

Optimalisasi Pendistribusian Besi Pada UD Logam Guru Dengan Penerapan Algoritma Particle Swarm Optimization

Mulia Agung Permana¹ Yusmet Rizal²

¹²Program Studi Matematika, Universitas Negeri Padang
e-mail: mulia.agung38@gmail.com¹, yusmet_adurrahman@yahoo.com^{1*}

Abstrak

Secara umum distribusi merupakan penyaluran bahan atau komoditas dari satu titik ke titik lainnya dan diterima oleh pihak yang berkepentingan. UD Logam Guru merupakan salah satu jenis usaha dagang yang bergerak pada bidang penjualan dan pendistribusian besi. Besi tersebut akan didistribusikan oleh UD Logam Guru dengan berkeliling ke delapan pengecer yang berada di Kota Padang. Pada penelitian ini kendala yang akan dijawab adalah pemilihan rute distribusi optimal dengan jarak total minimum. Penelitian ini merupakan penelitian terapan dengan menggunakan algoritma *Particle Swarm Optimization* untuk menyelesaikan masalah distribusi besi dengan model *Travelling Salesman Problem*. Jenis data yang digunakan pada penelitian ini yaitu data sekunder yang diperoleh dari UD Logam Guru. Pada perhitungan menggunakan Matlab diperoleh rute optimal yaitu mulai dari Gudang UD Logam Guru, UD Wiya, UD Das, Toko Rajawali, UD Sentosa, UD Pengumpulan, UD Antika, Toko Sumber Baru, UD Wiya, UD Tetes Logam, Kembali lagi ke UD Logam Guru dengan total jarak 24.335 km. Rute ini lebih optimal dari rute yang biasa digunakan UD Logam Guru yaitu dengan jarak tempuh 29.345 km. Rute optimal ini lebih pendek 5.01 km dibandingkan rute yang biasa digunakan oleh UD Logam Guru. Dengan demikian, rute yang direkomendasikan dapat dipertimbangkan oleh UD Logam Guru untuk mengurangi jarak tempuh dan dapat meminimalkan biaya distribusi.

Kata kunci: *Distribusi, Travelling Salesman Problem, Particle Swarm Optimization*

Abstract

In general, distribution refers to the delivery of materials or commodities from one point to another and received by the interested parties. UD Logam Guru is a type of trading business engaged in the sale and distribution of iron. The iron will be distributed by UD Logam Guru by traveling to eight retailers located in the city of Padang. This research addresses the constraint of selecting the optimal distribution route with the minimum total distance. This applied research uses the Particle Swarm Optimization algorithm to solve the iron distribution problem using the Travelling Salesman Problem model. The type of data used in this research is secondary data obtained from UD Logam

Guru. Using calculations with Matlab, the optimal route obtained is as follows: starting from the UD Logam Guru warehouse, UD Wiya, UD Das, Toko Rajawali, UD Sentosa, UD Pengumpulan, UD Antika, Toko Sumber Baru, UD Wiya, UD Tetesan Logam, and returning to UD Logam Guru, with a total distance of 24.335 km. This route is more optimal than the usual route used by UD Logam Guru, which has a distance of 29.345 km. The optimal route is 5.01 km shorter than the usual route used by UD Logam Guru. Therefore, the recommended route can be considered by UD Logam Guru to reduce travel distance and minimize distribution costs.

Keywords: *Distribution, Travelling Salesman Problem, Particle Swarm Optimization*

PENDAHULUAN

Perkembangan layanan pengiriman yang semakin cepat merupakan salah satu permasalahan yang dihadapi oleh hampir semua perusahaan (Fuadiyah, 2018). Hal ini berdampak pada persaingan bisnis yang cukup ketat. Maka dari itu setiap perusahaan harus memiliki strategi yang tepat dalam melakukan distribusi terhadap produknya.

