

ALGORITMA *SIMULATED ANNEALING* UNTUK MENENTUKAN RUTE KENDARAAN HETEROGEN (STUDI KASUS)

Andriansyah*¹, Rizky Novatama², Prima Denny Sentia³

^{1,2,3}Teknik Industri, Fakultas Teknik, Universitas Syiah Kuala
E-mail: ¹andriansyah@unsyiah.ac.id, ²rizkynovatama@gmail.com, ³primadennysentia@unsyiah.ac.id
*Penulis Korespondensi

(Naskah masuk: 16 Mei 2019, diterima untuk diterbitkan: 05 Oktober 2020)

Abstrak

Permasalahan transportasi dalam *supply chain management* sangat penting untuk dikaji karena dapat menimbulkan biaya logistik yang sangat besar. Salah satu cara untuk mengurangi biaya transportasi adalah dengan penentuan rute kendaraan atau dikenal dengan istilah *vehicle routing problem*. Objek yang menjadi kajian merupakan perusahaan yang bergerak pada bidang distribusi produk untuk area kota Banda Aceh dan sekitarnya. Dalam proses distribusi, perusahaan ini menggunakan dua jenis kendaraan dengan kapasitas dan biaya operasional yang berbeda sehingga permasalahan menjadi *heterogeneous fleet vehicle routing problem*. Penentuan rute kendaraan dalam penelitian ini dilakukan dengan tiga metode, yaitu metode analitik, algoritma *insertion heuristic* sebagai metode heuristik, dan algoritma *simulated annealing* sebagai metode metaheuristik. Berdasarkan hasil yang diperoleh dari data uji coba, algoritma *simulated annealing* merupakan algoritma yang paling baik dalam menyelesaikan permasalahan. Secara rata-rata, algoritma *simulated annealing* dapat menghasilkan kualitas solusi yang sama dengan metode analitik, namun dengan waktu komputasi yang lebih singkat. Selain itu, algoritma *simulated annealing* menghasilkan kualitas solusi yang lebih baik dibandingkan algoritma *insertion heuristic* yang dikembangkan dalam penelitian dan dapat meningkatkan kualitas solusi sebesar 20,18% dari penelitian sebelumnya dengan waktu komputasi 19,27 detik.

Kata kunci: *Vehicle Routing Problem, Heterogeneous Fleet Vehicle Routing Problem, Metaheuristic, Simulated Annealing*

SIMULATED ANNEALING ALGORITHM FOR HETEROGENEOUS VEHICLE ROUTING PROBLEM (CASE STUDY)

Abstract

Transportation problems in *supply chain* are very important to be discussed because they can raise enormous logistic cost. Route determination of the vehicles known as *vehicle routing problem* is the one of ways to reduce transportation cost. The object discussed in this study is the distribution company for Banda Aceh city and its surroundings. The company uses two types of vehicle to distribute the product for customers. The differences each vehicle are vehicle capacity and operational cost. To cover these differences, the problem becomes *heterogeneous fleet vehicle routing problem*. The study uses three methods to solve the problem. Analytical method, *insertion heuristic algorithm* as heuristic method and *simulated annealing algorithm* as metaheuristic method are the methods used. According to the results, *simulated annealing algorithm* produces the better solutions than two others. On average, solutions produced by *simulated annealing algorithm* from dataset have same quality with analytical method, but with faster computation. Furthermore, *simulated annealing algorithm* produces better quality of solutions than *insertion heuristic algorithm* both from this study and previous study. The solution improves 20,18% with computation time 19,27 seconds.

Keywords: *Vehicle Routing Problem, Heterogeneous Fleet Vehicle Routing Problem, Metaheuristic, Simulated Annealing*

1. PENDAHULUAN

Salah satu strategi yang dilakukan perusahaan untuk meningkatkan kualitas bisnisnya adalah dengan mengelola *Supply Chain Management* atau

SCM (Borade & Bansod, 2007). SCM mengelola jaringan dari perusahaan-perusahaan yang saling berkolaborasi dalam menyuplai bahan baku, memproduksi barang, maupun mengirimkannya ke konsumen. Salah satu bagian penting dari SCM

adalah manajemen distribusi dan transportasi atau sering juga disebut sebagai manajemen logistik (Pujawan & Er, 2017). Didalam perencanaan logistic, salah satu komponen penting yang harus diperhatikan yaitu masalah transportasi. Transportasi bisa saja memiliki biaya 33% - 67% dari total biaya logistik. Hal inilah yang menyebabkan perusahaan untuk mengurangi biaya transportasi, salah satunya adalah merencanakan rute kendaraan dengan biaya yang minimum. Masalah transportasi ini lebih dikenal dengan istilah *Vehicle Routing Problem* atau VRP (Wirdianto, Jonrinaldi, & Surya, 2007).

VRP klasik dikenalkan pertama kali oleh Dantzig dan Ramser pada tahun 1959 dimana solusi yang ingin dicapai dari VRP klasik yaitu dapat menentukan rute optimal untuk kendaraan dengan jenis yang sama (Koç, Bektaş, Jabali, & Laporte, 2015). Menurut Hoff, Andersson, Christiansen, Hasle, & Løkketangen (2010) sebuah industri bisa saja memiliki kendaraan yang berbeda, dimana kendaraan tersebut memiliki karakteristik yang berbeda, teknologi yang berbeda dan biaya depresiasi yang berbeda, permasalahan ini dikenal dengan istilah *Heterogeneous Fleet*. Oleh karena itu pada tahun 1984 Golden beserta peneliti lainnya memperkenalkan salah satu varian VRP yaitu *Heterogeneous Fleet Vehicle Routing Problem* atau HFVRP (Koç dkk, 2015).

