

---

**RUTE TERPENDEK ALGORITMA PARTICLE SWARM OPTIMIZATION DAN  
BRUTE FORCE UNTUK OPTIMASI TRAVELLING SALESMAN PROBLEM**

**Muchamad Kurniawan<sup>1\*</sup>, Farida<sup>2</sup>, Siti Agustini<sup>3</sup>**

<sup>1,2,3</sup>Teknik Informatika, Fakultas Teknik Elektro dan Teknologi Informasi

<sup>1,2,3</sup>Institut Teknologi Adhi Tama Surabaya

<sup>1,2,3</sup>Jl. Arief Rahman Hakim No.100, Surabaya

E-mail: <sup>1\*</sup>[muchamad.kurniawan@itats.ac.id](mailto:muchamad.kurniawan@itats.ac.id), <sup>2</sup>[farida@itats.ac.id](mailto:farida@itats.ac.id), <sup>3</sup>[sitiagustini@itats.ac.id](mailto:sitiagustini@itats.ac.id)

---

**ABSTRACT**

---

**Artikel:**

Diterima: 11 Januari, 2021

Direvisi: 20 Desember, 2021

Diterbitkan: 17 Januari, 2022

---

**\*Alamat Korespondensi:**

[muchamad.kurniawan@itats.ac.id](mailto:muchamad.kurniawan@itats.ac.id)

Distribution becomes an important measure of marketing success. Traveling Salesman Problem is an example of a case that can be implemented in a distribution case study to get the shortest route through which a distributor passes. The distributor must pass each node (address or city) once in a while and then return to the node where he started. Traveling salesman problems emerge as part of logistical and transportation problems that have developed and utilized in the current period which is growing in various sectors. This research proposes using the Particle Swarm Optimization and Brute Force method to compare the performance of the two methods to get the shortest route. The study was conducted in several experiments the number of points (nodes) namely 5, 10, 15, 20, 25, and 30 nodes. Overall experiments, the Particle Swarm Optimization algorithm is superior to Brute Force. The route produced by Particle Swarm Optimization has a shorter distance than Brute Force

**Keywords:** *Travelling Salesman Problem, Particle Swarm Optimization, Brute Force*

---

## ABSTRAK

Distribusi menjadi ukuran penting dari keberhasilan pemasaran. Traveling Salesman Problem adalah contoh kasus yang dapat diimplementasikan dalam studi kasus distribusi untuk mendapatkan rute terpendek yang dilewati oleh seorang distributor. Distributor harus melewati tiap-tiap node (alamat atau kota) tersebut satu kali dalam kemudian kembali ke node dimana dia memulai. Traveling salesman problem muncul sebagai bagian dari permasalahan logistik dan transportasi yang telah berkembang dan dimanfaatkan pada kurun waktu saat ini yang tumbuh di berbagai sektor. Pada penelitian ini diusulkan menggunakan metode Particle Swarm Optimization dan Brute Force untuk dibandingkan performansi kedua metode tersebut untuk mendapatkan rute terpendek. Penelitian dilakukan dalam beberapa percobaan jumlah titik (node) yaitu 5, 10, 15, 20, 25, dan 30 node. Secara keseluruhan percobaan, algoritma Particle Swarm Optimization lebih unggul dibandingkan Brute Force. Rute yang dihasilkan Particle Swarm Optimization memiliki jarak tempuh lebih pendek dibandingkan Brute Force.

**Kata Kunci:** *Travelling Salesman Problem, Particle Swarm Optimization, Brute Force*

---

## I. PENDAHULUAN

Salah satu indikator kesuksesan sebuah perusahaan dapat diukur dari distribusi produknya. Pendistribusian produk yang luas akan memaksimalkan keuntungan perusahaan. Keterlambatan dalam pendistribusian produk akan merugikan perusahaan dan juga konsumen. Tingkat penjualan akan menurun sehingga target penjualan yang telah ditentukan tidak tercapai. Permasalahan ini akan membuat pendapatan perusahaan tidak sesuai dengan yang diharapkan. Kerugian dari sisi konsumen adalah terbatasnya barang yang tersedia di pasaran.

Untuk mendukung distribusi yang lancar maka diperlukan pemilihan dan penentuan jalur distribusi. Namun, penentuan jalur distribusi yang efektif tidaklah mudah. Penentuan jalur distribusi yang salah akan menyebabkan terjadinya penumpukan produk pada gudang sehingga menyebabkan biaya simpan yang besar dan menghambat produk sampai ke konsumen yang dapat memungkinkan kerusakan produk semakin besar. Oleh karena itu, penentuan jalur distribusi yang efektif sangat penting bagi perusahaan dalam distribusi produk. Rute pengiriman produk yang efektif dari satu titik ke titik yang lain dapat meminimalkan jarak

tempuh pengiriman barang sehingga biaya kirim produk juga minim.

