

Analisa Local Search untuk Perbaikan Solusi Varian Vrp pada Permasalahan Optimasi

Sapti Wahyuningsih

Jurusan Matematika FMIPA Universitas Negeri Malang
sapti.wahyuningsih.fmipa@um.ac.id

Info Artikel

Riwayat Artikel:

Diterima: 15 Mei 2017
Direvisi: 1 Juni 2017
Diterbitkan: 31 Juli 2017

Kata kunci:

local search
varian VRP
algoritma
perbaikan solusi

ABSTRACT

Teori graf mempunyai medan aplikasi yang luas khususnya pada permasalahan optimasi. Permasalahan *Vehicle Routing Problem* (VRP) merupakan kajian dalam teori graf yang dapat diaplikasikan pada permasalahan optimasi. *Multi Depot Vehicle Routing Problem* (MDVRP), *Vehicle Routing Problem with time window* (VRPTW) dan gabungannya yaitu MDVRPTW merupakan varian VRP yang akan dianalisa perbaikan solusinya dengan strategi *local search*. Metode yang digunakan untuk menyelesaikan permasalahan varian VRP ini adalah algoritma *self-developed*, algoritma *interchange descent* dan algoritma *exchange local search*. Strategi penyelesaian varian VRP adalah tahap inisialisasi, tahap pengembangan dan tahap optimalisasi. Pada tahap pengembangan perbaikan solusi dianalisa dengan *local search*. Hasil solusi akhir algoritma *self developed* lebih baik dibandingkan algoritma *interchange descent*. Algoritma *interchange descent* melakukan perbaikan *inter route* dan *intra route* mengakibatkan proses iterasi yang lama tetapi diperoleh kemungkinan solusi hasil perbaikan lebih banyak jika dibanding *exchange local search*. Penentuan total waktu tempuh dapat dilihat dari seberapa lama kendaraan melakukan perjalanan. Algoritma *self developed* menghasilkan total waktu tempuh kendaraan yang minimal dibandingkan dengan algoritma *interchange descent*. Analisa *local search* untuk perbaikan solusi MDVRP, VRPTW dan MDVRPTW berguna untuk identifikasi solusi terbaik pada implementasi permasalahan optimasi.

Copyright © 2017 SIMANIS.
All rights reserved.

Corresponding Author:

Sapti Wahyuningsih
Jurusan Matematika FMIPA
Universitas Negeri Malang,
Jl. Semarang 5 Malang Jawa Timur, Indonesia 65145
Email: sapti.wahyuningsih.fmipa@um.ac.id

1. PENDAHULUAN

Dalam teori graf pembahasan *Vehicle Routing Problem* (VRP) merupakan salah satu bentuk permasalahan yang dapat diaplikasikan pada masalah transportasi yang melibatkan sistem distribusi. Permasalahan optimasi rute kendaraan mengenai sejumlah *customer* (pelanggan) pada lokasi tertentu dan harus dilayani oleh suatu depot dengan menggunakan sejumlah kendaraan dengan kapasitas yang terbatas. Terdapat bermacam-macam varian VRP bergantung kepada variabel, konstrain dan tujuan yang akan dioptimalkan. *Multi Depot Vehicle Routing Problem* (MDVRP), *Vehicle Routing Problem with Time Window* (VRPTW), *Multi Depot Vehicle Routing Problem with Time Window* (MDVRPTW) merupakan varian VRP yang akan menjadi fokus kajian artikel ini.

Untuk mendapatkan solusi optimal, terdapat bermacam-macam metode yang dapat digunakan untuk mencari solusi terbaik pada bervariasi varian VRP. Pada [1] penentuan solusi VRPTW dengan algoritma genetik. Pada [13] dan [16] penulis mengkaji penentuan solusi MDVRP; solusi MDVRP dengan algoritma

genetik dapat dilihat pada [13] sedangkan penentuan solusi MDVRP dengan algoritma *Cellular Ant* dapat dilihat pada [16]. Dari pengembangan VRPTW dan MDVRP akan diperoleh varian baru yaitu MDVRPTW yang dapat dilihat pada [6], [7], [9], [15]. Penentuan solusi MDVRPTW dengan algoritma *tabu search* yang dikombinasikan dengan *extended saving* dapat dilihat pada [14].

