

ALGORITMA GENETIKA SEBAGAI SOLUSI OPTIMASI DALAM TRAVELLING SALESMAN PROBLEM

Putu Manik Prihatini

Jurusan Teknik Elektro, Politeknik Negeri Bali
Bukit Jimbaran, P.O.Box 1064 Tuban Badung – Bali
Phone:+62-361-701981, Fax:+62-361-701128
e-Mail: gek_anik@yahoo.com

Abstrak: *Travelling Salesman Problem (TSP)* merupakan permasalahan optimasi yang mempunyai konsep sederhana dan mudah dipahami. Pada TSP, optimasi yang diinginkan adalah ditemukannya jalur perjalanan terpendek (optimal) untuk melewati sejumlah kota dengan jalur tertentu sehingga setiap kota hanya terlewati satu kali dan perjalanan diakhiri dengan kembali ke kota semula. Pendekatan yang digunakan untuk menyelesaikan masalah ini adalah dengan menerapkan langkah-langkah teori evolusi yang ada pada Algoritma Genetika yang melibatkan proses pemilihan orang tua sebagai representasi dari permasalahan TSP yang akan digunakan sebagai kandidat pada proses pindah silang sehingga menghasilkan anak-anak dengan kualitas terbaik yang nantinya akan menggantikan orang tuanya sebagai populasi baru. Melalui pendekatan ini, diharapkan agar TSP dapat dipecahkan sehingga diperoleh jalur yang paling optimal untuk meminimalkan biaya yang harus dikeluarkan dalam mengunjungi sejumlah kota.

Kata Kunci: *Travelling Salesman Problem, Optimasi, Algoritma Genetika*

Genetic Algorithm As The Optimization Solution In A Travelling Salesman Problem

Abstract: *Travelling Salesman Problem (TSP)* talking about optimization, which has simple concept and easy to understand. The optimization in TSP which is to find shortest path to passed some cities with routed path in order to make sure only one city passed and ended in the starting city. To solve this problem is implementing evolution theory of Genetic Algorithm relating parent selection as the symbol of the TSP problem used as the candidate in the crossover process that will result offspring with the best quality that will replace the parent as the new population. This method will solve the TSP which consequently the optimum path to lower the cost of visiting cities.

Key words: *Travelling Salesman Problem, Optimize, Genetic Algorithm*

I. PENDAHULUAN

Manusia diciptakan dengan kecerdasan yang luar biasa. Sejak dilahirkan, manusia mengalami perkembangan kecerdasan dengan sangat pesat, baik kecerdasan kognitif, kecerdasan emosional, maupun kecerdasan spiritualnya. Sampai saat ini, belum ada satu mesin pun yang bisa menyamai kecerdasan manusia secara keseluruhan.

Artificial Intelligence atau kecerdasan buatan merupakan cabang ilmu yang berusaha memahami kecerdasan manusia. Kecerdasan buatan berusaha membangun entitas-entitas cerdas yang memiliki kemampuan meniru kecerdasan manusia. Dengan didukung perkembangan *hardware* dan *software* yang sangat beragam, kecerdasan buatan telah menghasilkan banyak produk yang sangat penting dan berguna bagi kehidupan manusia.

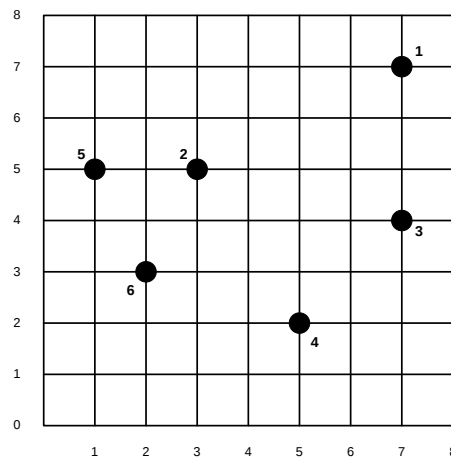
Produk-produk tersebut dikelompokkan ke dalam empat teknik, yaitu: *searching*, *reasoning*, *planning* dan *learning* [5].