HFVRP adalah salah satu jenis VRP yang kendaraannya memiliki kapasitas, *fixed cost*, serta *variable cost* yang berbeda (Imran, Salhi, & Wassan, 2009). Menurut Caric & Gold (2008) VRP juga merupakan masalah optimasi kombinatorial yang sulit dalam pencarian solusi optimal. Oleh karena itu, metode analitik hanya dapat diselesaikan jika cakupan distribusinya kecil. Sebagai contoh penentuan rute kendaraan sederhana dengan 10 titik distribusi, jika menggunakan metode analitik maka jumlah solusi yang ada adalah sebesar 10 faktorial atau 3.628.800 jumlah solusi. Jika diumpamakan 1 solusi didapatkan dengan waktu selama 1 detik, maka untuk menemukan solusi yang terbaik dengan kasus 10 titik distribusi dibutuhkan waktu selama 42 hari. Menurut Wibisono (2015), metode analitik membutuhkan waktu yang lama, sehingga para peneliti saat ini beralih ke metode heuristik dan metaheuristik.

Salah satu algoritma heuristik yang cukup terkenal dan banyak digunakan dalam menyelesaikan kasus VRP adalah algoritma *Insertion Heuristic* (IH). Algoritma ini mudah diterapkan dan bahkan dapat dimodifikasi sedemikian rupa agar dapat memenuhi varian VRP yang semakin kompleks. Algoritma IH terkenal karena cepat, menghasilkan solusi yang baik, mudah digunakan, dan dapat dimodifikasi untuk mengatasi batasan yang lebih kompleks karena algoritma ini merupakan algoritma konstruksi yang menghasilkan solusi tunggal (Andriansyah & Sentia, 2018). Andriansyah & Sentia (2018) menggunakan

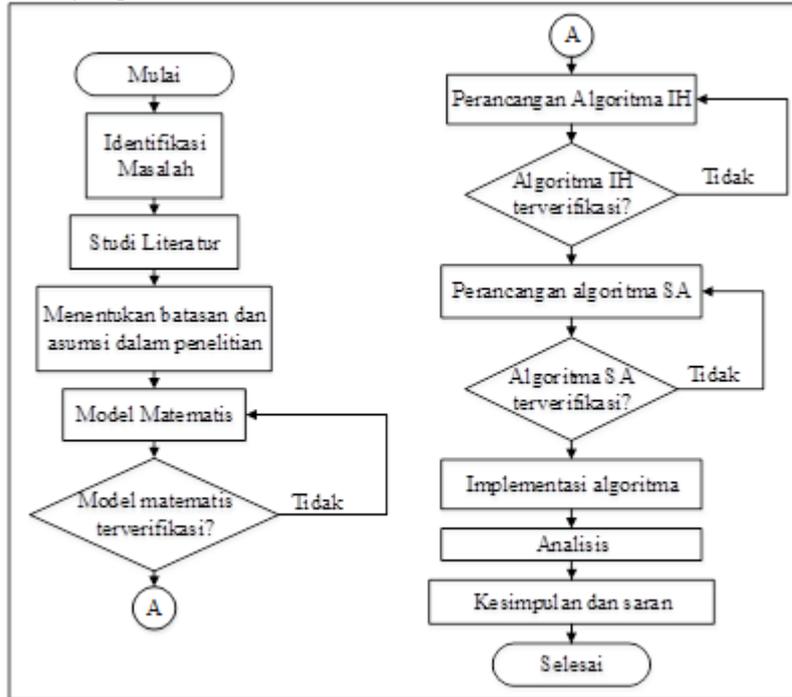
algoritma IH sebagai solusi awal dan perbaikan solusi menggunakan algoritma *local search* (LS) pada varian CVRP. Algoritma IH juga salah satu algoritma yang sering dipilih untuk membangkitkan sebuah solusi awal yang digunakan pada metode metaheuristik (Campbell & Savelsbergh, 2004). Pada penelitian sebelumnya algoritma IH terbukti dapat menghasilkan solusi yang lebih baik jika dibandingkan dengan solusi menggunakan metode heuristik lainnya (Dell'Amico, Monaci, Pagani, & Vigo, 2007; Dullaert, Janssens, Sörensen, & Vernimmen, 2002).

Algoritma *Simulated Annealing* (SA) banyak digunakan untuk menyelesaikan berbagai masalah optimasi kombinatorial yang sulit. Filosofi algoritma SA berasal dari proses *annealing* di *metallurgy* (Elhaddad, 2012). Menurut Talbi (2009) proses *annealing* dimulai dari pemanasan baja yang kemudian didinginkan secara perlahan supaya atom baja mencapai suatu struktur kristal yang kuat. Kekuatan dari struktur tersebut bergantung pada *cooling rate* dari proses pendinginan, struktur yang kuat berasal dari proses pendinginan secara perlahan dan hati-hati. Oleh karena itu, proses pendinginan memiliki peranan paling penting agar algoritma SA efektif dan efisien. Proses *annealing* tersebut kemudian dianalogikan ke dalam masalah optimasi, fungsi objektif pada masalah optimasi dianalogikan sebagai keadaan energi dari sistem, kemudian solusi dari masalah dianalogikan sebagai posisi atom dari baja. Kelebihan dari algoritma SA adalah dapat keluar dari solusi lokal optimal dengan cara melakukan "*hill-climbing moves*" (menerima solusi yang lebih buruk) dengan harapan mencapai solusi global optimal, semakin rendah suhu maka *hill-climbing moves* lebih jarang terjadi (Gendreau & Potvin, 2010). Beberapa penelitian sebelumnya yang menggunakan algoritma SA membuktikan bahwa algoritma ini dapat memperbaiki solusi awal menjadi solusi yang mendekati global optimal. Solusi tersebut dapat ditemukan dalam waktu komputasi yang singkat (Afifi, Dang, & Moukrim, 2013; Birim, 2016; Leung, Zhang, Zhang, Hua, & Lim, 2013; Yu, Redi, Hidayat, & Wibowo, 2017).

Objek yang menjadi kajian dalam penelitian adalah perusahaan distribusi yang beroperasi di Banda Aceh dan sekitarnya. Pada proses distribusinya, perusahaan ini menggunakan dua jenis kendaraan (A dan B). Kedua kendaraan ini memiliki biaya operasional yang berbeda sehingga kasus VRP pada perusahaan ini termasuk dalam HFVRP (Saputra dkk, 2017).

