IMPLEMENTASI ALGORITMA *CHEAPEST INSERTION HEURISTICS* BERBASISKAN *ANDROID* DAN *GOOGLE MAPS* PADA PT.XYZ

IMPLEMENTATION CHEAPEST INSERTION HEURISTICS ALGORITHM BASED ON ANDROID AND GOOGLE MAPS AT PT. XYZ

Shah Khadafi [1], Ovilia Rosa Saputri[2]

[1,2)] *Program Studi Sistem Informasi, Fakultas Teknik Elektro Dan Teknologi Informasi, Institut Teknologi Adhi Tama Surabaya*
Email : khadafi@itats.ac.id[1)], oviliarosa9@gmail.com[2)]

Abstrak

Pada sebuah perusahaan eksedisi seorang kurir diperlukan aktivitasnya untuk mendukung kegiatan operasional perusahaan, terutama ketika seorang kurir mengantar pengiriman paket ke alamat tujuan yang sesuai. PT.XYZ adalah sebuah perusahaan ekspedisi dengan area kerja sekitar kota Surabaya dan sekitarnya. Permasalahan yang dihadapi oleh perusahaan tersebut, adalah beberapa kurirnya belum memahami dan mengetehui beberapa area dan jalan yang ada di kota Surabaya. Fokus penelitian ini pada topik Travelling Salesman Problem (TSP) yang bertujuan membentuk rute pengiriman yang tercepat dan efisien bagi kurir, yang nantinya bisa menjadi panduan bagi kurir yang kesulitan menemukan rute alamat paket pengirimannya, sehingga. mampu memberikan solusi dalam penentuan rute dengan jarak total paling minimum. Metode yang digunakan dalam pembuatan sistem ini yaitu Heuristic dengan algoritma Cheapest Insertion Heuristic (CIH). Hasil perhitungan algoritma CIH memanfaatkan Google maps sebagai rute yang telah terbentuk dan dapat dijadikan panduan bagi kurir. Pengujian sistem yang dilakukan terhadap tiga kurir terhadap efisiensi jarak tempuh dengan menggunakan CIH yaitu, kurir 1 efiisiensi yang dihasilkan sebesar 34,2 %, kurir 2 dengan efisiensi sebesar 28 %, dan kurir 3 efisiensi sebesar 21,31 %. Sedangkan total keseluruhan efisiensi yang dihasilkan oleh system ini sebesar 83,51%.

Kata kunci: *Cheapest Insertion Heuristic, Heuristic, Kurir, Traveling Salesman Problem*

Abstract

In an exedition company, a courier needs his activities to support the company's operational activities, especially when a courier delivers parcel deliveries to the appropriate destination address. PT. XYZ is an expedition company with a work area around the city of Surabaya and its surroundings. The problem faced by the company is that some of the couriers do not understand and know several areas and roads in the city of Surabaya. The focus of this research is on the topic of the Traveling Salesman Problem (TSP) which aims to form the fastest and most efficient delivery route for couriers, which can later become a guide for couriers who have difficulty finding the route of the delivery package address. able to provide solutions in determining the route with the minimum total distance. The method used in making this system is Heuristic with the Cheapest Insertion Heuristic (CIH) algorithm. The results of the calculation of the CIH algorithm utilize Google maps as a route that has been formed and can be used as a guide for couriers. System testing carried out on three couriers on the efficiency of mileage using CIH, namely, courier 1 produced efficiency of 34.2%, courier 2 with efficiency of 28%, and courier 3 efficiency of 21.31%. Meanwhile, the total efficiency generated by this system is 83.51%.

Keywords : *Cheapest Insertion Heuristic, Courir, Heuristic, Traveling Salesman Problem*

1. PENDAHULUAN

Dengan berkembangnya teknologi saat ini, banyak masyarakat telah memanfaatkan dan telah melakukan penerapan dalam pengembangan bisnisnya. Salah satu bentuk bisnis yang sering dilakukan masyarakat adalah berjualan *online*. Menurut Badan Pusat Statistik (2020), bulan Maret 2020, penjualan *online* melonjak 320% dari total penjualan online awal tahun. Lonjakan semakin tajam terjadi, penjualan *online* April 2020 tercatat meningkat 480% dari Januari 2020[1].

Beberapa penelitian yang terkait dengan permasalahan TSP (*Travelling Salesman Problem*) saat ini telah banyak menjadi perhatian akademisi. Beberapa algoritma telah diadopsi untuk dapat menyelesaikan kompleksnya permasalahan TSP. Kecepatan dan ketepatan seorang

kurir merupakan tantangan yang harus dapat diselesaikan oleh perusahaan tersebut agar paket pengiriman dapat terkirim sesuai dengan alamatnya yang benar [2].

PT. XYZ merupakan perusahaan ekspedisi di kota Surabaya. Seorang kurir diharuskan mengirimkan paket dengan waktu yang sangat singkat dengan pemilihan rute-rute yang terpendek dan efisien. Seorang kurir harus dapat mengenal dan mengetahui beberapa area dan jalan-jalan di kota Surabaya. Beberapa daerah di kota Surabaya terjadi kerawanan macet terutama di wilayah utara-selatan. Beberapa kurir PT.XYZ kebanyakan masih belum memahami dan mengenal arek kota Surabaya, hal ini dikarenakan beberapa kurirnya domisilinya berasal dari luar kota Surabaya. Permasalahan kurir menjadi menjadi landasan peneliti untuk menerapkan *Travelling Saleman Problem* (TSP) menggunakan algoritma *Cheapest Insertion Heuristic* (CIH) [3]. Penggunaan algoritma Cheapest Insertion Heuristic (CIH) ini dipilih karena memiliki langkah-langkah sederhana dalam penyelesaian masalah ini dibanding algoritma lain, dengan cara mencari sisipan terkecil dari setiap rute. Dalam penelitian ini, algoritma *Cheapest Insertion Heuristic* dapat membantu pembentukan rute dan pengurutan pengiriman paket yang dikemas dalam bentuk sistem android. Penelitian ini bertujuan membuat sebuah rute pengiriman yang terpendek [4], dan memudahkan kurir perusahaan PT.XYZ berbasiskan Google maps dan sistem Android [5], yang juga bisa menjadi panduan bagi usernya [6], dalam hal ini usernya adalah kurir.

2. DASAR TEORI

2.1. Traveling Salesman Problem (TSP)

TSP adalah permasalahan optimasi yang menarik perhatian para peneliti, dikarenakan kasus TSP mudah didefinisikan namun sulit diselesaikan. Mengambil contoh permasalahan seorang *sales* yang melewati kota-kota tujuan setiap tujuan dikunjungi satu kali dan selanjutnya kembali ke kota asal dan total jarak yang ditempuh haruslah seminimum mungkin[7].

