

## KOMPARASI ALGORITMA *CHEAPEST INSERTION HEURISTIC* (CIH) DAN *GREEDY* DALAM OPTIMASI RUTE PENDISTRIBUSIAN BARANG

L. Virginayoga Hignasari<sup>1</sup>

<sup>1</sup>Program Studi Teknik Industri, Fakultas Teknik, Universitas Mahendradatta Bali  
Jl. Ken Arok No 12, Peguyangan Denpasar Utara, Bali 80115  
Email : ginahignasari@gmail.com

**Abstrak** - Penelitian ini bertujuan untuk membandingkan algoritma yang secara efektif dapat memberikan solusi lebih baik terkait masalah penentuan rute terpendek dalam distribusi barang. Penelitian ini merupakan penelitian kualitatif. Objek penelitian adalah rute pengiriman barang sebuah usaha yang bergerak di bidang printing dan konveksi. Algoritma yang dibandingkan dalam penelitian ini adalah algoritma *Cheapest Insertion Heuristic* (CIH) dan *Greedy*. Kedua algoritma memiliki kelebihan dan kekurangan dalam pencarian rute terpendek. Dari hasil analisis menggunakan kedua algoritma tersebut, algoritma *Cheapest Insertion Heuristic* (CIH) dan *Greedy* mampu memberikan hasil optimasi yang hampir sama. Yang membedakan hanyalah pemilihan rute perjalanannya saja. Adapaun kelebihan dari algoritma *Greedy* adalah langkah perhitungannya yang lebih sederhana daripada algoritma *Cheapest Insertion Heuristic* (CIH). Sedangkan kekurangan dari algoritma *Greedy* adalah kurang tepat digunakan untuk mencari rute terpendek dengan jumlah tempat yang dikunjungi relatif banyak. Kelebihan algoritma *Cheapest Insertion Heuristic* (CIH) adalah algoritma ini masih stabil digunakan untuk jumlah tempat yang dikunjungi relatif banyak. Sedangkan kekurangan algoritma *Cheapest Insertion Heuristic* (CIH) adalah prinsip penghitungannya yang rumit dan relatif lebih lama daripada algoritma *Greedy*.

**Kata Kunci** : *cheapest insertion heuristic, greedy, optimasi, traveling salesman problem*

**Abstract** - This study was aimed to compare algorithms that can effectively provide better solutions related to the problem of determining the shortest route in the distribution of goods. This research was a qualitative research. The object of research was the route of shipping goods of a business that is engaged in printing and convection. The algorithms compared in this study were *Cheapest Insertion Heuristic* (CIH) and *Greedy* algorithms. Both algorithms have advantages and disadvantages in finding the shortest route. From the results of the analysis using these two algorithms, the *Cheapest Insertion Heuristic* (CIH) and *Greedy* algorithm can provide almost the same optimization results. The difference was only the selection of the journey. The strength of the *Greedy* algorithm was that the calculation steps are simpler than the *Cheapest Insertion Heuristic* (CIH) algorithm. While the disadvantage of the *Greedy* algorithm was that it is inappropriate to find the shortest route with a relatively large number of places visited. The advantage of the *Cheapest Insertion Heuristic* (CIH) algorithm was that this algorithm is still stable, used for the relatively large number of places visited. While the lack of *Cheapest Insertion Heuristic* (CIH) algorithm was a complicated principle of calculation and was relatively longer than the *Greedy* algorithm.

**Keywords**: *cheapest insertion heuristic, greedy, optimization, traveling salesman problem*

### PENDAHULUAN

Penentuan lintasan pada proses distribusi barang merupakan salah satu masalah yang menjadi konsen di setiap perusahaan yang bergerak di bidang distribusi. Hal tersebut menjadi konsen

dikarenakan terkait dengan permasalahan efisiensi pembiayaan yang dikeluarkan. Dalam pendistribusian barang, efisiensi pengiriman barang salah satunya ditentukan oleh lintasan yang dilalui untuk mengirimkan barang tersebut. Permasalahan pemilihan rute terpendek ini

dikenal dengan istilah *Travelling Salesman Problem (TSP)* merupakan masalah klasik yang mencoba mencari rute terpendek yang bisa dilalui salesman yang ingin mengunjungi beberapa kota tanpa harus mendatangi kota yang sama lebih dari satu kali. *Travelling Salesman Problem (TSP)* merupakan aplikasi dari Teori Graf. TSP terdiri dari *Symmetric TSP (STSP)* dan *Asymmetric TSP (ATSP)*. STSP adalah TSP dimana jalur pergi dan jalur pulang antara dua kota selalu sama. Sedangkan ATSP adalah TSP dimana jalur pergi dan jalur pulang tidak selalu sama (Caturiyati, 2008).

Solusi optimal dari permasalahan TSP ini, akan sangat membantu perusahaan pengiriman barang untuk mengefisienkan proses pengiriman barang, baik dari segi waktu maupun biaya. Terdapat banyak algoritma untuk melakukan pencarian rute terpendek. Pemilihan algoritma yang paling optimum selalu menjadi permasalahan dalam pencarian rute terpendek, dimana setiap algoritma memiliki kelebihan dan kekurangannya masing-masing. Algoritma adalah sebuah prosedur komputasi yang mentransformasikan sejumlah input menjadi sejumlah output (Wibowo, 2012). Sebuah algoritma dikatakan benar jika untuk setiap inputnya menghasilkan output yang benar pula. Dalam hal ini algoritma dapat digunakan sebagai metode untuk mengetahui langkah-langkah secara urut untuk mencapai tujuan.

