

PENERAPAN *FIREFLY ALGORITHM* PADA PROSES PENENTUAN RUTE DAN PEMBERANGKATAN KENDARAAN DI PT PERTAMINA TBBM SURABAYA GROUP

Edo Rizaldi, Abdullah Alkaff, dan Mochammad Sahal

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: edo.rizaldi12@mhs.ee.its.ac.id, alkaff@ee.its.ac.id, sahal@ee.its.ac.id

Abstrak—Proses *vehicle routing and dispatching* sangat menentukan ketersediaan Bahan Bakar Minyak (BBM) di tiap-tiap Stasiun Pengisian Bahan Bakar Umum (SPBU). Proses *vehicle routing* akan membuat rute distribusi tiap SPBU dengan mempertimbangkan permintaan (*demand*) BBM tiap SPBU, jarak tempuh dari depo ke SPBU, serta jarak tempuh antar SPBU. Rute-rute yang terbentuk akan digunakan sebagai data masukan untuk proses *vehicle dispatching*, yaitu proses pengalokasian kendaraan per nomor kendaraan untuk menjalankan tiap rute yang terbentuk. Kendaraan yang digunakan dalam proses distribusi terbagi dalam kelompok (*cluster*) sesuai dengan kapasitas maksimalnya yang menandakan permasalahan ini termasuk *capacitated vehicle routing problem*. Pada penelitian Tugas Akhir ini, digunakan *firefly algorithm* untuk menyelesaikan permasalahan *vehicle routing and dispatching* dengan mempertimbangkan deviasi jarak tempuh tiap kendaraan yang terpakai. Dari hasil perbandingan menggunakan metode *nearest neighbor*, didapatkan hasil bahwa *firefly algorithm* mampu menghasilkan jumlah rute dan nilai deviasi jarak tempuh yang lebih baik, yaitu 24 rute dengan deviasi jarak tempuh sebesar 1.677 jam.

Kata Kunci—*Capacitated Vehicle Routing Problem, Firefly Algorithm, Proses Distribusi, Vehicle Dispatching*

I. PENDAHULUAN

Distribusi merupakan domain penting dalam kehidupan kita sehari-hari. Hal itu dikarenakan distribusi sangat berpengaruh pada aspek sosial maupun ekonomi. Lebih jauh lagi, distribusi juga memegang peranan penting pada ketersediaan bahan baku. Penurunan kecil dalam jarak tempuh operasi logistik secara harian dapat berdampak secara langsung pada pengurangan biaya distribusi.

Kebutuhan akan pasokan Bahan Bakar Minyak (BBM) pada tiap-tiap Stasiun Pengisian Bahan Bakar Umum (SPBU) merupakan bagian dari permasalahan distribusi. PT Pertamina sebagai perusahaan utama dalam negeri yang memasok BBM dari Terminal BBM (TBBM) ke tiap-tiap SPBU, menyadari hal tersebut sebagai suatu permasalahan utama. Beberapa komponen yang sangat mempengaruhi kecepatan pemenuhan BBM ke tiap SPBU adalah ketersediaan mobil tangki, rute

pendistribusian dan penjadwalan, yang dalam hal ini diartikan sebagai *Vehicle Routing*. Jika sistem *Vehicle Routing* itu tidak tertata dengan baik, maka sistem distribusi BBM tidak akan berjalan dengan baik dan mengakibatkan harga satuan BBM di masyarakat akan naik apabila terjadi kelangkaan BBM. Lebih jauh lagi, kejadian tersebut mengakibatkan turunnya citra Pertamina di mata masyarakat sehingga dikhawatirkan masyarakat bisa berpindah membeli BBM ke perusahaan asing lainnya yang mampu memberikan pelayanan yang lebih baik. Berangkat dari kasus tersebut, dapat diangkat suatu usulan penelitian mengenai penentuan rute dan penjadwalan kendaraan yang dapat mengoptimalkan kinerja dari proses distribusi BBM.