PT. X adalah sebuah perusahaan air mineral yang telah memiliki banyak outlet untuk memenuhi kebutuhan konsumen setiap hari. Perusahaan ini menuntut pendistribusian air mineral yang cepat kepada pelanggan yang telah ditetapkan. Distributor harus melewati setiap node (alamat atau kota) tersebut satu kali kemudian kembali ke node dimana dia memulai. Salah satu solusi yang dapat diterapkan untuk mengatasi masalah ini adalah Travelling Salesman Problem (TSP). TSP dapat digunakan oleh distributor untuk mencari dan menentukan rute terpendek. TSP merupakan bagian dari solusi permasalahan logistik dan transportasi saat ini dalam berbagai sektor [1-2].

TSP diimplementasikan untuk menghasilkan nilai optimal atau rute terpendek. Salah satu metode TSP adalah teknik brute force. Teknik ini bekerja dengan mencoba semua kemungkinan rute yang ada sehingga teknik ini tidak relevan ketika node atau tempat tujuan terlalu banyak dikarenakan kompleksitas waktu dalam TSP ini masuk dalam NP-hard. Karena itu, dibutuhkan sebuah teknik heuristic. Teknik heuristic adalah teknik untuk mencari solusi dengan sebuah pendekatan nilai optimal dengan harapan hasil yg diharapkan mendekati

optimal dengan kompleksitas waktu yg relevan. Salah satu teknik heuristic untuk menyelesaikan problem TSP yg sering digunakan adalah Particle Swarm Optimization (PSO). PSO diperkenalkan oleh Kennedy dan Eberhart [3] dengan mengadaptasi perilaku sekumpulan burung yang mencari makanan.

Pemecahan masalah mengenai TSP telah dilakukan melalui pendekatan komputasi tradisional [4]. Penelitian tersebut mengimplementasikan metode Brute Force. Metode Brute Force melakukan pencarian dari semua hasil kemungkinan yang ada. Karena itu, metode Brute Force membutuhkan waktu lama untuk menghasilkan solusi optimal. Pada penelitian tersebut, telah dilakukan percobaan dengan 12 node dan menghasilkan solusi terbaik dalam waktu 12.5 detik. Penelitian lain dilakukan oleh [5] yang membandingkan performa algoritma genetika (GA) dengan metode Particle Swarm Optimization (PSO). Hasil dari penelitian ini menunjukkan bahwa PSO memiliki convergence speed yang lebih tinggi daripada GA karena GA mudah terjebak dalam local optimum. Penelitian serupa yang membandingkan metode GA, PSO dan Simulated Annealing (SA) juga memberikan hasil yang paling optimal adalah PSO [6]. PSO juga mempunyai akurasi yang lebih baik dibandingkan Genetic Algorithm [7] [8]. Analisa akurasi PSO lebih bagus dikarenakan update individu yang ada di PSO didasari oleh gradient, dikarenakan berdasarkan gradient maka penyelidikan setiap individu lebih detail sehingga searching yang dilakukan lebih menyeluruh.

Manfaat penelitian ini dapat digunakan untuk penentuan rute terpendek pada studi kasus yang lain, misalkan studi kasus pengiriman agen besar elpiji kea gen kecil, pengiriman paket dan surat, marketing atau pengiriman dari Gudang ke kios-kios, ataupun dapat digunakan untuk aplikasi pengiriman buah.

Penelitian ini membahas tentang penyelesaian masalah optimasi TSP menggunakan metode brute force dan PSO sebagai pembandingan. Data yang digunakan adalah lokasi-lokasi pelanggan PT.X yang ada di Surabaya. Data lokasi konsumen akan diolah menjadi sebuah graf. Hasil akhir yang diharapkan adalah rute optimal atau rute terpendek.

## II. METODOLOGI

### 2.1 Brute Force

Brute Force merupakan algoritma sederhana yang melakukan pencarian menyeluruh terhadap kemungkinan yang ada [9]. Brute force memanfaatkan Faktorial (!) untuk menghitung banyaknya atau jumlah susunan yang dapat dibentuk dari sekumpulan atau populasi tanpa memperhatikan urutannya. Teknik Brute Force digunakan untuk mencari penyelesaian TSP dengan mencari semua kemungkinan rute yang ada. Dimana Faktorial dapat direpresentasikan dalam persamaan (1).

