

PERENCANAAN RUTE PENGIRIMAN MENGGUNAKAN METODE PARALLEL INSERTION DAN EXHAUSTIVE SEARCH PADA PT. STARMASS LOGISTICS

Irvan Maulana, Rifa Arifati

Fakultas Teknik UPN "Veteran" Jakarta Program Studi Teknik Industri

E-mail: rifahs@yahoo.com

Abstract

PT. Starmass Logistics is a company engaged in freight forwarding/ packages and documents. The company provides both domestic and international services. In the places of delivery to destination, PT. Starmass Logistics use based solely on the preferences and experience of courier only. Within delivery order record dated September 9, 2013, the company delivers 13.71 m3 and 1713.47 kg goods to 36 locations. With that large number of customers, companies need to optimize delivery route in order to save the transportation costs. Classical heuristics is an approach that is easy to use to find the Basic Feasible Solution to this Capacitated Vehicle Routing Problem. With saving and parallel insertion we got 453.8 km mileage routes. By doing Tour Improvement with Exhaustive Search, we can minimize mileage up to 360 km..

Keywords : *Capacitated Vehicle Routing Problem, Saving, Parallel Insertion, Exhaustive Search*

PENDAHULUAN

Perencanaan logistik dikatakan baik apabila barang atau jasa sampai pada tempat yang tepat, di waktu yang tepat dan dalam kondisi yang diinginkan pelanggan. Transportasi merupakan komponen yang vital dalam manajemen logistik suatu perusahaan. Salah satu faktor yang menentukan dalam manajemen logistik adalah penentuan jalur distribusi yang akan berpengaruh terhadap biaya transportasi. Pada umumnya biaya transportasi menyerap persentase biaya logistik yang lebih besar daripada aktivitas logistik lainnya. Ballou (2000), dalam Toth dan Vigo (2002), menyatakan bahwa sistem transportasi merupakan salah satu sistem yang banyak menyerap biaya (sekitar setengah sampai dua per tiga dari total biaya logistik). Permasalahan di bidang transportasi lebih dikenal dengan istilah Vehicle Routing Problem (VRP), berkaitan dengan penentuan rute-rute kendaraan yang perlu ditempuh oleh suatu kendaraan.

Oleh karena itu, untuk mengurangi biaya transportasi, diperlukan sistem transportasi yang efisien. Dengan menurunnya biaya transportasi, perusahaan dapat lebih mudah bersaing dengan para kompetitor dalam hal harga. Permasalahan yang bertujuan untuk membuat dan menentukan suatu rute yang optimal, untuk suatu kelompok kendaraan, agar dapat melayani sejumlah konsumen

disebut sebagai *Vehicle Routing Problem (VRP)*. *Vehicle Routing Problem* adalah sebuah cakupan masalah yang di dalamnya ada sebuah problem dimana ada sejumlah rute untuk sejumlah kendaraan yang berada pada satu atau lebih depot yang harus ditentukan jumlahnya agar tersebar secara geografis supaya bisa melayani konsumen-konsumen yang tersebar.

Di dalam penyelesaian VRP suatu perusahaan, metode-metode heuristics telah banyak dikembangkan, diantaranya adalah metode *heuristics* klasik (*classical heuristics*). Metode *heuristics* klasik banyak menjadi acuan bagi perkembangan metode heuristics lain. Metode ini mampu menghasilkan solusi yang cukup baik dengan waktu komputasi (waktu perhitungan) yang singkat. Banyak dari metode *heuristics* klasik yang dapat dengan mudah diadaptasikan dengan keberagaman kendala yang muncul dalam kondisi real. Untuk itu, penggunaan metode ini masih sering dan banyak digunakan. Penyelesaian permasalahan kendaraan dilakukan dalam dua tahap, yaitu: *tour construction* dan *tour improvement*.

Tour construction merupakan tahap penentuan metode untuk membuat solusi (rute perjalanan) awal. Metode *heuristics* klasik yang digunakan pada tahapan ini adalah *Clarke and Wright Savings Algorithm (metode savings)*. Algoritma metode ini sederhana (mudah

diimplementasikan) serta relevan dalam menjawab persoalan VRP karena menghasilkan solusi yang cukup baik dan masih banyak juga pihak yang menggunakan serta mengembangkannya secara berkelanjutan agar melalui metode ini dihasilkan solusi yang semakin baik.

Tour improvement merupakan tahapan untuk memperbaiki solusi awal, *Metode improvement* yang digunakan yaitu *Exhaustive Search*. *Exhaustive Search* adalah teknik pencarian solusi secara *brute force* pada masalah yang melibatkan pencarian elemen dengan sifat khusus, biasanya di antara objek-objek kombinatorik seperti permutasi, kombinasi, atau himpunan bagian dari sebuah himpunan. *Algoritma exhaustive search* memeriksa secara sistematis setiap kemungkinan solusi satu per satu dalam pencarian solusinya. Meskipun *algoritma exhaustive* secara teoritis menghasilkan solusi, namun waktu atau sumberdaya yang dibutuhkan dalam pencarian solusinya sangat besar. Di dalam beberapa literatur strategi algoritmik, contoh masalah yang sering diasosiasikan dengan *exhaustive search* atau *brute force* adalah masalah (TSP) *Travelling Salesperson Problem* (Munir, 2004). Pada tahap ini, urutan perjalanan (dalam satu rute) yang telah dihasilkan dari solusi awal akan mengalami perpindahan. Perpindahan yang dilakukan bertujuan untuk mencari solusi yang lebih baik dari solusi awal.

PT. STARMASS LOGISTICS merupakan perusahaan yang bergerak di bidang jasa pengiriman barang/ paket dan dokumen. Perusahaan ini menyediakan layanan baik untuk domestik maupun internasional. Dalam pengiriman ke tempat-tempat tujuan, PT. Starmass Logistics menggunakan rute yang hanya didasarkan pada preferensi dan pengalaman kurir saja. Perusahaan belum mempunyai prosedur penentuan rute yang optimal. Dengan banyaknya jumlah customer, perusahaan membutuhkan rute pengiriman yang optimal agar dapat menghemat biaya transportasi. Oleh karena itu, diperlukan suatu program ataupun metode khusus yang dapat memberikan solusi untuk menentukan rute optimal dalam setiap melakukan pengiriman.

