



Penentuan Lintasan Terpendek Perjalanan Pengiriman Barang Menggunakan Algoritma *Cheapest Insertion Heuristic* (Studi Kasus PT. Indah Logistik Cargo Bandar Lampung)

Ahya Shofa Ananda, Notiragayu*, Wamiliana, Muslim Ansori

Jurusan Matematika, Fakultas MIPA, Universitas Lampung

*Correspondence: E-mail: notiragayu@fmipa.unila.ac.id

ABSTRAK

Peningkatan jumlah anggota *online shop* membuat nilai transaksi *e-commerce* di Indonesia mengalami kenaikan. Hal ini mengakibatkan meningkatnya pendistribusian barang pada PT. Indah Logistik Cargo. Kendala pada PT. Indah Logistik Cargo adalah keterlambatan penyaluran paket ke para konsumen akibat jalur lintasan yang tidak efisien. Permasalahan pendistribusian pada perusahaan jasa pengiriman ini termasuk permasalahan *Travelling Salesman Problem* (TSP) yang dapat diselesaikan menggunakan Algoritma *Cheapest Insertion Heuristic* (CIH). Pada tulisan ini, penyelesaian masalah tersebut dilakukan secara manual dan juga menggunakan bahasa pemrograman *Python*. Data faktual yang dimiliki PT. Indah Logistik Cargo bahwa total jarak pengiriman barang yaitu 51.600 meter. Sementara, berdasarkan hasil perhitungan dengan algoritma CIH secara manual diperoleh total jarak 51.550 meter, yang lebih kecil dengan selisih 50 meter dari data faktual. Sedangkan, hasil perhitungan dengan bahasa pemrograman *Python* diperoleh 47.150 meter, atau mempunyai selisih 4.450 meter. Oleh karena itu, penggunaan algoritma CIH terbukti memberikan lintasan dengan total jarak tempuh yang lebih kecil.

© 2023 Kantor Jurnal dan Publikasi UPI

ABSTRACT

The growth in the number of *online shop* members has caused the value of *e-commerce* transactions in Indonesia to increase. This resulted in increased distribution of goods at PT. Indah Logistik Cargo. One of the constraints at PT. Indah Logistik Cargo is inefficient routes the distribution of packages. Distribution problem at this delivery service company include the *Traveling Salesman Problem* (TSP), which can be solved using the *Cheapest Insertion Heuristic* (CIH) algorithm. In this paper, the solution is carried out manually and also utilizes the *Python* programming language to optimize the delivery service travel path. Factual data owned by PT. Indah Logistik Cargo states that the total distance for delivering goods is 51.600 meters. Meanwhile, based on the results of manual calculations using the CIH algorithm, a total distance of 51.550 meters was obtained, which is smaller by a difference of 50 meters from the factual data. Furthermore, the results of calculations using the *Python* programming obtained 47.150 meters, or a difference of 4.450 meters. Therefore, the use of the CIH algorithm is proven to provide a path with a smaller total distance traveled.

© 2023 Kantor Jurnal dan Publikasi UPI

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima 13 Oktober 2023
Direvisi 20 Oktober 2023
Disetujui 28 Oktober 2023
Tersedia online 1 November 2023
Dipublikasikan 1 Desember 2023

Kata Kunci:

Algoritma CIH,
Lintasan Terpendek,
Pengiriman Barang.

Keywords:

CIH Algorithm,
Goods Delivery,
Shortest Path.

1. PENDAHULUAN

Dalam transaksi *e-commerce* diciptakan transaksi bisnis yang lebih praktis tanpa kertas (*paperless*) dan dalam transaksi *e-commerce* dapat tidak bertemu secara langsung (*face to face*) para pihak yang melakukan transaksi. Oleh karena itu, *e-commerce* menjadi penggerak ekonomi baru dalam bidang teknologi. Penggunaan internet dalam *e-commerce* ini memberikan dampak yang sangat positif, yakni dalam kecepatan dan kemudahan serta kecanggihan dalam melakukan interaksi global tanpa batasan tempat dan waktu yang kini menjadi hal yang biasa. Transaksi bisnis yang lebih praktis tanpa perlu kertas dan pena, perjanjian *face to face* (bertemu secara langsung) pelaku bisnis kini tidak diperlukan lagi, sehingga dapat dikatakan perdagangan elektronik atau *e-commerce* ini menjadi penggerak ekonomi baru dalam bidang teknologi khususnya di Indonesia. Kegiatan *e-commerce* tidak hanya jual beli barang dan jasa secara online tetapi ada industri lain di dalamnya seperti penyediaan jasa pengiriman barang atau logistic, produksi perangkat pintar, periklanan, *provider* telekomunikasi dan lain-lain (Nasution et al., 2020).

