

**APLIKASI TRAVELLING SALESMAN PROBLEM PADA
PENGEDROPAN BARANG DI ANJUNGAN MENGGUNAKAN
METODE INSERTION**

Priska Sari Dewi

Jurusan Matematika, FMIPA Universitas Jenderal Soedirman
priska.saridewi@gmail.com

Triyani

Jurusan Matematika, FMIPA Universitas Jenderal Soedirman

Siti Rahmah Nurshiami

Jurusan Matematika, FMIPA Universitas Jenderal Soedirman

ABSTRACT. *Travelling Salesman Problem (TSP) is a problem to find the shortest path a salesman visits all the cities exactly once, and returns to the starting city. In this reseach, the methods for TSP used are the nearest insertion method, the cheapest insertion method, and the farthest insertion method. The program for deciding the minimum TSP from three insertion methods was created with the help of the function of Software R. The TSP using three insertion methods do not always have the same weight and route.*

Key words: *Travelling Salesman Problem, Nearest Insertion Method, Cheapest Insertion Method, Farthest Insertion Method, and Software R.*

ABSTRAK. *Travelling Salesman Problem (TSP) merupakan permasalahan mencari lintasan terpendek seorang salesman harus mengunjungi semua kota yang akan dituju tepat satu kali, dan kembali ke kota awal. Pada penelitian ini, penyelesaian TSP menggunakan metode *nearest insertion*, metode *cheapest insertion*, dan metode *farthest insertion*. Program minimum TSP dari ketiga metode dibuat dengan bantuan fungsi *Software R*. Hasil TSP dengan menggunakan tiga metode *insertion* tidak selalu menghasilkan bobot dan rute lintasan yang sama namun bergantung pada data yang digunakan.*

Kata kunci: *Travelling Salesman Problem, Metode Nearest Insertion, Metode Cheapest Insertion, Metode Farthest Insertion, dan Software R.*

1. PENDAHULUAN

Anjungan lepas pantai mempunyai peran dan fungsi dalam kegiatan eksplorasi bahan tambang. Tercukupinya kebutuhan barang pada anjungan lepas pantai menjadi salah satu pendukung utama jalannya kegiatan eksplorasi. Pengedropan barang pada anjungan lepas pantai melalui jalur laut dari sumber ke

titik tujuan menggunakan rute terpendek. Masalah rute pengedropan barang dari titik awal menuju tempat tujuan tepat sekali dan kembali ke titik awal sering disebut dengan *Travelling Salesman Problem* (Rosenkrantz dkk, 1977).

Travelling Salesman Problem (TSP) dikemukakan pada tahun 1800-an oleh dua orang matematikawan, Irlandia William Rowan Hamilton dan Thomas Penyngton Kirkman. Masalah ini dapat direpresentasikan sebagai siklus terpendek dari sebuah graf lengkap dengan n titik pada masalah pencarian rute optimum suatu perjalanan pengiriman barang dari suatu sumber ke tujuan. Permasalahan ini menjadi sangat kompleks ketika 100 atau lebih kota yang harus dikunjungi.

Banyak peneliti telah mengkaji penyelesaian *Travelling Salesman Problem*, namun sampai saat ini penyelesaian *Travelling Salesman Problem* masih belum menghasilkan solusi yang optimal. Salah satunya yaitu Hahsler & Hornik (2007) yang telah mengkaji metode *insertion* untuk tiga metode yaitu metode *nearest insertion*, metode *cheapest insertion* dan metode *farthest insertion* untuk menyelesaikan *Travelling Salesman Problem* dengan menggunakan matriks jarak namun tidak dijelaskan secara terperinci. Untuk itu dalam penelitian ini, akan dikaji penyelesaian *Travelling Salesman Problem* dengan menggunakan metode *insertion* secara rinci untuk ketiga metode yaitu metode *nearest insertion*, metode *cheapest insertion* dan metode *farthest insertion*. Ketiga metode ini digunakan untuk menyelesaikan *Travelling Salesman Problem* dengan mengambil n data jarak koordinat antar titik. Selanjutnya menyusun program minimum dari ketiga metode *insertion* menggunakan bantuan fungsi *Software R*.

2. METODE PENELITIAN

Penelitian ini dilakukan dengan perangkat keras laptop individu dengan Prosesor Intel® Core™ i5-8250U CPU @1.60GHz, perangkat lunak dengan Sistem Operasi Windows 10 Home 64-bit, *Studio R Version* 1.3.1093, dan Microsoft Excel 2019 Profesional Plus. Data yang digunakan adalah data jarak koordinat antar titik dalam bentuk file penyimpanan *File Comma Separated Values* (CSV). Adapun langkah-langkah yang dilakukan dalam penelitian ini adalah mengkaji dan menyelesaikan TSP menggunakan metode *nearest insertion*,

metode *cheapest insertion*, dan metode *farthest insertion* menggunakan data jarak koordinat antar titik dan membuat program *software R* untuk meminimumkan TSP dari metode *nearest insertion*, metode *cheapest insertion*, dan metode *farthest insertion*.

3. HASIL DAN PEMBAHASAN

Persoalan *Travelling Salesman Problem* dapat direpresentasikan untuk menentukan sirkuit Hamilton yang memiliki bobot minimum pada graf lengkap (K_n) berbobot dengan n titik (Munir, 2010). Titik pada graf lengkap dapat berupa koordinat kartesius, sehingga jarak (d) antara dua buah titik (x_1, y_1) dan (x_2, y_2) harus dihitung secara *Euclidean* dengan persamaan

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \quad (1)$$

Secara nyata, sampai saat ini penyelesaian *Travelling Salesman Problem* masih belum menghasilkan solusi yang optimal. Salah satu algoritma yang digunakan dalam penyelesaian *Travelling Salesman Problem* adalah algoritma *heuristic* (Rosenkrantz, Stearns, & Lewis, 1977). Metode *heuristic* memberikan pendekatan dalam menyelesaikan permasalahan optimasi kombinatorial di mana diperoleh hasil yang mungkin dan mendekati optimal (Hansler & Kornik, 2007).

