

ANALISIS PENGARUH MUTASI TERHADAP PERFORMANCE ALGORITMA GENETIKA

Erianto Ongko

Prodi Teknik Informatika, Akademi Teknologi Industri Immanuel
e-Mail: erianto_ongko@yahoo.co.id

Abstrak

Algoritma genetika pada umumnya digunakan untuk pencarian solusi dari suatu permasalahan yang menuntut pencarian solusi yang optimal pada suatu permasalahan. Salah satu tahapan di dalam algoritma genetika adalah proses mutasi. Proses mutasi dilakukan setelah proses crossover dilakukan. Proses mutasi melibatkan pertukaran gen untuk beberapa gen yang berada pada posisi tertentu. Proses mutasi dimaksudkan untuk mencegah hasil terjebak di dalam kondisi local optima. Penelitian ini dimaksudkan untuk melihat pengaruh dari proses mutasi terhadap performance dari algoritma genetika. Studi kasus di dalam permasalahan ini adalah permasalahan Travelling Salesman Problem (TSP), dimana menggunakan library Berlin52.tsp. Hasil penelitian ini diharapkan dapat memberikan gambaran mengenai pengaruh mutasi terhadap performance algoritma genetika dengan memvariasikan nilai probabilitas mutasi.

Kata Kunci: Algoritma Genetika, Mutasi, Performance, Travelling Salesman Problem

1. PENDAHULUAN

Penelitian mengenai Algoritma Genetika dimulai pada Tahun 1975, ketika John Holland mengemukakan hasil penelitiannya mengenai proses komputasi yang menggunakan konsep evolusi di dalam bukunya yang berjudul "*Adaptation in Natural and Artificial Systems*" (Negnevitsky, 2005).

Algoritma genetika merupakan contoh permasalahan *soft computing* yang sering digunakan untuk penyelesaian masalah optimasi. Ketika membahas mengenai algoritma genetika ada tiga parameter utama yang harus didefinisikan yaitu ukuran populasi, probabilitas *crossover*, dan probabilitas mutasi. Ketiga parameter ini harus didefinisikan dengan baik sehingga tidak terjadi konvergensi dini atau local optima yaitu dimana individu-individu dalam populasi konvergen pada suatu solusi optimum lokal sehingga hasil paling optimum tidak dapat ditemukan (Muzid, 2014).

Schrempf dan Hobolth (2017) mengemukakan Model Wright-Fisher untuk proses mutasi pada mutasi dengan probabilitas mutasi yang kecil.

Penelitian ini dimaksudkan untuk melihat pengaruh dari proses mutasi terhadap performance dari algoritma genetika. Studi kasus di dalam permasalahan ini adalah

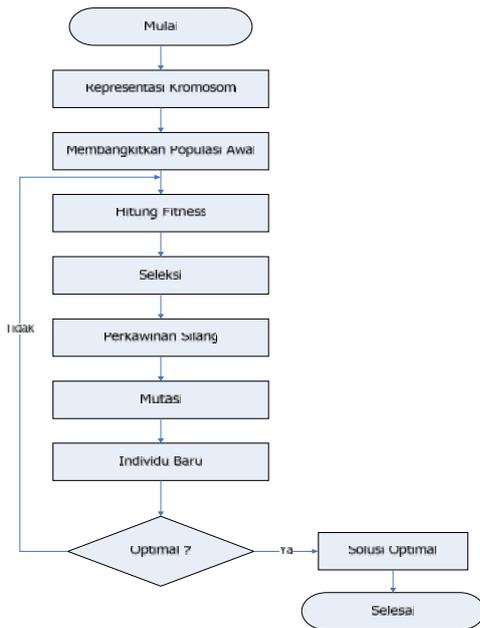
permasalahan Travelling Salesman Problem (TSP), dimana menggunakan library Berlin52.tsp. Hasil penelitian ini diharapkan dapat memberikan gambaran mengenai pengaruh mutasi terhadap performance algoritma genetika dengan memvariasikan nilai probabilitas mutasi dan juga jumlah generasi.