Pencarian solusi masalah dengan menggunakan teknik *searching*, *reasoning* dan *planning*, membutuhkan pendefinisian aturan yang benar dan lengkap untuk masalah tersebut. Tetapi, pada masalah tertentu, terkadang aturan-aturan tersebut tidak dapat didefinisikan karena data-data yang diperlukan tidak lengkap. Melalui teknik *learning*, aturan-aturan dapat diperoleh secara otomatis dan diharapkan berlaku umum untuk data-data yang belum pernah diketahui. Teknik *learning* telah digunakan pada berbagai bidang, seperti transportasi, *speech processing*, *computer vision*, *robotics* dan sebagainya. Salah satu implementasi teknik ini di bidang transportasi adalah masalah *Travelling Salesman Program* (TSP).

TSP dapat didefinisikan sebagai pencarian urutan kota yang harus dikunjungi untuk meminimalkan biaya dimana setiap kota hanya boleh dikunjungi maksimum satu kali dan pencarian diakhiri dengan kembali ke kota semula. TSP memiliki ruang masalah yang sangat besar. Untuk 20 kota saja, TSP mempunyai ruang masalah sebanyak 121.645.100.408.832.000 jalur yang mungkin.

Ada dua jenis TSP, yaitu asimetris dan simetris. Pada TSP asimetris, biaya dari kota 1 ke kota 2 tidak sama dengan biaya dari kota 2 ke kota 1. Pada TSP simetris, biaya dari kota 1 ke kota 2 adalah sama dengan dari kota 2 ke kota 1.

Sebagai contoh sederhana, misalkan terdapat enam kota dengan koordinat kartesian seperti terlihat pada gambar 1 dibawah. Dengan mengambil batasan bahwa kasus ini merupakan TSP simetris, maka permasalahannya adalah bagaimanakah langkah-langkah untuk memperoleh jalur yang paling optimal berdasarkan biaya berupa jarak kartesian?



Gambar 1.1 Kasus TSP

Untuk menyelesaikan permasalahan TSP diatas, dapat menggunakan salah satu metode *learning* yang ada yaitu Algoritma Genetika (AG). Melalui metode ini diharapkan permasalahan TSP diatas dapat terpecahkan dengan diperolehnya jalur yang paling optimal sehingga dapat meminimalkan biaya yang dikeluarkan.

II. METODE PENELITIAN

2.1 Metode Studi Pustaka

Metode ini dilakukan untuk menambah wawasan dari buku-buku, artikel dan sumber-sumber lain yang layak, seperti naskah-naskah yang tersedia di Internet untuk menunjang pemecahan masalah penelitian ini.

2.2 Metode Analisa dan Perancangan

Metode ini dilakukan untuk menganalisa hasil studi pustaka yang telah dilakukan dan merancang langkah-langkah metode optimasi terhadap permasalahan TSP menggunakan algoritma genetika berdasarkan hasil analisa.

2.3 Konsep Kecerdasan Buatan

Kecerdasan buatan atau *artificial intelligence* merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia [3].

Untuk mengembangkan aplikasi kecerdasan buatan, ada dua bagian utama yang sangat dibutuhkan, yaitu:

1. Basis pengetahuan, berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
2. Motor/mesin inferensi, yaitu kemampuan menarik kesimpulan berdasarkan pengalaman.

2.4 Konsep Algoritma Genetika

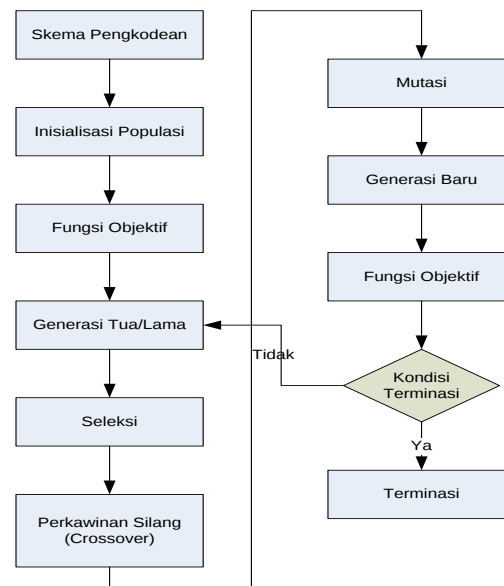
Algoritma genetika adalah suatu algoritma pencarian yang berbasis pada mekanisme seleksi alam dan genetika. Algoritma genetika merupakan salah satu algoritma yang sangat tepat digunakan dalam menyelesaikan masalah optimasi kompleks, yang sulit dilakukan dengan metode konvensional.

