

Penyelesaian Capacitated Vehicle Routing Problem (CVRP) Menggunakan Algoritma Tabu Search HGreX Crossover

Suheri

*Jurusan Teknik Elektro, Politeknik Negeri Pontianak, Jalan Jendral Ahmad Yani,
Pontianak Tenggara, Kota Pontianak, 78124
e-mail: chinsuvent@gmail.com*

Abstrak: Kebutuhan akan adanya metode yang otomatis dapat menentukan rute optimal dalam kasus CVRP, menuntun munculnya berbagai pendekatan yaitu eksak, heuristik, dan metaheuristik. Salah satu algoritma dalam pendekatan heuristik/metaheuristik yang telah banyak digunakan adalah Tabu Search. Tabu Search memanfaatkan teknik local search solution space dan tabu list untuk dapat terhindar dari local optimum atau konvergensi yang prematur, namun memerlukan proses yang panjang dikarenakan pencarian yang bersifat sekuensial. Oleh karena itu, digunakanlah operator HGreX Crossover untuk meningkatkan kualitas solusi yang dihasilkan dan mengurangi waktu eksekusi. Hasil pengujian pada dataset Augerat dengan kode A-n32-k5 menunjukkan bahwa penggunaan algoritma Tabu Search HGreX Crossover mampu meningkatkan kualitas solusi secara berturut-turut sebesar 14.88%, 19.08%, dan 14.04% untuk 100, 500, dan 1000 iterasi jika dibandingkan dengan Tabu Search konvensional dan waktu yang dibutuhkan Tabu Search HGreX Crossover secara berturut-turut lebih cepat 4.43%, 4.47%, dan 3.44% untuk 100, 500, dan 1000 iterasi dibandingkan dengan Tabu Search konvensional.

Kata Kunci: CVRP, Tabu Search, HGreX Crossover

The Solving Capacitated Vehicle Routing Problem (CVRP) Using Tabu Search HGreX Crossover Algorithm

Abstract: The need for methods that can be automatically applied to determine optimal route in case of CVRP, leading the emergence of a variety of approaches such as exact, heuristics, and metaheuristics. One of the algorithm in heuristic/Metaheuristic approach that has been widely used is Tabu Search. Tabu Search utilizing local search solution space and taboo list to be able to avoid a local optimum or premature convergence, but requires a long process because the search is sequential. Therefore, HGreX Crossover operator are used to improve the quality of the resulting solution and reduce the execution time. The test results on the dataset Augerat with code A-n32-k5 showed that the use of Tabu Search algorithm HGreX Crossover able to improve the quality of the solution, respectively by 14.88%, 19.08%, and 14.04% for 100, 500, and 1000 iterations when compared with conventional Tabu Search and Tabu Search HGreX Crossover respectively more faster 4.43 %, 4.47%, and 3.44% for 100, 500, and 1000 iterations compared with conventional Tabu Search.

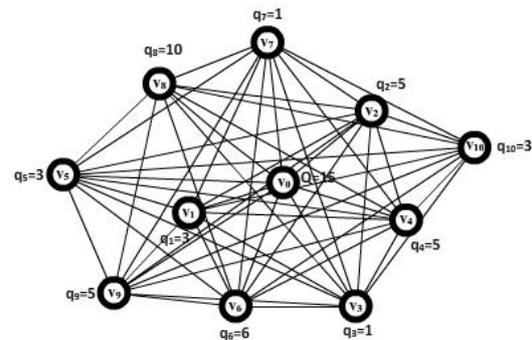
Keywords: CVRP, Tabu Search, HGreX Crossover

Vehicle Routing Problem (VRP) merupakan istilah umum yang digunakan untuk semua permasalahan yang melibatkan perancangan rute optimal untuk armada kendaraan yang melayani sejumlah *customer* dengan batasan-batasan tertentu (Baldacci, R., Mingozzi, A. and Roberti, R., 2012) yang merupakan perluasan dari permasalahan *Traveling Salesman Problem* (TSP) (Toth, P. and Vigo, D., 2002a). Terdapat banyak contoh aplikasi dari permasalahan VRP seperti pengumpulan dan pengantaran barang, pembersihan jalan, penentuan rute bus sekolah, pengumpulan sampah, *dial a ride systems*, transportasi orang cacat, dan penentuan rute penjualan [10].

VRP dapat dibagi ke dalam beberapa tipe yang tergantung dari varian batasan yang digunakan, salah satunya adalah *Capacitated Vehicle Routing Problem* (CVRP). Awal mula munculnya CVRP dimulai dari penelitian yang dilakukan oleh Dantzing and Ramser (1959) berkaitan dengan permasalahan distribusi *gasoline*. CVRP merupakan bentuk dasar dari VRP (Toth, P. and Vigo, D., 2002a) dengan batasan berupa kapasitas muatan dari kendaraan yang digunakan untuk pendistribusian barang.

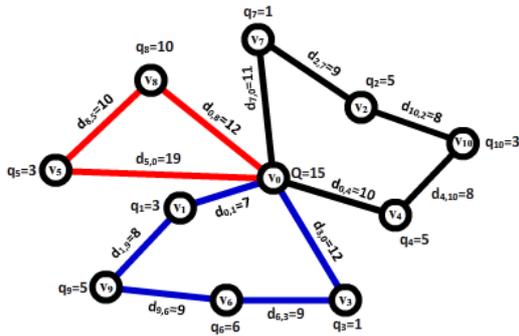
Menurut El Hassani et al., (2008), CVRP dapat dinyatakan sebagai sebuah *weighted directed graph* $G = (V, A)$ dimana $V = \{v_0, v_1, v_2, \dots, v_n\}$ menunjukkan kumpulan *vertex* dan $A = \{(v_i, v_j) : i \neq j\}$ menunjukkan kumpulan *arcs* atau busur yang menghubungkan 2 (dua) buah *vertex*. *Vertex* v_0 menunjukkan depot dan lainnya menunjukkan konsumen. Sebuah nilai *non negative* d_{ij} dihubungkan dengan setiap *arc* (v_i, v_j) . Nilai ini menunjukkan jarak antara *vertex* v_i dan *vertex* v_j pada konteks biaya atau waktu antara dua buah *vertex*. Sebuah data permintaan (*demand*) q_i dan atau waktu pelayanan

$\delta_i (q_0 = 0, \delta_0 = 0)$ dihubungkan dengan tiap konsumen (*vertex*) v_i . Pada CVRP, tujuan yang ingin dicapai adalah untuk meminimalkan total biaya rute berupa jarak dan dalam waktu bersamaan dapat menyelesaikan kendala-kendala seperti: 1) setiap konsumen hanya dikunjungi tepat satu kali dan hanya oleh satu kendaraan; 2) semua jalan/rute berawal dan berakhir di depot; 3) jumlah total permintaan konsumen di tiap jalan/rute tidak melebihi kapasitas setiap kendaraan. Adapun contoh ilustrasi dari kasus *Capacitated Vehicle Routing Problem* (CVRP) disajikan pada Gambar 1.



