

BAB II

KAJIAN PUSTAKA

2.1 *Travelling Salesman Problem* (TSP)

2.1.1 pengertian *Travelling Salesman Problem* (TSP)

Travelling Salesman Problem (TSP) dikemukakan pada tahun 1800 oleh matematikawan Irlandia, William Rowan Hamilton dan matematikawan Inggris, Thomas Penyngton. TSP dikenal sebagai suatu permasalahan optimasi yang bersifat klasik dan dimana tidak ada penyelesaian yang paling optimal selain mencoba seluruh kemungkinan penyelesaian yang ada. Permasalahan ini melibatkan seorang *Travelling Salesman* yang harus melakukan kunjungan sekali pada semua kota dalam sebuah lintasan sebelum dia kembali ketitik awal, sehingga perjalanannya dikatakan sempurna.

Travelling Salesman Problem (TSP) adalah permasalahan umum dalam optimasi kombinatorial dimana seorang salesman harus mengunjungi sejumlah N kota, disyaratkan setiap kota hanya dikunjungi sekali. Salesman ini harus memilih rute sehingga jarak total yang dia tempuh minimum (Budi Santosa, 2017)

Menurut Smith, dalam jurnal Utomo, dkk (2004) *Traveling Salesman Problem* (TSP) dapat dengan mudah diubah dalam bentuk *network problem* dengan formulasi yang serupa dengan model rute terpendek. Konsumen yang dikunjungi diidentifikasi sebagai simpul-simpul (*node*) dari jaringan. Sedangkan menurut Rabi', Persoalan *Travelling Salesman* (TSP) adalah persoalan optimasi yang dinyatakan sebagai mencari rute perjalanan termurah untuk mengunjungi *node* (konsumen), dimana setiap konsumen dikunjungi secara pasti satu kali.

Bahwa penentuan rute perjalanan merupakan salah satu permasalahan yang sering dihadapi dalam kehidupan sehari - hari. Salah satu contoh yaitu rute manakah yang memiliki biaya paling murah untuk dilalui seorang *salesman* ketika harus mengunjungi sejumlah daerah. Tiap daerah tersebut harus dikunjungi tepat satu kali kemudian kembali lagi ke tempat semula. Permasalahan tersebut dikenal sebagai *Traveling Salesman Problem* (TSP) yaitu mencari rute terpendek dengan syarat kendaraan berawal dan berakhir di depo yang sama dan setiap kota dikunjungi tepat satu kali.

Berikut adalah aturan-aturan yang mengidentifikasi bahwa permasalahan tersebut adalah TSP:

1. Perjalanan dimulai dan diakhiri di kota yang sama sebagai kota asal sales.
2. Seluruh kota harus dikunjungi tanpa satupun kota yang terlewatkan.
3. Sales tidak boleh kembali ke kota asal sebelum seluruh kota dikunjungi.

2.1.2 Model Matematika TSP

Formulasi matematika permasalahan TSP adalah sebagai berikut :

$$\text{Minimasi } z = \sum_{i=1}^n \sum_{j=1}^n C_{ij} X_{ij}, (1)$$

Dibatasi oleh :

$$\sum_{i=1}^n X_{ij} = 1, j = 1, 2, \dots, n, (2)$$

$$\sum_{j=1}^n X_{ij} = 1, i = 1, 2, \dots, n, (3)$$

$$X_{ij} \in \{0,1\}, i, j = 1, 2, \dots, n \quad (4)$$

\tilde{x} adalah bentuk Hamilton cycle (5)