Perusahaan ini memiliki kebijakan bahwa kendaraan B akan digunakan jika seluruh kendaraan A telah digunakan. Hal ini dikarenakan perusahaan memiliki anggapan kendaraan tipe A memiliki kapasitas yang lebih besar dibanding kendaraan tipe B, sehingga dapat mendistribusikan produk ke banyak pelanggan. Seharusnya, perusahaan juga

mempertimbangkan untuk menggunakan kendaraan tipe B yang memiliki biaya operasional lebih kecil



Gambar 1. Diagram Alir Penelitian

agar biaya operasional keseluruhan lebih rendah jika dikombinasikan. Selain itu, perusahaan juga tidak memiliki kebijakan pemilihan rute optimal dalam proses distribusi, hal ini menyebabkan jarak tempuh dari kendaraan terlalu jauh dan menghabiskan banyak biaya (Saputra dkk, 2017).

Dalam penelitian ini kasus HFVRP dalam perusahaan tersebut akan diselesaikan menggunakan tiga pendekatan, yaitu analitik, heuristik, dan metaheuristik. Pendekatan secara analitik akan diselesaikan dengan menggunakan bantuan perangkat lunak optimasi LINGO, kemudian pendekatan secara heuristik dengan menggunakan algoritma IH, dan pendekatan secara metaheuristik dengan algoritma SA. Solusi yang didapatkan dari algoritma IH akan dijadikan solusi awal di algoritma SA. Solusi dari pendekatan secara analitik, heuristik, dan metaheuristik akan dibandingkan untuk melihat solusi yang baik dengan biaya terendah dan waktu komputasi yang sedikit. Hasil dari penelitian ini juga diharapkan dapat menentukan rute optimal untuk pendistribusian produk di perusahaan dengan meminimalkan biaya operasional perusahaan.

2. METODE PENELITIAN

2.1 Identifikasi Masalah dan Studi Literatur

Identifikasi masalah dilakukan dengan cara memahami permasalahan HFVRP di perusahaan yang telah dijelaskan di penelitian terdahulu oleh (Saputra dkk, 2017). Kemudian melakukan studi literatur untuk mendapatkan model matematis yang relevan terhadap kasus HFVRP yang diteliti, dan mencari metode atau

algoritma untuk memecahkan masalah tersebut. Setelah identifikasi masalah dan studi literatur, maka akan ditetapkan beberapa batasan dan asumsi agar objek yang diteliti lebih sederhana.

2.2 Pengumpulan Data

Pada tahap ini akan dilakukan pengumpulan data-data yang dibutuhkan untuk menyelesaikan kasus. Dalam penelitian ini data sekunder diperoleh dari penelitian sebelumnya yang telah dilakukan oleh (Saputra dkk, 2017). Data yang diperoleh berupa data permintaan, data matriks jarak, data biaya operasional, data jenis dan kapasitas kendaraan. Selain data tersebut akan dibangkitkan beberapa set data yang dapat mewakili kasus, data ini dinamakan data hipotetik.

2.3 Model Matematis

Model matematis yang digunakan dalam penelitian ini adalah model matematis dari penelitian (Saputra dkk, 2017). Model matematis ini berguna untuk menentukan fungsi tujuan dan batasan-batasan dari kasus HFVRP yang tidak boleh dilanggar. Kemudian melakukan verifikasi satuan, dimana satuan dari model matematis di ruas kiri harus sama dengan satuan dari model matematis di ruas kanan. Kemudian akan dilakukan verifikasi kelogisan model matematis dengan bantuan perangkat lunak optimasi LINGO 11.0 menggunakan algoritma *branch-and-bound*. Verifikasi ini dilakukan untuk memeriksa apakah hasil dari model matematis tidak melanggar seluruh batasan-batasan yang telah ditentukan

2.4 Algoritma *Insertion Heuristic*

Algoritma IH akan dirancang dengan menggunakan bahasa pemrograman Matlab R2015a. Didalam merancang algoritma IH akan diberlakukan *soft constraint* untuk batasan kapasitas dan *hard constraints* untuk batasan lainnya, sehingga hasil boleh melanggar batasan tersebut dan jika melanggar maka akan ada perhitungan biaya penalti. Hal ini dilakukan karena didalam penelitian ini hasil dari algoritma IH hanya digunakan sebagai solusi awal dalam algoritma SA, dimana solusi tersebut akan di *improve* pada algoritma SA. Algoritma IH dirancang menggunakan bahasa pemrograman MATLAB R2015a. Algoritma IH dikatakan terverifikasi apabila hasil dari algoritma tersebut tidak melanggar *hard constraints* yang telah ditentukan di model matematisnya.

2.4 Algoritma *Simulated Annealing*

Algoritma SA yang telah dirancang oleh Afifi dkk. (2013) akan dijadikan acuan dalam merancang algoritma SA di penelitian ini. Algoritma SA dirancang menggunakan bahasa pemrograman MATLAB R2015a. Algoritma SA dikatakan terverifikasi apabila hasil dari algoritma tersebut tidak melanggar seluruh batasan yang telah ditentukan di model matematisnya.

2.6 Implementasi Algoritma dan Analisis

Dalam pengolahan data spesifikasi komputer yang digunakan yaitu, *processor* Intel Core i7-7700HQ CPU @2.80GHz, sistem operasi windows 10, dan kapasitas RAM 16.0 GB. Analisis dilakukan dengan cara membandingkan hasil dari penyelesaian secara analitik, heuristik dan metaheuristik. Hasil penelitian ini juga akan dibandingkan dengan hasil penelitian sebelumnya yang telah dilakukan oleh (Saputra dkk, 2017) menggunakan metode heuristik, hal ini dilakukan untuk mengetahui besarnya peningkatan solusi yang telah dilakukan dari penelitian. Dalam penelitian ini akan dilakukan analisis sensitivitas untuk beberapa parameter algoritma SA. Analisis sensitivitas dilakukan untuk melihat bagaimana pengaruh solusi yang akan diperoleh jika nilai dari parameter SA berubah.