2. LANDASAN TEORI

2.1. Proses Algoritma Genetika

Algoritma genetika adalah suatu algoritma *stochastic* yang memodelkan permasalahan evolusi dari spesies biologi melalui seleksi alam (Konar, 2005). Secara umum, proses penentuan populasi dilakukan secara acak dan solusi dibangkitkan sesudah tahapan konsektif dari proses *crossover* dan mutasi. Setiap individu dari populasi memiliki nilai yang diasosiasikan kedalam suatu nilai *fitness*, di dalam kaitannya untuk menyelesaikan suatu permasalahan (Rabunal, 2006).

Adapun diagram alir dari algoritma genetika standart dapat dilihat pada Gambar 2.1.



Gambar 2.1. Diagram Alir dari Proses

Algoritma Genetika (Negnevitsky, 2005)
Secara umum struktur dari suatu Algoritma Genetika dapat didefinisikan dengan langkah-langkah sebagai berikut: (Negnevitsky, 2005)

1. Membangkitkan populasi awal
Populasi awal ini dibangkitkan secara acak sehingga didapatkan solusi awal. Populasi itu sendiri terdiri atas sejumlah kromosom yang merepresentasikan solusi yang diinginkan.
2. Menghitung Fitness dari Tiap Generasi.
Pada tiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang dinamakan *fitness*. Nilai *fitness* suatu kromosom menggambarkan kualitas kromosom dalam populasi tersebut. Fungsi *Fitness* tersebut dapat dilihat pada Persamaan 1.
$$Fitness = 1/(1 + Fungsi\ Objektif \dots\dots\dots(1)$$

Dari persamaan 1, nilai *fitness* ditentukan oleh nilai fungsi objektif. Fungsi objektif tersebut menunjukkan hasil penjumlahan jarak pada tiap kromosom. Semakin tinggi nilai *fitness* akan semakin besar kemungkinan kromosom tersebut terpilih ke generasi berikutnya. Jadi nilai fungsi objektif berbanding terbalik dengan nilai *fitness*, semakin kecil nilai fungsi objektif semakin besar nilai *fitness*nya.
3. Evaluasi solusi
Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom dan mengevaluasinya

sampai terpenuhi kriteria berhenti. Bila kriteria berhenti belum terpenuhi maka akan dilanjutkan dengan proses perkawinan. Beberapa kriteria berhenti sering digunakan antara lain: berhenti pada generasi tertentu, berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai *fitness* tertinggi tidak berubah, berhenti dalam ngenerasi tidak didapatkan nilai *fitness* yang lebih tinggi.

4. Proses *Crossover*
Menentukan nilai *Pc* (*Probability Crossover*) dan kemudian menentukan pasangan kromosom yang akan terlibat di dalam proses *crossover* berdasarkan nilai *PC* yang dibangkitkan tersebut dengan menggunakan salah satu metode *crossover*.
5. Proses Mutasi
Menentukan nilai *mutation rate*, dan kemudian berdasarkan nilai bilangan *random* yang dibangkitkan akan dapat ditentukan gen-gen yang terlibat di dalam proses mutasi tersebut.
6. Menjadikan Kromosom Anak hasil dari Proses *Crossover* dan Mutasi sebagai Populasi Baru.

2.2. Proses Mutasi

Mutasi adalah proses perubahan gen di dalam sebuah kromosom. Proses Mutasi akan dimulai setelah proses *crossover* selesai dilakukan. Mutasi akan mengubah *offspring* hasil dari proses *crossover* dengan mengubah *bit* 1 menjadi 0 atau *bit* 0 menjadi 1. Mutasi dapat melibatkan setiap posisi *bit* dari *gen* di dalam *kromosom* dengan beberapa probabilitas yang umumnya berukuran kecil. Mutasi dimaksudkan untuk mencegah hasil pencarian mengarah pada keadaan *local optima* di dalam sebuah area pencarian (Shaikh, 2012).