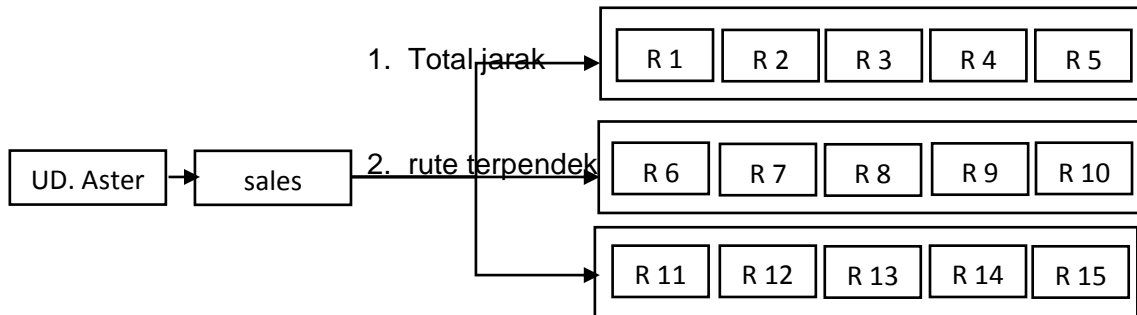
1. Dimana, C_{ij} adalah indeks biaya dan X_{ij} adalah variabel keputusan yang berkaitan dengan penugasan elemen i ke posisi j . Jika $X_{ij} = 1$, maka elemen i dihubungkan dengan elemen j , yang berarti terbentuk rute dari kota i ke kota j . Fungsi tujuan z pada persamaan.
2. merepresentasikan biaya total yang harus diminimasi. Pembatas pada persamaan.
3. memastikan bahwa setiap posisi j diisi hanya oleh satu kota. Pembatas.
4. menjamin bahwa setiap kota i ditugaskan tepat pada satu posisi. Pembatas.
5. memastikan bahwa X_{ij} merupakan anggota dari himpunan biner, 0 atau 1. Pembatas.
6. memastikan bahwa rute setiap kota hanya dikunjungi sekali.

2.1.3 Karakteristik sistem

Karakteristik sistem *Travelling Salesman Problem* (TSP) dengan menggunakan metode Particle Swarm Optimization (PSO) pada UD. Aster yang mempunyai satu sales dengan 15 *retailer* tersaji pada gambar 2.1 dimana pada saat proses pengiriman dari perusahaan ke *retailer* ada beberapa hal yang perlu diketahui :

1. Jarak dari perusahaan ke retailer
2. Waktu tempuh pada waktu proses pengiriman
3. Biaya perjalanan sales
4. Kapasitas kendaraan

5. Jumlah retailer 15
6. Pembayaran produk bolu aster secara tunai
7. Rute terpendek



Gambar 2.1 Karakteristik sistem

Keterangan :

Dimana R – 1 sampai dengan R – 15 adalah retailer dari UD. Aster

2.2 Optimasi

2.2.1 Pengertian Optimasi

Menurut Budi Santosa (2017) Optimasi memegang peran penting dalam mendesain suatu sistem, dengan optimasi, bisa menghasilkan ongkos yang lebih murah atau profit yang tinggi, dan sangat diperlukan bantuan *software* untuk menyelesaikan permasalahan yang ditemukan untuk mendapatkan solusi yang optimal dengan waktu komputasi yang tidak terlalu lama. Aplikasi dari teknik optimasi telah menjamur diberbagai bidang secara cepat. Keberhasilan penerapan teknik optimasi, paling tidak memerlukan tiga syarat. Syarat-syarat tersebut adalah kemampuan membuat model matematika dari permasalahan yang dihadapi, pengetahuan teknik optimasi dan pengetahuan akan program komputer. Pengertian optimasi ialah suatu kumpulan formula matematis dan

metoda numerik untuk menemukan dan mengidentifikasi kandidat terbaik dari sekumpulan alternatif tanpa harus secara eksplisit menghitung dan mengevaluasi semua alternatif yang mungkin.

Masalah optimasi biasanya dinyatakan dalam bentuk fungsi matematik. Optimasi adalah proses memaksimalkan atau meminimasi suatu fungsi tujuan dengan tetap memperhatikan pembatas yang ada. Suatu fungsi didefinisikan sebagai suatu aturan yang menugaskan setiap pilihan nilai x dengan satu nilai unik $y=f(x)$. Dalam hal ini x adalah variabel independent dan y adalah variabel dependent.

