

EVALUASI PENERAPAN *EVOLUTIONARY ALGORITHM* UNTUK PEMECAHAN *TRAVELING SALESMAN PROBLEM*

Herman Ruswan Suwarman

Program Studi Teknik Industri, Sekolah Tinggi Teknologi Bandung

suwarman.mac@gmail.com

Abstrak

Komputasi dalam memecahkan *Traveling Salesman Problem* (TSP) dihadapkan pada kombinasi solusi yang meningkat secara eksponensial seiring dengan bertambahnya node pada rute. Oleh karena itu, diperlukan pengembangan metode dalam memecahkannya. Penelitian ini akan menggunakan *Evolutionary Algorithm* (EA), salah satu algoritma heuristik untuk menangani beban komputasi tersebut. Walaupun demikian algoritma heuristik selalu dihadapkan dengan tantangan konvergensi. Konvergensi solusi yang diperoleh dari algoritma eksak seperti *Linear Programming* (LP) tidak selalu bisa dicapai oleh algoritma heuristik. Solver merupakan *add in program* Microsoft Excel yang mampu memecahkan masalah komputasi TSP melalui pendekatan EA (heuristik) dan *simplex LP* (eksak). Tujuan penelitian ini adalah pertama mendapatkan gambaran mengenai kinerja suatu metode heuristik EA dalam memecahkan TSP. Kedua, mendapatkan gambaran mengenai kompleksitas yang dihasilkan akibat penambahan jumlah node pada TSP. Sampel yang digunakan adalah jarak gudang dan 14 gerai minimarket terdekat di Kota Bandung. TSP untuk 15 node tersebut dipecahkan menggunakan EA dan kemudian dibandingkan dengan luaran dari hasil komputasi LP. Hasil perbandingan dianalisis secara deskriptif yaitu dengan melihat pola-pola yang dihasilkan oleh kedua metode. Semua komputasi baik EA dan LP menggunakan fitur yang disediakan oleh Microsoft Excel Solver. Hasil penelitian menunjukkan bahwa EA memiliki kualitas dalam mendekati konvergensi LP dengan baik dan layak digunakan dalam mendapatkan solusi pada jumlah node yang banyak. Hal ini ditunjukkan oleh pola luaran dari kedua metode dan linearitas yang dihasilkannya. Selain itu juga ditunjukkan pada node yang cukup banyak pada kasus ini, LP menghasilkan solusi tidak layak, di pihak lain solusi tersebut masih bisa diberikan oleh EA.

Kata Kunci : TSP, komputasi, heuristik, *evolutionary algorithm*, Microsoft Excel Solver, deskriptif

Abstract

Computing in solving the Traveling Salesman Problem (TSP) is faced with a combination of solutions that increases exponentially as the number of nodes on the route increases. It is necessary to develop a method to solve it. One of them is Evolutionary Algorithm (EA). Computing the Traveling Salesman Problem has been challenged by the increase of solution combination exponentially as the increasing number of nodes. Therefore, development of methods and algorithm has been called for years to solve it. This research used Evolutionary Algorithm (EA) that one of the heuristic approach to handle this problem. However, it comes with convergence problem which is the necessary condition that have to be met to achieve optimal condition as we found in exact algorithm (e.g. linear programming). Solver is add in program of Microsoft Excel that has ability to solve the TSP computation using EA and simplex LP. The aims of this research are, first, to get the explanation of EA performance to solve TSP. Secondly, to describe the complexity that is resulted by the increase of node in TSP. The research's sample took the distance data of a warehouse and 14 outlets of one the branded minimarket in Bandung city. TSP of the 15 nodes were solved using EA and compared with the result that was yielded by LP computation. Both results were analysed descriptively to identify patterns of these two methods. All computation was done by the features that is served in Microsoft Excel Solver. The evidence of linearity patterns shows the EA performance is good quality to approach LP's convergence. Aside from that, LP's solution showed the infeasibility in a high number of nodes that EA can handle this problem.

Keywords: TSP, computation, heuristic, evolutionary algorithm, Microsoft Excel Solver, descriptive

I. PENDAHULUAN

Traveling Saleman Problem (TSP) merupakan model jaringan untuk menemukan jarak tempuh ataupun biaya perjalanan paling minimum dimana hanya mengijinkan suatu perjalanan melewati sebuah titik (node) satu kali saja. Hal ini yang memberikan tingkat kompleksitas tersendiri dalam memecahkannya terutama jika berhadapan dengan sejumlah besar node beserta keterhubungannya satu sama lain. Dalam mengambil keputusan TSP dihadapkan pada kombinasi alternatif yang sangat banyak. Oleh karena itu, model-model pemecahan TSP berkembang dari mulai menggunakan *exact algorithm* dan bersifat deterministik seperti *Linear Programming* (LP) hingga model heuristik.