3. HASIL DAN PEMBAHASAN

3.1 Definisi Masalah

Model HFVRP didefinisikan sebagai sebuah grafik terarah $G = (N, A)$, dimana $N = \{0, \dots, n\}$ adalah himpunan node, dan $A = \{(i, j): 0 \leq i, j \leq n, i \neq j\}$ adalah himpunan busur. $N = 0$ adalah depot, sedangkan $N = 1, \dots, n$ merupakan pelanggan. Setiap pelanggan i memiliki permintaan yang tidak negatif sebesar d_i . c_{ij}^k merupakan biaya operasional

perjalanan di busur $(i, j) \in A$ oleh kendaraan k . $K = \{1, \dots, k\}$ merupakan himpunan dari jumlah kendaraan yang dimiliki dengan f^k dan U^k adalah biaya tetap kendaraan dan kapasitas kendaraan.

3.2 Model Matematis

$$\text{MIN } TC = \sum_{k=1}^K f^k z^k + \sum_{k=1}^K \sum_{i=1}^n \sum_{j=1}^n c_{ij}^k x_{ij}^k \quad (1)$$

$$\sum_{k=1}^K y_i^k = 1, \quad \forall i \in N \setminus \{0\} \quad (2)$$

$$\sum_{k=1}^K y_0^k \leq K \quad (3)$$

$$\sum_{j \in V \setminus \{i\}} x_{ij}^k = y_i^k \quad \forall i \in N, k = 1, \dots, K \quad (4)$$

$$\sum_{j \in V \setminus \{i\}} x_{ji}^k = y_i^k \quad \forall i \in N, k = 1, \dots, K \quad (5)$$

$$\sum_{i \in V} d_i y_i^k \leq U^k z^k \quad k = 1, \dots, K \quad (6)$$

$$\sum_{i \in S} \sum_{j \in S \setminus \{i\}} x_{ij}^k \leq |S| - 1 \quad \forall S \subseteq N \setminus \{0\}, |S| \geq 2, k = 1, \dots, K \quad (7)$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall i, j = 1, 2, \dots, n \quad \forall k = 1, 2, \dots, K \quad (8)$$

$$y_i^k \in \{0, 1\}, \quad \forall k = 1, 2, \dots, K \quad (9)$$

$$z^k \in \{0, 1\}, \quad \forall k = 1, 2, \dots, K \quad (10)$$

Persamaan (1) merupakan fungsi tujuan dari model matematis yang meminimumkan biaya operasional. Pembatas (2) membatasi bahwa setiap kendaraan k hanya boleh mengunjungi lokasi i satu kali oleh kendaraan k , kecuali depot. Pembatas (3) mengatakan bahwa kendaraan k yang keluar dari depot untuk melakukan proses distribusi tidak melebihi jumlah kendaraan yang tersedia sebanyak K . Pembatas (4) dan (5) memastikan bahwa jika kendaraan k pergi ke pelanggan i , maka kendaraan k akan meninggalkan pelanggan i menuju ke pelanggan j . Kemudian jika kendaraan k pergi dari pelanggan i menuju pelanggan j maka ada perhitungan biaya operasional dari pelanggan i ke pelanggan j , begitu pula jika kendaraan k pergi dari pelanggan j menuju pelanggan i . Pembatas (6) mengatakan bahwa kapasitas angkut kendaraan tidak boleh melebihi kapasitas kendaraan. Pembatas (7) merupakan batasan *sub-tour* yang membatasi setiap kendaraan yang mengunjungi suatu lokasi kecuali depot, sebagai contoh jika suatu kendaraan k mengunjungi dua titik lokasi maka jumlah perjalanan kendaraan k hanya ada 1, jika kendaraan k mengunjungi tiga titik lokasi maka jumlah perjalanan kendaraan k hanya ada 2. Pembatas (8), (9), dan (10) menyatakan bahwa variabel x , y , dan z merupakan variabel keputusan yang hanya bernilai 0 atau 1.

3.3 Verifikasi Model Matematis

Verifikasi dilakukan dengan melihat apakah model matematis yang telah dirancang telah konsisten antara satuan ruas kiri dan satuan ruas kanan dan hasil yang diperoleh telah tidak melanggar batasan.

Tabel 1. Hasil Verifikasi Model Matematis

Data	Rute	Batasan (2)	Batasan (3)	Batasan (4)	Batasan (5)	Batasan (6)	Batasan (7)
Hipotetik 1	1-2-7-6-5-1	TM	TM	TM	TM	TM	TM
	1-8-3-4-1	TM	TM	TM	TM	TM	TM
Hipotetik 2	1-7-3-10-5-8-1	TM	TM	TM	TM	TM	TM
	1-9-4-6-2-1	TM	TM	TM	TM	TM	TM

Tabel 2. Hasil Verifikasi Algoritma IH

Data	Rute	Batasan (2)	Batasan (3)	Batasan (4)	Batasan (5)	Batasan (6)	Batasan (7)
Hipotetik 1	1-4-6-2-1	TM	TM	TM	TM	TM	TM
	1-5-7-8-3-1	TM	TM	TM	TM	TM	TM
Hipotetik 2	1-5-4-2-1	TM	TM	TM	TM	TM	TM
	1-8-6-10-3-7-9-1	TM	TM	TM	TM	TM	TM

Tabel 3. Hasil Verifikasi Algoritma SA

Data	Rute	Batasan (2)	Batasan (3)	Batasan (4)	Batasan (5)	Batasan (6)	Batasan (7)
Hipotetik 1	1-2-7-6-5-1	TM	TM	TM	TM	TM	TM
	1-8-3-4-1	TM	TM	TM	TM	TM	TM
Hipotetik 2	1-7-3-10-5-8-1	TM	TM	TM	TM	TM	TM
	1-2-6-4-9-1	TM	TM	TM	TM	TM	TM

Keterangan:

TM : Tidak Melanggar

M : Melanggar

Berdasarkan tabel 1 dan tabel 4 dapat diketahui bahwa satuan ruas kiri dan ruas kanan dari model matematis adalah sama, kemudian solusi yang diperoleh menggunakan metode analitik tidak melanggar batasan-batasan yang telah ditentukan. Sehingga disimpulkan bahwa model matematis telah terverifikasi.