2.2. Cheapest Insertion Heuristic

Salah satu algoritma heuristik yang efektif digunakan menyelesaikan pencarian rute terpendek adalah algoritma *cheapest insertion heuristic* (CIH). Algoritma CIH adalah algoritma yang membentuk suatu *tour* dengan membuat rute jalur terpendek dengan bobot minimal dan secara berturut-turut ditambah dengan tempat baru. Pemilihan titik baru dilakukan bersamaan dengan pemilihan sisi sehingga didapatkan nilai penyisipan minimum[8].

Langkah-langkah pengerjaan algoritma *Cheapest Insertion Heuristic*:

1. Penelusuran dimulai dari lokasi pertama dihubungkan dengan lokasi terakhir.
2. Bangun *subtour* antara 2 lokasi. *Subtour* adalah perjalanan lokasi pertama berakhir di lokasi terakhir. Direpresentasikan dengan: $(s, d_3) \rightarrow (d_3, d_1) \rightarrow (d_1, d_2) \rightarrow (d_2, \ldots) \rightarrow (d_x, s)$
   Dimana s adalah titik awal, d_1, d_2, d_3, …, d adalah tempat tujuan hingga tempat ke-n
3. Ganti arah hubungan busur dari repreentasi dua lokasi dengan kombinasi dua busur, yaitu busur (*i, j*) dengan busur (*i, k*) dan busur (*k, j*), *i* merupakan titik busur awal, *j* merupakan titik busur yang dituju dan *k* merupakan titik busur yang belum masuk subtour dan dengan nilai sisipan terkecil. Penentuan nilai sisipan dengan cara:
   $C_{ik} + C_{kj} - C_{ij}$, …… (1)
   C_{ik} adalah jarak dari lokasi *i* ke lokasi *k*, , C_{kj} adalah jarak dari lokasi *k* ke lokasi *j*, dan C_{ij} adalah jarak dari lokasi *i* ke lokasi *j*.
4. Ulangi langkah ke 3 hingga seluruh tempat destinasi telah dimasuki dalam *subtour*.

2.3. Matrix Jarak

Jarak ini merupakan selisih dari posisi awal dan posisi akhir. Pada pengukuran jarak ini dapat diselesaikan menggunakan metode perhitungan Haversine yang menggunakan rumus (2).

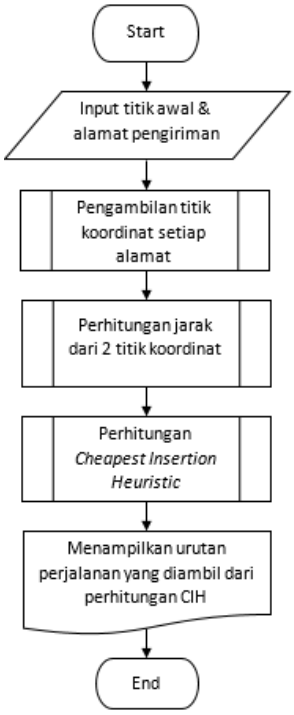
$$\Delta\partial = 2arcsin\left(\sqrt{sin^2\left(\frac{\Delta\emptyset}{2}\right) + \cos\emptyset_s \cos\emptyset_f \, sin^2\left(\frac{\Delta\lambda}{2}\right)}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Dimana:

$\Delta\sigma$ / d = *Interior Spherical Angle* / Jarak; $\Delta\phi$ = Latitude2 – Latitude1; Φ_s = Latitude 1; Φ_f = Latitude 2; $\Delta\lambda$ = Longtitude2 – Longtitude1; R = radius bumi (rata-rata radius = 6.371 km).

3. METODOLOGI PENELITIAN

3.1. Perancangan Sistem



Gambar 1. *Flowchart Sistem*

Rancangan system *flowchart system* nampak pada gambar 1. *Flowchart system* identifikasi awal alamat pengiriman beserta nilai koordinatnya Latitude dan Longitude memanfaatkan Google Maps. Nilai-nilai tersebut dimasukkan dalam matrix jarak 2 dimensi, dihitung jarak masing-masing koordinat mewakili masing-masing alamat. Kemudian dilakukan perhitungan CIH.

3.2. *Flowchart* Sub Sistem Request Koordinat Alamat Pengantaran

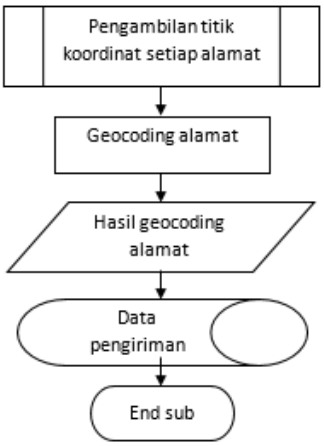
Data-data alamat pengiriman kurir didapatkan dari sebuah perusahaan ekspedisi PT.XYZ di kota Surabaya, yang area pengirimannya mencakup di sekitar area kota Surabaya. Sebagai contoh, data alamat-alamat pengiriman kurir seperti yang terlihat pada tabel 1.

Tabel 1. *Alamat Pengantaran Kurir*

No	Alamat Paket
1	Jl. suko semolo VIII F 23,
2	Jl. Lebak indah jaya I no.27
3	Jl. Mulyosari prima 12A no.26
4	Jl. Manyar kertoadi no 29
5.	Jl. Klampis indah gg I no. 6
6	Jl. Laguna kejawan putih no.7
7	Jl. Klimbungan gg IV no. 5
8	Jl. kedung mangu selatan 5/21

Flowchart sub system pengambilan titik koordinat merupakan proses *request* ke Google Maps menggunakan fitur Geocoding *Application Programing Interface* (API). Hasilnya berupa nilai *Latitude* dan *Longtitude* masing-masing alamat paket pengantaran kurir. Google Maps adalah *platform* peta digital, namun Google maps menghubungan data dan peta geografis dari produsen

dan pengguna data geografis [9]. Google maps API berupa library yang menggunakan kode bahasa pemrograman *JavaScript* [10]. *Flowchart* sub system ini pengambilan nilai koordinat disajikan pada gambar 2.