Dalam menentukan rute terpendek dari TSP, konsep dasar yang menjadi acuan untuk mencari penyelesaiannya adalah teori graf. Graf adalah himpunan titik (*vertices*) dan himpunan ruas garis (*edges*). Titik-titik dalam sebuah graf dihubungkan satu sama lain melalui ruas. Suatu graf  $G$  terdiri dari himpunan tidak kosong dari elemen-elemen yang disebut titik (*vertices*) dan suatu daftar pasangan node yang tidak terurut disebut ruas (*edge*). Himpunan *vertices* dari suatu graf  $G$  dinotasikan dengan  $V(G)$  sedangkan daftar himpunan *edges* dari graph  $G$  dinotasikan dengan  $E(G)$ . Untuk selanjutnya, suatu graph  $G$  dapat dinotasikan dengan  $G = (V, E)$  yang artinya graf  $G$  mempunyai  $V$  *vertices* dan  $E$  *edges*.

Terdapat berbagai algoritma yang dapat digunakan untuk menentukan rute terpendek suatu lintasan. Dalam penelitian ini, algoritma yang akan dibandingkan adalah algoritma *Cheapest Insertion Heuristic (CIH)* dan *Greedy*. Berdasarkan

hal tersebut, adapun rumusan masalah dalam penelitian ini adalah 1) Bagaimana aplikasi algoritma *Cheapest Insertion Heuristic (CIH)* dan *Greedy* dalam penentuan rute terpendek? 2) Algoritma manakah yang memberikan hasil yang optimal dalam penentuan rute terpendek suatu lintasan?

## METODE PENELITIAN

Penelitian ini merupakan penelitian kualitatif. Data penelitian diambil dengan menggunakan metode survei dan wawancara. Dalam penelitian ini, peneliti mengambil rute dari perjalanan karyawan Cahaya Bintang *Setting and Advertising*. Cahaya Bintang adalah badan usaha yang bekerja dalam bidang *advertising*. Cahaya Bintang beralamat di Jl. Tukad Pakerisan GG 18 No.3 Panjer, Denpasar Selatan, Bali. Untuk setiap harinya karyawan Cahaya Bintang akan mengunjungi beberapa tempat di daerah Denpasar untuk membeli bahan baku seperti kertas, tinta dll serta mendistribusikan barang hasil produksi. Variabel eksternal seperti kecepatan kendaraan, kondisi lalu lintas dan cuaca dianggap konstan dan memberikan pengaruh yang sama disetiap rute perjalanan.

Tahap-tahap yang dilakukan dalam penelitian ini adalah identifikasi masalah, studi literature, pengumpulan data, representasi dalam graf dan penentuan rute optimal. Pada identifikasi masalah, permasalahan yang dibahas adalah menentukan rute optimal distribusi barang dengan mengkopmarasi antara algoritma *Cheapest Insertion Heuristic (CIH)* dan algoritma *Greedy*. Rute distribusi akan ditentukan dengan menggunakan *google map*. Pengitungan jarak masing-masing tempat dari tempat asal juga ditentukan dengan menggunakan *google map*. Setelah jarak masing-masing tempat sudah didapatkan proses berlanjut pada pengaplikasian algoritma *Cheapest Insertion Heuristic (CIH)* dan *Greedy*. Dari hasil pengaplikasian algoritma tersebut, kemudian dibandingkan hasil yang didapat.

## HASIL DAN PEMBAHASAN

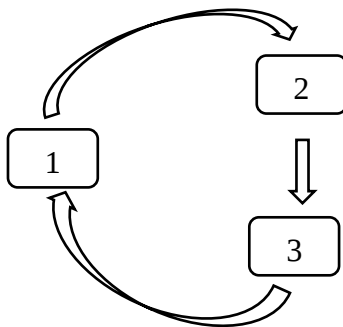
Algoritma *Cheapest Insertion Heuristics (CIH)* merupakan salah satu algoritma yang dapat digunakan untuk menyelesaikan masalah *Travelling Sales Problem*. Algoritma *Cheapest Insertion Heuristics (CIH)* memiliki kompleksitas waktu  $O(n^3)$ . Algoritma ini memiliki konsep menyisipkan

kota yang belum terlewati dengan tambahan jarak minimum hingga semua kota terlewati untuk mendapatkan solusi permasalahan (Rotlauff, 2011).

Sebuah perjalanan Algoritma *Cheapest Insertion Heuristics* (CIH) dimulai dari suatu simpul awal (1) menuju ke semua simpul (2,3,...,n) dan kembali ke simpul awal (1) tanpa ada simpul yang dikunjungi lebih dari satu kali dengan memperhitungkan tambahan jarak yang minimum ketika satu simpul disisipkan kedalam *partial tour* yang ada.