Rute yang dipilih dalam sistem distribusi merupakan faktor paling penting dalam menentukan jarak dan biaya transportasi yang harus dikeluarkan (Fuadiyah, 2018). Pentingnya memilih rute yang tepat dalam mendistribusikan barang dapat membantu mengatasi permasalahan yang ada. Pemilihan rute dengan jarak yang pendek akan mempersingkat waktu pengiriman sehingga dapat mengurangi biaya distribusi dan laba yang diperoleh perusahaan akan meningkat.

UD Logam Guru merupakan salah satu jenis usaha dagang yang bergerak pada bidang penjualan dan pendistribusian besi. Besi tersebut akan didistribusikan oleh UD Logam Guru ke delapan pengecer yang berada di Kota Padang. Proses distribusi produk ini dilakukan satu kali dalam tiap bulannya. Dengan banyaknya titik distribusi tentu akan membutuhkan waktu yang lama karena jarak tempuh yang cukup jauh. Setelah melakukan wawancara pada tanggal 18 Agustus 2023 dengan Ibu Misriani Rahim sebagai pemilik UD Logam Guru. Ternyata UD Logam Guru memiliki masalah dalam mendistribusikan barang. Masalah yang terjadi yaitu keterlambatan pada saat mengantarkan besi untuk para pengecer karena harus berkeliling ke delapan titik dengan jarak yang cukup jauh. Maka peneliti memutuskan bahwa UD Logam Guru tepat dipilih sebagai objek penelitian karena mengalami kendala yang sama dengan apa yang akan diteliti oleh peneliti. Kendala yang akan dijawab adalah pemilihan rute distribusi optimal dengan jarak total minimum.

Permasalahan pemilihan rute termasuk kedalam *Travelling Salesman Problem* (TSP). TSP merupakan salah satu bentuk masalah optimasi yang bertujuan untuk menemukan rute perjalanan mulai dari lokasi awal dan mengunjungi beberapa lokasi yang telah ditentukan, kemudian kembali lagi ke lokasi asal dengan jarak total yang ditempuh minimum dan setiap lokasi dikunjungi tepat satu kali (Natalia dkk, 2019). Permasalahan TSP ini dapat diselesaikan dengan metode algoritma PSO.

PSO merupakan teknik optimasi untuk mencari lintasan optimal berdasarkan perilaku sosial dari binatang, seperti sekawanan burung saat terbang (Immanuel, 2015). Sehingga dalam kasus ini yang akan diteliti yaitu mencari rute optimal dengan jarak terpendek dalam mendistribusikan besi oleh UD Logam Guru.

Berdasarkan pemaparan dari latar belakang di atas, maka peneliti akan mengangkat masalah tersebut ke dalam sebuah skripsi dengan judul "Optimalisasi Pendistribusian Besi pada UD Logam Guru dengan Penerapan Algoritma *Particle Swarm Optimization*". Dari uraian yang dijelaskan pada latar belakang, maka rumusan masalah yang akan dibahas pada penelitian ini yaitu "Bagaimana pemilihan rute yang optimal dalam kasus pendistribusian besi UD Logam Guru dengan penerapan algoritma PSO?"

METODE

Penelitian ini merupakan penelitian terapan. Penelitian terapan merupakan penelitian yang diawali dengan studi kepustakaan berdasarkan teori yang ada dan dilanjutkan dengan pengambilan data dan penerapannya. Data yang digunakan dalam penelitian ini yaitu data sekunder yang didapatkan dari UD Logam Guru. Data yang diperoleh meliputi data nama toko-toko pengecer, data lokasi pengecer, dan rute yang biasa dilalui UD Logam Guru dalam mendistribusikan barang diperoleh menggunakan google maps. Pada penelitian ini akan digunakan metode *Particle Swarm Optimization* untuk mendapatkan rute yang optimal dalam mendistribusikan barang oleh UD Logam Guru.

HASIL DAN PEMBAHASAN

Pada penelitian ini data yang digunakan yaitu titik lokasi pengecer dan jarak tiap pengecer. Data yang diperoleh terdapat Gudang dan delapan pengecer yang tersebar dikota padang dapat dilihat pada Tabel 1.

