

Penyelesaian Permasalahan *Multi-tour Inventory Routing Problem* dengan *Particle Swarm Optimization*

Syarif Daniel Budiman¹, I. Gede Agus Widyadana²

Abstract: The Inventory Routing Problem (IRP) has become a problem that thoroughly studied recently. One of the IRP models that have been developed was an IRP multi-tour, which described a single distribution center r with a set of salespoints. The salespoints is served by a limited number of vehicles with finite load capacities. Vehicle will replenish the salespoints by making a multi-tour route. Each multi-tour consists of a set of sub-tour involving salespoints to be replenished. The multi-tour solution has increase the complexity of the model compared to the normal tour IRP model. Therefore a Particle Swarm Optimization (PSO) algorithm has been proposed for solving a sequential problem in allocating salespoints to the route. Some heuristic methods also been developed for creating the multi-tour. The validity of the algorithm, along with comparison with other existing algorithm, has been provided to validate the proposed PSO algorithm. Result obtained has shown that the PSO algorithm having a promising performance in solving the problem.

Keywords: Inventory routing problem (IRP), single distribution center, salespoint, multi-tour, sub-tour, PSO.

Pendahuluan

Semakin berkembangnya industri menuju arah yang lebih kompetitif membuat *supply chain management* (SCM) menjadi sebuah faktor yang esensial bagi dunia industri (Rau, *et.al.* [8]). Peranan manajemen logistik dalam dalam SCM telah dianggap sebagai hal yang penting dalam memberikan kepuasan pelanggan (Campbell, *et.al.* [4]).

Permasalahan *Inventory Routing Problem* (IRP) akan mengintegrasikan hubungan yang saling bergantung antara aktivitas pengalokasian persediaan dan penentuan rute pengiriman. Menurut Aghezzaf, *et.al.* [1] penyelesaian masalah IRP bertujuan untuk melakukan perencanaan pengiriman barang dengan meminimalkan jumlah kendaraan yang digunakan, rata-rata total distribusi, dan *inventory holding cost* tanpa menyebabkan terjadinya *stockout*. Penyelesaian masalah IRP sesuai untuk diaplikasikan dalam menjalankan sistem SCM seperti *vendor managed inventory* atau VMI.

Menurut Campbell, *et.al.* [4], IRP memiliki kesesuaian dengan karakteristik dasar VMI dimana *vendor* melakukan penekanan biaya distribusi dengan melakukan koordinasi pengiriman barang ke sejumlah pembeli yang berbeda. Dalam hal ini aplikasi penyelesaian masalah IRP dapat menguntungkan pihak pembeli dan *vendor*.

Penyelesaian permasalahan IRP dengan berbagai macam kondisi telah dikembangkan. Archetti, *et.al.* [2] menyelesaikan permasalahan IRP (menyelesaikan sistem VMI) dimana permasalahan IRP untuk kendaraan tunggal diselesaikan memakai metode *branch and cut*. Campbell, *et.al.*, [6] melakukan penyelesaian masalah IRP untuk permasalahan berskala besar. Penyelesaian dilakukan dengan mendekomposisi permasalahan IRP menjadi 2 fase (*integer programming*, dan melakukan penjadwalan). Penelitian mengenai penyelesaian berbagai macam kondisi masalah multivehicle IRP (MIRP) dengan menggunakan metode *branch and cut* juga telah dilakukan oleh C. Coelho & Laporte [3]. Kasus IRP *multi-tour* telah dikembangkan oleh Aghezzaf, *et.al.* [1] yang akhirnya diselesaikan dengan memakai metode *column generation*.

Pengembangan IRP juga telah berkembang dengan mempertimbangkan adanya faktor-faktor lain. Ming Chen, *et.al.* mengembangkan model HSRIS untuk menyelesaikan permasalahan IRP dengan permintaan *stochastic*. Raa & Aghezzaf [7] mengembangkan model IRP yang mempertimbangkan *real*

^{1,2} Fakultas Teknologi Industri, Program Studi Teknik Industri, Universitas Kristen Petra. Jl. Siwalankerto 121-131, Surabaya 60236. Email: sdaniel.budiman@yahoo.com, gedeaw@gmail.com.

life constraint seperti *minimal cycle time*, kapasitas terbatas dari persediaan customer.

Masalah IRP sendiri termasuk dalam kategori *combinatorial optimization* yaitu sebuah permasalahan dengan sejumlah besar kemungkinan penyelesaian (*finite possible solution*). Dalam aplikasinya, pencarian solusi dengan metode mencoba semua kemungkinan sulit untuk dapat dilakukan karena besarnya jumlah solusi yang ada. Banyaknya batasan dan dimensi keputusan yang harus diambil untuk menyelesaikan permasalahan. membuat masalah IRP sangat sulit untuk diselesaikan (Campbell, *et.al.* [4]; Melissa Campbell & Savelsbergh [6]).