Algoritma genetika diperkenalkan pertama kali oleh John Holland pada tahun 1975 dari Universitas Michigan. John Holland menyatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan ke dalam terminologi genetika. Goldberg (1989) mendefinisikan algoritma genetika sebagai suatu pencarian algoritma berdasarkan mekanisme seleksi alam dan genetika alam.

Algoritma genetika bekerja dari populasi yang merupakan himpunan solusi yang dihasilkan secara acak. Setiap anggota himpunan yang merepresentasikan satu solusi masalah dinamakan kromosom. Kromosom dalam suatu populasi berevolusi dalam iterasi yang dinamakan generasi. Tiap kromosom dievaluasi berdasarkan fungsi evaluasi (*fitness function*). Pada algoritma genetika, fungsi evaluasi biasanya berupa fungsi objektif dari masalah yang akan dioptimasi.

Kromosom-kromosom diseleksi menurut nilai fungsi evaluasi masing-masing. Kromosom yang kuat mempunyai kemungkinan tinggi untuk bertahan hidup pada generasi berikutnya, tetapi tidak menutup kemungkinan juga bagi kromosom lemah

untuk tetap bertahan hidup. Proses seleksi tersebut kemudian ditentukan oleh kromosom-kromosom baru (*offspring*) melalui proses pindah silang (*crossover*) dan mutasi dari kromosom yang terpilih (*parents*). Dari dua proses tersebut diatas maka terbentuk suatu generasi baru yang akan diulangi terus menerus hingga mencapai suatu konvergensi, yaitu sebanyak generasi yang diinginkan.



Gambar 2. Struktur Umum Algoritma Genetika

III. HASIL DAN PEMBAHASAN

TSP merupakan persoalan yang mempunyai konsep sederhana dan mudah dipahami. Optimasi pada suatu penyelesaian fungsi berarti penentuan jarak lokasi minimum atau maksimum dari fungsi tersebut. Pada TSP, optimasi diinginkan agar ditemukan rute perjalanan terpendek untuk melewati sejumlah kota dengan jalur tertentu sehingga setiap kota hanya terlewati satu kali dan perjalanan diakhiri dengan kembali ke kota semula.

Ada dua macam bentuk persoalan TSP ditinjau dari jumlah pelaku perjalanan, Bentuk pertama dengan satu orang yang berkeliling dan dengan lebih dari satu (banyak) orang yang berkeliling sebagai bentuk kedua. Dalam pembahasan ini hanya akan dibahas bentuk pertama.

3.1 Skema Pengkodean

Suatu permasalahan harus dikonversi dulu ke dalam bentuk individu yang diwakili oleh satu atau lebih kromosom dengan kode tertentu dengan menggunakan salah satu dari beberapa skema pengkodean yang ada. Penentuan skema pengkodean yang digunakan harus disesuaikan dengan solusi yang ingin dicapai dari permasalahan tersebut.

Pada kasus TSP diatas, solusi yang ingin dicari adalah urutan kota yang harus dikunjungi (maksimum satu kali) dengan total biaya minimal. Jadi solusinya adalah berupa urutan, bukan suatu nilai. Untuk mencapai solusi ini tidak dapat dilakukan dengan skema pengkodean yang umum digunakan dalam algoritma genetika.

Skema pengkodean khusus yang dapat digunakan pada kasus ini adalah skema *permutation encoding*. Pada skema ini, setiap kromosom merupakan sekumpulan nomor yang merepresentasikan nomor dalam urutan.

Ruang pencarian dinyatakan dengan jumlah total dari kemungkinan permutasi, dan populasi dinyatakan dengan sekumpulan permutasi yang ada.

