

Optimasi pada Rute Truk Peti Kemas dengan Algoritma Optimasi Koloni Semut

Feisy D. Kambey
feisy.kambey@yahoo.co.id

Abstrak— Perdagangan global yang semakin berkembang mengakibatkan terjadinya peningkatan arus peti kemas. Salah satu bagian penting dari transportasi peti kemas adalah truk peti kemas yang harus dioptimalkan untuk menunjang kelancaran arus peti kemas. Makalah ini mengajukan salah satu bentuk optimasi transportasi peti kemas, yaitu dengan penentuan rute optimal truk peti kemas. *Ant Colony Optimization* sebagai salah satu metode optimasi dengan pendekatan stokastik dipilih sebagai metode optimasi untuk penentuan rute optimal truk peti kemas. Metode optimasi tersebut terinspirasi dari proses biologis perilaku sekumpulan semut yang bekerja sama untuk mencari makan. Perjalanan sekumpulan semut yang menyebar mencari makan dari sarangnya menuju sumber makanan kemudian kembali lagi ke sarangnya dan pada akhirnya konvergen ke satu jalur terpendek menjadi inspirasi perancangan algoritma untuk penentuan rute optimal dari truk peti kemas ini. *Rank-Based Ant System* sebagai salah satu algoritma *Ant Colony Optimization* dikenakan pada sistem penentuan rute truk peti kemas ini untuk menentukan algoritma yang memberikan hasil terbaik.

Kata Kunci — *Ant Colony Optimization*, optimasi, rute optimal, truk peti kemas.

I. PENDAHULUAN

TRANSPORTASI peti kemas semakin berkembang dengan adanya kebutuhan perpindahan barang dalam jumlah yang besar sehubungan dengan perdagangan global. Indonesia yang turut serta dalam perdagangan global pun mulai memperhatikan pembangunan sistem manajemen pelabuhan yang baik. Dalam hal ini, terminal peti kemas di Pelabuhan Tanjung Priok adalah salah satu terminal peti kemas yang cukup padat. Data dari PT. PELINDO II, arus peti kemas keseluruhan yang mencakup peti kemas MTI, Tanjung Priok, TPK Koja dan JICT pada tahun 2011 adalah sebesar 5,8 juta TEU. Arus peti kemas ini diperkirakan akan meningkat menjadi 18 juta TEU pada tahun 2030, dengan asumsi pertumbuhan 6% per tahun [3].

Arus peti kemas akan dimulai dengan pemuatan barang ke dalam peti kemas, kemudian diantar oleh truk peti kemas ke pelabuhan, lalu dijemput lagi oleh truk peti kemas untuk dibawa ke penerima peti kemas. Dari keseluruhan perjalanan peti kemas ini, truk peti kemas mengambil bagian yang cukup penting di dalamnya. Jumlah truk yang melayani pelabuhan Tanjung Priok menurut data yang tercatat di Unit Angkutan

Khusus Pelabuhan (Angsuspel) Organda DKI Jakarta memperlihatkan kecenderungan yang semakin meningkat dari sejumlah 9.081 unit pada tahun 2010, pada 2011 naik menjadi 9.900 unit, kemudian pada 2012 tercatat 11.015 unit, dan mencapai 12.000 unit sampai Agustus 2013. Oleh karena itu optimasi transportasi truk peti kemas perlu dilakukan untuk kelancaran arus peti kemas secara khusus juga kelancaran transportasi pada umumnya.

Makalah ini mengajukan salah satu bentuk optimasi transportasi truk peti kemas, yaitu dengan penentuan rute optimal truk peti kemas. Dalam hal ini, tujuan optimasi adalah meminimumkan total waktu operasi truk secara keseluruhan.

II. PEMODELAN SISTEM

A. Identifikasi Model

Sistem ini dirancang untuk model transportasi truk peti kemas dengan multi depot dan multi terminal. Dalam hal ini depot adalah lokasi dimana peti kemas kosong dapat disimpan dan truk-truk diparkirkan. Ada dua tipe peti kemas, yaitu peti kemas masuk (*inbound*) dan keluar (*outbound*). Peti kemas masuk adalah peti kemas yang berlokasi di terminal yang harus dibawa ke penerima atau depot. Sedangkan peti kemas yang berada di pelanggan (pengirim) atau depot yang harus dibawa ke pelabuhan, disebut peti kemas keluar. Selanjutnya, setiap tipe peti kemas terbagi atas peti kemas yang berisi barang kiriman (*full*) dan kosong (*empty*). Dalam hal ini, peti kemas kosong adalah peti kemas yang diekspor atau diimpor karena adanya kebutuhan peti kemas.