Fokus permasalahan yang dijadikan sumber pengerjaan tugas akhir ini adalah bagaimana menyelesaikan permasalahan penentuan rute dan penjadwalan pengiriman mobil tangki per nomor kendaraan dari TBBM (dalam hal ini bisa dikatakan depot) ke tiap SPBU dengan mempertimbangkan deviasi jarak tempuh tiap-tiap kendaraan.

Pembuatan sistem tersebut juga mempertimbangkan tipe kendaraan distribusi yang ada, tipe permintaan (*demand*) tiap-tiap SPBU, serta sistem giliran (*shift*) yang ada pada pengiriman BBM dari depo ke SPBU.

II. DASAR TEORI

A. Bin Packing Problem [1], [7]

Bin Packing Problem (BPP) adalah teori dasar yang melandasi konsep *vehicle dispatching* yang ada pada tugas akhir ini. BPP adalah suatu kasus dimana beberapa obyek yang punya volume masing-masing, harus dimasukkan kedalam suatu *bin* atau kotak yang punya keterbatasan kapasitas. Fungsi objektif dari permasalahan BPP adalah untuk meminimumkan jumlah penggunaan dari kotak tersebut. BPP sendiri adalah kasus dengan karakteristik mempunyai 1 distributor, dengan kapasitas yang homogen, dan nilai dari matriks biaya $c_{ij} = 0$ untuk setiap rute dari pelanggan i ke pelanggan j . Dalam kasus BPP, diberikan sejumlah n barang dan m kotak dengan:

$$w_j = \text{Berat dari barang } j$$

$$C = \text{Kapasitas dari tiap kotak}$$

Masukkan setiap barang kedalam 1 kotak sehingga berat total dari semua barang yang masuk kotak tersebut tidak

melebihi dari nilai C tiap kotak. Selain itu, permasalahan BPP selalu difungsikan untuk meminimumkan jumlah penggunaan kotak. Model matematis permasalahan ini adalah sebagai berikut:

$$\text{Min } Z = \sum_{i=1}^n y_i \quad (2.1)$$

B. Capacitated Vehicle Routing Problem [5], [8]

Kasus CVRP adalah suatu permasalahan gabungan dari kasus *Vehicle Routing Problem* dengan *Bin Packing Problem (BPP)*.

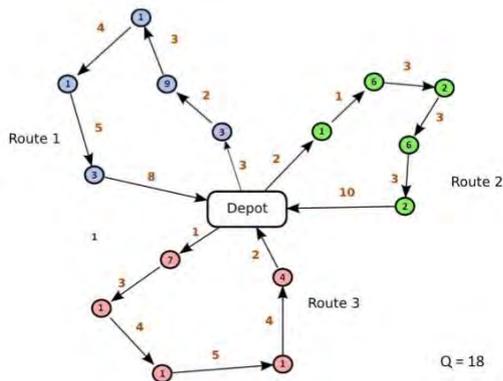
CVRP adalah varian yang paling umum dan dasar dari kasus *vehicle routing problem*. Dalam varian ini, kendaraan berkapasitas tetap m ($m \geq 1$) harus melayani *demand* pelanggan, dengan tambahan batasan bahwa tiap kendaraan ini punya kapasitas tertentu. Kendaraan ini dapat memuat suatu barang (*demand* dari pelanggan) sampai dengan kapasitas maksimalnya.

Heterogeneous CVRP mempunyai beberapa tipe kendaraan yang tiap tipenya memiliki kapasitas tertentu. Sifat dari CVRP *Heterogeneous Fleet* inilah yang mendasari terciptanya konsep *vehicle routing and dispatching*. Model permasalahan CVRP *Heterogeneous Fleet* bisa dituliskan sebagai berikut:

$$\left(\sum_{j=1}^K d_j \right) \leq Q \quad (2.2)$$

Dengan batasan:

$$\begin{aligned} d_j &= \text{demand dari pelanggan } v_j \\ d_j &\leq Q, \quad (1 \leq j \leq n) \\ \left(\sum_{j=1}^n d_j \right) &\leq m * Q \end{aligned} \quad (2.3)$$