2.2.2 beberapa contoh penerapan optimasi

1. penjadwalan

Suatu perusahaan mendapatkan order untuk mengerjakan suatu proyek. Proyek ini terdiri dari 10 *job* dan dikerjakan di saty mesin. Masing-masing *job* mempunyai waktu operasi o_j due date d_j dan pinalti P_j jika terjadi keterlambatan (tardiness). Permasalahannya, bagaimana urutan *job-job* ini harus dikerjakan sehingga meminimumkan total keterlambatan? Dalam hal ini, minimasi total keterlambatan sering disebut dengan fungsi tujuan. Yaitu, sesuatu yang dijadikan sebagai ukuran apakah masalah diselesaikan dengan baik atau tidak. Dalam hal ini minimasi semakin kecil nilai fungsi tujuan semakin baik solusi yang didapatkan.

2. *Travelling Salesman Problem* (TSP)

TSP adalah permasalahan umum dalam optimasi kombinatorial dimana seorang salesman harus mengunjungi sejumlah N kota, disyartakan setiap kota hanya

dikunjungi sekali. Salesman ini harus memilih rute sehingga jarak total yang dia tempuh minimum (Budi Santosa, 2017)

2.3 Distribusi

Pengertian Distribusi Dalam jurnal Puteri, dkk Distribusi adalah sebuah aktifitas yang sering orang lain lakukan, baik dalam bidang bisnis maupun non bisnis di kehidupan sehari - harinya. Menyampaikan produk ke pelanggan merupakan salah satu aktivitas dari pendistribusian barang. (Hahury, 2010). Proses distribusi di bagi menjadi 2 diantaranya : (Anonim, 2014)

1. Distribusi Langsung

Proses yang dilakukan dari produsen untuk pelanggan secara langsung dan tidak membutuhkan perantara lain.

2. Distribusi Tidak Langsung

Proses yang dilakukan oleh seseorang yang mempunyai tanggung jawab pada produsen untuk menyalurkan barang ke pelanggan, adanya orang ketiga ataupun perantara.

2.4 Algoritma Particle Swarm Optimization

Pengertian Algoritma Particle Swarm Optimization Menurut Budi Santosa (2017), PSO, didasarkan pada perilaku sebuah burung atau ikan. Algoritma PSO meniru perilaku sosial organisme ini. Perilaku sosial terdiri dari tindakan individu dan pengaruh dari individu-individu lain dalam suatu kelompok. Kata partikel menunjukkan, misalnya, seekor burung dalam kawanan burung. Setiap individu atau partikel berperilaku dengan cara menggunakan kecerdasannya (*intelligence*) sendiri dan dipengaruhi perilaku kolektifnya. Dengan demikian, jika satu partikel

atau seekor burung menemukan jalan yang tepat atau pendek menuju sumber makanan, sisa kelompok yang lain juga akan dapat segera mengikuti jalan tersebut meskipun lokasi mereka jauh di kelompok tersebut. *“Perilaku seekor hewan dalam kawanan (swarm) dipengaruhi perilaku individu dan juga kelompoknya”*.

Dalam *particle swarm optimization* (PSO), kawanan diasumsikan mempunyai ukuran tertentu dengan setiap partikel posisi awalnya terletak disuatu lokasi yang acak dalam ruang multidimesi. Setiap partikel bergerak dalam ruang/space tertentu dan mengingat posisi terbaik yang pernah dilalui atau ditemukan terhadap sumber makanan atau nilai fungsi objektif. Setiap partikel menyampaikan informasi atau posisi dan kecepatan masing-masing berdasarkan informasi yang diterima mengenai posisi tersebut. Sebagai contoh, misalnya perilaku burung-burung dalam kawanan burung. Meskipun setiap burung mempunyai keterbatasan dalam hal kecerdasan, biasanya ia akan mengikuti kebiasaan (rule) seperti berikut :

1. Seekor burung tidak berada terlalu dekat dengan burung yang lain
2. Burung tersebut akan mengarah terbangnya ke arah rata-rata keseluruhan burung.
3. Akan memposisikan diri dengan rata-rata posisi burung yang lain dengan menjaga sehingga jarak antar burung dalam kawanan itu tidak terlalu jauh

Dengan demikian perilaku kawanan burung akan didasarkan pada kombinasi dari 3 faktor berikut :

1. Kohesi – terbang bersama
2. Separasi - jangan terlalu dekat

3. Penyesuaian (*alighment*) – mengikuti arah bersama

Jadi PSO dikembangkan dengan berdasarkan pada model berikut :