Tabel 4. Verifikasi satuan

Pembatas (persamaan/pertidaksamaan)	Satuan	
	Ruas Kiri	Ruas Kanan
(1)	Rp	$R_p \times \text{unitless} + R_p \times \text{unitless}$
(2)	Unitless	Unitless
(3)	Unitless	Unitless
(4)	Unitless	Unitless
(5)	Unitless	Unitless
(6)	$\text{Cm}^3 \times \text{unitless}$	$\text{Cm}^3 \times \text{unitless}$
(7)	Unitless	Unitless
(8)	Unitless	Unitless
(9)	Unitless	Unitless
(10)	Unitless	Unitless

3.4 Algoritma Insertion Heuristic

Gambar 2 adalah *pseudocode* dari algoritma IH yang telah dirancang.

```

1 Set rute awal depot-depot untuk setiap Rk
2 for i = 1:pelanggan
3   Sisipkan pelanggan
4   if Ka > Kk
5     Batasan terlanggar
6   else

```

```

7   Pilih rute dengan biaya terendah
8   endif
9   Cek kapasitas tersedia
10  if Td > Kt
11    Sisipkan pelanggan
12    Beri biaya penalty
13    Pilih rute dengan biaya terendah
14  endif
15  Tandai lokasi i
16 endfor

```

Gambar 2. *Pseudocode* Algoritma IH

Keterangan:

Rk : Rute kendaraan

Ka : Kapasitas angkut

Kk : Kapasitas kendaraan

Td : Total permintaan pelanggan belum terlayani

Kt : Kapasitas kendaraan yang tersedia

3.5 Verifikasi Algoritma Insertion Heuristic

Berdasarkan tabel 2 dapat diketahui bahwa solusi yang diperoleh dari algoritma IH tidak melanggar batasan. Sehingga dapat disimpulkan algoritma IH terverifikasi secara kelogisan algoritma. Hasil dari algoritma IH ini akan dijadikan sebagai solusi awal dalam penyelesaian menggunakan algoritma SA.

3.6 Algoritma Simulated Annealing

Gambar 3, 4 dan 5 adalah *pseudocode* dari algoritma SA yang telah dirancang.

```

1  $\sigma_0 = \text{insertion\_heuristic}$ 
2  $\sigma_{best} = \sigma_0$ 
3 Inisiasi nilai T_awal,  $\alpha$ , T_akhir
4 T = T_awal
5 for i = 1:it_max
6   r = random(1,3)
7   if r == 1
8     Relocation move
9   elseif r == 2
10    Route exchange move

```

```

11 elseif r == 3
12   Route interchanging move
13 endif
14 if  $\sigma_{new} < \sigma_0$ 
15    $\sigma_0 = \sigma_{new}$ 
16 else
17    $\Delta = \sigma_{new} - \sigma_0$ 
18    $p = e^{-\frac{\Delta}{T}}$ 
19   u = random uniform(0,1)
20   if u ≤ p
21      $\sigma_0 = \sigma_{new}$ 
22   endif
23 endif
24 if  $\sigma_0 < \sigma_{best}$ 
25    $\sigma_{best} = \sigma_0$ 
26 endif
27 T = T × α
28 if T < T_akhir
29   break for
30 endif
31 endfor
    
```

Gambar 3. Pseudocode Algoritma SA

```

1 a = random(1,2)
2 if a == 1
3   Node = random(pelanggan)
4   Rute = [a b c Node d a]
5   Rute_baru = [a Node b c d a]
6 elseif a == 2
7   Node = random(pelanggan)
8   Rute1 = [a b c Node d a]
9   Rute2 = [a e g a]
10  Rute1 = [a b c d a]
11  Rute2 = [a e Node g a]
12 endif
    
```

Gambar 4. Pseudocode Relocation Move

```

1 Node1 = random(pelanggan)
2 Node2 = random(pelanggan)
3 if rute_node1 = rute_node2
4   Rute = [a Node1 b c Node2 d a]
5   Rute_baru = [a Node2 b c Node1 d a]
6 else
7   Rute_node1 = [a b Node1 d a]
8   Rute_node2 = [a Node2 e f a]
9   Rute_node1 = [a b Node2 d a]
10  Rute_node2 = [a Node1 e f a]
11 endif
    
```

Gambar 5. Pseudocode Route Exchange Move

Parameter algoritma SA yang digunakan dalam penelitian ini dapat dilihat pada tabel 4.

Tabel 4. Parameter Algoritma SA

Parameter SA	Nilai
T awal	10.000
T akhir	0
Cooling Factor (α)	0,95
Iterasi Setiap Temperatur	200
Iterasi Maksimum	500

```

1 Node1 = random(pelanggan)
2 Node2 = random(pelanggan)
3 if Rute_node1 = Rute_node2
4   Rute = [a Node1 b c d Node2 a]
5   Rute_baru = [a Node2 d c b Node1 a]
6 else
7   Rute_node1 = [a b Node1 c a]
8   Rute_node2 = [a d e Node2 f g a]
9   Rutebaru_node1 = [a b Node2 f g a]
10  Rutebaru_node2 = [a d e Node1 c a]
    
```

```

11 endif
    
```

Gambar 6. Pseudocode Route Interchanging Move

Keterangan:

- p : Probabilitas menerima solusi buruk
- u : Bilangan random berdistribusi *uniform*
- T : Temperatur saat ini
- α : Nilai *cooling factor*
- Δ : Selisih antara σ_{new} dan σ_0
- σ_0 : Biaya operasional solusi awal
- σ_{new} : Biaya operasional solusi yang ditemukan
- σ_{best} : Biaya operasional solusi terbaik

3.7 Verifikasi Algoritma Simulated Annealing

Berdasarkan tabel 3 solusi dari algoritma SA tidak melanggar seluruh batasan yang ada. Sehingga algoritma SA dapat dikatakan terverifikasi.