Gambar 2.1 Representasi *Capacitated Vehicle Routing Problem*

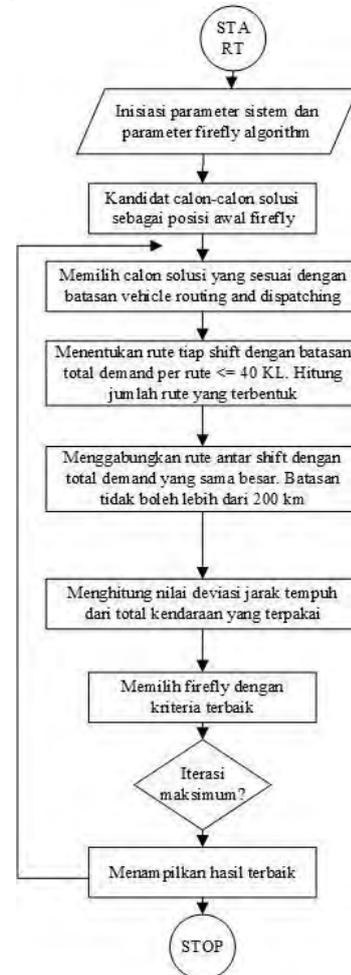
C. Firefly Algorithm [12], [13], [14]

Firefly Algorithm dibentuk dan dikembangkan oleh Dr. Xin She Yang di Cambridge University pada tahun 2007 untuk memecahkan masalah optimasi. *Firefly Algorithm* merupakan

algoritma yang didasarkan pada perilaku kunang-kunang. Kunang-kunang pada umumnya menghasilkan sinar dalam durasi yang pendek dan memiliki ritme tertentu. Ritme dari sinar dan durasi waktu penyinaran dari kunang-kunang merupakan bagian sinyal dari sistem kunang-kunang yang dibawa oleh kunang-kunang baik yang jantan maupun betina. Terdapat dua fungsi dasar sinar kunang-kunang, yaitu untuk menarik perhatian kunang-kunang yang lain atau bertahan dari serangan pemangsa.

Beberapa aturan yang diadopsi dan disintesis untuk membentuk *Firefly Algorithm* adalah:

1. Semua *firefly* bersifat *unisex*, dimana *firefly* punya ketertarikan satu dengan yang lain tanpa menghiraukan jenis kelamin.
2. Daya pikat dari *firefly* bersifat proporsional, bergantung pada tingkat terang dari sinar yang dipancarkan. Daya pikat *firefly* semakin berkurang pada saat jarak semakin bertambah. Jika diantara kunang-kunang tidak ada yang bersinar lebih terang, maka kunang-kunang bergerak secara *random*.
3. Terang yang ditimbulkan kunang-kunang dipengaruhi atau ditentukan oleh susunan dari fungsi objektif atau fungsi tujuan.



Gambar 2.2 Diagram alir algoritma kunang-kunang yang diterapkan pada *vehicle routing and dispatching*

III. PERANCANGAN SISTEM

A. Model Matematis Vehicle Routing and Dispatching

Model matematis digunakan untuk menginterpretasikan permasalahan dalam bentuk yang lebih detail. Model matematis ini menjelaskan mengenai fungsi objektif, parameter, variabel, dan *constraint* yang ada.

$$MinZ = \left(\sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^N d_{ij} X_{ijk} - Y \right)^2 \quad (3.1)$$

Dimana:

- K : Jumlah total kendaraan
- N : Jumlah Total SPBU
- d_{ij} : Jarak tempuh dari SPBU i ke SPBU j
- m_i : Demand SPBU i
- q_k : Kapasitas kendaraan k
- Y : Waktu tempuh maksimal kendaraan

$$X_{ijk} = \begin{cases} 1 & \text{kendaraan } k \text{ mengunjungi SPBU } i \text{ dan SPBU } j \\ 0 & \text{kendaraan } k \text{ hanya mengunjungi salah satu SPBU} \end{cases}$$