Pemilihan fokus kajian tentang permasalahan MDVRP, VRPTW dan MDVRPTW karena ketiga varian ini mempunyai tiga prinsip dasar dalam menentukan solusi permasalahan yaitu tahap inisialisasi, tahap pengembangan dan tahap optimalisasi. Pada masalah MDVRP tahap inisialisasi berupa *grouping*, tahap pengembangan berupa *routing* dan tahap optimalisasi berupa *scheduling*. Metode yang akan digunakan algoritma *self-developed*, algoritma *Interchange descent* dan algoritma *exchange local search*. Pemilihan metode ini diperlukan untuk strategi yang akan dipilih yaitu analisa *local search* pada tahap pengembangan. Pada tahap pengembangan metode *local search* salah satu cara yang dapat digunakan untuk perbaikan solusi. Misalnya dapat dilihat pada [2], [3], [4] dan [11]. Metode *local search* dapat diadaptasi dari varian TSP dapat dilihat pada [10].

Metode *local search* untuk perbaikan solusi MDVRP, VRPTW dan MDVRPTW yang dikaji pada artikel ini adalah *inter route* dan *intra route* untuk mengidentifikasi solusi akhir yang lebih baik. Hal ini berguna pada implementasi permasalahan optimasi.

2. METODE

Permasalahan MDVRP, VRPTW dan MDVRPTW dapat diselesaikan dengan algoritma algoritma *self-developed*, algoritma *interchange descent* dan algoritma *exchange local search*. Algoritma *Self-Developed* merupakan algoritma yang dibuat dengan menambahkan beberapa modifikasi untuk *saving algorithm* berdasarkan pada intuisi untuk banyak rute. Algoritma *Self-Developed* menghitung *saving* dengan tabel waktu akhir penyelesaian sehingga rute yang dihasilkan dapat menunjukkan waktu yang dibutuhkan untuk setiap rute yang terbentuk serta jarak dan permintaan yang dibutuhkan. Selanjutnya rute-rute dijadwalkan pada tahap *scheduling* dan akhirnya mendapatkan kumpulan rute pada masing-masing depot [2].

Adapun langkah-langkah Algoritma *Self-Developed* yaitu inisialisasi dan iterasi.

Tahap inisialisasi (semua *customer* yang ada pada daftar dan diurutkan menurun berdasarkan pada permintaan. Semua rute penjemputan harus *layak* dengan memperhatikan waktu perjalanan. Jika suatu rute tidak *layak* maka tidak terdapat solusi dari suatu permasalahan). Sedangkan tahap iterasi (kombinasi rute, penghitungan *saving*, dan uji kelayakan)

Algoritma *interchange descent* merupakan suatu kombinasi *heuristic* yang bermacam-macam untuk memeriksa daerah solusi untuk menjadikan daerah solusi yang lebih baik [15]. *Interchange Descent* terdiri dari perbaikan *inter-route* dan *intra-route*. Langkah-langkah algoritma *Interchange Descent* terdiri dari tahap inialisasi, iterasi, dan perbaikan rute. Tahap inialisasi dimulai dengan membuat rute sementara untuk setiap *customer*. Tahap iterasi akan dihitung *savings* penghematan, lalu akan dilakukan penggabungan rute berdasarkan daftar *saving*. Setelah solusi awal terbentuk akan dilakukan perbaikan rute.

Algoritma *Exchange Local Search* merupakan salah satu *improving heuristic* yang digunakan untuk mengoptimalkan rute yang telah dibentuk. Secara umum, pada metode ini dilakukan perpindahan titik dari suatu rute sehingga terbentuk rute yang baru. Algoritma *Exchange Local Search* ini merupakan permutasi antara dua titik *customer* [8].