Sehingga, terdapat empat jenis peti kemas yaitu : *inbound full* (IF), *outbound full* (OF), *inbound empty* (IE), dan *outbound empty* (OE). Pertama, peti kemas IF adalah peti kemas yang harus dijemput di terminal dan diangkut ke truk, diantar ke penerimanya, diturunkan dan dibongkar di tempat penerimanya. Suatu peti kemas IF yang telah selesai ditangani di tujuannya akan menghasilkan peti kemas kosong yang harus dikembalikan ke depot. Kedua, peti kemas OF adalah barang kiriman yang harus diantar dan berlokasi di tempat pelanggan. Karena itu harus diantar peti kemas kosong ke lokasi pelanggan, muat barang ke dalam peti kemas dan mengantarnya ke terminal. Ketiga, peti kemas IE pada awalnya berlokasi di terminal. Peti kemas ini harus dijemput di terminal dan ditransportasikan ke depot. Terakhir peti kemas OE yang merupakan peti kemas kosong yang harus diantar ke terminal.

Perumusan Graf Berarah

Masalah penjadwalan rute truk peti kemas dirumuskan secara matematis menggunakan graf berarah, $G = \{V, A\}$. Pemodelan menggunakan perumusan graf berarah untuk mendapatkan model dan fungsi objektif masalah optimasinya didasarkan pada penelitian Zhang, dkk [4], dengan sedikit modifikasi untuk menyesuaikan dengan sistem transportasi truk peti kemas di Pelabuhan Tanjung Priok.

Himpunan titik adalah $V = V_D \cup V_C$. Dalam hal ini V_D adalah himpunan titik awal/balik, dan V_C adalah titik peti kemas. Sedangkan himpunan busur adalah $A = \{(i, j) | i \in V_D, j \in V_C, \text{atau } i \in V_D \cup V_C\}$.

Graf G dirumuskan berdasarkan rangkaian kegiatan-kegiatan. Dalam perumusan graf ini, proses pemecahan masalahnya yaitu menemukan rute-rute (busur) yang optimal. Busur menghubungkan satu titik peti kemas ke titik yang lain, atau satu titik awal/balik ke satu titik peti kemas. Akan tetapi tidak ada busur antara dua titik awal/balik.

Setiap titik awal/balik, $i \in V_D$, terkait dengan suatu depot i , dimana $i = 1, \dots, m$. Atribut dari titik awal/balik hanyalah jumlah awal dari truk yang berlokasi di depot. Suatu titikpeti kemas, $i \in V_C$, adalah serangkaian kegiatan dari peti kemas yang bersesuaian, dimana $i = m + 1, \dots, m + n$. Indeks yang sama, yaitu i , digunakan untuk depot ($i = 1, \dots, m$) dan peti kemas ($i = m + 1, \dots, m + n$) karena baik depot maupun peti kemas berkaitan dengan suatu titik. Dalam hal ini, m adalah jumlah depot dan n adalah jumlah peti kemas.

Semua terminal dan lokasi pelanggan, baik pengirim maupun penerima, memiliki *time window* untuk setiap peti kemas tetapi depot tidak memiliki *time window*. Akibatnya, semua peti kemas masuk/keluar yang berisi memiliki dua *time window*, yaitu *time window* di asal dan tujuannya. Sementara peti kemas masuk/keluar yang kosong hanya memiliki satu *time window*.