Subject to:

$$\sum_{k=1}^K \sum_{j=1}^N x_{ijk} \leq K, \quad \text{for } i = 0 \quad (3.2)$$

$$\sum_{j=1}^N x_{ijk} = 1, \quad \text{for } i = 0 \text{ and } k \in \{1, \dots, K\} \quad (3.3)$$

$$\sum_{j=1}^N x_{jik} = 1, \quad \text{for } i = 0 \text{ and } k \in \{1, \dots, K\} \quad (3.4)$$

$$\sum_{k=1}^K \sum_{j=0}^N x_{ijk} = 1, \quad \text{for } i \in \{1, \dots, N\} \quad (3.5)$$

$$\sum_{k=1}^K \sum_{i=0}^N x_{ijk} = 1, \quad \text{for } j \in \{1, \dots, N\} \quad (3.6)$$

$$\sum_{i=1}^N m_i \sum_{j=0}^N x_{ijk} = q_k, \quad \text{for } k \in \{1, \dots, K\} \quad (3.7)$$

Fungsi Objektif di persamaan (3.1) adalah untuk meminimumkan deviasi jarak tempuh tiap kendaraan yang terpakai. Deviasi yang dimaksud adalah perbedaan waktu tempuh dari waktu tempuh yang seharusnya dipenuhi oleh tiap kendaraan, yaitu 8 jam. Disamping harus memenuhi fungsi objektif, sistem ini juga harus memenuhi semua batasan yang ada.

Pada batasan (3.2) menjelaskan bahwa untuk setiap k kendaraan, jumlah *demand* pada SPBU yang disuplai tidak

boleh melebihi kapasitas angkut kendaraan. Batasan (3.3) menjelaskan untuk setiap SPBU i yang dilayani, kendaraan hanya bisa mengunjungi tepat satu kali. Batasan (3.4) ini menjelaskan untuk setiap k kendaraan yang mengunjungi j SPBU, kendaraan tersebut harus memenuhi *demand* dari SPBU i . Batasan (3.5) menjelaskan kebalikan dari batasan (3.4), yaitu setiap k kendaraan yang mengunjungi i SPBU, kendaraan tersebut harus memenuhi *demand* dari SPBU j . Pada batasan (3.6) dijelaskan setiap rute yang terbentuk selalu berawal dari depo. Sedangkan pada batasan (3.7) dijelaskan bahwa setiap rute yang terbentuk berakhir di depo.

B. Firefly Algorithm untuk menyelesaikan proses Vehicle routing and dispatching

Firefly algorithm digunakan untuk mencari nilai objektif yang paling baik dari sejumlah iterasi yang telah dibuat. Dalam sistem ini dicari nilai deviasi jarak tempuh paling minimum dari keseluruhan iterasi. Parameter *Firefly* yang digunakan dijelaskan pada tabel.

Tabel 3.1 Parameter *Firefly*

Parameter <i>Firefly</i>	Nilai
MaxGeneration (iterasi)	50
Populasi <i>Firefly</i> (n)	200
Gamma (γ)	1000

C. Pengelompokan SPBU

Proses diawali dengan menentukan secara random urutan pengiriman dari 95 SPBU yang ada. Setelah didapat urutan random dari ke-95 SPBU, proses selanjutnya adalah menentukan di *shift* mana SPBU tersebut dilayani. Diasumsikan, untuk SPBU yang mendapat urutan 1-29, maka SPBU tersebut dilayani di *shift* 1. SPBU yang mendapat urutan 30-57, maka SPBU tersebut dilayani di *shift* 2. Dan untuk SPBU yang mendapat urutan 58-85 dilayani di *shift* 3. SPBU yang mendapat urutan 86 sampai 95 diasumsikan tidak dilayani. Rute pertama yang terbentuk dalam setiap *shift* adalah rute dari depo ke SPBU yang mendapat urutan random nomor 1. SPBU urutan 30 dan 58 di *shift* 2 dan 3 diasumsikan mendapat urutan random nomor 1 di *shift* tersebut.

D. Pembuatan Rute Tiap Shift

Setelah dilakukan proses pengelompokan SPBU, proses selanjutnya adalah proses penambahan SPBU kedalam suatu rute. Proses ini dilakukan dengan menambahkan SPBU yang masih *available* atau tersedia kedalam suatu rute sesuai dengan urutan yang dibentuk dari urutan *random*. Proses ini juga mempertimbangkan total *demand* yang terbentuk dari setiap rute. Apabila penambahan SPBU baru kedalam suatu rute mengakibatkan *total demand* melebihi kapasitas angkut kendaraan, maka SPBU baru tersebut tidak bisa ditambahkan ke dalam rute tersebut. Ketika SPBU tersebut tidak bisa ditambahkan pada suatu rute, maka SPBU tersebut menginisiasi terbentuknya rute baru.