Sampai saat ini, PT. Starmass Logistics masih melakukan upaya adaptasi dan peningkatan sistem-sistem yang ada, salah satunya adalah sistem transportasi atau *Transportation Management System* (TMS). Berdasarkan kondisi tersebut dan dengan melihat pentingnya peranan transportasi untuk menunjang aliran bisnis, maka fokus pada

penelitian kali ini adalah pada TMS PT. Starmass Logistics, yaitu pada permasalahan rute kendaraan (VRP) PT. Starmass Logistics.

Berdasarkan penjelasan tersebut maka pada penelitian kali tiap metode dalam *tour construction* akan dipadukan dengan tiap metode dalam *tour improvement*. Kemudian hasil yang diperoleh dari kombinasi metode-metode tersebut akan dibandingkan untuk akhirnya diketahui metode terbaik bagi permasalahan jalur/ rute kendaraan pada PT. STARMASS LOGISTICS, yaitu metode yang menghasilkan rute-rute dengan jarak tempuh terpendek.

TINJAUAN PUSTAKA

Vehicle Routing Problem

VRP atau *Vehicle Routing Problem* merupakan suatu permasalahan yang berfokus pada pendistribusian barang dari depot (gudang) perusahaan kepada pelanggannya. Pengiriman barang tersebut menyangkut pelayanan yang diberikan perusahaan dalam kurun waktu yang telah ditentukan kepada sejumlah pelanggan dengan menggunakan kendaraan tertentu dimana lokasi depot dapat berada pada satu atau lebih lokasi. Kendaraan dikemudikan oleh pengemudi melewati jalan yang memungkinkan untuk dilewati. Solusi dari VRP berupa rute-rute yang dapat ditempuh kendaraan untuk mengantarkan seluruh permintaan pelanggan dimana setiap rute ditempuh oleh satu kendaraan yang berawal dan berakhir di depot.

Karakteristik VRP

Menurut (Toth dan Vigo, 2002) ada beberapa karakteristik dalam VRP yang perlu diperhatikan. Yang pertama adalah komponen-komponen yang berkaitan dalam VRP, yaitu:

1. Pelanggan

- Lokasi pelanggan
- Permintaan pelanggan
- Rentang waktu dimana pelanggan boleh dilayani (*time windows*)
- Waktu yang dibutuhkan untuk bongkar-muat barang
- Perkiraan jenis kendaraan yang digunakan untuk melayani pelanggan (mempertimbangkan kondisi jalan yang ditempuh dan karakteristik permintaan yang dibawa)

2. Depot

- Lokasi depot
- Jam operasional depot
- Kendaraan yang ada di depot

3. Kendaraan

- Jumlah dan kapasitas angkut tiap kendaraan
- Kendaraan berawal dan berakhir di depot
- Setiap rute hanya dilayani oleh satu kendaraan
- Total permintaan pelanggan yang dibawa suatu kendaraan di rute tertentu (tidak boleh melebihi kapasitas angkut kendaraan tersebut)
- Biaya-biaya yang terkait dengan kendaraan (misalkan: biaya persatuan jarak atau per satuan waktu tempuh)

4. Pengemudi

Pengemudi yang dipekerjakan harus memenuhi keseluruhan syarat yang dimiliki perusahaan, seperti jam kerja harian, jam istirahat selama melakukan pelayanan kepada pelanggan, lama maksimum mengemudi, overtime, dan lain-lain.

5. Rute Kendaraan (jalan yang ditempuh)

Rute yang ditempuh perlu diperhatikan kondisi aslinya, apakah memungkinkan untuk dilewati atau tidak. Selain itu, perlu juga memperhatikan kondisi barang yang dibawa apakah memungkinkan untuk melewati rute tersebut. Hal ini bertujuan untuk mencegah waktu perjalanan yang lama dan juga kerusakan yang mungkin timbul akibat kondisi jalan yang ditempuh.

Karakteristik berikutnya dari VRP adalah dalam hal kendala yang ada dalam VRP tersebut. Berdasarkan batasan atau kendala yang ada, VRP dalam penelitian ini adalah:

CVRP (*Capacitated Vehicle Routing Problem*) yang merupakan model dasar dalam VRP dengan kapasitas angkut kendaraan sebagai kendala yang dihadapi. Semua permintaan pelanggan diketahui di awal dan pengantaran permintaan tersebut, untuk setiap pelanggan, dilakukan pada satu rute yang sama (keseluruhan permintaan suatu pelanggan diletakkan pada rute yang sama). Kendaraan yang digunakan adalah identik dan hanya terdapat satu depot sebagai lokasi awal dan akhir setiap kendaraan.

Model Dasar VRP

Menurut (Toth dan Vigo, 2002) berikut ini merupakan model matematis dasar dari VRP yaitu memiliki kendala pada kapasitas angkut kendaraan. Apabila terdapat kendala lain, maka model ini bisa dikembangkan sesuai kebutuhan:

= Node (titik) depot dan pelanggan, $N = (0, 1, 2, \dots, n)$
 $N = 0$ Node gudang

$N =$ Node pelanggan

$K =$ Himpunan kendaraan,

$K = (0, 1, 2, \dots, k)$

$V_{zk} =$ Kapasitas maksimum kendaraan k

$d_j =$ Total permintaan pelanggan j

$c_{ijk} =$ Biaya/jarak untuk menempuh lokasi pelanggan i ke pelanggan j menggunakan kendaraan k

Fungsi tujuan adalah meminimumkan biaya pengiriman total:

$$\text{Min. } Z = \sum_j \sum_k c_{ijk} x_{ijk} \quad (2.1)$$

Satu titik hanya dikunjungi oleh satu kendaraan :

$$\sum_i \sum_k x_{ijk} = 1, \forall j \quad (2.2)$$

Hanya satu kendaraan yang keluar dari satu titik :

$$\sum_i \sum_k x_{ijk} = 1, \forall i \quad (2.3)$$

Terdapat K kendaraan yang keluar dari depot/gudang:

$$\sum_i \sum_k x_{oijk} = K \quad (2.4)$$

Terdapat K kendaraan yang masuk ke depot/gudang:

$$\sum_i \sum_k x_{oijk} = K \quad (2.5)$$

Total barang yang diangkut oleh satu kendaraan tidak melebihi kapasitas angkutnya:

$$\sum_i \sum_k a_{ij} x_{ijk} \leq V_{zk}, \forall k \quad (2.6)$$

$$x_{ijk} = \begin{cases} 1 & (\text{Lintasan jalan } i, j \text{ dilalui oleh kendaraan } k) \\ 0 & (\text{Lintasan } i, j \text{ tidak dilalui oleh kendaraan } k) \end{cases}$$

Metode Heuristics Klasik untuk penyelesaian VRP

Heuristics merupakan suatu pendekatan atau uji coba yang dilakukan untuk memecahkan suatu permasalahan. Permasalahan yang dicoba diselesaikan dengan metode ini merupakan permasalahan yang tidak mempunyai solusi yang pasti. Permasalahan tersebut dicoba untuk dicarikan solusi yang sedapat mungkin optimal. Dengan demikian, metode pendekatan ini akan senantiasa berkembang guna mencari solusi yang lebih baik dari sebelumnya.