Gambar 1. Contoh Ilustrasi Kasus CVRP

Pada Gambar 1 di atas diilustrasikan sebuah kasus CVRP dengan 1 buah simpul v_0 sebagai depot, kapasitas muatan armada kendaraan $Q = 15$, sejumlah simpul atau konsumen sebanyak 10 ($v_1, v_2, v_3, \dots, v_{10}$) dengan masing-masing permintaan $q_1=3, q_2=5, q_3=1, q_4=5, q_5=3, q_6=6, q_7=1, q_8=10, q_9=5$, dan $q_{10}=3$. Sedangkan Gambar 2 berikut ini merupakan salah satu contoh ilustrasi sebuah solusi CVRP berdasarkan contoh kasus pada Gambar 1.



Gambar 2. Contoh Solusi Kasus CVRP

Pada Gambar 2 di atas menggambarkan sebuah solusi dari kasus CVRP berdasarkan pada Gambar 1, yaitu $(v_0, v_4, v_{10}, v_2, v_7, v_0, v_8, v_5, v_0, v_1, v_9, v_6, v_3, v_0)$ yang terdiri dari 3 (tiga) rute, yaitu rute 1 $(v_0, v_4, v_{10}, v_2, v_7, v_0)$, rute 2 (v_0, v_8, v_5, v_0) , dan rute 3 $(v_0, v_1, v_9, v_6, v_3, v_0)$.

CVRP merupakan jenis permasalahan kombinatorial yang diklasifikasikan sebagai *NP-hard problem*. Oleh karena itu, diperlukan suatu metode khusus untuk dapat menyelesaikan permasalahan CVRP. Terdapat beberapa pendekatan yang dapat digunakan untuk menyelesaikan permasalahan CVRP, yaitu pendekatan eksak, heuristik, dan metaheuristik. Salah satu algoritma dari pendekatan metaheuristik yang telah banyak digunakan dan mampu menyelesaikan kasus CVRP dengan kinerja yang baik adalah algoritma *Tabu Search* (Guan, C., Cao, Y. and Shi, J., 2010), (Jia, H., Li, Y., Dong, B. and Ya, H., 2013), (Toth, P. and Vigo, D., 2002b).

Algoritma *Tabu Search* konvensional (Pirim, H., Bayraktar, E. and Eksioğlu, B., 2008) bekerja dengan memanfaatkan teknik pencari ruang solusi *local search* dan menggunakan *tabu list* untuk dapat terhindar dari *local optimum* atau konvergensi yang prematur, namun memerlukan proses yang panjang dikarenakan pencarian

menggunakan teknik *local search* yang bersifat sekuensial. Oleh karena itu, penulis mengusulkan ide penggunaan operator *crossover* sebagai satu tahapan yang perlu dilakukan untuk memperbaiki solusi yang digunakan dalam pembentukan solusi tetangga (*neighborhood*) dengan harapan bahwa proses penemuan solusi yang mendekati dan atau *global optimum* akan menjadi lebih cepat. Adapun operator *crossover* yang digunakan adalah HGreX, dimana operator HGreX merupakan salah satu dari 3 (tiga) operator *crossover* algoritma genetika yang memberikan hasil terbaik dari 8 (delapan) operator *crossover* yang diuji kinerjanya untuk penyelesaian kasus VRP (Puljić, K. and Manger, R., 2013) dan sejauh ini belum ada penelitian yang mencoba menggabungkan algoritma *Tabu Search* konvensional dengan operator *HGreX Crossover*. Oleh karena itu, penelitian ini menggabungkan kedua buah metode tersebut dengan harapan operator *HGreX Crossover* dapat meningkatkan kinerja dari algoritma *Tabu Search* konvensional.

METODE

Pada penelitian ini, konsep CVRP yang peneliti gunakan mengacu pada penelitian yang dilakukan oleh El Hassani et al., (2008), yaitu CVRP dapat dinyatakan sebagai sebuah *weighted directed graph* $G = (V, A)$ dan $V = \{v_0, v_1, v_2, \dots, v_n\}$ menunjukkan kumpulan *vertex* serta $A = \{(v_i, v_j) : i \neq j\}$ menunjukkan kumpulan *arcs* atau busur yang menghubungkan 2 (dua) buah *vertex*. *Vertex* v_0 menunjukkan depot dan lainnya menunjukkan konsumen. Sebuah nilai *non negative* d_{ij} dihubungkan dengan setiap *arc* (v_i, v_j) . Nilai ini menunjukkan jarak antara *vertex* v_i dan *vertex* v_j pada konteks biaya atau waktu antara dua buah *vertex*.

Sebuah data permintaan (*demand*) q_i dan atau waktu pelayanan δ_i ($q_0 = 0, \delta_0 = 0$) dihubungkan dengan tiap konsumen (*vertex*) v_i . Pada CVRP, tujuan yang ingin dicapai adalah untuk meminimalkan total biaya rute berupa jarak dan dalam waktu bersamaan dapat menyelesaikan kendala-kendala seperti: 1) setiap konsumen hanya dikunjungi tepat satu kali dan hanya oleh satu kendaraan; 2) semua jalan/rute berawal dan berakhir di depot; 3) jumlah total permintaan konsumen di tiap jalan/rute tidak melebihi kapasitas setiap kendaraan.

Berdasarkan uraian di atas, deskripsi dari daftar variabel yang perlu diketahui adalah sebagai berikut:

- $G = (V, A)$
- $V = \{v_0, v_1, v_2, \dots, v_n\}$ dimana v_0 menunjukkan depot dan v_1, \dots, v_n menunjukkan semua konsumen
- $A = \{(v_i, v_j) : i \neq j\}$ merupakan kumpulan *arcs* atau busur yang menghubungkan *vertex* v_i ke *vertex* v_j
- q_i menunjukkan permintaan dari konsumen $i, i \in V$
- d_{ij} menunjukkan jarak (*cost*) antara konsumen (*vertex*) v_i dan v_j
- $K = \{k_1, k_2, \dots, k_m\}$ mewakili armada kendaraan, m tidak lebih kecil dari m_{min} yang merupakan jumlah minimum kendaraan yang diperlukan untuk melayani semua konsumen, dimana nilai m_{min} dapat ditentukan dengan rumus $m_{min} = \lceil (\sum_{i=1}^n d_i / Q) \rceil$ dengan n merupakan jumlah konsumen (Ralphs, T.K., Kopman, L., Pulleyblank, W.R. and Trotter, L.E., 2003).
- Q merupakan kapasitas dari setiap kendaraan $k_i \in K$ (armada bersifat homogen).

