# **BAB III**

# **ANALISIS DAN PERANCANGAN SISTEM**

1. **Analisis Sistem**

Dalam penerapan *Traveling Salesman Problem* pada sistem yang akan dibangun ialah sistem informasi yang berbasis android dengan memanfaatkan *Google Maps API* yang berfungsi sebagai alat bantu dalam penentuan titik koordinat setiap destinasi, dan sebagai alat bantu dalam menampilkan denah jalan yang dituju atau biasa disebut juga dengan peta. Sistem akan menganalisa jarak yang ditempuh dari titik awal ke setiap destinasi, yang dimana nantinya sistem akan memberikan rute destinasi secara runtut dengan total jarak yang minimum. Sistem ini dilengkapi dengan fitur perhitungan *Cheapest Insertion Heuristic* yang digunakan untuk mendapatkan nilai jarak tempuh minimum.

1. **Analisis Kebutuhan**

Di dalam suatu proses sistem diperlukan identifikasi kebutuhan fungsional maupun non-fungsional guna untuk mengetahui fungsi-fungsi yang terdapat pada sistem, penjabaran kebutuhan tersebut sebagai berikut:

1. Kebutuhan Fungsional
2. Sistem dapat melakukan pencarian rute terpendek.
3. Sistem dapat menyajikan urutan perjalanan/tujuan dari jarak terpendek beserta waktu yang ditempuh.
4. Kebutuhan Non-Fungsional
5. Ponsel yang digunakan berbasis *Android.*
6. Penggunaan minimum OS. Andorid API 23 (Marshmellow)
7. **Diagram Alir Sistem**

Dalam suatu sistem diperlukannya alur diagram yang dimana dapat memudahkan pengguna dalam memahami jalannya sistem. Diagram alir sistem dapat dilihat pada **Gambar 3.1**.

**Gambar 3. 1** Diagram alir sistem

Start

Input titik awal & alamat pengiriman

Pengambilan titik koordinat setiap alamat

Perhitungan jarak dari 2 titik koordinat

Perhitungan *Cheapest Insertion Heuristic*

Menampilkan urutan perjalanan yang diambil dari perhitungan CIH

End

**Gambar 3. 1** Diagram alir sistem

1. **Pengambilan titik koordinat**

Dalam proses tahap pengambilan titik koordinat setiap lokasi ini dibantu dengan fitur *Google Maps Geocoding API* yang dimana berisikan nilai *latitude* dan *longtitude*. Berikut hasil dari Geocoding alamat pengiriman kurir yang dapat dilihat pada **Tabel 3.1**.

**Tabel 3. 1**

Hasil pengambilan titik koordinat

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

Dalam proses ini dilakukannya *HTTP Request* pada *Geocoding API* yang dimana nanti akan memberikan responses, kemudian responses yang berupa titik koordinat *latitude* dan *longtitude* akan dimasukkan kedalam sistem dan di simpan ke dalam *database*, Terdapat diagram alir untuk memudahkan mengetahui jalannya proses dalam penyimpanan data alamat pengiriman, dapat dilihat pada Gambar **3.2**.

End sub

Geocoding alamat

Hasil geocoding alamat

Data pengiriman

Pengambilan titik koordinat setiap alamat

**Gambar 3. 2** Pengambilan titik koordinat

1. ***Matrix* Jarak**

Matrix jarak merupakan matrix 2 dimensi yang berisikan nilai dua titik koordinat pada masing-masing alamat. Berikut diagram alir menghitung jarak yang dimana nanti hasil perhitungannya digunakan sebagai matrix 2 dimensi, dapat dilihat pada **Gambar 3.3**.

**Gambar 3. 3** Diagram alir menghitung jarak

Perhitungan jarak dari 2 titik koordinat

Pembentukkan *matrix* 2 dimensi

Hitung jarak antar alamat pengiriman

End sub

Pada tahap ini penulis memberikan contoh menghitung dengan titik koordinat di Jl. Raya Mastrip, Kedurus, Kec. Karang Pilang, Kota SBY, Jawa Timur 60223 sebagai s (titik awal) dengan *latitutde* -7,32056 dan *longitude* 112,7099 dan dihitung dengan Rumus Formula Haversin, rumus dapat dilihat pada **Gambar 2.3**.