3.8 Perbandingan Model Matematis dan Algoritma

Perbandingan solusi dari metode analitik, algoritma IH, dan algoritma SA dilakukan untuk melihat perbedaan solusi (gap) yang didapatkan dari ketiga metode tersebut. Selain itu, perbandingan ini juga dilakukan untuk melihat metode mana yang paling baik untuk digunakan dalam menyelesaikan permasalahan VRP. Suatu metode dikatakan bagus untuk digunakan jika solusi yang didapatkan merupakan solusi yang paling baik dan waktu komputasi yang dibutuhkan lebih singkat. Data uji coba yang digunakan dalam perbandingan ini sebanyak 8 jenis data yang mewakili permasalahan HFVRP, yaitu data hipotetik 1, hipotetik 2, hipotetik 3, hipotetik 4, hipotetik 5, hipotetik 6, hipotetik 7 dan hipotetik 8. Setiap data memiliki karakteristik kasus HFVRP yang berbeda. Dalam penyelesaian kasus HFVRP pencarian solusi akan dibatasi selama 8 jam, dikarenakan 8 jam merupakan waktu yang sangat lama untuk memperoleh solusi dari kasus HFVRP yang diteliti berdasarkan horison perencanaan dari perusahaan.

Tabel 6 menunjukkan hasil percobaan dari setiap metode untuk data uji coba. Secara rata-rata, Metode analitik dan algoritma SA menghasilkan kualitas solusi yang paling baik. Namun, pada data dengan karakteristik jumlah pelanggan dan atau jumlah jenis kendaraan yang meningkat, metode analitik menghasilkan solusi dengan komputasi yang lebih lama dari algoritma SA, bahkan pada data H7 dan H8 metode tidak menghasilkan solusi optimal selama delapan jam. Hal ini disebabkan metode ini mencari semua kemungkinan solusi yang ada, sehingga membutuhkan waktu yang cukup lama ketika ada penambahan indeks baik titik pelanggan, maupun tipe kendaraan.

Algoritma IH menghasilkan solusi yang sangat singkat pada setiap data uji coba berdasarkan tabel 6. Namun, solusi yang diperoleh dari algoritma IH sangat buruk dibandingkan solusi dari kedua metode lainnya yang ditunjukkan dengan nilai gap yang besar padat tabel 6. Hal ini dikarenakan, algoritma IH hanya

membentuk satu kandidat solusi saja yang akan diperbaiki oleh algoritma SA.

Algoritma SA menghasilkan solusi yang sangat baik dan konsisten dari segi kualitas solusi maupun waktu komputasi. Algoritma SA juga mampu

menghasilkan solusi dalam waktu komputasi yang lebih singkat pada data H3, H4, H5, dan H6. Algoritma ini menghasilkan solusi yang cukup baik karena bekerja dengan filosofi *hill climbing*.

Tabel 6. Perbandingan Solusi

Data	Metode						Gap MM-IH (%)	Gap MM-SA (%)	Gap SA-IH (%)
	Model Matematis (MM)		Algoritma IH		Algoritma SA				
	Solusi (Rp)	WK	Solusi (Rp)	WK	Solusi (Rp)	WK			
H 1	23.600*	1 detik	27.900	0,01 detik	23.600*	18,9 detik	18,2	0	18,2
H 2	35.772*	8 detik	43.696	0,01 detik	35.772*	19,6 detik	22,1	0	22,1
H 3	29.205*	5,2 menit	35.470	0,01 detik	29.205*	20,2 detik	21,4	0	21,4
H 4	26.180*	40,2 menit	40.830	0,01 detik	26.180*	21,1 detik	55,9	0	55,9
H 5	27.460*	5,3 jam	46.280	0,01 detik	27.460*	21,9 detik	68,5	0	68,5
H 6	48.470*	11,2 menit	75.120	0,01 detik	48.470*	22,6 detik	54,9	0	54,9
H 7	~	8 jam	46.900	0,01 detik	25.070*	23 detik	-	-	87,07
H 8	~	8 jam	63.620	0,01 detik	31.950*	24,1 detik	-	-	99,1

Keterangan:

WK : Waktu komputasi

* Solusi terbaik

Deskripsi kasus:

H 1 : 2 tipe kendaraan dengan jumlah 2 kendaraan dan 7 pelanggan

H 2 : 2 tipe kendaraan dengan jumlah 2 kendaraan dan 9 pelanggan

H 3 : 2 tipe kendaraan dengan jumlah 2 kendaraan dan 11 pelanggan

H 4 : 3 tipe kendaraan dengan jumlah 3 kendaraan dan 11 pelanggan

H 5 : 4 tipe kendaraan dengan jumlah 4 kendaraan dan 11 pelanggan

H 6 : 5 tipe kendaraan dengan jumlah 5 kendaraan dan 9 pelanggan

H 7 : 5 tipe kendaraan dengan jumlah 5 kendaraan dan 11 pelanggan

H 8 : 2 tipe kendaraan dengan jumlah 5 kendaraan dan 11 pelanggan

3.9 Implementasi Model Matematis dan Algoritma

Permasalahan perusahaan yang dikaji terdapat 20 pelanggan yang harus dilayani, kemudian proses distribusi dilakukan menggunakan 2 tipe jenis kendaraan yaitu tipe A berjumlah 4 unit dan tipe B berjumlah 1 unit sehingga jumlah kendaraan yang digunakan adalah 5 unit. Kasus HFVRP di perusahaan mirip dengan kasus pada data hipotetik 8, dimana data hipotetik 8 memiliki 11 pelanggan yang harus dilayani dan 2 tipe kendaraan dengan jumlah 5 kendaraan yang melakukan proses distribusi, sedangkan kasus sebenarnya memiliki 20 pelanggan yang harus dilayani.

Berdasarkan hasil yang disajikan di tabel 6 maka disimpulkan bahwa kasus di perusahaan tidak dapat diselesaikan dengan metode analitik atau penyelesaian menggunakan model matematis. Hal ini dikarenakan metode analitik tidak mampu mendapatkan solusi untuk data hipotetik 8 dengan waktu komputasi yang dibatasi selama 8 jam.

Hasil dari algoritma IH menunjukkan bahwa semua kendaraan akan digunakan untuk proses

distribusi dengan total biaya operasional sebesar Rp. 326.180,75,-. Solusi dari algoritma IH selanjutnya akan dijadikan sebagai solusi awal dalam algoritma SA. Sedangkan dalam penyelesaian menggunakan algoritma SA total biaya operasional yang digunakan turun dari Rp. 326.180,75,- menjadi Rp. 219.927,5. Algoritma SA menghasilkan solusi dengan menggunakan 3 unit kendaraan tipe A saja untuk melakukan proses distribusi ke seluruh pelanggan, sedangkan kendaraan tipe B tidak digunakan sama sekali. Algoritma SA juga mampu menyelesaikan kasus yang diteliti hanya dalam waktu selama 19,27 detik saja.