Menurut Daniel dkk (1977), konsep metode *insertion* dalam menyelesaikan TSP adalah membangun *tour* dari himpunan kota atau *subtour* dengan cara memasukkan sebuah kota ke kota asal sehingga menjadi *subtour*, dan berulang sampai semua kota terhubung. Diberikan graf TSP (N, d) dimana N merepresentasikan jumlah kota dan d merepresentasikan jarak antar kota. Sebuah *tour* T pada *subset* dari N disebut *subtour* dari (N, d) dimana $a \in T$ berarti $a \in N$. Sebuah *subtour* T dan sebuah kota k di N tetapi tidak ada di T , didefinisikan $TOUR(T, k)$, dan akan menjadi sebuah *subtour* dengan ketentuan berikut :

a. Jika *subtour* T melewati lebih dari 1 kota, maka

1) mencari sisi (i, j) di T yang meminimalkan

$$F = d(i, k) + d(k, j) - d(i, j) \quad (2)$$

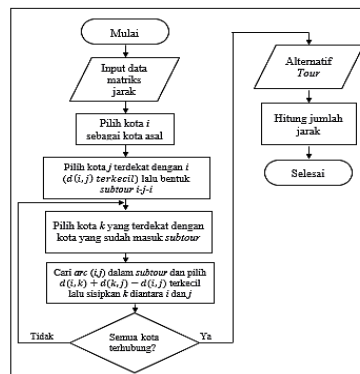
F : Jarak tambahan i : Kota pertama dalam *insertion*

k : Kota yang belum terpilih j : Kota kedua dalam *insertion*

- 2) menghapus sisi (i, j) dan menambahkan sisi (i, k) dan (k, j) untuk memperoleh $TOUR(T, k)$
- b. Jika *subtour* T melewati satu kota i saja maka, untuk membuat $TOUR(T, k)$ dengan 2 kota, digunakan 2 buah *arc* (himpunan dari pasangan terurut i, j yang mempunyai arah dari titik i ke j) yaitu $arc(i, j)$ dan $arc(j, i)$ yang didefinisikan sebagai $(i, j) \rightarrow (j, i)$, yang berarti terdapat lintasan dari kota i ke j begitupun sebaliknya.

3.1 Penyelesaian TSP dengan Metode Nearest Insertion

Metode *nearest insertion* yaitu metode menyisipkan kota terdekat dari suatu kota yang sudah termuat pada *subtour* yang sudah ada untuk masuk ke dalamnya sampai seluruh kota terhubung (Yuan Ze University, 2015). Berikut diagram alir langkah metode *nearest insertion*.



Gambar 1. Diagram Alir Metode Nearest Insertion

Berikut diberikan contoh penyelesaian TSP dengan menggunakan Metode *Nearest Insertion*. Misal diberikan data koordinat 14 anjungan seperti pada Tabel 1.

Tabel 1. Data Koordinat 14 Anjungan

Kota	x	y	Kota	x	y
A	14	8	H	4	17
B	9	4	I	12	8
C	6	18	J	19	16
D	17	13	K	7	9
E	7	5	L	4	7
F	3	10	M	10	16
G	11	9	N	8	4

Jarak antar anjungan dalam bentuk koordinat diubah ke dalam jarak *Euclidean* menggunakan Persamaan (1) dan dapat dilihat pada Tabel 2 berikut.

Tabel 2. Jarak *Euclidean* 14 Anjungan

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A	0	6,403124	12,806248	5,830952	7,615773	11,18034	3,162278	13,453624	2	9,433981	7,071068	10,049876	8,944272	7,211103
B	6,403124	0	14,317821	12,041595	2,236068	8,485281	5,385165	13,928388	5	15,620499	5,385165	5,830952	12,041595	1
C	12,806248	14,317821	0	12,083046	13,038405	8,544004	10,29563	2,236068	11,661904	13,152946	9,055385	11,18034	4,472136	14,142136
D	5,830952	12,041595	12,083046	0	12,806248	14,317821	7,211103	13,601471	7,071068	3,605551	10,77033	14,317821	7,615773	12,727922
E	7,615773	2,236068	13,038405	12,806248	0	6,403124	5,656854	12,369317	5,830952	16,278821	4	3,605551	11,401754	1,414214
F	11,18034	8,485281	8,544004	14,317821	6,403124	0	8,062258	7,071068	9,219544	17,088007	4,123106	3,162278	9,219544	7,81025
G	3,162278	5,385165	10,29563	7,211103	5,656854	8,062258	0	10,630146	1,414214	10,630146	4	7,28011	7,071068	5,830952
H	13,453624	13,928388	2,236068	13,601471	12,369317	7,071068	10,630146	0	12,041595	15,033296	8,544004	10	6,082763	13,601471
I	2	5	11,661904	7,071068	5,830952	9,219544	1,414214	12,041595	0	10,630146	5,09902	8,062258	8,246211	5,656854
J	9,433981	15,620499	13,152946	3,605551	16,278821	17,088007	10,630146	15,033296	10,630146	0	13,892444	17,492856	9	16,278821
K	7,071068	5,385165	9,055385	10,77033	4	4,123106	4	8,544004	5,09902	13,892444	0	3,605551	7,615773	5,09902
L	10,049876	5,830952	11,18034	14,317821	3,605551	3,162278	7,28011	10	8,062258	17,492856	3,605551	0	10,816654	5
M	8,944272	12,041595	4,472136	7,615773	11,401754	9,219544	7,071068	6,082763	8,246211	9	7,615773	10,816654	0	12,165525
N	7,211103	1	14,142136	12,727922	1,414214	7,81025	5,830952	13,601471	5,656854	16,278821	5,09902	5	12,165525	0

Selanjutnya penyelesaian TSP dilakukan dengan langkah-langkah berikut.

Langkah 1 Pada pembahasan kali ini dipilih anjungan C sebagai anjungan asal.

Langkah 2 Anjungan H terpilih sebagai anjungan terdekat dengan anjungan C karena memiliki jarak yang paling minimum yaitu 2,236068. *Subtour* sementara $C - H - C$ atau dapat ditulis $(C, H) \rightarrow (H, C)$.

Langkah 3 Berikut ditunjukkan jarak antar anjungan dengan anjungan C dan H.