Penerapan metode heuristik secara umum mampu mengurangi kompleksitas pemecahan TSP[1][6], walaupun belum tentu bisa mendapatkan solusi yang optimal (konvergensi terhadap titik optimal global. Microsoft Excel menampilkan suatu program *add in* bernama Solver dimana sudah menggunakan metode heuristik yaitu *Evolutionary Algorithm* (EA) untuk memecahkan masalah optimisasi disamping metode lainnya yaitu metode non linear GRG (*Generalized Reduced Gradient Methods*) dan Simplex LP (*Linear Programming*). Pada penelitian ini akan digunakan metode heuristik (EA)

dan Simplex LP dengan tujuan mendapatkan perbandingan dalam menilai kinerja metode heuristik yang tersedia dalam program Solver Microsoft Excel.

Adapun tujuan dalam penelitian ini adalah sebagai berikut:

1. Mendapatkan gambaran mengenai kinerja suatu metode heuristik *Evolutionary Algorithm (EA)* dalam memecahkan TSP dalam kasus pengiriman barang dari sebuah gudang mini market ke gerai-gerainya.
2. Mendapatkan gambaran mengenai kompleksitas yang dihasilkan akibat penambahan jumlah node pada sebuah TSP.

II. TINJAUAN PUSTAKA

1. *Traveling Salesman Problem*

Dalam sebuah literatur cukup lama (1954), Dantzig dkk memberikan pemahaman TSP yang bisa digambarkan dengan rumusan berikut ini: “Bagaimana menemukan rute terpendek seorang *salesman* yang berangkat dari sebuah kota, mengunjungi beberapa kota dan kemudian kembali ke kota asal” [2]. Dalam bahasa matematis, masalah TSP ini adalah situasi menemukan rute terpendek dalam kunjungan ke n-buah kota dimana setiap kota dikunjungi hanya sekali saja [10]. Secara spesifik, n-kota tersebut didefinisikan [10][3]:

$$x_{ij} = \begin{cases} 1, & \text{jika kota } j \text{ dicapai dari kota } i \\ 0 & \text{jika sebaliknya} \end{cases}$$

diberikan d_{ij} adalah jarak dari kota i ke j , model TSP diberikan sebagai

$$\text{Minimasi } z = \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \quad , d_{ij} = \infty \text{ untuk semua } i = j \quad (1)$$

dengan pembatas

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n$$

$$x_{ij} = (0, 1)$$

Dimana solusi dari formula *Integer Linear Programming (ILP)* ini akan menghasilkan sebuah urutan rute kunjungan *salesman* ke n-kota. Pemecahan solusi ini bisa digunakan prosedur pemecahan ILP seperti *Branch and Bound (BB)* dan *Cutting Plane (CP)* selain juga bisa dilaksanakan dengan metode *Dynamic Programming* [10]. Kendala dalam menggunakan prosedur pemecahan ini adalah munculnya masalah kombinatorial pada *constrain* (pembatas). Kombinasi permutasi p pada matrik $a(i, j)$ yang berasal dari matriks jarak antar kota sejumlah n akan muncul mengikuti persamaan berikut ini [2]:

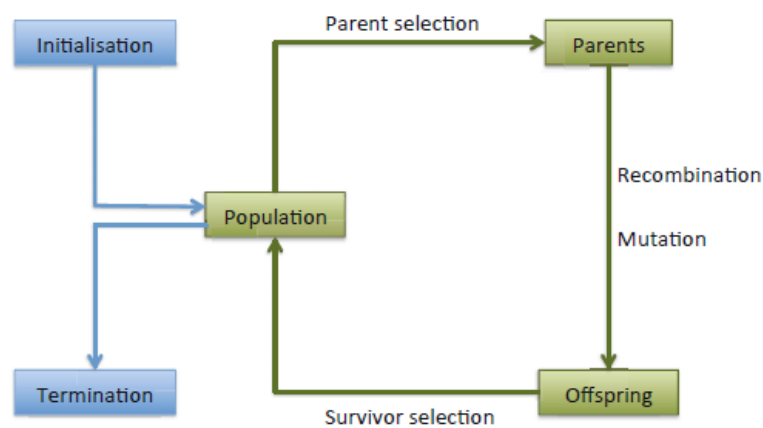
$$\max_p \sum_{i=1}^n a_{i,p(i)} \quad (2)$$