Dua *time window*, yaitu *time window* dari lokasi asal dan tujuan, harus dikenakan pada titik peti kemas IF dan OF. Karena itu, dua *time window* dari titik i ($P_i \in \{IF', OF'\}$) dikombinasikan dengan memperhatikan waktu layanan peti kemas di asalnya (t_{oi}) dan di tujuannya (t_{di}). Pertama-tama, aktifitas di titik i ($P_i \in \{IF', OF'\}$) harus dimulai pada *time window*-nya sendiri, $[a_i, b_i]$, dalam hal ini merupakan kombinasi antara *time window* di asal, $[a_{oi}, b_{oi}]$, dan *time window* di tujuan, $[a_{di}, b_{di}]$. Kedua, kegiatan ini harus dimulai sebelum waktu $b_{di} - t_{oi}$ agar ada cukup waktu untuk aktivitas di lokasi asal. Ketiga, aktifitas ini harus dimulai setelah waktu $a_{di} - t_{oi}$ agar supaya waktu tunggu di lokasi tujuan sebelum waktu a_{di} dapat dihindari. Maka *time window*, $[a_i, b_i]$, dan waktu pemrosesan (τ_i) pada suatu $node_i \in V_C$ dapat diperoleh sebagaimana yang dirumuskan pada (1), (2), dan (3).

$$a_i = \begin{cases} \min[\max(a_{oi}, a_{di} - t_{oi} - t(O_i, D_i)), b_{oi}] & \text{jika } P_i \in \{IF', OF'\} \\ a_{oi} & \text{jika } P_i \in \{IE'\} \\ a_{di} & \text{jika } P_i \in \{OE'\} \end{cases} \quad (1)$$

$$b_i = \begin{cases} \min(b_{oi}, b_{di} - t_{oi} - t(O_i, D_i)) & \text{jika } P_i \in \{IF', OF'\} \\ b_{oi} & \text{jika } P_i \in \{IE'\} \\ b_{di} & \text{jika } P_i \in \{OE'\} \end{cases} \quad (2)$$

TABELI
RUMUSAN WAKTU TRANSFER

$i \setminus j$	Depot	IF/IE	OF	OE
Depot	-	$t(H_i, O_j)$	$t(H_i, H_{kj}) + t_{ul} + t(H_{kj}, O_j)$	$t(H_i, H_{kj}) + t_{ul} + t(H_{kj}, O_j)$
IF	$t(D_i, H_{ki}) + t_{ul} + t(H_{ki}, H_j)$	$t(D_i, H_{ki}) + t_{ul} + t(H_{ki}, O_j)$	$t(D_i, H_{ki}) + 2t_{ul} + t(H_{ki}, H_{kj}) + t(H_{kj}, O_j)$	$t(D_i, H_{ki}) + 2t_{ul} + t(H_{ki}, H_{kj}) + t(H_{kj}, D_j)$
IE	$t(O_i, H_{ki}) + t_{ul} + t(H_{ki}, H_j)$	$t(O_i, H_{ki}) + t_{ul} + t(H_{ki}, O_j)$	$t(O_i, H_{ki}) + 2t_{ul} + t(H_{ki}, H_{kj}) + t(H_{kj}, O_j)$	$t(O_i, H_{ki}) + 2t_{ul} + t(H_{ki}, H_{kj}) + t(H_{kj}, D_j)$
OF/OE	$t(D_i, H_j)$	$t(Tr_i, Tr_j)$	$t(D_i, H_{kj}) + t_{ul} + t(H_{kj}, O_j)$	$t(D_i, H_{kj}) + t_{ul} + t(H_{kj}, D_j)$

$$\tau_i = \begin{cases} \max([a_{di} - b_{oi}], [t_{oi} + t(O_i, D_i)]) + t_{di} & \text{jika } P_i \in \{IF', OF'\} \\ t_{oi} & \text{jika } P_i \in \{IE'\} \\ t_{di} & \text{jika } P_i \in \{OE'\} \end{cases} \quad (3)$$

Suatu busur $(i, j) \in A$ didefinisikan sebagai transfer dari titik i ke titik j yang juga terdiri dari berbagai rangkaian kegiatan, di mana waktu transfer (τ_{ij}) dirumuskan sesuai Tabel I. Dalam hal ini, waktu yang diperlukan untuk berpindah antara dua lokasi dilambangkan dengan $t(\blacksquare, \blacksquare)$. Lokasi asal dan tujuan dari titik i masing-masing dilambangkan dengan O_i dan D_i . Depot yang berperan sebagai titik awal/balik i dilambangkan dengan H_i , sedangkan depot untuk mengembalikan atau mengambil peti kemas kosong untuk titik i dilambangkan dengan H_{ki} . Terakhir, terminal yang menjadi tujuan atau asal dari peti kemas di titik i dilambangkan dengan Tr_i .