Tabel 3. Anjungan Terdekat Penambah *Subtour* Pertama Metode *Nearest Insertion*

Jarak dari	Ke	Jarak	Jarak dari	Ke	Jarak
C	A	12,8062	H	A	13,4536
C	B	14,3178	H	B	13,9284
C	D	12,083	H	D	13,6015
C	E	13,0384	H	E	12,3693
C	F	8,544	H	F	7,07107
C	G	10,2956	H	G	10,6301
C	I	11,6619	H	I	12,0416
C	J	13,1529	H	J	15,0333
C	K	9,05539	H	K	8,544
C	L	11,1803	H	L	10
C	M	4,47214	H	M	6,08276
C	N	14,1421	H	N	13,6015

Melihat Tabel 3, terpilih anjungan M sebagai anjungan terdekat dengan anjungan C karena memiliki jarak paling minimum yaitu 4,472136, lalu anjungan M disisipkan dalam *subtour* dan dihitung menggunakan Persamaan (2) yang menghasilkan jarak tambahan terkecil.

Tabel 4. Arc Penambah *Subtour* Pertama pada Metode *Nearest Insertion*

Arc awal	Arc yang akan ditambahkan ke <i>subtour</i>	Jarak tambahan
(C, H)	(C, M) → (M, H)	$d_{CM} + d_{MH} - d_{CH} = 8,318831$
(H, C)	(H, M) → (M, C)	$d_{HM} + d_{MC} - d_{HC} = 8,318831$

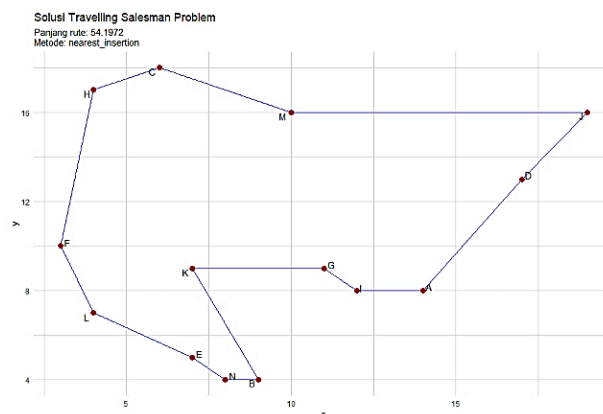
Melihat Tabel 4, dipilih *arc*(C, M) dengan *arc*(M, H) menggantikan *arc*(C, H) dengan jarak tambahan sebesar 8,318831. *Subtour* sementara yaitu (C, M) → (M, H) → (H, C). Belum semua anjungan masuk *subtour*, maka kembali seperti Langkah 3. Adapun hasil langkah berikutnya ada pada Tabel 5 berikut.

Tabel 5. Penambah *Subtour* pada Metode *Nearest Insertion*

Langkah	Tambahan anjungan	Terdekat dari	Jarak tambahan	<i>Subtour</i> sementara
4	G	M	7,071068	(C, M) → (M, G) → (G, H) → (H, C)
5	I	G	2,589357	(C, M) → (M, I) → (I, G) → (G, H) → (H, C)
6	A	I	2,698061	(C, M) → (M, I) → (I, G) → (G, H) → (H, C)
7	K	G	1,913858	(C, M) → (M, A) → (A, I) → (I, G) → (G, K) → (K, H) → (H, C)
8	L	K	5,061547	(C, M) → (M, A) → (A, I) → (I, G) → (G, K) → (K, L) → (L, H) → (H, C)
9	E	L	4	(C, M) → (M, A) → (A, I) → (I, G) → (G, K) → (K, E) → (E, L) → (L, H) → (H, C)
10	N	E	2,513234	(C, M) → (M, A) → (A, I) → (I, G) → (G, K) → (K, N) → (N, E) → (E, L) → (L, H) → (H, C)
11	B	N	1,286145	(C, M) → (M, A) → (A, I) → (I, G) → (G, K) → (K, B) → (B, N) → (N, E) → (E, L) → (L, H) → (H, C)
12	F	L	0,683346	(C, M) → (M, A) → (A, I) → (I, G) → (G, K) → (K, B) → (B, N) → (N, E) → (E, L) →

Langkah	Tambahan anjungan	Terdekat dari	Jarak tambahan	Subtour sementara
13	D	A	4,502453	$(L, F) \rightarrow (F, H) \rightarrow (H, C)$ $(C, M) \rightarrow (M, D) \rightarrow (D, A) \rightarrow$ $(A, I) \rightarrow (I, G) \rightarrow (G, K) \rightarrow$ $(K, B) \rightarrow (B, N) \rightarrow (N, E) \rightarrow$ $(E, L) \rightarrow (L, F) \rightarrow (F, H) \rightarrow (H, C)$
14	J	-	4,989778	$(C, M) \rightarrow (M, J) \rightarrow (J, D) \rightarrow$ $(D, A) \rightarrow (A, I) \rightarrow (I, G) \rightarrow$ $(G, K) \rightarrow (K, B) \rightarrow (B, N) \rightarrow$ $(N, E) \rightarrow (E, L) \rightarrow (L, F) \rightarrow$ $(F, H) \rightarrow (H, C)$

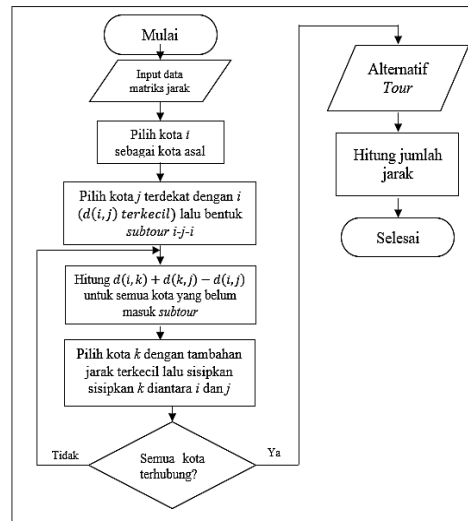
Melihat Tabel 5, penyelesaian TSP 14 anjungan menggunakan metode *nearest insertion* adalah $(C, M) \rightarrow (M, J) \rightarrow (J, D) \rightarrow (D, A) \rightarrow (A, I) \rightarrow (I, G) \rightarrow (G, K) \rightarrow (K, B) \rightarrow (B, N) \rightarrow (N, E) \rightarrow (E, L) \rightarrow (L, F) \rightarrow (F, H) \rightarrow (H, C)$ dengan jarak 54,1972.