Dengan kondisi titik ekstrim pada himpunan *convex* akan berjumlah sebanyak n^2 . Hal ini yang menyebabkan prosedur pemecahan ILP konvensional akan memakan komputasi yang cukup lama dalam mengatasi sekian banyak solusi alternatif. Prosedur ILP seperti ini sering disebut dengan algoritma eksak (*exact algorithm*) yang akan selalu terkait dengan prosedur-prosedur iteratif yang pasti dan tertentu arahnya. Pada perkembangannya telah muncul prosedur-prosedur heuristik yang lebih memudahkan analisis dalam melakukan komputasi. Contoh dari prosedur heuristik adalah *Nearest Neighbor Algorithm (NNA)*, *Nearest Insertion Algorithm (NIA)*, *2-opt Algorithm (2OPT)*, *Simulated Annealing*, *Microcanonic Annealing*, *Trashhold Accepting Method*, *Noising Method*, *Tabu Search*, *Evolutionary Computation*, *Genetic Algorithm*, dan masih banyak lagi [1][7][9]. Dalam perkembangan terakhir, walaupun TSP merupakan permasalahan linear, namun komputasinya bisa menggunakan model non linear seperti *Mixed Integer Non Linear Programming (MINLP)* yang digunakan *open-source optimizer software COUENNE*. Kemampuan MINLP ini adalah memecahkan LP yang mempunyai fungsi tujuan dan pembatas bersifat non linear dan *non-convex* [7][3].

2. *Evolutionary Algorithm*

Dalam penelitian ini digunakan suatu algoritma pemecahan TSP heuristik yang bernama *Evolutionary Algorithm (EA)*. EA secara umum adalah sebuah algoritma optimisasi yang terinspirasi dari teori Darwin mengenai kapabilitas alami sebuah individu bisa beradaptasi dengan baik di alam [1][6]. Gagasan teknik EA ini adalah: suatu populasi individu dalam suatu lingkungan (*environment*) dimana didalamnya mereka mempunyai keterbatasan sumber daya, berkompetisi untuk mendapatkan sumber daya tersebut sehingga mengakibatkan seleksi alam (*survival of the fittest*). Keadaan ini menyebabkan sebuah kemunculan populasi yang berisi individu yang paling mampu beradaptasi (*survival of the fittest*) [6]. Adapun komponen dari EA adalah sebagai berikut [1][6]:

- Representation* – mendefinisikan individu yang melibatkan representasi genotip dan pemetaan genotip-fenotip. Contoh : jika kita akan mereprestasikan sebuah integer (fenotip) sebagai array biner (genotip), semisal 19 menjadi 10011 dan 106 menjadi 1101010. Jika dimutasikan pada bit pertama, maka diperoleh ‘3’ (00011) dan ‘42’ (0101010).
- Evaluation function (fitness function)* – kebutuhan/persyaratan populasi agar mampu beradaptasi. Dalam contoh model matematis : memaksimalkan/meminimumkan x^2 .
- Population*- peran populasi adalah sebagai sumber pembangkitan solusi. Sebuah populasi terdiri dari himpunan-himpunan genotip. Populasi ini membentuk unit pada proses evolusi.
- Parent Selection Mechanism*- peran dari *parent selection* ini adalah membedakan individu yang ada berdasarkan kualitas dan secara khusus akan menjadikan individu yang lebih baik untuk menjadi *parent* di generasi berikutnya.
- Variation Operators (Mutation and Recombination)* – peran dari *variation operators* ini adalah menciptakan individu baru dari individu yang lama. Untuk mengisi ruang fenotip sejumlah individu baru ini dibangkitkan untuk menjadi kandidat-kandidat solusi yang baru. *Variation Operators* terbagi dalam 2 jenis berdasarkan *arity*, *distinguish unary* (mutasi), dan *n-ary* (rekombinasi)
- Mutation*- salah satu *variation operator* adalah *mutation*. Mutasi ini dilaksanakan terhadap genotip dan menghasilkan *mutant* (anak, *child/offspring*).
- Recombination* – disebut juga *crossover*. Sebuah operator stokastik yang memilih bagian mana dari setiap *parents* yang bisa dikombinasikan dan bagaimana cara melakukannya.
- Survivor Selection Mechanism (Replacement)*- hampir sama dengan dengan langkah *parent selection* , hanya saja langkah ini dilakukan setelah *child* sudah dibentuk.