Contoh perhitungan antar titik koordinat alamat titik s (titik awal) dan d1 (destinasi 1) sebagai berikut:

d(s,d1) =2\*6371 \* ASIN(SQRT((SIN((D5\*(3.14159/180)B5\*(3.14159/180))/2))^2+

COS(D5\*(3.14159/180))\*COS(B5\*(3.14159/180))\*SIN(((E5\*

(3.14159/80)-C5\*(3.14159/180))/2))^2))

d(s,d1) = 8.0638

Dalam bentuk format excel

d(s,d1) = 2 \* 6371 \* ASIN (SQRT((SIN((-7.30285\*(3.14159/180)- -7.32056\*

(3.14159/180))/2))^2+COS(-7.30285\*(3.14159/180))\*COS(-7.32056\*

(3.14159/180))\*SIN(((112.7808\*(3.14159/180)- 112.7099\*

(3.14159/180))/2))^2))

d(s,d1) = 8.0638

Penjabaran perhitungan dari excel (memasukkan nilai)

Hasil seluruh penghitungan jarak dapat dilihat pada **Tabel 3.2**.

**Tabel 3. 2**

Hasil perhitungan jarak antar destinasi

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **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 |

1. ***Cheapest Insertion Heuristic***

Tahap selanjutnya setelah menghitung jarak antar koordinat ialah melakukan perhitungan Algoritma *Cheapest Insertion Heuristic* yang nantinya akan mendapatkan nilai optimasi dari jalur pegiriman yang telah ditentukan. Dalam perhitungan Algoritma CIH dalam sistem ini diperlukannya langkah awal yaitu menginisialisai totalSubtour yang dimana berisikan total destinasi. Kemudian menginisialisai arcGanti yang dimaksud ialah sebuah arc(edge atau sisi) di subtour tersebut (katakanlah arc(*i,j*), sedangkan menginisialisasi arcSisip ini ialah kombinasi dari 2 arcs, yaitu (*i,k*) dan (*k,j*), dengan *k* tidak berada dalam subtour saat ini. Tahap selanjutnya dilakukannya perhitungan sesuai rumus Algoritma CIH, yaitu: *Cik* + *Ckj* – *Cij.*Dari hasil perhitungan yang sudah didapatkan, maka akan dicari nilai paling kecil yang dimana nantinya arcSisip dengan nilai paling kecil akan disimpan pada subTourIndex, dan perhitungan ini akan terus berulang hingga totalSubtourIndex sama dengan totalSubtour yang dari awal sudah diinisialisasikan. Saat semua proses sudah terpenuhi maka urutan rute destinasi akan ditampilkan.

Terdapat diagram alir perhitungan *Cheapest Insertion Heuristic* yang bisa dilihat pada **Gambar 3.5**.

Perhitungan *Cheapest Insertion Heuristic*

Menginisialisasi totalSubtour

Menginisialisasi arcGanti, arcSisip

Proses perhitungan CIH

*Cik* + *Ckj* – *Cij*

Proses pencarian hasil paling kecil & arcSisip akan disimpan pada subTourIndex

totalSubtourIndex < totalSubtour

Hasil rute brupa subtourIndex, dan total jarak yang ditempuh

Sub end

Ya

Tidak

**Gambar 3. 4** Perhitungan Algoritma Cheapest Insertion Heuristic

1. **Perhitungan Menggunakan Algoritma Cheapest Insertion Heuristic Dengan Studi Kasus Perhitungan**

Berikut ini adalah contoh scenario kasus dan penyelesaiannya dengan menggunakan Algoritma *Cheapest Insertion Heuristic*.

Langkah awal Admin telah melakukan inputan data pengiriman berupa file yang berformatkan excel atau csv. Di dalam file berisikan alamat-alamat pengiriman yang akan dituju. Data yang didapatkan terdiri dari 8 alamat tujuan, seperti yang terlihat pada **Tabel 3.3**.

