	1	80	yes	520	0,68	521	240	240	150	671	750	870	yes	no	22	0,1	0,1	5,0	400	5000	3,0	151	30
2	1	0	0	yes	671	1,66	672	200	200	0	672	0	0	yes	no	0	0,0	0,0	0,0	0				

Biaya tetap	10
Biaya transportasi	6617
Biaya persediaan	1290
Biaya energi	271156
Biaya Penalti	0
Total	279073

Tabel 4 Rekapitulasi Penyusunan Rute dan Evaluasi Fungsi Tujuan

	0	1	2	3	4	Fungsi tujuan
1	0	3	4	1	2	139,896
2	0	3	2	4	1	279,073
3	0	1	3	4	2	239,604
4	0	3	1	4	2	299,731
5	0	1	3	2	4	253,153
6	0	4	1	3	2	279,073

6) Memutakhirkan nilai P_{best} partikel dan G_{best} kawanan: Iterasi pertama, P_{best} sama dengan nilai awal partikel. Selanjutnya P_{best} ini akan disimpan dan dibandingkan dengan iterasi P_{best} selanjutnya. P_{best} yang disimpan adalah bilangan random yang dibangkitkan, bukan rutenya. Sementara penentuan G_{best} berdasar pertikel dengan nilai fungsi tujuan terkecil (karena fungsi obyektifnya adalah meminimalisasi). Penentuan P_{best} dan G_{best} ditunjukkan pada Tabel 5. Pada contoh ini P_{best} partikel 1 adalah [.000 .207 .645 .042 .171]. Sementara G_{best} berada di posisi partikel 1 (fungsi obyektif paling minimum Tabel 5 sehingga G_{best} = [.000 .207 .645 .042 .171].
7) Memperbarui bobot inersia, kecepatan dan posisi partikel: Kecepatan diperbarui diperoleh berdasar (2.21), bobot inersia diperbarui berdasar (2.22), sedangkan posisi partikel diperbarui berdasar (2.23). Kecepatan serta posisi partikel diperbarui ditunjukkan pada Tabel 6.

Tabel 5 Penentuan P_{best} dan G_{best}

	0	1	2	3	4	Fungsi tujuan
1	0	0,207	0,645	0,042	0,171	139,896
2	0	0,709	0,172	0,171	0,447	279,073
3	0	0,223	0,83	0,299	0,547	239,604
4	0	0,694	0,888	0,602	0,713	299,731
5	0	0,31	0,466	0,431	0,51	253,153
6	0	0,046	0,921	0,178	0,016	279,073

Tabel 6 Kecepatan dan Posisi Partikel Diperbarui

	0	1	2	3	4
1	0	0,21	0,21	0,21	0,21
2	0	0	0,494	0,133	0,044
3	0	0,2	0,099	0,056	0
4	0	0	0,064	0	0
5	0	0,148	0,317	0	0,007
6	0	0,306	0,044	0,129	0,303

(a) Kecepatan diperbarui

	0	1	2	3	4
1	0	0,417	0,855	0,252	0,381
2	0	0,709	0,666	0,304	0,491
3	0	0,423	0,929	0,355	0,547
4	0	0,694	0,952	0,602	0,713
5	0	0,458	0,783	0,431	0,516
6	0	0,352	0,965	0,307	0,319

(b) Posisi diperbarui

8) Memeriksa batas atas dan bawah posisi partikel: Apabila tidak memenuhi maka sesuaikan nilai partikel berdasar batas atas dan bawah yang ditetapkan.
9) Kembali ke langkah 5. Jika iterasi telah memenuhi maka hentikan

Tabel 7 Perbandingan Solusi Algoritma PSO dan Hybrid Metaheuristik PSO

Problem	PSO murni Total Biaya (Rp. Dalam ribu)	Hybrid PSO Total Biaya (Rp. Dalam ribu)	Gap
C101	2,105	2,081	1,16%
C102	1,8	1,85	0,00%
C103	1,83	1,835	0,00%
C104	1,558	1,558	0,00%
C105	1,867	1,828	2.16%
R101	2,79	2,786	0,16%
R102	2,879	2,764	4.16%
R103	2,657	2,47	7.56%
R104	2,038	1,932	5.50%
R105	2,533	2,533	0,00%

Tabel 8 Perbandingan Waktu Komputasi dan Jumlah Iterasi Algoritma PSO dan Hybrid Metaheuristik PSO