Gambar 2. TSP 14 Anjungan dengan Menggunakan Metode *Nearest Insertion*

3.2 Penyelesaian TSP dengan Metode *Cheapest Insertion*

Perbedaan dengan metode sebelumnya yaitu pemilihan kota yang akan disisipkan, metode *nearest insertion* berdasarkan jarak terdekat dengan kota yang sudah masuk *subtour*, sedangkan metode *cheapest insertion* berdasarkan jarak tambahan terkecil. Berikut diagram alir langkah metode *cheapest insertion*:



Gambar 3. Diagram Alir Metode *Cheapest Insertion*

Berikut contoh penyelesaian TSP dengan metode *Cheapest Insertion* menggunakan data jarak Tabel 2.

Langkah 1 Pada pembahasan kali ini dipilih anjungan C sebagai anjungan asal.

Langkah 2 Anjungan H terpilih sebagai anjungan terdekat dengan anjungan C karena memiliki jarak paling minimum yaitu 2,236068. *Subtour* sementara $C - H - C$ atau dapat ditulis $(C, H) \rightarrow (H, C)$.

Langkah 3 Memasukan anjungan $k \in \{A, B, D, E, F, I, J, K, L, M, N\}$ yang belum masuk dalam *subtour*. Adapun hasil langkah berikutnya ada pada Tabel 6 berikut.

Tabel 6. Penambah Subtour pertama pada Metode *Cheapest Insertion*

Arc awal	Arc yang akan ditambahkan ke <i>subtour</i>	Jarak tambahan
(C, H)	$(C, A) \rightarrow (A, H)$	$d_{CA} + d_{AH} - d_{CH} = 24,023804$
(C, H)	$(C, B) \rightarrow (B, H)$	$d_{CB} + d_{BH} - d_{CH} = 26,010141$
(C, H)	$(C, D) \rightarrow (D, H)$	$d_{CD} + d_{DH} - d_{CH} = 23,448449$
(C, H)	$(C, E) \rightarrow (E, H)$	$d_{CE} + d_{EH} - d_{CH} = 23,171654$
(C, H)	$(C, F) \rightarrow (F, H)$	$d_{CF} + d_{FH} - d_{CH} = 13,379004$
(C, H)	$(C, G) \rightarrow (G, H)$	$d_{CG} + d_{GH} - d_{CH} = 18,689708$
(C, H)	$(C, I) \rightarrow (I, H)$	$d_{CI} + d_{IH} - d_{CH} = 21,467431$
(C, H)	$(C, J) \rightarrow (J, H)$	$d_{CJ} + d_{JH} - d_{CH} = 25,950174$
(C, H)	$(C, K) \rightarrow (K, H)$	$d_{CK} + d_{KH} - d_{CH} = 15,363321$
(C, H)	$(C, L) \rightarrow (L, H)$	$d_{CL} + d_{LH} - d_{CH} = 18,944272$
(C, H)	$(C, M) \rightarrow (M, H)$	$d_{CM} + d_{MH} - d_{CH} = 8,318831$
(C, H)	$(C, N) \rightarrow (N, H)$	$d_{CN} + d_{NH} - d_{CH} = 25,507539$

Arc awal	Arc yang akan ditambahkan ke <i>subtour</i>	Jarak tambahan
(H, C)	$(H, A) \rightarrow (A, C)$	$d_{HA} + d_{AC} - d_{HC} = 24,023804$
(H, C)	$(H, B) \rightarrow (B, C)$	$d_{HB} + d_{BC} - d_{HC} = 26,010141$
(H, C)	$(H, D) \rightarrow (D, C)$	$d_{HD} + d_{DC} - d_{HC} = 23,448449$
(H, C)	$(H, E) \rightarrow (E, C)$	$d_{HE} + d_{EC} - d_{HC} = 23,171654$
(H, C)	$(H, F) \rightarrow (F, C)$	$d_{HF} + d_{FC} - d_{HC} = 13,379004$
(H, C)	$(H, G) \rightarrow (G, C)$	$d_{HG} + d_{GC} - d_{HC} = 18,689708$
(H, C)	$(H, I) \rightarrow (I, C)$	$d_{HI} + d_{IC} - d_{HC} = 21,467431$
(H, C)	$(H, J) \rightarrow (J, C)$	$d_{HJ} + d_{JC} - d_{HC} = 25,950174$
(H, C)	$(H, K) \rightarrow (K, C)$	$d_{HK} + d_{KC} - d_{HC} = 15,363321$
(H, C)	$(H, L) \rightarrow (L, C)$	$d_{HL} + d_{LC} - d_{HC} = 18,944272$
(H, C)	$(H, M) \rightarrow (M, C)$	$d_{HM} + d_{MC} - d_{HC} = 8,318831$
(H, C)	$(H, N) \rightarrow (N, C)$	$d_{HN} + d_{NC} - d_{HC} = 25,507539$

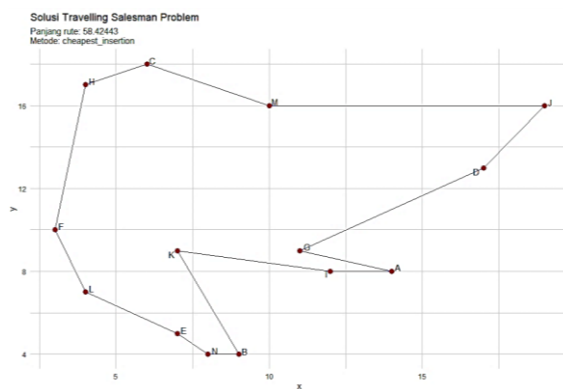
Melihat Tabel 6, dipilih $arc(H, M)$ dengan $arc(M, C)$ menggantikan $arc(H, C)$ dengan jarak tambahan sebesar 8,318831. *Subtour* sementara yaitu $(C, H) \rightarrow (H, M) \rightarrow (M, C)$. Belum semua anjungan masuk *subtour*, maka kembali seperti Langkah 3. Adapun hasil langkah berikutnya ada pada Tabel 7 berikut.