Dalam penelitian ini algoritma *Particle Swarm Optimization* (PSO) akan digunakan untuk menyelesaikan model tersebut. PSO merupakan algoritma yang dapat menyelesaikan permasalahan optimasi yang mendekati kondisi optimal dengan waktu penyelesaian yang singkat sehingga digunakan dalam penelitian ini. Algoritma PSO akan mengalami penyesuaian dengan masalah IRP yang akan diselesaikan. Hal ini dikarenakan bermacam-macamnya jenis variabel keputusan yang terdapat dalam model. Pembahasan dalam jurnal ini terdiri beberapa bagian. Bagian metode penelitian yang menunjukkan metode dan dasar pengembangan model PSO. Bagian hasil dan pembahasan yang menunjukkan pengembangan algoritma PSO dan analisa kemampuan algoritma. Bagian simpulan yang berisi mengenai kesimpulan dari penelitian.

Metode Penelitian

Bagian ini membahas mengenai metodologi yang akan dipakai dalam penelitian ini. Penelitian akan berusaha untuk mengembangkan sebuah metode penyelesaian untuk permasalahan IRP tertentu. Model IRP yang akan diselesaikan adalah model IRP *multi-tour* yang telah dikembangkan oleh Aghezzaf, *et. al.* [1]. Dasar model PSO dalam penelitian ini dikembangkan dari perumusan algoritma PSO M1 yang dikembangkan oleh Farhang, *et. al.* [5]. Definisi dimensi yang terdapat dalam PSO M1 merupakan konsep yang terutama dipakai dalam pengembangan algoritma dalam penelitian.

IRP Multi-tour

Pembuatan model dalam IRP dapat disesuaikan dengan berbagai macam kondisi dan batasan permasalahan yang ada. Salah satu bentuk pemodelan IRP yang dijadikan dasar dalam penelitian ini adalah model IRP *multi-tour* yang dikembangkan oleh Aghezzaf, *et.al.* [1].

Konsep IRP Multi-tour

Pembuatan model dalam IRP dapat disesuaikan dengan berbagai macam kondisi dan batasan permasalahan yang ada. Salah satu bentuk pemodelan IRP yang dijadikan dasar dalam penelitian ini adalah model IRP *multi-tour* yang dikembangkan oleh Aghezzaf, *et.al.* [1].

Model yang dibuat Aghezzaf, *et.al.* [1] dikembangkan untuk menyelesaikan masalah IRP dengan konsep *vehicle multi-tour*. Konsep ini merujuk pada kondisi satu kendaraan dapat melakukan perencanaan perjalanan (*tour*) ke serangkaian titik penjualan C .

Hal ini membuat kendaraan dapat melakukan perjalanan bolak-balik dari *distribution center* ke sejumlah titik penjualan yang berbeda-beda untuk setiap perjalanan (*tour*) yang terdapat dalam serangkaian titik penjualan C . Didalam satu *tour* atau disebut sebagai *multi-tour* (C) terdiri dari beberapa *sub-tour* ke titik penjualan yang berbeda-beda. Dapat dikatakan bahwa sebuah *multi-tour* C terdiri dari beberapa *sub-tour* $C_1 \dots C_m$.

Konsep waktu siklus dalam IRP Multi-tour

Aspek penting lain dalam model adalah rentang waktu antara *tour* yang berdekatan (*consecutive tour*) yang didefinisikan sebagai waktu siklus (*cycle time*). Waktu siklus memiliki batas bawah yang berasal dari waktu yang diperlukan untuk bisa melakukan sebuah *multi-tour* oleh sebuah kendaraan dan disebut sebagai $T_{min}(C)$. Perumusan dapat dilihat pada rumus (1)

Batas atas dari waktu siklus terjadi karena adanya kapasitas maksimal dari kendaraan. Saat waktu berjalan sepanjang waktu siklus $T(C)$ atau sama dengan waktu ke t^* dimana $T(C) = t^*$, kendaraan harus menghantarkan kuantitas sebesar q_i (dimana $q_i = d_i \cdot t^*$) ke titik penjualan i pada *multi-tour* C . Notasi d_i adalah laju permintaan titik penjualan i . Jumlah muatan kendaraan dalam sebuah *sub-tour* tidak boleh melebihi kapasitas maksimal kendaraan (k).

Waktu siklus maksimal dari rute C adalah nilai minimal dari waktu siklus maksimal setiap *sub-tour* yang terjadi pada *multi-tour* C seperti pada rumus (2). Pengambilan nilai minimal dilakukan agar jumlah barang yang dikirim tidak melebihi kapasitas maksimal dari kendaraan.

Terdapat persyaratan $T_{min}(C) \leq T_{max}(C)$ agar *multi-tour* tersebut *feasible*. Hal ini membuat setiap *multi-tour* memiliki batas atas dan batas bawah dari waktu siklus. Pemenuhan permintaan hanya dapat memenuhi jumlah permintaan dalam batas waktu yang ada.

