PENGEMBANGAN ALGORITMA HYBRID NEAREST NEIGHBOUR & ANT COLONY OPETIMIZATION UNTUK MENYELESAIKAN TRAVELLING SALESMAN PROBLEM

Oleh :
Ilham Alpihuda
198030015
Program Magister Teknik Industri
Universitas Pasundan

Abstract. *Traveling Salesman Problem (TSP) dikenal sebagai salah satu masalah optimasi yang banyak menarik perhatian para peneliti sejak beberapa dekade terdahulu. Meskipun TSP sepertinya mudah dinyatakan, akan tetapi sangat sulit untuk diselesaikan terutama untuk persoalan dengan jumlah kota yang banyak. TSP dikarakteristikan kedalam kelas NP-hard (Nondeterministik Polynomial – hard), dimana jika titik yang harus dikunjungi berjumlah banyak. Untuk menyelesaikan permasalahan TSP, selain menggunakan metode heuristik ada metode yang lebih cerdas dalam mengeskplorasi solusi terbaik yaitu dengan menggunakan algoritma metaheurstik. Banyak algoritma metaheuristik yang berguna telah dikembangkan oleh para peneliti terdahulu untuk menyelesaikan masalah TSP, di penenelitian ini akan akan dibahas sebuah tema mengenai Pengembangan Algoritma Hybrid Nearest Neighbour (NN) & Ant Colony Optimization (ACO) untuk Menyelesaikan Travelling Salesman Problem (TSP). Algoritma yang telah dikembangkan diterjemahkan ke dalam bahasa pemrograman, sehingga penyelesaian TSP dengan menggunakan NN dan ACO dapat dilakukan dengan bantuan komputer. Penulisan program dilakukan di GNU Octave. Dari hasil pengujian pada algoritma Nearest Neighbour, Ant Colony Optimization juga Hybrid Nearest Neighbour & Ant Colony Optimization dari beberapa Instance yang dilakukan, maka keluar nilai jarak yang bervariatif berdasarkan perbandingan solusi, untuk setiap instance yang diselesaikan, algoritma yang dikembangkan tidak mampu menghasilkan total panjang seluruh rute yang lebih baik dibandingkan dengan algoritma-algoritma dari penelitian sebelumnya. Total panjang seluruh rute dari instance 2l_cvrp0101 merupakan total panjang seluruh rute terbaik yang mampu dihasilkan dibandingkan dengan penelitian-penelitian sebelumnya, dengan nilai perbandingan -1,43%. Total panjang seluruh rute terburuk yang mampu dihasilkan adalah total panjang seluruh rute dari instance 2l_cvrp1801, dengan nilai perbandingan -124,91% dibandingkan dengan GTS, -127,18% dibandingkan dengan ACO dan -127,90% dibandingkan dengan GRASPxELS. Secara rata-rata, total panjang seluruh rute yang dihasilkan mempunyai nilai perbandingan -63,72% dibandingkan dengan GTS, -68,28% dibandingkan dengan ACO dan -70,58% dibandingkan dengan GRASPxELS.*

Keywords: ***Traveling Slesman Problem, Nearest Naigbour, Ant Colony Optimization, Hybrid Algoritma***

1. Pendahuluan

Traveling Salesman Problem (TSP) dikenal sebagai salah satu masalah optimasi yang banyak menarik perhatian para peneliti sejak beberapa dekade terdahulu. Pada mulanya, TSP dinyatakan sebagai permasalahan dalam mencari jarak minimal sebuah tour tertutup terhadap sejumlah n kota dimana kota-kota yang ada hanya dikunjungi sekali dengan kota awal juga merupakan kota akhir (tujuan). TSP merupakan persoalan yang banyak diaplikasikan pada berbagai persoalan dunia nyata. Hingga saat ini, TSP diaplikasikan pada persoalan perencanaan pembangunan, perencanaan produksi, rute pengambilan surat dari kotak pos, rute pengisian uang pada mesin ATM, rute patroli polisi, rute pesawat terbang dan sebagainya.