Gambar 2. *Flowchart Sub System Request* Koordinat Alamat

Gambar 2 dijelaskan *request* ke Google maps menggunakan API memanfaatkan *library* google maps API yang digunakan. *Library* tersebut yaitu [11]: **(1) Geocoding**: merupakan *library* Google Maps API digunakan mendapatkan nilai koordinat. Fungsi mencari *geocoding* ditemukan pada *Geocoder object*. *Geocode object* memiliki satu *method*: geocode(), *method* ini mengembalikan dua nilai fungsi *callback*: *result* dan *status*. **(2) Geometry**: merupakan *object* Google Maps API memiliki *property* yang penting yaitu *location*, berisikan nilai posisi alamat sebagai *LatLngobject*, mempunyai nilai *latitude* dan *longitude* sebuah alamat. **(3) *request* Address**, fitur distance matrix API merupakan layanan menyediakan jarak dan waktu perjalanan antara alamat asal dan alamat tujuan. Elemen Google maps *distance matrix* API: *origin_address*, *destination_address*, *distance*.

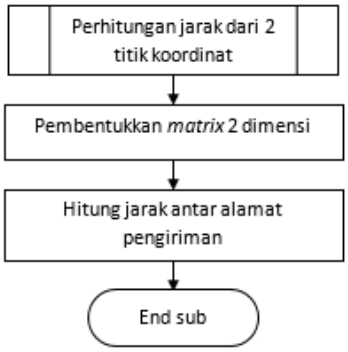
Hasil dari proses *request* koordinat alamat yaitu nilai *latitude, longitude* beserta alamat valid yang disebut *call back value* yang nampak pada tabel 2.

Tabel 1. *Informasi Data Latitude dan Longitude Alamat Pengiriman Kurir*

No	Alamat	Lattitude	Longtitude
1	Jl. Suko Semolo VIII Bok F No. 23, Sukolilo, Kota SBY, Jawa Timur 60119, Indonesia	-7.30285	112.7808
2	Jl. Lb. Indah Jaya I No. 27, Tambaksari, Kota SBY, Jawa Timur 60134, Indonesia	-7.24225	112.783
3	Jalan Mulyosari Prima, Mulyorejo, Kota Surabaya, Jawa Timur 60116	-7.26011	112.79968
4	Jl. Manyar Kertoadi, Klampis Ngasem, Kec. Sukolilo , Kota SBY, Jawa Timur 60116	-7.28217	112.77837
5.	Jl. Klampis Indah I No. 6, Sukolilo, Kota SBY, Jawa Timur 60117, Indonesia	-7.2945	112.7753
6	Jl. Lagguna Kjw Putih, Mulyorejo, Kota SBY, Jawa Timur 60112, Indonesia	-7.2769	112.811
7	Jl. Klimbungan IV No. 5, Genteng, Kota SBY, Jawa Timur 60274, Indonesia	-7.24862	112.746
8	Jl. Kedung Mangu Sel. V No. 21, Kejneran, Kota SBY, Jawa Timur 60128, Indonesia	-7.22742	112.76

3.3. *Flowchart* Sub Sistem Perhitungan Jarak

Flowchart sub system perhitungan jarak ini biasa disebut perhitungan Matrix jarak. Matrik jarak ini bentuknya matrix 2 dimensi yang berisikan nilai dua titik koordinat alamat [12]. Data-data pengiriman kurir yang sebelumnya telah, selanjutnya dijadikan sebuah matriks 2 dimensi untuk memudahkan penghitungan matriks jarak. *Flowchart* sub system perhitungan jarak dengan datanya dari nilai koordinat masing-masing alamat pengantaran kurir disajikan pada gambar 3.



Gambar 2. *Diagram Alir Menghitung Jarak*

3.4. Pembentukan *Matrix* Jarak Antar Alamat

Tabel 2. *Matrix Jarak*

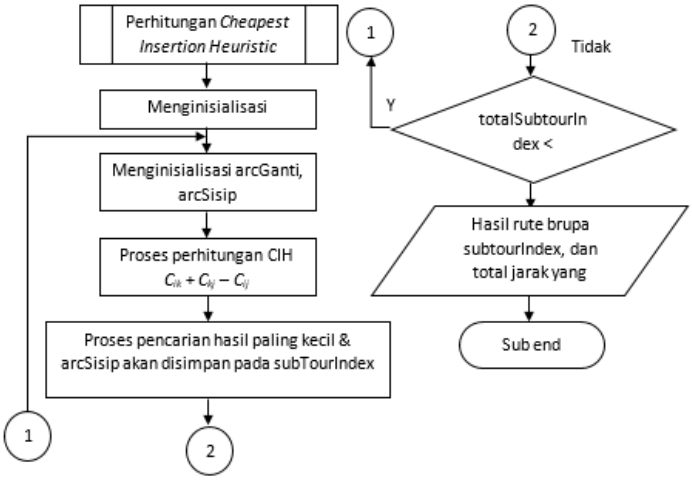
	s	d1	d2	d3	d4	d5	d6	d7	d8
s	0	8.064	11.87	11.97	8.675	7.773	12.16	8.936	11.74
d1	8.064	0	6.743	5.189	2.315	1.109	4.407	7.148	8.696
d2	11.87	6.743	0	2.707	4.468	5.872	4.938	4.142	3.026
d3	11.97	5.189	2.707	0	3.397	4.675	2.246	6.057	5.69
d4	8.675	2.315	4.468	3.397	0	1.412	3.646	5.164	6.416
d5	7.773	1.109	5.872	4.675	1.412	0	4.397	6.039	7.647
d6	12.16	4.407	4.938	2.246	3.646	4.397	0	7.829	7.869
d7	8.936	7.148	4.142	6.057	5.164	6.039	7.829	0	2.818
d8	11.74	8.696	3.026	5.69	6.416	7.647	7.869	2.818	0

Penulisan d1, d2, d3, dst, pada tabel 3 merupakan penulisan dari alamat yang terdapat pada tabel 1. Masing-masing alamat ini sesuai dengan baris dan kolomnya kemudian dilakukan perhitungan jarak antar masing-masing alamatnya menggunakan rumus Haversin (rumus 1). Hasil dari perhitungannya bisa dilihat pada tabel 3.

Pada tahapan ini titik awal (disimbolkan "s"), alamat kantor cabang Surabaya PT.XYZ berada di Jl. Raya Mastrip, Kedurus, Kec. Karang Pilang, Kota SBY, Jawa Timur 60223 dengan nilai *latitutde* -7,32056 dan *longitude* 112,7099. Selanjutnya, dari data-data tabel 1, dibuat matriks 2 dimensi menghitung antar koordinat masing-masing alamatnya (*Lattitude* dan *Longitude*). Hasil perhitungan antar koordinat menghasilkan matrik jarak [13]. *Distrance matrix* ditulis dengan $D = (d_{ij})$ adalah sebuah matriks nxn, dan d_{ij} adalah panjang dari $vertex^i$ dan $vertex^j$ [14].