Dalam batasan ini waktu dilakukan formulasi untuk mencari waktu siklus optimal untuk bisa meminimalkan biaya. Waktu siklus ini dinotasikan sebagai $T_{EOQ}(C)$. $T_{EOQ}(C)$ berusaha untuk meminimalkan empat elemen biaya yang didefinisikan dalam IRP. Biaya operasi tetap (*fix operating cost*) yang dinotasikan sebagai ψ per jam. Biaya transportasi (*transportation cost*) $\delta \cdot v \cdot T_{min}(C)$ dimana δ adalah biaya transportasi per km dan v adalah kecepatan rata-rata (Km/jam). Biaya *delivery handling* yang dinotasikan sebagai φ . Biaya penyimpanan (*stock holding cost*), $\frac{1}{2} \eta_i d_i T^v$ dimana η adalah biaya *holding cost* pada titik penjualan i . Notasi d_i adalah laju permintaan (ton/jam) di titik penjualan i , sedangkan T^v adalah waktu siklus multi-tour C . Biaya-biaya yang ada ini akan dikonversikan menjadi biaya per jam sehingga akan didapatkan perumusan seperti pada rumus (5).

Nilai $T_{EOQ}(C)$ yang berada diluar batas atas dan bawah waktu siklus membuat waktu siklus ditentukan paling mendekati $T_{EOQ}(C)$ namun masih dalam batas. Perumusan $T_{EOQ}(C)$ dapat dilihat pada rumus (4).

Formulasi biaya IRP multi-tour

Perumusan model matematis yang diformulasikan oleh Aghezzaf, et. al. [1] merupakan *mixed integer programming*. Rumus (5) adalah fungsi tujuan yang akan diminimalkan.

Batasan pada rumus (6) sampai dengan rumus (13) adalah batasan yang diselesaikan dengan metode *column generation*. Penyelesaian dengan metode PSO membuat beberapa batasan seperti (6), (7), (10) akan terpenuhi apabila PSO dapat menghasilkan solusi *multi-tour* yang *feasible*. Solusi yang *feasible* adalah solusi yang dapat memenuhi semua batasan.

Waktu siklus minimal untuk setiap *multi-tour*

$$T_{min}(C) = \sum_{i=1, \dots, m} T_{min}(C_i) \tag{1}$$

Waktu siklus maksimal untuk setiap *sub-tour*

$$T_{max}(C_i) = k / \sum_{j \in C_i} d_j \tag{2}$$

Waktu siklus maksimal untuk setiap *multi-tour*

$$T_{max}(C) = \min_{i=1 \dots m} \{T_{max}(C_i)\} \tag{3}$$

Waktu siklus optimal untuk setiap *multi-tour*

$$T_{EOQ}(C) = \sqrt{\frac{\delta \cdot v \cdot T_{min}(C) + \sum_{i \in C} \varphi_i}{\sum_{i \in C} (\eta_i \cdot d_i / 2)}} \tag{4}$$

Fungsi tujuan model

$$Z = \sum_{v \in V} \left[\psi^v y^v + \frac{1}{T^v} \left(\sum_{i \in S^+} \sum_{j \in S^+} \delta v t_{ij} x_{ij}^v \right) + \sum_{i \in S} \left(\varphi_i \frac{1}{T^v} + \frac{1}{2} \eta_i d_i T^v \right) \left(\sum_{j \in S^+} x_{ij}^v \right) \right] \tag{5}$$

Batasan model

$$\sum_{v \in V} \sum_{i \in S^+} x_{ij}^v = 1 \tag{6}$$

Untuk semua $j \in S$

$$\sum_{v \in V} x_{ij}^v - \sum_{k \in S^+} x_{jk}^v = 0 \tag{7}$$

Untuk semua $v \in V, j \in S^+$

$$\sum_{i \in S^+} \sum_{j \in S^+} t_{ij}^v x_{ij}^v - T^v \leq 0 \tag{8}$$

Untuk semua $v \in V$

$$\sum_{v \in V} \sum_{i \in S^+} z_{ij}^v - \sum_{v \in V} \sum_{k \in S^+} z_{jk}^v = d_j \tag{9}$$

Untuk semua $j \in S$

$$x_{rj}^v - y^v \leq 0 \tag{10}$$

Untuk semua $v \in V, j \in S$

$$T^v z_{ij}^v \leq k(v) \tag{11}$$

Untuk semua $v \in V, j \in S$

$$z_{ij}^v - \left(\sum_{k \in S} d_k \right) x_{ij}^v \leq 0 \tag{12}$$

Untuk semua $v \in V, j \in S^+$

$$x_{ij}^v \in \{0,1\}, z_{ij}^v \geq 0, y^v \in \{0,1\}, T^v \geq 0 \tag{13}$$

Untuk semua $v \in V, i, j \in S^+$

Algoritma PSO M1

PSO merupakan sebuah algoritma optimasi yang bertujuan untuk menyelesaikan permasalahan dengan variabel real atau *continuous variable*. Dalam IRP terdapat permasalahan pengalokasian urutan titik penjualan. Prinsip pengurutan lokasi yang akan dituju dalam algoritma PSO M1 oleh Farhang Moghaddam, et. al. [5] menjadi dasar pengembangan algoritma PSO dalam penelitian yang dilakukan.

