

Penyelesaian *Colored Traveling Salesman Problem* Menggunakan Algoritma Genetika *Hill-Climbing*

Fakhrana Nadhilah*, Khusnul Novianingsih dan Kartika Yulianti

Departemen Pendidikan Matematika
Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam
Universitas Pendidikan Indonesia
*Surel : fakhnanadhilah@yahoo.com

ABSTRAK. *Colored Traveling Salesman Problem* (CTSP) adalah pengembangan dari MTSP dimana terdapat dua wilayah kerja yaitu wilayah umum yang dapat dikunjungi oleh setiap pekerja, dan wilayah pribadi yang berlaku hanya untuk pekerja yang ditugaskan di wilayah tersebut. Pada CTSP rute dari beberapa pekerja akan dibagi dengan mempertimbangkan wilayah umum dan wilayah pribadinya. Pada kajian ini, CTSP diselesaikan dengan Algoritma Genetika *Hill-Climbing*, yang merupakan penggabungan dari Algoritma Genetika dengan Algoritma *Hill-Climbing* dengan tujuan menghasilkan solusi yang lebih baik. Selanjutnya, model CTSP menggunakan Algoritma Genetika *Hill-Climbing* diimplementasikan pada kasus pengumpulan paket suatu perusahaan ekspedisi di Kota Bandung. Hasil dari kajian ini yaitu diperoleh rute terpendek untuk kasus pengumpulan paket suatu perusahaan ekspedisi. Selain itu, dengan membandingkan Algoritma Genetika *Hill-Climbing* dengan Algoritma Genetika Klasik, diperoleh hasil bahwa Algoritma Genetika *Hill-Climbing* memberikan solusi dengan jarak yang lebih pendek meskipun membutuhkan waktu komputasi yang lebih lama.

Kata Kunci: *Multiple Traveling Salesman Problem*, Algoritma Genetika, *Hill-Climbing*

TITLE. *Settlement of colored traveling salesman problem using genetic algorithm hill-climbing*

ABSTRACT. *Colored Traveling Salesman Problem* (CTSP) is developed from MTSP that have two types of workspaces that called as shared workspace that can be visited by all workers, and exclusive workspace that only can be visited by assigning worker. The route at CTSP will be divided by considering the shared workspace and exclusive workspace. In this study, the *Hill-Climbing Genetic Algorithm*

is used to solve the CTSP. Hill-Climbing Genetic Algorithm is optimization methods that combining Genetic Algorithm and Hill-Climbing Algorithm to result better solution. Then, CTSP using the Hill-Climbing Genetic Algorithm is implemented to package collection at one of shipping company in Bandung. Based on the computational results, we found that the Hill-Climbing Genetic Algorithm gives better solution then Classic Genetic Algorithm although it takes longer computing time.

Keywords: *Multiple Traveling Salesman Problem, Genetic Algorithm, Hill-Climbing*

1. PENDAHULUAN

Multiple Traveling Salesman Problem (MTSP) adalah perluasan dari *Traveling Salesman Problem* (TSP) yang merupakan salah satu masalah optimisasi kombinatorik yang bertujuan untuk membagi rute beberapa pekerja atau subjek agar tidak terjadi pertumpukan pada salah satu lokasi kerja atau objek yang sama, atau dua subjek bekerja pada objek yang sama. Li [1] telah mengembangkan permasalahan MTSP yaitu *Colored Traveling Salesman Problem* (CTSP). Pada CTSP setiap rute pekerja dinyatakan oleh sebuah warna agar tidak terjadi tabrakan atau tumpang tindih dalam pembagian rute.