Berdasarkan laporan pada *website* Kementerian Keuangan, peningkatan jumlah anggota *online shop* membuat nilai transaksi *e-commerce* di Indonesia mengalami peningkatan pada tahun 2018 lalu mencapai Rp 77,766 triliun. Angka ini meroket 151% dibandingkan tahun sebelumnya yang mencapai Rp 30,942 triliun (<https://bppk.kemenkeu.go.id/>). Hal ini membuktikan bahwa bisnis melalui *e-commerce* semakin menjanjikan dan menguntungkan banyak pihak, salah satunya adalah perusahaan jasa pengiriman barang PT. Indah Logistik Cargo di kota Bandar Lampung.

Perusahaan jasa pengiriman barang menggunakan sistem distribusi dalam proses kegiatannya. Meningkatnya jumlah kegiatan jual beli *online* mengakibatkan pendistribusian pada perusahaan jasa pengiriman barang juga meningkat. Salah satu kendala pada PT. Indah Logistik Cargo adalah keterlambatan penyaluran paket dari jual beli *online* dari produsen ke konsumen akibat lintasan perjalanan yang tidak efisien sehingga perlu adanya lintasan lain untuk mengatasi kendala tersebut.

Pada lintasan yang menghubungkan tempat produksi menuju tempat distributor dapat direpresentasikan ke dalam bentuk graf. Tempat pendistribusian oleh distributor dapat diasumsikan sebagai titik (*vertex*). Kemudian, jalan yang menjadi penghubung antara tempat produksi dan tempat distributor dapat disebut sisi (*edge*).

Permasalahan pendistribusian pada perusahaan jasa pengiriman adalah permasalahan *Travelling Salesman Problem* (TSP). TSP masalah optimasi dengan tujuan untuk menentukan rute perjalanan yang dimulai dari lokasi awal kemudian mengunjungi beberapa lokasi hingga kembali lagi pada lokasi asal dengan setiap lokasi yang dikunjungi tepat hanya satu kali (Chaerunnissa, 2021).

TSP dapat diselesaikan menggunakan beberapa Algoritma, diantaranya *Ant Colony Optimimization Algorithm*, *Genetic Algorithm* (GA), *Nearest Neighbourhood Heuristic* (NNH) dan *Cheapest Insertion Heuristic* (CIH). Optimisasi koloni semut merupakan metode *meta heuristik* yang cukup sulit untuk menyelesaikan masalah TSP namun waktu yang diperlukan dalam eksekusi lebih cepat dibandingkan dengan metode eksak (Silalahi, Fathiah, & Supriyo, 2019). Algoritma Genetika cukup stabil dalam menyelesaikan masalah TSP (Ramadhania & Rani, 2021; Wiyanti, 2013), dan mampu memberikan jarak terpendek meski dengan jumlah kota yang besar. Akan tetapi, Algoritma Genetik ini sangat bergantung pada pemilihan parameter input yaitu ukuran populasi, besar maksimum generasi, ukuran peluang crossover dan mutation (Wiyanti, 2013). Algoritma *Nearest Neighbourhood Heuristic* (NNH)

merupakan algoritma yang sederhana dan cepat untuk membangun panjang tour awal yang layak bagi TSP. Algoritma ini menemukan kemungkinan terbaik di setiap langkah (Nuraiman *et al.*, 2018). Algoritma NNH lebih efektif dalam hal pencarian jarak terpendek dalam masalah TSP jika dibandingkan dengan algoritma CIH (Sinaga & Marpaung, 2023). Algoritma CIH berhasil diimplementasikan dengan mengetahui jumlah kota yang terhubung dan jaraknya masing-masing. Akan tetapi, algoritma CIH ini belum mampu diterapkan jika ada dua kota yang bobotnya berbeda (tidak simetri) dalam arah berbalikan dan juga jika ada kota yang tidak terhubung (Kusrini & Istiyanto 2007). CIH dapat mempercepat pencarian solusi masalah lintasan terpendek dengan membangun perjalanan (*subtour*) yang membuat lintasan perjalanan pendek dengan bobot minimal dan secara berturut-turut ditambah dengan tempat baru. Pemilihan titik baru tersebut dilakukan bersamaan dengan pemilihan sisi sehingga nilai penyisipan minimum (Saleh, Helmi, & Prihandono, 2015).

Proses pencarian solusi dengan algoritma CIH dapat menggunakan bahasa pemrograman Python (Sucilawati & Kurnianda, 2021). Penggunaan bahasa pemrograman Python diharapkan dapat membantu menyelesaikan persoalan pencarian lintasan terpendek untuk mengoptimalkan lintasan perjalanan jasa pengiriman PT. Indah Logistik Cargo dengan waktu yang cukup singkat.