3.5. *Flowchart Sub* Sistem Algoritma *Cheapest Insertion Heuristic* (CIH)

Flowchart sub system algoritma CIH, awalnya menginisialisai total subtour berisikan total keseluruhan jarak destinasi. Selanjutnya, inisialisasi ArcGanti dan Arcganti sebuah arc (edge atau sisi) di subtour (ditulis dengan arc(i,j)). ArcSisip adalah kombinasi dari 2 arcs, yaitu (i,k) dan (k,j), dengan k tidak berada dalam subtour saat ini. Selanjutnya menghitung jarak antar alamat-alamat pengiriman pada matrix mengunakan rumus Algoritma CIH, yaitu: Cik + Ckj – Cij. Perhitungan ini mencari nilai terkecil yang nantinya arcSisip diganti dengan nilai paling kecil akan disimpan pada subTour Index. Perhitungan ini terus berulang hingga total Subtour Index sama dengan total *subtour* yang sudah diinisialisasikan.



Gambar 3. *Perhitungan Algoritma Cheapest Insertion Heuristic*

3.6. Pembentukan Rute Pengiriman Menggunakan *Cheapest Insertion Heuristic* (CIH)

Pada bab ini ditunjukkan detail pembentukan rute pengiriman kurir dengan teknik penyisipan sesuai konsep algoritma CIH, berikut ini adalah langkah-langkahnya :

1. Membuat rute awal, diambil dari s (titik awal) dan nilai paling kecil antar tempat tujuan, yang dapat dituliskan dengan : (s, d1).
2. Masukkan rute tersebut kedalam *subtour*. Kondisi *subtour* saat ini: (s, d1).
3. Membuat sisipan baru dari setiap kombinasi diluar *subtour*, dapat dilihat pada Tabel 4.

Tabel 3. *Penyisipan terhadap subtour 1[7]*

Arc yang diganti	Penyisipan Arc ke subtour	Penambahan Jarak (km)
(s, d1)	(s, d2) → (d2, d1)	$C_{s,d2} + C_{d2,d1} - C_{s,d1}$ 11.87 + 6.743 – 8.064 = 10.55
(s, d1)	(s, d3) → (d3, d1)	$C_{s,d3} + C_{d3,d1} - C_{s,d1}$ 11.97 + 5.189 – 8.064 = 9.095
(s, d1)	(s, d4) → (d4, d1)	$C_{s,d4} + C_{d4,d1} - C_{s,d1}$ 8.675 + 2.315 – 8.064 = 2.926
(s, d1)	(s, d5) → (d5, d1)	Cs,d5 + Cd5,d1 - Cs,d1 7.773 + 1.109 – 8.064 = 0.818
(s, d1)	(s, d6) → (d6, d1)	Cs,d6 + Cd6,d1 - Cs,d1 12.16 + 4.407 – 8.064 = 8.503
(s, d1)	(s, d7) → (d7, d1)	Cs,d7 + Cd7,d1 - Cs,d1 8.936 + 7.148 – 8.064 = 8.02
(s, d1)	(s, d8) → (d8, d1)	Cs,d8 + Cd8,d1 - Cs,d1 11.74 + 8.696 – 8.064 = 12.372

Hasil penambahan jarak atau bobot paling kecil penyisipan (s, d1) adalah 0.81. Maka Arc (s, d1) diganti dengan (s, d5) – (d5, d1). Kondisi *subtour* saat ini: (s, d5) – (d5, d1).

4. Memeriksa tujuan yang belum ada pada *subtour*, yaitu d2, d3, d4, d6, d7, d8. Dibuat nilai sisipan terbaru dari setiap kombinasi diluar *subtour*, dapat dilihat pada Tabel 5.

Tabel 4. *Penyisipan terhadap subtour 2[7]*

Arc yang diganti	Penyisipan Arc ke subtour	Penambahan Jarak (km)
(s, d5)	(s, d2) → (d2, d5)	$C_{s,d2} + C_{d2,d5} - C_{s,d5}$ 11.87 + 5.872 – 7.773 = 9.969
(s, d5)	(s, d3) → (d3, d5)	$C_{s,d3} + C_{d3,d5} - C_{s,d5}$ 11.97 + 4.675 – 7.773 = 8.772
(s, d5)	(s, d4) → (d4, d5)	$C_{s,d4} + C_{d4,d5} – C_{s,d5}$ 8.675 + 1.412 – 7.773 = 2.314
(s, d5)	(s, d6) → (d6, d5)	Cs,d6 + Cd6,d5 – Cs,d5 12.16 + 4.397 – 7.773 = 8.784

(s, d5)	(s, d7) → (d7, d5)	Cs,d7 + Cd7,d5 – Cs,d5 8.936 + 6.039 – 7.773 = 7.202
(s, d5)	(s, d8) → (d8, d5)	Cs,d8 + Cd8,d5 – Cs,d5 11.74 + 7.647 – 7.773 = 11.614
(d5, d1)	(d5, d2) → (d2, d1)	Cd5,d2 + Cd2,d1 - Cd5,d1 5.872 + 6.743 – 1.109 = 11.506
(d5, d1)	(d5, d3) → (d3, d1)	Cd5,d3 + Cd3,d1 - Cd5,d1 4.675 + 5.189 – 1.109 = 8.755
(d5, d1)	(d5, d4) → (d4, d1)	Cd5,d4 + Cd4,d1 - Cd5,d1 1.412 + 2.315 – 1.109 = 2.618
(d5, d1)	(d5, d6) → (d6, d1)	Cd5,d6 + Cd6,d1 - Cd5,d1 4.397 + 4.407 – 1.109 = 7.695
(d5, d1)	(d5, d7) → (d7, d1)	Cd5,d7 + Cd7,d1 - Cd5,d1 6.039 + 7.148 – 1.109 = 12.078
(d5, d1)	(d5, d8) → (d8, d1)	Cd5,d8 + Cd8,d1 - Cd5,d1 7.647 + 8.696 – 1.109 = 15.234

Hasil penambahan jarak atau bobot paling kecil penyisipan adalah 2.618. Maka Arc (d5, d1) diganti dengan (d5, d4) → (d4, d1). Kondisi subtour saat ini: (s, d5) → (d5, d4) → (d4, d1).

5. Memeriksa tujuan yang belum ada pada *subtour*, yaitu d2, d3, d6, d7, d8. Dibuat nilai sisipan terbaru dari setiap kombinasi diluar *subtour*, dapat dilihat pada Tabel 6.