Berdasarkan deskripsi di atas, permasalahan CVRP dapat dimodelkan secara matematis yang meliputi variabel keputusan, fungsi biaya, fungsi objektif, dan batasan-batasan.

Variabel Keputusan

Variabel keputusan digunakan untuk menentukan urutan kunjungan konsumen oleh sebuah armada kendaraan yang didefinisikan dalam bentuk persamaan (1) dan (2) sebagai berikut:

$$x_{i,j}^k = \begin{cases} 1, & \text{jika kendaraan } k \text{ mengunjungi} \\ & \text{konsumen } j \text{ secara langsung} \\ & \text{setelah } i \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$y_i^k = \begin{cases} 1, & \text{jika konsumen } i \text{ telah} \\ & \text{dilayani oleh kendaraan } k \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Persamaan (1) menunjukkan sebuah kondisi apakah armada kendaraan k melewati jalur dari konsumen i ke konsumen j , dimana untuk setiap busur (i, j) , $i \neq j$, $i \neq n + 1$, dan $j \neq 0$. Persamaan (2) menunjukkan bahwa apakah armada kendaraan k sudah melewati atau melayani konsumen i .

Fungsi Biaya

Fungsi biaya merupakan fungsi untuk menghitung total jarak dari solusi $X = (x_{ij}^k) \forall i, j \in V, k \in K$, yaitu:

$$Cost(X) = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij}^k \quad (3)$$

dimana jarak d_{ij} dapat dihitung dengan menggunakan fungsi *euclidean distance* dengan asumsi posisi konsumen i dan j berada pada ruang 2 dimensi dengan koordinat $i(x_1, y_1)$ dan $j(x_2, y_2)$, sehingga dapat dirumuskan *euclidean distance* dari d_{ij} adalah sebagai berikut:

$$d_{ij} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (4)$$

Fungsi Objektif

Fungsi objektif merupakan fungsi yang ingin dicapai. Pada kasus CVRP, fungsi objektifnya adalah meminimalkan fungsi biaya, yaitu:

$$\text{Min Cost}(X) = \text{Min} \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} d_{ij} x_{i,j}^k \quad (5)$$

Batasan-Batasan

Adapun batasan-batasan dalam kasus CVRP disajikan dalam persamaan (6) sampai dengan (11) sebagai berikut:

$$\sum_{k \in K} \sum_{j \in V} x_{i,j}^k = 1, \forall i \in V \quad (6)$$

$$\sum_{j \in V} x_{i,j}^k - \sum_{j \in V} x_{j,i}^k = 0, \forall i \in V, k \in K \quad (7)$$

$$\sum_{j \in V} x_{0,j}^k = 1, \forall k \in K \quad (8)$$

$$\sum_{j \in V} x_{j,n+1}^k = 1, \forall k \in K \quad (9)$$

$$x_{i,j}^k = 1 \Rightarrow y_i - q_j = y_j, \forall i, j \in V, k \in K \quad (10)$$

$$y_0 = Q, 0 \leq y_i, \forall i \in V \quad (11)$$

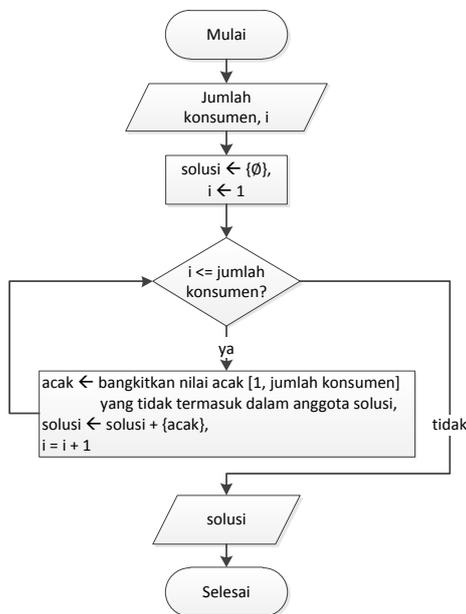
Persamaan (6) menyatakan batasan bahwa untuk setiap konsumen hanya dikunjungi tepat satu kali. Persamaan (7) menyatakan bahwa setelah kendaraan tiba pada konsumen j dari konsumen i maka selanjutnya berangkat untuk menuju tujuan lainnya. Persamaan (8) menyatakan bahwa semua armada kendaraan k berangkat dari depot. Persamaan (9) menyatakan bahwa pada akhirnya semua kendaraan harus kembali ke depot $n + 1$. Persamaan (10) menyatakan bahwa jika kendaraan k mengunjungi j dari i , maka muatan kendaraan k ketika sampai di konsumen j akan sama dengan muatan kendaraan k pada konsumen i dikurangi dengan permintaan konsumen j . Persamaan (11) menyatakan bahwa semua armada kendaraan pada saat berada di depot memiliki kapasitas muatan lebih kecil atau sama dengan Q dan lebih besar atau sama dengan 0 ketika armada kendaraan berada di konsumen i .

Pada penelitian ini, peneliti menggunakan algoritma *Tabu Search HGreX* yang merupakan kombinasi dari algoritma *Tabu Search* (Pirim, H.,

Bayraktar, E. and Eksioğlu, B., 2008) dan operator *HGreX Crossover* (Puljić, K. and Manger, R., 2013) untuk menyelesaikan kasus CVRP dengan dataset yang digunakan adalah Augerat (1995) kode A-n32-k5. Adapun langkah-langkah dalam algoritma *Tabu Search HGreX Crossover* meliputi: (1) Algoritma menerima masukan berupa koordinat konsumen (simpul), jumlah permintaan, jumlah kendaraan, dan maksimum iterasi yang digunakan; (2) Inisialisasi solusi awal S secara acak, menetapkan S sebagai solusi terbaik *bestSolusi*, evaluasi S menggunakan fungsi *cost* dan menetapkannya sebagai *bestCost*, dan menjadikan S sebagai anggota dari Tabu List *tabuList*; (3) Penetapan S sebagai *currBestSolusi* dan evaluasi *currBestSolusi* menggunakan fungsi *cost* dan menetapkannya sebagai *currBestCost*; (4) Pengecekan apakah *currBestSolusi* sama dengan *bestSolusi* atau tidak. Jika *currBestSolusi* sama dengan *bestSolusi*, maka dilakukan proses mutasi menggunakan operator *order based mutation* untuk melakukan perubahan solusi *currBestSolusi*. Jika *currBestSolusi* tidak sama dengan *bestSolusi*, maka ditetapkan *currCS* sebagai hasil dari *crossover* dengan menggunakan operator *HGreX Crossover* berdasarkan *bestSolusi* dan *currBestSolusi*; (5) Pengecekan apakah *currCS* lebih kecil atau sama dengan *currBestSolusi*. Jika *currCS* lebih kecil atau sama dengan *currBestSolusi*, maka ditetapkan *currCS* sebagai S , S sebagai *currBestSolusi*, dihitung nilai *currBestCost* menggunakan fungsi *cost* dari S . Jika *currCS* tidak lebih kecil atau sama dengan *currBestSolusi*, maka dilanjutkan dengan proses berikutnya; (6) Pembentukan *neighborhood* dari S ; (7) Penetapan nilai *currBestSolusi* sebagai S berdasarkan hasil penentuan solusi terbaik *neighborhood* dari S ; (8)