2. METODE PENELITIAN

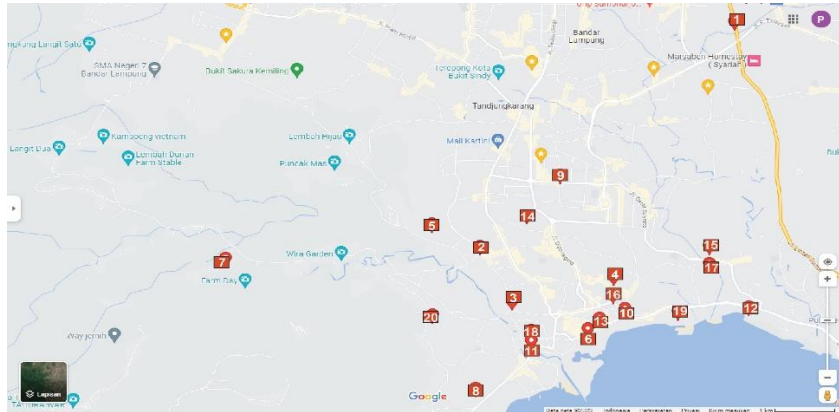
Langkah-langkah yang digunakan dalam penelitian ini adalah:

- a. Mengidentifikasi Algoritma *Cheapest Insertion Heuristic* sebagai metode untuk menyelesaikan masalah TSP.
- b. Data yang digunakan adalah data jarak dari lokasi A ke lokasi B pada jalur lintasan yang telah digunakan oleh PT. Indah Logistik Cargo Bandar Lampung di Kecamatan Teluk Betung. Sejumlah 20 titik lokasi yang digunakan oleh PT. Indah Logistik Cargo Bandar Lampung ini direpresentasikan dalam bentuk graf lengkap.
- c. Mencari lintasan terpendek pada permasalahan TSP menggunakan algoritma CIH dengan langkah-langkah sebagai berikut:
 - i) Dimulai dari sebuah lokasi yang dianggap sebagai titik awal dan akan terhubung dengan titik akhir.
 - ii) Kemudian, bangun *subtour* antar 2 titik (lokasi A ke lokasi B). *Subtour* yang dimaksud adalah lokasi awal mengunjungi suatu titik lokasi dan berakhir pada titik awal itu pula. Contoh *subtour* dapat direpresentasikan seperti: (1, 2) – (2, 3) – (3, 4) – (4, 1).
 - iii) Ubah salah satu busur (*arc*) antara 2 tempat. Lakukan penyisipan dengan menggunakan metode kombinasi pada busur (i, j) dengan busur (i, k) dan busur (k, j), dimana k adalah bukan anggota *subtour*. Dengan hasil kombinasi yang didapati menghasilkan jalur paling kecil. *Subtour* baru yang terbentuk menjadi $C_{ik} + C_{kj} - C_{ij}$ dimana C_{ik} adalah jarak dari tempat i ke tempat k , C_{kj} adalah jarak dari tempat k ke tempat j , dan C_{ij} adalah jarak dari tempat i ke tempat j .
 - iv) Ulangi langkah ke iii) sehingga seluruh tempat destinasi termuat dalam *subtour*.
- d. Mencari jalur lintasan dan total jarak tempuh lintasan menggunakan algoritma CIH dengan bahasa pemrograman *Python*.
- e. Melakukan perbandingan antara jarak lintasan yang digunakan PT. Indah Logistik Cargo Bandar Lampung di Kecamatan Teluk Betung dengan hasil jarak lintasan menggunakan algoritma CIH cara manual dan dengan pemrograman *Python*.

3. HASIL DAN PEMBAHASAN

3.1 Deskripsi Data

Bagian ini berisi data yang digunakan yaitu 20 titik lokasi pengantaran armada PT. Indah Logistik Cargo Bandar Lampung di Kecamatan Teluk Betung yang digambarkan dengan peta wilayah Bandar Lampung dan ditandai oleh tanda berwarna merah (Gambar 1). Selanjutnya data dipresentasikan dalam Tabel 1. Bobot sisi yang menghubungkan antar lokasi 20 titik pengiriman barang diambil menggunakan *Google Maps* dengan asumsi kondisi geografis normal tanpa hambatan dan jarak yang simetris (Tabel 2).



Gambar 1. Peta Wilayah Kecamatan Teluk Betung.