Definisi Dimensi

Partikel yang terdapat dalam algoritma PSO M1 akan mewakili urutan lokasi yang akan dituju oleh kendaraan yang tersedia. Sebuah partikel akan memiliki dimensi dengan ukuran $3n$, dimana n adalah jumlah lokasi yang akan dituju atau *salespoints* yang ada. Deret pertama berisi urutan lokasi yang akan dituju deret kedua dan ketiga berisi mengenai pengalokasian lokasi ke kendaraan yang ada. Setiap dimensi adalah bilangan bulat dengan rentang antara 0 sampai dengan 99.

Fitness Distance Ratio (FDR)

Pengaturan bobot informasi yang diterima partikel juga dipengaruhi oleh perumusan *fitness distance ratio* (FDR). Pendekatan dengan menggunakan (FDR) yang telah dirumuskan akan membuat partikel tersebut mengalami pembelajaran. Pengaruh setiap informasi pada partikel akan bergantung pada pembobotan yang akan diberikan pada informasi yang ada. Perumusan tersebut menjabarkan kondisi yang dapat mengatur bobot informasi yang diberikan pada setiap partikel. Bobot informasi dari sebuah kecepatan tersebut adalah inersia kecepatan dari partikel. perumusan inersia memakai FDR dapat dilihat pada rumus (14).

FDR akan membuat bobot informasi yang diberikan ke partikel dapat menyesuaikan seiring dengan berjalannya iterasi. Rumus (14) secara tidak langsung menunjukkan bahwa semakin mendekati akhir iterasi, bobot inersia semakin kecil. Hal ini menyebabkan partikel dapat melakukan pencarian di daerah yang sudah mendekati optimal dengan lebih seksama.

Hasil dan Pembahasan

Bagian ini membahas algoritma PSO yang dikembangkan dan verifikasi performa dari algoritma. Dalam algoritma PSO, dimensi yang dihasilkan oleh PSO akan dikonversikan menjadi sebuah *multi-tour* dengan metode pembobotan heuristik.

Dalam algoritma PSO yang dibuat, terdapat p partikel dengan s dimensi (dijelaskan lebih lanjut pada bagian Definisi Dimensi). Pergerakan setiap partikel dipengaruhi oleh dua informasi yaitu informasi kognitif atau *personal best*, dan informasi sosial atau *global best*. Kedua informasi memiliki pembobotan dimana pembobotan secara berturut-turut dinotasikan sebagai $c1$ dan $c2$. Prinsip FDR diterapkan dalam algoritma PSO dan hasil perumusan dapat dilihat pada rumus (15) dan (16).

Selama belum ditemukannya solusi yang *feasible*, nilai inersia tetap menggunakan nilai inersia awal (dihasilkan kecepatan yang tinggi). Hal ini dapat

dilihat pada rumus (15). Rumus (16) merupakan kondisi inersia bila ditemukan solusi yang *feasible*.

Kecepatan dipengaruhi oleh inersia informasi kognitif dan sosial. Rumus kecepatan dan pengaturan posisi dapat dilihat pada rumus (17) dan (18).

Langkah-langkah algoritma PSO dalam bagian ini dituliskan dalam bentuk *pseudo-code* untuk dimasukkan ke dalam bahasa pemrograman komputer. *Pseudo-code* dapat dilihat pada Tabel 1.

Inisialisasi termasuk penentuan parameter dalam PSO, penentuan nilai *global best* dan *personal best* ($pvalue$ dan $gvalue$) dan nilai dimensinya ($g(k)$; $p(j, k)$). Parameter seperti bobot kognitif ($c1$), sosial ($c2$), inersia awal (w_{in}), inersia akhir (w_f) juga ditentukan. Parameter lain seperti *penalty value* untuk nilai *fitness* bila solusi tidak *feasible* juga ditentukan.

Dalam algoritma PSO dilakukan pencarian solusi dengan memakai dua metode pembobotan heuristik yang berbeda. Hal ini membuat dilakukan pencarian solusi memakai metode pembobotan metode 1 dan setelah itu menggunakan metode 2. Dalam *pseudo-code* hal ini diwakili dengan notasi 1 dalam Tabel 1. Notasi ini menentukan pencarian PSO menggunakan metode pembobotan heuristik yang mana. Notasi *bestvalue* dan *bestparticle* merupakan nilai dan dimensi dari solusi terbaik diantara kedua metode pembobotan heuristik.

$$w(\alpha) = w(F) + \frac{\alpha - F}{1 - F}(w(I) - w(F)) \quad (14)$$

$$w(\alpha) = w(F) \quad (15)$$

$$w(\alpha) = w(F) + \frac{\alpha}{F}(w(I) - w(F)) \quad (16)$$

$$v_{ps}(\alpha + 1) = w(\alpha) \times v_{ps}(\alpha) + u[0,1] \times c1(\alpha) \times (x(g)_s - x_{ps}(\alpha)) + u[0,1] \times c2(\alpha) \times (x(l)_{ps} - x_{ps}(\alpha)) \quad (17)$$