Algoritma *Cheapest Insertion Heuristics* (CIH) menurut Kusri, 2007 adalah sebagai berikut:

1. Penelusuran  
Dimulai dari sebuah tempat pertama yang dihubungkan dengan tempat terakhir.
2. Dibuat hubungan subtour  
Sebuah hubungan subtour dibuat antara 2 tempat tersebut. Yang dimaksud subtour adalah perjalanan dari kota pertama dan berakhir di kota pertama, misal (1,3) → (3,2) → (2,1) seperti tergambar dalam Gambar 1.



**Gambar 1.** Subtour

### 3. Mengganti arah hubungan

Salah satu arah hubungan (*arc*) dari dua kota diganti dengan kombinasi dua *arc*, yaitu *arc*(i,j) dengan *arc*(i,k) dan *arc*(k,j), dengan k diambil dari kota yang belum masuk subtour, dan dengan tambahan jarak terkecil. Jarak diperoleh dari Persamaan 1 :

$$c_{ik} + c_{kj} - c_{ij} \quad (1)$$

Keterangan :

$c_{ik}$  adalah jarak dari kota i ke kota k

$c_{kj}$  adalah jarak dari kota k ke kota j

$c_{ij}$  adalah jarak dari kota i ke kota j

4. Ulangi langkah 3 sampai seluruh kota masuk dalam subtour.

Algoritma *Greedy* adalah algoritma yang memecahkan langkah demi langkah dan merupakan salah satu metode dalam masalah optimasi. Menurut Dian (2013:2) algoritma *Greedy* merupakan algoritma yang dapat menentukan sebuah jalur terpendek antara node-node yang akan digunakan dengan mengambil secara terus menerus dan menambahkannya ke dalam jalur yang akan dilewati. Mengacu pada konsep *Greedy* yang menganggap bahwa pada setiap langkah akan dipilih tempat atau kota yang belum pernah dikunjungi, dimana tempat atau kota tersebut memiliki jarak terdekat dari tempat atau kota sebelumnya. Algoritma ini tidak mempertimbangkan nilai heuristic, yang dalam hal ini bisa berupa jarak langsung antar dua tempat (Aristi,2014). Algoritma *Greedy* membentuk solusi langkah per langkah sebagai berikut :

1. Terdapat banyak pilihan yang perlu dieksplorasi pada setiap langkah solusi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Keputusan yang telah diambil pada satu langkah tidak dapat diubah lagi pada langkah selanjutnya.
2. Pendekatan yang digunakan algoritma *Greedy* adalah membuat pilihan yang terlihat memberikan perolehan terbaik, yaitu dengan membuat pilihan optimum lokal pada setiap langkah dan diharapkan akan mendapatkan solusi optimum global (Hayati,2014)

Algoritma *Greedy* didasarkan pada pemindahan *edge* (*arc*) per *edge* (*arc*) dan pada setiap langkah yang diambil tidak memikirkan konsekuensi ke depan, *Greedy* tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada serta sebagian masalah *Greedy* tidak selalu berhasil memberikan solusi yang benar-benar optimum tapi pasti memberikan solusi yang mendekati nilai optimum. Masalah optimasi dalam konteks algoritma *Greedy*

disusun oleh elemen-elemen sebagai berikut :

1. Himpunan Kandidat : Himpunan ini berisi elemen-elemen yang memiliki peluang untuk pembentuk solusi. Pada persoalan lintasan terpendek dalam graf, himpunan kandidat ini adalah himpunan simpul dari graf tersebut.
2. Himpunan Solusi : Himpunan ini berisi solusi dari permasalahan yang diselesaikan dan elemennya terdiri dari elemen dalam himpunan kandidat, namun tidak semuanya atau dengan kata lain himpunan solusi ini adalah bagian dari himpunan kandidat.
3. Fungsi Seleksi : Fungsi yang setiap langkahnya memilih kandidat yang paling mungkin untuk menghasilkan solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.
4. Fungsi Kelayakan : Fungsi yang memeriksa apakah suatu kandidat yang telah dipilih (diseleksi) dapat memberikan solusi yang layak.
5. Fungsi Obyektif : Fungsi yang memaksimumkan atau meminimumkan nilai solusi. Tujuannya adalah memilih satu saja solusi terbaik dari masing-masing anggota himpunan solusi.

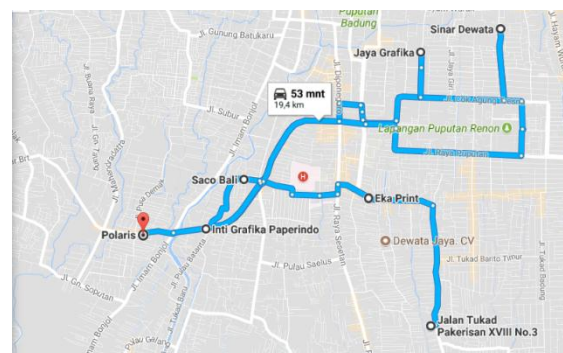
Dengan kata lain: algoritma *Greedy* melibatkan pencarian sebuah himpunan bagian,  $S$ , dari himpunan kandidat,  $C$ ; yang dalam hal ini,  $S$  harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan  $S$  dioptimisasi oleh fungsi obyektif (Munir, 2004). Berikut adalah cara kerja algoritma *Greedy* :