Jenis mutasi yang akan digunakan di dalam penelitian ini adalah Mutasi dalam Pengkodean Biner. Mutasi di dalam pengkodean biner dilakukan dengan cara melakukan operasi yang menginversi nilai bit pada posisi tertentu yang terpilih secara acak (atau menggunakan skema tertentu) pada kromosom, yang disebut *inverse bit*.

Tabel 2.1. Contoh Mutasi pada Pengkodean Biner

Kromosom sebelum mutasi	10010111
Kromosom setelah mutasi	10010011

2.3. Travelling Salesman Problem

Permasalahan matematika tentang *Traveling Salesman Problem* dikemukakan pada tahun 1800 oleh matematikawan Irlandia William Rowan Hamilton dan matematikawan Inggris Thomas Penyngton. Permasalahan TSP ini merupakan permasalahan di mana seorang *salesman* harus mengunjungi semua kota di mana tiap kota hanya dikunjungi sekali, dan dia harus mulai dan kembali ke kota asal. Tujuan yang ingin dicapai pada permasalahan TSP adalah mencari rute terpendek bagi seorang *salesman*. (Biggs *et.al*, 1976).

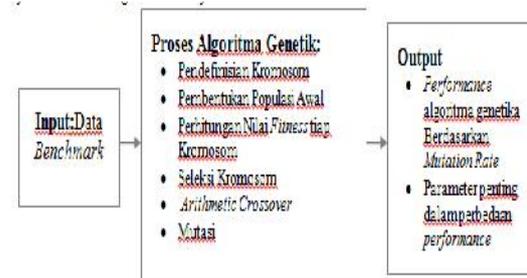
3. METODE PENELITIAN

Travelling Salesman Problem termasuk ke dalam kelas permasalahan NP (*NonDeterministic Polynomial*) kategori sulit karena memiliki kompleksitas $O(n!)$. Permasalahan utama dari TSP adalah bagaimana seorang *salesman* dapat mengatur rute perjalanannya untuk mengunjungi sejumlah kota yang diketahui jarak satu kota dengan kota lainnya sehingga jarak yang ditempuh merupakan jarak minimum di mana seorang *salesman* hanya dapat mengunjungi kota tersebut tepat satu kali. Salah satu metode yang dapat digunakan di dalam menyelesaikan permasalahan TSP yaitu algoritma genetika.

Proses mutasi dimaksudkan untuk mencegah hasil dari algoritma genetika terjebak di dalam kondisi *local optima*. Penelitian ini dimaksudkan untuk melihat pengaruh dari proses mutasi terhadap *performance* dari algoritma genetika.

Data yang digunakan merupakan data *benchmark* yang diambil dari *Travelling Salesman Problem Library* (TSPLIB). Adapun data yang digunakan yaitu data berlin52.tsp.

Adapun prosedur kerja yang dilakukan oleh peneliti dari penelitian ini dapat dilihat secara keseluruhan pada Gambar 2.



Gambar 3.1 Metode Penelitian

Pada Gambar 3.1, dapat dilihat bahwa proses penelitian dimulai dari penentuan *input* yang dalam hal ini peneliti menggunakan data *benchmark* yang sudah ada, yang bersumber dari TSPLIB yaitu *Berlin52.tsp*. Tahapan penelitian yang dilakukan dimulai dari pendefinisian kromosom, pembentukan populasi awal, perhitungan nilai *fitness* tiap kromosom, penyeleksian kromosom, pemilihan metode kawin silang (*crossover*) yang dalam hal ini menggunakan *arithmetic crossover*, dan permutasian. Penelitian ini akan fokus pada *output* berupa *performance* berdasarkan pada proses mutasi.