Untuk semua $p; s$

$$x_{ps}(\alpha + 1) = x_{ps}(\alpha) + v_{ps}(\alpha + 1) \quad (18)$$

Untuk semua $p; s$

Tabel 1. Pseudo code algoritma PSO

```

Inisialisasi
for l=1:2
  for i=1:iterasi_maksimal
    if gvalue<penalty value
      w(i)=w_f+(i)/(iterasi_maksimal)*(w_f-w_in)
    else
      w(i)=w_in
    end
    for j=1:jumlah_partikel
      for k=1:jumlah_dimensi
        v(j,k)=v(j,k)*w(i)+rand(0,1)*c1*(x(j,k)-
          p(j,k))+rand(0,1)*c2*(x(j,k)-g(k))
        x(j,k)=x(j,k)+v(j,k)
      end
      Pembentukan multi-tour(j), heuristik(l)
      Perhitungan fitness(j)
      if fitness(j)<pvalue(j)
        pvalue(j)=fitness(j)
        for k=1:jumlah dimensi
          p(j,k)=x(j,k)
        end
      end
      if fitness(j)<gvalue
        gvalue=fitness(j)
        for k=1:jumlah dimensi
          g(k)=x(j,k)
        end
      end
    end
  end
  if gvalue<bestvalue
    bestvalue=gvalue
    for k=1:jumlah dimensi
      bestparticle(k)=g(k)
    end
  end
end
end

```

Definisi Dimensi

Sama halnya dengan yang telah dijelaskan pada bagian Metode Penelitian, dimensi dari partikel adalah sebesar jumlah titik penjualan yang ada.

Terdapat sekumpulan S titik penjualan yang akan dilayani oleh sebuah DC r . Urutan titik penjualan yang akan dituju oleh *vendor*, dinotasikan sebagai $s = \{N, Re\}$. dimana $N = (1 \dots S)$ dan Re merupakan nilai dimensi pada setiap partikel yang berupa bilangan real. Nilai dari setiap dimensi ini kemudian akan diurutkan sehingga membentuk urutan yang akan dituju. Hasil pengurutan ini nantinya akan dialokasikan ke sejumlah kendaraan yang ada. Notasi s diurutkan berdasarkan nilai Re dari urutan nilai terbesar ke terkecil.

Kondisi sebelum diurutkan dapat dilihat pada Tabel 2, dimana nilai Re dari setiap lokasi adalah bilangan

real. Nilai ini diurutkan dan hasil pengurutan berupa Seq dapat dilihat pada Tabel 3.

Tabel 2. Keadaan awal dari sebuah partikel

Urutan	1	2	3	4	5
N	1	2	3	4	5
Re	0,44	0,85	0,87	0,02	0,97

Tabel 3. Hasil pengurutan dari notasi s

Urutan	4	1	2	3	5
Seq	0,02	0,44	0,85	0,87	0,97

Metode Pengalokasian Kendaraan

Sebelum dilakukan pembentukan *multi-tour* dilakukan alokasi titik penjualan ke sejumlah kendaraan yang ada. Langkah pertama dalam pembagian urutan adalah memberikan bobot pada semua titik penjualan yang ada. Pembagian bobot menggunakan dua jenis cara. Hal ini dilakukan untuk mengurangi kemungkinan algoritma PSO terjebak ke dalam kondisi konvergen prematur atau mencapai kondisi *local optimum* saja.

Metode Pembobotan Heuristik 1

Cara pertama adalah melakukan pembobotan dengan mempertimbangkan permintaan pada titik penjualan saja. Perumusan dapat dilihat pada rumus (19).

$$we_i = d_i \times v \times t_{ri}; i \in \{Seq_1 \dots Seq_n\} \quad (19)$$

Metode Pembobotan Heuristik 2

Bagian ini mempertimbangkan pembobotan dengan permintaan pada titik penjualan dan juga bobot antara kemungkinan lokasi sebelum dan sesudah penjualan untuk setiap titik penjualan. Terdapat dua pertimbangan dalam pembobotan yaitu kondisi asal kendaraan sebelum berangkat ke titik penjualan i dan kondisi tujuan berikut dari kendaraan setelah mengunjungi titik penjualan i . Setiap kondisi tersebut akan memiliki dua kemungkinan. Kemungkinan pertama adalah kendaraan berangkat dari DC r atau kendaraan itu melanjutkan rute ke *salespoint* berikutnya. Hal ini membuat kendaraan dari *salespoint* i memiliki empat kemungkinan rute yaitu $r - i; (i - 1) - i; i - r; i - (i + 1)$. Perumusan kemungkinan ini dapat dilihat pada rumus (20).