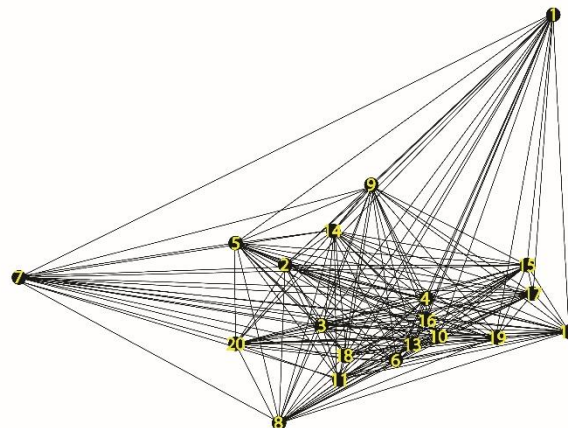
Tabel 1. Nama Titik Lokasi

No.	Nama Titik Lokasi	Titik
1	Gudang	1
2	Rumah Dinas PN Lampung, TBS	2
3	Kantor DPW hidayatullah, TBS	3
4	Sentra Komputer	4
5	Jl. P. Emir Gg. Karya Muda, TBS	5
6	PT. Sungai Budi, TBS	6
7	Ponpes Madarijul Ulum, TBB	7
8	Mushola Nurul Iman, TBB	8
9	Jl. Jend. Sudirman 68	9
10	Jl. Laksamana Malahayati 83	10
11	Jl. Ikan Sebelah Gg. Masjid, TBS	11
12	Graha Wangsa, BW	12
13	Jl. Laksamana Malahayati 160	13
14	Perum. Rasuna Hills Residence	14
15	Jl. Gatot Subroto, BW	15
16	Hazmy Motor, BW	16
17	Jl. Slamet Riyadi II, BW	17
18	Jl. Laksamana Malahayati 37B, TBS	18
19	Jl. Ikan Julung SKIP Rahayu, BW	19
20	Citra Garden Olivine, TBU	20

Tabel 2. Bobot Sisi Antar Lokasi Titik

Kota	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	10700	10500	8500	10300	11000	14600	13500	6500	9300	12100	8200	9800	8400	8000	9500	8100	11900	9800	12500
2	10700	0	1500	2900	1200	3300	5600	3900	3600	3600	2500	5900	3300	2100	5600	3800	5700	2300	4700	3500
3	10500	1500	0	2600	2400	2000	5900	2500	4600	3100	1100	5400	3000	3200	5500	2700	5400	800	4000	2100
4	8500	2900	2600	0	3900	1800	8300	4300	3100	750	2800	3000	1900	2600	3200	450	3000	2600	1700	4700
5	10300	1200	2400	3900	0	4300	6400	4900	3400	4600	3500	6900	4300	3100	6300	4700	6500	3300	5700	4500
6	11000	3300	2000	1800	4300	0	7600	2700	4400	800	1200	3600	350	4000	3800	1100	3600	1100	2300	3800
7	14600	5600	5900	8300	6400	7600	0	6900	9100	8900	6700	11200	8800	7600	11000	9000	11200	6500	9800	5600
8	13500	3900	2500	4300	4900	2700	6900	0	6300	4800	1500	7000	3100	5600	7200	4400	7000	2000	5700	3900
9	6500	3600	4600	3100	3400	4400	9100	6300	0	4100	4900	5300	4600	1600	3600	4000	3700	4700	5400	6500
10	9300	3600	3100	750	4600	800	8900	4800	4100	0	3300	3000	450	3900	3200	500	3000	3100	1700	5100
11	12100	2500	1100	2800	3500	1200	6700	1500	4900	3300	0	5600	1700	4300	5800	2900	5600	350	4200	2900
12	8200	5900	5400	3000	6900	3600	11200	7000	5300	3000	5600	0	3700	6000	1800	3300	2100	5400	2200	7400
13	9800	3300	3000	1900	4300	350	8800	3100	4600	450	1700	3700	0	4200	3900	750	3700	1500	2400	5000
14	8400	2100	3200	2600	3100	4000	7600	5600	1600	3900	4300	6000	4200	0	4200	3700	4400	4000	4900	5300
15	8000	5600	5500	3200	6300	3800	11000	7200	3600	3200	5800	1800	3900	4200	0	3500	750	4300	2400	7600
16	9500	3800	2700	450	4700	1100	9000	4400	4000	500	2900	3300	750	3700	3500	0	3300	2700	1900	4800
17	8100	5700	5400	3000	6500	3600	11200	7000	3700	3000	5600	2100	3700	4400	750	3300	0	5400	2200	7400
18	11900	2300	800	2600	3300	1100	6500	2000	4700	3100	350	5400	1500	4000	4300	2700	5400	0	4100	2700
19	9800	4700	4000	1700	5700	2300	9800	5700	5400	1700	4200	2200	2400	4900	2400	1900	2200	4100	0	6100
20	12500	3500	2100	4700	4500	3800	5600	3900	6500	5100	2900	7400	5000	5300	7600	4800	7400	2700	6100	0