Tabel 7. Penambah *Subtour* pada Metode *Cheapest Insertion*

Langkah	Tambahan anjungan	Jarak tambahan	<i>Subtour</i> sementara
4	K	10,077014	$(C, H) \rightarrow (H, K) \rightarrow (K, M) \rightarrow (M, C)$
5	F	2,65017	$(C, H) \rightarrow (H, F) \rightarrow (F, K) \rightarrow (K, M) \rightarrow (M, C)$
6	L	2,644723	$(C, H) \rightarrow (H, F) \rightarrow (F, L) \rightarrow (L, K) \rightarrow (K, M) \rightarrow (M, C)$
7	G	3,455295	$(C, H) \rightarrow (H, F) \rightarrow (F, L) \rightarrow (L, K) \rightarrow (K, G) \rightarrow (G, M) \rightarrow (M, C)$
8	I	2,513234	$(C, H) \rightarrow (H, F) \rightarrow (F, L) \rightarrow (L, K) \rightarrow (K, I) \rightarrow (I, G) \rightarrow (G, M) \rightarrow (M, C)$
9	A	3,748064	$(C, H) \rightarrow (H, F) \rightarrow (F, L) \rightarrow (L, K) \rightarrow (K, I) \rightarrow (I, A) \rightarrow (A, G) \rightarrow (G, M) \rightarrow (M, C)$
10	E	4	$(C, H) \rightarrow (H, F) \rightarrow (F, L) \rightarrow (L, E) \rightarrow (E, K) \rightarrow (K, I) \rightarrow (I, A) \rightarrow (A, G) \rightarrow (G, M) \rightarrow (M, C)$
11	N	2,513234	$(C, H) \rightarrow (H, F) \rightarrow (F, L) \rightarrow (L, E) \rightarrow (E, N) \rightarrow (N, K) \rightarrow (K, I) \rightarrow (I, A) \rightarrow (A, G) \rightarrow (G, M) \rightarrow (M, C)$
12	B	1,286145	$(C, H) \rightarrow (H, F) \rightarrow (F, L) \rightarrow (L, E) \rightarrow (E, N) \rightarrow (N, B) \rightarrow (B, K) \rightarrow (K, I) \rightarrow (I, A) \rightarrow (A, G) \rightarrow (G, M) \rightarrow (M, C)$
13	D	7,755808	$(C, H) \rightarrow (H, F) \rightarrow (F, L) \rightarrow (L, E) \rightarrow (E, N) \rightarrow (N, B) \rightarrow (B, K) \rightarrow (K, I) \rightarrow (I, A) \rightarrow (A, G) \rightarrow (G, D) \rightarrow (D, M) \rightarrow (M, C)$

Langkah	Tambahan anjungan	Jarak tambahan	Subtour sementara
14	J	4,989778	$(C, H) \rightarrow (H, F) \rightarrow (F, L) \rightarrow (L, E) \rightarrow (E, N) \rightarrow (N, B) \rightarrow (B, K) \rightarrow (K, I) \rightarrow (I, A) \rightarrow (A, G) \rightarrow (G, D) \rightarrow (D, J) \rightarrow (J, M) \rightarrow (M, C)$

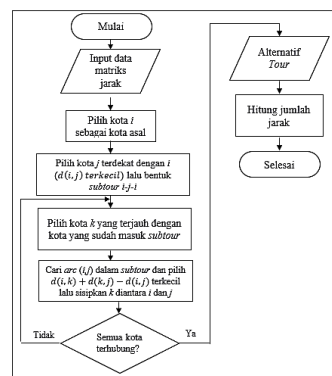
Melihat Tabel 7, penyelesaian TSP 14 anjungan menggunakan metode *cheapest insertion* adalah $(C, H) \rightarrow (H, F) \rightarrow (F, L) \rightarrow (L, E) \rightarrow (E, N) \rightarrow (N, B) \rightarrow (B, K) \rightarrow (K, I) \rightarrow (I, A) \rightarrow (A, G) \rightarrow (G, D) \rightarrow (D, J) \rightarrow (J, M) \rightarrow (M, C)$ dengan jarak 58,42443.



Gambar 4. TSP 14 Anjungan dengan Menggunakan Metode *Cheapest Insertion*

3.3 Penyelesaian TSP dengan Metode *Farthest Insertion*

Metode *farthest insertion* sama halnya metode *nearest insertion* namun berbeda pengambilan kota yaitu menggunakan kota terjauh bukan terdekat. Berikut diagram alir langkah metode *farthest insertion*:



Gambar 5. TSP 14 Anjungan dengan Menggunakan Metode *Nearest Insertion*

Berikut contoh penyelesaian TSP dengan metode *Farthest Insertion* menggunakan data jarak Tabel 2.

Langkah 1 Pada pembahasan kali ini dipilih anjungan C sebagai anjungan asal.

Langkah 2 Anjungan B terpilih sebagai anjungan terjauh dengan anjungan C karena memiliki jarak yang paling maksimum yaitu 14,317821. *Subtour* sementara $C - B - C$ atau dapat ditulis $(C, B) \rightarrow (B, C)$.

Langkah 3 Berikut ditunjukkan jarak antar anjungan lain dengan anjungan B dan C.

Tabel 8. Anjungan Terdekat Penambah *Subtour* Pertama Metode *Farthest Insertion*

Jarak dari	Ke	Jarak	Jarak dari	Ke	Jarak
B	A	6,403124	C	A	12,806248
B	D	12,041595	C	D	12,083046
B	E	2,236068	C	E	13,038405
B	F	8,485281	C	F	8,544004
B	G	5,385165	C	G	10,29563
B	H	13,928388	C	H	2,236068
B	I	5	C	I	11,661904
B	J	15,620499	C	J	13,152946
B	K	5,385165	C	K	9,055385
B	L	5,830952	C	L	11,18034
B	M	12,041595	C	M	4,472136
B	N	1	C	N	14,142136

Melihat Tabel 8, terpilih anjungan J sebagai anjungan terjauh dengan anjungan B karena memiliki jarak paling maksimum yaitu 15,620499, lalu anjungan J disisipkan dalam *subtour* dan dihitung menggunakan Persamaan (2) yang menghasilkan jarak tambahan terkecil.