Untuk menentukan solusi optimal pada kasus yang diteliti maka dilakukan perbandingan hasil dari penyelesaian secara analitik, heuristik dan metaheuristik dalam menyelesaikan kasus HFVRP di perusahaan. Dikarenakan kasus yang diteliti tidak dapat diselesaikan secara analitik maka perbandingan hanya dilakukan pada penyelesaian metaheuristik dengan algoritma SA dan hasil penelitian sebelumnya oleh Saputra dkk (2017) yang menggunakan metode IH. Tabel 9 merupakan tabel perbandingan solusi dari kasus yang diteliti.

Tabel 7. Hasil Implementasi Algoritma IH

Tipe Kendaraan	Unit Kendaraan	Rute	Biaya (Rp)	Waktu Komputasi
A	1	1 - 17 - 14 - 4 - 1	70.711,00	0,01 detik
	2	1 - 18 - 8 - 19 - 1	67.796,25	
	3	1 - 15 - 11 - 9 - 1	65.503,75	
	4	1 - 13 - 10 - 20 - 21 - 1	62.228,75	

Tipe Kendaraan	Unit Kendaraan	Rute	Biaya (Rp)	Waktu Komputasi
B	5	1 - 16 - 5 - 3 - 7 - 12 - 6 - 2 - 1	59.941,00	
Total			326.180,75	

Tabel 8. Hasil Implementasi Algoritma SA

Tipe Kendaraan	Unit Kendaraan	Rute	Biaya (Rp)	Waktu Komputasi
A	1	Tidak digunakan	0,00	19,2 detik
	2	1 - 15 - 3 - 5 - 10 - 1	61.246,25	
	3	1 - 20 - 21 - 7 - 14 - 17 - 18 - 2 - 19 - 8 - 11 - 9 - 6 - 12 - 1	88.559,75	
	4	1 - 13 - 4 - 16 - 1	70.121,50	
B	5	Tidak digunakan	0,00	
Total			219.927,50	

Dalam penelitian Saputra dkk. (2017) penyelesaian kasus tidak membatasi waktu komputasi, sehingga tidak diketahui berapa lama waktu yang dibutuhkan untuk mendapatkan solusi tersebut. Algoritma SA mampu menghasilkan solusi yang lebih baik dibandingkan hasil penelitian tersebut dengan gap 20,18%, kemudian waktu komputasi yang dibutuhkan untuk mendapatkan solusi tersebut sangat cepat yaitu 19,27 detik. Sehingga disimpulkan solusi yang paling optimal adalah menggunakan algoritma SA, karena biaya operasional dari solusi yang didapatkan lebih sedikit dan waktu komputasi yang dibutuhkan sangat cepat yaitu 19,27 detik.

Tabel 9. Penentuan Solusi Optimal

Algoritma SA		Saputra dkk. (2017)		Gap (%)
Solusi (Rp)	WK	Solusi (Rp)	WK	
219.927,5	19,27 detik	264.311	-	20,18

3.10 Analisis Sensitivitas

Menurut Elhaddad (2012) menentukan nilai parameter dalam algoritma SA merupakan salah satu penentu terhadap hasil yang baik atau buruk. Sehingga dalam penelitian ini akan dilakukan analisis sensitivitas dengan melakukan beberapa kali percobaan untuk melihat pengaruh hasil yang didapatkan jika nilai parameter SA berubah.

a. Temperatur awal

Tabel 10 menunjukkan bahwa nilai temperatur awal yang rendah mendapatkan solusi yang lebih buruk dibanding nilai temperatur awal yang tinggi. Solusi yang didapatkan dari temperatur awal 5.000 dan temperatur awal 10.000 adalah sama yaitu sebesar Rp. 219.927,5,- dan waktu komputasi yang diperlukan juga sama. Dari uji sensitivitas dapat disimpulkan bahwa nilai temperatur awal dapat mempengaruhi kualitas solusi yang dihasilkan, tetapi tidak berpengaruh dengan lamanya waktu komputasi.

Tabel 10. Analisis Sensitivitas Temperatur Awal

Temperatur Awal	Solusi (Rp)	Waktu Komputasi (detik)
100	237.391	19,5
500	239.814,5	19,1
1.000	220.877,25	19,2
5.000	219.927,5	19,5
10.000	219.927,5	19,2

b. Iterasi setiap temperatur

Tabel 11 menunjukkan bahwa nilai iterasi setiap temperatur yang berbeda berpengaruh terhadap hasil solusi yang baik atau buruk walaupun selisih dari solusi tersebut sedikit. Hal ini dikarenakan jika nilai iterasi terlalu rendah maka algoritma SA hanya memiliki waktu yang sedikit untuk mendapatkan solusi yang baik. Kemudian semakin tinggi nilai iterasi setiap temperatur maka waktu komputasi juga semakin lama

Tabel 11. Analisis Sensitivitas Iterasi

Iterasi Setiap Temperatur	Solusi (Rp)	Waktu Komputasi (detik)
50	221.204,75	5
100	220.582,5	18,8
200	219.927,5	19,2
500	219.927,5	48,3
1000	219.927,5	99,8

c. Cooling factor

Tabel 12 menunjukkan bahwa adanya perubahan solusi yang didapatkan jika adanya perubahan nilai parameter *cooling factor*, walaupun perubahan solusi tersebut tidak terlalu signifikan. Kemudian waktu komputasi yang dibutuhkan juga tidak jauh berbeda. Selain itu, dari analisis sensitivitas *cooling factor* juga ditemukan bahwa semakin tinggi nilai *cooling rate* maka suhu dari algoritma SA akan menurun secara perlahan dan hasil yang dihasilkan dari setiap iterasi menurun secara perlahan hingga mencapai nilai minimumnya. Sedangkan nilai *cooling rate* yang rendah menunjukkan sebaliknya dimana suhu semakin cepat menurun dan hasil dari setiap iterasi menurun secara cepat hingga hasilnya tidak bisa menurun lagi, karena itu hasil yang diperoleh lebih buruk. Hal ini sesuai dengan proses *annealing*, dimana agar atom membentuk kristal yang kuat dibutuhkan pendinginan secara perlahan dan hati-hati.