Tabel 1. Gudang dan Deapan Pengecer UD Logam Guru

No	Tempat	Lokasi
1	Gudang	Simpang GIA Lanud
2	UD Wiya	Balai Baru
3	Toko Rajawali	Kuranji
4	UD Das	Pisang
5	UD Pengumpulan	Pasar Raya
6	UD Antika	Bandar Olo
7	Toko Sumber Baru	Belakang Tangsi
8	UD Tetesan Logam	Kampung Nias
9	UD Sentosa	Veteran

UD Logam Guru dalam mendistribusikan besinya menggunakan rute dari gudang ke UD Wiya sebagai pengecer pertama sampai ke UD Sentosa sebagai

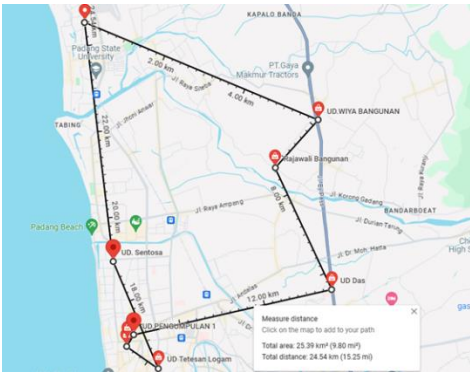
pengecer terakhir lalu kembali lagi ke gudang. Untuk jarak tempuh antar lokasi dapat dilihat pada Tabel 2.

Tabel 2. Jarak Tempuh Antar Lokasi (dalam km)

	Nama	Gudang Logam Guru	UD Wiya	Toko Rajawali	UD Das	UD Pengumpulan	UD Antika	Toko Sumber Baru	UD Tetesan Logam	UD Sentosa
1	Gudang Logam Guru	0	5.84	8.4	5.53	7.38	8.18	7.3	5.56	7.55
2	UD Wiya	5.84	0	3.92	1.48	6.7	6.82	6.53	5.75	6.84
3	Toko Rajawali	8.4	3.92	0	3.09	4.31	5.04	5.36	5.21	5.36
4	UD Das	5.53	1.48	3.09	0	5.09	4.68	4.4	4.9	4.9
5	UD Pengumpulan	7.38	6.7	4.31	5.09	0	0.178	0.318	0.971	1.76
6	UD Antika	8.18	6.82	5.04	4.68	0.178	0	0.172	0.988	1.82
7	Toko Sumber Baru	7.3	6.53	5.36	4.4	0.318	0.172	0	0.895	1.99
8	UD Tetesan Logam	5.56	5.75	5.21	4.9	0.971	0.988	0.895	0	2.69
9	UD Sentosa	7.55	6.84	5.36	4.9	1.76	1.82	1.99	2.69	0

Pada tabel diatas dapat dilihat rute yang digunakan UD Logam Guru dimulai dari UD Wiya kemudian ke UD Das, setelah itu ke Toko Rajawali, lanjut ke UD Antika, kemudian ke, UD Tetes Logam, setelah itu ke UD Pengumpulan, dan lanjut ke UD Sentosa, dan lanjut ke Toko Sumber Baru, dan terakhir kembali ke UD Logam Guru. Dengan melihat rute yang biasa digunakan UD Logam Guru dalam mendistribusikan besi ke seluruh pengecer maka pada penelitian ini akan dilakukan optimasi menggunakan metode PSO untuk mencari rute optimal yang dapat digunakan UD Logam Guru dalam mendistribusikan besi agar lebih efesien. Peta rute awal distribusi UD Logam Guru dapat dilihat pada Gambar 1.

Gambar 1. Peta Rute Awal Distribusi Besi UD Logam Guru



Pembentukan Rute menggunakan Algoritma PSO dengan *Software Matlab*.