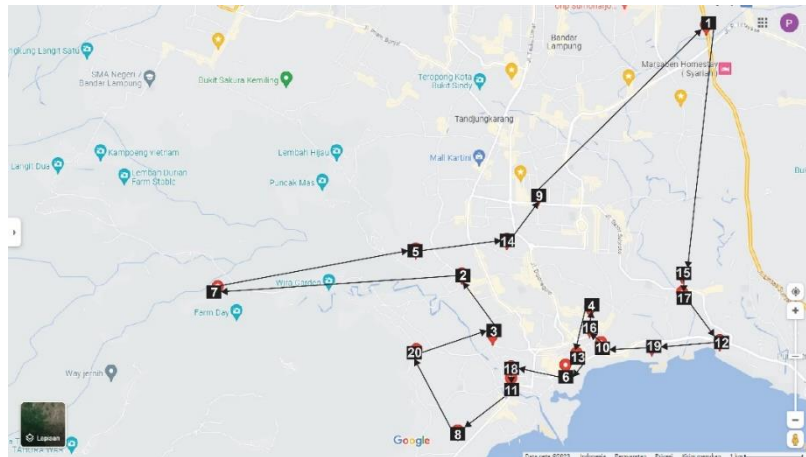
Diasumsikan nilai bobot sisi dari lokasi A ke lokasi B sama dengan bobot sisi dari lokasi B ke lokasi A. Diasumsikan pula nilai bobot sisi dari A ke B maupun sebaliknya adalah sama walaupun dalam pengambilan waktu data yang berbeda. Graf yang terbentuk adalah graf lengkap dengan $n = 20$, dimana masing-masing titik pada graf tersebut berderajat sama yaitu $n - 1 = 19$ (Gambar 2).



Gambar 2. Representasi 20 Titik Lokasi Dalam Bentuk Graf Lengkap.

3.2. Jarak Tempuh Perjalanan Kurir PT. Indah Logistik Cargo Bandar Lampung

Melalui data yang diperoleh dari kurir ekspedisi PT. Indah Logistik Cargo Bandar Lampung, jalur lintasan pengiriman barang yang digunakan oleh PT. Indah Logistik Cargo Bandar Lampung adalah $(1, 15) - (15, 17) - (17, 12) - (12, 19) - (19, 10) - (10, 16) - (16, 4) - (4, 13) - (13, 6) - (6, 18) - (18, 11) - (11, 8) - (8, 20) - (20, 3) - (3, 2) - (2, 7) - (7, 5) - (5, 14) - (14, 9) - (9, 1)$ sehingga menghasilkan total jarak tempuh sebesar 51.600 meter. Lintasan direpresentasikan pada Gambar 3.



Gambar 3. Bentuk Lintasan Perjalanan Kurir PT. Indah Logistik Cargo.

3.2. Penerapan Algoritma CIH

Algoritma CIH menghitung jarak efektif melalui lintasan perjalanan yang terbentuk dari titik 1 yakni Gudang PT. Indah Logistik Cargo dengan melewati titik-titik lainnya (titik 2, 3, ..., 20) hingga kembali ke titik 1 semula. Berikut langkah-langkah algoritma CIH:

Iterasi 1

1. Pilih titik 1 – 9 – 1 sebagai perjalanan awal (jarak terpendek antar titik = 6500 meter pada Tabel 2).
2. Buat *subtour* ke 1 yaitu (1,9) – (9,1). Kemudian akan dibentuk *subtour* yang menghubungkan kedua titik dengan mengganti busur (i, j) ke dalam bentuk kombinasi dua busur (i, k) dan (k, j). Titik i adalah titik asal yaitu titik 1, titik j adalah titik tujuan yaitu titik 9, dan titik k adalah titik yang belum dilewati dalam *subtour* saat ini.
3. Buat tabel hasil sisipan dari langkah 2 dan pilih titik (k=14) dengan tambahan jarak terkecil kemudian sisipkan sehingga diperoleh *subtour* ke 2 (Tabel 3).

Tabel 3. Hasil Penyisipan Simpul Pada Busur dalam Iterasi 1

Busur/Arc	Subtour	Sisipan ($C_{ik} + C_{kj} - C_{ij}$)	Hasil Sisipan
(1, 9)	(1, 2)-(2, 9)	$10700 + 3600 - 6500$	7800
(1, 9)	(1, 3)-(3, 9)	$10500 + 4600 - 6500$	8600
(1, 9)	(1, 4)-(4, 9)	$8500 + 3100 - 6500$	5100
(1, 9)	(1, 5)-(5, 9)	$10300 + 3400 - 6500$	7200
(1, 9)	(1, 6)-(6, 9)	$11000 + 4400 - 6500$	8900
(1, 9)	(1, 7)-(7, 9)	$14600 + 9100 - 6500$	17200
(1, 9)	(1, 8)-(8, 9)	$13500 + 6300 - 6500$	13300
(1, 9)	(1, 10)-(10, 9)	$9300 + 4100 - 6500$	6900
(1, 9)	(1, 11)-(11, 9)	$12100 + 4900 - 6500$	10500
(1, 9)	(1, 12)-(12, 9)	$8200 + 5300 - 6500$	7000
(1, 9)	(1, 13)-(13, 9)	$9800 + 4600 - 6500$	7900