4. HASIL DAN PEMBAHASAN

Analisis Pembangkitan Populasi Awal

Permasalahan *Berlin52.TSP* yang digunakan di dalam penelitian ini meliputi 51 kota yang akan dikunjungi dari kota asal. Jumlah populasi yang digunakan adalah sebanyak 10 dan terdapat 51 kromosom pada tiap populasi, yang mewakili jumlah kota yang harus dikunjungi.

Analisis Proses Penghitungan Nilai *Fitness*

Untuk dapat menghitung nilai *fitness* maka kita harus menghitung nilai fungsi objek terlebih dahulu. Fungsi objektif di dalam permasalahan ini adalah merupakan total jarak perjalanan yang dilalui oleh seorang *salesman*. Adapun jarak antara 1 kota dengan kota lainnya dapat dihitung dengan menggunakan perhitungan *euclidean distance*. Adapun persamaan untuk *euclidean distance* dapat dilihat pada Persamaan 1.

$$d_{(i,j)} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \dots\dots(1)$$

Dimana:

x_i = Koordinat x kota i
 x_j = Koordinat x kota j
 y_i = Koordinat y kota i
 y_j = Koordinat y kota j

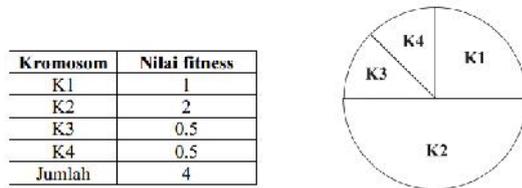
Setelah menghitung nilai fungsi objektif, maka nilai *fitness* dapat dihitung dengan menggunakan Persamaan 2.¹

$$Fitness = 1 / (1 + Fungsi_Objektif) \dots (2)$$

Analisis Proses Seleksi

Proses seleksi di dalam penelitian menggunakan metode *roulette wheel selection*.

Metode *roulette wheel selection* dapat dilihat pada Gambar 3.



Sumber: Kumar, 2012

Gambar 4.1. Metode *Roulette Wheel Selection*

Analisis Proses Crossover

Metode *crossover* yang digunakan di dalam penelitian ini adalah metode *arithmtic crossover*. Jenis *arithmtic crossover* yang digunakan adalah *Whole Arithmetic Crossover*.

Pada Metode *Whole Arithmetic Crossover*, gen pada kromosom *offspring* diperoleh dari hasil operasi aritmatika gen pada kromosom *parent*, di mana proses aritmatika yang dilakukan sesuai dengan persamaan 3. Ilustrasi dari proses *whole arithmetic crossover* dapat dilihat pada Tabel 2.

$$Child = \left\langle r \cdot x + (1 - r) \cdot y \right\rangle \dots (3)$$

Tabel 4.1 *Whole Arithmetic Crossover*

Kromosom Parent 1	0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
-------------------	-------------------------------------

Kromosom Parent 2	0.3 0.2 0.3 0.2 0.3 0.2 0.3 0.2 0.3
	0.5
Kromosom Offspring	0.2 0.2 0.3 0.3 0.4 0.4 0.5 0.5 0.6

Analisis Proses Mutasi

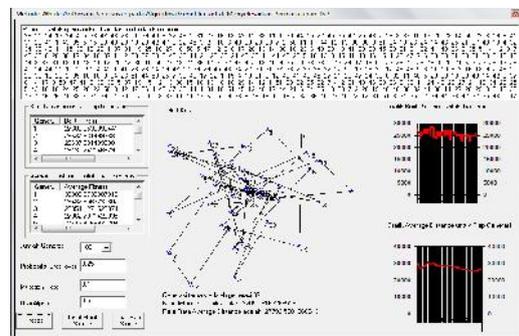
Proses mutasi yang akan digunakan di dalam penelitian ini adalah mutasi dalam pengkodean biner. Proses mutasi akan divariasikan nilai probabilitas mutasi sehingga dapat diketahui pengaruh proses mutasi terhadap *performance* dari algoritma genetika