Rangkaian kegiatan transfer ini terutama adalah perpindahan dari lokasi tujuan peti kemas sebelumnya ke lokasi asal peti kemas selanjutnya. Dalam beberapa kasus juga termasuk menaikkan atau menurunkan peti kemas, dengan waktu t_{ul} . Setelah menangani peti kemas inbound, truk harus ke depot untuk mengembalikan peti kemas kosong yang muatannya telah diturunkan di penerima, sehingga terdapat tambahan waktu untuk mengembalikan peti kemas kosong ke depotnya. Di sisi lain, sebelum menangani peti kemas outbound, truk harus mengambil peti kemas kosong terlebih dahulu ke depot yang dipesan oleh pengirim, sehingga terdapat waktu tambahan untuk mengambil peti kemas kosong.

Dengan perumusan graf ini, maka dapat ditentukan bobot untuk perpindahan dari titik i ke titik j , yang merupakan bobot waktu total, T_{ij} sebagai gabungan waktu pemrosesan pada (3) dan waktu transfer di Tabel 1 menjadi (4).

$$T_{ij} = \begin{cases} \tau_{ij}, & i \in V_D, j \in V_C \\ \tau_{ij} + \tau_i, & i \in V_C, j \in V_D \cup V_C \end{cases} \quad (4)$$

III. ALGORITMA ANT COLONY OPTIMIZATION

Ant colony optimization (ACO) merupakan salah satu metode optimasi dengan pendekatan stokastik yang mendapatkan inspirasi dari perilaku beberapa spesies semut dalam mencari makan. Semut ini menyetor feromon di tanah untuk menandai jalur yang baik yang harus diikuti oleh anggota lain dari koloni. *Ant colony optimization* memanfaatkan mekanisme yang sama untuk memecahkan masalah optimasi.

Dalam ACO, sejumlah semut tiruan membangun solusi untuk masalah optimasi dan bertukar informasi mengenai kualitas solusi tersebut melalui skema komunikasi yang diadopsi dari perilaku semut nyata. *Ant colony optimization* telah diformalkan ke dalam bentuk metaheuristik untuk masalah optimasi yang ditunjukkan dalam bentuk algoritma pada Gambar 1. Setelah inisialisasi, metaheuristik akan beriterasi atas tiga fase, yaitu *ConstructAntSolutions*, *UpdatePheromones*, dan *DaemonActions*.

Pada fase pertama, sejumlah solusi dibangun oleh satu himpunan semut tiruan. Pemilihan solusi ini dituntun oleh mekanisme stokastik, yang dibiarkan oleh feromon yang terdapat pada calon solusi tersebut.

Fase berikutnya adalah proses dimana jejak feromon dimodifikasi. Tujuan dari modifikasi ini adalah untuk meningkatkan nilai feromon yang berhubungan dengan solusi yang baik atau menjanjikan, dan untuk mengurangi nilai feromon yang berhubungan dengan solusi yang buruk. Dari sudut pandang praktis, penambahan feromon meningkatkan kemungkinan bahwa komponen yang terpilih oleh paling tidak satu semut dan menghasilkan solusi yang sangat baik tersebut akan dipilih lagi oleh semut yang akan datang. Dengan cara yang berbeda, penguapan feromon mengimplementasikan bentuk "*forgetting*", yang menghindari konvergensi algoritma yang terlalu cepat ke daerah sub optimal, sehingga memperkenankan eksplorasi area pencarian yang baru.

Fase terakhir adalah prosedur "*daemon*", yang merupakan langkah opsional, tergantung pada masalah yang akan diselesaikan. Contohnya adalah observasi jalur yang ditemukan oleh setiap semut dalam koloni dan memilih satu atau beberapa semut yang membangun solusi terbaik dalam iterasi algoritma, yang kemudian akan diijinkan menambah feromon pada komponen (jalur) yang digunakan semut tersebut.