Pembentukan rute dengan bantuan software Matlab akan dilakukan sebanyak 50 kali iterasi dengan 50 partikel. Hal ini dilakukan untuk mendapatkan solusi yang lebih optimum yaitu rute distribusi dengan jarak terpendek.

Step 1. Inisialisasi variabel

Inisialisasi variabel dalam implementasi algoritma PSO adalah langkah penting yang harus dilakukan sebelum memulai iterasi optimasi.

Gambar 2. Kode Fungsi Inisialisasi Variabel Awal

```
% Inisialisasi variable
rute_optimum = [];
jarak_minimum = Inf;
t = 0;
t_awal = cputime;
% Inisialisasi partikel secara acak
x = zeros(np, size(matriks_jarak, 1)); % Inisialisasi matriks posisi
partikel
for i = 1:np
    x(i, :) = randperm(size(matriks_jarak, 1)); % Urutan acak kota
end
x_awal = x; % Simpan posisi partikel yang diinisialisasi
```

Step 2. Inisialisasi secara random partikel x_i dan kecepatan v_i dalam ruangan pencarian problem p-dimensi.

Inisialisasi secara random partikel x_i dan kecepatan v_i dalam ruang pencarian problem dimensi adalah salah satu langkah dalam algoritma PSO yang bertujuan untuk mendistribusikan solusi kandidat (partikel) secara acak dalam ruang pencarian yang ditentukan.

Gambar 3. Kode Inisialisasi Partikel Secara Acak

```
% Inisialisasi partikel secara acak
x = zeros(np, size(matriks_jarak, 1)); % Inisialisasi matriks posisi partikel
for i = 1:np
    x(i, :) = randperm(size(matriks_jarak, 1)); % Urutan acak kota
end
x_awal = x; % Simpan posisi partikel yang diinisialisasi

% Inisialisasi kecepatan partikel
vmax = 0.2 * (bb - ba); % Inisialisasi kecepatan maksimum
v = rand(np, size(matriks_jarak, 1)) - 0.5; % Inisialisasi kecepatan dengan
nilai acak antara -0.5 dan 0.5
v = 2 * vmax * v; % Skala nilai kecepatan ke rentang (-vmax, vmax)
v = min(max(v, -vmax), vmax); % Batasi nilai kecepatan sesuai dengan
batasan vmax
v_awal = v; % Simpan kecepatan partikel yang diinisialisasi
```

Step 3. Melakukan uji fungsi jartsp (jarak sebelum optimasi)

Sebelum melakukan optimasi, kita perlu mengukur jarak total awal dari rute yang dihasilkan oleh posisi partikel awal. Fungsi ini akan membantu kita memahami seberapa baik algoritma PSO mengoptimalkan rute.

Gambar 4. Kode Uji Fungsi Jartsp

```
rute = [1,2,3,4,5,6,7,8,9];
jarak = Jartsp(rute, matriks_jarak);
disp('Rute Awal:');
disp(rute);
disp('Jarak total Sebelum optimasi:');
disp(jarak);
```

Step 4. Inisialisasi variabel tambahan untuk menyimpan data iterasi.

Kita perlu menginisialisasi beberapa variabel tambahan untuk menyimpan data hasil iterasi seperti nilai terbaik dari setiap iterasi dan posisi terbaik global.

Gambar 5. Kode Inisialisasi Variabel Tambahan

```
% Inisialisasi variabel tambahan untuk menyimpan data iterasi
semua_matriks = cell(itmax, 1);
semua_pbest = cell(itmax, 1);
semua_gbest = cell(itmax, 1);
nilai_rute_iterasi = zeros(itmax, 1); % Variabel untuk menyimpan nilai
rute tiap iterasi
```

Step 5. Melakukan loop PSO utama

Pada tahap ini adalah inti dari algoritma PSO dimana kita akan memperbarui posisi dan kecepatan partikel, serta melacak posisi terbaik setiap partikel dan posisi terbaik global.