Meskipun TSP sepertinya mudah dinyatakan, akan tetapi sangat sulit untuk diselesaikan terutama untuk persoalan dengan jumlah kota yang banyak. TSP dikarakteristikan kedalam kelas NP-hard (***Nondeterministik Polynomial – hard***), dimana jika titik yang harus dikunjungi berjumlah banyak, maka untuk mendapatkan rute terpendeknya diperlukan waktu yang sangat lama. Sebagai contoh jika

terdapat 50 titik yang harus dikunjungi maka akan terdapat 50! (3,04140932017134 x 10^{64}) alternatif rute, dengan alternatif rute sebanyak ini tentunya akan memerlukan waktu yang sangat lama untuk mendapatkan rute terbaiknya, sekalioun perhitungannya menggunakan super komputer tercanggih (D.L. Applegate, R.E. Bixby, V. Chvatal and W.J. Cook : 2006).

Algoritma yang perhitungannya cepat walaupun "hanya" menghasilkan solusi yang mendekati optimal, merupakan opsi yang sangat baik untuk menyelesaikan TSP . Algoritma jenis ini dikenal dengan sebutan Heuristik. "Heuristik" merupakan sebuah kata dari bahasa Yunani yang berarti "menemukan" atau "mengeksplorasi". Heuristik disebut sebagai *approximate techniques*. Tujuan utama dari Heuristik adalah untuk membangun sebuah model optimasi yang mudah dipahami dan mampu memberikan solusi yang baik dalam waktu perhitungan yang wajar (K. Kumar, D. Zindani and J. Paulo Davim : 2020). Beberapa Heuristik yang cukup populer untuk menyelesaikan TSP yaitu *Savings Algorithm, Nearest Neighbour, Nearest Insertion,* dan *Minimum Spanning Tree Heuristic*.

Untuk menyelesaikan permasalahan TSP, selain menggunakan metode heuristik ada metode yang lebih cerdas dalam mengeskplorasi solusi terbaik yaitu dengan menggunakan algoritma metaheurstik. Metode metaheuristik pertama sekali diperkenalkan pada tahun 1986, dan digambarkan sebagai alternatif strategi pencarian yang sangat baik, diatas ruang pencarian dengan harapan bisa menemukan hasil yang optimal. Banyak algoritma metaheuristik yang berguna telah dikembangkan seperti *Simulated Anneling, Algoritma Genetika, Tabu Search, Ant Colony* dan lain sebagainya.

Berdasarkan uraian diatas, maka pada penelitian ini akan dibahas sebuah tema mengenai ***Pengembangan Algoritma Hybrid Nearest Neighbour (NN) & Ant Colony Optimization (ACO) untuk Menyelesaikan Travelling Salesman Problem (TSP)***.

1.1 Perumusan Masalah

Berdasarkan tema yang telah ditentukan, penelitian ini diharapkan dapat menjawab beberapa pertanyaan berikut ini:

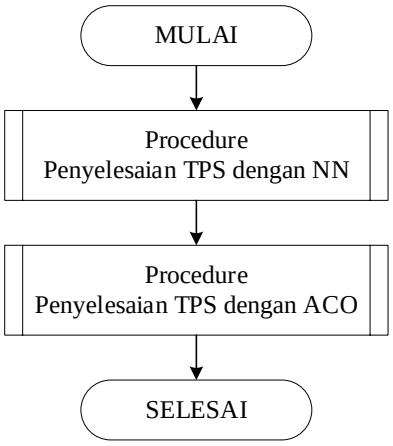
1. Bagaimana hasil solusi terbaik penyelesaian TSP menggunakan *Algoritma Nearest Neighbour* (NN) pada data *instance* yang telah ditetapkan ?
2. Bagaimana hasil solusi terbaik penyelesaian TSP menggunakan *Algoritma Ant Colony Optimization* (ACO) pada data *instance* yang telah ditetapkan?
3. Bagaimana Algoritma *Hybrid Nearest Neighbor & Ant Colony Optimization* yang dikembangkan dapat menyelesaikan TSP pada data *instance* yang telah ditetapkan?