Tabel 6. *Penyisipan terhadap subtour 3[7]*

Arc yang diganti	Penyisipan Arc ke subtour	Penambahan Jarak (km)
(d5, d4)	(d5, d2) → (d2, d4)	$C_{d5,d2} + C_{d2,d4} - C_{d5,d4}$ 5.872 + 4.468 – 1.412 = 8.928
(d5, d4)	(d5, d3) → (d3, d4)	$C_{d5,d3} + C_{d3,d4} - C_{d5,d4}$ 5.872 + 3.397 – 1.412 = 7.857
(d5, d4)	(d5, d6) → (d6, d4)	$C_{d5,d6} + C_{d6,d4} - C_{d5,d4}$ 4.397 + 3.646 – 1.412 = 6.631
(d5, d4)	(d5, d7) → (d7, d4)	$C_{d5,d7} + C_{d7,d4} - C_{d5,d4}$ 6.039 + 5.164 – 1.412 = 9.791
(d5, d4)	(d5, d8) → (d8, d4)	$C_{d5,d8} + C_{d8,d4} - C_{d5,d4}$ 7.647 + 6.416 – 1.412 = 12.651
(d4, d1)	(d4, d2) → (d2, d1)	$C_{d4,d2} + C_{d2,d1} - C_{d4,d1}$ 4.468 + 6.743 – 2.315 = 8.896
(d4, d1)	(d4, d3) → (d3, d1)	$C_{d4,d3} + C_{d3,d1} - C_{d4,d1}$ 3.397 + 5.189 – 2.315 = 6.271
(d4, d1)	(d4, d6) → (d6, d1)	$C_{d4,d6} + C_{d6,d1} - C_{d4,d1}$ 3.646 + 4.407 – 2.315 = 5.738
(d4, d1)	(d4, d7) → (d7, d1)	$C_{d4,d7} + C_{d7,d1} - C_{d4,d1}$ 5.164 + 7.148 – 2.315 = 9.997
(d4, d1)	(d4, d8) → (d8, d1)	$C_{d4,d8} + C_{d8,d1} - C_{d4,d1}$ 6.416 + 8.696 – 2.315 = 12.797
(s, d5)	(s, d2) → (d2, d5)	$C_{s,d2} + C_{d2,d5} – C_{s,d5}$ 11,87 + 5.872 – 7.773 = 9.969
(s, d5)	(s, d3) → (d3, d5)	$C_{s,d3} + C_{d3,d5} – C_{s,d5}$ 11.97 + 4.675 – 7.773 = 8.872
(s, d5)	(s, d6) → (d6, d5)	Cs,d6 + Cd6,d5 – Cs,d5 12.16 + 4.397 – 7.773 = 8.789
(s, d5)	(s, d7) → (d7, d5)	Cs,d7 + Cd7,d5 – Cs,d5 8.936 + 6.039 – 7.773 = 7.202
(s, d5)	(s, d8) → (d8, d5)	Cs,d8 + Cd8,d5 – Cs,d5 11.74 + 7.647 – 7.773 = 11.614

Hasil penambahan jarak atau bobot paling kecil penyisipan adalah 5.738. Maka Arc (d4, d1) diganti dengan (d4, d6) → (d6, d1). Kondisi subtour saat ini: (s, d5) → (d5, d4) → (d4, d6) → (d6, d1).

6. Memeriksa tujuan yang belum ada pada *subtour*, yaitu d2, d3, d7, d8. Dibuat nilai sisipan terbaru dari setiap kombinasi diluar *subtour*, dapat dilihat pada Tabel 7.

Tabel 7. *Penyisipan terhadap subtour 4[7]*

Arc yang diganti	Penyisipan Arc ke subtour	Penambahan jarak (km)
(s, d5)	(s, d2) → (d2, d5)	$C_{s,d2} + C_{d2,d5} - C_{s,d5}$ 11.87 + 5.872 – 7.773 = 9.969
(s, d5)	(s, d3) → (d3, d5)	$C_{s,d3} + C_{d3,d5} - C_{s,d5}$ 11.97 + 4.675 – 7.773 = 8.872
(s, d5)	(s, d7) → (d7, d5)	$C_{s,d7} + C_{d7,d5} - C_{s,d5}$ 8.936 + 6.039 – 7.773 = 7.202
(s, d5)	(s, d8) → (d8, d5)	$C_{s,d8} + C_{d8,d5} - C_{s,d5}$ 11.74 + 7.647 – 7.773 = 11.614
(d5, d4)	(d5, d2) → (d2, d4)	$C_{d5,d2} + C_{d2,d4} - C_{d5,d4}$ 5.872 + 4.468 – 1.412 = 8.928
(d5, d4)	(d5, d3) → (d3, d4)	$C_{d5,d3} + C_{d3,d4} - C_{d5,d4}$ 4.675 + 3.397 – 1.412 = 6.66
(d5, d4)	(d5, d7) → (d7, d4)	$C_{d5,d7} + C_{d7,d4} - C_{d5,d4}$ 6.039 + 5.164 – 1.412 = 9.791
(d5, d4)	(d5, d8) → (d8, d4)	$C_{d5,d8} + C_{d8,d4} - C_{d5,d4}$ 7.647 + 6.416 – 1.412 = 12.651
(d4, d6)	(d4, d2) → (d2, d6)	$C_{d4,d2} + C_{d2,d6} - C_{d4,d6}$ 4.468 + 4.938 – 3.646 = 5.76
(d4, d6)	(d4, d3) - (d3, d6)	$C_{d4,d3} + C_{d3,d6} - C_{d4,d6}$ 3.397 + 2.246 – 3.646 = 1.997
(d4, d6)	(d4, d7) → (d7, d6)	$C_{d4,d7} + C_{d7,d6} - C_{d4,d6}$ 5.164 + 7.829 – 3.646 = 9.347
(d4, d6)	(d4, d8) → (d8, d6)	$C_{d4,d8} + C_{d8,d6} - C_{d4,d6}$ 6.416 + 7.869 – 3.646 = 10.639
(d6, d1)	(d6, d2) → (d2, d1)	$C_{d6,d2} + C_{d2,d1} - C_{d6,d1}$ 4.938 + 6.743 – 4.407 = 7.274
(d6, d1)	(d6, d3) → (d3, d1)	$C_{d6,d3} + C_{d3,d1} - C_{d6,d1}$ 2.246 + 5.189 – 4.407 = 3.028
(d6, d1)	(d6, d7) → (d7, d1)	$C_{d6,d7} + C_{d7,d1} - C_{d6,d1}$ 7.829 + 7.148 – 4.407 = 10.57
(d6, d1)	(d6, d8) → (d8, d1)	$C_{d6,d8} + C_{d8,d1} - C_{d6,d1}$ 7.869 + 8.696 – 4.407 = 12.158

Hasil penambahan jarak atau bobot paling kecil penyisipan adalah 1.997. Maka Arc (d4, d6) diganti dengan (d4, d3) → (d3, d6). Kondisi subtour saat ini: (s, d5) → (d5, d4) → (d4, d3) → (d3, d6) → (d6, d1).