Gambar 6. Kode Program Loop PSO Utama

```
% Loop PSO utama
for iter = 1:itmax
    % Perbarui posisi dan kecepatan partikel
    for i = 1:np
        xi = x(i, :);
        % Hitung komponen kognitif dan sosial
        cognitive = c1 * rand(1, size(matriks_jarak, 1)) .* (pbest(i, :) - xi);
        social = c2 * rand(1, size(matriks_jarak, 1)) .* (gbest - xi);
        % Hitung kecepatan baru dengan konstriksi
        v(i, :) = w * (v(i, :) + cognitive + social);
        v(i, :) = min(max(v(i, :), -vmax), vmax);
        % Perbarui posisi
        x(i, :) = x(i, :) + v(i, :);
        % Terapkan batasan posisi
        x(i, :) = round(max(min(x(i, :), size(matriks_jarak, 1)), 1));
        % Pastikan bahwa nilai kota asal dan tujuan tidak bernilai 0
        if x(i, 1) == 0
            x(i, 1) = 1;
        end
        if x(i, end) == 0
            x(i, end) = 1;
        end
    end

    % Perbaiki rute yang memiliki kota yang muncul lebih dari sekali
    for i = 1:np
        for j = 1:size(x, 2)
            if nnz(x(i, :) == x(i, j)) > 1
                kota_tidak_dikunjungi = setdiff(1:size(matriks_jarak, 1), x(i, :));
                for k = 1:length(kota_tidak_dikunjungi)
                    if ~ismember(kota_tidak_dikunjungi(k), x(i, :))
                        x(i, j) = kota_tidak_dikunjungi(k);
                        break;
                    end
                end
            end
        end
    end
end
```

```
        end
    end

    % Evaluasi nilai fitness untuk semua partikel
    jarak = zeros(np, 1);
    for i = 1:np
        try
            jarak(i) = Jartsp(x(i, :), matriks_jarak);
        catch err
            warning(['Error occurred while calculating fitness for particle ',
num2str(i), ': ', err.message]);
            jarak(i) = Inf;
            continue;
        end
    end
end
```

Step 6. Memperbarui posisi terbaik untuk setiap partikel dan global

Selama iterasi, kita harus terus memperbarui posisi terbaik individu dan global berdasarkan evaluasi fungsi objektif.

Gambar 7. Kode Program Memperbarui Posisi Terbaik Tiap Partikel

```
% Perbarui posisi terbaik untuk setiap partikel
for i = 1:np
    if jarak(i) < Jartsp(pbest(i, :), matriks_jarak)
        pbest(i, :) = x(i, :);
    end
end

% Perbarui posisi terbaik global (gbest)
[jarak_min_baru, idx_min] = min(jarak);
if jarak_min_baru < jarak_minimum
    jarak_minimum = jarak_min_baru;
    rute_optimum = x(idx_min, :);
    gbest = x(idx_min, :);
end
```

Step 7. Menyimpan data hasil iterasi

Untuk analisis dan visualisasi kinerja algoritma, kita menyimpan nilai terbaik dari setiap iterasi dan posisi terbaik global.

Gambar 8. Kode Program Menyimpan Data Hasil Iterasi

```
% Simpan data iterasi
semua_matriks{iter} = x;
semua_pbest{iter} = pbest;
semua_gbest{iter} = gbest;

% Simpan nilai rute tiap iterasi
nilai_rute_iterasi(iter) = jarak_minimum;
```

Step 8. Menampilkan nilai rute sementara

Menampilkan nilai rute sementara pada algoritma PSO, berarti memonitor dan menampilkan rute terbaik yang ditemukan sejauh ini selama proses iterasi dari algoritma PSO tersebut.