E. Penggabungan Rute Antar Shift

Prosedur untuk melakukan proses penggabungan rute adalah sebagai berikut:

1. Data masukan berupa rute distribusi tiap *shift*, total *demand* tiap rute, dan jarak tempuh tiap rute
2. Lakukan pengelompokan untuk tiap rute yang memiliki kesesuaian *demand*
3. Lakukan penggabungan rute untuk tiap rute yang bersesuaian total *demand* antar *shift* yang berbeda
4. Apabila penggabungan rute menghasilkan jarak tempuh melebihi 200 km, maka penggabungan rute tersebut tidak bisa dilakukan.
5. Mencari rute lain dalam kelompok rute yang sama di *shift* yang berbeda
6. Proses penggabungan rute selesai ketika semua rute yang ada telah diproses

F. Pemilihan Kendaraan

Langkah-langkah dalam proses pemilihan kendaraan adalah sebagai berikut:

1. Data masukan berupa hasil rute gabungan yang didapat di proses penggabungan rute
2. Total *demand* tiap rute menentukan tipe kendaraan yang digunakan untuk menjalankan rute tersebut
3. Kendaraan yang ditugaskan untuk setiap tipe selalu diawali dari kendaraan nomor 1
4. Ketersediaan kendaraan bernilai fix atau konstan
5. Apabila dalam pemilihan kendaraan didapat suatu kasus depo kehabisan stok kendaraan di tipe tertentu, maka rute yang mendapat jatah kendaraan tersebut tidak bisa dilayani

IV. IMPLEMENTASI SISTEM

A. Pengumpulan Data

Dari proses pengumpulan data, didapatkan suatu hasil:

1. Sistem menggunakan seluruh SPBU yang berada di wilayah Surabaya dengan jumlah 95 SPBU. Dalam pengerjaan program, 10 SPBU diasumsikan tidak *supply*.
2. Terdapat 153 kendaraan dengan rincian:
 - a. Tipe 16 : 17 Unit
 - b. Tipe 24 : 66 Unit
 - c. Tipe 32 : 56 Unit
 - d. Tipe 40 : 14 Unit
3. *Shift* pengiriman digunakan untuk mengelompokkan SPBU kedalam slot waktu yang tersedia. Karakteristik *shift* pengiriman dapat dilihat pada tabel 4.1.

Tabel 4.1 Karakteristik *Shift* Pengiriman

<i>Shift</i>	Slot Waktu
<i>Shift</i> 1	24.01 – 08.00
<i>Shift</i> 2	08.01 – 16.00
<i>Shift</i> 3	16.01 – 24.00

4. Jarak Tempuh SPBU

Jarak tempuh antar SPBU diperoleh dengan asumsi penulis. Sedangkan jarak Depo dengan SPBU diperoleh dari pihak Pertamina TBBM Surabaya Group

5. *Demand* tiap SPBU

Demand BBM tiap SPBU ditentukan secara random di nilai 16 KL, 24 KL, dan 32 KL.

B. Hasil Pembuatan Rute

Pada proses pembuatan rute, pada *shift* 1 terbentuk 21 rute. Sedangkan *shift* 2 dan *shift* 3 masing-masing terbentuk 22 dan 21 rute. Tabel 4.2 adalah tabel rute *shift* 1.