$$\begin{aligned}
 we_{1i} &= \begin{cases} d_i \times v \times t_{i(i+1)}; i \in \{Seq_1 \dots Seq_{n-1}\} \\ 0; i \in \{Seq_n\} \end{cases} \\
 we_{2i} &= \begin{cases} d_i \times v \times t_{(i-1)i}; i \in \{Seq_2 \dots Seq_n\} \\ 0; i \in \{Seq_1\} \end{cases} \\
 we_{3i} &= d_i \times v \times t_{ir}; i \in \{Seq_1 \dots Seq_n\} \\
 we_{4i} &= d_i \times v \times t_{ri}; i \in \{Seq_1 \dots Seq_n\}
 \end{aligned} \quad (20)$$

Bobot untuk setiap salespoint

Keempat kemungkinan itu akan dijumlahkan dan menjadi bobot pada titik penjualan *i*. Perumusan dapat dilihat pada rumus (21).

$$we_i = we_{1i} + we_{2i} + we_{3i} + we_{4i}; i \in \{Seq_1 \dots Seq_n\} \tag{21}$$

Pengalokasian kendaraan berdasar bobot

Total bobot dari semua titik penjualan yang didapat baik menggunakan metode pembobotan heuristik 1 dan 2 kemudian dijumlahkan. Hasil penjumlahan untuk setiap *salespoint* dapat dilihat pada rumus (22)

Didefinisikan sebuah notasi yang dapat membagi rata titik penjualan berdasarkan bobotnya terhadap jumlah kendaraan yang ada. Batas pembagian bobot dinotasikan sebagai *Wn* dan notasi *V* adalah jumlah kendaraan. Perumusan dapat dilihat dapat dilihat pada rumus (23).

$$W = \sum_{i \in S} we_i \tag{22}$$

$$Wn = W/V \tag{23}$$

Rumus (23) digunakan untuk bisa melakukan pengalokasian *salespoint* ke sejumlah kendaraan. Contoh pembobotan dapat dilihat dengan menggunakan informasi jarak dan laju permintaan pada Tabel 4 dan Tabel 5. Diasumsikan tersedia 2 kendaraan yang akan dipakai untuk melakukan pemenuhan permintaan.

Dilakukan penjumlahan secara berurutan dari awal lokasi sampai mendekati selisih terkecil dengan batas pembagian bobot. Kumpulan *we_i* *salespoint* yang ada dijumlahkan, dimana akumulasi yang memiliki selisih kecil dengan *Wn* menjadi alokasi kelompok *salespoint* ke sebuah kendaraan. Tabel 6 adalah nilai *we_i* dan *Wn* dari *salespoint* untuk kedua metode heuristik. Langkah ini dilakukan secara berulang untuk sejumlah kendaraan yang ada, dengan melanjutkan alokasi mulai dari kota yang belum dialokasikan secara berurutan. Hasil pengalokasian dapat dilihat pada Tabel 7.

Metode Pembentukan Sub-tour Dalam Multi-tour

Hasil pengalokasian lokasi yang ada ke kendaraan dilanjutkan dengan tahapan berikutnya yaitu pembentukan *sub-tour* pada setiap *multi-tour*. pembentukan dilakukan dengan metode randomisasi jumlah lokasi dari angka 1 sampai dengan jumlah lokasi dalam sebuah *multi-tour*. Contoh dari pengalokasian dari metode heuristik 1 terdiri dari 2

multi-tour. *Multi-tour* 1 hanya akan memiliki 1 *sub-tour* karena hanya ada satu *salespoint*. *Multi-tour* 2 dapat digunakan sebagai contoh randomisasi. Langkah pertama, dilakukan randomisasi antara 1 sampai 4, dan didapat nilai random berturut-turut adalah {2,4,1,1}. Hal ini membuat *sub-tour* 1 pada *multi-tour* 2 adalah 2 *salespoint* pertama. Angka 4 diabaikan karena melebihi dari sisa jumlah *salespoint* yang belum dialokasikan. Angka random berikutnya membuat terbentuknya *sub-tour* 2 dan *sub-tour* 3 yang mengalokasikan lokasi *salespoint* secara berurutan. Setiap *sub-tour* diawali dari DC *r* dan kemblai ke DC *r*. Rangkuman hasil pembentukan *sub-tour* dapat dilihat pada Tabel 8.