**Tabel 3. 3**

Data alamat pengiriman

|  |  |
| --- | --- |
| **No** | **Alamat** |
| 1 | Jl. Suko Semolo VIII Bok F No. 23, Sukolilo, Kota SBY, Jawa Timur 60119, Indonesia |
| 2 | Jl. Lb. Indah Jaya I No. 27, Tambaksari, Kota SBY, Jawa Timur 60134, Indonesia |
| 3 | Jalan Mulyosari Prima, Mulyorejo, Kota Surabaya, Jawa Timur 60116 |
| 4 | Jl. Manyar Kertoadi, Klampis Ngasem, Kec. Sukolilo , Kota SBY, Jawa Timur 60116 |
| 5. | Jl. Klampis Indah I No. 6, Sukolilo, Kota SBY, Jawa Timur 60117, Indonesia |
| 6 | Jl. Lagguna Kjw Putih, Mulyorejo, Kota SBY, Jawa Timur 60112, Indonesia |
| 7 | Jl. Klimbungan IV No. 5, Genteng, Kota SBY, Jawa Timur 60274, Indonesia |
| 8 | Jl. Kedung Mangu Sel. V No. 21, Kejneran, Kota SBY, Jawa Timur 60128, Indonesia |

Dilakukannya inisialisasi *variable,* seperti yang terlihat pada **Tabel 3.5**.

**Tabel 3. 4**

Pemberian nama variabel

|  |  |
| --- | --- |
| **Variable Node** | **Keterangan** |
| s | Titik awal |
| d1 | Tempat tujuan dengan ID 1 |
| d2 | Tempat tujuan dengan ID 2 |
| d3 | Tempat tujuan dengan ID 3 |
| d4 | Tempat tujuan dengan ID 4 |
| d5 | Tempat tujuan dengan ID 5 |
| d6 | Tempat tujuan dengan ID 6 |
| d7 | Tempat tujuan dengan ID 7 |
| d8 | Tempat tujuan dengan ID 8 |

Kemudian dilakukannya perhitungan nilai dua titik koordinat yang dibentuk menjadi *matrix* jarak (*matrix* 2 dimensi) dengan bantuan *Google Maps Geocoding API*, telah didapatkan jarak antara satu tempat dengan tempat lainnya yang direpresentasika dalam bentuk tabel, seperti yang terlihat pada **Tabel 3.2**.

Selanjutnya akan dilakukannya proses algoritma *Cheapest Insertion Heuristic*, dengan menggunakan tabel jarak yang dapat dilihat pada **Tabel 3.2**. Berikut langkah-langkah yang dilakukan:

1. Membuat rute awal, diambil dari s (titik awal) dan nilai paling kecil antar tempat tujuan. Dapat dituliskan seperti berikut: (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 3.5**.

**Tabel 3. 5**

Iterasi 1 - Penyisipan terhadap subtour

|  |  |  |
| --- | --- | --- |
| **Arc yang akan diganti** | **Arc yang akan disisipkan ke subtour** | **Penambahan jarak (km)** |
| (s, d1) | (s, d2) → (d2, d1) | Cs,d2 + Cd2,d1 - Cs,d1  11.87 + 6.743 – 8.064 = 10.55 |
| (s, d1) | (s, d3) → (d3, d1) | Cs,d3 + Cd3,d1 - Cs,d1  11.97 + 5.189 – 8.064 = 9.095 |
| (s, d1) | (s, d4) → (d4, d1) | Cs,d4 + Cd4,d1 - Cs,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 |

Dari tabel 3.6, dapat diambil bahwa hasil dari penambahan jarak atau bobot paling kecil dari penyisipan (s, d1) ialah 0.819. Maka Arc (s, d1) diganti dengan (s, d5) – (d5, d1). Kondisi *subtour* saat ini: (s, d5) – (d5, d1).

1. Melakukan pengecekan tempat tujuan yang belum ada pada *subtour*, dan terdapat pada d2, d3, d4, d6, d7, d8. Kemudian dibuatlah nilai sisipan terbaru dari setiap kombinasi diluar *subtour*. Dapat dilihat pada **tabel 3.6**.