Tabel 4.2 Tabel Hasil Pembuatan Rute *Shift* 1

RUTE	SPBU	JARAK TEMPUH (km)	TOTAL DEMAND	CAR
1	DEPO – 30 - DEPO	42.888	3	24(1)
2	DEPO – 51 – DEPO	70.332	4	32(1)
3	DEPO – 4 – 92 – DEPO	65.034	4	32(2)
4	DEPO – 13 – 90 – DEPO	57.857	5	40(1)
5	DEPO – 23 – DEPO	30.06	2	16(1)
6	DEPO – 75 – DEPO	52.062	4	32(3)
7	DEPO – 42 – DEPO	75.59	4	32(4)
8	DEPO – 16 – 32 – DEPO	84.973	4	32(5)
9	DEPO – 94 – 76 – DEPO	70.461	5	40(2)
10	DEPO – 65 – 22 – DEPO	63.366	5	40(3)
11	DEPO – 95 – 83 – DEPO	80.789	5	40(4)
12	DEPO – 85 – DEPO	73.712	3	24(2)
13	DEPO – 50 – DEPO	60.634	4	32(6)
14	DEPO – 64 – 54 – DEPO	85.835	5	40(5)
15	DEPO – 58 – DEPO	66.262	4	32(7)
16	DEPO – 56 – DEPO	75.786	4	32(8)
17	DEPO – 80 – 15 – DEPO	81.937	5	40(6)
18	DEPO – 37 – DEPO	40.24	4	32(9)
19	DEPO – 86 – DEPO	74.366	3	24(3)
20	DEPO – 35 – DEPO	41.914	4	32(10)
21	DEPO – 73 – DEPO	58.31	3	24(4)

C. Hasil Penggabungan Rute

Pada proses penggabungan rute *shift* 2 dan *shift* 3, terbentuk 24 rute yang memenuhi total 85 permintaan. Hasil proses ini dapat dilihat pada tabel 4.3.

Tabel 4.3 Tabel Hasil Penggabungan Rute

DEMAND	SHIFT 1	SHIFT 2	SHIFT 3
16 (1)	DEPO – 23 – DEPO	DEPO – 43 – DEPO	DEPO – 81 – DEPO
24 (1)	DEPO – 30 – DEPO	DEPO – 88 – DEPO	DEPO – 53 – DEPO
24 (2)	DEPO – 85 – DEPO	DEPO – 55 – DEPO	DEPO – 36 – DEPO
24 (3)	DEPO – 86 – DEPO	DEPO – 33 – DEPO	DEPO – 69 – DEPO
24 (4)	DEPO – 73 – DEPO	DEPO – 18 – DEPO	DEPO – 91 – DEPO
24 (5)	-	DEPO – 46 – DEPO	DEPO – 78 – DEPO
24 (6)	-	DEPO – 48 – DEPO	DEPO – 52 – DEPO
24 (7)	-	DEPO – 47 – DEPO	DEPO – 21 – DEPO
32 (1)	DEPO – 51 – DEPO	DEPO – 41 – DEPO	DEPO – 72 – DEPO
32 (2)	DEPO – 4 – 92 – DEPO	DEPO – 28 – DEPO	DEPO – 82 – 49 – DEPO
32 (3)	DEPO – 75 – DEPO	DEPO – 26 – DEPO	DEPO – 1 – 25 – DEPO
32 (4)	DEPO – 42 – DEPO	DEPO – 66 – 20 – DEPO	-
32 (5)	DEPO – 16 – 32 – DEPO	DEPO – 39 – DEPO	DEPO – 14 – DEPO
32 (6)	DEPO – 50 – DEPO	DEPO – 12 – DEPO	DEPO – 70 – DEPO
32 (7)	DEPO – 58 – DEPO	DEPO – 93 – DEPO	DEPO – 2 – DEPO
32 (8)	DEPO – 56 – DEPO	DEPO – 27 – DEPO	DEPO – 3 – DEPO
32 (9)	DEPO – 37 – DEPO	DEPO – 11 – DEPO	DEPO – 31 – DEPO
32 (10)	DEPO – 35 – DEPO	-	DEPO – 38 – 34 – DEPO
40 (1)	DEPO – 13 – 90 – DEPO	DEPO – 40 – 84 – DEPO	DEPO – 29 – 63 – DEPO
40 (2)	DEPO – 94 – 76 – DEPO	DEPO – 68 – 19 – DEPO	DEPO – 71 – 59 – DEPO