Suatu nilai tidak boleh berulang dalam suatu kromosom tunggal. Sebagai contoh, jika diberikan sekumpulan gen {A, B, C, D, E}, maka beberapa kemungkinan kromosom yang ada sebagai berikut:

Kromosom 1 : ABCDE

Kromosom 2 : DBCAE

Kromosom 3 : CBADE, dan seterusnya.

Untuk kasus TSP diatas dengan menggunakan skema *permutation encoding*, setiap individu berisi satu kromosom yang berisi gen-gen yang merepresentasikan nomorurut kota.

Jumlah gen dalam setiap individu adalah sebanyak jumlah kota yang ada. Oleh karena jumlah kota yang ingin dikunjungi sebanyak 6 kota, maka setiap individu berisi 6 gen yang nilainya berbeda (karena setiap kota hanya boleh dikunjungi satu kali).

Setiap nilai gen (*allele*) menyatakan nomerurut kota dan posisi gen menyatakan urutan kunjungan. Misalkan setiap individu dinyatakan dengan notasi D, maka D1 merepresentasikan suatu solusi dengan suatu urutan kunjungan. Berikut contoh individu-individu untuk kasus diatas.

D1	6	2	4	1	5	3
----	---	---	---	---	---	---

D2	2	5	3	4	6	1
----	---	---	---	---	---	---

Gambar 3. Representasi Individu untuk TSP

3.2 Penentuan Nilai *Fitness*

Pada evolusi dunia nyata, individu bernilai *fitness* tinggi akan bertahan hidup, sedangkan individu bernilai *fitness* rendah akan mati. Pada algoritma genetika, suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran nilai *fitness*-nya.

Pada kasus TSP diatas, karena tujuan TSP adalah meminimalkan total biaya (masalah minimasi), maka fungsi *fitness* yang bisa digunakan adalah:

$$\frac{1}{\text{total biaya}}$$

Dalam hal ini, yang dimaksud dengan total biaya adalah jumlah jarak kartesian antara satu kota dengan kota lainnya secara melingkar.

Misalkan, untuk sejumlah n kota, total biayanya adalah jarak kartesian dari kota 1 ke kota 2 ditambah jarak kartesian dari kota 2 ke kota 3 dan seterusnya sampai jarak kartesian dari kota n ke kota 1. Jarak kartesian antara kota A dan kota B dinyatakan dengan rumus berikut (Suyanto, 2007):

$$\|A - B\| = \sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2}$$

(X_A, Y_A) menyatakan posisi koordinat kota A dan (X_B, Y_B) menyatakan posisi koordinat kota B.

3.3 Proses Seleksi Orang Tua

Proses pemilihan dua individu sebagai orang tua biasanya dilakukan secara proporsional berdasarkan nilai-nilai *fitness*-nya.

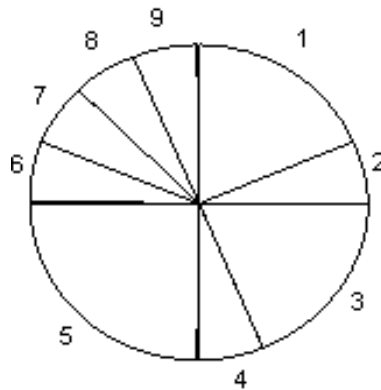
Untuk kasus TSP diatas, proses seleksi orang tua dapat menggunakan metode seleksi yang umum digunakan yaitu *roulette-wheel*.

Pada metode ini, masing-masing individu menempati potongan lingkaran pada roda *roulette* secara proporsional sesuai dengan nilai *fitness*-nya. Sebagai contoh, pada gambar berikut menggambarkan contoh penggunaan metode *roulette-wheel* pada TSP.

Individu D5 yang bernilai *fitness* paling besar menempati seperempat lingkaran roda *roulette*. Dengan demikian, D5 memiliki peluang sebesar 0.25 untuk terpilih sebagai orang tua.

Tabel 1. Representasi Nilai *Fitness* Individu

Individu	Nilai Fitness
D1	0.6
D2	0.2
D3	0.6
D4	0.2
D5	0.8
D6	0.2
D7	0.2
D8	0.2
D9	0.2
Jumlah	3.2

Gambar 4. Metode *Roulette-Wheel* untuk TSP

3.4 Proses Pindah Silang

Pada proses pindah silang terjadi kombinasi pewarisan gen-gen dari induknya, dimana gen-gen dari kedua induk dapat bercampur sehingga dihasilkan susunan kromosom yang baru.