Tetapkan parameter, inisialisasi jejak feromon
while kondisi terminasi belum tercapai **do**
 ConstructAntSolutions
 UpdatePheromones
 DaemonActions
endwhile

Gbr. 1. Metaheuristik *Ant Colony Optimization*

Dalam perkembangannya, *Ant Colony Optimization* telah dimodifikasi dalam berbagai bentuk algoritma untuk memberikan hasil yang lebih baik. Beberapa algoritma ACO, antara lain *Ant System*, *Elitist Ant System*, dan *Rank-Based Ant System*.

A. *Ant System*

Dua fase utama dari algoritma AS adalah konstruksi solusi dan perbaruan nilai feromon. Di algoritma AS, m semut tiruan membangun solusinya. Pada awalnya semut-semut tersebut ditempatkan pada titik awal masing-masing yang dipilih secara acak. Pada setiap langkah konstruksi, semut k mengaplikasikan aturan pemilihan probabilistik, yang disebut aturan random proportional, untuk menentukan titik mana yang akan dikunjungi berikutnya. Dalam hal ini, probabilitas semut k yang berada di titik i untuk bergerak ke titik j adalah

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, \text{ jika } j \in \mathcal{N}_i^k \quad (5)$$

dimana $\eta_{ij} = 1/d_{ij}$ adalah nilai heuristik yang tersedia sebelumnya, α dan β adalah dua parameter yang menentukan pengaruh relatif dari jejak feromon dan informasi heuristik, dan \mathcal{N}_i^k adalah titik-titik tetangga dari semut k yang berada di titik i .

Penguapan feromon diimplementasikan dengan (6).

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij}, \quad \forall (i, j) \in L \quad (6)$$

di mana $0 < \rho \leq 1$ adalah tingkat penguapan feromon. Setelah penguapan, semua semut menambahkan feromon di jalur yang telah dilalui sesuai (7).

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k, \quad \forall (i, j) \in L \quad (7)$$

di mana $\Delta\tau_{ij}^k$ adalah jumlah feromon yang ditambahkan oleh semut k di jalur yang dilaluinya, yang didefinisikan dengan (8).

$$\Delta\tau_{ij}^k = \begin{cases} 1/C^k & \forall (i, j) \in T^k \\ 0 & \text{lainnya} \end{cases} \quad (8)$$

di mana C^k , panjang jalur T^k yang dibangun oleh semut k

B. *Elitist Ant System*

Perbedaan dari algoritma ini dengan *Ant System* adalah pada proses deposit feromon. Dalam hal ini, gagasannya adalah untuk memberikan penguatan tambahan pada cabang yang menjadi bagian dari solusi terbaik yang ditemukan sejak

algoritma dimulai, di mana solusi ini selanjutnya akan dilambangkan dengan T^{bs} .

Penguatan tambahan pada solusi T^{bs} dicapai dengan menambahkan nilai e/C^{bs} pada jalurnya, di mana e adalah parameter yang mendefinisikan bobot yang diberikan pada solusi T^{bs} , dan C^{bs} adalah panjang jalurnya. Maka (7) diubah menjadi (9).

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k + e\Delta\tau_{ij}^{bs}, \quad (9)$$

di mana $\Delta\tau_{ij}^k$ didefinisikan seperti pada (8) dan $\Delta\tau_{ij}^{bs}$ didefinisikan seperti (10).

$$\Delta\tau_{ij}^{bs} = \begin{cases} 1/C^{bs} & \forall (i,j) \in T^{bs} \\ 0 & \text{lainnya} \end{cases} \quad (10)$$

Sedangkan untuk penguapan feromon pada algoritma ini diimplementasikan sebagaimana yang dinyatakan oleh (6).

C. Rank-Based Ant System

Pada algoritma ini juga, modifikasi dari algoritma *Ant System* terjadi pada proses deposit feromon. Dalam hal ini, sebelum memperbarui jejak feromon, semut-semut diurutkan sesuai kualitas solusi dan jumlah feromon yang ditambahkan oleh suatu semut akan diberikan bobot berdasarkan pada peringkat r dari semut tersebut. Pada setiap iterasi hanya $(w-1)$ semut dengan peringkat terbaik dan semut yang menghasilkan solusi best-so-far yang dimungkinkan untuk menambahkan feromonnya. Semut yang menghasilkan solusi best-so-far tidak harus berada pada kumpulan semut di iterasi terakhir. Maka aturan perbaruan feromon pada algoritma ASrank adalah sesuai dengan (11).