Problem	PSO Murni		Hybrid PSO	
	Waktu Komputasi	Banyak Iterasi	Waktu Komputasi	Banyak Iterasi
C101	1738	444	80	39
C102	1382	500	65	39
C103	2062	561	98	40
C104	8401	1250	3101	83
C105	7984	1250	3581	712
R101	34012	1180	4262	1106
R102	37284	1750	28877	1991
R103	33704	1750	29789	1888
R104	31990	1750	20127	1748
R105	35286	1925	24131	1830

3. Hasil dan Pembahasan

Algoritma Hybrid NN dan PSO diterjemahkan ke dalam *coding* dengan menggunakan software Microsoft Excel 2013 Visual Basic Application. Eksperimen dilakukan dengan menggunakan komputer dengan spesifikasi Intel Core i5,3.18 Hz RAM 8192 MB. Data numerik penelitian ini menggunakan data Solomon (http://web.cba.neu.edu/msolomon/problems.htm). Data Solomon yang digunakan dalam uji numerik penelitian ini adalah data kelas C1 dan R1. Kedua kelas data ini ini mewakili karakteristik penyebaran lokasi pelanggan, klaster (C1) atau *random* (R1). Percobaan numerik menggunakan kelas data yang berbeda dilakukan karena akan memengaruhi *running* dan penyelesaian model. Perbedaan karakteristik kelas data akan memengaruhi perilaku model, dilihat dari waktu perhitungan serta penyelesaian

model Müller [5]. Setiap data uji akan diselesaikan dengan menggunakan algoritma PSO murni dan hibrid PSO masing-masing sebanyak 20 replikasi.

Dari rekap hasil uji eksperimen yang telah dilakukan pada Tabel 7 dan Tabel 8 terdapat 3 kriteria performa, yaitu total biaya, iterasi, waktu. Total biaya merepresentasikan solusi yang ditemukan oleh masing-masing algoritma di setiap replikasi. Iterasi merepresentasikan jumlah iterasi yang dibutuhkan untuk menemukan solusi, sedangkan waktu merepresentasikan waktu yang dibutuhkan untuk menemukan solusi di setiap replikasinya. Dari contoh kasus Solomon tersebut, algoritma PSO murni mampu menghasilkan empat solusi yang sama dengan solusi terbaik yaitu untuk data Solomon C101, C103, C104, dan R105. Untuk enam contoh kasus lainnya, walaupun PSO tidak menemukan solusi optimal, tetapi solusi yang dihasilkan cukup kompetitif dengan gap dapat menghasilkan solusi dengan gap antara 0,10 – 7.56%.

Di sisi lain, algoritma hibrid PSO terlihat lebih baik daripada algoritma PSO murni. Algoritma ini dapat menemukan sepuluh solusi terbaik pada data Solomon C101, C102, C103, C104, C105, R101,R102, R103, R104, R105. Hal ini menunjukkan bahwa algoritma hibrid metaheuristik NN-PSO yang dikembangkan mampu menghasilkan solusi yang lebih baik jika dibandingkan dengan algoritma metaheuristik PSO murni. Algoritma hibrid metaheuristik tersebut mampu menuju fungsi obyektif yang lebih kecil daripada PSO murni (G_{best} lebih baik). Sampai dengan 20 iterasi yang dilakukan hibrid metaheuristik mampu menuju konvergenitas G_{best} yang lebih baik daripada PSO murni.

Keefisienan penyelesaian juga beresiko rute yang dihasilkan oleh algoritma hibrid metaheuristik NN-PSO terjebak dalam *local optima*. Hal ini disebabkan G_{best} tidak bergerak ke arah fungsi obyektif yang lebih baik dari awal iterasi sampai iterasi ke-20. Sementara untuk solusi yang dihasilkan dari PSO murni mampu bergerak ke arah fungsi obyektif yang lebih baik. Bisa dikatakan pergerakan kawanan algoritma hibrid metaheuristik NN-PSO untuk mendapatkan solusi optimal jauh lebih lambat daripada PSO murni.