Salah satu kelompok dalam heuristics adalah heuristics klasik (*classical heuristics*). Heuristics klasik berkembang pada umumnya di tahun 1960-1990. Hampir sebagian besar konstruksi standard (konstruksi awal) dan perbaikan suatu rute terdapat pada heuristics kelompok ini. Metode *heuristics* klasik melakukan pencarian yang sederhana (terbatas) dari permasalahan yang terjadi namun mampu menghasilkan solusi yang cukup baik

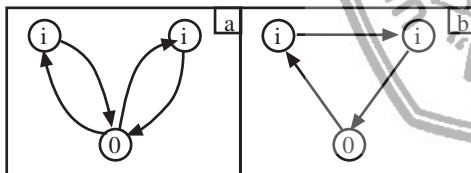
dengan waktu komputasi yang singkat. Banyak dari metode ini dapat dikembangkan lebih lanjut sesuai dengan kendala-kendala yang datang dalam praktek nyatanya. Karena itu, *heuristics* klasik masih banyak digunakan sekalipun pada saat yang bersamaan telah berkembang metode-metode lain. *Heuristics* klasik dalam penelitian kali ini, seperti yang dijelaskan pada bagian pendahuluan, tahapan pembuatan rute terbagi ke dalam dua tahap : *Tour Construction* dan *Tour Improvement*.

Tour Construction

Pada tahap ini, dilakukan upaya untuk menciptakan solusi awal dari permasalahan rute kendaraan. Dalam menciptakan solusi tersebut, metode yang digunakan berasal dari golongan *construction method* adalah *Clarke and Wright Savings Algorithm*.

Algoritma ini tergolong dalam *construction method*, yaitu metode yang secara berangsur-angsur (bertahap) memasukkan setiap pelanggannya ke dalam suatu rute. Metode ini, sesuai namanya, di publikasikan oleh *Clarke* dan *Wright* dengan berdasarkan pada prinsip penghematan (*savings*). Penghematan yang dimaksud adalah penghematan yang diperoleh apabila menggabungkan dua rute menjadi satu. Dua rute yang memiliki penghematan terbesar yang pertama kali mendapat kesempatan untuk dimasukkan ke dalam rute.

Gambar 2.1 Ilustrasi Konsep *Savings*



(sumber: Toth dan Vigo, 2002)

Keterangan:

i, j = pelanggan i , pelanggan j
 0 = depot

Gambar 2.1 (a) menunjukkan bahwa pelanggan i dan pelanggan j dilewati (dikunjungi) oleh rute yang berbeda. Pada Gambar 2.1 (b), pelanggan i dan pelanggan j berada pada rute yang sama. Dengan mengacu pada ilustrasi tersebut, maka bisa dicari penghematan yang bisa diperoleh dari (a) menuju (b). Penghematan yang dimaksud berupa penghematan biaya atau jarak (tergantung data yang diketahui).

$$D_a = C_{0i} + C_{i0} + C_{0j} + C_{j0}$$

$$D_b = C_{0i} + C_{ij} + C_{j0}$$

$$S_{ij} = D_a - D_b$$

$$= (C_{0i} + C_{i0} + C_{0j} + C_{j0}) - C_{0i} - C_{ij}$$

$$= C_{i0} + C_{0j} - C_{ij}$$

Keterangan:

D_a = biaya atau jarak total dua rute (rute pelanggan I ditambah dengan rute pelanggan j)
 D_b = biaya atau jarak total satu rute (rute pelanggan i dan j)

c_{oi} = biaya atau jarak dari depot ke pelanggan i
 c_{io} = biaya atau jarak dari pelanggan i ke depot
 c_{oj} = biaya atau jarak dari depot ke pelanggan j
 C_{jo} = biaya atau jarak dari pelanggan j ke depot
 c_{ij} = biaya atau jarak dari pelanggan i ke pelanggan j
 S_{ij} = penghematan biaya atau jarak antara pelanggan i dan j

Tiap dua rute dicari terlebih dahulu nilai *savings*-nya lalu dibuat *savings list* (suatu daftar yang berisi nilai *savings* dari yang terbesar hingga terkecil). Pelanggan dengan nilai *savings* terbesar akan diprioritaskan untuk masuk terlebih dahulu ke dalam rute.

Ada dua versi dalam proses memasukkan pelanggan ke dalam rute :

Sequential version

Versi ini membentuk rute demi rute secara bertahap. Semua nilai *savings* yang terdapat pada *savings list* akan dibaca dan pelanggan-pelanggan yang terkait dengan nilai tersebut dicoba untuk dimasukkan ke dalam rute. Apabila *feasible* (sesuai dengan kendala yang ada), maka pelanggan-pelanggan tersebut akan masuk ke dalam rute. Apabila tidak memungkinkan lagi untuk memasukkan pelanggan pada rute tersebut, maka rute tersebut dianggap selesai dibangun kemudian mulai membentuk rute yang baru dengan cara yang sama (mengulang pembacaan nilai *savings* lalu mencoba memasukkan pelanggan-pelanggan yang belum masuk ke dalam rute). Prioritas pembentukan rute kendaraan terkonsentrasi pada pembentukan satu rute terlebih dahulu berdasarkan nilai *savings* terbesar.

Parallel version

Pada versi ini, rute yang terbentuk bisa lebih dari satu pada setiap kali pembacaan nilai *savings*. Prioritas pembentukan kendaraan sangat

terkonsentrasi pada nilai savings terbesar. (Lysgaard, 1997) menampilkan cara menghitung solusi versi sequential dan parallel yang mudah dipahami. Keduanya menghasilkan solusi yang berbeda.