(1, 9)	(1, 14)-(14, 9)	8400 + 1600 – 6500	3500*
(1, 9)	(1, 15)-(15, 9)	8000 + 3600 – 6500	5100
(1, 9)	(1, 16)-(16, 9)	9500 + 4000 – 6500	7000
(1, 9)	(1, 17)-(17, 9)	8100 + 3700 – 6500	5300
(1, 9)	(1, 18)-(18, 9)	11900 + 4700 – 6500	10100
(1, 9)	(1, 19)-(19, 9)	9800 + 5400 – 6500	8700
(1, 9)	(1, 20)-(20, 9)	12500 + 6500 – 6500	12500

4. Terbentuk subtour ke 2 yaitu (1,14) – (14,9) – (9,1). Ulangi langkah 2 karena belum melewati seluruh titik.

Iterasi 2

1. Buat subtour ke 2 yaitu (1,14) – (14,9) – (9,1) dan sisipkan titik k titik yang belum dilewati dalam subtour ini.
2. Buat Tabel hasil sisipan dari langkah 1 dan pilih titik (k=4) dengan tambahan jarak terkecil kemudian sisipkan sehingga diperoleh subtour ke 3.

Tabel 4. Hasil Penyisipan Simpul Pada Busur dalam Iterasi 2

Busur/Arc	Subtour	Sisipan ($C_{ik} + C_{kj} - C_{ij}$)	Hasil Sisipan
(1, 14)	(1, 2)-(2, 14)	10700 + 2100 – 8400	4400
(1, 14)	(1, 3)-(3, 14)	10500 + 3200 – 8400	5300
(1, 14)	(1, 4)-(4, 14)	8500 + 2600 – 8400	2700
(1, 14)	(1, 5)-(5, 14)	10300 + 3100 – 8400	5000
(1, 14)	(1, 6)-(6, 14)	11000 + 4000 – 8400	6600
(1, 14)	(1, 7)-(7, 14)	14600 + 7600 – 8400	13800
(1, 14)	(1, 8)-(8, 14)	13500 + 5600 – 8400	10700
(1, 14)	(1, 10)-(10, 14)	9300 + 3900 – 8400	4800
(1, 14)	(1, 11)-(11, 14)	12100 + 4300 – 8400	8000
(1, 14)	(1, 12)-(12, 14)	8200 + 6000 – 8400	5800
(1, 14)	(1, 13)-(13, 14)	9800 + 4200 – 8400	5600
(1, 14)	(1, 15)-(15, 14)	8000 + 4200 – 8400	3800
(1, 14)	(1, 16)-(16, 14)	9500 + 3700 – 8400	4800
(1, 14)	(1, 17)-(17, 14)	8100 + 4400 – 8400	4100
(1, 14)	(1, 18)-(18, 14)	11900 + 4000 – 8400	7500
(1, 14)	(1, 19)-(19, 14)	9800 + 4900 – 8400	6300
(1, 14)	(1, 20)-(20, 14)	12500 + 5300 – 8400	9400
(14, 9)	(14, 2)-(2, 9)	2100 + 3600 – 1600	4100
(14, 9)	(14, 3)-(3, 9)	3200 + 4600 – 1600	6200
(14, 9)	(14, 4)-(4, 9)	2600 + 3100 – 1600	4100
(14, 9)	(14, 5)-(5, 9)	3100 + 3400 – 1600	4900
(14, 9)	(14, 6)-(6, 9)	4000 + 4400 – 1600	6800
(14, 9)	(14, 7)-(7, 9)	7600 + 9100 – 1600	15100
(14, 9)	(14, 8)-(8, 9)	5600 + 6300 – 1600	10300

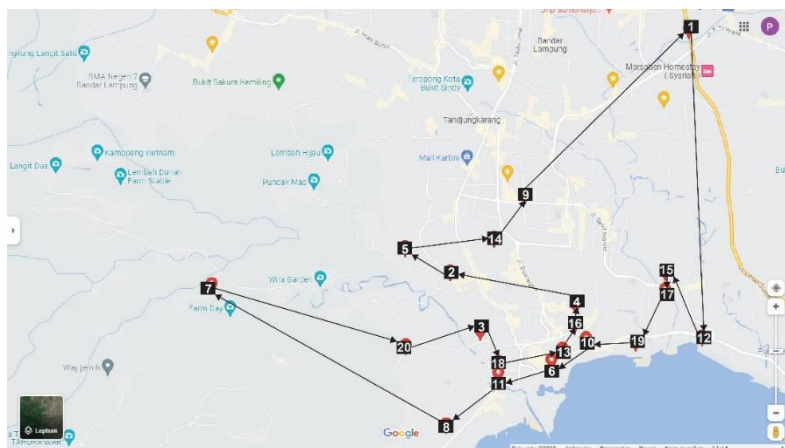
(14, 9)	(14, 10)-(10, 9)	3900 + 4100 – 1600	6400
(14, 9)	(14, 11)-(11, 9)	4300 + 4900 – 1600	7600
(14, 9)	(14, 12)-(12, 9)	6000 + 5300 – 1600	9700
(14, 9)	(14, 13)-(13, 9)	4200 + 4600 – 1600	7200
(14, 9)	(14, 15)-(15, 9)	4200 + 3600 – 1600	6200
(14, 9)	(14, 16)-(16, 9)	3700 + 4000 – 1600	6100
(14, 9)	(14, 17)-(17, 9)	4400 + 3700 – 1600	6500
(14, 9)	(14, 18)-(18, 9)	4000 + 4700 – 1600	7100
(14, 9)	(14, 19)-(19, 9)	4900 + 5400 – 1600	8700
(14, 9)	(14, 20)-(20, 9)	5300 + 6500 – 1600	10200