3. HASIL DAN DISKUSI

Multi Depot Vehicle Routing Problem (MDVRP) merupakan varian VRP berupa permasalahan menentukan keseluruhan rute untuk sejumlah kendaraan dari lebih dari satu depot (*multi depot*) untuk suatu himpunan *customer* (pelanggan) dan kembali ke depot yang sama dengan total jarak pengiriman yang minimum tanpa melanggar kendala kapasitas. Tujuan dari permasalahan MDVRP adalah mencari sejumlah rute minimum pada setiap depot dimana *customer* dilayani tepat sekali dan kendaraan berawal juga berakhir pada depot yang sama serta tidak melanggar kendala kapasitas.

Merujuk pada [16] permasalahan MDVRP yang didefinisikan oleh graph komplit $G = (V, A)$ adalah suatu himpunan titik yaitu $V = \{v_1, v_2, \dots, v_m, v_{m+1}, v_{m+2}, \dots, v_{m+n}\}$ dengan v_1 sampai v_m adalah titik berkorespondensi ke m depot dan v_{m+1} sampai v_{m+n} adalah titik yang berkorespondensi ke *customer*. Untuk masing-masing titik *customer*, ada suatu permintaan q_i yang bernilai positif. Himpunan A adalah suatu himpunan sisi yaitu $A = \{(v_i, v_j): v_i, v_j \in V, i \neq j\}$. Untuk masing-masing sisi (v_i, v_j) mempunyai biaya c_{ij} non negatif yang menyatakan biaya dari *customer* i ke *customer* j . Selanjutnya, suatu armada dari K kendaraan ditempatkan pada m depot. Kendaraan ke k mempunyai muatan Q_k . Fungsi tujuan untuk MDVRP adalah

$$\text{Min } \sum_{i \in I \cup D} \sum_{d \in I \cup J} \sum_{k \in K} C_{id} X_{id}^k = C_{1d} X_{1d}^1 + \dots + C_{nd} X_{nd}^n$$

$$X_{id}^k = \begin{cases} 1, & \text{jika kendaraan dijalankan dari titik } i \text{ ke } d \\ 0, & \text{untuk yang lain} \end{cases}$$

$$Z_{id} = \begin{cases} 1, & \text{jika customer } i \text{ dialokasikan untuk depot } d \\ 0, & \text{untuk yang lain} \end{cases}$$

Z_{id} adalah banyaknya pengelompokan pada tiap depot.

Merujuk pada [1] permasalahan VRPTW merupakan varian VRP berupa permasalahan meminimalkan total biaya perjalanan, tanpa mengabaikan batasan kapasitas kendaraan dan time window. Desain rute dilakukan sedemikian hingga setiap customer hanya dikunjungi sekali oleh satu kendaraan, dan setiap kendaraan memulai dan mengakhiri rutenya pada suatu depot. Permasalahan VRPTW dapat didefinisikan pada graph yaitu misalkan $G = (N, E)$ adalah suatu graph dimana $N = \{0, 1, \dots, n\}$ adalah himpunan titik dan $E \subseteq N \times N$ adalah himpunan sisi. Jika $(i, j) \in E$, maka mungkin untuk melakukan perjalanan dari i ke j , dengan waktu tempuh t_{ij} . Titik 0 adalah suatu depot. $V = \{1, \dots, m\}$ adalah himpunan kendaraan yang digunakan. Sedangkan titik pada himpunan $N / \{0\}$ merepresentasikan himpunan customer yang akan dilayani. Setiap dari customer tersebut memiliki permintaan q_i dan T merupakan maximum waktu tempuh yang disediakan oleh perusahaan.