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{r=1}^{w-1} (w-r)\Delta\tau_{ij}^r + w\Delta\tau_{ij}^{bs}, \quad (11)$$

Di mana $\Delta\tau_{ij}^r = 1/C^r$ dan $\Delta\tau_{ij}^{bs} = 1/C^{bs}$.

IV. PERANCANGAN SISTEM PENENTUAN RUTE OPTIMAL TRUK PETI KEMAS DENGAN ALGORITMA ANT COLONY

Sistem penentuan rute ini dirancang menggunakan MATLAB®2011b (7.13.0.564), dimana data peti kemas diimpor dari dan hasilnya diekspor ke Microsoft Excel 2010. Sedangkan program ini dijalankan di Notebook TOSHIBA Satellite L645, dengan prosesor intelCOREi3.

Sistem penentuan rute ini dirancang dengan tujuan untuk mendapatkan waktu total minimum yang dibutuhkan oleh semua truk. Untuk penjadwalan ini dibutuhkan data sebagai sebagaimana diuraikan sebelas poin berikut.

Poin pertama adalah Tipe peti kemas, selanjutnya adalah *Time window* asal peti kemas (TW terminal untuk IF, dan TW pengirim untuk OF), *Time window* tujuan peti kemas (TW penerima untuk IF dan TW terminal untuk OF), Waktu untuk

pemrosesan bongkar atau muat barang, yang kelima adalah Waktu dari asal ke tujuan peti kemas. Poin keenam adalah data Depot untuk mengembalikan atau mengambil peti kemas kosong, selanjutnya Terminal yang menjadi tujuan atau asal peti kemas, Waktu dari asal peti kemas ke masing-masing depot yang ada, Waktu dari tujuan peti kemas ke masing-masing depot yang ada, Waktu dari depot ke depot dan poin yang terakhir adalah Waktu dari terminal ke terminal.

Keunggulan dari sistem ini terutama adalah kemudahan untuk memperoleh rute truk peti kemas dengan total waktu minimal. Selain itu tiga keunggulan lain dari sistem ini adalah dapat meminimalkan jumlah truk yang dioperasikan, menentukan waktu dimulainya kegiatan di tiap titik dengan memperhatikan *time window* dan keunggulan yang ketiga adalah Sistem penjadwalan truk kontainer berlaku untuk multi depot dan multi terminal.

Perhitungan bobot waktu pada sistem ini didasarkan pada model yang diajukan oleh Zhang, dkk (2010) tentang penjadwalan rute truk berbasis heuristik tetapi dengan sedikit modifikasi karena model yang diajukan Zhang, dkk ini menganggap tipe "IE" dan "OE" adalah sama, tetapi pada model dalam penelitian ini kedua tipe peti kemas tersebut dianggap berbeda. Sedangkan untuk algoritma optimasi, Zhang, dkk menggunakan pendekatan deterministik dan metode window partition based agar dapat menentukan waktu mulai kegiatan di tiap titik yang tentu saja akan menjadi rumit dalam pemrograman jika peti kemas yang harus ditangani sangat banyak. Sementara sistem penjadwalan truk dengan algoritma *Ant Colony Optimization* tetap dapat menentukan waktu mulai kegiatan di tiap titik tanpa membuat partisi *time window*.

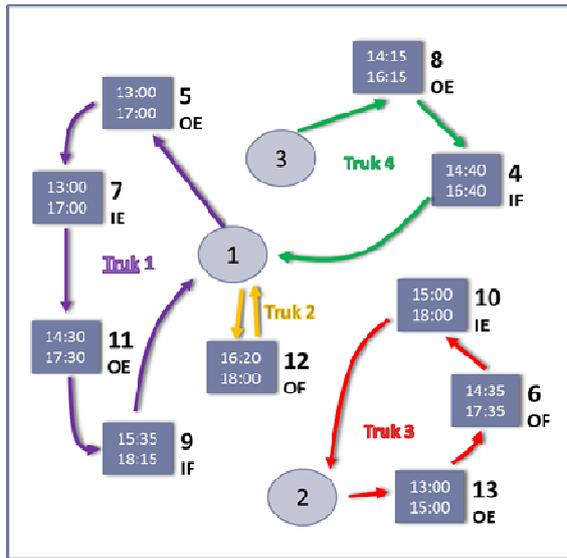
V. HASIL KOMPUTASI

Rank Based Ant System sebagai salah satu algoritma ACO menunjukkan hasil yang terbaik dibanding algoritma ACO yang lain untuk sistem penentuan rute optimal truk peti kemas ini. Oleh karena itu, dalam pengujian ini algoritma yang digunakan adalah *Rank Based Ant System*.