Tabel 9. Arc Penambah *Subtour* Pertama pada Metode *Farthest Insertion*

Arc awal	Arc yang akan ditambahkan ke <i>subtour</i>	Jarak tambahan
(C, B)	$(C, J) \rightarrow (J, B)$	$d_{CJ} + d_{JB} - d_{CB} = 14,455624$
(B, C)	$(B, J) \rightarrow (J, C)$	$d_{BJ} + d_{JC} - d_{BC} = 14,455624$

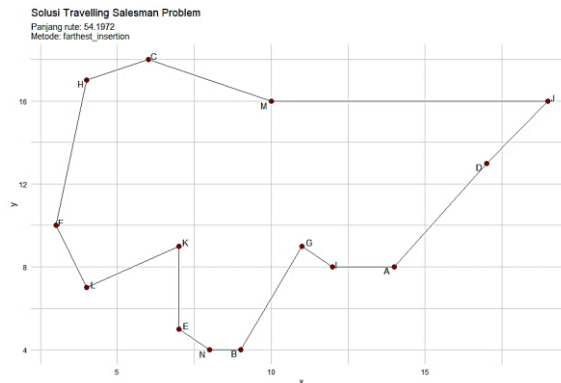
Melihat Tabel 9, dipilih $arc(C, J)$ dengan $arc(J, B)$ menggantikan $arc(C, B)$ dengan jarak tambahan sebesar 14,455624. *Subtour* sementara yaitu $(C, J) \rightarrow (J, B) \rightarrow (B, C)$. Belum semua anjungan masuk *subtour*, maka kembali seperti Langkah 3. Adapun hasil langkah berikutnya ada pada Tabel 10 berikut.

Tabel 10. Penambah *Subtour* pada Metode *Farthest Insertion*

Langkah	Tambahan anjungan	Terjauh dari	Jarak tambahan	<i>Subtour</i> sementara
4	L	J	17,492856	$(C, J) \rightarrow (J, B) \rightarrow (B, L) \rightarrow (L, C)$
5	F	J	0,525942	$(C, J) \rightarrow (J, B) \rightarrow (B, L) \rightarrow (L, F) \rightarrow (F, C)$
6	E	J	0,010667	$(C, J) \rightarrow (J, B) \rightarrow (B, E) \rightarrow (E, L) \rightarrow (L, F) \rightarrow (F, C)$
7	N	J	1,913858	$(C, J) \rightarrow (J, B) \rightarrow (B, N) \rightarrow (N, E) \rightarrow (E, L) \rightarrow (L, F) \rightarrow (F, C)$
8	H	J	0,763132	$(C, J) \rightarrow (J, B) \rightarrow (B, N) \rightarrow (N, E) \rightarrow (E, L) \rightarrow (L, F) \rightarrow (F, H) \rightarrow (H, C)$
9	D	L	0,026647	$(C, J) \rightarrow (J, D) \rightarrow (D, B) \rightarrow (B, N) \rightarrow (N, E) \rightarrow (E, L) \rightarrow (L, F) \rightarrow (F, H) \rightarrow (H, C)$
10	K	J	4	$(C, J) \rightarrow (J, D) \rightarrow (D, B) \rightarrow (B, N) \rightarrow (N, E) \rightarrow (E, K) \rightarrow (K, L) \rightarrow (L, F) \rightarrow (F, H) \rightarrow (H, C)$
11	A	H	0,192481	$(C, J) \rightarrow (J, D) \rightarrow (D, A) \rightarrow (A, B) \rightarrow (B, N) \rightarrow (N, E) \rightarrow (E, K) \rightarrow (K, L) \rightarrow (L, F) \rightarrow (F, H) \rightarrow (H, C)$
12	M	N	0,31919	$(C, M) \rightarrow (M, J) \rightarrow (J, D) \rightarrow (D, A) \rightarrow (A, B) \rightarrow (B, N) \rightarrow (N, E) \rightarrow (E, K) \rightarrow (K, L) \rightarrow (L, F) \rightarrow (F, H) \rightarrow (H, C)$
13	I	H	0,596876	$(C, M) \rightarrow (M, J) \rightarrow (J, D) \rightarrow (D, A) \rightarrow (A, I) \rightarrow (I, B) \rightarrow (B, N) \rightarrow (N, E) \rightarrow (E, K) \rightarrow (K, L) \rightarrow (L, F) \rightarrow (F, H) \rightarrow (H, C)$
14	G	-	1,799379	$(C, M) \rightarrow (M, J) \rightarrow (J, D) \rightarrow (D, A) \rightarrow (A, I) \rightarrow (I, G) \rightarrow (G, B) \rightarrow (B, N) \rightarrow (N, E) \rightarrow (E, K) \rightarrow (K, L) \rightarrow (L, F) \rightarrow (F, H) \rightarrow (H, C)$

Melihat Tabel 10, penyelesaian TSP 14 anjungan menggunakan metode *farthest insertion* adalah $(C, M) \rightarrow (M, J) \rightarrow (J, D) \rightarrow (D, A) \rightarrow (A, I) \rightarrow (I, G) \rightarrow (G, B) \rightarrow$

$(B, N) \rightarrow (N, E) \rightarrow (E, K) \rightarrow (K, L) \rightarrow (L, F) \rightarrow (F, H) \rightarrow (H, C)$ dengan jarak 54,1972.



Gambar 6. TSP 14 Anjungan dengan Menggunakan Metode *Farthest Insertion*

3.4 Program R untuk Minimum TSP dari Tiga Metode *Insertion*

Berikut langkah-langkah pemrograman R untuk minimum TSP dari tiga metode *insertion* :

Langkah 1 Langkah awal pada pemrograman R yaitu dengan mengekspor data excel ke *data frame* di R. Data awalnya disimpan dalam bentuk file penyimpanan CSV pada *Microsoft Excel* lalu diekspor menjadi *data frame* pada R dengan contoh nama 'mydata'.

```

Console Terminal x Jobs x
G:/D/ ↗
> mydata<-read.table(file.choose(),sep=";",header=TRUE,row.names=1)
> View(mydata)