1. Periksa semua edge yang langsung bersesuaian dengan verteks a. Pilih edge yang bobotnya terkecil. Edge ini menjadi lintasan terpendek, sebut saja  $L(1)$ .
2. Tentukan lintasan terpendek kedua dengan cara berikut :
  - a. Hitung  $d(i) = \text{Panjang } L(1) + \text{bobot edge dari verteks akhir } L(1) \text{ ke verteks } i \text{ yang lain}$
  - b. Pilih  $d(i)$  yang terkecil
  - c. Bandingkan  $d(i)$  dengan bobot edge  $(a,i)$  lebih kecil daripada  $d(i)$ ,

makan  $L(2) = L(1) \cup (\text{edge dari verteks akhir } L(i) \text{ ke verteks } i)$

3. Dengan cara yang sama, ulangi langkah (2) untuk menentukan lintasan terpendek berikutnya.

Dalam penelitian ini, peneliti mengambil rute dari perjalanan karyawan dari Cahaya Bintang *Setting and Advertising*. Untuk setiap harinya karyawan Cahaya Bintang akan mengunjungi beberapa tempat di daerah Denpasar untuk membeli bahan baku seperti kertas, tinta serta mendistribusikan barang hasil produksi. Data yang diperoleh dari hasil observasi dan wawancara akan dianalisis dengan menggunakan algoritma CIH. Untuk menentukan tempat dan rute dari distribusi barang dari Cahaya Bintang akan dibantu dengan menggunakan aplikasi *google map*. Untuk menentukan jarak dan rute pada aplikasi *google map* adalah dengan menginput nama tempat atau alamat pada kolom search. *Google map* akan otomatis menampilkan jarak terpendek dari beberapa opsi rute antara tempat yang satu dengan yang lain. Berikut tampilan tempat-tempat yang akan dikunjungi oleh karyawan Cahaya Bintang pada Gambar 2.



**Gambar 2.** Rute Perjalanan dari Aplikasi Google Map

Setelah menentukan tempat yang akan dikunjungi, langkah selanjutnya adalah menghitung jarak setiap tempat dari titik asal yang disajikan pada Tabel 1.

**Tabel 1. Jarak Tempat Distribusi dari Titik**

| Asal               |       |
|--------------------|-------|
| Tempat             | Jarak |
| Cahaya Bintang (1) | 0 km  |

|                  |        |
|------------------|--------|
| Eka Print (2)    | 2,3 km |
| Inti Grafika (3) | 4,8 km |
| Jaya Grafika (4) | 4,9 km |
| Sinar Dewata (5) | 6,2 km |
| Saco Bali (6)    | 4,2 km |
| Polaris (7)      | 5,6 km |

Selanjutnya daftar tempat distribusi akan dikonversi menjadi angka yang tertera pada Tabel 2.

**Tabel 2. Konversi Jarak Tempat Distribusi dari Titik Asal**

| Tempat | Jarak  |
|--------|--------|
| 1      | 0 km   |
| 2      | 2,3 km |
| 3      | 4,8 km |
| 4      | 4,9 km |
| 5      | 6,2 km |
| 6      | 4,2 km |
| 7      | 5,6 km |

Langkah selanjutnya yaitu menentukan jarak dari masing-masing tempat yang disajikan dalam Tabel 3 berikut :

**Tabel 3. Jarak dari Masing-Masing Tempat**

| Tempat Asal | Tempat Tujuan | Jarak  |
|-------------|---------------|--------|
| 1           | 2             | 2,3 km |
| 1           | 3             | 4,8 km |
| 1           | 4             | 4,9 km |
| 1           | 5             | 6,2 km |
| 1           | 6             | 4,2 km |
| 1           | 7             | 5,6 km |
| 2           | 3             | 2,5 km |
| 2           | 4             | 2,5 km |
| 2           | 5             | 3,9 km |
| 2           | 6             | 1,8 km |
| 2           | 7             | 3,3 km |
| 3           | 4             | 4,5 km |
| 3           | 5             | 6,5 km |
| 3           | 6             | 1,1 km |
| 3           | 7             | 0,8 km |
| 4           | 5             | 2,4 km |
| 4           | 6             | 4,1 km |
| 4           | 7             | 5,5 km |
| 5           | 6             | 5,9 km |
| 5           | 7             | 7,4 km |
| 6           | 7             | 2,0 km |

Untuk mencari jarak terpendek dari tempat 1 ke tempat 7, adapun langkah-langkahnya adalah sebagai berikut :

1. Ambil perjalanan dari tempat 1 ke 7
2. Buat subtour (1,7) → (7,1)

3. Buat tabel yang menyimpan tempat yang bisa disisipkan dalam subtour beserta tambahan jaraknya, seperti ditampilkan dalam Tabel 4 berikut.