Penetapan S sebagai anggota dari Tabu List $tabuList$; (9) Pengecekan apakah $currBestCost$ lebih kecil dari $bestCost$. Jika $currBestCost$ lebih kecil dari $bestCost$, maka ditetapkan $currBestCost$ sebagai $bestCost$ dan $currBestSolusi$ sebagai $bestSolusi$. Jika $currBestCost$ tidak lebih kecil dari $bestCost$, maka dilakukan langkah ke-3.

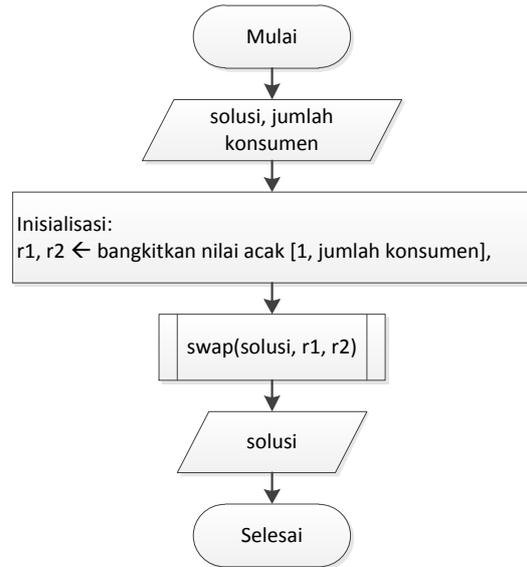
Pembentukan solusi awal pada langkah (2) dilakukan melalui proses membangkitkan nilai acak yang masing-masing nilai acaknya mewakili konsumen. Proses pembentukan solusi awal disajikan pada Gambar 3.



Gambar 3. Diagram alur pembentukan solusi awal

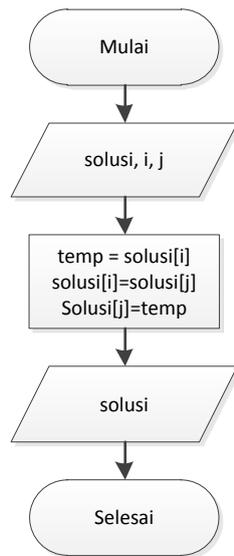
Proses mutasi sebuah solusi pada langkah (4) dilakukan dengan menggunakan teknik mutasi *interchanging* (Mohammed, M.A., Ahmad, M.S. and Mostafa, S.A., 2012) dengan masukan berupa sebuah solusi yang akan dimutasi dan jumlah konsumen. Proses mutasi terjadi diawali dengan membangkitkan 2 (dua) buah nilai acak yang berlainan dengan jangkauan [1, jumlah konsumen] untuk

menentukan konsumen mana yang akan ditukarkan. Selanjutnya, dilakukan proses penukaran 2 (dua) konsumen menggunakan teknik *swap* berdasarkan nilai acak yang telah dibangkitkan. Hasil dari proses mutasi ini adalah berupa sebuah solusi yang sudah mengalami perubahan posisi 2 konsumen. Adapun proses mutasi yang terjadi disajikan pada Gambar 4.