Fungsi tujuan untuk meminimumkan total biaya semua kendaraan model matematika dari VRPTW adalah:

$$\sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} X_{ij}^k$$

$$X_{ij}^k = \begin{cases} 1, & \text{jika kendaraan } k \text{ dijalankan dari titik } i \text{ ke titik } j \text{ dengan } i \neq j \\ 0, & \text{jika kendaraan } k \text{ tidak dijalankan dari titik } i \text{ ke titik } j \end{cases}$$

Dengan batasan-batasan khusus yang membedakan dengan varian VRP yang lain adalah Setiap kendaraan k memulai pelayanan pada titik j dengan memenuhi

$$\sum_{k \in K} \sum_{i \in V} X_{ij}^k (b_i^k + w_i + f_i + t_{ij}) = b_j^k, \quad \forall j \in V \setminus \{0\}$$

Pelayanan dilakukan berdasarkan batasan waktu tiap-tiap customer

$$e_i \leq b_i^k + w_i + f_i \leq l_i, \quad \forall i \in V, k \in K$$

Fungsi tujuan untuk meminimumkan total biaya semua kendaraan model matematika dari VRPTW

$$\sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij} X_{ij}^k$$

$$X_{ij}^k = \begin{cases} 1, & \text{jika kendaraan } k \text{ dijalankan dari titik } i \text{ ke titik } j \text{ dengan } i \neq j \\ 0, & \text{jika kendaraan } k \text{ tidak dijalankan dari titik } i \text{ ke titik } j \end{cases}$$

Dengan batasan-batasan khusus yang membedakan dengan varian VRP yang lain adalah Setiap kendaraan k memulai pelayanan pada titik j dengan memenuhi

$$\sum_{k \in K} \sum_{i \in V} X_{ij}^k (b_i^k + w_i + f_i + t_{ij}) = b_j^k, \quad \forall j \in V \setminus \{0\}$$

Pelayanan dilakukan berdasarkan batasan waktu tiap-tiap customer

$$e_i \leq b_i^k + w_i + f_i \leq l_i, \quad \forall i \in V, k \in K$$

Suatu himpunan $arcA = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$. Untuk masing-masing $arc(v_i, v_j)$, ada suatu travel timenon-negatif atau cost c_{ij} . Suatu armada dari K kendaraan ditempatkan pada m depot, masing-masing customer dilayani oleh satu dan hanya satu kendaraan, total muatan dan durasi dari kendaraan k tidak melebihi Q_k dan T_k , pelayanan customer i mulai di dalam time-window $[e_i, l_i]$ dan masing-masing rute mulai dan berakhir di dalam time window depot yang berkaitan [7].

Penerapan algoritma dengan perbaikan solusi pada varian VRP dapat dijelaskan berikut. Algoritma *Interchange Descent* merupakan keluarga *local search* yang terdiri dari perbaikan *inter-route* dan *intra route*. Algoritma *Interchange Descent* digunakan untuk memperbaiki rute awal yang telah terbentuk. Hal yang harus dilakukan untuk menggunakan Algoritma *Interchange Descent* yaitu tahap pertama melakukan inisialisasi

neighbor list yang memuat perpindahan *inter-route*. Enam struktur *inter-route neighborhood* sebagai berikut *shift (1,0)*: satu *customer i* dipindahkan dari rute 1 ke rute 2. *Swap (1,1)*: pertukaran antara *customer i* dari rute 1 dan *customer j* dari rute 2. *Shift (2,0)*: dua *customer* terhubung *i* dan *j* (atau sisi (i,j)) dipindahkan dari rute 1 ke rute 2. *Swap (2,1)*: pertukaran antara *customer* terhubung *i* dan *j* dari rute 1 dengan *customer k* dari rute 2. *Swap (2,2)*: pertukaran antara *customer* terhubung *i* dan *j* dari rute 1 dengan dua *customer* terhubung lain *k* dan *l* dari rute 2. *Cross*: sisi antara *customer* terhubung *i* dan *j* dari rute 1 dan sisi antara *customer k* dan *l* dari rute 2 dihapus. Kemudian, dua sisi (i,l) dan (k,j) disisipkan.