$$n! = 1 \times 2 \times \dots \times (n-2) \times (n-1) \times n \quad (1)$$

### 2.2 Particle Swarm Optimization

Particle Swarm Optimization didasarkan oleh perilaku social sebuah kawanan burung atau ikan. Perilaku sosial ini terdiri dari perilaku individu dan pengaruhnya terhadap individu lain dalam suatu kawanan [10]. Particle mendefinisikan sebuah individu dalam kawanan (swarm) burung. Sebagai contoh, jika satu partikel (burung) menemukan jalan atau rute terpendek menuju sumber makanan maka kawanannya lain juga akan segera mengikuti jalan tersebut mesti lokasi mereka jauh. Pada algoritma PSO, pencarian solusi dilakukan oleh populasi yang terdiri dari beberapa partikel. Populasi dibuat secara acak dengan batasan nilai terkecil dan terbesar. Sedangkan partikel mencerminkan posisi atau solusi dari permasalahan yang dihadapi. Setiap partikel melakukan pencarian solusi yang optimal melalui ruang pencarian (search space). Sebuah partikel direpresentasikan seperti sebuah titik dalam dimensi ruang tertentu. Hal ini dilakukan dengan memperhatikan partikel secara individu dan partikel terbaik secara global selama proses pencarian solusi. Kemudian, dilakukan pencarian posisi terbaik dari setiap partikel dalam sejumlah iterasi tertentu sampai diperoleh posisi relative steady.

Algoritma PSO secara umum dapat dilihat pada Gambar 1. Proses pertama adalah Data Preparation, pada proses ini berungsi untuk mengenerate individu awal. Proses selanjutnya adalah Inisialisasi Parameter, pada proses ini terdapat parameter maksimum iterasi,  $c1$  dan  $c2$ ,  $w$  (inersia) yang perlu diatur nilainya. Setelah itu terdapat proses utama PSO dimana harus melewati syarat jika iterasi < epoch. Pada proses utama terdapat proses penentuan Global

Best (Gbest) dan Local Best (Pbest), setelah penentuan Pbest dan Gbest Langkah seterusnya adalah update nilai Pbest dan Gbest yang telah ada. Jika Pbest Baru lebih baik maka nilai yang baru menggantikan. Proses terakhir adalah menghitung nilai kecepatan dan merubah posisi berdasarkan kecepatan yang didapatkan.

Terdapat 2 faktor yang memberikan karakter terhadap status partikel pada search space yaitu posisi dan kecepatan artikel. Persamaan 2 dan 3 merupakan formula untuk menentukan posisi dan kecepatan partikel pada ruang tertentu.

$$V_i(t) = V_i(t - 1) + c_1 r_1 (X_i^L - X_i(t - 1)) + c_2 r_2 (X^G - X_i(t - 1)) \quad (2)$$

$$X_i(t) = V_i(t) + X_i(t - 1) \quad (3)$$

dimana :

$V_i(t)$  = Kecepatan partikel pada iterasi ke-t

$X_i(t)$  = Posisi partikel pada iterasi ke-t

$c_1$  dan  $c_2$  = konstanta bernilai positif (learning factor)

$r_1$  dan  $r_2$  = bilangan random antara 0 sampai 1

$X_i^L = X_{i1}^L, X_{i2}^L, \dots, X_{iN}^L$  = local best dari partikel ke-i

$X^G = X_1^G, X_2^G, \dots, X_N^G$  = global best dari seluruh kawanan



Gambar 1. Algoritma PSO

Persamaan 1 digunakan saat menghitung kecepatan partikel yang baru berdasarkan kecepatan sebelumnya, jarak antara posisi sekarang dan posisi terbaik partikel (local best) dan posisi saat ini dengan posisi terbaik populasi (global best). Kemudian persamaan 2 dijalankan dengan sejumlah at iterasi tertentu hingga mencapai kriteria sehingga didapatkan solusi yang merupakan solusi global terbaik (global best). Pada setiap iterasi solusi yang direpresentasikan oleh posisi partikel kemudian dilakukan evaluasi performa

dengan cara memasukkan solusi tersebut ke dalam fitness function.

### III. HASIL DAN PEMBAHASAN

#### 3.1 Skenario Uji Coba

Pada penelitian ini dilakukan beberapa tahap untuk mencapai hasil penelitian yang dapat dilihat pada Gambar 2. Yang pertama ada Data Preparation yaitu data yang disiapkan berupa titik atau alamat lokasi dari pelanggan. Tabel 1 merupakan contoh dari 5 titik alamat pelanggan. Alamat yang telah ditentukan diproses lebih lanjut untuk diperoleh jarak antar masing-masing titik atau alamat. Untuk memperoleh jarak antar titik sebagai acuan bobot atau cost maka digunakan bantuan dari google maps API. Hasil penentuan jarak antar titik dapat dilihat dari Tabel 2.