Li [1] menerapkan empat algoritma genetika untuk menyelesaikan masalah CTSP ini yaitu Algoritma Genetika Klasik, Algoritma Genetika *Greedy*, Algoritma Genetika *Hill-Climbing* dan Algoritma Genetika *Simulated Annealing*. Li [1] menyatakan bahwa Algoritma Genetika *Simulated Annealing* memberikan hasil yang paling optimal yaitu rute dengan total jarak terpendek namun komputasinya membutuhkan waktu yang cukup lama, sedangkan Algoritma Genetika *Hill-Climbing* memberikan hasil yang optimal dengan membutuhkan waktu lebih sedikit.

Algoritma Genetika *Hill-Climbing* adalah pengembangan dari Algoritma Genetika dengan menambahkan Algoritma *Hill-Climbing* dalam tahapannya. Algoritma Genetika adalah salah satu algoritma yang sering digunakan dalam penyelesaian masalah optimisasi kombinatorik karena meskipun dalam proses Algoritma Genetika menggunakan pencarian acak, namun Algoritma Genetika menggunakan teori-teori evolusi untuk memperoleh solusi yang lebih baik. Pada Algoritma Genetika *Hill-Climbing*, solusi-solusi yang dibangkitkan akan melewati tahap Algoritma *Hill-Climbing* sebelum memasuki tahapan dalam Algoritma Genetika Klasik seperti Seleksi, *Crossover* dan Mutasi. Algoritma *Hill-Climbing* mencari solusi yang mungkin dan memiliki hasil yang lebih optimal dibandingkan dengan solusi acak yang dibangkitkan. Kemudian solusi baru yang telah terpilih melalui Algoritma *Hill-Climbing* memasuki proses Seleksi, *Crossover* dan Mutasi. Algoritma *Hill-Climbing* pada Algoritma Genetika *Hill-Climbing* digunakan dengan tujuan agar solusi yang dihasilkan lebih cepat mencapai konvergen karena solusi-solusi yang digunakan dalam tahap Seleksi merupakan solusi yang terpilih.

Pada makalah ini *Colored Traveling Salesman Problem* diselesaikan menggunakan Algoritma Genetika *Hill-Climbing*. Berbeda dengan Li [1] yang tidak mengimplementasikan masalah *Colored Traveling Salesman* pada masalah nyata, pada penelitian hal tersebut diimplementasikan pada

penyelesaian masalah pengumpulan paket dari suatu perusahaan ekspedisi di Bandung.

2. METODOLOGI

2.1 Model CTSP

Berdasarkan Li [1], pada model optimisasi *Colored Traveling Salesman Problem* digunakan graf $G = (V, E)$ dengan V adalah himpunan simpul dan E adalah himpunan sisi. Simpul menyatakan kota, dan sisi menyatakan jalan yang menghubungkan dua buah kota. Kemudian dicari rute dengan jarak terpendek dengan m -pekerja dan n -kota, C_{ij} merupakan nilai bobot dari kota i ke kota j . Himpunan P_k berisi kota-kota dari wilayah pribadi pekerja k dan U merupakan himpunan kota-kota dari wilayah umum. Fungsi tujuan dari CTSP yaitu:

$$f = \sum_{k=1}^m \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} C_{ij} x_{ijk} \quad (1)$$

Kendala dari model CTSP yaitu:

1. Setiap pekerja berangkat dari depot dan kembali pada depot. Kendala ini diekspresikan sebagai:

$$\sum_{j=1}^{n-1} x_{0jk} = 1, k \in Z_m \quad (2)$$

$$\sum_{i=1}^{n-1} x_{i0k} = 1, k \in Z_m \quad (3)$$

2. Setiap kota selain depot harus dikunjungi tepat satu kali. Kendala ini diekspresikan sebagai:

$$\sum_{j=0}^{n-1} \sum_{k=1}^m x_{ijk} = 1, i \neq j, i \in V \setminus \{0\} \quad (4)$$

$$\sum_{j=0}^{n-1} \sum_{k=1}^m x_{jik} = 1, i \neq j, i \in V \setminus \{0\} \quad (5)$$