**Tabel 4. Arc Penambahan Subtour ke 1**

| Arc yang akan diganti | Arc yang akan ditambahkan ke Subtour | Tambahan Jarak                   |
|-----------------------|--------------------------------------|----------------------------------|
| (1,7)                 | (1,2) → (2,7)                        | $c_{12} + c_{27} - c_{17} = 0$   |
| (1,7)                 | (1,3) → (3,7)                        | $c_{13} + c_{37} - c_{17} = 0$   |
| (1,7)                 | (1,4) → (4,7)                        | $c_{14} + c_{47} - c_{17} = 4,8$ |
| (1,7)                 | (1,5) → (5,7)                        | $c_{15} + c_{57} - c_{17} = 8$   |
| (1,7)                 | (1,6) → (6,7)                        | $c_{16} + c_{67} - c_{17} = 6$   |
| (7,1)                 | (7,2) → (2,1)                        | $c_{72} + c_{21} - c_{71} = 0$   |
| (7,1)                 | (7,3) → (3,1)                        | $c_{73} + c_{31} - c_{71} = 0$   |
| (7,1)                 | (7,4) → (4,1)                        | $c_{74} + c_{41} - c_{71} = 4,8$ |
| (7,1)                 | (7,5) → (5,1)                        | $c_{75} + c_{51} - c_{71} = 8$   |
| (7,1)                 | (7,6) → (6,1)                        | $c_{76} + c_{61} - c_{71} = 6$   |

Dari tabel tersebut terdapat dua subtour yang memiliki jarak terpendek. Karena jaraknya sama oleh sebab itu dapat dipilih salah satu dari keduanya. Jadi dipilih untuk arc (1,7) diganti dengan arc (1,2) dan arc (2,7) atau arc (7,1) diganti dengan arc (7,2) dan arc (2,1). Dari penggantian subtour tersebut maka diperoleh subtour yang baru adalah (1,2) → (2,7) → (7,1).

Selanjutnya ulangi langkah di atas untuk menyisipkan subtour baru pada rute baru tersebut. Penambahan subtour baru disajikan pada Tabel 5.

**Tabel 5. Arc Penambahan subtour ke 2**

| Arc yang akan diganti | Arc yang akan ditambahkan ke Subtour | Tambahan Jarak                   |
|-----------------------|--------------------------------------|----------------------------------|
| (1,2)                 | (1,3) → (3,2)                        | $c_{13} + c_{32} - c_{12} = 5$   |
| (1,2)                 | (1,4) → (4,2)                        | $c_{14} + c_{42} - c_{12} = 5,1$ |
| (1,2)                 | (1,5) → (5,2)                        | $c_{15} + c_{52} - c_{12} = 7,8$ |
| (1,2)                 | (1,6) → (6,2)                        | $c_{16} + c_{62} - c_{12} = 3,7$ |
| (2,7)                 | (2,3) → (3,7)                        | $c_{23} + c_{37} - c_{27} = 0$   |
| (2,7)                 | (2,4) → (4,7)                        | $c_{24} + c_{47} - c_{27} = 3$   |
| (2,7)                 | (2,5) → (5,7)                        | $c_{25} + c_{57} - c_{27} = 8$   |
| (2,7)                 | (2,6) → (6,7)                        | $c_{26} + c_{67} - c_{27} = 0,5$ |
| (7,1)                 | (7,4) → (4,1)                        | $c_{74} + c_{41} - c_{71} = 4,8$ |
| (7,1)                 | (7,5) → (5,1)                        | $c_{75} + c_{51} - c_{71} = 8$   |
| (7,1)                 | (7,6) → (6,1)                        | $c_{76} + c_{61} - c_{71} = 6$   |

Dari tabel di atas diperoleh jarak terkecil adalah 0 dengan menggantikan arc (2,7) dengan arc(2,3) dan arc(3,7),

sehingga subtour baru yang dihasilkan adalah  $(1,2) \rightarrow (2,3) \rightarrow (3,7) \rightarrow (7,1)$ .

Lakukan langkah di atas agar semua tempat dilalui. Pencarian subtour baru akan disajikan pada tabel 6 di bawah :

**Tabel 6. Arc Penambahan subtour ke 3**

| Arc yang akan diganti | Arc yang akan ditambahkan ke Subtour | Tambahan Jarak                    |
|-----------------------|--------------------------------------|-----------------------------------|
| (1,2)                 | (1,4) $\rightarrow$ (4,2)            | $c_{14} + c_{42} - c_{12} = 5,1$  |
| (1,2)                 | (1,5) $\rightarrow$ (5,2)            | $c_{15} + c_{52} - c_{12} = 7,8$  |
| (1,2)                 | (1,6) $\rightarrow$ (6,2)            | $c_{16} + c_{62} - c_{12} = 3,7$  |
| (2,3)                 | (2,4) $\rightarrow$ (4,3)            | $c_{24} + c_{43} - c_{23} = 4,5$  |
| (2,3)                 | (2,5) $\rightarrow$ (5,3)            | $c_{25} + c_{53} - c_{23} = 7,9$  |
| (2,3)                 | (2,6) $\rightarrow$ (6,3)            | $c_{26} + c_{63} - c_{23} = 0,4$  |
| (3,7)                 | (3,4) $\rightarrow$ (4,7)            | $c_{34} + c_{47} - c_{37} = 9,2$  |
| (3,7)                 | (3,5) $\rightarrow$ (5,7)            | $c_{35} + c_{57} - c_{37} = 13,1$ |
| (3,7)                 | (3,6) $\rightarrow$ (6,7)            | $c_{36} + c_{67} - c_{37} = 2,3$  |
| (7,1)                 | (7,4) $\rightarrow$ (4,1)            | $c_{74} + c_{41} - c_{71} = 4,8$  |
| (7,1)                 | (7,5) $\rightarrow$ (5,1)            | $c_{75} + c_{51} - c_{71} = 8$    |
| (7,1)                 | (7,6) $\rightarrow$ (6,1)            | $c_{76} + c_{61} - c_{71} = 6$    |