Tabel 2.1 Jarak dan Permintaan Pelanggan

Dari	Tujuan						Permintaan
	0	1	2	3	4	5	
0 (Depot)	-	28	31	20	25	34	0
1		-	21	29	26	20	37
2			-	38	20	32	35
3				-	30	27	30
4					-	25	25
5						-	32

(sumber: Toth dan Vigo, 2002)

Tabel 2.2 Nilai Savings Pelanggan

Saving	Pelanggan
42	1-5
38	1-2
36	2-4
34	4-5
33	2-5
27	1-4
27	3-5
19	1-3
15	3-4
13	2-3

(sumber: Toth dan Vigo, 2002)

Pada Tabel 2.1 terdapat informasi mengenai sebuah depot dan lima pelanggan dengan diketahui jumlah permintaan tiap pelanggan dan besarnya jarak antar titik. Berdasarkan rumus *savings*, diperoleh nilai savings dengan urutan dari yang terbesar hingga terkecil seperti yang terdapat pada Tabel 2.2. Kendaraan yang tersedia diasumsikan berkapasitas 100 unit. Jika menggunakan *sequential version*, dihasilkan solusi jarak total 187.

Rute I : 0-1-5-4-0 (jarak = 98)

Rute II : 0-2-3-0 (jarak = 89)

Jika menggunakan *parallel version*, solusi yang dihasilkan berjarak total 171:

Rute I : 0-1-5-3-0 (jarak = 95)

Rute II : 0-2-4-0 (jarak = 76)

Gambar 2.2 Perbandingan *Sequential* dan *Parallel Version*

Problem	Sequential	Parallel
E051-05e	625.56	584.64
E076-10e	1005.25	900.26
E101-08e	982.48	886.83
E101-10c	939.99	833.51
E121-07c	1291.33	1071.07
E151-12c	1299.39	1133.43
E200-17c	1708.00	1395.74
D051-06c	670.01	618.40
D076-11c	989.42	975.46
D101-09c	1054.70	973.94
D101-11c	952.53	875.75
D121-11c	1646.60	1596.72
D151-14c	1383.87	1287.64
D200-18c	1671.29	1538.66

(sumber: Toth dan Vigo, 2002)

Gambar 2.2 merupakan sebuah tabel yang diberikan

Toth dan Vigo yang menunjukkan bahwa hasil perhitungan *parallel version* lebih baik daripada *sequential version*.

Dalam pengolahan data selanjutnya, metode *savings* yang akan digunakan adalah *parallel version* karena terbukti menghasilkan solusi yang lebih baik dari *sequential version*.

Tour Improvement

Pada tahap ini, solusi awal yang telah dibuat pada *tour construction* mengalami perbaikan dengan menggunakan metode yang tergolong dalam *improvement method*. Perbaikan terhadap solusi tersebut dilakukan salah satunya dengan menukarkan letak node (titik) yang berada dalam satu solusi (*single route improvement*). Letak atau urutan node berpindah-pindah sedemikian rupa menurut aturan tertentu. Selama perpindahan terjadi, apabila solusi yang lebih baik ditemukan, maka solusi yang terdahulu akan digantikan dengan solusi yang baru.

Exhaustive Search adalah teknik pencarian solusi secara *Brute force* pada masalah yang melibatkan pencarian elemen dengan sifat khusus, biasanya di antara objek-objek kombinatorik seperti permutasi, kombinasi, atau himpunan bagian dari sebuah himpunan. Berdasarkan definisi ini, maka *Exhaustive Search* adalah *brute force* juga (Munir, 2004).

Metode *Exhaustive Search* dapat dirumuskan langkah-langkahnya sebagai berikut:

1. Enumerasi (list) setiap solusi yang mungkin dengan cara yang sistematis.
2. Evaluasi setiap kemungkinan solusi satu per satu, mungkin saja beberapa kemungkinan solusi

yang tidak layak dikeluarkan, dan simpan solusi terbaik yang ditemukan sampai sejauh ini (*the best solusi found so far*).

3. Bila pencarian berakhir, umumkan solusi terbaik (*the winner*).

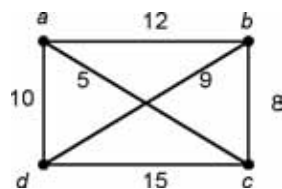
Algoritma Exhaustive Search memeriksa secara sistematis setiap kemungkinan solusi satu per satu dalam pencarian solusinya. Meskipun algoritma *Exhaustive Search* secara teoritis menghasilkan solusi, namun waktu atau sumber daya yang dibutuhkan dalam pencarian solusinya sangat besar. Di dalam beberapa literatur strategi algoritmik, contoh masalah yang sering diasosiasikan dengan *Exhaustive Search* atau *Brute Force* adalah masalah *Travelling Salesperson Problem* (TSP). Untuk mengingat kembali masalah *Travelling Salesperson Problem* (TSP) ini, berikut diulang kembali deskripsi masalahnya. TSP: diberikan buah kota serta diketahui jarak antara setiap kota satu sama lain. Temukan perjalanan (tour) terpendek yang melalui setiap kota lainnya hanya sekali dan kembali lagi ke kota asal keberangkatan. Jika setiap kota direpresentasikan dengan simpul dan jalan yang menghubungkan antar kota sebagai sisi, maka persoalan TSP ini dimodelkan dengan graf lengkap dengan n buah simpul.

Bobot pada setiap sisi menyatakan jarak antar setiap kota. Persoalan TSP tidak lain adalah menemukan sirkuit Hamilton dengan bobot minimum.

Algoritma Exhaustive Search untuk persoalan TSP ini adalah:

1. Enumerasikan (*list*) semua sirkuit Hamilton dari graf lengkap dengan n buah simpul.
2. Hitung (evaluasi) bobot setiap sirkuit Hamilton yang ditemukan pada langkah 1 (satu).
3. Pilih sirkuit Hamilton yang mempunyai bobot terkecil.

Contoh: $n = 4$



Misalkan simpul a adalah kota tempat dimulainya perjalanan (starting city).

Enumerasikan semua sirkuit Hamilton sebagai berikut:

Tabel 2.3 Enumerasi Sirkuit Hamilton

Rute perjalanan (tour)	Bobot
$a \rightarrow b \rightarrow c \rightarrow d \rightarrow a$	$12+9+15+10 = 45$
$a \rightarrow b \rightarrow d \rightarrow c \rightarrow a$	$12+9+15+5 = 41$
$a \rightarrow c \rightarrow b \rightarrow d \rightarrow a$	$5+8+9+10 = 32$
$a \rightarrow c \rightarrow d \rightarrow b \rightarrow a$	$5+15+9+12 = 41$
$a \rightarrow d \rightarrow b \rightarrow c \rightarrow a$	$10+9+8+5 = 32$
$a \rightarrow d \rightarrow c \rightarrow b \rightarrow a$	$10+15+8+12 = 45$

(sumber: Munir, 2004)

Untuk 4 kota, terdapat 6 buah kemungkinan rute perjalanan (atau sirkuit Hamilton).