1. Ketika burung mendekati target atau makanan (atau bisa minimum atau maximum suatu fungsi tujuan) secara cepat mengirim informasi kepada burung-burung yang lain dalam kawanan tertentu
2. Burung yang lain akan mengikuti arah menuju ke makanan tetapi tidak secara langsung
3. Ada komponen yang tergantung pada pikiran setiap burung, yaitu memorinya tentang apa yang sudah dilewati pada waktu sebelumnya

Pada algoritma PSO ini, pencarian solusi dilakukan oleh suatu populasi dari beberapa partikel. Populasi dibangkitkan secara random dengan batasan nilai terkecil dan terbesar. Setiap partikel merepresentasikan posisi atau solusi dari permasalahan yang dihadapi. Setiap partikel melakukan pencarian solusi yang optimal dengan melintasi ruang pencarian (*local space*). Hal ini dilakukan dengan cara setiap partikel melakukan penyesuaian terhadap posisi terbaik dari partikel tersebut (*local best*) dan penyesuaian terhadap posisi terbaik dari seluruh kawanan terbaik (*global best*) selama melintasi ruang pencarian. Jadi penyebaran atau informasi didalam partikel itu sendiri dan antara suatu partikel dengan partikel terbaik dari seluruh kawanan selama proses pencarian solusi. Setelah itu, dilakukan proses pencarian untuk mencari posisi terbaik setiap partikel dalam sejumlah iterasi tertentu semua didapatkan posisi yang relatif steady atau mencapai batas iterasi yang telah ditetapkan. Pada setiap iterasi, setiap solusi yang di representasikan oleh posisi partikel, dievaluasi

performansinya dengan cara memasukkan solusi tersebut kedalam *fitness function*.

Setiap partikel diperlakukan seperti sebuah titik pada suatu dimensi ruang tertentu. Kemudian terdapat dua faktor yang memberikan karakter terhadap status partikel pada ruang pencarian yaitu posisi partikel dan kecepatan partikel [Kennedy and Eberhart, 1995]

Berikut ini merupakan formulasi matematika yang menggambarkan posisi dan kecepatan partikel pada suatu dimensi ruang tertentu :

$$x_i(t) = x_{i1}(t)x_{i2}(t) \dots, x_{iN}(t) \quad [2.1]$$

$$v_i(t) = v_{i1}(t)v_{i2}(t) \dots, v_{iN}(t) \quad [2.2]$$

Dimana :

X = posisi partikel

V = kecepatan partikel

i = indeks partikel

T = iterasi ke-t

N = Ukuran dimesi ruang

Berikut ini merupakan model matematika yang menggambarkan mekanisme updating status partikel. [Kennedy and Eberhart 1995]

$$V_i(t) = V_i(t-1) + c_1 r_1 (X_i^l - X_i(T-1)) + c_2 r_2 (X^G - X_i(t-1)) \quad [2.3]$$

$$X_i(t) = V_i(t) + X_i(t-1) \quad [2.4]$$

Dimana

$X_i^l = X_{i1}^l, X_{i2}^l, \dots, X_{iN}^l$ merepresentasikan *local best* dari partikel ke-i, sedangkan $X^G = X_{i1}^G, X_{i2}^G, \dots, X_{iN}^G$ merepresentasikan *global best* dari seluruh kawan. Sedangkan c_1 dan c_2 adalah suatu konstanta yang bernilai positif yang biasanya disebut sebagai *learning factor*. Kemudian r_1 dan r_2 adalah suatu bilangan random yang bernilai antara 0 sampai 1. Persamaan [2.3] digunakan untuk menghitung kecepatan sebelumnya, jarak antara posisi saat ini dengan posisi terbaik partikel (*local best*), dan jarak antara posisi saat ini dengan posisi terbaik kawan (*global best*). Kemudian partikel terbang menuju posisi yang baru berdasarkan persamaan [2.2]. setelah algoritma PSO ini dijalankan dengan sejumlah iterasi tertentu hingga mencapai kriteria pemberhentian, maka akan didapatkan solusi yang terletak pada *global best*.