Gambar 2. Blok Diagram Penelitian

Tabel 1. Data Alamat Pelanggan

Titik ke-	Alamat
1	Jl. Bubutan I, Alun-alun Contong, Bubutan, Kota SBY, Jawa Timur 60174
2	Jl. Kanginan III No.5, Ketabang, Genteng, Kota SBY, Jawa Timur 60272
3	Jl. Maspati IV No.55, RT.002/RW.07, Bubutan, Kota SBY, Jawa Timur 60174
4	Jl. Dinoyo no. 20, Tegalsari, Kota SBY, Jawa Timur 60265
5	Jl. Raya Mulyosari No.328, Mulyorejo, Kota SBY, Jawa Timur 60112

Tabel 2. Data Jarak antar Alamat Pelanggan

Titik ke-	1	2	3	4
1	0	2756	1766	3842
2	2756	0	3950	3350
3	1766	3950	0	4913
4	3842	3350	4913	0
5	10727	9109	11799	7617

Setelah jarak antar titik didapatkan, maka data preparation diolah menggunakan PSO dan Brute Force. Jumlah titik atau node yang digunakan dalam penelitian adalah 5, 10, 15, 20, 25, dan 30 node. Hasil dari percobaan dibandingkan berdasarkan perolehan jarak terpendek.

#### 3.2 Perbandingan Kinerja PSO dan Brute Force

Perbandingan kinerja antara PSO dan Brute Force dilakukan dengan jumlah node mulai 5 node hingga 30 node dengan kelipatan 5 node. Waktu pemrosesan dibatasi dalam waktu 2 jam. Hasil yang ditunjukkan pada Tabel 2 menunjukkan perolehan jarak dari rute yang dihasilkan oleh PSO dan Brute Force pada jumlah node 5 dan 10 memiliki jarak yang sama. Namun, mulai jumlah node 15 sampai 30 perolehan jarak rute terpendek mulai berbeda. Hasil perolehan rute terpendek PSO memiliki jarak yang lebih pendek dari Brute Force di segala jumlah node. Selain itu, dalam selisih hasil total rute di bawah menunjukkan semakin banyak jumlah node yang terlibat makan semakin besar selisih total jarak antara PSO dan Brute force. Hal ini membuat PSO lebih unggul dari Brute Force.

Pencarian solusi TSP menggunakan Brute Force diawali dari Tabel 2 yang berisi jarak antar titik (alamat). Pada table 2 memuat 4 titik (alamat). Teknik Brute Force akan banyaknya mencari susunan yang dapat dibentuk dari keempat titik tersebut. Maka, banyaknya susunan rute yang dapat dicapai sesuai persamaan (1) adalah  $4! = 1 \times 2 \times 3 \times 4 = 24$  susunan. Detail susunan rute dapat dilihat pada Tabel 4. Dari ke 24 susunan rute yang dihasilkan dipilih rute terpendek sebagai hasilnya yaitu no 4 (1, 3, 4, 2).

Tabel 3. Hasil Perbandingan Kinerja PSO dan Brute Force

Jumlah Node	Total Jarak Rute (meter)		Selisih Jarak (meter)	Rute Terpendek	
	Brute Force	PSO		Brute Force	PSO
5	42942	42942	0	1,5,3,4,2	1,5,3,4,2
10	56938	56938	0	1,5,9,7,4,10,2,3,6,8	1,5,9,7,4,10,2,3,6,8
15	98910	72847	26063	1,2,3,4,5,6,15,9,12,7, 13,10,14,11,8,1	1,15,5,9,12,7,10,13, 4,2,3,14,6,11,8
20	148334	98319	50015	1,2,3,4,5,6,7,8,9,10,12, 13,19,17,14,18,16,20,15, 11,1	1,15,8,18,11,6,5,20,9, 12,7,17,2,10,13,4,19,14, 16,3
25	208879	170644	38235	1,2,3,4,5,6,7,8,9,10,11,12, 13,14,15,16,20,18,17,19,22, 21,25,24,23,1	1,11,5,9,14,18,25,21,20, 12,7,4,19,3,17,2,13,22,10, 23,15,24,16,6,8
30	254389	171875	82514	1,2,3,4,5,6,7,8,9,10,11,12,13, 14,15,16,17,18,19,20,22,30,21, 25,27,28,26,23,24,29,1	1,24,29,5,12,9,25,27,16,3, 20,11,26,10,2,7,13,4,19,22, 17,23,8,15,21,30,14,18,28,6

Tabel 4. Susunan Rute Hasil Brute Force

No.	Rute	Jarak (meter)	No.	Rute	Jarak (meter)
1	1, 2, 3, 4	54334	13	3, 2, 1, 4	56457
2	1, 2, 4, 3	42811	14	3, 2, 4, 1	45734
3	1, 3, 2, 4	47534	15	3, 1, 2, 4	42811
4	1, 3, 4, 2	38930	16	3, 1, 4, 2	47270
5	1, 4, 3, 2	56457	17	3, 4, 1, 2	54434
6	1, 4, 2, 3	47270	18	3, 4, 2, 1	38930
7	2, 1, 3, 4	38930	19	4, 2, 3, 1	47270
8	2, 1, 4, 3	56457	20	4, 2, 1, 3	38930
9	2, 3, 1, 4	47270	21	4, 3, 2, 1	56457
10	2, 3, 4, 1	54334	22	4, 3, 1, 2	42811
11	2, 4, 3, 1	42811	23	4, 1, 3, 2	47534
12	2, 4, 1, 3	47534	24	4, 1, 2, 3	54334

Dalam *Particle Swarm Optimization* ada beberapa parameter yang harus di inialisasi yaitu :

1. Jumlah partikel dan Rute awal

Jumlah partikel/individu yang akan memberikan kandidat solusi dibangkitkan secara acak, misal jumlah partikel yang akan dibuat adalah 3.