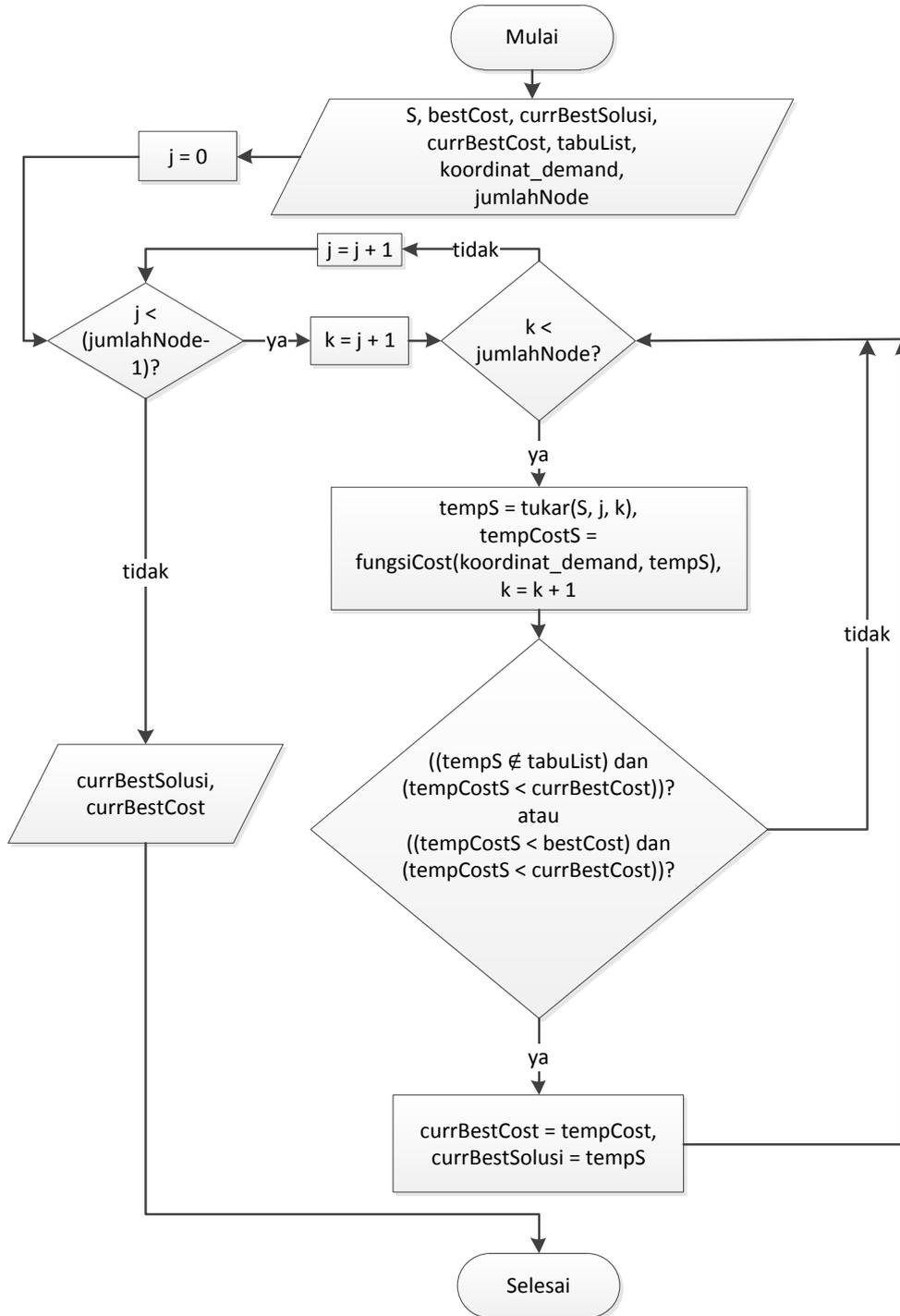
Gambar 4. Diagram alur proses mutasi

Pada proses mutasi, terdapat teknik *swap* yang digunakan untuk menukarkan posisi antara 2 (dua) konsumen dalam sebuah solusi, yaitu dengan cara dimulai dari menerima masukan berupa sebuah solusi dan 2 (dua) nilai acak yang diwakili oleh i dan j yang diperoleh dari proses sebelumnya pada Gambar 4. Selanjutnya dilakukan proses penukaran posisi antara 2 (dua) konsumen dalam sebuah solusi berdasarkan nilai i dan j . Hasil dari proses *swap* ini adalah berupa sebuah solusi yang sudah mengalami perubahan posisi 2 (dua) konsumen. Adapun ilustrasi dari teknik *swap* disajikan pada Gambar 5.



Gambar 5. Diagram alur proses swap

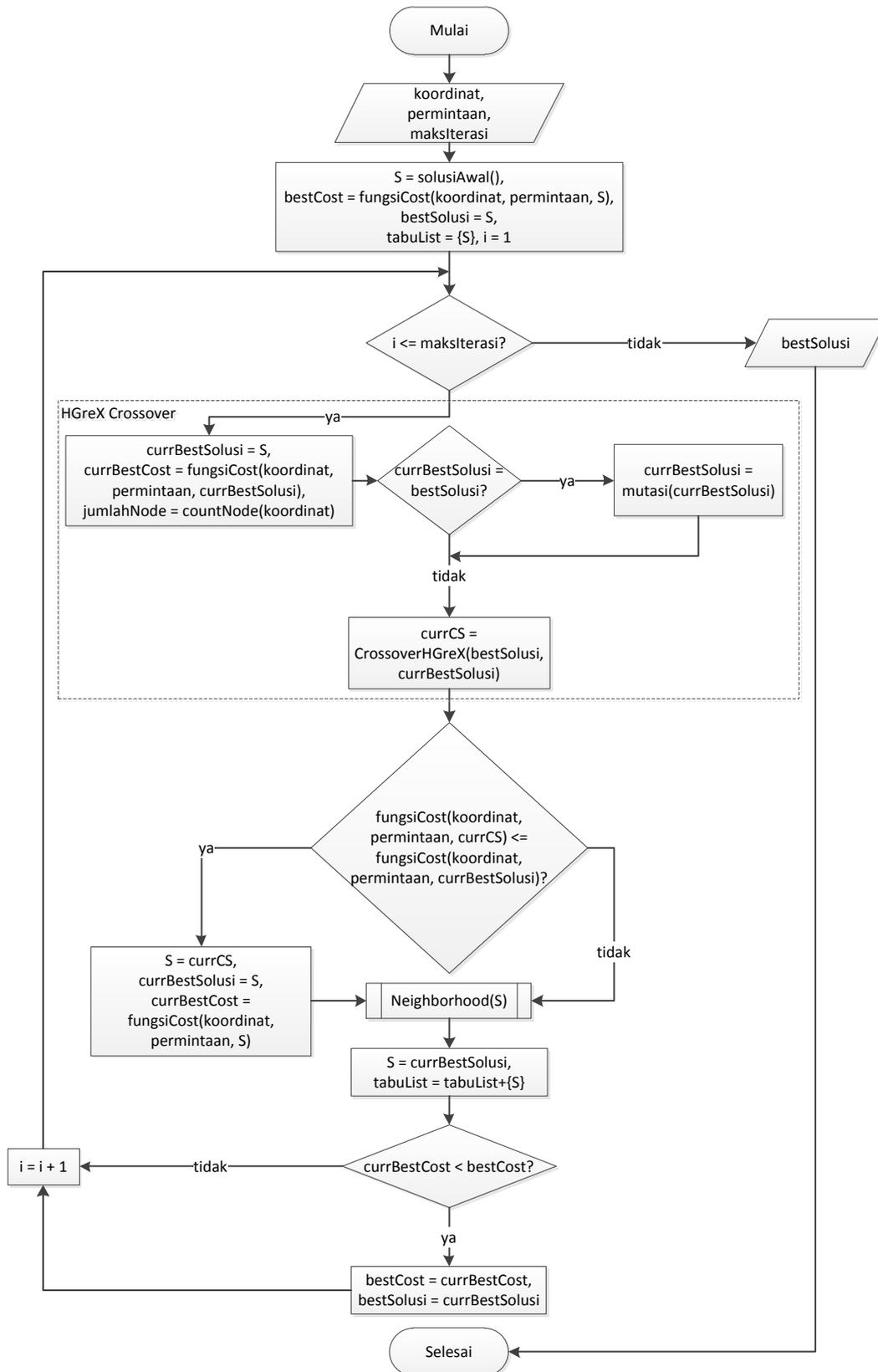
Pembentukan *neighborhood* pada langkah (6) dilakukan dengan menggunakan konsep permutasi, yaitu dengan menukarkan posisi 2 konsumen, dimana jumlah solusi tetangga yang dapat terbentuk sebanyak $\frac{n!}{2!(n-2)!}$ kombinasi, dimana n menunjukkan jumlah konsumen. Selanjutnya setiap solusi tetangga dievaluasi dengan menggunakan fungsi objektif untuk menentukan *currBestSolusi* dan *currBestCost*. Adapun diagram alur dari pembentukan *neighborhood* disajikan dalam flowchart pada Gambar 6.