1.2 Tujuan Penelitian

Penelitian ini mempunyai beberapa tujuan yang akan dilakukan, yaitu:
Untuk menguji Algoritma *Hybrid Nearest Neighbor & Ant Colony Optimization* yang dikembangkan dengan membandingkan hasil solusi terbaiknya degan solusi TSP *Nearest Neighbor* serta solusi TSP *Ant Colony Optimization.*

2. Pengembangan Algoritma ACO

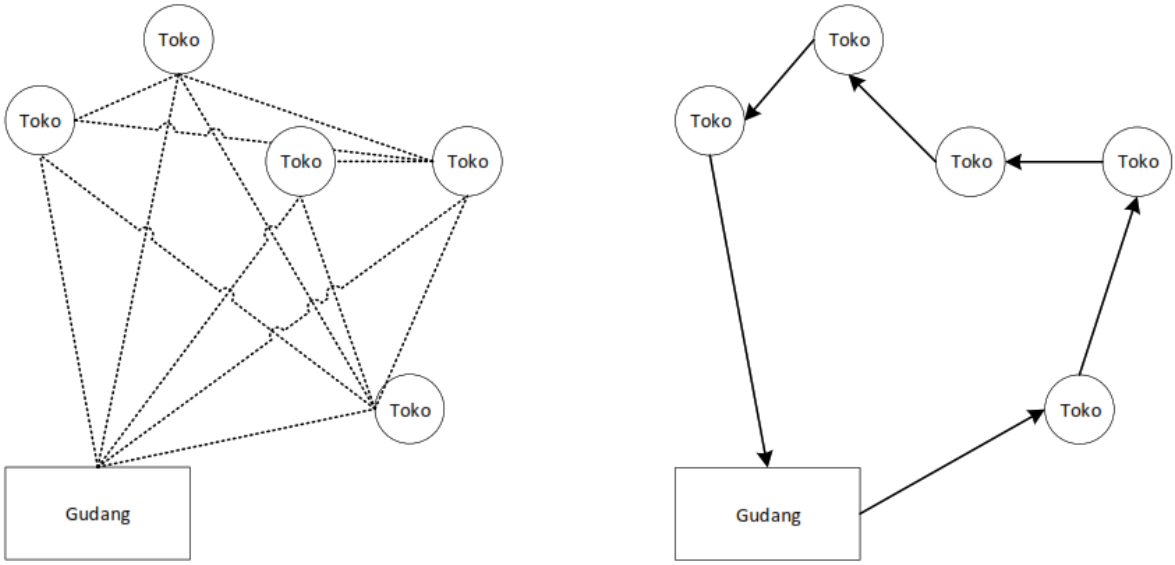
Untuk penyelesaan TSP, pada penelitian ini akan dikembangkan sebuah algoritma gabungan berdasarkan Algoritma *Nearest Neighbor* dan Algoritma ACO. Penggabungan dari kedua alogritma tersebut disebut sebagai Algoritma *hybrid Nearest Neigbor* & ACO. *Flowchart* utama dari pengembangan algoritma ini dapat digambarkan seperti pada gambar 1.



Gambar 1 *Flowchart* Utama

2.1 Langkah Algoritma Nearest Neighbour Untuk Penyelesaian TSP

Nearest neighbor metode heuristik yang digunakan untuk pemecahan masalah sebagai dasar untuk penentuan rute metode NN juga banyak digunakan. Algoritma heuristik yang memang berkinerja signifikan lebih baik dan realistis dalam pembentukan rute. Untuk sejumlah kecil kota, masalah dapat dengan mudah dan cepat diselesaikan dengan algoritma *nearest neighbor*.
Langkah pertama memasukkan tujuan ke dalam rute pengiriman, hal pertama yang harus dilakukan adalah mengurutkan nilai terkecil yang telah diperoleh mulai dari yang terbesar hingga yang terendah. Pada komputasi NN memiliki kinerja yang sangat cepat. NN ditemukan oleh Solomon pada tahun 1987 yang konsepnya adalah mengunjungi lokasi terdekat dari masing-masing lokasi yang sedang dikunjungi.