4. Simpulan

Model matematis yang dikembangkan telah merepresentasikan permasalahan pengiriman penentuan rute kendaraan yang mengangkut produk perishable. Algoritma *Particle Swarm Optimization* (PSO) dan *hybrid* metaheuristik *Nearest Neighbor* (NN)-PSO yang dikembangkan mampu digunakan untuk menyelesaikan permasalahan penentuan rute dengan *soft time windows*. Algoritma PSO mampu menghasilkan solusi optimal pada set data C101, C103, C104, dan R105. Solusi yang ditemukan pada set data Solomon lainnya C102, C105, R101, R102, R103, R104, cukup kompetitif dengan gap 0,10 – 7.56%. Terbukti bahwa Algoritma hybrid metaheuristik NN-PSO mampu menghasilkan solusi optimal pada semua set data C101, C102, C103, C104, C105, R101,R102, R103, R104, R105. Dari segi waktu komputasi algoritma hybrid metaheuristik NN-PSO, lebih unggul dibandingkan PSO untuk semua eksperimen. Penelitian *routing problem* dengan *time window* masih belum banyak dilakukan sehingga dapat dilakukan pengembangan model dan diaplikasikan ke permasalahan nyata. Selain itu, juga dapat dilakukan pengembangan metode atau algoritma metaheuristik lain untuk menyelesaikan permasalahan ini sehingga dapat menemukan solusi-solusi terbaik untuk dijadikan acuan penelitian berikutnya.

Daftar Notasi

I_n : Himpunan pelanggan= {0,…., *i*,…, *n*); dimana *I* = {1,…., *i*,…, *n*) dan {0} = sentral depot
L : Himpunan kendaraan = {1,…, *l*,…., *v*)
M : Himpunan produk *perishable* = {1,….., *m*,…., *s*}
f : Biaya tetap penggunaan kendaraan yang digunakan per unit
P : Biaya resiko terjadi *loss* per unit *item*
R : Biaya transportasi kendaraan *l* per unit jarak
T_m : Temperatur penyimpanan produk m; m ϵ M
A : Luasan permukaan ruang penyimpanan kendaraan/*cold storage* (m^2)
s : Ketebalan permukaan ruang penyimpanan kendaraan/*cold storage* (m)
k : koefisien insulating material; k = 0.002
C : spesifikasi panas muatan; C = 0.77
Cl : panas laten; Cl = 60
e : Temperatur eksternal cold storage; diasumsikan 32°C
T_0 : Temperatur netral (0°C)
r_i : Waktu awal pelayanan pada time windows pelanggan i; i ϵ I
s_i : Waktu akhir pelayanan *time windows* pelanggan *i*; $i \epsilon I$
S_i : Waktu akhir pelayanan yang masih diterima pelanggan *i* dengan penalti
u_i : Durasi pelayanan di pelanggan *i*; $i \epsilon I$
$\overline{t}_{ij}^{l}$: Lamanya perjalanan yang diharapkan pelanggan i ke j; i, j ϵ I0
β_0 : Invers kecepatan kendaraan tanpa kepadatan lalu lintas
β_1,β_2 : Invers kecepatan kendaraan dengan kepadatan lalu lintas, $\beta_1 \leq \beta_2$
p : Probabilitas terjadi kepadatan lalu lintas
d_i : Permintaan pelanggan i yang harus dipenuhi; $i \epsilon I$
K : Kapasitas kendaraan
h_m : *Shelf life* produk m; $m \epsilon M$
$F(-)$: Fungsi kumulatif probabilitas loss f(y),
$G(d_i)$: Probabilitas loss, mengikuti shelf life produk dan durasi pelayanan
t_{lm} : Setting temperatur kendaraan $l \epsilon L$ yang membawa item $m \epsilon M$
y_{0lm} : Waktu kembali ke sentral depot kendaraan $l \epsilon L$ yang membawa item $m \epsilon M$
y_{ilm} : Waktu kedatangan ke pelanggan *i*; $i \epsilon I$, kendaraan $l \epsilon L$ yang membawa item $m \epsilon M$
w_{lm} : Muatan yang dibawa kendaraan $l \epsilon L$ yang membawa item $m \epsilon M$
l : jumlah kendaraan yang digunakan
q_{ijlm} : 1; jika rute (i, j) dilewati kendaraan *l* temperatur *g* memuat item *m*; 0 jika tidak
z_{ilm} : 1; jika kendaraan *l* memuat item *m* melayani pelanggan *i*

Referensi

[1] J.-P. Rodrigue, C. Comtois, and B. Slack, The geography of transport systems: Routledge, 2009.
[2] X. Li, P. Tian, and S. C. Leung, "Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm," International Journal of Production Economics, vol. 125, pp. 137-145, 2010.
[3] P. Toth and D. Vigo, "The vehicle routing problem, ser. SIAM monographs on discrete mathematics and applications," Society for Industrial and Applied Mathematics, 2002.