Tabel 12. Analisis Sensitivitas *Cooling Factor*

<i>Cooling Factor</i>	Solusi (Rp)	Waktu Komputasi (detik)
0,8	220.549,75	19,04
0,85	220.549,75	19,1
0,9	219.927,5	19,4
0,95	219.927,5	19,2
0,99	219.927,5	19,3

4. KESIMPULAN DAN SARAN

Dalam penelitian ini penyelesaian kasus HFVRP dilakukan dengan tiga metode yaitu analitik, heuristik, dan metaheuristik. Penyelesaian dengan menggunakan algoritma SA menghasilkan solusi dengan biaya operasional sebesar Rp. 219.927,5,-. Penyelesaian menggunakan metode analitik menghasilkan kualitas solusi yang baik, namun dengan komputasi yang lama. Pada data H8 yang merepresentasi data pada objek yang dikaji, metode analitik tidak mampu menghasilkan solusi optimal dengan selama delapan jam waktu komputasi sehingga dibutuhkan metode alternatif yaitu heuristik dan metaheuristik.

Metode metaheuristik menggunakan algoritma SA merupakan metode yang menghasilkan solusi yang paling baik untuk menyelesaikan kasus baik dari segi kualitas solusi maupun waktu komputasi. Metode ini mampu meningkatkan solusi sebesar 20,18% dari penelitian sebelumnya. Penelitian selanjutnya dapat mempertimbangkan batasan lainnya seperti *time windows*, *multi depot* untuk HFVRP atau dapat menggunakan algoritma lain yang seperti *genetic algorithm*, *particle swarm optimization*, *cross entropy* untuk dibandingkan.

UCAPAN TERIMA KASIH

Penelitian ini didukung oleh Universitas Syiah Kuala, Kementerian Riset, Teknologi, dan Pendidikan Tinggi melalui Program Hibah Laboratorium 2019 dengan Surat Kontrak Hibah Laboratorium Tahun Anggaran 2019 Nomor: 305/UN11.2/PP/PNBP/SP3/2019. Terima kasih dan apresiasi yang tinggi untuk Rektor, Laboratorium Terpadu dan LPPM Universitas Syiah Kuala.

DAFTAR PUSTAKA

- AFIFI, S., DANG, D.-C. AND MOUKRIM, A. 2013. A Simulated Annealing Algorithm for the Vehicle Routing Problem with Time Windows and Synchronization Constraints. *Regional Council of Picardie and the European Regional Development Fund*, pp. 259–265.
- ANDRIANSYAH dan SENTIA, P. D., 2018. Penentuan Rute Kendaraan Pada Sistem Distribusi Logistik Pasca Bencana (Studi Kasus). *Jurnal Manajemen Industri dan Logistik*, 2(1), pp. 75–83.
- BIRIM, Ş., 2016. Vehicle Routing Problem with Cross Docking: A Simulated Annealing

Approach. *Procedia - Social and Behavioral Sciences*, 235(October), pp. 149–158.

- BORADE, A. B. dan BANSOD, S. V., 2007. Domain Of Supply Chain Management - A State Of Art. *Journal of Technology Management & Innovation*, 2(4), pp. 109–121.
- CAMPBELL, A. M. dan SAVELSBERGH, M., 2004. Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems. *Transportation Science*, 38(3), pp. 369–378.
- CARIC, T. AND GOLD, H., 2008. *Vehicle Routing Problem*. In-Teh.
- DELL'AMICO, M. *ET AL.*, 2007. Heuristic Approaches for the Fleet Size and Mix Vehicle Routing Problem with Time Windows. *Transportation Science*, 41(4), pp. 516–526.
- DULLAERT, W. *ET AL.*, 2002. New heuristics for the Fleet Size and Mix Vehicle Routing Problem with Time Windows. *Journal of the Operational Research Society*, 53(11), pp. 1232–1238.
- ELHADDAD, Y. R., 2012. Combined Simulated Annealing and Genetic Algorithm to Solve Optimization Problems. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 6(8), pp. 1047–1049.
- GENDREAU, M. dan POTVIN, J. Y., 2010. *Handbook of Metaheuristics*. 2nd edn. Springer.
- HOFF, A. *ET AL.*, 2010. Industrial aspects and literature survey: Fleet composition and routing. *Computers and Operations Research*. Elsevier, 37(12), pp. 2041–2061.
- IMRAN, A., SALHI, S. AND WASSAN, N. A., 2009. A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*. Elsevier B.V., 197(2), pp. 509–518.
- KOÇ, Ç. *ET AL.*, 2015. Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research*, pp. 1–21.
- LEUNG, S. C. H. *ET AL.*, 2013. A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints. *European Journal of Operational Research*. Elsevier B.V., 225(2), pp. 199–210.
- PUJAWAN, I. N. dan ER, M., 2017. *Supply Chain Management*. 3rd edn. ANDI Yogyakarta.
- SAPUTRA, N., SENTIA, P. D., ANDRIANSYAH, 2017. Penentuan Rute Kendaraan Heterogen Menggunakan Algoritma Insertion Heuristic. *Jurnal Optimasi Sistem Industri*.
- TALBI, E.-G., 2009. *Metaheuristics From Design To Implementatiton*. A John Wiley & Sons, Inc.
- WIBISONO, E., 2015. Pengembangan heuristik pada kasus heterogeneous vehicle routing

- problems with time windows and fixed costs. pp. 22–34.
- WIRDIANTO, E., JONRINALDI AND SURYA, B., 2007. Penerapan Algoritma Simulated Annealing pada Penjadwalan Distribusi Produk. *Jurnal Optimasi Sistem Industri*, 7(1), pp. 7–20.
- YU, V. F. *ET AL.*, 2017. A simulated annealing heuristic for the hybrid vehicle routing problem. *Applied Soft Computing Journal*. Elsevier B.V., 53, pp. 119–132.