**Tabel 3. 6**

Iterasi 2 - Penyisipan terhadap subtour

|  |  |  |
| --- | --- | --- |
| **Arc yang akan diganti** | **Arc yang akan disisipkan ke subtour** | **Penambahan jarak (km)** |
| (s, d5) | (s, d2) → (d2, d5) | Cs,d2 + Cd2,d5 - Cs,d5  11.87 + 5.872 – 7.773 = 9.969 |
| (s, d5) | (s, d3) → (d3, d5) | Cs,d3 + Cd3,d5 - Cs,d5  11.97 + 4.675 – 7.773 = 8.772 |
| (s, d5) | (s, d4) → (d4, d5) | Cs,d4 + Cd4,d5 – Cs,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 |

Dari tabel 3.6, dapat diambil bahwa hasil dari penambahan jarak atau bobot paling kecil dari penyisipan ialah 2.618. Maka Arc (d5, d1) diganti dengan (d5, d4) → (d4, d1). Kondisi *subtour* saat ini: (s, d5) → (d5, d4) → (d4, d1).

1. Melakukan pengecekan tempat tujuan yang belum ada pada *subtour*, dan terdapat pada d2, d3, d6, d7, d8. Kemudian dibuatlah nilai sisipan terbaru dari setiap kombinasi diluar *subtour*. Dapat dilihat pada **tabel 3.7**.

**Tabel 3. 7**

Iterasi 3 - Penyisipan terhadap subtour

|  |  |  |
| --- | --- | --- |
| **Arc yang akan diganti** | **Arc yang akan disisipkan ke subtour** | **Penambahan jarak (km)** |
| (d5, d4) | (d5, d2) → (d2, d4) | Cd5,d2 + Cd2,d4 - Cd5,d4  5.872 + 4.468 – 1.412 = 8.928 |
| (d5, d4) | (d5, d3) → (d3, d4) | Cd5,d3 + Cd3,d4 - Cd5,d4  5.872 + 3.397 – 1.412 = 7.857 |
| (d5, d4) | (d5, d6) → (d6, d4) | Cd5,d6 + Cd6,d4 - Cd5,d4  4.397 + 3.646 – 1.412 = 6.631 |
| (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, d1) | (d4, d2) → (d2, d1) | Cd4,d2 + Cd2,d1 - Cd4,d1  4.468 + 6.743 – 2.315 = 8.896 |
| (d4, d1) | (d4, d3) → (d3, d1) | Cd4,d3 + Cd3,d1 - Cd4,d1  3.397 + 5.189 – 2.315 = 6.271 |
| (d4, d1) | (d4, d6) → (d6, d1) | Cd4,d6 + Cd6,d1 - Cd4,d1  3.646 + 4.407 – 2.315 = 5.738 |
| (d4, d1) | (d4, d7) → (d7, d1) | Cd4,d7 + Cd7,d1 - Cd4,d1  5.164 + 7.148 – 2.315 = 9.997 |

**Tabel 3.7**

Iterasi 3 - Penyisipan terhadap subtour (Lanjutan)

|  |  |  |
| --- | --- | --- |
| **Arc yang akan diganti** | **Arc yang akan disisipkan ke subtour** | **Penambahan jarak (km)** |
| (d4, d1) | (d4, d8) → (d8, d1) | Cd4,d8 + Cd8,d1 - Cd4,d1  6.416 + 8.696 – 2.315 = 12.797 |
| (s, d5) | (s, d2) → (d2, d5) | Cs,d2 + Cd2,d5 – Cs,d5  11,87 + 5.872 – 7.773 = 9.969 |
| (s, d5) | (s, d3) → (d3, d5) | Cs,d3 + Cd3,d5 – Cs,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 |

Dari tabel 3.8, dapat diambil bahwa hasil dari penambahan jarak atau bobot paling kecil dari penyisipan ialah 5.738. Maka Arc (d4, d1) diganti dengan (d4, d6) → (d6, d1). Kondisi *subtour* saat ini: (s, d5) → (d5, d4) → (d4, d6) → (d6, d1).

1. Selanjutnya melakukan pengecekan tempat tujuan yang belum ada pada *subtour*, yang terdapat pada d2, d3, d7, d8. Kemudian dibuatlah nilai sisipan terbaru dari setiap kombinasi diluar *subtour*. Dapat dilihat pada **Tabel 3.8**.