Jika terdapat perbaikan, maka perbaikan yang terbaik akan menggantikan solusi awal. Selanjutnya dilakukan perbaikan dengan *intra-route*. Empat struktur *intra-route neighborhood* sebagai berikut. *Or-opt* : satu, dua atau tiga *customer* yang terhubung dihapus dan disisipkan kembali ke dalam posisi yang berbeda pada rute. *2-opt*: dua sisi yang tidak terhubung dihapus dan dua sisi yang lain ditambahkan sedemikian sehingga rute baru terbentuk. *Exchange*: permutasi antara dua customer. Ini merupakan tipe *swap(1,1)* untuk *intra-route*. *Reinsertion* : satu *customer* dihapus dan disisipkan kembali ke dalam posisi yang berbeda pada rute.

Penentuan solusi varian VRP dengan algoritma *self-developed* merupakan algoritma yang dibuat dengan menambahkan beberapa modifikasi-modifikasi untuk *savings algorithm* berdasarkan pada intuisi untuk banyak rute. Permasalahan MDVRP dapat diselesaikan dengan Algoritma *Self-Developed*. Pencarian solusi dimulai dengan mengelompokkan pelanggan pada depot yang terdekat sehingga terbentuk beberapa kelompok rute yang akan dipasangkan untuk setiap depot (*grouping*). Kemudian menghitung serta mengurutkan rute berdasarkan *savings* dari terbesar ke terkecil, mengkombinasikan rute-rute tersebut dengan melihat kendala waktu serta kapasitas setiap rute yang dipasangkan (*routing*). Dari proses tersebut dihasilkan rute yang layak pada setiap depot yang berawal dan berakhir di depot tersebut. Setelah itu menjadwalkan pengiriman berdasarkan rute yang terbentuk dan telah dipasangkan dengan kendaraan yang ada (*scheduling*).

Permasalahan MDVRPTW merupakan perluasan dari VRP dengan banyaknya depot untuk melayani *customer* lebih dari satu dan terdapat kendala kapasitas dan waktu. Penyelesaian permasalahan MDVRPTW serupa dengan penyelesaian MDVRP yaitu *grouping*, *routing*, dan *scheduling*. Pada langkah *routing* permasalahan MDVRPTW diselesaikan dengan algoritma *self-developed* dan algoritma *Interchange descent*. Pada tahap inisialisasi dimulai dengan menyusun semua rute penjemputan dengan setiap kendaraan melayani tepat satu *customer*. Setelah itu dilanjutkan pada langkah iterasi, yaitu pengkombinasian rute dengan perhitungan *savings*. Penghitungan *savings* dikembangkan dengan modifikasi *farthest insertion*. Dari daftar *savings* yang terbentuk, *savings* yang terpilih adalah *savings* dengan nilai terbesar. Setelah semua *customer* terpilih dan terpasangkan ke kendaraan maka terbentuk solusi awal.

Perbaikan rute menggunakan *Exchange Local Search*, dengan cara pertukaran (permutasi) dari dua titik *customer* pada tiap rute. Setelah rute akhir hasil *local search* terbentuk maka dilanjutkan dengan menjadwalkan kumpulan rute tersebut dan dipasangkan pada tiap kendaraan pada masing-masing depot.