Gambar 6. Diagram alur pembentukan neighborhood

Adapun keseluruhan proses algoritma *Tabu Search HGreX Crossover* untuk penyelesaian

Capacitated Vehicle Routing Problem (CVRP) disajikan pada Gambar 7.



Gambar 7. Diagram Alur Algoritma Tabu Search HGreX Crossover

HASIL DAN PEMBAHASAN

Hasil penelitian yang diperoleh diuraikan berdasarkan pencapaian tujuan utama penelitian, yaitu menerapkan algoritma Tabu Search yang telah dioptimasi dengan menerapkan operator HGreX Crossover untuk menyelesaikan Capacitated Vehicle Routing Problem, sehingga dapat meningkatkan kualitas solusi dan mengurangi waktu eksekusi.

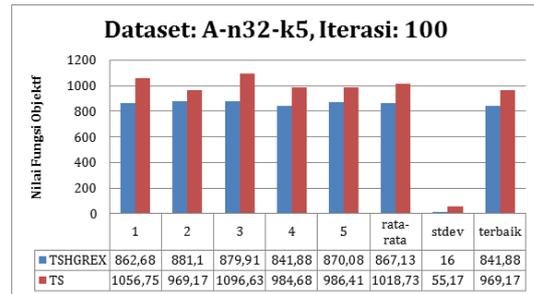
Skenario Pengujian

Skenario pengujian dilakukan dengan menerapkan algoritma *Tabu Search HGreX Crossover* terhadap 1 (satu) buah dataset uji sebanyak 5 kali untuk setiap jumlah iterasi [100; 500; 1000] dan dengan cara yang sama pula untuk algoritma *Tabu Search* konvensional, sehingga dapat diketahui kinerja dari algoritma *Tabu Search HGreX Crossover* terkait nilai fungsi objektif dan waktu eksekusi.

Hasil Pengujian

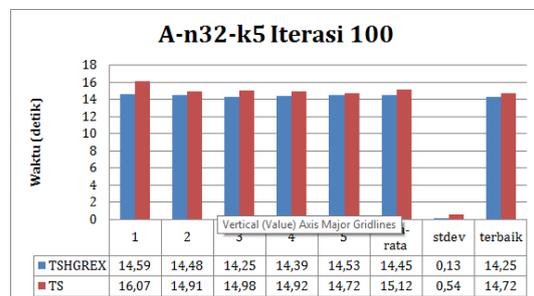
Hasil pengujian diperoleh dengan mengimplementasikan algoritma *Tabu Search HGreX Crossover* dan algoritma *Tabu Search* konvensional sebanyak 5 kali terhadap dataset Augerat (1995) dengan kode A-n32-k5 untuk parameter jumlah iterasi [100; 500; 1000].

Pada pengujian dengan jumlah iterasi sebesar 100 iterasi, hasil dari masing-masing algoritma untuk setiap pengujiannya tidak memiliki perbedaan yang signifikan. Hal ini dapat dilihat dari nilai standar deviasi yang kecil. Jika dibandingkan dengan algoritma *Tabu Search* konvensional, algoritma *Tabu Search HGreX Crossover* memberikan hasil yang lebih baik, dimana nilai fungsi objektif yang diperoleh lebih kecil dibandingkan dengan algoritma *Tabu Search* konvensional. Hasil dari pengujian ini disajikan pada Gambar 8.



Gambar 8. Hasil pengujian nilai fungsi objektif pada dataset A-n32-k5 dengan 100 iterasi

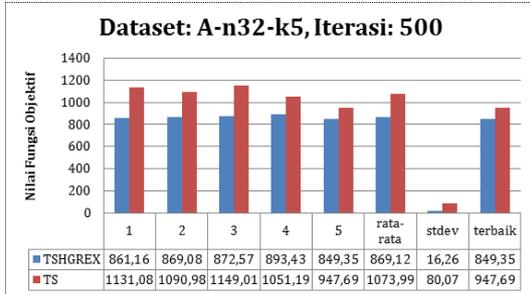
Berkaitan dengan waktu eksekusi dari algoritma *Tabu Search HGreX Crossover* dan *Tabu Search* konvensional untuk tiap percobaan dengan iterasi 100 iterasi yang dilakukan untuk hasil pengujian pada Gambar 8 tidak memiliki perbedaan yang signifikan. Namun jika dibandingkan dengan *Tabu Search* konvensional, *Tabu Search HGreX Crossover* memperoleh waktu eksekusi yang lebih cepat. Hasil pengujian ini disajikan pada Gambar 9.



Gambar 9. Hasil pengujian waktu pada dataset A-n32-k5 dengan 100 iterasi

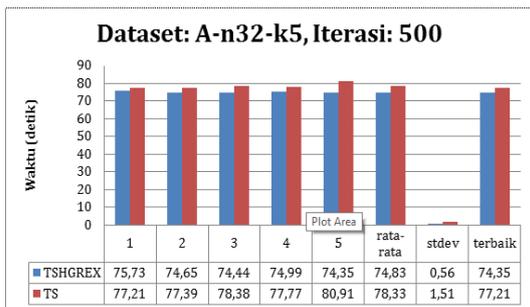
Pengujian selanjutnya dilakukan dengan jumlah iterasi 500 iterasi. Pada pengujian ini, hasil dari masing-masing algoritma untuk setiap pengujiannya juga tidak memiliki perbedaan yang signifikan. Hal ini dapat dilihat dari nilai standar deviasi yang tergolong kecil. Jika dibandingkan dengan algoritma *Tabu Search* konvensional, algoritma *Tabu Search HGreX Crossover* memberikan hasil yang lebih baik, dimana nilai fungsi objektif yang

diperoleh lebih kecil dibandingkan dengan algoritma *Tabu Search* konvensional. Hasil dari pengujian ini disajikan pada Gambar 10.



Gambar 10. Hasil pengujian nilai fungsi objektif pada dataset A-n32-k5 dengan 500 iterasi

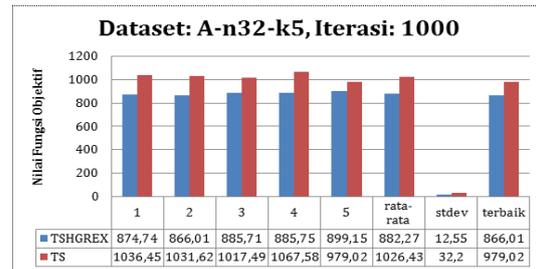
Waktu eksekusi yang diperoleh dari algoritma *Tabu Search HGreX Crossover* dan *Tabu Search* konvensional untuk tiap percobaan dengan jumlah iterasi 500 iterasi untuk hasil pengujian pada Gambar 10 juga tidak memiliki perbedaan yang signifikan. Namun jika dibandingkan dengan *Tabu Search* konvensional, *Tabu Search HGreX Crossover* memperoleh waktu eksekusi yang lebih cepat. Hasil pengujian ini disajikan pada Gambar 11.