Gambar 2. Bentuk Rute *Nearest Neighbor*

Pada gambar 3.3 merupakan penentuan rute dengan menggunakan metode nearest neighbor dengan mencari jarak terdekat dari lokasi gudang kemudian berpindah dari satu toko ke toko yang lain Berikut pemecahan masalah dengan pada pendistribusian menggunakan NN kumpulan dari perjalanan atau rute yang tersimpan dalam urutan adalah hasil dari algoritma ini :

1. Dimulai dari gudang di setiap perjalanan atau rute pengiriman.
2. Mencari tujuan pengiriman barang yang belum dikunjungi dengan jarak yang paling terdekat dari lokasi awal dan tidak melebihi kapasitas kendaraan.

a. Jika tujuan pengiriman barang terpilih dan masih memiliki sisa kapasitas maka kembali ke langkah 2 dan diubah sebagai lokasi awal.
b. Jika kendaraan sudah tidak memiliki sisa kapasitas maka kembali ke langkah 1 buat perjalanan atau rute baru.

3. Jika semua tujuan pengiriman telah di kunjungi satu kali pada algoritma selesai

2.2 Langkah Algoritma Ant Colony Untuk Penyelesaian TSP

Untuk mempermudah pemecahan masalah TSP, maka harus dimodelkan ke dalam Graf. Secara umum, strategi ACO yang digunakan untuk masalah TSP adalah sebagai berikut. Pada tahap awal, tempatkan semut secara acak pada setiap simpul dan berikan feromon yang sama banyak setiap sisi dalam graf. Pada setiap langkah, setiap semut harus memilih simpul berikutnya yang akan dikunjungi. Simpul yang akan dikunjungi tersebut harus merupakan simpul yang belum ada dalam memori semut itu.Fungsi untuk memilih simpul berikutnya ini dinyatakan dalam Aturan Kemungkinan Transisi (*Probabilistic Transition Rule*). Jumlah feromon pada sisi yang dilalui oleh semut akan diperbaharui jumlahnya, yakni dikurangi dan ditambah. Fungsi untuk mengurangi dan menambahkan feromon yang ditinggalkan oleh semut dinyatakan dalam Aturan Pembaharuan Feromon Lokal (*Local Pheromon Updating Rule*). Kemudian, jika setiap semut telah mengunjungi semua simpul dan kembali lagi ke simpul asal, maka akan diperoleh lebih dari satu kemungkinan solusi jarak tempuh karena ada lebih dari satu semut yang melintasi setiap sisi dalam graf. Keadaan dimana setiap semut telah mengunjungi setiap simpul dalam graf dan kembali lagi ke simpul asal dinamakan satu iterasi. Pada strategi ACO ini, digunakan lebih dari satu kali iterasi. Pada akhir setiap iterasi akan dihasilkan lebih dari satu kemungkinan solusi.Langkah selanjutnya adalah memilih solusi yang terbaik sejauh ini, yaitu solusi dengan jarak tempuh minimum. Kemudian, tambahkan feromon pada lintasan terpendek yang dihasilkan dari solusi tersebut. Fungsi untuk menambahkan feromon ini dinyatakan dalam Aturan Pembaharuan Feromon Global (*Global Pheromon Updating Rule*). Tujuan penambahan feromon global ini adalah untuk mendorong semut – semut pada iterasi berikutnya agar memilih lintasan terpendek yang telah dicapai sejauh ini. Selanjutnya, jalankan metode di atas untuk setiap iterasi. Setelah iterasi yang terakhir selesai dijalankan, pilihlah jarak tempuh minimum dari berbagai kemungkinan solusi yang dihasilkan pada setiap iterasi. Jarak tempuh minimum itulah yang merupakan solusi atas masalah TSP.