4. IMPLEMENTASI PADA PERMASALAHAN OPTIMASI

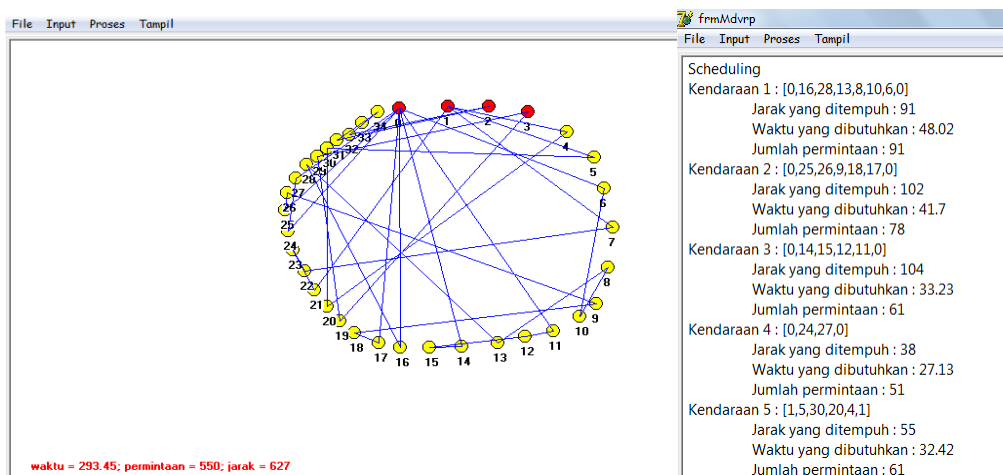
Implementasi permasalahan optimasi distribusi ini dapat menggunakan modul ICT dari hasil penelitian hibah bersaing [17].

Tabel 1. Data Jarak antar *customer*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1																							
2	2		7	8	3	2	5	6	8	9	11	14	10	12	15	16	12	17	18	19	12	11	9
3	3	7		19	12	11	4	15	17	12	11	14	16	18	12	19	10	14	16	12	13	16	17
4	2	8	19		18	14	12	15	13	18	17	19	10	14	15	16	17	12	12	10	19	21	22
5	5	3	12	18		14	15	15	12	11	11	13	14	16	12	18	16	17	19	12	13	14	12
6	4	2	11	14	14		10	18	15	16	12	19	20	23	22	21	17	19	15	16	13	12	16
7	3	5	4	12	15	10		17	18	19	10	21	22	23	24	25	26	21	22	20	21	24	23
8	7	6	15	15	15	18	17		21	21	23	24	28	26	23	27	21	29	28	21	23	21	25
9	5	8	17	13	12	15	18	21		16	15	18	19	10	21	22	21	22	24	25	26	27	29
10	6	9	12	18	11	16	19	21	16		35	36	38	40	42	46	40	35	22	20	24	27	26
11	7	11	11	17	11	12	10	23	15	35		27	31	32	35	37	30	30	31	29	28	27	20
12	5	14	14	19	13	19	21	24	18	36	27		30	31	28	29	20	32	31	27	28	29	20
13	6	10	16	10	14	20	22	28	19	38	31	30		37	29	31	31	32	33	35	36	37	38
14	9	12	18	14	16	23	23	26	10	40	32	31	37		36	35	34	31	32	30	35	34	31
15	8	15	12	15	12	22	24	23	21	42	35	28	29	36		30	31	33	34	32	36	35	38
16	12	16	19	16	18	21	25	27	22	46	37	29	31	35	30		30	32	37	34	31	32	29
17	10	12	10	17	16	17	26	21	21	40	30	20	31	34	31	30		36	39	37	31	30	33
18	11	17	14	12	17	19	21	29	22	35	30	32	32	31	33	32	36		17	19	20	23	25
19	12	18	16	12	19	15	22	28	24	22	31	31	33	32	34	37	39	17		31	30	20	21
20	13	19	12	10	12	16	20	21	25	20	29	27	35	30	32	34	37	19	31		27	22	26
21	14	12	13	19	13	13	21	23	26	24	28	28	36	35	36	31	31	20	30	27		23	26

Tabel 2. Data permintaan, kapasitas kendaraan dan kecepatan

Titik	Permintaan
4	18
5	12
6	13
7	14
8	16
9	10
10	19
11	18
12	12
13	13
14	16
15	17
16	19
17	12
18	14