3. Pada CTSP wilayah kerja terbagi menjadi dua, salah satunya yaitu wilayah pribadi yang hanya boleh dikunjungi oleh pekerja yang ditugaskan, sehingga pekerja k tidak boleh mengunjungi wilayah pribadi milik pekerja lain dari wilayah pribadinya, dan pekerja selain pekerja k tidak boleh mengunjungi wilayah pribadi pekerja k yang tersisa. Kendala ini diekspresikan sebagai:

$$\sum_i \sum_j x_{ijk} = 0, i \in P_k, j \in V \setminus (U \cup P_k), k \in Z_m \quad (6)$$

$$\sum_i \sum_j x_{jik} = 0, i \in P_k, j \in V \setminus (U \cup P_k), k \in Z_m \quad (7)$$

4. Pekerja $l (l \neq k)$ tidak boleh berangkat dari ataupun kembali ke wilayah pribadi pekerja k . Kendala ini diekspresikan sebagai:

$$\sum_i \sum_j x_{ijl} = 0, i \neq j, k \neq l, i \in P_k, j \in V, l \in Z_m \quad (8)$$

$$\sum_i \sum_j x_{jil} = 0, i \neq j, k \neq l, i \in P_k, j \in V, l \in Z_m \quad (9)$$

5. Wilayah umum merupakan wilayah yang dapat dikunjungi oleh setiap pekerja. Seorang pekerja dapat mengunjungi wilayah umum dan sepasang akses untuk masuk dan keluar jika dibutuhkan. Kendala ini diekspresikan sebagai:

$$\sum_l x_{jlk} = \sum_i x_{ijk}, i \neq j \neq l, j \in U, l \in P_k \cup U \quad (10)$$

6. Rute dari setiap pekerja harus terhubung dan tidak boleh ada sub-rute yang tidak terhubung pada kota-kota $V \setminus \{0\}$ sehingga

$$(u_{ik} - u_{jk} + n) \times x_{ijk} \leq n - 1, j \neq i, i, j \in V \setminus \{0\}, k \in Z_m \quad (11)$$

dengan u_{ik} adalah banyaknya kota yang dikunjungi pekerja k sebelum mencapai kota i dan u_{jk} adalah banyaknya kota yang dilewati pekerja k sebelum mencapai kota j .

2.2 Algoritma Genetika *Hill-Climbing*

Colored Traveling Salesman Problem merupakan salah satu permasalahan optimasi kombinatorial yang termasuk sebagai *NP-Hard Problem*. Permasalahan ini memiliki banyak kemungkinan solusi dan membutuhkan waktu yang lama dalam penyelesaiannya agar menemukan solusi yang optimal, sehingga dibutuhkan metode penyelesaian alternatif agar permasalahan ini dapat diselesaikan dengan cepat.

Pada makalah ini, model CTSP pada persamaan (1) hingga (11) diselesaikan dengan menggunakan Algoritma Genetika *Hill-Climbing*. Algoritma Genetika *Hill-Climbing* adalah sebuah metode yang telah dikembangkan dari Algoritma Genetika Klasik, dengan menyisipkan teknik *Hill-Climbing* dengan tujuan lebih mengoptimalkan solusi.

Contoh representasi kromosom pada individu, misalkan terdapat 28 kota dimana 7 kota berada di masing-masing 3 wilayah pribadi dan 6

kota berada di wilayah umum dan 1 kota sebagai depot dengan rincian sebagai berikut:

- Depot = {0}
- Wilayah Umum = {1,2,3,4,5,6}
- Wilayah Pribadi 1 = {7,8,9,10,11,12,13}
- Wilayah Pribadi 2 = {14,15,16,17,18,19,20}
- Wilayah Pribadi 3 = {21,22,23,24,25,26,27}

Kromosom 1	8	2	7	3	10	11	9	12	13
Kromosom 2	1	14	15	19	17	20	16	18	5
Kromosom 3	24	27	23	25	22	21	26	4	6