**Tabel 3. 8**

Iterasi 4 - Penyisipan terhadap subtour

|  |  |  |
| --- | --- | --- |
| **Arc yang akan diganti** | **Arc yang akan disisipkan ke subtour** | **Penambahan jarak (km)** |
| (s, d5) | (s, d2) → (d2, d5) | Cs,d2 + Cd2,d5 – Cs,d5  11.87 + 5.872 – 7.773 = 9.969 |

**Tabel 3. 8**

Iterasi 4 - Penyisipan terhadap subtour (Lanjutan)

|  |  |  |
| --- | --- | --- |
| **Arc yang akan diganti** | **Arc yang akan disisipkan ke subtour** | **Penambahan jarak (km)** |
| (s, d5) | (s, d3) → (d3, d5) | Cs,d3 + Cd3,d5 – Cs,d5  11.97 + 4.675 – 7.773 = 8.872 |
| (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, d4) | (d5, d2) → (d2, d4) | Cd5,d2 + Cd2,d4 - Cd5,d4  5.872 + 4.468 – 1.412 = 8.928 |
| (d5, d4) | (d5, d3) → (d3, d4) | Cd5,d3 + Cd3,d4 - Cd5,d4  4.675 + 3.397 – 1.412 = 6.66 |
| (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, d6) | (d4, d2) → (d2, d6) | Cd4,d2 + Cd2,d6 - Cd4,d6  4.468 + 4.938 – 3.646 = 5.76 |
| (d4, d6) | (d4, d3) - (d3, d6) | Cd4,d3 + Cd3,d6 - Cd4,d6  3.397 + 2.246 – 3.646 = 1.997 |
| (d4, d6) | (d4, d7) → (d7, d6) | Cd4,d7 + Cd7,d6 - Cd4,d6  5.164 + 7.829 – 3.646 = 9.347 |
| (d4, d6) | (d4, d8) → (d8, d6) | Cd4,d8 + Cd8,d6 - Cd4,d6  6.416 + 7.869 – 3.646 = 10.639 |
| (d6, d1) | (d6, d2) → (d2, d1) | Cd6,d2 + Cd2,d1 - Cd6,d1  4.938 + 6.743 – 4.407 = 7.274 |

**Tabel 3.8**

Iterasi 4 - Penyisipan terhadap subtour (Lanjutan)

|  |  |  |
| --- | --- | --- |
| **Arc yang akan diganti** | **Arc yang akan disisipkan ke subtour** | **Penambahan jarak (km)** |
| (d6, d1) | (d6, d3) → (d3, d1) | Cd6,d3 + Cd3,d1 - Cd6,d1  2.246 + 5.189 – 4.407 = 3.028 |
| (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 |

Dari tabel 3.8, dapat diambil bahwa hasil dari penambahan jarak atau bobot paling kecil dari penyisipan ialah 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).

1. Selanjutnya melakukan pengecekan tempat tujuan yang belum ada pada *subtour*, yang terdapat pada d2, d7, d8. Kemudian dibuatlah nilai sisipan terbaru dari setiap kombinasi diluar *subtour*. Dapat dilihat pada **Tabel 3.9.**

**Tabel 3. 9**

Iterasi 5 - Penyisipan terhadap subtour

|  |  |  |
| --- | --- | --- |
| **Arc yang akan diganti** | **Arc yang akan disisipkan ke subtour** | **Penambahan jarak (km)** |
| (s, d5) | (s, d2) → (d2, d5) | Cs,d2 + Cd2,d5 – Cs,d5  11.87 + 5.872 – 7.773 = 9.969 |
| (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 |

. **Tabel 3.9**

Iterasi 5 - Penyisipan terhadap subtour (Lanjutan)

|  |  |  |
| --- | --- | --- |
| **Arc yang akan diganti** | **Arc yang akan disisipkan ke subtour** | **Penambahan jarak (km)** |
| (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 |

Dari tabel 3.9, dapat diambil bahwa hasil dari penambahan jarak atau bobot paling kecil dari penyisipan ialah 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).