Gambar 1 Skema umum Flow Chart EA (sumber:

(sumber: Eiben, A.E., dan Smith, J.E., Introduction to Evolutionary Computing, Springer, Heidelberg, 2015)

Untuk menjalankan komputasi ini EA (gambar 1) dimulai dengan tahap *initialisation* dimana populasi awal harus dibuat (*population seed*) dengan cara membangkitkan secara random individu-individu. Setelah itu harus diteapkan kondisi *termination* yaitu bagaimana suatu komputasi bisa menerima solusi saat mencapai *optimal fitness*.

3. Microsoft Excel Solver

Komputasi pada penelitian ini menggunakan *add in program* dari Microsoft Excel bernama Solver. Dalam memecahkan komputasi, Solver menyediakan tiga algoritma yaitu: *Evolutionary Algorithm*(EA), *Generalized Reduced Gradient* (GRG), dan *Simplex Linear Programming* (Simplex LP). Dalam memecahkan TSP bisa digunakan EA dan Simplex LP [5][9]. Program ini berada pada *Tool* pada Excel. Jika tidak ada bisa melakukan instalasi dari Excel. Solver ini dibangun oleh Frontline System Inc suatu perusahaan yang membangun program serupa untuk Lotus 123 dan QuatroPro [9]. Untuk mendapatkan informasi lebih lanjut bisa mengunjungi situs perusahaan tersebut (<http://www.solver.com>). Model matematika yang diterapkan pada *spreadsheet* ini disebut dengan model *spreadsheet*. Dalam mengolah model matematika *spreadsheet* ini bisa digunakan sel-sel yang terpisah dalam merepresentasikan variabel keputusan, menuliskan formula pada sel untuk mengekspresikan fungsi tujuan dan menuliskan formula pada sel untuk merepresentasikan sistem persamaan pembatas [5]. Beberapa *paper terpilih* terlampir yang bisa digunakan sebagai panduan untuk menggunakan Solver [5][9]. Secara umum langkah-langkah dari penggunaan Solver adalah sebagai berikut [5][9]:

- Membuat *Excel Worksheet*- disesuaikan dengan matriks fungsi tujuan dan sistem persamaan pembatas.
- Menggunakan Solver dialogs- mendefinisikan tujuan (objective), variabel, parameter, dan set eksperimen)
- Menemukan dan menggunakan solusi yang dihasilkan- menjalankan program Solver untuk mendapatkan solusi.

Dalam memecahkan TSP, tingkat kompleksitas dari masalah ini adalah tidak boleh terjadi kunjungan terhadap node lebih dari 1 kali, oleh karena itu pemecahan dilakukan dengan mendefinisikan variabel sebagai keputusan biner selain juga mendefinisikan pembatas yang tidak boleh mengijinkan node dilalui lebih dari 1 kali. Mengenai hal terakhir, Solver menyediakan fitur “all different” yang tidak mengijinkan node bisa dilewatkan lebih dari 1 kali [5][9].

III. HASIL DAN PEMBAHASAN

Penelitian ini bertujuan untuk mendapatkan gambaran umum penggunaan algoritma heuristik pada suatu studi terpilih. Algoritma heuristik yang dipilih adalah Algoritma *Evolutionary*. Dalam penelitian ini, konvergensi algoritma tersebut akan dibandingkan secara kualitatif dengan metode deterministik yaitu *Linear Programming*. Penyelidikan dilakukan dengan mengolah data dari beberapa kondisi jumlah node dari 5 node hingga 15 node, untuk melihat kemungkinan pola yang muncul. Adapun komputasi yang dilakukan selama penelitian ini berlangsung dilaksanakan dengan bantuan program Solver yang merupakan *add-in program* dalam Microsoft Excel.

Adapun sampel yang digunakan adalah pengiriman barang dari gudang mini market ke 14 gerai terdekat. Dalam hal ini posisi gudang selalu menjadi sebuah node awal dan akhir. Untuk data jarak antara ke 15 node ini diperoleh dari situs <https://maps.google.com>.

