

BAB 2

TINJAUAN PUSTAKA

2.1. OPTIMASI

Menurut Kamus Besar Bahasa Indonesia (Optimasi), optimasi diartikan sebagai optimalisasi, yaitu proses, cara, dan kreasi untuk menghasilkan hal-hal terkini. Sedangkan optimasi juga berasal dari kamus bahasa Inggris yaitu *Optimization* yang artinya optimasi. Menurut Suprodjo dan Purwandi, 1982, Tarmizi, 2005, Optimasi matematis adalah suatu teknik untuk mencapai harga ekstrim maksimum atau minimum untuk suatu fungsi tertentu dengan beberapa faktor pembatas (Ekonomi Manajemen). Menurut Sugioko (2013 : 113), Optimasi merupakan sebuah bidang ilmu dalam matematika yang berfokus pada pencarian nilai minimum atau maksimum secara terencana dari suatu fungsi, peluang, atau pencarian nilai lainnya dalam berbagai situasi. Dalam beberapa konteks, optimasi juga bisa mengacu pada upaya untuk meningkatkan kinerja sehingga dicapai kualitas terbaik dan hasil kerja terbaik. Secara matematis, optimasi adalah proses mencari nilai ekstrim suatu fungsi, baik maksimum maupun minimum, dengan mempertimbangkan batasan-batasannya.

Optimasi secara umum berarti memaksimalkan atau mengoptimalkan sesuatu dengan tujuan mengelola sesuatu yang telah dilakukan. Oleh karena itu, optimasi merupakan kata benda yang berasal dari kata kerja, dan bergantung pada tujuan yang perlu dimaksimalkan, optimasi dapat dianggap sebagai ilmu dan seni. Contoh masalah yang dimaksimalkan adalah masalah keuntungan, dan contoh masalah yang diminimalkan adalah masalah biaya, persediaan, dan sebagainya. Kendala yang sering ditemui antara lain terbatasnya bahan baku, terbatasnya tenaga kerja, dan lain-lain. Kendala tersebut dapat dinyatakan dalam bentuk himpunan persamaan linier atau pertidaksamaan variabel. Oleh karena itu, fungsi yang dioptimalkan adalah solusi dengan nilai fungsi tujuan yang diinginkan. Nilai yang diinginkan adalah nilai maksimum yaitu nilai maksimum fungsi tujuan, dan nilai minimum yaitu nilai minimum fungsi tujuan.

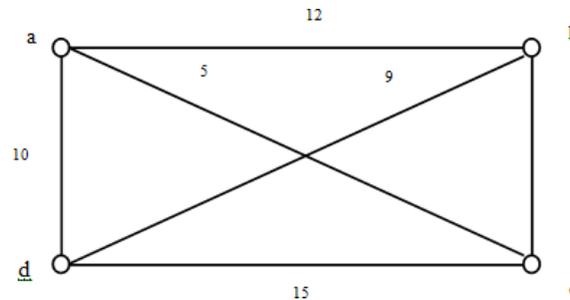
2.2. TEORI ROUTING DALAM ALGORITMA GENETIKA

Teori *routing* dalam algoritma genetika merupakan pendekatan yang memadukan konsep-konsep dari genetika dan evolusi biologis untuk menemukan solusi optimal atau yang mendekati optimal pada masalah *routing* seperti *Traveling Salesman Problem* (TSP) atau *Vehicle Routing Problem* (VRP). Dalam algoritma genetika, solusi diwakili sebagai kromosom atau individu yang terdiri dari berbagai gen yang bisa mewakili urutan kota dalam TSP atau urutan pengiriman dalam VRP (Yusron Mubarak and Chotijah 2021). Algoritma genetika memperbaiki populasi solusi dengan operasi genetik seperti mutasi, seleksi, dan *crossover*. Seleksi digunakan untuk memilih individu terbaik sebagai orangtua dalam reproduksi berikutnya. *Crossover* menggabungkan informasi dari dua individu untuk menciptakan solusi baru, sementara mutasi mengubah gen secara acak untuk mencegah stagnasi dalam pencarian (Muhandhis et al. 2023). Setiap individu dievaluasi berdasarkan kriteria kecocokan yang sesuai dengan masalah *routing* yang dihadapi. Proses ini berulang secara iteratif sampai mencapai kondisi berhenti yang ditentukan, seperti jumlah generasi atau mencapai solusi yang memenuhi kriteria tertentu. Melalui pendekatan ini, algoritma genetika dapat menemukan solusi yang optimal atau mendekati optimal untuk masalah *routing* dengan menggunakan konsep evolusi biologis secara efisien.