2.2.1 Aturan Kemungkinan Solusi

Kemungkinan simpul s akan dipilih oleh semut k yang sedang berada pada simpul r dinyatakan dalam perhitungan matematika secara heuristik sebagai berikut:

$$s = \begin{cases} \arg\max_{u \in M_k} \left\{ [\tau(r,u)] \cdot [\eta(r,u)]^{\beta} \right\} & \text{jika } q < q_0 \text{ (Eksploitasi)} \\ S & \text{jika } q >= q_0 \text{ (Eksplorasi)} \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots (3.1)$$

Keterangan :

τ(r,u)	: jumlah feromon pada sisi dari simpul r ke simpul s.
η(r,u)	: (panjang sisi dari simpul r ke simpul s)- 1
β	: parameter perbandingan jumlah feromon relatif terhadap jarak (merupakan parameter yang telah ditentukan sebelumnya)
Mk	: himpunan yang berisi simpul – simpul yang telah dikunjungi oleh semut
U	: simpul yang berada dalam Mk
q	: bilangan random
q0	: parameter perbandingan eksploitasi terhadap eksplorasi
S	: simpul berikutnya yang dipilih berdasarkan persamaan (3.2)
Eksploitasi	: semut mengeksploitasi simpul yang telah dikunjungi sebelumnya
Eksplorasi	: semut mengeksplorasi simpul yang belu pernah dikunjungi sebelumnya.

2.2.2 Aturan Pembaharuan Lokal

Jumlah feromon $\tau(r,s)$ akan berubah sesuai dengan aturan dibawah ini :
$\tau(r,s) \leftarrow (1 - \rho) \times \tau(r,s) + \rho \times \Delta\tau(r,s)$

Keterangan :
$\tau(r,s)$: jumlah feromon pada sisi dari simpul r ke simpul s.
ρ : parameter lokal feromon yang hilang
$\Delta\tau(r,s)$: jumlah total feromon pada sisi dari simpul r ke s

2.2.3 Aturan Pembaharuan Feromon Global

Jumlah feromon $\tau(r,s)$ akan berubah sesuai dengan aturan dibawah ini :
$\tau(r,s) \leftarrow (1 - \alpha) * \tau(r,s) + \alpha * \Delta\tau(r,s)$

Keterangan :
$\tau(r,s)$: jumlah feromon pada sisi dari simpul r ke simpul s.
α : parameter global feromon yang hilang
$\Delta\tau(r,s)$: 1 / jarak tempuh lintasan terpendek sejauh ini (*best solution so far*) jika simpul r dan simpul s ada dalam lintasan tersebut.

2.3 Pengembangan Algoritma *Hybrid Nearest Neighbour* & *Ant Colony*

Pengembangan Algortima Hybrid NN & ACO bertujuan untuk membangun solusi TPS yang lebih baik dari hasil solusi yang didapatkan dari masing-masing Metode NN dan ACO.

Pada metode ACO, tahap inisiasi awal / solusi awal didapat dari iterasi pertama pada semut yang ditempatkan secara random. Pengembangan algoritma Hybrid yang dilakukan yaitu pada tahap inisiasi awal/solusi awal penempatan semut berdasarkan dari hasil solusi NN yang telah di komputerisasi sebelumnya.

3.3 Pengujian Algoitma

Algoritma yang telah dikembangkan diterjemahkan ke dalam bahasa pemrograman, sehingga penyelesaian TSP dengan menggunakan NN dan ACO dapat dilakukan dengan bantuan komputer. Penulisan program dilakukan di GNU Octave. Program yang ditulis berdasarkan algoritma yang dikembangkan. Dan Instance yang akan digunakan pada penelitian ini adalah instance yang telah digunakan di penelitian-penelitian sebelumnya dalam menguji algoritma yang dikembangkannya.