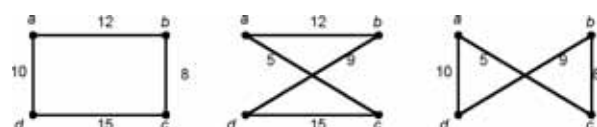
Rute perjalananan terpendek adalah $a \rightarrow c \rightarrow b \rightarrow d \rightarrow a$ atau $a \rightarrow d \rightarrow b \rightarrow c \rightarrow a$ dengan bobot = 32. Karena perjalanan berawal dan berakhir pada simpul yang sama, maka untuk n buah simpul semua rute perjalanan yang mungkin dibangkitkan dengan permutasi dari $n - 1$ buah simpul. Permutasi dari $n - 1$ buah simpul adalah $(n - 1)!$. Pada contoh di atas, untuk $n = 4$ akan terdapat $(4 - 1)! = 3! = 6$ buah rute perjalanan.

Jika persoalan TSP diselesaikan dengan metode *Exhaustive Search*, maka kita harus mengenumerasi sebanyak $(n-1)!$ buah sirkuit Hamilton, menghitung setiap bobotnya, dan memilih sirkuit Hamilton dengan bobot terkecil. Kemudian untuk menghitung bobot setiap sirkuit Hamilton dibutuhkan waktu $O(n)$, maka kompleksitas waktu algoritma *exhaustive search* untuk persoalan TSP sebanding dengan $(n-1)!$ dikali dengan waktu untuk menghitung bobot setiap sirkuit Hamilton. Dengan kata lain, kompleksitas waktu *algoritma exhaustive search* untuk persoalan TSP adalah $O(n \cdot n!)$.

Kita dapat menghemat jumlah komputasi dengan mengamati bahwa setengah dari rute perjalanan adalah hasil pencerminan dari setengah rute yang lain, yakni dengan mengubah arah rute perjalanan : (1 dan 6), (2 dan 4), (3 dan 5), maka dapat dihilangkan setengah dari jumlah permutasi (dari 6 menjadi 3).

Ketiga buah sirkuit Hamilton yang dihasilkan adalah seperti gambar di bawah ini:

Gambar 2.3 Sirkuit Hamilton



(sumber: Munir, 2004)

Dengan demikian, untuk graf dengan n buah simpul, kita hanya perlu mengevaluasi sirkuit Hamilton sebanyak $(n - 1)!/2$. Jelaslah bahwa kebutuhan waktu *algoritma Exhaustive Search* adalah dalam orde ekponensial. Algoritma ini hanya bagus untuk ukuran masukan (n) yang kecil sebab kebutuhan waktunya masih realistis. Untuk ukuran masukan yang besar, *algoritma Exhaustive Search* menjadi sangat tidak mangkus. Pada persoalan TSP misalnya, untuk jumlah simpul $n = 20$ akan terdapat $(19!)/2 = 6 \times 10^{16}$ sirkuit Hamilton yang harus dievaluasi satu per satu. Sayangnya, untuk persoalan *Travelling Salesperson Problem* (TSP) tidak ada algoritma lain yang lebih baik daripada *algoritma exhaustive search*. Algoritma yang mangkus selalu mempunyai kompleksitas waktu dalam orde polynomial (Munir, 2004). *Exhaustive Search* sering disebut-sebut di dalam bidang kriptografi, yaitu sebagai teknik yang digunakan penyerang untuk menemukan kunci enkripsi dengan cara mencoba semua kemungkinan kunci. Serangan semacam ini dikenal dengan nama *Exhaustive Search Attack* atau *Brute Force Attack*. Misalnya pada algoritma kriptografi DES (*Data Encryption Standard*), panjang kunci enkripsi adalah 64 bit (atau setara dengan 8 karakter). Dari 64 bit tersebut, hanya 56 bit yang digunakan (8 bit paritas lainnya tidak dipakai). Karena ada 56 posisi pengisian bit yang masing-masing memiliki dua kemungkinan nilai, 0 atau 1, maka jumlah kombinasi kunci yang harus dievaluasi oleh pihak lawan adalah sebanyak: $(2)(2)(2)(2)(2) \dots (2)(2)$ (sebanyak 56 kali) $= 256 = 7.205.759.403.7927.936$ buah.

Meskipun *algoritma Exhaustive Search* tidak mangkus (masih belum efisien), namun nilai plusnya terletak pada keberhasilannya yang selalu menemukan solusi jika diberikan waktu yang cukup (Munir, 2004).

METODE PENELITIAN

Penelitian ini bertujuan untuk mencari metode terbaik untuk sistem transportasi PT. Starmass Logistics. Dimana melaluinya akan dihasilkan rute kendaraan yang memiliki jarak tempuh terpendek. Metode yang digunakan adalah metode heuristics klasik, dimana metode ini memang digunakan untuk membangun solusi/rute awal (*Tour Construction*) dari permasalahan yang ada, kemudian menggabungkannya dengan *Metode Exhaustive Search* untuk membuat perbaikan rute awal (*Tour Improvement*).

Data berasal dari PT. Starmass Logistics yang merupakan data permintaan/pengiriman harian, karena data tersebut cukup untuk mewakili kondisi permintaan pelanggan yang bersifat harian. Data sekunder yang didapat dari perusahaan disusun terlebih dahulu sebelum diolah pada *Microsoft Office Excel*. Pelanggan-pelanggan yang alamatnya tidak terdapat dalam data sekunder dieliminasi (tidak diikutsertakan dalam pengolahan data) kemudian lokasi pelanggan-pelanggan tersebut diperkirakan pada peta dan dibuat matriks jaraknya dengan menggunakan *Google Maps*. Memasukkan matriks jarak dari masing-masing lokasi pengiriman/permintaan (*demand*), serta volume untuk mengoptimalkan kapasitas angkut kendaraan yang digunakan kedalam *Microsoft Office Excel*, kemudian mengolahnya berdasarkan algoritma dari masing-masing metode. Setelah didapatkan hasil maka dilakukan perbandingan terhadap hasil dari masing-masing metode tersebut.