Langkah-langkah utama dari algoritma PSO adalah sebagai berikut :

1. Bangkitkan posisi awal sejumlah partikel sekaligus kecepatan awalnya secara random.
2. Evaluasi *fitness* dari masing – masing partikel berdasarkan posisinya.
3. Tentukan partikel dengan *fitness* terbaik dan di tetapkan sebagai *Gbest*.
Untuk setiap partikel, *Pbest* awal akan sama dengan posisi awal.

Ulangi langkah – langkah berikut sampai stopping criteria dipenuhi

1. Menggunakan *Pbest* dan *Gbest* yang ada, perbarui kecepatan setiap partikel menggunakan persamaan [2.3]. Lalu dengan kecepatan baru yang didapat, perbarui posisi setiap partikel menggunakan persamaan [2.4]
2. Evaluasi nilai *fitness* dari setiap partikel

3. Tentukan partikel dengan fitness terbaik, dan tetapkan sebagai G_{best} . Untuk setiap partikel, tentukan P_{best} dengan membandingkan posisi sekarang dengan P_{best} dari iterasi sebelumnya.
4. Cek stopping kriteria, jika dipenuhi, berhenti. Jika tidak, kembali ke 1

2.5 Prosedur PSO Untuk TSP

1. Lakukan inisialisasi : jumlah partikel, rute awal, jumlah iterasi, kecepatan awal, ba , bb
2. Evaluasi jarak total dari masing – masing rute yang dihasilkan dari setiap partikel berdasarkan matrik jarak
3. Tentukan nilai partikel dengan jarak total terkecil, dan tetapkan partikel ini sebagai G_{best} . Untuk setiap partikel, gunakan nilai awalnya sebagai P_{best} .

Ulangi langkah berikut sampai *stopping* kriteria dipenuhi

1. Menggunakan P_{best} dan G_{best} yang ada, perbarui kecepatan setiap partikel menggunakan persamaan [2.3] lalu dengan kecepatan baru yang didapat, perbarui nilai dari setiap partikel menggunakan persamaan [2.4]
2. Jika nilai elemen setiap partikel diluar batas yang diijinkan, yaitu antara bb dan ba , maka lakukan penyesuaian,

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

3. Evaluasi jarak dan setiap rute yang dihasilkan oleh setiap partikel
4. Tentukan partikel dengan jarak total minimum, dan tetapkan partikel yang bersangkutan sebagai G_{best} . Untuk setiap partikel, tentukan P_{best} dengan membandingkan partikel sekarang dengan P_{best} dari iterasi sebelumnya berdasarkan jarak total minimumnya.

5. Cek stopping kriteria. Jika dipenuhi, berhenti. Jika tidak, kembali ke 1.

2.6 software matlab

Matlab merupakan perangkat lunak yang digunakan untuk pemrograman, analisis serta komputasi teknis dan matematis berbasis matriks. *matlab* adalah singkatan dari *Matrix Laboratory* karena mampu menyelesaikan masalah perhitungan dalam bentuk matriks. *Matlab* versi pertama dirilis pada tahun 1970 oleh Cleve Moler. Pada awalnya *matlab* didesain untuk menyelesaikan masalah-masalah persamaan aljabar linear. Seiring berjalannya waktu, program ini terus mengalami perkembangan dari segi fungsi dan performa komputasi.

Bahasa pemrograman yang kini dikembangkan oleh *MathWorks Inc.* menggabungkan proses pemrograman, komputasi dan visualisasi melalui lingkungan kerja yang mudah digunakan. *Matlab* juga memiliki keunggulan umum lainnya seperti analisis dan eksplorasi data, pengembangan algoritma, pemodelan dan simulasi, visualisasi plot dalam bentuk 2D dan 3D, hingga pengembangan aplikasi antar muka grafis. Dalam ruang lingkup perguruan tinggi, *matlab* digunakan sebagai alat pembelajaran pemrograman matematika, teknik dan sains pada level pengenalan dan lanjutan.

Matlab dapat dioperasikan pada sistem operasi *Windows*, *Linux* maupun *macOS*. Selain itu, *matlab* juga bisa dihubungkan dengan aplikasi atau bahasa pemrograman eksternal lainnya seperti *C*, *Java*, *.NET* dan *Microsoft Excel*. Dalam *matlab* tersedia pula kotak kakas (*toolbox*) yang dapat digunakan untuk aplikasi-aplikasi khusus seperti pengolahan sinyal, sistem kontrol, logika *fuzzy*,

jaringan saraf tiruan, optimasi, pengolahan citra digital, bioinformatika, simulasi dan berbagai teknologi lainnya.