- Membangkitkan populasi partikel dengan bilangan acak antara 0 hingga 1.

$$pop = \begin{bmatrix} 0.8023 & 0.0732 & 0.0392 & 0.5178 & 0.9350 \\ 0.4242 & 0.0591 & 0.9463 & 0.9942 & 0.4795 \\ 0.7289 & 0.9102 & 0.7637 & 0.8549 & 0.2138 \end{bmatrix}$$

- Untuk setiap partikel, urutkan bilangan random dari yang terkecil. Pengurutan ini akan menghasilkan rute.

$$popurut = \begin{bmatrix} 0.0392 & 0.0732 & 0.5178 & 0.8023 & 0.9350 \\ 0.4242 & 0.4795 & 0.5910 & 0.9463 & 0.9942 \\ 0.2318 & 0.7289 & 0.7637 & 0.8549 & 0.9120 \end{bmatrix}$$

- Setelah masing-masing partikel diurutkan maka akan didapat rute yang dibentuk dari posisi bilangan random tersebut sebelum diurutkan.

$$rute = \begin{bmatrix} 3 & 2 & 4 & 1 & 5 \\ 1 & 5 & 2 & 3 & 4 \\ 5 & 1 & 3 & 4 & 2 \end{bmatrix}$$

- Untuk menghasilkan tur/sirkuit maka angka pertama akan ditambahkan di bagian akhir, seperti contoh pada partikel pertama akan menghasilkan solusi 3-2-4-1-5-3.

2. Kecepatan partikel

Setiap partikel akan mendapat nilai kecepatan yang sama pada saat inialisasi. Misal kecepatan awal diberikan :

$$v_0 = \begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix}$$

3. Nilai konstanta pertama dan kedua

Kontanta adalah nilai bersifat positif atau biasa disebut *learning rate* untuk kemampuan individu (kognitif) dan pengaruh sosial (kawanannya).

4. *Epoch*

Jumlah iterasi yang digunakan sebagai *stopping criteria*.

5. Inersia

Nilai yang berguna untuk meredam laju kecepatan pada formula *update* agar terhindar dari terlewatnya nilai minimum fungsi tujuan.

6. Nilai random pertama dan kedua ( $r_1$  dan  $r_2$ )

Nilai random yang bernilai antara 0 dan 1. Sebagai contoh dalam sistem ini nilai  $r_1 = 0.4$  dan  $r_2 = 0.6$ .

Setelah inisialisasi, proses selanjutnya adalah menentukan *Local Best* dan *Global Best* yaitu:

$$x_1 = 3 - 2 - 4 - 1 - 5 - 3$$

$$= 3950 + 3350 + 3842$$

$$+ 10727 + 11799 = 3368$$

$$x_2 = 1 - 5 - 2 - 3 - 4 - 1$$

$$= 10727 + 9109 + 3950$$

$$+ 4913 + 3842 = 32541$$

$$x_3 = 5 - 1 - 3 - 4 - 2 - 5$$

$$= 10727 + 1766 + 4913$$

$$+ 3350 + 9190 = 29946$$

Untuk setiap partikel, *local best* (Lbest) akan sama dengan nilai partikel awal karena dalam proses iterasi pertama dan *global best* (Gbest) adalah solusi optimal dan ada pada partikel ke 3 karena memiliki nilai paling minimum.

Langkah selanjutnya adalah update kecepatan :

- Kecepatan partikel pertama

$$v_1 = (0.9)([0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1]) +$$

$$0.4 \left( \frac{[0.8023 \ 0.0732 \ 0.0392 \ 0.5178 \ 0.9350] - [0.8023 \ 0.0732 \ 0.0392 \ 0.5178 \ 0.9350]}{[0.8023 \ 0.0732 \ 0.0392 \ 0.5178 \ 0.9350]} \right) +$$

$$0.6([0.7289 \ 0.9102 \ 0.7673 \ 0.8549 \ 0.2318] - [0.8023 \ 0.0732 \ 0.0392 \ 0.5178 \ 0.9350])$$