Adapun asumsi yang digunakan dalam penelitian ini adalah:

- Gudang merupakan node asal (sumber) dan node akhir (tujuan). Setiap iterasi yang menghasilkan gudang tidak di posisi node asal (sumber) dan node (akhir) akan diolah agar tetap mengikuti asumsi ini dengan cara mengubah urutannya.
- Menggunakan data jarak 1 gudang distribusi dan 14 gerai yang terdapat di situs <https://maps.google.com>.
- Setiap node terhubung dengan semua node, Sehingga representasi jumlah node adalah jumlah busur (variabel keputusan) yang berjumlah $(n-1)! n$.

Penelitian bermula dengan mengambil data gudang dan 14 gerai minimarket di Kota Bandung yang diperoleh dari situs <https://maps.google.com> (tabel I).

TABEL I
Matriks Jarak Gudang dan 14 Gerai Salah Sebuah Perusahaan Minimarket Terdekat di Kota Bandung dalam km

No	Alamat Indomaret	Kode Node	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Gudang Indomaret Bandung Jl. Sukawangi No.05/06	0		1,3	3,2	2,2	2	2,6	1,8	2,3	2,5	2,7	3	2	1,5	1,5	2,2
2	Indomaret Padasuka bawah (Surapati Core) Jl. PH.H. Mustofa No. 45 - (021) 15	1	1,3		2,4	1,6	1,4	1,4	1	1,7	1,8	2,1	2,4	2	0,85	0,65	1,6
3	Indomaret Plus Terusan Jakarta Jl. Terusan Jakarta No.31	2	3,2	2,4		3,1	0,65	1,7	1,5	2,4	1,3	3,1	1	3,5	2,4	2,4	3,1
4	Ah Indomaret Nasution 888 Jl. Jendral Ahmad Yani No.888	3	2,2	1,6	3,1		1,6	2,2	1,4	1,1	2,1	0,5	2,3	2,2	1,1	1,1	1,8
5	Indomaret Jl. Sulaksana	4	2	1,4	0,65	1,6		1,9	0,8	1,6	1,5	2,9	0,95	2,8	1,6	1,7	2,3
6	Indomaret Cikutra Jl. Cikutra No.116	5	2,6	1,4	1,7	2,2	1,9		1,1	2,1	0,45	3,4	2,9	3,3	2,1	1,9	2,8
7	Indomaret Jl. Jendral Ahmad Yani No.723	6	1,8	1	1,5	1,4	0,8	1,1		1,5	0,7	2,7	1,8	2,6	1,5	1,5	2,2
8	Indomaret Jl. Jatihandap	7	2,3	1,7	2,4	1,1	1,6	2,1	1,5		3,2	1,6	3,4	1,5	0,02	1,4	1,1
9	Indomaret Cikutra Sukamulya Jl. Cikutra	8	2,5	1,8	1,3	2,1	1,5	0,45	0,7	3,2		0,65	2,5	3,3	2,1	2,2	2,9
10	Indomaret Jl. Jendral Ahmad Yani	9	2,7	2,1	3,1	0,5	2,9	3,4	2,7	1,6	0,65		2,1	1,7	0,6	0,65	1,3
11	Indomaret Ters.Jakarta 108 Jl. Ters Jakarta No.108	10	3	2,4	1	2,3	0,95	2,9	1,8	3,4	2,5	2,1		2,9	3	3	3,4
12	Indomaret AH Nasution 68 Jl. A.H. Nasution No.68	11	2	2	3,5	2,2	2,8	3,3	2,6	1,5	3,3	1,7	2,9		2,2	2,3	0,45
13	Indomaret Jatihandap Bawah	12	1,5	0,85	2,4	1,1	1,6	2,1	1,5	0,02	2,1	0,6	3	2,2		1,4	1,1
14	Indomaret Hybrid Padasuka 2 Jl. Padasuka No.59	13	1,5	0,65	2,4	1,1	1,7	1,9	1,5	1,4	2,2	0,65	3	2,3	1,4		1,8
15	Indomaret Cikadut Jl. A.Nasution No.97, Cicahem, Kiara Condong	14	2,2	1,6	3,1	1,8	2,3	2,8	2,2	1,1	2,9	1,3	3,4	0,45	1,1	1,8	

Nama alamat atau gerai yang ada dalam data ini kemudian selanjutnya digunakan dalam kode angka untuk komputasi selanjutnya. Gudang diberikan kode node “0”, dimana hasil komputasi *Traveling Salesman Problem* pada penelitian ini,, node 0 (sumber) selalu menjadi node asal dan akhir.