DEMAND	SHIFT 1	SHIFT 2	SHIFT 3
40 (3)	DEPO – 65 – 22 – DEPO	DEPO – 45 – 77 – DEPO	DEPO – 9 – 17 – DEPO
40 (4)	DEPO – 95 – 83 – DEPO	DEPO – 60 – 57 – DEPO	DEPO – 67 – 89 – DEPO
40 (5)	DEPO – 64 – 54 – DEPO	DEPO – 7 – 79 – DEPO	-
40 (6)	DEPO – 80 – 15 – DEPO	-	-

D. Hasil Pemilihan Kendaraan

Pada proses pemilihan kendaraan, data hasil proses penggabungan rute dicocokkan dengan *database* ketersediaan kendaraan. Hasil proses ini dapat dilihat pada tabel 4.4.

Tabel 4.4 Tabel Hasil Pemilihan Kendaraan

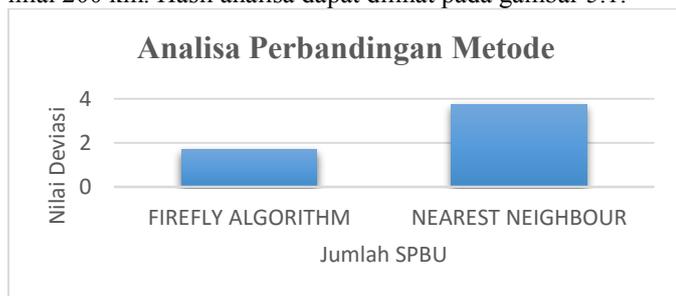
TIPE	NOMOR	TOTAL JARAK TEMPUH (Km)	DEVIASI (Jam)
16	L9086UH	160.720	1.677
	L9126UZ	156.752	
24	L8565UZ	183.146	
	L8970UZ	198.614	
	L8410UU	173.542	
	AG8734R	125.28	
	M8516UH	113.902	
	M8591UH	105.098	
32	AG9044US	154.876	
	AG9748US	162.338	
	N8164UA	197.159	
	N8442UC	159.101	
	AG8956US	205.109	
	L9434UB	205.88	
	L9442UB	180.104	
	L9508UB	186.458	
	L9527UB	149.682	
	L9528UB	122.929	
40	L9501UB	197.4241	
	L9585UG	151.0941	
	N8540UB	174.3124	
	N9098UF	190.5243	
	N8024UE	153.56	
	N8467UE	161.4365	

V. PENGUJIAN DAN HASIL ANALISIS PENGUJIAN

Proses pengujian yang dilakukan adalah dengan membandingkan hasil deviasi jarak tempuh yang dihasilkan oleh *firefly algorithm* dengan hasil deviasi jarak tempuh *nearest neighbor*. Tujuan dari analisa sistem ini adalah untuk menguji tingkat akuratan hasil dari algoritma *firefly*.

A. Perbandingan dengan Nearest Neighbor

Dari hasil analisa didapatkan suatu kesimpulan bahwa *firefly algorithm* menghasilkan nilai deviasi yang jauh lebih baik dari *nearest neighbor*. *Nearest Neighbor* menghasilkan nilai deviasi jarak tempuh sebesar 3.722 jam. Hal ini bisa terjadi dikarenakan metode *nearest neighbor* menggunakan jarak terpendek pada proses penentuan rutenya. Sehingga pada rute-rute awal, jarak tempuh yang didapat memiliki *range* yang pendek dan mengakibatkan jarak tempuhnya sangat jauh dari nilai 200 km. Hasil analisa dapat dilihat pada gambar 5.1.



Gambar 5.1 Hasil Analisa Perbandingan Metode

VI. PENUTUP

A. Kesimpulan

1. *Firefly Algorithm* dapat digunakan sebagai suatu metode penyelesaian permasalahan *vehicle routing and dispatching*.
2. Dalam penyelesaian permasalahan *vehicle routing and dispatching* dengan dataset 85 pelanggan, *firefly algorithm* mampu menghasilkan 24 rute dengan nilai deviasi jarak tempuh kendaraan yang terpakai sebesar 1.677 jam.
3. *Firefly algorithm* mampu menghasilkan nilai deviasi yang lebih baik dari *nearest neighbor* dikarenakan *firefly algorithm* sebagai salah satu varian metaheuristik, menggunakan proses *random* di setiap iterasi dalam proses pencarian nilai terbaik. Sedangkan *nearest neighbor* adalah varian dari metode heuristik yang berangkat dari karakteristik tiap-tiap algoritma dalam pencarian solusi terbaik.