```

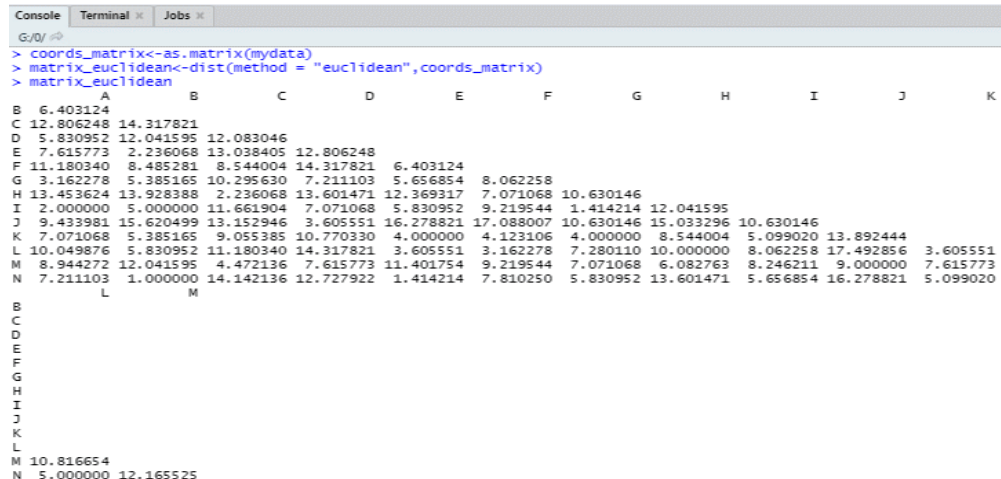
Gambar 7. Ekspor Data Excel ke *Data Frame* di R

	X	Y
A	14	8
B	9	4
C	6	18
D	17	13
E	7	5
F	3	10
G	11	9
H	4	17
I	12	8
J	19	16
K	7	9
L	4	7
M	10	16
N	8	4

Showing 1 to 14 of 14 entries, 2 total columns

Gambar 8. Ekspor Data Excel ke *Data Frame* di R

Langkah 2 Mengkonversi *Data frame* ke dalam bentuk matriks jarak menggunakan jarak *Euclidean*. Hasil matriks *Euclidean* 14 anjungan ada pada **Gambar 9**.



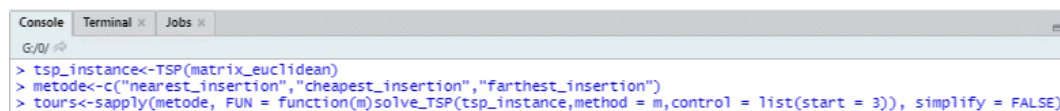
```

Console Terminal Jobs
G:/0/ #
> coords_matrix<-as.matrix(mydata)
> matrix_euclidean<-dist(method = "euclidean",coords_matrix)
> matrix_euclidean
  B      A      C      D      E      F      G      H      I      J      K
B 6.403124
C 12.806248 14.317821
D 5.830952 12.041595 12.083046
E 7.615773 2.236068 13.038405 12.806248
F 11.180340 8.485281 8.544004 14.317821 6.403124
G 3.162278 5.385165 10.295630 7.211103 5.656854 8.062258
H 13.453624 13.928388 2.236068 13.601471 12.369317 7.071068 10.630146
I 2.000000 5.000000 11.661904 7.071068 5.830952 9.219544 1.414214 12.041595
J 9.433981 15.620499 13.152946 3.605551 16.278821 17.088007 10.630146 15.033296 10.630146
K 7.071068 5.385165 9.055385 10.770330 4.000000 4.123106 4.000000 8.544004 5.099020 13.892444
L 10.049876 5.830952 11.180340 14.317821 3.605551 3.162278 7.280110 10.000000 8.062258 17.492856 3.605551
M 8.944272 12.041595 4.472136 7.615773 11.401754 9.219544 7.071068 6.082763 8.246211 9.000000 7.615773
N 7.211103 1.000000 14.142136 12.727922 1.414214 7.810250 5.830952 13.601471 5.656854 16.278821 5.099020
  L      M
B
C
D
E
F
G
H
I
J
K
L
M 10.816654
N 5.000000 12.165525

```

Gambar 9. Matriks Jarak *Euclidean*

Langkah 3 Matriks *Euclidean* dibuat menjadi objek TSP diikuti pendeskripsian metode yang akan digunakan yaitu metode *cheapest insertion*, metode *nearest insertion* dan metode *farthest insertion*. Pada program terdapat pilihan untuk menentukan titik awal bergerak dari anjungan mana sesuai kebutuhan.



```

Console Terminal Jobs
G:/0/ #
> tsp_instance<-TSP(matrix_euclidean)
> metode<-c("nearest_insertion","cheapest_insertion","farthest_insertion")
> tours<-sapply(metode, FUN = function(m)solve_TSP(tsp_instance,method = m,control = list(start = 3)), simplify = FALSE)

```

Gambar 10. Pembuatan Objek dan Metode TSP

Langkah 4 Mencari ‘panjang_tur’ untuk masing-masing metode lalu dicari mana metode yang mempunyai panjang tur/bobot paling minimal (‘min_tur’) dari ketiga metode. Hasil *tour* dapat berupa *integer* apun *data frame*, namun untuk dibuat plot *tour* maka menggunakan hasil *tour* dalam bentuk *data frame*.

```

Console Terminal Jobs
G:/0/ ↵
> panjang_tur<-sapply(tours,tour_length)
> min_tur<-panjang_tur[which.min(panjang_tur)]
> tur_min<-tours[[names(min_tur)]]
> as.integer(tur_min)
[1] 3 13 10 4 1 9 7 2 14 5 11 12 6 8
> rute_tur_min<-mydata[tur_min,]
> rute_tur_min
  X  Y
C  6 18
M 10 16
J 19 16
D 17 13
A 14  8
I 12  8
G 11  9
B  9  4
N  8  4
E  7  5
K  7  9
L  4  7
F  3 10
H  4 17

```

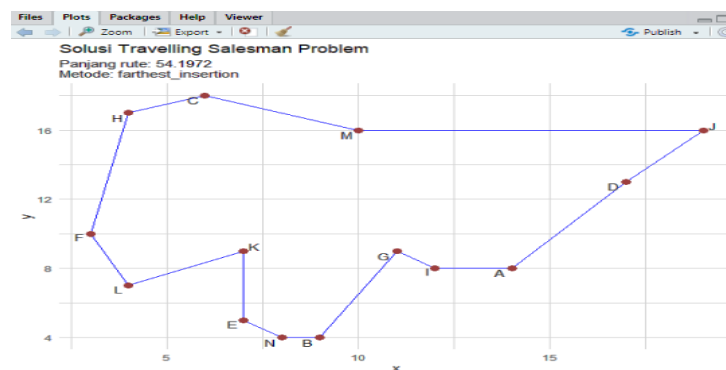
Gambar 11. Hasil *Tour* TSP

Langkah 5 Langkah terakhir yaitu membuat plot dari metode yang memiliki panjang *tour*/bobot paling minimum.