Gambar 9. Kode Program Menampilkan Hasil Sementara

```
% Tampilkan nilai rute sementara
disp(['Iterasi ke-', num2str(iter), ': Jarak sementara = ',
num2str(jarak_minimum)]);
end

% Perbarui waktu komputasi
t = cputime - t_awal;
```

Step 9. Membuat plot grafik perjalanan nilai rute tiap iterasi

Membuat plot grafik perjalanan nilai rute tiap iterasi adalah cara yang bagus untuk memvisualisasikan bagaimana nilai rute terbaik berubah seiring dengan berjalannya algoritma optimasi. Plot ini membantu dalam memahami konvergensi algoritma dan mengevaluasi kinerja algoritma optimasi

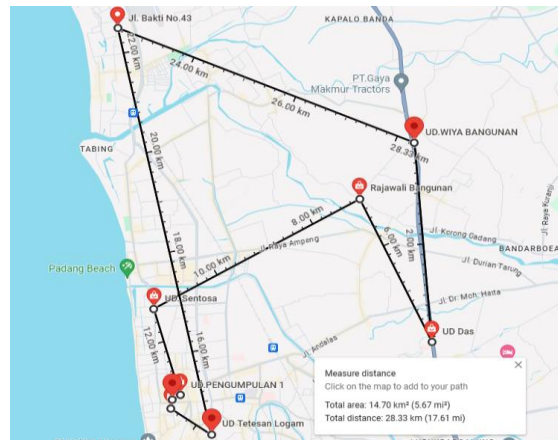
Gambar 10. Kode Membuat Plot Grafik Perjalanan Rute Tiap Iterasi

```
% Plot grafik perjalanan nilai rute tiap iterasi
figure;
plot(1:itmax, nilai_rute_iterasi, 'b-', 'LineWidth', 2);
xlabel('Iterasi');
ylabel('Jarak Minimum');
title('Perjalanan Nilai Rute Tiap Iterasi');
end
```

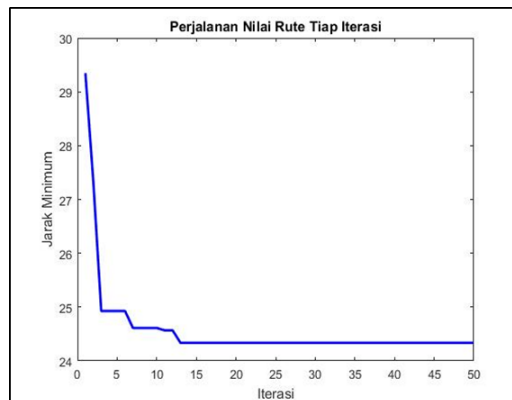
Pada penerapan implementasi metode PSO untuk TSP diatas setelah dilakukan analisis menggunakan 50 iterasi dan 50 partikel (X_i) maka didapatkan hasil rute

minimum sebesar 24.335 km dengan urutan rute 1 – 2 – 4 – 3 – 9 – 5 – 6 – 7 – 8 – 1. Dimana rute awal mulai dari Gudang UD Logam Guru, UD Wiya, UD Das, Toko Rajawali, UD Sentosa, UD Pengumpulan, UD Antika, Toko Sumber Baru, UD Wiya, UD Tetes Logam, kembali lagi ke Gudang UD Logam Guru. Peta rute distribusi setelah optimasi menggunakan metode PSO dapat dilihat pada Gambar 11.

Gambar 11. Peta Rute Distribusi Setelah Optimasi Menggunakan Metode PSO



Selanjutnya grafik hasil dari 50 iterasi jarak minimum yang dijalankan model PSO dapat dilihat pada Gambar 12.



Gambar 12. Grafik Iterasi Jarak Minimum

SIMPULAN

Dari hasil penelitian pada optimalisasi pendistribusian besi pada UD Logam Guru dengan penerapan algoritma *Particle Swarm Optimization* dapat disimpulkan pada perhitungan MATLAB menggunakan metode PSO diperoleh rute terbaik yaitu dari Gudang UD Logam Guru, UD Wiya, UD Das, Toko Rajawali, UD Sentosa, UD Pengumpulan, UD Antika, Toko Sumber Baru, UD Tetesan Logam, Kembali lagi ke Gudang UD Logam Guru, dengan total jarak 24.335 km. Rute yang diperoleh lebih optimal dari rute yang biasa ditempuh UD Logam Guru yaitu dengan jarak tempuh 29.345 km. Sehingga diperoleh rute optimal lebih pendek 5.01 km. Rute ini dapat dijadikan pertimbangan oleh UD Logam Guru untuk mencoba rute dan solusi yang penulis rekomendasikan agar jarak yang ditempuh lebih pendek sehingga biaya distribusi yang diperlukan pada saat mengantarkan besi menjadi lebih kecil.