Gambar 11. Hasil pengujian waktu pada dataset A-n32-k5 dengan 500 iterasi

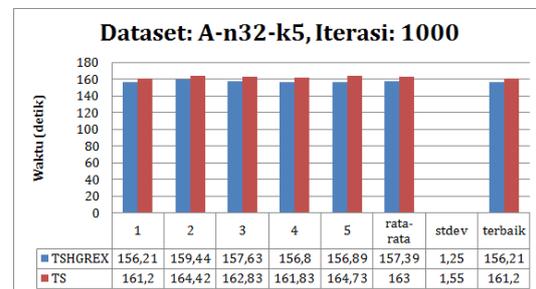
Pada pengujian dengan jumlah iterasi sebesar 1000 iterasi, hasil dari algoritma *Tabu Search* konvensional menunjukkan tidak terdapat perbedaan yang signifikan pada nilai fungsi objektif yang diperoleh untuk tiap percobaannya.

Hal ini dapat dilihat dari nilai standar deviasi yang kecil. Jika dibandingkan dengan algoritma *Tabu Search* konvensional, algoritma *Tabu Search HGreX Crossover* memberikan hasil yang lebih baik, dimana nilai fungsi objektif yang diperoleh lebih kecil dibandingkan dengan algoritma *Tabu Search* konvensional. Hasil dari pengujian ini disajikan pada Gambar 12.



Gambar 12. Hasil pengujian nilai fungsi objektif pada dataset A-n32-k5 dengan 1000 iterasi

Berdasarkan hasil pengujian pada Gambar 12, waktu eksekusi yang diperoleh untuk algoritma *Tabu Search HGreX Crossover* dan *Tabu Search* konvensional juga tidak memiliki perbedaan yang signifikan. Namun jika dibandingkan dengan *Tabu Search* konvensional, *Tabu Search HGreX Crossover* memperoleh waktu eksekusi yang lebih cepat. Hasil pengujian ini disajikan pada Gambar 13.



Gambar 13. Hasil pengujian waktu pada dataset A-n32-k5 dengan 1000 iterasi

Berdasarkan hasil pengujian secara keseluruhan untuk algoritma *Tabu Search*

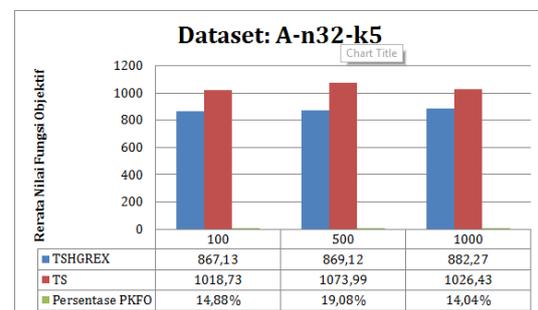
HGreX Crossover dan *Tabu Search* konvensional terhadap dataset Augerat (1995) dengan kode A-n32-k5, dapat diketahui bahwa jumlah iterasi yang diberikan dapat mempengaruhi hasil yang diperoleh terkait dengan nilai fungsi objektif dan waktu eksekusi. Selanjutnya untuk mengetahui kinerja dari algoritma *Tabu Search HGreX Crossover* yang diusulkan, dilakukan evaluasi terhadap nilai fungsi objektif dan waktu eksekusi yang diperoleh dengan cara membandingkan hasil pengujiannya dengan algoritma *Tabu Search* konvensional.

Evaluasi nilai fungsi objektif dilakukan untuk mengetahui apakah terdapat peningkatan atau penurunan kualitas nilai fungsi objektif (solusi) dari kasus CVRP yang diperoleh dengan menggunakan algoritma *Tabu Search HGreX Crossover* dibandingkan dengan menggunakan algoritma *Tabu Search* konvensional. Oleh karena fungsi objektif dari CVRP adalah meminimalkan total jarak tempuh dari seluruh rute yang diperoleh, maka jika nilai fungsi objektif yang diperoleh semakin kecil maka kualitas nilai fungsi objektif meningkat dan jika nilai fungsi objektif yang diperoleh semakin besar maka kualitas nilai fungsi objektif menurun. Persentase peningkatan atau penurunan kualitas ini dapat diketahui dengan mencari selisih dari nilai fungsi objektif yang diperoleh menggunakan algoritma *Tabu Search* konvensional dengan nilai fungsi objektif yang diperoleh menggunakan algoritma *Tabu Search HGreX Crossover* dan selanjutnya dibagi dengan nilai fungsi objektif yang diperoleh menggunakan algoritma *Tabu Search* konvensional dan selanjutnya dikalikan dengan 100%.

Pada penelitian ini, percobaan dilakukan berdasarkan parameter jumlah iterasi yang digunakan, yaitu 100 iterasi, 500 iterasi, dan 1000 iterasi. Masing-

masing iterasi dilakukan 5 (lima) kali percobaan, sehingga peningkatan atau penurunan kualitas nilai fungsi objektif diperoleh dengan mencari selisih dari rerata nilai fungsi objektif yang diperoleh menggunakan algoritma *Tabu Search* konvensional dengan rerata nilai fungsi objektif yang diperoleh menggunakan algoritma *Tabu Search HGreX Crossover* dan selanjutnya dibagi dengan rerata nilai fungsi objektif yang diperoleh menggunakan algoritma *Tabu Search* konvensional dan kemudian hasilnya dikalikan dengan 100%.

Hasil pengujian yang telah dilakukan terhadap dataset Augerat (1995) dengan kode A-n32-k5 menunjukkan bahwa algoritma *Tabu Search HGreX Crossover* mampu meningkatkan kualitas solusi sebesar 14.88% untuk 100 iterasi, 19.08% untuk 500 iterasi, dan 14.04% untuk 1000 iterasi jika dibandingkan dengan *Tabu Search* konvensional yang disajikan pada Gambar 14, dimana Persentase PKFO merupakan persentase peningkatan kualitas rerata nilai fungsi objektif yang diperoleh menggunakan algoritma *Tabu Search HGreX Crossover* dibandingkan dengan rerata nilai fungsi objektif yang diperoleh menggunakan algoritma *Tabu Search* konvensional.