Dari tabel di atas diperoleh jarak terkecil adalah 0,4 dengan menggantikan  $arc(2,3)$  dengan  $arc(3,6)$  dan  $arc(6,7)$ , sehingga subtour baru yang dihasilkan adalah:  $(1,2) \rightarrow (2,6) \rightarrow (6,3) \rightarrow (3,7) \rightarrow (7,1)$

Lakukan langkah di atas agar semua tempat dilalui. Pencarian subtour baru akan disajikan pada Tabel 7 di bawah :

**Tabel 7. Arc Penambahan subtour ke 4**

| Arc yang akan diganti | Arc yang akan ditambahkan ke Subtour | Tambahan Jarak                    |
|-----------------------|--------------------------------------|-----------------------------------|
| (1,2)                 | (1,4) $\rightarrow$ (4,2)            | $c_{14} + c_{42} - c_{12} = 5,1$  |
| (1,2)                 | (1,5) $\rightarrow$ (5,2)            | $c_{15} + c_{52} - c_{12} = 7,8$  |
| (2,6)                 | (2,4) $\rightarrow$ (4,6)            | $c_{24} + c_{46} - c_{26} = 4,8$  |
| (2,6)                 | (2,5) $\rightarrow$ (5,6)            | $c_{25} + c_{56} - c_{26} = 8$    |
| (6,3)                 | (6,4) $\rightarrow$ (4,3)            | $c_{64} + c_{43} - c_{63} = 7,5$  |
| (6,3)                 | (6,5) $\rightarrow$ (5,3)            | $c_{65} + c_{53} - c_{63} = 11,3$ |
| (3,7)                 | (3,4) $\rightarrow$ (4,7)            | $c_{34} + c_{47} - c_{37} = 9,2$  |
| (3,7)                 | (3,5) $\rightarrow$ (5,7)            | $c_{35} + c_{57} - c_{37} = 13,1$ |
| (7,1)                 | (7,4) $\rightarrow$ (4,1)            | $c_{74} + c_{41} - c_{71} = 4,8$  |
| (7,1)                 | (7,5) $\rightarrow$ (5,1)            | $c_{75} + c_{51} - c_{71} = 8$    |

Dari tabel di atas diperoleh jarak terkecil adalah 4,8. Karena terdapat 2 subtour dengan jarak terpendek maka dapat dipilih salah satunya. Pada perhitungan kali ini yang akan dipilih adalah  $arc(2,6)$  yang akan diganti dengan  $arc(2,4)$  dan  $arc(4,6)$ , sehingga subtour baru yang dihasilkan adalah:  $(1,2) \rightarrow (2,4) \rightarrow (4,6) \rightarrow (6,3) \rightarrow (3,7) \rightarrow (7,1)$

Lakukan langkah serupa agar semua tempat dilalui. Pencarian subtour baru akan disajikan pada Tabel 8 di bawah :

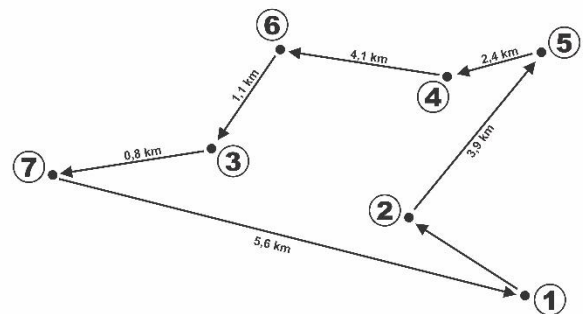
**Tabel 8. Arc Penambahan subtour ke 5**

| Arc yang akan diganti | Arc yang akan ditambahkan ke Subtour | Tambahan Jarak                    |
|-----------------------|--------------------------------------|-----------------------------------|
| (1,2)                 | (1,5) $\rightarrow$ (5,2)            | $c_{15} + c_{52} - c_{12} = 7,8$  |
| (2,4)                 | (2,5) $\rightarrow$ (5,4)            | $c_{25} + c_{54} - c_{24} = 3,8$  |
| (4,6)                 | (4,5) $\rightarrow$ (5,6)            | $c_{45} + c_{56} - c_{46} = 4,2$  |
| (6,3)                 | (6,5) $\rightarrow$ (5,3)            | $c_{65} + c_{53} - c_{63} = 11,3$ |
| (3,7)                 | (3,5) $\rightarrow$ (5,7)            | $c_{35} + c_{57} - c_{37} = 13,1$ |
| (7,1)                 | (7,5) $\rightarrow$ (5,1)            | $c_{75} + c_{51} - c_{71} = 8$    |

Dari tabel di atas diperoleh jarak terkecil adalah 3,8 dengan menggantikan  $arc(2,4)$  dengan  $arc(2,5)$  dan  $arc(5,4)$ , sehingga subtour baru yang dihasilkan adalah:

$(1,2) \rightarrow (2,5) \rightarrow (5,4) \rightarrow (4,6) \rightarrow (6,3) \rightarrow (3,7) \rightarrow (7,1)$ .