$$v_1 = [0.0460 \ 0.5922 \ 0.5269 \ 0.2927 \ -0.3320]$$

- Kecepatan partikel kedua

$$v_1 = (0.9)([0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1]) +$$

$$0.4 \left( \frac{[0.4242 \ 0.5910 \ 0.9463 \ 0.9942 \ 0.4795] - [0.4242 \ 0.5910 \ 0.9463 \ 0.9942 \ 0.4795]}{[0.4242 \ 0.5910 \ 0.9463 \ 0.9942 \ 0.4795]} \right) +$$

$$0.6([0.7289 \ 0.9102 \ 0.7673 \ 0.8549 \ 0.2318] - [0.4242 \ 0.5910 \ 0.9463 \ 0.9942 \ 0.4795])$$

$$v_1 = [0.2728 \ 0.2815 \ -0.0174 \ 0.0064 \ -0.0586]$$

- Kecepatan partikel ketiga

$$v_1 = (0.9)([0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1]) +$$

$$0.4 \left( \frac{[0.7289 \ 0.9102 \ 0.7673 \ 0.8549 \ 0.2318] - [0.7289 \ 0.9102 \ 0.7673 \ 0.8549 \ 0.2318]}{[0.7289 \ 0.9102 \ 0.7673 \ 0.8549 \ 0.2318]} \right) +$$

$$0.6([0.7289 \ 0.9102 \ 0.7673 \ 0.8549 \ 0.2318] - [0.7289 \ 0.9102 \ 0.7673 \ 0.8549 \ 0.2318])$$

$$v_1 = [0.09 \ 0.09 \ 0.09 \ 0.09 \ 0.09]$$

Sehingga didapat kecepatan akhir berupa

$$v_1 = \begin{bmatrix} 0.0460 & 0.5922 & 0.5269 & 0.2927 & -0.3320 \\ 0.2728 & 0.2815 & -0.0174 & 0.0064 & -0.0586 \\ 0.09 & 0.09 & 0.09 & 0.09 & 0.09 \end{bmatrix}$$

Kemudian laikan perubahan nilai dari partikel dengan rumus

$$pop_1 = pop_0 + v_1$$

Jika nilai elemen setiap partikel di luar batas yang diijinkan, yaitu antara  $bb$  (batas bawah) bernilai 0 dan  $ba$  (batas atas) bernilai 1, maka akan melakukan penyesuaian

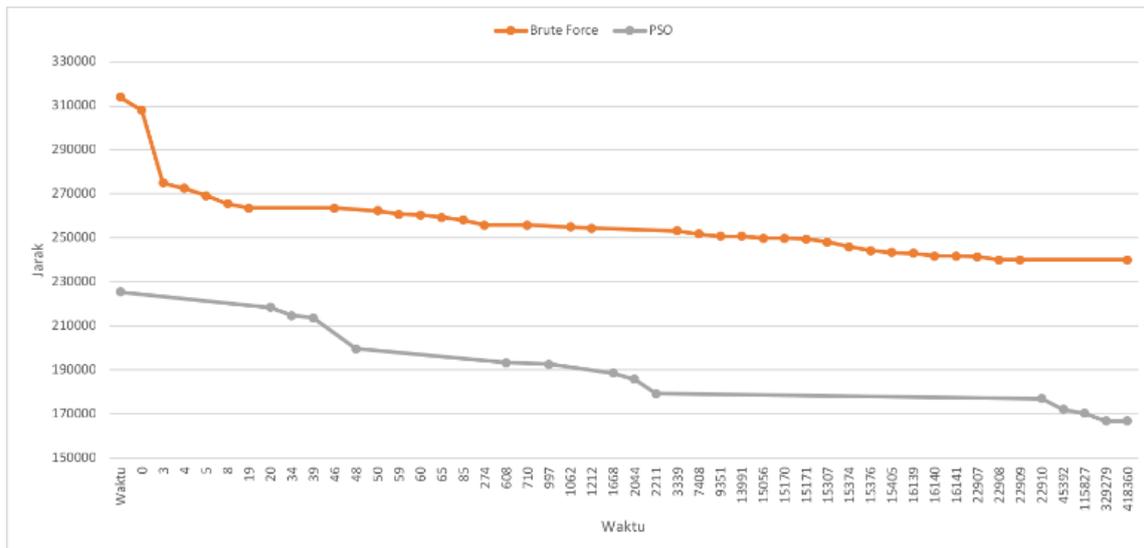
$$pop(i, j) = \begin{cases} bb & \text{jika } pop(i, j) < bb \\ ba & \text{jika } pop(i, j) > ba \end{cases}$$

$$= \begin{bmatrix} 0.8023 & 0.0732 & 0.0392 & 0.5178 & 0.9350 \\ 0.4242 & 0.0591 & 0.9463 & 0.9942 & 0.4795 \\ 0.7289 & 0.9102 & 0.7637 & 0.8549 & 0.2138 \\ 0.0460 & 0.5922 & 0.5269 & 0.2927 & -0.3320 \\ 0.2728 & 0.2815 & -0.0174 & 0.0064 & -0.0586 \\ 0.09 & 0.09 & 0.09 & 0.09 & 0.09 \end{bmatrix}$$