1. Komputasi rute terpendek TSP dengan menggunakan EA

Tabel II menunjukkan contoh hasil komputasi menggunakan Program Solver untuk distribusi dari gudang ke 7 gerai dengan hasil sebagai berikut:

Hasil komputasi menggunakan EA :

- Total jarak : 10,65 km
- Jalur : 0-1-6-5-2-4-7-3

TABEL II
HASIL KOMPUTASI EA UNTUK PENGIRIMAN KE 7 BUAH GERAI

Node	0	1	2	3	4	5	6	7
0	0	1,3	3,2	2,2	2	2,6	1,8	2,3
1	1,3	0	2,4	1,6	1,4	1,4	1	1,7
2	3,2	2,4	0	3,1	0,65	1,7	1,5	2,4
3	2,2	1,6	3,1	0	1,6	2,2	1,4	1,1
4	2	1,4	0,65	1,6	0	1,9	0,8	1,6
5	2,6	1,4	1,7	2,2	1,9	0	1,1	2,1
6	1,8	1	1,5	1,4	0,8	1,1	0	1,5
7	2,3	1,7	2,4	1,1	1,6	2,1	1,5	0

EA	0	1	6	5	2	4	7	3	0
	1,3	1	1,1	1,7	0,65	1,6	1,1	2,2	

Total jarak	10,65
-------------	-------

Dengan cara yang sama diperoleh data total jarak dan lintasan optimum variasi jumlah node. Tabel III menunjukkan hasil rekapitulasi hasil komputasi menggunakan EA berbagai jumlah node dan total jarak yang dihasilkannya.

TABEL III
TOTAL JARAK (KM) DAN LINTASN OPTIMUM HASIL KOMPUTASI MENGGUNAKAN *EVOLUTIONARY ALGORITHM*

Banyaknya Node	Total Jarak (Km)	Lintasan Optimum
5	8,15	0 3 4 2 1 0
6	8,85	0 1 5 2 4 3 0
7	9,25	0 1 2 4 5 3 6 0
8	10,65	0 1 6 5 2 4 7 3 0
9	10,70	0 1 5 8 2 4 6 3 7 0
10	10,45	0 1 6 4 2 5 8 9 3 7 0
11	11,75	0 1 6 4 10 2 5 8 9 3 7 0
12	12,95	0 11 7 3 9 8 5 2 # 4 6 1 0
13	12,97	0 1 6 4 10 2 5 8 9 3 # 7 11 0
14	13,82	0 13 1 6 4 10 2 5 8 9 3 12 7 11 0
15	14,12	0 6 4 10 2 8 5 1 # 3 9 12 7 14 11 0

2. Komputasi rute terpendek TSP dengan menggunakan metode LP

Komputasi rute terpendek TSP menggunakan program Solver dilakukan menggunakan prosedur pada umumnya saat melakukan komputasi menggunakan metode Integer Linear Programming. Pada komputasi ini, hasil komputasi dari Solver tidak langsung menghasilkan 1 lintasan atau terjadi *subtour*. Prosedur manual dilakukan dengan menambahkan pembatas integer baru yang berisikan variabel-variabel yang terlibat pada jaringan *subtour*.

Tabel IV merupakan contoh tampilan luaran komputasi Solver di Microsoft Excel menggunakan metode simplex LP.

TABEL IV
LUARAN HASIL KOMPUTASI SOLVER PADA MICROSOFT EXCEL DENGAN METODE LP

Node	0	1	2	3	4	5	6	7
0	1000	1,3	3,2	2,2	2	2,6	1,8	2,3
1	1,3	1000	2,4	1,6	1,4	1,4	1	1,7
2	3,2	2,4	1000	3,1	0,65	1,7	1,5	2,4
3	2,2	1,6	3,1	1000	1,6	2,2	1,4	1,1
4	2	1,4	0,65	1,6	1000	1,9	0,8	1,6
5	2,6	1,4	1,7	2,2	1,9	1000	1,1	2,1
6	1,8	1	1,5	1,4	0,8	1,1	1000	1,5
7	2,3	1,7	2,4	1,1	1,6	2,1	1,5	1000

z	10,85
---	-------

Node	0	1	2	3	4	5	6	7	Subject to
0	0	0	0	1	0	0	0	0	1
1	1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	0	1
3	0	0	0	0	0	0	0	1	1
4	0	0	1	0	0	0	0	0	1
5	0	1	0	0	0	0	0	0	1
6	0	0	0	0	0	1	0	0	1
7	0	0	0	0	1	0	0	0	1
	1	1	1	1	1	1	1	1	1