3. Terbentuk subtour ke 3 yaitu (1, 4) – (4, 14) – (14, 9) – (9, 1). Ulangi langkah 2 karena belum melewati seluruh titik.

Iterasi ke-3

1. Iterasi ke-3 mendapatkan titik nomor 4 yang terpilih untuk disisipkan pada *subtour* selanjutnya yaitu membentuk *subtour* (1, 4) – (4, 14) – (14, 9) – (9,1).
2. Dengan cara yang sama lakukan iterasi selanjutnya hingga seluruh titik sudah terlewati.

Dalam proses penghitungan manual ini, dilakukan sebanyak 18 kali iterasi hingga seluruh titik telah dilewati sekali dan kembali ke titik awal atau Gudang PT. Indah Logistik Cargo. Berdasarkan hasil penghitungan yang telah dilakukan, diperoleh jalur lintasan dengan urutan yang terbentuk adalah (1, 12) – (12, 15) – (15, 17) – (17, 19) – (19, 10) – (10, 6) – (6, 11) – (11, 8) – (8, 7) – (7, 20) – (20, 3) – (3, 18) – (18, 13) – (13, 16) – (16, 4) – (4, 2) – (2, 5) – (5, 14) – (14, 9) – (9, 1). Total jarak tempuh yang berhasil didapatkan dari lintasan yang terbentuk melalui 20 titik ini adalah 51.550 meter. Lintasan perjalanan yang telah diperoleh ditunjukkan pada Gambar 4.



Gambar 4. Bentuk Lintasan Perjalanan Pada Penghitungan Manual.

3.3. Penerapan Algoritma CIH dengan Python

Pada tahap ini dilakukan pencarian lintasan perjalanan dengan bahasa pemrograman *Python* menggunakan algoritma CIH. Penghitungan ini diawali dengan membuat matriks jarak yang

diambil dari jarak bobot sisi dari lokasi i ke lokasi j dengan asumsi A ke B adalah sama dengan B ke A sesuai urutan pada tabel (Gambar 5). Selanjutnya adalah membuat model perutean, *callback* jarak dan penetapan bobot, penetapan parameter penelusuran, dan diakhiri *print* solusi.

```

from ortools.constraint_solver import routing_enums_pb2
from ortools.constraint_solver import pywrapcp

def create_data_model():
    """Stores the data for the problem."""
    data = {}
    data['distance_matrix'] = [
        [0, 2451, 713, 1018, 1631],
        [2451, 0, 1745, 1524, 831],
        [713, 1745, 0, 355, 920],
        [1018, 1524, 355, 0, 700],
        [1631, 831, 920, 700, 0],
    ] # yapf: disable
    data['num_vehicles'] = 1
    data['depot'] = 0
    return data

```

Gambar 5. Bagian script program algoritma CIH

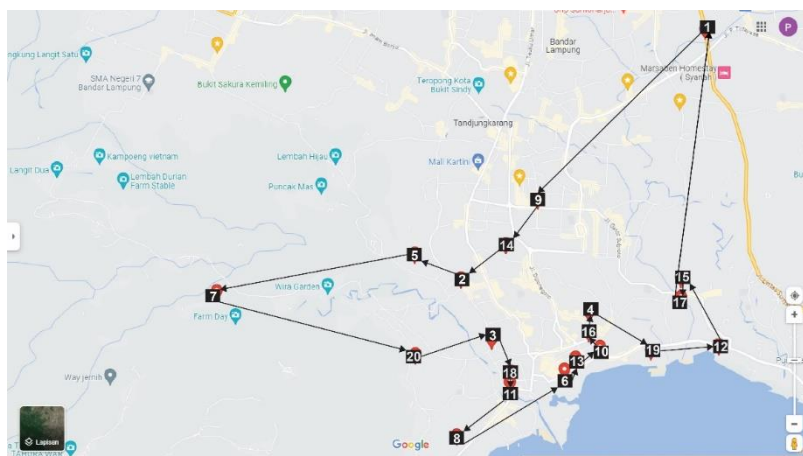
Pada *script* algoritma CIH dengan 20 titik, menghasilkan jalur lintasan dan total jarak tempuh lintasan yaitu:

Objective: 47150 meters

Route for vehicle 0:

0 -> 8 -> 13 -> 1 -> 4 -> 6 -> 19 -> 2 -> 17 -> 10 -> 7 -> 5 -> 12 -> 9 -> 15 -> 3 -> 18 -> 11 -> 14 -> 16 -> 0 atau lintasan dapat dituliskan sebagai berikut:

(1, 9) – (9, 14) – (14, 2) – (2, 5) – (5, 7) – (7, 20) – (20, 3) – (3, 18) – (18, 11) – (11, 8) – (8, 6) – (6, 13) – (13, 10) – (10, 16) – (16, 4) – (4, 19) – (19, 12) – (12, 15) – (15, 17) – (17, 1) dengan jarak tempuh 47.150 meter. Lintasan direpresentasikan pada Gambar 6.



Gambar 6. Bentuk Lintasan Perjalanan Pada Penghitungan *Python*.

4. KESIMPULAN

Berdasarkan hasil dan pembahasan yang telah dilakukan, diperoleh total jarak tempuh lintasan perjalanan pengiriman barang yang digunakan oleh PT. Indah Logistik Cargo Bandar Lampung adalah sebesar 51.600 meter. Total jarak yang diperoleh menggunakan algoritma

CIH secara manual adalah 51.550 meter. Kemudian dengan menggunakan bahasa pemrograman *Python*, total jarak tempuh yang diperoleh adalah 47.150 meter. Jika dibandingkan dengan total jarak tempuh lintasan yang digunakan PT. Indah Logistik Cargo yaitu 51.600 meter dan algoritma *Cheapest Insertion Heuristic* secara manual masih memiliki total jarak yang lebih minimum dengan selisih 50 meter dan dengan bahasa pemrograman *Python* dengan selisih 4.450 meter. Maka, penggunaan algoritma CIH terbukti memberikan lintasan dengan total jarak tempuh yang lebih optimal.

5. DAFTAR PUSTAKA

- Chaerunnissa, H. (2021). Penyelesaian travelling salesman problem dengan menggunakan algoritma artificial bee colony. *Jurnal Riset Matematika*, 1(1), 37-45.
- Kusrini dan Istiyanto, J.E. (2007). Penyelesaian travelling salesman problem dengan algoritma cheapest insertion heuristics dan basis data. *Jurnal Informatika*, 8(2), 109–114.
- Nasution, E. Y., Hariani, P., Hasibuan, L. S., & Pradita, W. (2020). Perkembangan transaksi bisnis e-commerce terhadap pertumbuhan ekonomi di Indonesia. *Jurnal Ekonomi dan Ekonomi Syariah*, 3(2), 506-519.
- Nuraiman, D., Ilahi, F., Dewi, Y., & Hamidi, E. A. Z. (2018, July). A new hybrid method based on nearest neighbor algorithm and 2-Opt algorithm for traveling salesman problem. In *2018 4th International Conference on Wireless and Telematics (ICWT)* (pp. 1-4). IEEE.
- Ramadhania, S. E., & Rani, S. (2021). Implementasi kombinasi algoritma genetika dan tabu search untuk penyelesaian travelling salesman problem. *Automata*, 2(1), 1-8.
- Saleh, K., Helmi, dan Prihandono, B. (2015). Penentuan rute terpendek dengan menggunakan algoritma cheapest insertion heuristic (Studi kasus: PT. Wicaksana Overseas International Tbk. Cabang Pontianak). *Buletin Ilmiah Matematika, Statistika dan Terapannya*, 4(3), 295–304.
- Silalahi, B. P., Fathiah N., dan Supriyo P. T. (2019). Use of ant colony optimization algorithm for determining travelling salesman problem routes. *Jurnal Matematika*, 5(2), 100-111.
- Sinaga, R. P. dan Marpaung, F. (2023). Perbandingan algoritma cheapest insertion heuristic dan nearest neighbor dalam menyelesaikan travelling salesman problem. *Jurnal Riset Rumpun Matematika dan Ilmu Pengetahuan Alam*, 2(2), 238-247.
- Sucilawati, T., dan Kurnianda, N. R. (2021). Implementation of shortest route file delivery on the messenger population and civil registration of DKI Jakarta using TSP backtracking method. *International Journal of Computer Techniques*, 8(2), 18-26.
- Wiyanti D. T. (2013). Algoritma optimasi untuk penyelesaian travelling salesman problem. *Jurnal Transformatika*, 11(1), 1-6.