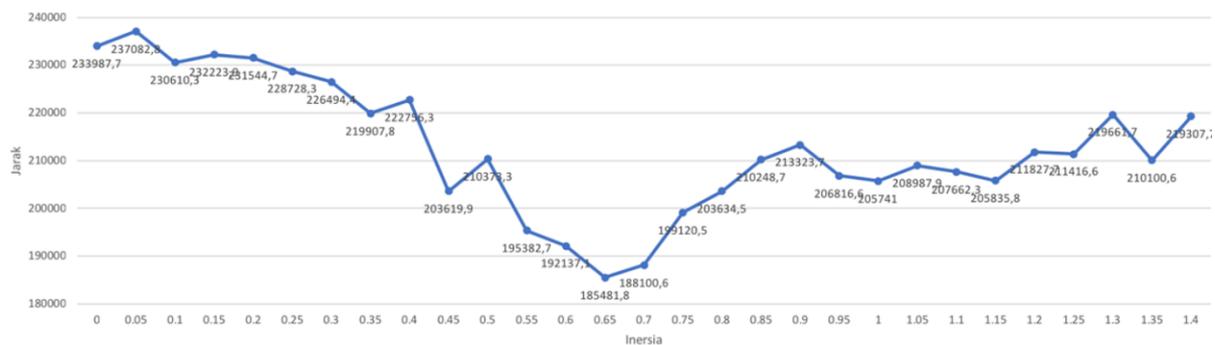
$$pop_1 = \begin{bmatrix} 0.8483 & 0.6654 & 0.5661 & 0.8105 & 0.603 \\ 0.697 & 0.3406 & 0.9289 & 1 & 0.4209 \\ 0.8189 & 0.9089 & 0.8537 & 0.9449 & 0.3038 \end{bmatrix}$$

Kemudian cek apakah solusi sudah konvergen dimana nilai  $x$  saling dekat dan cek apakah iterasi sudah melebihi *epoch*. Jika belum maka lanjutkan iterasi ke langkah langkah dimulai dari penentuan *local best* dan *global best*.

Hasil yang diperoleh dari penelitian ini perbandingan waktu dengan jarak antara metode PSO dengan Bruto Force dapat dilihat pada Gambar 3, dan Gambar 4 merupakan hasil uji coba parameter Inersia yang dilakukan dengan maximum 10000 iterasi.



Gambar 3. Hasil perbandingan waktu dengan jarak.



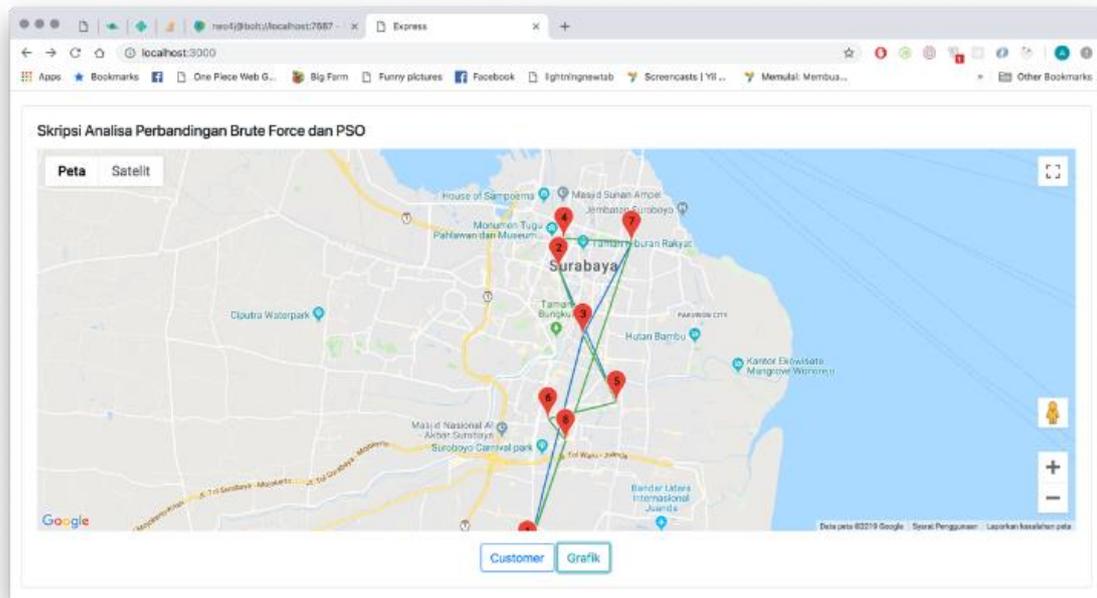
Gambar 4. Nilai parameter Inersia ( $w$ ) dengan jarak Algoritma PSO

Analisa hasil yang diperoleh pada penelitian ini sesuai dengan penelitian [5][6], bahwa metode PSO mampu untuk menyelesaikan permasalahan TSP dengan hasil yang lebih baik. Hasil yang lebih baik ditunjukkan dengan waktu dan akurasi (jarak minimal) dibandingkan dengan metode brute force atau GA.