7. Memeriksa tujuan yang belum ada pada *subtour*, yaitu d2, d7, d8. Dibuat nilai sisipan terbaru dari setiap kombinasi diluar *subtour*, dapat dilihat pada Tabel 8**.**

Tabel 8. *Penyisipan terhadap subtour 5[7]*

Arc yang diganti	Penyisipan Arc ke subtour	Penambahan Jarak (km)
(s, d5)	(s, d2) → (d2, d5)	$C_{s,d2} + C_{d2,d5} - C_{s,d5}$ 11.87 + 5.872 – 7.773 = 9.969
(s, d5)	(s, d7) → (d7, d5)	$C_{s,d7} + C_{d7,d5} - C_{s,d5}$ 8.936 + 6.039 – 7.773 = 7.202
(s, d5)	(s, d8) → (d8, d5)	$C_{s,d8} + C_{d8,d5} - C_{s,d5}$ 11.74 +7.647 – 7.773 = 11.614
(d5, d4)	(d5, d2) → (d2, d4)	Cd5,d2 + Cd2,d4 - Cd5,d4 5.872 + 4.468 – 1.412 = 8.928
(d5, d4)	(d5, d7) → (d7, d4)	Cd5,d7 + Cd7,d4 - Cd5,d4 6.039 + 5.164 – 1.412 = 9.791
(d5, d4)	(d5, d8) → (d8, d4)	Cd5,d8 + Cd8,d4 - Cd5,d4 7.647 + 6.416 – 1.412 = 12.651
(d4, d3)	(d4, d2) → (d2, d3)	Cd4,d2 + Cd2,d3 - Cd4,d3 4.468 + 2.707 – 3.397 = 3.778
(d4, d3)	(d4, d7) → (d7, d3)	Cd4,d7 + Cd7,d3 - Cd4,d3

		5.164 + 6.057 – 3.397 = 7.824
(d4, d3)	(d4, d8) → (d8, d3)	Cd4,d8 + Cd8,d3 - Cd4,d3 6.416 + 5.69 – 3.397 = 8.709
(d3, d6)	(d3, d2) → (d2, d6)	Cd3,d2 + Cd2,d6 - Cd3,d6 2.707 + 4.938 – 2.246 = 5.399
(d3, d6)	(d3, d7) → (d7, d6)	Cd3,d7 + Cd7,d6 - Cd3,d6 6.057 + 7.829- 2.246 = 11.64
(d3, d6)	(d3, d8) → (d8, d6)	Cd3,d8 + Cd8,d6 - Cd3,d6 5.69 + 7.869 – 2.246 = 11.313
(d6, d1)	(d6, d2) → (d2, d1)	Cd6,d2 + Cd2,d1 - Cd6,d1 4.938 + 6.743 – 4.407 = 7.274
(d6, d1)	(d6, d7) → (d7, d1)	Cd6,d7 + Cd7,d1 - Cd6,d1 7.829 + 7.148 – 4.407 = 10.57
(d6, d1)	(d6, d8) → (d8, d1)	Cd6,d8 + Cd8,d1 - Cd6,d1 7.869 + 8.696 – 4.407 = 12.158

Hasil penambahan jarak atau bobot paling kecil penyisipan adalah 3.778. Maka Arc (d4, d3) diganti dengan (d4, d2) → (d2, d3). Kondisi subtour saat ini: (s, d5) → (d5, d4) → (d4, d2) → (d2, d3) → (d3, d6) → (d6, d1).

8. Memeriksa tujuan yang belum ada pada *subtour*, yaitu d7, d8. Dibuah nilai sisipan terbaru dari setiap kombinasi diluar *subtour*, dapat dilihat pada Tabel 9.

Tabel 9. *Penyisipan terhadap subtour 6[7]*

Arc yang diganti	Penyisipan Arc ke subtour	Penambahan Jarak (km)
(s, d5)	(s, d7) → (d7, d5)	$C_{s,d7} + C_{d7,d5} - C_{s,d5}$ 8.936 + 6.039 – 7.773 = 7.202
(s, d5)	(s, d8) → (d8, d5)	$C_{s,d8} + C_{d8,d5} - C_{s,d5}$ 11.74 + 7.647 – 7.773 = 11.614
(d5, d4)	(d5, d7) → (d7, d4)	$C_{d5,d7} + C_{d7,d4} - C_{d5,d4}$ 6.039 + 5.164 – 1.412 = 9.791
(d5, d4)	(d5, d8) → (d8, d4)	$C_{d5,d8} + C_{d8,d4} - C_{d5,d4}$ 7.647 + 6.416 – 1.412 = 12.651
(d4, d2)	(d4, d7) → (d7, d2)	$C_{d4,d7} + C_{d7,d}2 - C_{d4,d2}$ 5.164 + 4.142 – 4.468 = 4.838
(d4, d2)	(d4, d8) → (d8, d2)	$C_{d4,d8} + C_{d8,d2} - C_{d4,d2}$ 6.416 + 3.026 – 4.468 = 4.974
(d2, d3)	(d2, d7) → (d7, d3)	$C_{d2,d7} + C_{d7,d3} - C_{d2,d3}$ 4.142 + 6.057 – 2.707 = 7.492
(d2, d3)	(d2, d8) → (d8, d3)	$C_{d2,d8} + C_{d8,d3} - C_{d2,d3}$ 3.026 + 5.69 – 2.707 = 6.009
(d3, d6)	(d3, d7) → (d7, d6)	$C_{d3,d7} + C_{d7,d6} - C_{d3,d6}$ 6.057 + 7.829 – 2.246 = 11.64
(d3, d6)	(d3, d8) → (d8, d6)	Cd3,d8 + Cd8,d6 - Cd3,d6 5.69 + 7.869 – 2.246 = 11.313
(d6, d1)	(d6, d7) → (d7, d1)	Cd6,d7 + Cd7,d1 - Cd6,d1 7.829 + 7.148 – 4.407 = 10.57
(d6, d1)	(d6, d8) → (d8, d1)	Cd6,d8 + Cd8,d1 - Cd6,d1 7.869 + 8.696 – 4.407 = 12.158

Hasil penambahan jarak atau bobot paling kecil penyisipan adalah 4.838. Maka Arc (d4, d2) diganti dengan (d4, d7) → (d7, d2). Kondisi subtour saat ini: (s, d5) → (d5, d4) → (d4, d7) → (d7, d2) → (d2, d3) → (d3, d6) → (d6, d1).

9. Memeriksa tujuan yang belum ada pada *subtour*, yaitu d8. Kemudian dibuat nilai sisipan terbaru dari setiap kombinasi diluar *subtour*, dapat dilihat pada Tabel 10.