PEMBAHASAN

Berdasarkan catatan data pengiriman (*Delivery Order*) pada tanggal 9 September 2013, terdapat 36 *Delivery Order* yang harus diantarkan oleh ke empat kendaraan perusahaan pada hari tersebut. Dengan jumlah 36 lokasi lokasi yang harus dikunjungi dengan total volume permintaan sebesar 13,71 m³ dan tonase seberat 1713.47 kg. Data pengiriman (*Delivery Order*) dapat dilihat pada tabel berikut :

Tabel 4.1. Permintaan Pelanggan

No	Nama	Alamat	Volume/m ³	Tonase/kg
0	PT. STARMASS LOGISTICS (Depot)	J. Jati, Jakarta Selatan	0.000000	46.85
1	PT. PROMEXX PUSAT STATIONARY	J. Gajah Mada	0.043943	22.56
2	SEJA KAWAN TE	Pasar Jemberga	0.092016	22.38
3	PT. TIKO WARNA GEMITA	J. Pluit Kembara, Pluit	0.217183	13.98
4	PT. TOKO GUNUNG AGUNG	Plaza Blok M	0.028953	56.29
5	PT. TOKO GUNUNG AGUNG	Semayan City	0.004820	57.29
6	PT. LOFFE MART INDONESIA	Kali Plaza	0.100054	48.94
7	PT. BANK MULIHO INDONESIA	Plaza BI	0.978454	36.02
8	PT. ELITE PRO MEDIA	Plaza Summarecon	2.017300	36.21
9	PT. STARMEDIA	J. Gading Elok Utara	0.770312	42.28
10	MERDEKA	Harjo Mangga Dua	0.008216	52.03
11	PT. CARREFOUR INDONESIA	J. TB Sempayan	0.074278	29.57
12	PT. PROMEXX PUSAT STATIONARY	J. I.R.H. Juanda, Jak-Pus	0.057798	25.74
13	PT. DATASCRIPT	J. Agung Karya, Tanjung Priuk	0.008845	71.24
14	PT. TISA DIAMANTA	ITC Mangga Dua	0.103347	17.88
15	TOKO PENAMAS	J. Pluit Selatan, Pluit	0.214480	72.87
16	CV. JAYA AGUNG BERSAMA	J. Mulyata Raya, Jak-Tim	0.028676	31.38
17	PT. TOKO GUNUNG AGUNG	J. Pulo Kambing, Jak-Tim	0.006894	26.75
18	BERKAH STATIONARY	J. Pulo Asri, Jak-Tim	0.004429	45.25
19	PT. PROMEXX PUSAT STATIONARY	J. KH. Husein Ashari	0.043804	43.37
20	KD APUNG	Pasar Pagi Mangga Dua	2.830000	52.06
21	PT. MITRA WIDHI SHAKTI UTAMA	J. Mangga Dua Raya	0.041380	43.49
22	FRANSISKA	Pasar Pagi Mangga Dua	0.943000	23.74
23	PT. PROMEXX PUSAT STATIONARY	Mall Emporium Pluit	0.028438	51.49
24	PT. PROMEXX PUSAT STATIONARY	Pluit Village	1.034629	39.43
25	PT. INDO STATIONERY RITEL UTAMA	Plaza Bintaro	0.010968	56.83
26	TOKO PENAMAS	Pasar Pagi Mangga Dua	0.030744	80.86
27	PT. TOKO GUNUNG AGUNG	Plaza Ardon	0.026737	38.03

(Sumber : PT. Starmass Logistics)

Tabel 4.3. Sampel *Savings List*

Pelanggan		Jarak (km)			Savings (km)
i	j	C_{ij}	C_{i0}	C_{0j}	S_{ij}
14	21	1.6	39.1	38	75.5
20	22	0.5	37.3	37.8	74.6
10	21	1.1	37.1	38	74
14	20	2.6	39.1	37.3	73.8
14	22	3.1	39.1	37.8	73.8
10	14	2.4	37.1	39.1	73.8
21	22	2.2	38	37.8	73.6
20	21	2	37.3	38	73.3
10	20	2.1	37.1	37.3	72.3
10	22	2.6	37.1	37.8	72.3
14	29	3.4	39.1	32.7	68.4
21	29	2.9	38	32.7	67.8
10	29	2.8	37.1	32.7	67
22	29	3.6	37.8	32.7	66.9
20	29	3.7	37.3	32.7	66.3
9	32	8.5	36.2	34.5	62.2
14	24	5.2	39.1	27.5	61.4
8	16	8.7	40.5	29.5	61.3
21	24	4.4	38	27.5	61.1
10	24	4.4	37.1	27.5	60.7
9	21	14.1	36.2	38	60.1
22	24	5.2	37.8	27.5	60.1
3	14	5.9	26.7	39.1	59.9
9	10	13.6	36.2	37.1	59.7
9	14	15.6	36.2	39.1	59.7

lagi untuk ditambahkan node (titik) berikutnya. Sedangkan pada *Parallel Insertion*, prioritas pembentukan rute kendaraan dibentuk secara berurutan dan sangat terkonsentrasi pada nilai savings terbesar, dan dengan jarak tempuh yang paling terdekat dengan node (titik) sebelumnya. Contoh dari beberapa tahap pembuatan rute *Parallel Insertion*:

Keterangan :

1. T1 = Truk 1, T2 = Truk 2, T3 = Truk 3, T4 = Truk 4.
2. Pada tahap ke 2 terdapat pertukaran node pada T2 (truk 2) karena dari node 22 ke 20 menghasilkan sirkuit (rute) yang lebih baik dibandingkan jika dimulai dari node 20 ke 22.
3. Pada tahap ke 4 juga terdapat pertukaran node pada T4 (truk 4) karena dari node 32 ke 9 menghasilkan sirkuit (rute) yang lebih baik dibandingkan jika dimulai dari node 9 ke 32

Tabel 4.4. Perbandingan *Sequential* dan *Parallel Insertion*

4.2 Pembuatan rute awal (*Sequential*) dan (*Parallel Insertion*)

Pada pembuatan *Sequential*, semua nilai savings yang terdapat pada *savings list* akan dibaca dan pelanggan-pelanggan yang terkait dengan nilai tersebut dicoba untuk dimasukkan ke dalam rute. Apabila *feasible* (sesuai dengan kendala yang ada), maka pelanggan-pelanggan tersebut akan masuk ke dalam rute. Apabila tidak memungkinkan lagi untuk memasukkan pelanggan pada rute tersebut, maka rute tersebut dianggap selesai dibangun kemudian mulai membentuk rute yang baru dengan cara yang sama (mengulang pembacaan nilai savings lalu mencoba memasukkan pelanggan-pelanggan yang belum masuk ke dalam rute). Prioritas pembentukan rute kendaraan terkonsentrasi pada pembentukan satu rute terlebih dahulu berdasarkan nilai savings terbesar.