[4] M. A. Figliozzi, "An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows," Transportation Research Part C: Emerging Technologies, vol. 18, pp. 668-679, 2010. [1] J.-P. Rodrigue, C. Comtois, and B. Slack, The geography of transport systems: Routledge, 2009.
[2] X. Li, P. Tian, and S. C. Leung, "Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm," International Journal of Production Economics, vol. 125, pp. 137-145, 2010.
[3] P. Toth and D. Vigo, "The vehicle routing problem, ser. SIAM monographs on discrete mathematics and applications," Society for Industrial and Applied Mathematics, 2002.
[4] M. A. Figliozzi, "An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows," Transportation Research Part C: Emerging Technologies, vol. 18, pp. 668-679, 2010.
[5] J. Müller, "Approximative solutions to the bicriterion vehicle routing problem with time windows," European Journal of Operational Research, vol. 202, pp. 223-231, 2010.
[6] H. I. Calvete, C. Galé, M.-J. Oliveros, and B. Sánchez-Valverde, "A goal programming approach to vehicle routing problems with soft time windows," European Journal of Operational Research, vol. 177, pp. 1720-1733, 2007.
[7] G. N. Yücenur and N. Ç. Demirel, "A new geometric shape-based genetic clustering algorithm for the multi-depot vehicle routing problem," Expert Systems with Applications, vol. 38, pp. 11859-11865, 2011.
[8] X.-S. Yang, Engineering optimization: an introduction with metaheuristic applications: John Wiley & Sons, 2010.
[9] J. Jie, J. Zeng, C. Han, and Q. Wang, "Knowledge-based cooperative particle swarm optimization," Applied mathematics and Computation, vol. 205, pp. 861-873, 2008.
[10] T. J. Ai and V. Kachitvichyanukul, "A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery," Computers & Operations Research, vol. 36, pp. 1693-1702, 2009.
[11] Trihardani, "Vehicle Routing Problem for Optimizing Multi Temperature Joint Distribution On Distribution of Perishable Products," International Conference on Operations and Supply Chain Management (OSCM), pp. 331-343, 2016.
[12] R. A. Broekmeulen and K. H. van Donselaar, "A heuristic to manage perishable inventory with batch ordering, positive lead-times, and time-varying demand," Computers & Operations Research, vol. 36, pp. 3013-3018, 2009.
[13] H.-S. Hwang, "A food distribution model for famine relief," Computers & Industrial Engineering, vol. 37, pp. 335-338, 1999.
[14] C. Tarantilis and C. Kiranoudis, "Distribution of fresh meat," Journal of Food Engineering, vol. 51, pp. 85-91, 2002.
[15] C. Tarantilis and C. Kiranoudis, "A meta-heuristic algorithm for the efficient distribution of perishable foods," Journal of food Engineering, vol. 50, pp. 1-9, 2001.
[16] J. Faulin, "Applying MIXALG procedure in a routing problem to optimize food product delivery," Omega, vol. 31, pp. 387-395, 2003.
[17] J. Belenguer, E. Benavent, and M. Martínez, "RutaRep: a computer package to design dispatching routes in the meat industry," Journal of food engineering, vol. 70, pp. 435-445, 2005.

[18] C.-I. Hsu, S.-F. Hung, and H.-C. Li, "Vehicle routing problem with time-windows for perishable food delivery," Journal of food engineering, vol. 80, pp. 465-475, 2007.
[19] A. Osvald and L. Z. Stirn, "A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food," Journal of food engineering, vol. 85, pp. 285-295, 2008.
[20] H.-K. Chen, C.-F. Hsueh, and M.-S. Chang, "Production scheduling and vehicle routing with time windows for perishable food products," Computers & operations research, vol. 36, pp. 2311-2319, 2009.
[21] A. Rong, R. Akkerman, and M. Grunow, "An optimization approach for managing fresh food quality throughout the supply chain," International Journal of Production Economics, vol. 131, pp. 421-429, 2011.
[22] J.-Y. Yan, Q. Ling, and D.-m. Sun, "A differential evolution with simulated annealing updating method," in Machine Learning and Cybernetics, 2006 International Conference on, 2006, pp. 2103-2106.
[23] S. Das, A. Konar, and U. K. Chakraborty, "Annealed differential evolution," in Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, 2007, pp. 1926-1933.
[24] B. Santosa and P. Willy, "Metoda Metaheuristik Konsep dan Implementasi," Guna Widya, Surabaya, 2011.