### 3.2 Implementasi Map Berbasis Website

Setelah menganalisa kinerja antara PSO dan Brute Force, langkah selanjutnya adalah implementasi siste ke dalam aplikasi berbasis web. Aplikasi ini memberikan informasi peta

dari Google Map sebagai visualisasi dan rute grafik untuk hasil visual hasil rute terpendek secara real time. Gambar 1 merupakan hasil visual implementasi peta. Implementasi dilakukan dengan membuat penanda lokasi pelanggan berdasarkan alamat dan longitude latitude yang diperoleh dari google map API. Hasil pencarian rute terpendek dari PSO dan Brute Force akan ditampilkan semua pada Google Map. Contoh pencarian rute terpendek pada gambar 3 ditunjukkan berupa garis berwarna biru untuk PSO sedangkan warna hijau untuk Brute Force.



Gambar 5. Visualisasi Rute Terpendek (PSO dan Brute Force)

#### IV. PENUTUP

Tujuan penelitian ini adalah untuk menemukan rute terpendek dari pengiriman barang pada PT.X, dimana dalam pengiriman tersebut memuat beberapa lokasi tujuan atau dalam permasalahan teori Graph disebut dengan TSP. Sesuai dengan hasil penelitian yang dilakukan Algoritma Particle Swarm Optimization (PSO) mampu menyelesaikan permasalahan TSP dengan hasil Total Jarak yang kompetitif. Pada proses pencarian rute dengan jarak terpendek dengan berbagai node, node dengan jumlah 5 dan 10 menghasilkan jarak yang sama saat diproses menggunakan metode Brute Force dan Particle Swarm Optimization. Node dengan jumlah 15 hingga 30, dengan mempertimbangkan kompleksitas komputasi dan batasan waktu, PSO memberikan rute dengan jarak lebih pendek dibandingkan dengan Brute Force.

#### DAFTAR PUSTAKA

- [1] B. Ahmadi dan D. Jayawati, "Rancang Bangun Decision Support System Untuk Pemilihan Rute Pengiriman Paket Pada Perusahaan Penyedia Jasa Logistik", *Jurnal Manajemen Industri dan Logistik*, Vol.1, No. 2, 2018.
- [2] M. Hasibuan dan Lusiana, "Pencarian Rute Terbaik Pada Travelling Salesman Problem (TSP) Menggunakan Algoritma Genetika pada Dinas Kebersihan dan Pertamanan kota Pekanbaru", *Jurnal Sains dan Teknologi Informasi*, Vol. 1, No.1, 2015.
- [3] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the 1995 IEEE International Conference on Neural Network*, pp. 1942–1948, 1995.
- [4] E. Baidoo and S. O. Oppong, "Solving the TSP using Traditional Approach", *International Journal of Computer Application*, Vol. 152, No. 8, 2016.
- [5] X. Yan, C. Zhang, W. Lou, W. Li, W. Chen, dan H. Liu, "Solve Traveling Salesman Problem Using Particle Swarm Optimization Algorithm", *International Journal of Computer Science*, Vol. 9, No.2, 2012.
- [6] M. Panda, "Performance Comparison of Genetic Algorithm, Particle Swarm Optimization & Simulated Annealing Applied to TSP", *Applied Journal of Applied Engineering Research*, Vol.13, No. 9, 2018.
- [7] M. Kurniawan dan N. Suciati, "Premise Parameter Optimization on Adaptive Network Based Fuzzy Inference System Using Modification Hybrid Particle Swarm Optimization and Genetic Algorithm", Vol. 22, No. 2, 2018.

- [9] M. Kurniawan and N. Suciati, "177-460-2-RV (1)," *INTEGER J. Inf. Technol.*, vol. 2, pp. 31–40, 2017.
- [10] K.A.F.A. Samah, N. Sabri, R. Hamzah, R. Roslan, N.A. Mangshor, A.A.M. Asri, "Brute Force Algorithm Implementation for Traveljoy Travelling Recommendation System", *Indonesian Journal of Electrical Engineering and Computer Science*, Vol.16, No.2, 2019.
- [11] B. Santosa dan P. Willy, "Metode Heuristik konsep dan Implementasi", Guna Widya, 2011.