Ada 11 instance dalam pengujian algoritma yang dilakukan sebagaimana berikut :

Burma		
NO	X	Y
1	16,47	96,1
2	16,47	94,44
3	20,09	92,54
4	22,39	93,37
5	25,23	97,24
6	22	96,05
7	20,47	97,02
8	17,2	96,29
9	16,3	97,38
10	14,05	98,12
11	16,53	97,38
12	21,52	95,59
13	19,41	97,13
14	20,09	94,55

No	Instance	Class	Number of Customers	Number of Items	Vehicle Capacity	Container Height	Container Width
111	2l_cvrp0304	4	20	44	85	40	20
112	2l_cvrp0404	4	20	50	58	40	20
113	2l_cvrp0504	4	21	41	6000	40	20
114	2l_cvrp0604	4	21	57	4000	40	20
115	2l_cvrp0704	4	22	51	4500	40	20
116	2l_cvrp0804	4	22	48	4500	40	20
117	2l_cvrp0904	4	25	63	48	40	20
118	2l_cvrp1004	4	29	72	4500	40	20
119	2l_cvrp1104	4	29	74	4500	40	20
120	2l_cvrp1204	4	30	82	68	40	20
121	2l_cvrp1304	4	32	78	38000	40	20
122	2l_cvrp1404	4	32	65	8000	40	20
123	2l_cvrp1504	4	32	84	8000	40	20
124	2l_cvrp1604	4	35	93	67	40	20
125	2l_cvrp1704	4	40	96	60	40	20
126	2l_cvrp1804	4	44	112	2010	40	20
127	2l_cvrp1904	4	50	134	160	40	20
128	2l_cvrp2004	4	71	178	30000	40	20
129	2l_cvrp2104	4	75	168	220	40	20
130	2l_cvrp2204	4	75	198	180	40	20
131	2l_cvrp2304	4	75	179	140	40	20
132	2l_cvrp2404	4	75	195	100	40	20
133	2l_cvrp2504	4	100	254	200	40	20
134	2l_cvrp2604	4	100	247	200	40	20
135	2l_cvrp2704	4	100	245	112	40	20
136	2l_cvrp2804	4	120	299	200	40	20
137	2l_cvrp2904	4	134	342	2210	40	20
138	2l_cvrp3004	4	150	366	200	40	20
139	2l_cvrp3104	4	199	513	200	40	20
140	2l_cvrp3204	4	199	497	200	40	20
141	2l_cvrp3304	4	199	499	200	40	20
142	2l_cvrp3404	4	240	604	200	40	20
143	2l_cvrp3504	4	252	634	1000	40	20
144	2l_cvrp3604	4	255	606	1000	40	20
145	2l_cvrp0105	5	15	45	90	40	20
146	2l_cvrp0205	5	15	48	55	40	20
147	2l_cvrp0305	5	20	49	85	40	20
148	2l_cvrp0405	5	20	62	58	40	20
149	2l_cvrp0505	5	21	57	6000	40	20
150	2l_cvrp0605	5	21	56	4000	40	20
151	2l_cvrp0705	5	22	55	4500	40	20
152	2l_cvrp0805	5	22	52	4500	40	20
153	2l_cvrp0905	5	25	91	48	40	20
154	2l_cvrp1005	5	29	86	4500	40	20
155	2l_cvrp1105	5	29	91	4500	40	20
156	2l_cvrp1205	5	30	101	68	40	20
157	2l_cvrp1305	5	32	102	38000	40	20
158	2l_cvrp1405	5	32	87	8000	40	20
159	2l_cvrp1505	5	32	114	8000	40	20
160	2l_cvrp1605	5	35	114	67	40	20

Setelah hasil running pemrograman tersebut, maka dapat dilihat pada table 1 bahwa jarak terpendek pada beberapa instance dari algoritma Nearest neighbour, Ant Colony Optimization dan hybrid Nearest Neighbour & Ant Colony sebagai berikut :

Tabel 1. Hasil Perhitungan Alogirtma

Instance	Jumlah Titik	NN	ACO	Hybrid
Burma	14	38,6	39,99	36,59
Berlin	52	8980,91	18278,82	8980,91
Eli	51	513,61	1097,15	5136,61