Sebagai contoh hasil komputasi, sistem penentuan rute ini dikenakan pada sistem yang terdiri dari 3 depot, 2 terminal, dengan sejumlah 10 peti kemas yang harus ditangani dalam satu hari. Dengan data dari tiap peti kemas yang dibangkitkan secara acak, maka rute dari truk peti kemas akan terlihat sebagaimana yang dijelaskan pada Gambar 2.

Dari hasil optimasi ini, maka untuk menangani 10 peti kemas dengan memperhatikan masing-masing *time window* dan tipenya, dibutuhkan 4 unit truk.

Tabel II menunjukkan perbandingan rute truk yang menangani 10 peti kemas dalam sehari, antara rute dengan ACO dan tanpa optimasi. Dengan asumsi bahwa jika tanpa optimasi satu truk akan menangani satu peti kemas, maka berarti dibutuhkan 10 truk peti kemas yang beroperasi sedangkan jika rute tersebut dioptimalkan dengan ACO maka hanya dibutuhkan 4 truk yang beroperasi.



Gbr. 2. Hasil Optimasi Route Truk Peti Kemas

Terdapat selisih waktu total 305 menit juga antara rute yang menggunakan ACO dan tanpa optimasi, yang tentunya juga berimbas pada penghematan BBM sebesar 50,83 liter untuk perkiraan konsumsi BBM truk peti kemas sebesar 10 liter/jam.

VI. KESIMPULAN

Dengan adanya optimasi menggunakan ACO, pada penentuan rute truk yang harus menangani 10 peti kemas, maka hanya dibutuhkan 4 truk dibandingkan dengan tanpa optimasi yang membutuhkan 10 truk. Sedangkan selisih total waktu operasi dengan ACO dan tanpa optimasi adalah sebesar 305 menit yang berimbas pada penghematan BBM sebanyak 50, 83 liter.

Sistem penentuan rute ini dapat lebih mengoptimalkan transportasi peti kemas di sisi darat jika ditunjang dengan sistem penentuan *time window* yang baik di terminal peti kemas.

REFERENSI

[1] M. Dorigo, & T. Stutzel, *Ant Colony Optimization*, A Bradford Book – The MIT Press, Cambridge, Massachusetts, 2004.
 [2] M. Dorigo, dkk, *Ant Colony Optimization : Artificial Ants as a Computational Intelligence Technique*, dalam *IEEE Computational Intelligence Magazine*, IEEE, 2006.
 [3] *Rencana Induk Pelabuhan Tanjung Priok* dalam Lampiran Peraturan Menteri Perhubungan. Nomor : PM. 38 Tahun 2012. Tanggal : 13 Juni 2012.
 [4] Zhang, dkk. *Heuristic-based truck scheduling for inland container transportation*, dalam *OR Spectrum* (2010) 32:787-808, Springer – Verlag, 2010.

TABELII
 PERBANDINGAN RUTE TRUK DENGAN ACO DAN TANPA OPTIMASI

Tanpa Optimasi				Dengan Optimasi			
Truk	Rute	Waktu Operasi (menit)	BBM (liter)	Truk	Rute	Waktu Operasi (menit)	BBM (liter)
1	1-4-1	245	40,83	1	1-5-7-11-9-1	510	85,00
2	1-5-1	125	20,83	2	1-12-1	205	34,17
3	2-6-2	225	37,50	3	3-8-4-1	300	50,00
4	3-7-3	115	19,17	4	2-13-6-10-2	380	63,33
5	3-8-3	130	21,67	Total	1395	232,5	
6	1-9-1	250	41,67				
7	2-10-2	105	17,50				
8	2-11-2	185	30,83				
9	1-12-1	205	34,17				
10	2-13-2	115	19,17				
Total		1700	283,33				