DAFTAR PUSTAKA

- Adventia, A., Novianingsih, K., & Serviana, H. (2016). *Penyelesaian Masalah Pendistribusian Barang Menggunakan Algoritma Bee Colony Optimization Solving the Distribution Problem Using Bee Colony Optimization Algorithm*. 64–72.
- Aldous, J. M., & Wilson, R. J. (2004). *Graphs and Applications And Introductory Approach*. Springer-Verlag London Berlin heiderlberg.
- Bustani, H. (2005). *Fundamental Operation Research*. PT Gramedia Pustaka Utama.
- Fuadiyah, M. (2018). *Algoritma Particle Swarm (PSO) dan Aplikasi pada Masalah Capacitated Vehicle Routing Problem (CVRP)*.
- Gunariantio, B. M. (2019). *Desain Lokasi Gudang Hub Dan Optimisasi Rute Distribusi Kendaraan Angkutan Produk Air Mineral Dalam Kemasan Menggunakan Particle Swarm Optimization (Studi Kasus: Pt. Muawanah Al-Masoem)*.
- Gutin, G., & Punnen, A. P. (2004). *The Travelling Salesman Problem and Its Variations*. Kluwe Academic Publishers.
- Heizer, J., & Render, B. (2011). *Operations Management* (10th ed.). Pearson Education.
- Hidayat, M., Cholissodin, I., & Rahayudi, B. (2020). *Optimasi Multiple Travelling Salesman Problem (M-TSP) pada Penentuan Rute Angkutan Sekolah menggunakan Algoritme Particle Swarm Optimization (PSO)*. 4(9), 2878–2885. <http://j-ptiik.ub.ac.id>.
- Immanuel, R. M. (2015). *Algoritma Particle Swarm Optimization Untuk Menyelesaikan Multi Depot Vehicle Routing Prolem Dengan Variabel Travel Time*. <https://repository.its.ac.id/52056/>.
- Lattan, B. W., Tupan, J. M., & Paillin, D. B. (2018). Pemecahan Traveling Salesmen Problem Menggunakan Teknik Branch and Bound Dan Cheapest Insertion Heuristic. *I Tabaos*, 1(1), 13–22. <https://doi.org/10.30598/i-tabaos.2021.1.1.13-22>.
- Munir, R. (2005). *Matematika Diskrit Edisi Ketiga*. Informatika.
- Natalia, D., Yundari, & Yudhi. (2019). Optimasi Jarak Penjemputan Penumpang Cv.

- Eira Saudara Menggunakan Metode *Particle Swarm Optimization* Studi Kasus : Travel Taxi Tiga Saudara Pontianak Kalbar. *Bimaster : Buletin Ilmiah Matematika, Statistika Dan Terapannya*, 8(3), 531–538.
<https://doi.org/10.26418/bbimst.v8i3.33852>
- Rao, S. S. (2009). *Engineering Optimization Theory and Practice* (Fourth Edi). John Willey and Sons.
- Robandi, I. (2019). *Artifical Intelligence-Mengupas Rekayasa Kecerdasan Tiruan* (M. Kika (ed.)). Penerbit ANDI.
- Russell, R. S., & Taylor, B. W. (2009). *Operation Management: Creating Value along the Supply Chain, 6th Edition*. John Willey and Sons.
- Suwarno, H. L. (2006). Sembilan Fungsi Saluran Distribusi: Kunci Pelaksanaan Kegiatan Distribusi yang Efektif. *Jurnal Manajemen*, 6.