Tabel 10. *Penyisipan terhadap subtour 7[7]*

Arc yang akan diganti	Arc yang akan disisipkan ke subtour	Penambahan jarak (km)
(s, d5)	(s, d8) → (d8, d5)	$C_{s,d8} + C_{d8,d5} - C_{s,d5}$ 11.74 + 7.647 – 7.773 = 11.614

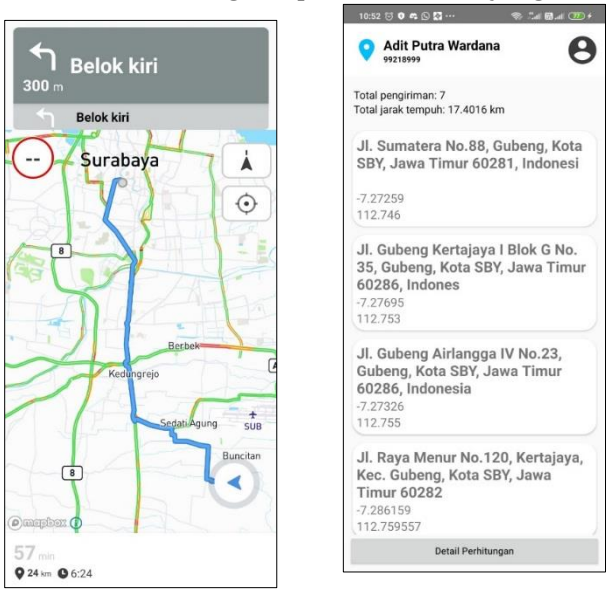
(d5, d4)	(d5, d8) → (d8, d4)	$C_{d5,d8} + C_{d8,d4} - C_{d5,d4}$ 7.647 + 6.416 – 1.412 = 12.651
(d4, d7)	(d4, d8) → (d8, d7)	$C_{d4,d8} + C_{d8,d7} - C_{d4,d7}$ 6.416 + 2.818 – 5.164 = 4.07
(d7, d2)	(d7, d8) → (d8, d2)	Cd7,d8 + Cd8,d2 - Cd7,d2 2.818 + 3.026 – 4.142 = 1.702
(d2, d3)	(d2, d8) → (d8, d3)	Cd2,d8 + Cd8,d3 - Cd2,d3 3.026 + 5.69 – 2.707 = 6.009
(d3, d6)	(d3, d8) → (d8, d6)	Cd3,d8 + Cd8,d6 - Cd3,d6 5.69 + 7.869 – 2.246 = 11.313
(d6, d1)	(d6, d8) → (d8, d1)	Cd6,d8 + Cd8,d1 - Cd6,d1 7.869 + 8.696 – 4.407 = 12.158

Hasil penambahan jarak atau bobot paling kecil penyisipan adalah 1.702. Maka Arc (d7, d2) diganti dengan (d7, d8) → (d8, d2). Kondisi subtour saat ini: (s, d5) → (d5, d4) → (d4, d7) → (d7, d8) → (d8, d2) → (d2, d3) → (d3, d6) → (d6, d1).

10. Hasil perhitungan algoritma CIH yaitu, urutan-urutan yang diambil *subtour route* perjalanan: (s, d5) → (d5, d4) → (d4, d7) → (d7, d8) → (d8, d2) → (d2, d3) → (d3, d6) → (d6, d1).

3.7. Implementasi Algoritma CIH Pada Smartphone Android

Hasil penelitian ini diimplementasikan pada *smartphone* Android yang membentuk sebuah rute pengiriman kurir, yang nampak pada gambar 5(a). Tampilan pada *smartphone* berupa urutan alamat-alamat pengiriman paket nampak pada gambar 5(b). Aplikasi ini digunakan sebagai panduan kurir yang merasa kesulitan mengirim paket ke alamat yang sesuai.



(a) (b)

Gambar 5. (a) *Tampilan Aplikasi Rute Pengiriman, (b) Tampilan Aplikasi Urutan Pengiriman*

4. PENGUJIAN DAN PEMBAHASAN

Pengujian pertama terkait perhitungan akhir algoritma CIH adalah untuk mendapatkan total *maximum distance* [15], jarak tempuhnya sebesar 29,553 Km. Untuk melakukan pengujian sistem dilakukan perbandingan *total maximum distance* antara hasil yang didapatkan dari perhitungan algoritma CIH dengan total jarak sesungguhnya. Hasilnya pada tabel 11, menunjukkan jarak tempuh rute terbentuk menggunakan algoritma CIH lebih kecil dibadingkan dengan jarak tempuh sesuungguhnya, yaitu 29,553 Km dengan 37,767 Km.

Tabel 51. *Perbandingan total jarak rute pengiriman tanpa menggunakan CIH dan menggunakan CIH*

Arc (tanpa CIH)	Jarak (tanpa CIH)	Arc (pakai CIH)	Jarak (pakai CIH)
(s, d1)	8.064	(s, d5)	7.773
(d1, d2)	6.743	(d5, d4)	1.412
(d2, d3)	2.707	(d4, d7)	5.164
(d3, d4)	3.397	(d7, d8)	2.818
(d4, d5)	1.412	(d8, d2)	3.026
(d5, d6)	4.397	(d2, d3)	2.707
(d6, d7)	7.829	(d3, d6)	2.246
(d7, d8)	2.818	(d6, d1)	4.407
	37.367		**29.553**

Pengujian yang kedua, pengujian keakuratan system. Pengujian ini menggunakan data-data alamat pengiriman paket sebanyak 33 alamat yang dibagi untuk 3 orang kurir. Pengujian ini bertujuan mengetahui keakuratan system melakukan pembentukan urutan dan perhitungan rute jarak tempuh yang paling kecil atau rute terpendek yang didapatkan dan mengetahui keefisiensian yang dihasilkan dari rute yang dibentuk dari system ini. Hasil dari pengujian lanjutan ini data-datanya disajikan pada tabel 12.

Tabel 62. *Pengujian keakuratan system antara total jarak rute pengiriman tanpa menggunakan CIH dan menggunakan CIH*

	Jarak tanpa CIH	**Jarak dengan CIH**	**Selisih Jarak (Km)**	**Efisiensi Jarak (persen)**
Kurir 1	28,85	18,99	9,87	34,2 %
Kurir 2	43,89	31,6	12,29	28 %
Kurir 3	14,67	11,54	3,127	21,31 %
	87,41	62,13	25,287	83,51 %

Hasil pengujian yang disajikan pada tabel 12 menunjukkan kurir 1 jarak tempuh yang didapatkan menggunakan perhitungan algoritma CIH lebih kecil dibadingkan dengan jarak tempuh sesungguhnya, yaitu 18,99 Km dengan 28,85 Km. Kurir 2 jarak tempuh yang didapatkan menggunakan perhitungan algoritma CIH lebih kecil dibadingkan dengan jarak tempuh sesungguhnya, yaitu 31,6 Km dengan 43,89 Km. Kurir 3 jarak tempuh yang didapatkan menggunakan perhitungan algoritma CIH lebih kecil dibadingkan dengan jarak tempuh sesungguhnya yaitu 11,54 Km dengan 14,67 Km. Efisiensi jarak tempuh yang dihasilkan menggunakan algoritma CIH terhadap jarak sesungguhnya untuk semua kurir tersebut yaitu, kurir 1 efiisiensi sebesar 34,2%, kurir 2 efisiensi sebesar 28%, dan kurir 3 efisiensi sebesar 21,31%.