Berikut representasi lintasan terpendek dalam bentuk graf yang ditampilkan pada Gambar 3.



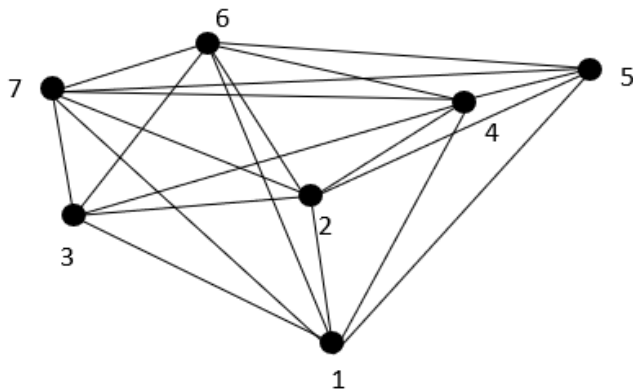
**Gambar 3. Lintasan Terpendek Distribusi Barang**

Berdasarkan perhitungan algoritma CIH dapat ditentukan bahwa jarak terpendek seorang karyawan Cahaya Bintang untuk mendistribusikan barang yaitu :

$$\begin{aligned}
 s &= s((1,2) + (2,5) + (5,4) + (4,6) + (6,3) + \\
 &\quad (3,7) + (7,1)) \\
 &= 2,3 + 3,9 + 2,4 + 4,1 + 1,1 + 0,8 + 5,6 \\
 &= 20,2 \text{ km}
 \end{aligned}$$

Adapun rute perjalanan yang menghasilkan jarak terpendek adalah dimulai dari titik awal menuju tempat nomor 2, selanjutnya nomor 5 selanjutnya nomor 4, kemudian nomor 6, kemudian nomor 3, dan yang terakhir adalah nomor 7 sebelum kembali ke titik awal.

Berikutnya akan dianalisis menggunakan algoritma Greedy. Pertama akan dirubah rute perjalanan karyawan dalam bentuk graf. Graf rute perjalanan karyawan Cahaya Bintang disajikan pada Gambar 4.



**Gambar 4.** Rute Tempat yang dikunjungi Karyawan Cahaya Bintang dalam Bentuk Graf.

Setelah merubah rute tempat yang dikunjungi karyawan Cahaya Bintang, kemudian akan ditentukan jarak minimal dari tempat 1 ke seluruh tempat dan kemudian kembali lagi ke 1. Jarak dari masing-masing titik ke titik yang lain dapat dilihat pada Tabel 3. Berdasarkan Tabel 3, maka terlebih dahulu ditentukan jarak minimum dari 1 ke titik yang lain ( $L(1)$ ). Sehingga sesuai dengan data yang disajikan dalam Tabel 3, jarak yang paling minimum adalah dari 1 ke 2 yaitu 2,3 km. Kemudian setelah mendapatkan  $L(1)$ , yang ditentukan selanjutnya adalah  $L(2)$ . Berdasarkan tabel maka jarak minimum adalah dari titik 2 ke titik 6 yaitu dengan jarak 1,8 km. Kemudian  $L(3)$  adalah jarak minimum dari titik 6 ke titik 3 yaitu dengan jarak 1,1 km.  $L(4)$  adalah jarak minimum dari titik 3 ke titik 7 yaitu dengan jarak 0,8 km.  $L(5)$  adalah jarak minimum dari titik 7 ke titik 4 yaitu dengan jarak 5,5 km.  $L(6)$  adalah jarak minimum dari titik 4 ke titik 5 yaitu dengan jarak 2,4 km. Terakhir yaitu  $L(7)$  adalah jarak minimum dari titik 5

kembali ke titik 1 yaitu dengan jarak 6,2 km. Sehingga dari jarak minimum lokal yang telah ditentukan dari setiap titik yang satu ke titik lainnya, maka nilai minimum Global adalah

$$L(1)+L(2)+L(3)+L(4)+L(5)+L(6)+L(7) = 2,3 \text{ km} + 1,8 \text{ km} + 1,1 \text{ km} + 0,8 \text{ km} + 5,5 \text{ km} + 2,4 \text{ km} + 6,2 \text{ km} = 20,1 \text{ km}$$

Rute minimum yang dihasilkan oleh algoritma Greedy adalah dari titik  $1 \rightarrow 2 \rightarrow 6 \rightarrow 3 \rightarrow 7 \rightarrow 4 \rightarrow 5 \rightarrow 1$  atau rute yang menghasilkan jarak minimum adalah dari Kantor Cahaya Bintang  $\rightarrow$  Eka Print  $\rightarrow$  Saco Bali  $\rightarrow$  Inti Grafika Paperindo  $\rightarrow$  Polaris  $\rightarrow$  Jaya Grafika  $\rightarrow$  Sinar Dewata  $\rightarrow$  Kantor Cahaya Bintang.