Optimum Path 0 3 7 4 2 6 5 1 0

Angka “1” dengan cetak tebal pada kolom pembatas (*subject to*) merupakan konstanta hasil dari pembatas integer tambahan dari variabel-variabel *subtour* yang terjadi pada komputasi sebelumnya. Dengan cara yang sama diperoleh data total jarak dan lintasan optimum dari variasi jumlah node. Tabel V adalah hasil rekapitulasi lintasan optimum dan jarak terdekat menggunakan *Linear Programming*.

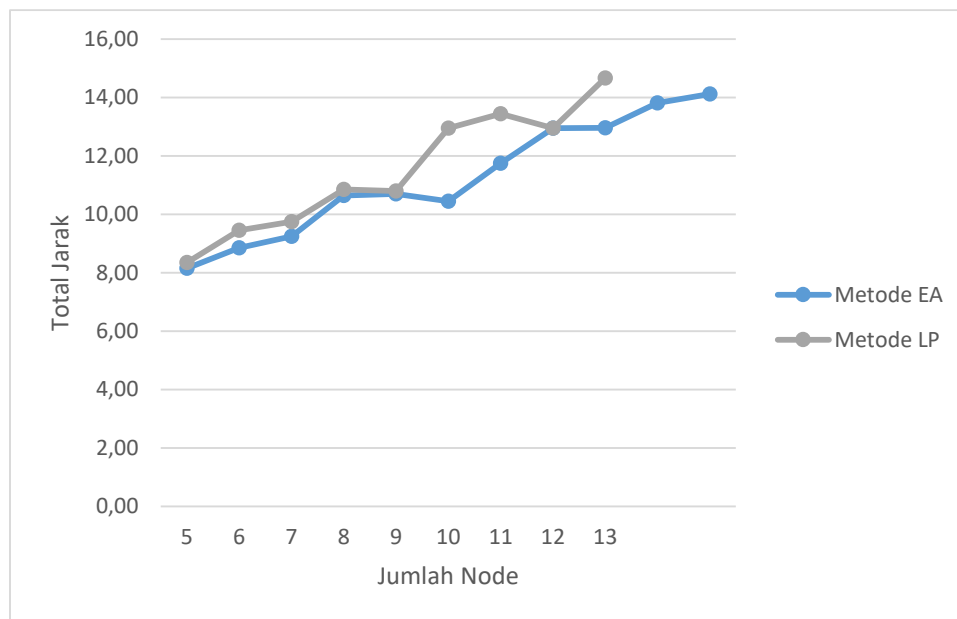
TABEL IV
TOTAL JARAK (KM) DAN LINTASAN OPTIMUM HASIL KOMPUTASI SOLVER MENGGUNAKAN METODE LP

Banyaknya Node	Total jarak (Km)	Lintasan Optimum
5	8,35	0 1 3 4 2 0
6	9,45	0 1 4 2 5 5 0
7	9,75	0 1 5 6 2 4 3 0
8	10,85	0 3 7 4 2 6 5 1 0
9	10,8	0 1 6 8 5 2 4 3 7 0
10	12,95	0 6 4 1 5 2 8 9 3 7 0
11	13,45	0 4 6 7 3 9 10 2 8 5 1 0
12	12,95	0 1 6 4 10 2 5 8 9 3 7 11 0
13	14,67	0 11 7 12 9 3 10 2 5 8 4 6 1 0
14	infeasible	
15	infeasible	

Secara umum dengan bertambahnya node, juga akan memberikan tambahan pada total jarak. Namun terjadi pengecualian pada jumlah node 9 buah (10,8 Km) dan jumlah node 8 buah (10,85 Km) serta jumlah node 12 buah (12,95 Km) dan jumlah node 11 buah (13,45 Km), yaitu total jarak lebih pendek pada node yang lebih banyak.

Ditemukan juga pada jumlah node 14 buah dan 15 buah, prosedur simpleks LP tidak menemukan solusi yang layak (*infeasible solution*).

Evaluasi terhadap EA ini dilakukan secara kualitatif dengan melihat pola-pola yang terjadi dan dibandingkan dengan metode LP. Gambar 2 perbandingan kedua (EA dan LP) dalam pengaruh total jarak optimal terhadap banyaknya node.