5. KESIMPULAN

Dari penelitian *Traveling Salesman Problem* yang dilakukan menggunakan algoritma perhitungan dan perancangan sistem penerapan *Traveling Salesman Problem* dengan metode *Cheapest Insertion Heuristic* pada PT. XYZ dapat disimpulkan bahwa:

1. Penggunaan algoritma CIH untuk pembentukan rute dan pengurutan pengiriman paket ke alamat masing-masing, mampu mempersingkat rute tempuh perjaanan kurir ketika proses pengirirman paket.
2. Data-data alamat pengiriman kurir yang digunakan, alangkah baiknya dilakukan pre-processing terlebih dahulu, agar mempercepat proses respon dari server Google map unuk memberian informasi yang akurat sebuah alamat.
3. Penggunaan library Google maps API pada sistem Android memudahkan kurir menggunakan sistem ketikan mencari alamat paket.
4. Penelitian ini menghasilkan nilai efisiensi jarak tempuh semua kurir yaitu sebesaar 83,51%.

DAFTAR PUSTAKA

[1] G. Mediatama, “BPS catat penjualan online melonjak tajam selama pandemi corona,”

kontan.co.id, Jun. 02, 2020. https://nasional.kontan.co.id/news/bps-catat-penjualan-online-melonjak-tajam-selama-pandemi-corona (accessed Jan. 09, 2023).
[2] S. Riyadi and I. Nurhaida, "Aplikasi Sistem Virtual Tour E-Panorama 360 Derajat Berbasis Android Untuk Pengenalan Kampus Mercu Buana," J. Teknol. Inf. Dan Ilmu Komput., vol. 9, no. 1, p. 17, Feb. 2022, doi: 10.25126/jtiik.2021864209.
[3] F. R. Fargiana, "Implementasi Algoritma Cheapest Insertion Heuristic dalam Menentukan Rute Pengiriman Barang," J. Ris. Mat., vol. 1, no. 2, pp. 129–136, 2021, doi: https://doi.org/ 10.29313/jrm.v1i2.483.
[4] S. Darina, A. T. Wibowo, and M. Ridwan, "Penggunaan Algoritma Simulated Annealing Untuk Menyelesaikan Masalah Vehicle Routing Pada Rute Distribusi Supermarket Simulated Annealing Algorithm For Solving Vehicle Routing Problems On Supermarket Distribution Routes," vol. 6, no. 2, 2021.
[5] P. D. Istiqomah and W. A. Kusuma, "Sistem Informasi Geografis Kurir Asi Di Kota Malang Berbasis Website (Studi Kasus : Simomi)," Tek. Eng. Sains J., vol. 2, no. 1, p. 25, Jun. 2018, doi: 10.51804/tesj.v2i1.224.25-32.
[6] S. Riyadi and I. Nurhaida, "Aplikasi Sistem Virtual Tour E-Panorama 360 Derajat Berbasis Android Untuk Pengenalan Kampus Mercu Buana," J. Teknol. Inf. Dan Ilmu Komput. JTIIK, vol. 9, no. 1, pp. 17–24, Jan. 2022, doi: http://dx.doi.org/10.25126/jtiik.2021864209.
[7] S. Khadafi, "Implementasi Algoritma Pso Untuk Probabilitas Urutan Pengiriman Paket Pengantaran Kurir," p. 6, 2016.
[8] I. B. Gede Dwidasmara, I. G. N. A. W. Putra, I. M. Widiartha, I. W. Santiyasa, I. B. Made Mahendra, and A. A. I. Ngurah Eka Karyawati, "Sistem Rekomendasi Tempat Wisata Menggunakan Algoritma Cheapest Insertion Heuristic Dan Naïve Bayes," JELIKU J. Elektron. Ilmu Komput. Udayana, vol. 10, no. 2, p. 227, Jan. 2022, doi: 10.24843/JLK.2021.v10.i02.p05.
[9] J.-C. Plantin, "Google Maps as Cartographic Infrastructure: From Participatory Mapmaking to Database Maintenance," Int. J. Commun., vol. 12, pp. 489–506, 2018.
[10] "Google Maps Platform API berdasarkan Platform," Google Developers. https://developers.google.com/maps/apis-by-platform?hl=id (accessed Jan. 10, 2023).
[11] T. J. Pattiasina, E. T. Setyoadi, and D. Wijayanto, "Saving Matrix Method for Efficient Distribution Route Based on Google Maps API," J. Telecommun. Electron. Comput. Eng., vol. 10, no. 2, pp. 183–188, Jun. 2018.
[12] D. C. Nugraha and S. Khadafi, "Penerapan Travelling Salesman Problem Untuk Optimasi Jarak Jalur Kurir Menggunakan Algoritma Ant Colony Optimization (Aco)," Semin. Nas. Sains Dan Teknol. Terap. IX 2021 - ITATS, pp. 259–266, 2021.
[13] S. L. Chasanah, E. Khairunnisa, M. Yusuf, and K. A. Sugeng, "Relationship between adjacency and distance matrix of graph of diameter two," Indones. J. Comb., vol. 5, no. 2, p. 63, Dec. 2021, doi: 10.19184/ijc.2021.5.2.1.
[14] M. Yusuf and K. A. Sugeng, "The relation between the square of the adjacency matrix and spectra of the distance matrix of a graph with diameter two," presented at the The 8th Annual Basic Science International Conference: Coverage of Basic Sciences toward the World's Sustainability Challanges, East Java, Indonesia, 2018, p. 060023. doi: 10.1063/1.5062787.
[15] D. Pertami, I. W. Nuarsa, and I. D. N. Nurweda Putra, "Pemetaan Perubahan Penggunaan Lahan Wilayah Pesisir Kecamatan Rungkut, Kota Surabaya, Tahun 2013 dan 2019," J. Mar. Res. Technol., vol. 5, no. 1, p. 10, Feb. 2022, doi: 10.24843/JMRT.2022.v05.i01.p03.