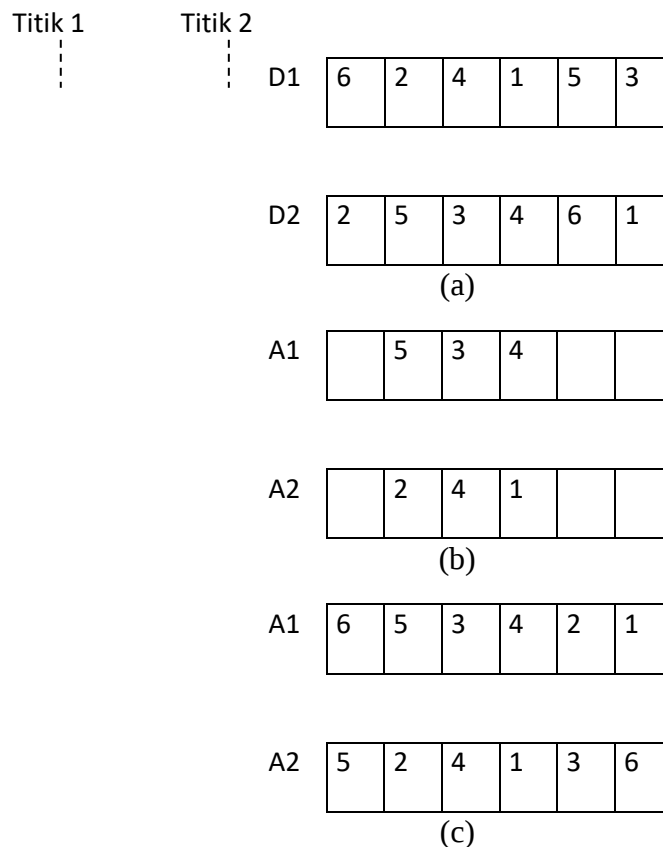
Dengan suatu skema pindah silang tertentu, dua individu dipilih sebagai orang tua. Selanjutnya, ditentukan titik pindah silang secara acak.

Jika diasumsikan L adalah panjang kromosom, maka titik pindah silang berada antara 1 sampai $L-1$. Kemudian beberapa bagian dari dua kromosom ditukar pada titik pindah silang yang dipilih. Pertukaran tersebut akan menghasilkan dua individu anak (*offspring*). Peluang keberhasilan proses pindah silang dinyatakan dengan probabilitas pindah silang atau p_c .

Untuk kasus TSP diatas, proses pindah silang dilakukan dengan menggunakan skema *order crossover*, yaitu:

1. Bangkitkan dua titik pindah silang pada dua orang tua yang terpilih, misalkan D1 dan D2
2. Gen-gen yang berada di antara kedua titik saling ditukarkan, sehingga menghasilkan calon anak, misalkan A1 dan A2 (5.a)
3. Gen-gen pada D1 yang belum ada di A1 dimasukkan ke tempat yang kosong secara berurutan (5.b)
4. Hal yang sama juga dilakukan untuk A2 (5.c)

Agar lebih jelasnya, dapat dilihat ilustrasi pada gambar berikut.

Gambar 5. Proses *Order Crossover* untuk TSP

3.5 Proses Mutasi

Proses mutasi diperlukan untuk mengembalikan informasi bit yang hilang akibat *crossover*. Mutasi diterapkan dengan probabilitas yang sangat kecil. Jika mutasi dilakukan terlalu sering, maka akan menghasilkan individu yang lemah karena konfigurasi gen pada individu yang unggul akan dirusak.

Untuk kasus TSP diatas, proses mutasi dapat dilakukan dengan metode *swapping mutation*. Pada metode ini, komponen mutasi biasanya diimplementasikan dengan menukarkan gen termutasi dengan gen lain yang dipilih secara acak. Sebagai contoh, suatu individu {2, 3, 4, 1, 5, 6} dapat termutasi dengan menukarkan gen ke-1 dan gen ke-3 sehingga menjadi {4, 3, 2, 1, 5, 6}.