Gambar 1. Contoh Individu

Nilai *fitness* mengukur seberapa optimal individu yang dibangkitkan. Pada permasalahan CTSP semakin tinggi nilai *fitness* maka semakin baik solusinya. Fungsi *fitness* yang digunakan yaitu

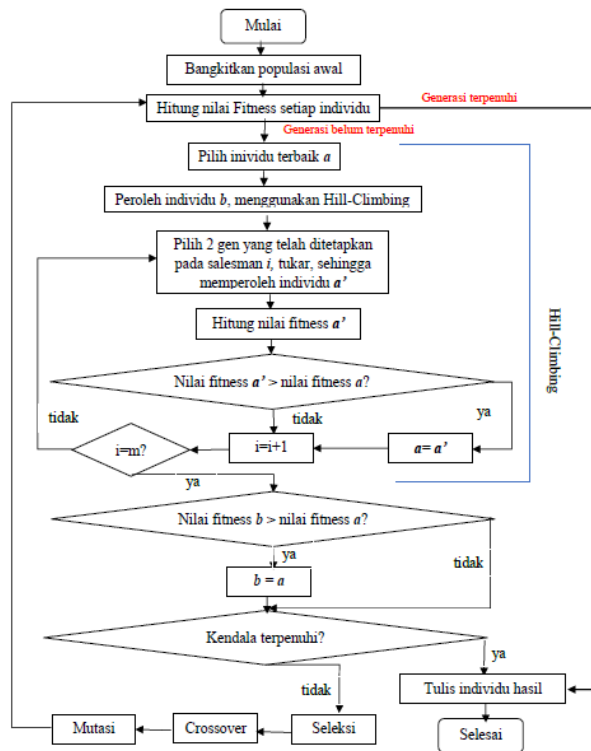
$$F(x) = \frac{1}{f(x)}$$

dimana $F(x)$ adalah fungsi *fitness* individu x dan $f(x)$ adalah fungsi tujuan dari CTSP.

Algoritma *Hill-Climbing* diterapkan pada masing-masing individu untuk menghasilkan individu baru. Langkah-langkah dari Algoritma *Hill-Climbing* sebagai berikut:

1. Tetapkan $a = 1$, dan $i = 1$.
2. Jika $i < m$ maka lakukan langkah 3. Jika $i = m$ maka $a = a + 1$ kemudian kembali ke langkah 3. Jika $a = p$ maka berhenti.
3. Pilih 2 gen (kota) yang telah ditugaskan pada pekerja i , dari kromosom kota pada individu a . Tukar kedua gen tersebut kemudian diperoleh individu a' .
4. Periksa apakah nilai *fitness* dari a' lebih besar dari a . Jika ya maka $a = a'$, sebaliknya maka $a = a$.
5. Tetapkan $i = i + 1$ dan kembali ke langkah 2.

Selanjutnya individu baru memasuki tahap seleksi menggunakan metode *Roulette Wheel*, *crossover* dengan *Order Crossover (OX)* dan mutasi menggunakan *Swapping Mutation*.



Gambar 2. Flowchart Algoritma Genetika Hill-Climbing

3. HASIL DAN PEMBAHASAN

Model CTSP dan penyelesaiannya menggunakan Algoritma Genetika *Hill-Climbing* diimplementasikan pada kasus pengumpulan paket. Data-data yang dibutuhkan untuk menyelesaikan kasus yaitu:

1. Data lokasi *outlet*

Data lokasi *outlet* menyatakan lokasi-lokasi *outlet* yang berada di wilayah Bandung. Terdapat 66 *outlet* yang terletak di wilayah Bandung dan terbagi menjadi 6 wilayah dengan 1 wilayah sebagai wilayah umum dan 5 wilayah pribadi. Pada wilayah umum atau wilayah Karees terdapat 1 *outlet* sebagai depot dan 12 *outlet*. Sedangkan pada wilayah pribadi yaitu: 13 *outlet* terletak pada wilayah Bojonagara, 13 *outlet* terletak pada wilayah Cibeunying, 10 *outlet* terletak pada wilayah Gedebage, 13 *outlet* terletak pada wilayah Tegallega, dan 4 *outlet* terletak pada wilayah Ujung Berung.