2.3. TRAVELING SALESMAN PROBLEM

Traveling Salesman Problem (TSP) adalah salah satu masalah paling populer dalam optimasi kombinatorial dan telah diterapkan pada masalah perencanaan kendaraan, penjadwalan, penempatan, dan perutean. TSP adalah masalah rute perjalanan ketika mengunjungi banyak kota dan setiap kota dimulai dan diakhiri di kota yang sama dengan jarak perjalanan yang paling sedikit. (Bryan 2000). TSP memiliki beberapa karakteristik, semua kota harus dilintasi tanpa kecuali, tidak dapat kembali ke kota awal jika masih ada kota yang belum dilintasi, dan hanya dapat dilintasi sekali (Sutanto, Soebroto, and Cholissodin 2023). Merencanakan jarak total minimum adalah masalah TSP. Karena ruang pencarian dimulai dari himpunan permutasi beberapa kota tertentu, penyelesaian

masalah ini tidak mudah. Akibatnya, TSP dianggap sebagai masalah non-polinomial. Gambaran sederhana dari pengertian TSP adalah :



Gambar 2.1 Posisi kota-kota yang akan dilewati

Permasalahan TSP dapat diselesaikan dengan menghitung $\frac{(n-1)!}{2}$, dimana n adalah jumlah kota dan memilih rute dengan panjang terpendek. Seperti terlihat pada Gambar 2.1, lokasi kota yang terdiri dari empat kota (a, b, c, d) mempunyai $\frac{(4-1)!}{2} = 3$ rute yang dapat dilalui, adalah :

R1 = Panjang rute (a, b, c, d, a) atau (a, d, c, b, a) adalah $12 + 8 + 15 + 10 = 45$

R2 = Panjang rute (a, c, d, b, a) atau (a, b, d, c, a) adalah $5 + 15 + 9 + 12 = 41$

R3 = Panjang rute (a, c, b, d, a) atau (a, d, b, c, a) adalah $5 + 8 + 9 + 10 = 32$

Jadi, rute terpendek adalah R3 = (a, c, b, d, a) atau (a, d, b, c, a) dan panjang rutenya adalah $5 + 8 + 9 + 10 = 32$. Ini adalah solusi permasalahan TSP dari Gambar 2.1. Oleh karena itu kita dapat menyimpulkan bahwa masalah dari TSP pada dasarnya terdiri dari menemukan rute paling optimal untuk diambil, dengan syarat setiap kota dikunjungi sebanyak satu kali (Wijanarko et al. 2021).

2.4. ALGORITMA GENETIKA

Algoritma Genetika terinspirasi oleh prinsip-prinsip genetika seleksi alam yang pertama kali ditemukan di Universitas Michigan, di Amerika Serikat, oleh Belanda pada tahun 1975 berkat penelitian, kemudian dipopulerkan oleh salah satu muridnya, David Golberg. Konsep dasar algoritma genetika dirancang untuk mensimulasikan proses dalam sistem alami oleh Charles Darwin, yaitu "*survival of the fit*". Teori ini menyatakan bahwa dalam suatu populasi, makhluk yang lebih

kuat akan mengalahkan makhluk yang lebih lemah dalam perlombaan untuk memperoleh sumber daya alam yang langka (Sutojo, dkk. 2011).

Algoritma Genetika merupakan suatu metode optimasi yang berasal dari gagasan tentang proses seleksi alam. Algoritma genetika diawali dengan menyajikan solusi dari suatu permasalahan yang ingin diselesaikan dalam suatu rangkaian kromosom yang kemudian menghitung nilai *fitness* dan memilih individu mana yang mempunyai nilai *fitness* tersebut. Algoritma genetika menerapkan langkah-langkah seleksi pada proses optimasi dengan menggabungkan beberapa pilihan secara acak dalam suatu populasi untuk memperoleh generasi baru dengan nilai *fitness* tertinggi. Generasi baru dihasilkan dengan menggunakan operator genetik dari individu yang ada dalam populasi. Representasi kromosom, seleksi, *crossover*, mutasi dan perhitungan fungsi *fitness* merupakan elemen dari algoritma genetika (Katoch, Chauhan, and Kumar 2021).

Berikut langkah-langkah penyelesaian dengan algoritma genetika :

1. Membangkitkan Populasi Awal

Bentuk awal dari populasi adalah bentuk individu yang dapat dibentuk selama proses tersebut (Sutanto, dkk. 2023). Populasi n kromosom umumnya dihasilkan secara acak dan dapat berukuran apa pun yang diinginkan, dari satu individu hingga ribuan individu.

Ada beberapa teknik untuk menghasilkan populasi awal, algoritma generator acak, beberapa pendekatan genetik dan permutasi. Semua teknik ini dapat diterapkan pada kasus TSP, termasuk algoritma generator acak. Inti dari algoritma generator acak adalah melibatkan pembangkitan bilangan acak untuk nilai setiap gen bergantung pada representasi kromosom yang digunakan. Contoh penggunaan Algoritma *Random Generator* dalam kasus TSP menurut Yulyantri adalah sebagai berikut :

Sebuah kromosom dengan sembilan kota direpresentasikan berikut :

$$\text{Kromosom A} = (X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9)$$