Gambar. Graph Hasil dan perhitungannya

5. KESIMPULAN

Berdasarkan analisis kinerja algoritma *self developed* dan algoritma *Interchange Descent* pada MDVRPTW, maka dapat disimpulkan bahwa algoritma *self developed* menghasilkan solusi akhir yang lebih minimal dibanding algoritma *interchange descent*. Hal ini disebabkan pada tahap penentuan solusi awal algoritma *self developed* menghasilkan solusi yang lebih baik karena adanya proses penentuan urutan *customer* yang hendak dilayani menggunakan metode *furthest insertion* (FI). Penggunaan FI ini bermaksud untuk memperoleh suatu rute dengan jarak tempuh minimum. Pada tahap perbaikan rute *local search* algoritma *interchange descent* melakukan perpindahan rute *inter route* dan *intra route*. Kemungkinan solusi hasil perbaikan rute *interchange descent* lebih banyak dibandingkan *exchange local search*. Tetapi karena pada solusi awal algoritma *self developed* menghasilkan jarak tempuh yang lebih minimal, maka pada hasil solusi akhir algoritma *self developed* masih menghasilkan solusi lebih baik dibandingkan algoritma *interchange descent*.

Setelah solusi awal terbentuk langkah perbaikan rute menggunakan *local search*. Perbaikan rute disini bertujuan untuk mengoptimalkan solusi awal yang telah terbentuk. Tipe *local search* yang digunakan *exchange local search*, *local search inter route* dan *intra route*. Pada proses iterasi *exchange local search* hanya melakukan pertukaran titik untuk *intra route* saja. Sedangkan untuk algoritma *interchange descent* melakukan perbaikan *inter route* dan *intra route*. Hal ini mengakibatkan pada Algoritma *interchange descent* menjadikan proses iterasi yang lama. Namun akan diperoleh kemungkinan solusi hasil perbaikan lebih banyak jika dibanding *exchange local search*.

Setelah dilakukan perbaikan *local search* hasil kinerja algoritma menunjukkan adanya perbaikan solusi dari solusi awal. Namun karena pada solusi awal algoritma *self developed* menghasilkan solusi yang lebih baik, maka pada hasil solusi akhir algoritma *self developed* masih tetap lebih unggul dibandingkan algoritma *interchange descent*. Penentuan total waktu tempuh dapat dilihat dari seberapa lama kendaraan melakukan perjalanan. Algoritma *self developed* menghasilkan total waktu tempuh kendaraan yang minimal dibandingkan dengan algoritma *interchange descent*.

Analisa *local search* untuk perbaikan solusi varian VRP pada artikel ini diterapkan pada MDVRP, VRPTW dan MDVRPTW. Pada pengembangannya dapat diterapkan pada varian yang lain misalnya CVRP, MTRVP, MFVRP, VRPB, VRPPD atau kombinasinya. Demikian juga metode yang digunakan dapat menggunakan algoritma lain [12], [18], [19].

DAFTAR PUSTAKA

- [1] Braysy, O.& Gendreau, M. 2005. Genetic Algorithms for Vehicle Routing Problem with Time Windows, Department of Mathematics and Statistic, University of Vasa, Finlandia. *Transportation Science*, (Online), 39(1): 119-139
- [2] Crevier, B., Jean-Francois Cordeau., & Gilbert Laporte. 2007. *The Multi-Depot Vehicle Routing Problem with inter-depot routes* (*European Journal of Operational Research* 176 (2007) 756-773). Canada: ELSEVIER (online),
- [3] En-nahli, L., Afifi, S., Allaoui, H. & Nouaouri, I. 2016. *Local Search Analysis for a Vehicle Routing Problem with Synchronization and Time Windows Constraints in Home Health Care Services*. International Federation of Automatic Control, France, 2016. (Online), 49(12): 1210-1215
- [4] Penna, P.V., Subramanian, A.&Ochi, L.S. 2011. *An Iterated Local Search Heuristik for The Heterogeneous Fleet Vehicle Routing Problem*, (Online),
- [5] Kiran, M, dkk. 2013. [A Modified Savings Algorithm Based Approach for Vehicle Routing Problem with Simultaneous Pick-up and Delivery](#). *International Journal of Emerging Technology and Advanced Engineering*(Online)
- [6] Li, J., Pardalos, P.& Li, Y. 2014. Multi Depot Vehicle Routing Problem With Time Windows Under Shared Depot Resources. College of Engineering, Nanjing Agricultural University, Nanjing, China. *Journal of Combinatorial Optimization*, (Online),31: 515,
- [7] Polacek, M.; Hartl, R.F.; dan Doerner, K. 2005. "A Variable Neighborhood Search for the Multi Depot Vehicle Routing Problem with TimeWindows." *Journal of Heuristics*, 10: 613-627.