Gambar 14. Perbandingan rerata nilai fungsi objektif setiap jumlah iterasi untuk dataset A-n32-k5

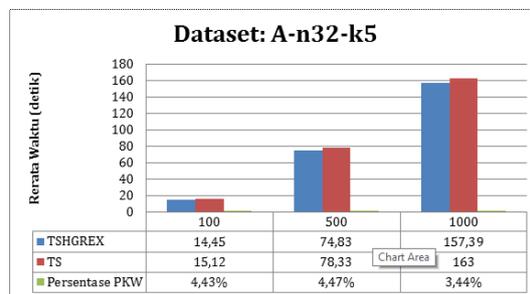
Evaluasi waktu eksekusi dilakukan untuk mengetahui apakah

terdapat peningkatan atau penurunan kualitas waktu eksekusi untuk memperoleh sebuah solusi dari kasus CVRP yang diperoleh dengan menggunakan algoritma *Tabu Search HGreX Crossover* dibandingkan dengan menggunakan algoritma *Tabu Search* konvensional. Kualitas waktu eksekusi akan semakin meningkat jika waktu eksekusi untuk memperoleh sebuah solusi dari kasus CVRP yang diperoleh dengan menggunakan algoritma *Tabu Search HGreX Crossover* lebih singkat dibandingkan dengan waktu eksekusi menggunakan algoritma *Tabu Search* konvensional. Sebaliknya, Kualitas waktu eksekusi akan semakin menurun jika waktu eksekusi untuk memperoleh sebuah solusi dari kasus CVRP yang diperoleh dengan menggunakan algoritma *Tabu Search HGreX Crossover* lebih lama dibandingkan dengan waktu eksekusi menggunakan algoritma *Tabu Search* konvensional. Persentase peningkatan atau penurunan kualitas ini dapat diketahui dengan mencari selisih dari waktu eksekusi yang diperoleh menggunakan algoritma *Tabu Search* konvensional dengan waktu eksekusi yang diperoleh menggunakan algoritma *Tabu Search HGreX Crossover* dan selanjutnya dibagi dengan waktu eksekusi yang diperoleh menggunakan algoritma *Tabu Search* konvensional dan selanjutnya dikalikan dengan 100%.

Pada penelitian ini, percobaan dilakukan berdasarkan parameter jumlah iterasi yang digunakan, yaitu 100 iterasi, 500 iterasi, dan 1000 iterasi. Masing-masing iterasi dilakukan 5 (lima) kali percobaan, sehingga peningkatan atau penurunan kualitas waktu eksekusi diperoleh dengan mencari selisih dari rerata waktu eksekusi yang diperoleh menggunakan algoritma *Tabu Search* konvensional dengan rerata waktu eksekusi yang diperoleh menggunakan

algoritma *Tabu Search HGreX Crossover* dan selanjutnya dibagi dengan rerata waktu eksekusi yang diperoleh menggunakan algoritma *Tabu Search* konvensional dan kemudian hasilnya dikalikan dengan 100%.

Hasil pengujian yang telah dilakukan terhadap dataset Augerat (1995) dengan kode A-n32-k5 menunjukkan bahwa algoritma *Tabu Search HGreX Crossover* mampu meningkatkan kualitas waktu eksekusi sebesar 4.43% untuk 100 iterasi, 4.47% untuk 500 iterasi, dan 3.44% untuk 1000 iterasi jika dibandingkan dengan *Tabu Search* konvensional yang disajikan pada Gambar 15, dimana Persentase PKW merupakan persentase peningkatan kualitas rerata waktu eksekusi yang diperoleh menggunakan algoritma *Tabu Search HGreX Crossover* dibandingkan dengan rerata waktu eksekusi yang diperoleh menggunakan algoritma *Tabu Search* konvensional.



Gambar 15. Perbandingan rerata waktu eksekusi setiap jumlah iterasi untuk dataset A-n32-k5

SIMPULAN

Berdasarkan pada pembahasan yang telah diuraikan sebelumnya, maka dapat diambil kesimpulan bahwa (1) Penerapan operator *HGreX Crossover* pada algoritma *Tabu Search* berdasarkan jarak dan waktu pada keseluruhan pengujian mendapatkan hasil yang lebih baik dibandingkan dengan algoritma *Tabu Search* konvensional; (2) Hasil pengujian pada dataset Augerat dengan

kode A-n32-k5 menunjukkan bahwa penggunaan algoritma *Tabu Search HGreX Crossover* lebih baik dibandingkan dengan algoritma *Tabu Search* konvensional, baik dari kualitas nilai fungsi objektif (solusi) maupun waktu eksekusinya.

DAFTAR PUSTAKA

- Baldacci, R., Mingozzi, A. and Roberti, R., 2012, Recent Exact Algorithms for solving the Vehicle Routing Problem under Capacity and Time Window Constraints. *European Journal of Operational Research*, 218, 1, 1–6.
- Dantzing, G.B. and Ramser, J.H., 1959, The Truck Dispatching Problem. *Management Science*, 6, 1, 80–91.
- El Hassani, A.H., Bouhafs, L. and Koukam, A., 2008, A Hybrid Ant Colony System Approach for the Capacitated Vehicle Routing Problem and the Capacitated Vehicle Routing Problem with Time Windows.
- Guan, C., Cao, Y. and Shi, J., 2010, A tabu search algorithm for the vehicle routing problem.
- Jia, H., Li, Y., Dong, B. and Ya, H., 2013, An Improved Tabu Search Approach to Vehicle Routing Problem. *Procedia - Social and Behavioral Sciences*, 96, Cictp, 1208–1217.
- Mohammed, M.A., Ahmad, M.S. and Mostafa, S.A., 2012, Using Genetic Algorithm in Implementing Capacitated Vehicle Routing Problem. 2012 International Conference on Computer & Information Science (ICCIS), 257–262.
- Pirim, H., Bayraktar, E. and Eksioğlu, B., 2008, Tabu Search: A Comparative Study. , October, 1–29.
- Puljić, K. and Manger, R., 2013, Comparison of eight evolutionary crossover operators for the vehicle routing problem. *Mathematical Communications*, 18, 359–375.
- Ralphs, T.K., Kopman, L., Pulleyblank, W.R. and Trotter, L.E., 2003, On the capacitated vehicle routing problem. *Mathematical Programming*, 94, 2-3, 343–359.
- Toth, P. and Vigo, D., 2002a, Models, Relaxations and Exact Approaches for the Capacitated Vehicle Routing Problem. *Discrete Applied Mathematics*, 123, 1-3, 487–512.
- Toth, P. and Vigo, D., 2002b, *The Vehicle Routing Problem*. SIAM, Philadelphia, USA.