Pemilihan *software* atau perangkat lunak *matlab* untuk membantu penelitian dalam menganalisis dari penemuan pada penelitian ini dikarenakan *software* ini memiliki *feature* yang diperlukan dalam penelitian, seperti *feature* Simulink yang berguna dalam sistem kendali dalam pembuatan blok diagram dan terdapat *m.file* yang digunakan untuk pengkodean dari algoritma metode *particle swarm optimization* (PSO) dan keduanya dapat digunakan secara bersamaan. Selain itu, dapat memperlihatkan grafik yang memberi informasi bahwa hasil pencarian solusi sudah konvergen atau belum konvergen.

Software MATLAB yang memiliki *feature* yang cukup lengkap, tentu saja memerlukan komputer atau laptop yang memiliki spesifikasi atau *system requirement* agar *software* ini dapat bekerja. Spesifikasi yang harus dimiliki oleh komputer atau laptop antara lain memiliki *processor* minimal dual-core 2.1 Ghz, OS windows XP/7 (32 bit dan 64 bit), RAM 1 GB, dan *free hard disk* 1GB.

2.7 Posisi Penelitian

Nama Peneliti	Tahun	Judul	Permasalahan	Metode	<i>Critical review</i>
Kusrini, Jazi Eko Istiyanto	2007	Penyelesaian Traveling Salesman Problem dengan algoritma cheapest	Bagaimana untuk menyelesaikan kasus Travelling Salesman	cheapest insertion heuristisch	Berdasarkan dari jurnal tersebut dengan menggunakan metode <i>Cheapest</i>

		insertion heuristisch dan basis data	Problem dengan menggunakan algoritma CIH.		<i>Insertion Heuristic</i> dapat menghasilkan rute yang lebih pendek akan tetapi sebaiknya dilakukan pengujian ulang dengan menggunakan metode algoritma yang lain.
Fatmawati, Bayu Prihandono, Evi Noviani	2015	Penyelesaian Travelling Salesman Problem dengan metode tabu search	mencari rute baru dari rute awal yang dihasilkan pada tahap sebelumnya dengan cara melakukan <i>neighborhood search</i> menggunakan aturan kombinasi.	Metode Tabu search	Di lakukan percobaan perhitungan sampai literasi ke 60, dan pada literasi ke 20 jarak tempuh dan waktu perjalanan paling minimum.
Shah Khadafi	2016	Implementasi algoritma PSO untuk probabilitas	Bagaimana untuk mendapatkan rute tercepat	<i>Particle swarm Optimization</i>	Seorang kurir Dalam pengiriman paket di

		urutan pengiriman paket pengantaran kurir	dalam pengiriman beberapa paket	(PSO)	batasi dengan waktu yang singkat tetapi tidak ada keterangan batasan waktu dalam jurnal tersebut dan Tidak adanya langkah-langkah dalam menggunakan metode PSO
Rinindya Nurtiara Puteri, dkk	2017	Optimasi Multiple Travelling Salesman Problem Pada Pendistribusian Air Minum Menggunakan Algoritme Particle Swarm Optimization	Pendistribusian yang kurang optimal menghambat proses pengiriman air minum ke tempat pelanggan	<i>Particle swarm Optimization</i> (PSO)	Didalam jurnal tersebut mempunyai 30 toko konsumen yang tersebar di kota malang tetapi tidak ada data alamat toko, hanya diketahui keterangan total jarak tempuh untuk

					ke empat sales
Henri fachrudin		Penyelesaian Permasalahan Travelling Salesman Problem (TSP) Pada UD. ASTER	Bagaimana cara untuk menentukan rute sales ke konsumen dengan total jarak tempuh paling minimal	<i>Nearest insertion</i>	Di dalam penelitiannya masih menggunakan pendekatan metode heuristik, dimana dalam mengerjakannya membutuhkan waktu yang lumayan lama.