Kemudian bilangan acak akan dibangkitkan sebanyak jumlah kota (gen) dalam kromosom. Misal bilangan acak adalah :

$$[0.13 \ 0.25 \ 0.11 \ 0.36 \ 0.74 \ 0.55 \ 0.64 \ 0.27 \ 0.83]$$

Dengan nilai acak di atas, nilai 0.11 merupakan nilai terkecil, sehingga kota ke-3 bertempat di urutan pertama. Selanjutnya 0.13 menempati urutan kedua dan seterusnya. Dengan demikian, dari nilai acak di atas dapat ditentukan lintasan yang terjadi adalah :

$$\text{Kromosom A} = (X_3, X_1, X_2, X_8, X_4, X_6, X_7, X_5, X_9)$$

2. Nilai *Fitness*

Fitness merupakan suatu nilai yang dimiliki setiap individu untuk menilai derajat kesesuaian setiap individu terhadap standar dan sasaran (objectives) dari masalah yang ingin dicapai. Nilai *fitness* kromosom mewakili kualitas kromosom dalam suatu populasi (Pratama 2020). Nilai *fitness* dihitung dengan persamaan 2.1 berikut

$$Fitness = \frac{1}{fungsi\ objektif(i)} \quad (2.1)$$

3. Proses Seleksi

Proses seleksi merupakan proses pemilihan calon generasi baru . Induk yang baik menghasilkan anak yang baik. Semakin tinggi nilai *fitness* individu , semakin besar kemungkinannya untuk terpilih (Pitaloka and Koedijarto 2022). Kemampuan algoritma genetika untuk secara bertahap menghasilkan kromosom yang lebih baik bergantung pada tekanan seleksi yang diberikan pada populasi (Andini and Astuti 2021). Ada bermacam-macam teknik untuk melakukan proses seleksi pada suatu permasalahan tergantung pada permasalahan apa yang akan diselesaikan. Di antara teknik-teknik untuk seleksi adalah *Roulette Wheel Selection*, *Rank Base Selection* dan *Steady State Selection*.

Teknik yang digunakan pada proses seleksi ini adalah teknik *Roulette Wheel Selection*. Teknik seleksi ini merupakan yang paling sederhana. Teknik ini biasa disebut *Stochastic Sampling with Replacement*. Langkah pertama yang dilakukan dalam seleksi adalah pencarian nilai *fitness*. Untuk mencari *fitness* relatif setiap kromosom dapat dihitung menggunakan persamaan 2.2.

$$Prob = \frac{fitness(i)}{total\ fitness} \quad (2.2)$$

4. Proses *Crossover*

Crossover adalah suatu mekanisme yang digunakan algoritma genetika dengan cara menggabungkan dua kromosom untuk menghasilkan kromosom anak yang mewarisi ciri-ciri dasar dari induknya (Pitaloka and Koedijarto 2022). *Crossover* memungkinkan individu menghasilkan keturunan baru dengan mengganti informasi tertentu dari induknya. Pada penelitian ini menggunakan metode *order crossover* (OX) dikarenakan metode ini bisa mencegah adanya gen ganda pada suatu individu (Yuda and Widiartha 2020). Langkah-langkah menggunakan metode *order crossover* :

1. Acak pilih dua titik pada masing-masing *parent*.
 2. Salin segmen antara titik tersebut dari salah satu *parent* ke *offspring*.
 3. Sisipkan gen dari *parent* lain ke *offspring* di antara segmen yang disalin, hindari gen yang sama.
 4. Isi sisa posisi di *offspring* dengan gen yang belum digunakan dari *parent* lain.
 5. Pastikan tidak ada gen yang sama dalam *offspring*.
 6. Hasilkan *offspring* sebagai hasil dari proses *order crossover*
7. Proses Mutasi

Mutasi adalah proses menciptakan gen baru dengan mengubah gen individu. Proses ini serupa dengan perubahan yang terjadi pada kehidupan alami (Muhandhis et al. 2023). Kemungkinan terjadinya mutasi genetik sangat rendah, serupa dengan kejadian nyata dalam kehidupan. Oleh karena itu, ada kemungkinan terjadinya mutasi genetik, pada tingkat yang rendah.

Penelitian ini akan menggunakan metode *Mutasi Inversion* yaitu metode mutasi pembalikan yang dilakukan pada *substring* yang terletak di antara dua titik pada kromosom. Kemudian pemilihan akan dilakukan secara acak. Selanjutnya, kita akan memilih sejumlah gen tertentu dalam gen *substring* yang posisinya akan dibalik.

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

Memilih gen yang dimutasi adalah gen ke-3, gen ke-4, gen ke-5 dan gen ke-6. Sebagai contoh gen akan di urutkan dalam kromosom tersebut menjadi seperti dibawah ini :

1	2	6	5	4	3	7	8
---	---	---	---	---	---	---	---

Terjadi perubahan yaitu gen ke-3 menempati posisi gen ke-6, gen ke-4 menempati posisi gen ke-5, gen ke-5 menempati posisi gen ke-4 dan gen ke-6 menempati posisi gen ke-3.