4. Analisa Hasil Pengujian Algoritma

Dari hasil pengujian pada algoritma Nearest Neighbour, Ant Colony Optimization juga Hybrid Nearest Neighbour & Ant Colony Optimization dari beberapa Instance yang dilakukan, maka keluar nilai jarak yang bervariatif berdasarkan perbandingan solusi, untuk setiap *instance* yang diselesaikan, algoritma yang dikembangkan tidak mampu menghasilkan total panjang seluruh rute yang lebih baik dibandingkan dengan algoritma-algoritma dari penelitian sebelumnya. Total panjang seluruh rute dari *instance 2l_cvrp0101* merupakan total panjang seluruh rute terbaik yang mampu dihasilkan dibandingkan dengan penelitian-penelitian sebelumnya, dengan nilai perbandingan -1,43%. Total panjang seluruh rute terburuk yang mampu dihasilkan adalah total panjang seluruh rute dari *instance 2l_cvrp1801*, dengan nilai perbandingan -124,91% dibandingkan dengan *GTS*, -127,18% dibandingkan dengan *ACO* dan -127,90% dibandingkan dengan GRASPxELS. Secara rata-rata, total panjang seluruh rute yang dihasilkan mempunyai nilai perbandingan -63,72% dibandingkan dengan *GTS*, -68,28% dibandingkan dengan *ACO* dan -70,58% dibandingkan dengan GRASPxELS.

Berdasarkan hasil Uji, algoritma Nearest Neighbour lah yang dapat menghasilkan jarak paling dekat diantara Ant Colony Optimization dan Hybrid Nearest Neighbour & Ant Colony Optimization pada 11 Instance. Hanya ada 1 instance yaitu Burma dengan 14 iterasi yang berhasil dapat memperbaiki algoritma Nearest Neighbour dengan algoritma Hybrid Nearest Neighbour & Ant Colony Optimization.

5. Kesimpulan

Setelah menganalisa terhadap algoritma yang dikembangkan berdasarkan kemampuannya menyelesaikan instance yang telah ditentukan dapat diperoleh beberapa kesimpulan sebagaimana berikut :

1. Algoritma Nearest Neighbour belum bisa diperbaiki sama Algoritma Ant Colony Optimization, sehingga hasilnya tetap saja tidak ada perbaikan.
2. Algoritma Ant Colony Optimization Belum baik dalam memperbaiki Nearest Neighbour dari 11 instance, dan hanya 1 instance yang berhasil yang mana iterasinya paling sedikit dari instance yang lainnya yaitu 14 iterasi, mungkin dikarenakan peneliti baru memakai satu kombinasi parameter. Untuk kedepannya bisa dicoba dengan menggunakan kombinasi parameter yang lain.

DAFTAR PUSTAKA

[1] Kumar, Kaushik., Zindani, Divya. & Davim, J. Paulo., 2020, Optimizing Engineering Problems through Heuristic Techniques, CRC Press, USA.

[2] Talbi, El-Ghazali., 2009, Metaheuristics from Design to Implementation, John Wiley & Sons, USA.

[3] Marco Dorigo and Thomas Stutzle, “Ant Colony Optimization," The MIT Pres Cambride London, 2004.

[4] Andreescu, Titu. & Andrica, Dorin., 2014, Complex Numbers from A to...Z, 2nd Edition, Springer Science + Business Media, USA.

[5] Hansen, Jesper Schmidt., 2011, GNU Octave Beginner's Guide, PACKT,Birmingham.

[6] Korte, Bernhard. & Vygen, Jens., 2018, Combinatorial Optimization Theory and Algorithms, 6 Edition, Springer, Germany.

[7] Kusumadewi, Sri., Penyelesaian Masalah Optimasi dengan Teknik-Teknik Heuristik.

[8] David L., Robet E. Vasek, William J, "The Travelling Salesman Problem," Princeton Univseristy Press, 2006.