- [8] Polat, O., Kalayci, C., Kulak, O., Gunther, H. 2015. A Perturbation Based Variable Neighborhood Search Heuristic For Solving The Vehicle Routing Problem With Simultaneous Pickup And Delivery With Time Limit. *European Journal of Operational Research*. S0377-2217(14)00814-5.
- [9] Ramalingam, A. dan Vivekanandan, K. 2014. "Genetic Algorithm based Solution Model for Multi-Depot Vehicle Routing Problem with Time Windows." *International Journal of Advanced Research in Computer and Communication Engineering*, 3: 8433-8499.
- [10] Rostami, Ali S., Mohanna, F., Keshavarz, H. & Hosseinabadi, A.A.R. 2015. Solving Multiple Traveling Salesman Problem using the Gravitational Emulation Local Search Algorithm. *Applied Mathematics & Information Sciences*, (Online), 9(2): 699-709,
- [11] Silva, M.M., Subramanian, A. & Ochi, L.S. 2015. An Iterated Local Search Heuristic for The Split Delivery Vehicle Routing Problem. *Computer & Operations Research* (Online),
- [12] Subramanian, A., Drummond, L.M.A., Bentes, C., Ochi, L.S. & Farias, R. 2010. A Parallel Heuristic for The Vehicle Routing Problem with Simultaneous Pickup and Delivery. *Computer & Operations Research* (Online), 37: 1899-1911,
- [13] Sumathi, S., & Surekha P. 2011. Solution To Multi-Depot Vehicle Routing Problem Using Genetic Algorithms. *WAP Journal*, (Online), 1(3) : 118-131,
- [14] Tamashiro, H.; Nakamura, M.; Okazaki, T.; dan Kang, D. 2007. "A Tabu Search Approach combined with An Extended Saving Method for Multi-depot Vehicle Routing Problems with Time Windows." *Biomedical Soft Computing and Human Sciences*, 15(1): 29-37.
- [15] Timmermann, T. 2010. An approach to solve the Multi Depot Vehicle Routing Problem with Time Windows (MDVRPTW) In Static And Dynamic Scenarios. *Proceedings of the 22. Workshop "Planen, Scheduling und Konfigurieren, Entwerfen*, (Online)
- [16] Wang, Y. 2013. "Research of Multi-Depot Vehicle Routing Problem by Cellular Ant Algorithm." *Journal of Computers*, 8(7): 1722-1727.
- [17] Wahyuningsih, S dan Satyananda, D (2014 & 2015). Pengembangan modul penerapan teori graph berbasis ICT sebagai pedoman PKL mahasiswa jurusan matematika di Industri. LP2M UM: laporan penelitian hibah bersaing.
- [18] Xing, W. 2016. An Improved Savings Method for Vehicle Routing Problem. *International Conference on Control Science and Systems Engineering* (Online)
- [19] Yeun, L.C.; Ismail, W.R.; Omar, K.; dan Zirour, M. 2008. "Vehicle Routing Problem: Models and Solutions." *Journal of Quality Measurement and Analysis*, 4(1): 205-218.