1. Selanjutnya melakukan pengecekan tempat tujuan yang belum ada pada *subtour*, yang terdapat pada d7, d8. Kemudian dibuatlah nilai sisipan terbaru dari setiap kombinasi diluar *subtour*. Dapat dilihat pada **tabel 3.10.**

**Tabel 3. 10**

Iterasi 6 - Penyisipan terhadap subtour

|  |  |  |
| --- | --- | --- |
| **Arc yang akan diganti** | **Arc yang akan disisipkan ke subtour** | **Penambahan jarak (km)** |
| (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, 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, d2) | (d4, d7) → (d7, d2) | Cd4,d7 + Cd7,d2 - Cd4,d2  5.164 + 4.142 – 4.468 = 4.838 |
| (d4, d2) | (d4, d8) → (d8, d2) | Cd4,d8 + Cd8,d2 - Cd4,d2  6.416 + 3.026 – 4.468 = 4.974 |
| (d2, d3) | (d2, d7) → (d7, d3) | Cd2,d7 + Cd7,d3 - Cd2,d3  4.142 + 6.057 – 2.707 = 7.492 |
| (d2, d3) | (d2, d8) → (d8, d3) | Cd2,d8 + Cd8,d3 - Cd2,d3  3.026 + 5.69 – 2.707 = 6.009 |
| (d3, d6) | (d3, d7) → (d7, d6) | Cd3,d7 + Cd7,d6 - Cd3,d6  6.057 + 7.829 – 2.246 = 11.64 |

**Tabel 3.10**

Iterasi 6 - Penyisipan terhadap subtour (Lanjutan)

|  |  |  |
| --- | --- | --- |
| **Arc yang akan diganti** | **Arc yang akan disisipkan ke subtour** | **Penambahan jarak (km)** |
| (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 |

Dari tabel 3.10, dapat diambil bahwa hasil dari penambahan jarak atau bobot paling kecil dari penyisipan ialah 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).

1. Selanjutnya melakukan pengecekan tempat tujuan yang belum ada pada *subtour*, yang terdapat pada d8. Kemudian dibuatlah nilai sisipan terbaru dari setiap kombinasi diluar *subtour*. Dapat dilihat pada **tabel 3.11.**

**Tabel 3. 11**

Iterasi 7 - Penyisipan terhadap subtour

|  |  |  |
| --- | --- | --- |
| **Arc yang akan diganti** | **Arc yang akan disisipkan ke subtour** | **Penambahan jarak (km)** |
| (s, d5) | (s, d8) → (d8, d5) | Cs,d8 + Cd8,d5 – Cs,d5  11.74 + 7.647 – 7.773 = 11.614 |
| (d5, d4) | (d5, d8) → (d8, d4) | Cd5,d8 + Cd8,d4 - Cd5,d4  7.647 + 6.416 – 1.412 = 12.651 |
| (d4, d7) | (d4, d8) → (d8, d7) | Cd4,d8 + Cd8,d7 - Cd4,d7  6.416 + 2.818 – 5.164 = 4.07 |

**Tabel 3.11**

Iterasi 7 - Penyisipan terhadap subtour (Lanjutan)

|  |  |  |
| --- | --- | --- |
| **Arc yang akan diganti** | **Arc yang akan disisipkan ke subtour** | **Penambahan jarak (km)** |
| (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 |

Dari tabel 3.11, dapat diambil bahwa hasil dari penambahan jarak atau bobot paling kecil dari penyisipan ialah 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).

1. Selanjutnya dilakukan kembali pengecekan tempat tujuan yang belum masuk dalam *subtour*. Pada tabel 3.11 terlihat bahwa seluruh tempat tujuan sudah terdapat pada *subtour*, ditambah dengan rute awal (s). Maka dari itu iterasi penyisipan diberhentikan, dan program akan menampilkan *Output* berupa nilai-nilai yang terdapat pada *subtour*.

Hasil *output* dari aplikasi berupa runtutan perjalanan yang diambil dari *subtour*, yakni:

Rute perjalanan: (s, d5) → (d5, d4) → (d4, d7) → (d7, d8) → (d8, d2) → (d2, d3) → (d3, d6) → (d6, d1).

Dengan total jarak perjalanan yang dapat dihitung, dapat dilihat pada **Tabel 3.12**.

**Tabel 3. 12**

Total jarak perjalanan menggunakan algoritma Cheapest Insertion Heuristic

|  |  |
| --- | --- |
| **Arc** | **Jarak** |
| (s, d5) | 7.773 |
| (d5, d4) | 1.412 |
| (d4, d7) | 5.164 |
| (d7, d8) | 2.818 |
| (d8, d2) | 3.026 |
| (d2, d3) | 2.707 |
| (d3, d6) | 2.246 |
| (d6, d1) | 4.407 |
| **Total Jarak (km)** | **29.553** |