2. Data jarak *outlet*

Data jarak *outlet* menyatakan besarnya jarak dari tiap-tiap *outlet* ke *outlet* lainnya baik yang dalam satu wilayah maupun berbeda wilayah. Data

ini dibutuhkan untuk menghitung keoptimalannya dengan mencari rute jarak terdekat. Data diperoleh dengan menggunakan aplikasi *Google Maps*.

3. Data kendaraan

Terdapat 5 kendaraan yang mengangkut paket dari setiap wilayah. Diasumsikan satu kendaraan cukup untuk mengumpulkan paket dari satu rute yang telah ditentukan.

Berikut adalah tahapan-tahapan Algoritma Genetika *Hill-Climbing* untuk penyelesaian masalah pengumpulan paket.

1) Pembangkitan Populasi Awal

Pada tahap awal dibangkitkan individu secara acak sesuai dengan jumlah populasi yang telah ditentukan. Selanjutnya ditentukan banyaknya generasi, probabilitas *crossover* dan probabilitas mutasi.

2) Nilai *Fitness*

Setiap individu dihitung nilai *fitness*, yaitu total jarak yang dibutuhkan individu untuk menjalankan rute. Kemudian periksa parameter generasi dari Algoritma Genetika. Apabila generasi belum terpenuhi maka lanjutkan ke tahap *Hill-Climbing*.

3) Algoritma *Hill-Climbing*

Tahap selanjutnya diterapkan Algoritma *Hill-Climbing* pada setiap individu. Langkah-langkah *Hill-Climbing* yang digunakan telah dijelaskan pada sub bab METODOLOGI.

4) Seleksi

Pada tahapan ini dilakukan seleksi dengan menggunakan metode *Roulette Wheel*. Langkah-langkah metode *Roulette Wheel* yang digunakan dapat dilihat pada [2].

5) *Crossover*

Tahapan selanjutnya yaitu dilakukan *crossover* pada individu yang terpilih dengan menggunakan probabilitas *crossover*. Metode *crossover* yang digunakan yaitu *Order Crossover (OX)*. Langkah-langkah metode *Order Crossover* yang digunakan dapat dilihat pada [2].

6) Mutasi

Mutasi dilakukan dengan menggunakan metode *Swapping Mutation*. Mutasi dilakukan pada gen yang dipilih menggunakan probabilitas mutasi. Langkah-langkah metode *Swapping Mutation* yang digunakan dapat dilihat pada [2].

7) Evaluasi

Pada tahap evaluasi diperiksa parameter generasi Algoritma Genetika. Apabila telah dipenuhi maka tahap selanjutnya yaitu menghitung nilai *fitness* tiap individu akhir. Kemudian individu dengan nilai *fitness*

paling tinggi terpilih sebagai solusi optimal dari Algoritma Genetika *Hill-Climbing*.

Tabel 1 menampilkan hasil dari pembagian rute pengumpulan paket menggunakan Algoritma Genetika *Hill-Climbing*. Dengan nilai parameter populasi = 40, probabilitas *crossover* = 0,75, probabilitas mutasi = 0,1, dan banyaknya generasi = 100, diperoleh total jarak minimum sebesar 194,31 km dengan rute sebagai berikut:

Tabel 1. Rute Terpendek Solusi Algoritma Genetika *Hill-Climbing*

Kendaraan	Rute	Total Jarak
1	Depot – Sudirman – Gunung Batu – Dakota – Sarijadi – Sutami – Mitra Kusumah Sejahtera – Barokah Gerong Hilir – Formula – Abank Cell – Setiabudhi – Dian Kreasindo – Waspada – Victori 1 – Depot	33,85
2	Depot – Griya Muslim – Felix – CV.KYMTECH SOLUSINDO – Ciumbuleuit 911 – Erisandev – Sam’s Tour – Dago Pojok – Sukaluyu – Agata Tour Travel – Apartemen Gateway Ahmad Yani – Cikutra Barat – DS_Nusantara Ekspedisi – Simply Fresh Laundry Cikutra – Depot	43
3	Depot – Rekana – Im & Co Rancabolang – Merkuri Utara IV – Boiz – Saturnus – Pluto – CV. Gatra Cipta Karya – Bahagia Permai – Sekelimus – Warbud – Agi Fresha – Batununggal – Pasirluyu – Depot	31,95
4	Depot – Rejeki Jaya Toys - Holis 32 – Cijerah Hilir – Holis Regency – Caringin 32 – Kopo 235 – Orens – Leuwi Panjang 52 – Mekar Wangi – Mekar Kencana – JoandFashion Holis – Rajawali – Otista – Depot	33,45
5	Depot – Turangga – Naripan – Gasmin – Buana Studio – Sariwates – Sri Ayu – Victori 2 – Puri Dago – BT-9 – Junior JR – Sukanegara – Lengkong Besar – Kejaksaan - Depot	52,06
Total Jarak		194,31

Kemudian solusi hasil dan waktu komputasi dari Algoritma Genetika *Hill-Climbing* dibandingkan dengan Algoritma Genetika Klasik. Parameter-parameter yang digunakan yaitu populasi = 40, probabilitas *crossover* = 0,75, probabilitas mutasi = 0,1, dan banyaknya generasi = 100. Berdasarkan Tabel 2 diperoleh hasil bahwa Algoritma Genetika *Hill-Climbing* memberikan solusi dengan jarak yang lebih pendek jika dibandingkan dengan Algoritma Genetika Klasik, tetapi membutuhkan waktu komputasi yang lebih lama.

Tabel 2. Perbandingan Solusi Algoritma Genetika *Hill-Climbing* dan Algoritma Genetika

	Total Jarak	Waktu
Algoritma Genetika <i>Hill-Climbing</i>	194,310	1667,459
Algoritma Genetika Klasik	352,300	1478,448

4. KESIMPULAN

CTSP dapat dimodelkan sebagai model *integer programming* dengan fungsi tujuan untuk meminimumkan jarak terpendek dari rute perjalanan beberapa orang pekerja jika terdapat wilayah umum dan wilayah pribadi.

Algoritma Genetika *Hill-Climbing* merupakan Algoritma Genetika Klasik yang menambahkan Algoritma *Hill-Climbing*. Algoritma ini bekerja dengan cara merepresentasikan kromosom membangkitkan populasi, menghitung nilai *fitness*, menerapkan Algoritma *Hill-Climbing*, proses seleksi, *crossover* dan mutasi.

Model CTSP dengan Algoritma Genetika *Hill-Climbing* berhasil diimplementasikan pada permasalahan pengumpulan paket suatu perusahaan ekspedisi dengan total jarak solusi optimal sebesar 194,31 km. Hasil komputasi menunjukkan bahwa Algoritma Genetika *Hill-Climbing* menghasilkan solusi lebih baik pada penyelesaian kasus pengumpulan paket dengan total jarak yang lebih pendek dibandingkan dengan Algoritma Genetika Klasik meskipun waktu komputasi yang dibutuhkan lebih lama.

5. DAFTAR PUSTAKA

- [1] Li, J., Zhou, M., Sun, Q., Dai, X., & Yu, X. *Colored Traveling Salesman Problem. IEEE* (2014): 2168-2267.
- [2] Martiana, Entin. (2011). *Bab 7 Algoritma Genetika*. [Online]. Diakses dari <http://entin.lecturer.pens.ac.id/Kecerdasan%20Buatan/Buku/Bab%207%20Algoritma%20Genetika.pdf>