Contoh pembuatan rute *Sequential* pada Truk 1:
Rute : $\rightarrow 0 \rightarrow 21 \rightarrow 14 \rightarrow 20 \rightarrow 29 \rightarrow 10 \rightarrow 0$
Pembuatan rute berhenti pada node 10, karena kapasitas kendaraan sudah tidak memungkinkan

Tabel 4.4. Menunjukkan perbandingan antara *Sequential* dan *Parallel Insertion*, dan hasilnya dapat terlihat perbandingan rute yang lebih baik. Maka dipilihlah *Parallel Insertion* sebagai rute awal, selanjutnya dilakukan kembali perbaikan rute awal (*Tour Construction*) dengan melakukan *Exhaustive Search* dan *Enumerasi* untuk menghasilkan perbaikan rute (*Tour Improvement*) yang terbaik.

4.3 Perbaikan rute *Parallel Insertion* (*Tour Construction*) dengan *Exhaustive Search* dan *Enumerasi* (*Tour Improvement*)

Pada tahap ini, solusi awal yang telah dibuat

pada *tour construction* mengalami perbaikan dengan menggunakan metode yang tergolong dalam *improvement method*. Perbaikan terhadap solusi tersebut dilakukan salah satunya dengan menukarkan letak node (titik) yang berada dalam satu solusi (*single route improvement*). Letak atau urutan node berpindah-pindah sedemikian rupa menurut aturan tertentu. Selama perpindahan terjadi, apabila solusi yang lebih baik ditemukan, maka solusi yang terdahulu akan digantikan dengan solusi yang baru. Pencarian rute terbaik dilakukan dengan bantuan *Exhaustive search* pada *Ms. Excel*.

Tabel 4.6. Sampel dari proses *Exhaustive Search* dan Enumerasi (Perbaikan rute *Parallel Insertion*)

Tabel 4.6. merupakan sampel dari proses *Exhaustive Search* dan Enumerasi (*Tour Improvement*) dalam melakukan perbaikan rute *Parallel Insertion* (*Tour Construction*)

4.4 Hasil dari proses Exhaustive Search dan Enumerasi

Berikut ini adalah hasil dari perbaikan rute *Parallel Insertion* (*Tour Construction*) yang akan digantikan dengan solusi yang baru dan lebih baik dari sebelumnya (*Tour Improvement*).

Tabel 4.7. Hasil Exhaustive Search dan Enumerasi

Improved Route										Distance (km)		
1st Truck	0	18	27	13	21	14	24	3	7	5	0	81.7
2nd Truck	0	36	19	22	20	15	33	34	25	4	0	105.4
3rd Truck	0	28	10	29	23	2	26	1	12	35	0	67.2
4th Truck	0	17	9	32	16	8	31	30	6	11	0	105.7
Total Distances :											340	

4.5 Perbandingan rute Parallel Insertion (Tour Improvement) dengan (Tour Construction)

Pada Tabel 4.7 terlihat perbaikan

(*Improvement*) dan perpindahan rute dari masing-masing kendaraan, dengan kapasitas volume yang tetap sama, hanya titik lokasinya yang berubah namun menghasilkan jarak yang terbaik. Pada Tabel 4.8 sampai dengan Tabel 4.11 berikut ini adalah hasil dan perbandingan rute sebelumnya (*parallel*) dengan rute yang sudah diperbaiki dengan *Exhaustive Search* dan Enumerasi.

Tabel 4.8. Perbandingan hasil dari *Tour Construction* dan *Tour Improvement 1st Truck*

1 st TRUCK												
INITIAL ROUTE (Parallel Insertion)												
Distance (km)	0	18	27	13	21	14	24	3	7	5	0	81.7
Volume (kg)	0	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	10000
Volume (kg)	0	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	10000
IMPROVED ROUTE (Exhaustive Search)												
Distance (km)	0	18	27	13	21	14	24	3	7	5	0	81.7
Volume (kg)	0	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	10000
Volume (kg)	0	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	10000

Tabel 4.9. Perbandingan hasil dari *Tour Construction* dan *Tour Improvement 2nd Truck*

2 nd TRUCK												
INITIAL ROUTE (Parallel Insertion)												
Distance (km)	0	36	19	22	20	15	33	34	25	4	0	105.4
Volume (kg)	0	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	10000
Volume (kg)	0	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	10000
IMPROVED ROUTE (Exhaustive Search)												
Distance (km)	0	36	19	22	20	15	33	34	25	4	0	105.4
Volume (kg)	0	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	10000
Volume (kg)	0	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	10000

Tabel 4.10. Perbandingan hasil dari *Tour Construction* dan *Tour Improvement 3rd Truck*

3 rd TRUCK												
INITIAL ROUTE (Parallel Insertion)												
Distance (km)	0	28	10	29	23	2	26	1	12	35	0	67.2
Volume (kg)	0	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	10000
Volume (kg)	0	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	10000
IMPROVED ROUTE (Exhaustive Search)												
Distance (km)	0	28	10	29	23	2	26	1	12	35	0	67.2
Volume (kg)	0	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	10000
Volume (kg)	0	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	10000

Tabel 4.11. Perbandingan hasil dari *Tour Construction* dan *Tour Improvement 4th Truck*

4 th TRUCK												
INITIAL ROUTE (Parallel Insertion)												
Distance (km)	0	17	9	32	16	8	31	30	6	11	0	105.7
Volume (kg)	0	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	10000
Volume (kg)	0	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	10000
IMPROVED ROUTE (Exhaustive Search)												
Distance (km)	0	17	9	32	16	8	31	30	6	11	0	105.7
Volume (kg)	0	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	10000
Volume (kg)	0	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	10000

ANALISA PERMASALAHAN

Analisa Hasil

Jarak yang dihasilkan dari metode *sequential (savings)*, *parallel insertion*, dan *exhaustive search* selalu menghasilkan solusi dengan jarak tempuh terpendek. Sekalipun waktu perhitungan metode tersebut tidak selalu unggul pada kelompoknya masing-masing, karena banyak membutuhkan sumber daya khususnya waktu dalam perhitungannya.