Gambar 2 Perbandingan total jarak optimal TSP yang dihasilkan oleh EA dan LP

Grafik pada gambar 2 menunjukkan bahwa metode heuristik EA mempunyai kinerja yang lebih baik dalam menghasilkan jarak yang terpendek secara umum. Berdasarkan pola total jarak yang dihasilkan EA menggambarkan bahwa metode heuristik lebih mendekati kondisi optimal yang dihasilkan oleh LP.

Dalam kasus ini telah ditunjukkan suatu kualitas konvergensi metode heuristik EA dengan perbedaan nilai solusi yang relatif rendah terhadap metode deterministik seperti LP. Di sisi lain, tabel 8 menunjukkan bahwa metode LP pada jumlah node yang lebih banyak (14 buah) menyebabkan solusi *infeasible*. Sehingga metode heuristik EA dalam kasus ini berperan juga dalam menjadi metode substitusi metode deterministik tersebut. Ketidaklayakan solusi pada metode deterministic LP pada kasus ini terjadi karena iterasi yang menghasilkan banyak *subtour* dan tidak bisa digabungkan secara simultan.

Linearitas antara biaya dan jumlah busur yang dihasilkan suatu metode dapat dilihat pada tabel vi

TABEL VI
PERBANDINGAN MODEL LINEAR DAN POLYNOMIAL

Metode	Model Linear
<i>Evolutionary Algorithm</i>	$y = 0,6049x + 7,6124$ $R^2 = 0,9673$
Linear Programming	$= 0,7547x + 7,6956$ $R^2 = 0,9453$

(data diolah menggunakan Microsoft Excel)

Secara signifikan kedua metode menghasilkan indikasi sebuah persamaan linear. Hal ini terlihat dari nilai R^2 masing-masing di persamaan linear yang mendekati nilai 1.

IV. KESIMPULAN

Dari hasil penelitian ini diperoleh beberapa kesimpulan berikut:

1. Pada kasus ini metode Heuristik *Evolutionary Algorithm* mempunyai konvergensi yang baik dalam memecahkan permasalahan TSP
2. Sifat polinomial dalam kasus ini tidak signifikan muncul dalam model TSP, sehingga tidak diperlukan metode optimisasi non linear dalam memecahkannya.
3. Metode heuristik *Evolutionary Algorithm* merupakan pendekatan yang memiliki tingkat hambatan komputasi yang relatif rendah dibandingkan metode deterministik Linear Programming.

REFERENSI

- [1] Boussaid, I., Lepagnot, J., Siarry, R., *A Survey an Optimization Metaheuristic*, Information Sciences 271(233), Elsevier. 2013 hal 82-117
- [2] Dantzig, G., Fulkerson., R. dan Johnson, S., *Solution of Large Scale Traveling Salesman Problem*, *Operation Research*. Vol 2, 1954. Hal 393-410
- [3] Diaby, M. *The Traveling Salesman Problem: A Linear Programming Formulation*, WSEAS Transaction On Mathematics, Issue 6, Vol 6. USA, 2007
- [4] Chandra, A. dan Setiawan, B., *Minimasi Jalur Distribusi di PT.XYZ dengan Metode Improved Cluster First Route Second*, Jurnal Metris (20). 2019
- [5] Chandrakantha, L. *Using Excel Solver in Optimization Problems*. White Paper. CUNY: Mathematics and Computer Science Department. 2011.
- [6] Eiben, A.E., dan Smith, J.E., *Introduction to Evolutionary Computing*, Springer, Heidelberg. 2015
- [7] Gentilini, I., Margot, F., Shimada, K., *The Travelling Salesman Problem with Neighborhood MINLP Solution*, Journal of Optimization Methods and Software, Vol. 8, Issue No. 2, 2012
- [8] Jadzack R., *Traveling Salesman Problem : Approach to Optimality*, De Gruyter Open, Vol. 15, Issu. 2, hal 157-169, University of Social Sciences Publishing House. 2014
- [9] Jiang, C. A., *Reliable Solver of Euclidean Traveling Salesman Problems in Microsoft Excel Add-in Tools for Small-size System.*, Journal of Software, Vol 5. No 7, 2010
- [10] Taha, H., *Operations Research*, 8th Edition, Pearson Education, Inc New Jersey. 2007