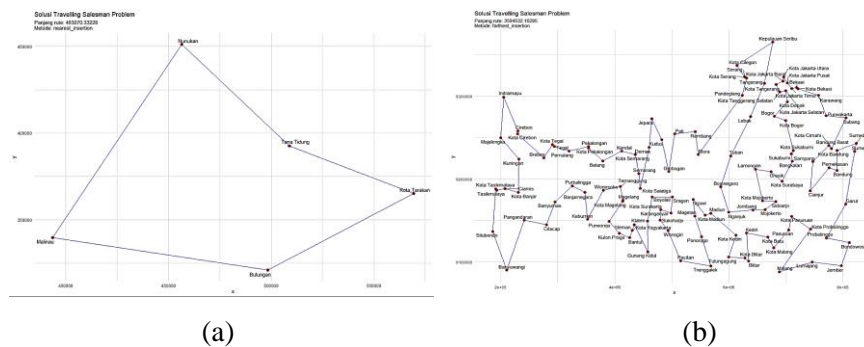
```

Console Terminal Jobs
G:/0/ ↵
> ggplot(rute_tur_min,aes(X,Y,label = rownames(rute_tur_min))) +
+ geom_polygon(colour = "blue", fill = NA) +
+ geom_point(colour = "firebrick", size = 3) +
+ geom_text_repel(size = 4) + #besar huruf
+ labs(x = "x",y = "y",title = "Solusi Travelling Salesman Problem",
+ subtitle = paste("Panjang rute:",round(min_tur, 5),"\nMetode:",names(min_tur))) +
+ theme_minimal()

```

Gambar 12. Pembuatan Plot Hasil *Tour* TSPGambar 13. Plot Hasil *Tour* TSP

Berdasarkan Gambar 13, penyelesaian *Travelling Salesman Problem* 14 Anjungan diketahui mempunyai panjang *tour*/bobot minimum sebesar 54,1972 dan tertulis metode *farthest insertion* yang artinya metode tersebut merupakan metode dengan bobot minimum dari ketiga metode untuk TSP 14 Anjungan.



Gambar 14. TSP : (a) Provinsi Kalimantan Utara, (b) Pulau Jawa

Contoh lainnya yaitu menggunakan koordinat titik di Provinsi Kalimantan Utara dan Pulau Jawa. Berdasarkan Gambar 14(a), penyelesaian TSP Kalimantan Utara dengan 5 titik mempunyai panjang rute minimum sebesar 463.070,33228 m dan tertulis metode *nearest insertion* yang artinya metode tersebut merupakan metode dengan bobot minimum. Pada Gambar 14(b), penyelesaian TSP Pulau Jawa dengan 119 titik diketahui panjang rute sebesar 3.594.532,16295 m dan tertulis metode *farthest insertion* yang artinya metode tersebut merupakan metode dengan bobot minimum.

4. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang diperoleh, hasil TSP untuk data yang sama dengan menggunakan tiga metode *insertion* tidak selalu menghasilkan rute dan panjang rute/bobot yang sama. Sebagai saran, penyelesaian TSP dapat diperoleh dengan menggunakan metode *insertion* lain seperti *arbitrary insertion* dan *nearest neighbor algorithm*. Selain itu, dan dapat dikaji lebih lanjut tentang permasalahan riil di anjungan seperti adanya waktu penyaluran muatan, prioritas yang paling membutuhkan, biaya angkut, dan banyaknya permintaan barang untuk setiap anjungan yang akan dikunjungi.

DAFTAR PUSTAKA

Applegate, D. L., Bixby, R. E., Chvátal, V., dan Cook, W. J., *The Traveling Salesman Problem : A Computational Study*, Princeton University Press, New Jersey, 2006.

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., dan Stein, C., *Introduction to Algorithms, Second Edition*, The MIT Press, London, 2001.
- Dorigo, M. dan Stützle, T., *Ant Colony Optimization*, The MIT Press, London, 2004.
- Goodaire, E. G., *Discrete Mathematics with Graph Theory*, Prentice Hall, Upper Saddle River, 1998.
- Hahsler, M. dan Hornik, K., *TSP-Infrastructure for the Travelling Salesperson Problem*, Journal of Statistical Software, **32**(2) (2007), 5-10.
- Hahsler, M. dan Hornik, K., *TSP-Infrastructure for the Travelling Salesperson Problem*, Cran R Project Org, Package 'TSP', 2020.
- Kusrini dan Istiyanto, J. E., *Penyelesaian Travelling Salesman Problem dengan Algoritma Cheapest Insertion Heuristics dan Basis Data*, Journal Informatika, **8**(2) (2007), 110-111.
- Munir, R., *Matematika Diskrit*, Edisi Ketiga, Informatika Bandung, Bandung, 2010.
- Prawidya, A., Pramono, B., dan Aksara, L. B., *Travelling Salesman Problem (TSP) untuk Menentukan Rute Terpendek bagi Kurir Kota Kendari Menggunakan Algoritma Greedy Berbasis Android*, **3**(1) (2017), 95-106.
- Rosen, K. H., *Discrete Mathematics and Its Application*, Edisi Kelima, WCB McGraw-Hill, New York, 2003.
- Rosenkrantz, D. J., Stearns, R. E., dan Lewis, P. M., *An Analysis of Several Heuristics for the Traveling Salesman Problem*, SIAM Journal on Computing, **6**(3) (1977), 570-577.
- Wilson, R. J. dan Watkins, J. J., *Graphs : An Introductory Approach*, John Willey and Sons, New York, 1990.
- Yuan Ze University, *The Traveling Salesman Problem (TSP) and its solving algorithm*, <http://logistics.iem.yzu.edu.tw/Courses/1041/TSP%20Problem.pdf>, diakses pada 13 April 2020.