B. Saran

1. Perlu adanya pemahaman lebih dalam terkait penggunaan metode metaheuristik pada kasus *vehicle routing problem*

2. Adanya pengembangan metode metaheuristik yang lain pada kasus *vehicle routing problem*, yang dapat menemukan solusi penyelesaian masalah dengan kecepatan pemrosesan data yang lebih baik dari *firefly algorithm*.
3. Kompleksitas permasalahan dapat dikembangkan dengan menambahkan penyimpanan data jarak tempuh yang telah dilalui oleh tiap kendaraan serta karakteristik tipe kendaraan tiap SPBU.

UCAPAN TERIMA KASIH

Penulis Edo Rizaldi mengucapkan terima kasih sebesar-besarnya kepada Bapak Abdullah Alkaff dan Bapak Mochammad Sahal atas bimbingannya selama ini sehingga pengerjaan tugas akhir ini menjadi lancar. Tak lupa pula penulis mengucapkan terima kasih kepada pihak PT Pertamina TBBM Surabaya *Group* atas kesediannya membantu penulis dalam hal penyediaan data-data terkait proses *vehicle routing and dispatching*.

DAFTAR PUSTAKA

- [1] Hillier, F. Lieberman, G. "Introduction to Operation Research", McGraw-Hill, New York, 2001.
- [2] Alkaff, A., Gamayanti, N. "Diktat Kuliah Penyelidikan Operasi", Surabaya.
- [3] Wu, N., R, Coppins. "Linear Programming and Extensions", McGraw-Hill, New York, 1981.
- [4] Carić, Tonči, Gold, Hrvoje. "Vehicle Routing Problem," Faculty of Traffic and Transport Sciences, University of Zagreb, 2008.
- [5] Toth, P., Vigo, D. "The Vehicle Routing Problem", SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia, 2002.
- [6] Immanuel, Randi Mangatas. "Algoritma Particle Swarm Optimization Untuk Menyelesaikan Multi Depot Vehicle Routing Problem Dengan Variabel Travel Time", Institut Teknologi Sepuluh Nopember, Thesis, 2015.
- [7] A. Borodin and R. El-Yaniv. "Online Computation and Competitive Analysis", Cambridge University Press, 1998.
- [8] Ralphs, Ted, "Capacitated Vehicle Routing and Some Related Problems," Industrial and Systems Engineering Lehigh University, 2005.
- [9] Bird, John. "Engineering Mathematics", Elsevier, Oxford, 2007.
- [10] Al-Kharabsheh, Khalid. "Review on Sorting Algorithms: A Comparative Study", International Journal of Computer Science and Security (IJCSS) Volume (7), 2013.
- [11] Anggara Putra, Rian. "Efektifitas Metode Sequential Insertion Dan Metode Nearest Neighbour Dalam Penentuan Rute Kendaraan Pengangkut Sampah Di Kota Yogyakarta", Thesis, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Yogyakarta, 2014.
- [12] Sudirwan, J., Fadlilah, S.N. "Aplikasi Hybrid Firefly Algorithm Untuk Pemecahan Masalah Traveling Salesman: Studi Kasus Pada PT Anugerah Mandiri Success", ComTech Vol. 5 No. 2, 2014.
- [13] Zainal Abidin, A.F., Ansor, M.A. "A Preliminary Study on Firefly Algorithm Approach for Travelling Salesman Problem," Science & Engineering Technology National Conference, 2013.
- [14] Jati, G.K., Suyanto, S. "Evolutionary discrete firefly algorithm for traveling salesman problem". ICAIS, 2011.
- [15] Wibowo, R.P. "Optimasi Pengiriman BBM Ke SPBU Pada Kasus Multi Depot dan Multi Product Di Sales Area Bandung", Thesis, Fakultas Teknik, Universitas Indonesia, 2011.