Tabel 4. Jarak antar lokasi (kilometer)

Lokasi	r	1	2	3	4	5
r	0	60	66.3	52.3	95.1	48.1
1	60	0	86.1	75.7	57.2	42.2
2	66.3	86.1	0	14	74.1	43.9
3	52.3	75.7	14	0	73	34.7
4	95.1	57.2	74.1	73	0	48.3
5	48.1	42.2	43.9	34.7	48.3	0

Tabel 5. Laju permintaan (ton/jam)

Salespoint	1	2	3	4	5
Demand rate	0,54	0,19	0,72	0,98	0,60

Tabel 6. Contoh perhitungan pembobotan dengan dua metode pembobotan heuristik

Seq	4	1	2	3	5	Wn
Heuristik 1	93,2	32,4	12,6	37,7	28,9	102,4
Heuristik 2	243	142	44,2	110	78,5	308,9

Tabel 7. Contoh perhitungan pengalokasian titik penjualan ke kendaraan

	Metode Heuristik 1	Metode Heuristik 2
<i>Multi-tour</i> 1	{4}	<i>Multi-tour</i> 1 {4}
<i>Multi-tour</i> 2	{1,2,3,5}	<i>Multi-tour</i> 2 {1,2,3}
		<i>Multi-tour</i> 3 {5}

Tabel 8. Contoh perhitungan pembentukan *multi-tour* kendaraan

Multi-tour	Sub-tour	Rute
<i>Multi-tour</i> 1	<i>Sub-tour</i> 1	{r,4,r}
<i>Multi-tour</i> 2	<i>Sub-tour</i> 1	{r,1,2,r}
<i>Multi-tour</i> 3	<i>Sub-tour</i> 2	{r,3,r}
<i>Multi-tour</i> 4	<i>Sub-tour</i> 3	{r,5,r}

Verifikasi Kemampuan PSO

Verifikasi bertujuan untuk melakukan pengujian apakah model algoritma PSO untuk menyelesaikan model IRP *multi-tour* sudah bisa menghasilkan solusi yang mendekati kondisi optimal. Verifikasi

dan validasi dilakukan dengan menggunakan studi kasus pada penelitian yang dilakukan oleh Aghezzaf. Parameter yang digunakan dalam algoritma PSO dapat dilihat pada Tabel 9.

Dimensi yang terdapat pada PSO disesuaikan dengan jumlah lokasi *salespoint* yang terdapat pada studi kasus dari Aghezzaf, et. al. [1]. Iterasi pembentukan *sub-tour* maksimal adalah 25 kali, apabila lebih dari 25 kali percobaan belum terbentuk solusi yang *feasible* maka akan dihasilkan solusi yang bersifat *infeasible* dan nilai biaya total adalah sebesar *penalty value*. Nilai *penalty value* sangat besar agar membuat algoritma PSO menghindari solusi yang *infeasible*.

Verifikasi model algoritma PSO dilakukan dengan beberapa cara yaitu kemampuan perbaikan solusi, konsistensi dan kualitas solusi yang dihasilkan.

Signifikansi perbaikan solusi

Perbandingan dilakukan dengan membandingkan solusi awal yang *feasible* dengan solusi akhir sehingga performa algoritma PSO bisa diketahui. Semakin besar penurunan biaya yang terjadi semakin menunjukkan bahwa terjadi perbaikan yang semakin signifikan dalam penemuan solusi yang dilakukan.

Tabel 10 menunjukkan bahwa bahwa algoritma PSO baik menggunakan algoritma pembobotan heuristik 1 ataupun 2, dapat menghasilkan perbaikan solusi yang signifikan. Peningkatan kualitas solusi apabila dirata-rata memiliki perbedaan 10% lebih baik dari solusi awal. Hal ini menunjukkan algoritma PSO mampu menghasilkan perbaikan solusi yang signifikan (lebih besar dari 0%).

Konsistensi solusi PSO

Semakin kecil perbedaan antara replikasi PSO menunjukkan bahwa solusi yang dihasilkan algoritma PSO semakin stabil dan mampu menghasilkan solusi yang konvergen. Pengujian konsistensi dilihat dari *coefficient of variance* (CV) atau rasio antara standar deviasi dengan nilai rata-rata antara setiap replikasi. CV dipakai sebagai pembanding karena dapat mewakili rasio penyimpanan yang terjadi dalam bentuk persentase. Antara replikasi akan dibandingkan persentase penyimpanan yang dapat terjadi. Pengukuran nilai CV dapat dilihat pada

Tabel 11.

Nilai CV menunjukkan bahwa dari 5 kali replikasi, algoritma PSO dapat dengan konsisten menghasilkan solusi dengan kualitas yang sama. Hasil perhitungan menunjukkan bahwa pembobotan heuristik metode 1 metode 2 memiliki konsistensi yang hampir sama (1,62%; 1,27%). Nilai CV untuk algoritma PSO metode 1 dimana rata-rata CV

adalah 1,44%. Hal ini menunjukkan bahwa rata-rata penyimpangan yang terjadi antara replikasi cukup baik memakai metode pembobotan heuristik 1 dan 2. Nilai persentase yang kecil ini (dibawah 5%) menunjukkan solusi akhir yang dihasilkan dari algoritma PSO bukan faktor kebetulan.

Perbandingan Kemampuan PSO

Bagian ini melakukan perbandingan antara solusi yang dihasilkan *column generation* yang diformulasikan Aghezzaf, et. al. [1] dengan algoritma PSO dalam penelitian. Perbandingan dilakukan dengan membandingkan biaya total yang dihasilkan *multi-tour* yang dihasilkan penelitian Aghezzaf, et. al., (2005) [1] karena adanya keterbatasan waktu.