3.6 Proses Penggantian Populasi

Proses penggantian populasi diperlukan untuk mengganti populasi yang lama dengan populasi hasil proses pindah silang sehingga pada akhirnya akan dihasilkan populasi yang benar-benar optimum.

Untuk kasus TSP diatas, proses penggantian dapat dilakukan dengan menggunakan proses penggantian berjenis *Steady State*, dimana proses penggantian dilakukan setiap kali dihasilkan dua *offspring* hasil *crossover*. *Offspring* menggantikan kromosom yang nilai *fitness*-nya paling kecil. Dengan demikian, populasi baru yang dihasilkan selalu memiliki individu-individu yang lebih baik dibandingkan dengan populasi lama. Pada gambar berikut diberikan algoritma untuk mengilustrasikan proses tersebut [5].

```
Bangkitkan populasi awal,  $P$  individu

Loop untuk  $P$  individu

    Dekodekan individu

    Evaluasi individu

End

Loop sampai kondisi berhenti

    Pilih dua individu sebagai parent 1 dan parent 2

    If perlu crossover then

        Offspring = crossover(parent1, parent2)

    End
```

3.7 Proses Penghentian Iterasi

Proses evolusi di dalam algoritma genetika berlangsung terus menerus dalam bentuk iterasi untuk menghasilkan populasi yang benar-benar optimum. Untuk menghentikan iterasi ini, dapat dilakukan dengan beberapa cara.

Untuk kasus TSP diatas, proses penghentian iterasi ini dapat dilakukan dengan menghitung kegagalan penggantian anggota populasi yang terjadi secara berurutan sampai jumlah tertentu. Misalkan setelah 100 iterasi, tidak ada penggantian individu dalam populasi karena individu anak yang dihasilkan selalu memiliki nilai *fitness* yang lebih rendah daripada orang tuanya, maka iterasi dapat dihentikan.

IV. KESIMPULAN

4.1 Kesimpulan

Algoritma genetika sebagai solusi optimasi dalam *Travelling Salesman Problem* dapat dilakukan dengan mengikuti serangkaian proses evolusi dengan langkah-langkah sebagai berikut:

Melakukan proses pengkodean permasalahan yang ingin dipecahkan menjadi bentuk individu dengan menggunakan skema *permutation encoding*

Menghitung nilai fitness yang merepresentasikan panjang jalur yang dilalui dari suatu kota ke kota berikutnya sebagai jarak kartesian dari kota A ke kota B

1. Melakukan proses seleksi orang tua dengan metode *roulette-wheel*
2. Melakukan proses pindah silang dengan metode *order crossover*

3. Melakukan proses mutasi dengan metode *swapping mutation*
4. Melakukan proses penggantian populasi dengan algoritma *Steady State*

Melalui proses evolusi diatas, *Travelling Salesman Problem* dapat dipecahkan sehingga diperoleh jalur yang paling optimal untuk meminimalkan biaya yang dikeluarkan dalam mengunjungi sejumlah kota.

4.2 Saran

Untuk memperoleh solusi yang paling optimal dalam menangani masalah optimasi pada *Travelling Salesman Problem* dapat dilakukan analisa perbandingan metode Algoritma Genetika dengan metode optimasi yang lainnya.

V. DAFTAR PUSTAKA

- [1] Desiani Anita, Arhami Muhammad. 2006. Konsep Kecerdasan Buatan. Andi Yogyakarta. Yogyakarta.
- [2] Goldberg, D. 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Company, Inc.
- [3] Kusumadewi Sri. 2003. Artificial Intelligence (Teknik dan Aplikasinya). Graha Ilmu. Yogyakarta.
- [4] Russell, Stuart; dan Norvig, Peter. 2003. Artificial Intelligence A Modern Approach. International Edition, Edisi 2. New Jersey: Pearson Prentice-Hall Education International.
- [5] Suyanto. 2007. Artificial Intelligence Searching, Reasoning, Planning and Learning. Informatika. Bandung.
- [6] <http://www.obitko.com/tutorials/genetic-algorithms/encoding.php>