Jika dibandingkan antara algoritma Cheapest Insertion Heuristics (CIH) dan Greedy jarak minimum yang dihasilkan dari kedua algoritma tersebut hampir sama. Semuanya memberikan hasil jarak minimum yang optimal, namun yang membedakan adalah pemilihan rute perjalanannya. Dengan adanya aplikasi menggunakan algoritma Cheapest Insertion Heuristics (CIH) dan Greedy, tentunya akan membuat perjalanan menjadi efektif dan efisien. Berdasarkan pencarian rute terpendek menggunakan dua algoritma tersebut adapun kelebihan dan kekurangan dari masing-masing algoritma adalah sebagai berikut :

Kelebihan :

Algoritma Cheapest Insertion Heuristics (CIH) : Algoritma ini masih stabil digunakan untuk kasus TSP dengan jumlah kota yang besar.

Algoritma Greedy :

- Pencarian lebih simple apabila kota tidak terlalu banyak
- Dapat sampai tepat waktu menuju tujuan karena prinsip pencarian lintasan terpendek memakai fungsi seleksi dan itu berguna untuk menentukan jalan tersingkat untuk menuju suatu tempat.

Kekurangan :

Algoritma *Cheapest Insertion Heuristics* (CIH) : Prinsip pencariannya lebih rumit dan lebih lama dibanding algoritma greedy karena memungkinkan terjadinya perulangan perhitungan tambahan jarak (probabilitas edge yang akan digantikan) pada iterasi yang berbeda.

Algoritma *Greedy* :

- Hanya mengambil solusi yang terbaik yang mereka temukan pada saat itu tanpa memperhatikan konsekuensi ke depan.
- Keputusan optimal di setiap sub langkah tidak menghantarkan kita ke solusi optimal global yang diinginkan. - Kurang cocok digunakan apabila dengan jumlah kota yang besar.

Dari hasil analisis rute terpendek disdtribusi barang, jumlah tempat yang dikunjungi sangat mempengaruhi pemilihan algoritma. Hal tersebut sesuai dengan hasil penelitian Aristi, 2014 yang menyatakan bahwa Jumlah kota yang dikunjungi mempengaruhi pemilihan algoritma. Menurut Aristi, 2014 untuk pemcarian rute terpendek jumlah tempat yang dikunjungi masih relative sedikit, algoritma *Greedy* lebih efektif untuk digunakan. Namun jika jumlah tempat yang dikunjungi lebih banyak maka algoritma *Cheapest Insertion Heuristics* CIH lebih efektif untuk digunakan (Aristi, 2014). Berdasarkan hasil analisis yang telah dilakukan, kedua algoritma mampu memberikan solusi dalam penentuan rute terpendek pendistribusian barang.

## KESIMPULAN

Algoritma *Cheapest Insertion Heuristics* (CIH) dan *Greedy* merupakan algoritma yang dapat digunakan untuk menyelesaikan masalah TSP. Algoritma CIH memiliki konsep menyisipkan kota yang belum terlewati dengan tambahan jarak minimum hingga semua kota terlewati untuk mendapatkan solusi permasalahan. Sedangkan algoritma *Greedy* memiliki

konsep kerja yaitu menentukan jarak minimum lokak dari setiap titik ke titik yang lainnya. Kedua aplikasi algoritma tersebut dalam menentukan jarak minimum dalam suatu rute perjalanan, sama-sama memberikan hasil yang hampir sama. Yang membedakan hanyalah pemilihan rute perjalanan yang dilalui.

Adapun saran yang dapat disampaikan adalah, sangat disarankan bagi peneliti selanjutnya agar membandingkan algoritma *Cheapest Insertion Heuristics* (CIH) atau algoritma *Greedy* dengan algoritma yang lain untuk mengetahui algoritma mana yang lebih efektif untuj menyelesaikan *traveling salesman problem*.

## DAFTAR RUJUKAN

- Aristi, G., 2014. Perbandingan algoritma greedy, algoritma cheapest insertion heuristics dan dynamic programming dalam penyelesaian travelling salesman problem. *Paradigma-Jurnal Komputer dan Informatika*, 16(2), pp.52-58.
- Caturiyati. 2008. *Penyelesaian Asymmetric Travelling Salesman Problem dengan Algoritma Hungarian dan Algoritma Cheapest Insertion Heuristic*. Prosiding Seminar Nasional Matematika dan Pendidikan Matematika.
- Dian. 2013. Algoritma Optimasi Untuk Penyelesaian Travelling Salesman Problem (Optimization Algorithm For Solving Travelling Salesman Problem). *Jurnal Transformatika. Jurusan Teknologi Informasi Fakultas Teknologi Informasi dan Komunikasi, Universitas Semarang*. Volume 11, No.1, Juli 2013.
- Hayati, E.N. and Yohanes, A., 2014. Pencarian Rute Terpendek Menggunakan Algoritma Greedy. *Prosiding Seminar Nasional IENACO* (hal.391-397)
- Kusrini, 2007. "Penyelesaian Travelling Salesman Problem dengan Algoritma Cheapest Insertion Heuristics dan Basis Data". *Jurnal Informaika*, Vol.8(hal 109-114).

Munir, R., 2004. *Algoritma Greedy*. Departemen Teknik Informatika Institut Teknologi Bandung.  
Rotlauff, Franz. 2011. *Design of Modern Heuristics*. London : Springer

Wibowo, Abdul Ghofur dkk. 2012. "Rancang Bangun Aplikasi untuk Menentukan Jalur Terpendek Rumah Sakit Purbalingga dengan Metode Algoritma Dijkstra". *Juita* ISSN : 2086-9398, Vol 2 No.1