Perbandingan kualitas solusi dapat dilihat pada Tabel 12. Kualitas solusi yang dihasilkan oleh algoritma PSO diuji dengan cara membandingkan kualitas solusi yang dihasilkan sama dengan solusi yang dihasilkan metode *column generation* yang dirumuskan Aghezzaf, et. al., (2005). Pengujian menggunakan metode statistik *one sample t*. Pengujian dilakukan dengan bantuan *software* Minitab 16. Hasil pengujian menunjukkan terjadi kondisi gagal tolak H_0 dengan nilai *p-value* sebesar 0,168 (lebih besar dari $\alpha = 0,05$). Dapat dikatakan bahwa kelima replikasi algoritma PSO menghasilkan solusi dengan kualitas yang sama dengan *column generation*. Algoritma PSO bahkan dapat menemukan solusi yang lebih baik (185,981) daripada *column generation* (186,7). Hal ini menunjukkan bahwa performa algoritma PSO yang dibuat dapat dikatakan mampu menghasilkan solusi dengan kualitas yang sama dengan metode *column generation* yang dibuat oleh Aghezzaf, et. al., (2005) [1].

Tabel 9. Parameter PSO

Parameter PSO metode 1	
Dimensi (jumlah titik penjualan)	15
Jumlah partikel	40
Iterasi maksimum	1000
c1	2
c2	1,5
w_in	0,8
w_f	0,4
Iterasi pembentukan <i>sub-tour</i>	25

Tabel 10. Persentase selisih *total cost* antara solusi awal iterasi yang *feasible* dengan solusi pada akhir

Metode	Pembobotan		
	Heuristik 1	Heuristik 2	Rata-rata
Rep 1	10,15%	10,58%	10,37%
Rep 2	14,48%	13,33%	13,91%
Rep 3	5,27%	12,88%	9,07%
Rep 4	6,09%	8,06%	7,08%
Rep 5	13,45%	7,03%	10,24%
Rata-rata replikasi	9,89%	10,38%	10,13%

Tabel 11. Hasil perhitungan CV antara replikasi

Metode	Jumlah Replikasi	Rata-rata	St. dev	CV
Pembobotan Heuristik 1	5	199,714	3,241	1,62%
Pembobotan Heuristik 2	5	200,097	2,534	1,27%
Rata-rata	-	199,905	2,888	1,44%

Tabel 12. Perbandingan *total cost* dari rute yang dihasilkan oleh PSO, dan *column generation*

Metode	Hasil per replikasi	Hasil terbaik
Algoritma PSO	190.233	185.981
	185.981	
	187.1	
	190.025	
	187.298	
<i>Column Generation</i> (Aghezzaf, dkk., 2005)	186,7	

Simpulan

IRP merupakan sebuah permasalahan *np-hard* dimana untuk menyelesaikan permasalahan ini dikembangkan metode-metode tertentu. dalam penelitian ini dikembangkan algoritma PSO untuk bisa memberikan solusi penyelesaian IRP *multi-tour* yang lebih kompleks dari IRP biasa. Algoritma PSO dalam penelitian merupakan pengembangan PSO agar dapat mengatasi masalah pengalokasian titik penjualan menjadi *multi-tour*.

Algoritma PSO yang dibuat memerlukan verifikasi dalam performa. Verifikasi dilihat melalui konsistensi kedua algoritma, signifikansi perbaikan solusi dari awal sampai akhir iterasi, kualitas solusi yang ditemukan. Didapatkan bahwa algoritma PSO memiliki kemampuan yang menjanjikan dimana dalam menghasilkan solusi terbaik, kualitas solusi yang dihasilkan sama dengan metode *column generation*.

Daftar Pustaka

1. Aghezzaf, E.-H., Raa, B., & Van Landeghem, H. (2005). Modeling Inventory Routing Problem In Supply Chains of High Consumption Products.
2. Archetti, C., Bertazzi, L., Laporte, G., & Grazia Speranza, M. (2006). A Branch-and-Cut Algorithm for a Vendor-Managed Inventory-Routing Problem.
3. C. Coelho, L., & Laporte, G. (2012). The exact solution of several classes of Inventory Routing Problems, introduction.
4. Campbell, A., Clarke, L., Kleywegt, A., & Savlesbergh, M. (1997). The Inventory Routing Problem.
5. Farhang Moghaddam, B., Ruiz, R., & Jafar Sadjadi, S. (2011). Vehicle Routing Problem With Uncertain Demands: An Advanced Particle Swarm Algorithm.
6. Melissa Campbell, A., & Savelsbergh, M. W. (2002). A Decomposition Approach for the Inventory-Routing Problem.
7. Raa, B., & Aghezzaf, E.-H. (2007). A practical solution approach for the cyclic Inventory Routing Problem.
8. Rau, H., Wu, M.-Y., & Wee, H.-M. (2003). Integrated inventory model for deteriorating items under a multi-echelon supply chain environment.