Analisa Matriks jarak

Saat pembuatan matriks jarak, dipilih lintasan yang memiliki jarak tempuh terpendek untuk menghubungkan tiap titik. Sesudah rute didapatkan, diketahui bahwa jarak solusi rute yang diolah secara matematis dengan jarak solusi rute. Namun kemungkinan google maps masih belum cukup akurat untuk menghitung jarak antar titik. Tetapi jarak yang berbeda dalam google maps adalah wajar terjadi mengingat bahwa google maps sendiri memberikan beberapa alternatif jalan untuk menghubungkan dua titik. Maka dalam pencarian jarak, rute terpendek yang diusulkan oleh google maps yang dipilih pada awal pembuatan matriks jarak.

Analisa Cost Saving

Algoritma savings Clark and Wright mampu menghasilkan solusi yang baik bagi permasalahan rute kendaraan. Dengan berdasarkan pada prinsip penghematan, maka penyusunan rute kendaraan akan lebih efisien. Hal ini dikarenakan prioritas pemasukan titik tujuan untuk membentuk suatu rute menjadi sangat jelas, yaitu mendahulukan titik-titik tujuan yang memberikan penghematan (*savings*) terbesar. Dengan prinsip ini, penyusunan rute kendaraan menjadi lebih terarah dan teratur. Pelanggan dengan nilai *savings* terbesar akan diprioritaskan untuk masuk terlebih dahulu ke dalam rute.

Analisa Tour Construction

Dalam *Tour Construction*, ada dua versi dalam proses memasukkan pelanggan ke dalam rute. Versi pertama adalah *Sequential version*. Versi ini membentuk rute demi rute secara bertahap. Semua nilai *savings* yang terdapat pada *savings list* akan dibaca dan pelanggan-pelanggan yang terkait dengan nilai tersebut dicoba untuk dimasukkan ke dalam rute. Apabila *feasible* (sesuai dengan kendala yang ada), maka pelanggan-pelanggan tersebut

akan masuk ke dalam rute.

Apabila tidak memungkinkan lagi untuk memasukkan pelanggan pada rute tersebut, maka rute tersebut dianggap selesai dibangun kemudian mulai membentuk rute yang baru dengan cara yang sama (mengulang pembacaan nilai *savings* lalu mencoba memasukkan pelanggan-pelanggan yang belum masuk ke dalam rute). Prioritas pembentukan rute kendaraan terkonsentrasi pada pembentukan satu rute terlebih dahulu berdasarkan nilai *savings* terbesar. Dari proses *Tour Construction* menggunakan *Sequential Version* dihasilkan total jarak tempuh sebesar 454.7 km.

Versi kedua yaitu *Parallel Insertion*. Pada versi ini, rute baru yang terbentuk bisa lebih dari satu pada setiap kali pembacaan nilai *savings*. Prioritas pembentukan kendaraan sangat terkonsentrasi pada nilai *savings* terbesar. Cara menghitung solusi versi *sequential* dan *Parallel Insertion* mudah dipahami, dan keduanya menghasilkan solusi yang menunjukkan perbandingan antara *Sequential* dan *Parallel Insertion*, dan hasilnya dapat terlihat perbandingan rute dan kapasitas volume yang lebih baik. Sedangkan dari proses *Tour Construction* menggunakan *Parallel Insertion* dihasilkan total jarak tempuh sebesar 453.8 km.

Analisa Tour Improvement

Berdasarkan hasil yang terlihat pada Tabel 4.5, maka dipilihlah *Parallel Insertion* sebagai rute awal, untuk selanjutnya akan dilakukan kembali perbaikan rute awal (*Tour Construction*) dengan melakukan *Exhaustive Search* dan *Enumerasi* untuk menghasilkan rute baru atau perbaikan rute (*Tour Improvement*) yang terbaik.

Pada tahap ini, solusi awal yang telah dibuat pada *tour construction* akan mengalami perbaikan dengan menggunakan metode yang tergolong dalam *improvement method*. Perbaikan terhadap solusi tersebut dilakukan salah satunya dengan menukarkan letak node (titik) yang berada dalam satu solusi (*single route improvement*). Letak atau urutan node berpindah-pindah sedemikian rupa menurut aturan tertentu. Selama perpindahan terjadi, apabila solusi yang lebih baik ditemukan, maka solusi yang terdahulu akan digantikan dengan solusi yang baru sampai ditemukan hasil yang terbaik. Dan dari proses *Tour Improvement* menggunakan *Exhaustive Search* dihasilkan total jarak tempuh sebesar 360 km.

Gambar 5.1. Hasil Tour Improvement Truk 1 pada *Google Maps*



Gambar 5.2. Hasil Tour Improvement Truk 2 pada *Google Maps*



Gambar 5.3. Hasil Tour Improvement Truk 3 pada *Google Maps*



Gambar 5.4. Hasil Tour Improvement Truk 4 pada *Google Maps*



diketahui bahwa metode *savings* dikombinasikan dengan metode perbaikan *Parallel Insertion*, *Exhaustive Search* dan *Enumerasi* mampu menghasilkan solusi (rute kendaraan) dalam membuat *Tour Improvement* dan *Tour Construction* dengan jarak tempuh yang lebih baik.

Tabel 6.1. Perbandingan Hasil/Total Jarak Tempuh Tiap Metode.

Metode	Total Jarak Tempuh (km)
Sequential Version	545,7
Parallel Insertion (Tour Construction)	553,8
Exhaustive Search (Tour Improvement)	360

Dengan demikian metode-metode ini mampu menghasilkan solusi yang semakin baik untuk permasalahan rute kendaraan PT. Starmass Logistics.

DAFTAR PUSTAKA

Ballou, R. H. (2004). *Business Logistics/ Supply Chain Management*. New Jersey: Prentice Hall.

Fauzy, Akhmad. (2008). "Statistik Industri". Erlangga : Jakarta

Helsgaun, K. (2006). *An Effective Implementation of K-opt Moves for the Lin Kernighan TSP Heuristic*. Roskilde: Computer Science, Roskilde University. Didapat dari : <http://akira.ruc.dk/~keld/research/LKH/KoptReport.pdf>

Lysgaard, J. (1997). *Clarke & Wright's Savings Algorithm*. Department of Management Science and Logistics, The Aarhus School of Business.

Munir, Rinaldi. (2004). *Strategi Algoritmik dan Algoritma Exhaustive Search*. Bandung : Institut Teknologi Bandung (ITB)

Toth dan Vigo (2002). *The Vehicle Routing Problem*. Philadelphia, SIAM Monographs on Discrete Mathematics and Application.

KESIMPULAN

Dari hasil penelitian penentuan metode heuristics klasik terbaik pada permasalahan rute kendaraan (studi kasus: PT. Starmass Logistics),