Analisis Proses Pengujian

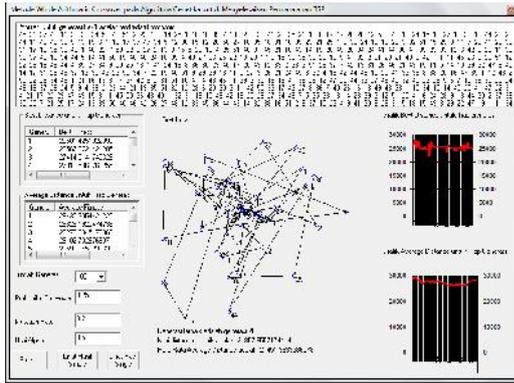
Pada penelitian ini akan dilakukan pengujian terhadap algoritma genetika. Metode kawin silang (*crossover*) yang digunakan adalah *Whole Arithmetic Crossover*. Penelitian ini akan menguji kaitan antara proses mutasi terhadap *performance* dari algoritma genetika. Proses mutasi akan diuji berdasarkan variasi dari nilai probabilitas mutasi (*mutation rate*). *Mutation rate* akan divariasikan pada nilai 0.1, 0.2, 0.3, dan 0.4. Hasil pengujian diharapkan dapat memberikan gambaran berupa pengaruh proses terhadap *performance* dari algoritma genetika.

Pengujian dilakukan dengan jumlah generasi sebanyak 100 generasi dengan nilai *probability crossover* 0.25 dan nilai *mutation rate* (Probabilitas Mutasi) sebesar 0.1 serta nilai sebesar 0.5 untuk melihat nilai *best distance* serta *average distance*.

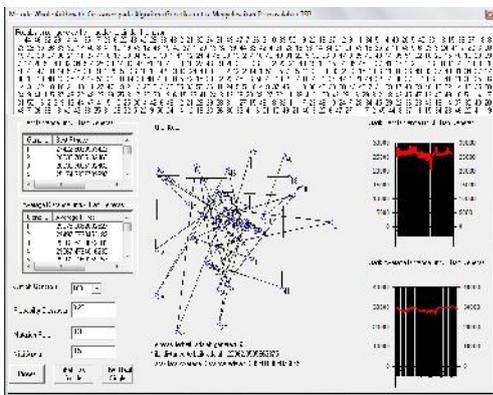
Adapun proses pengujian dapat dilihat pada Gambar 4.2, 4.3, 4.4, dan 4.5.



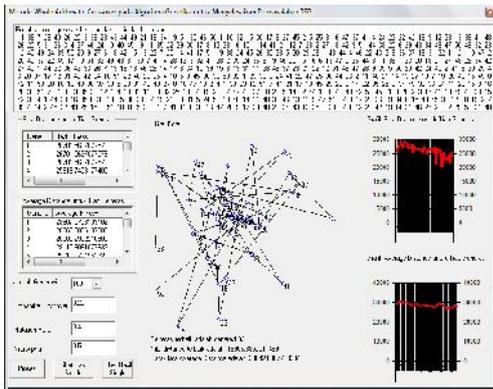
Gambar 4.2 Pengujian dengan Menggunakan Probabilitas Mutasi Sebesar 0.1



Gambar 4.3. Pengujian dengan Menggunakan Probabilitas Mutasi Sebesar 0.2



Gambar 4.4 Pengujian dengan Menggunakan Probabilitas Mutasi Sebesar 0.3



Gambar 4.5. Pengujian dengan Menggunakan Probabilitas Mutasi Sebesar 0.4

Hasil Pengujian dapat dilihat pada Tabel 4.2.

Tabel 4.2 Hasil Pengujian

Probabilitas Mutasi	Best Distance	Average Distance
0.1	22055.91	27792.58
0.2	21857.85	27451.98

0.3	20062.05	28950.68
0.4	19906.38	28642.60

Pada Tabel 4.2, dapat dilihat bahwa nilai *distance* yang semakin rendah adalah semakin baik yang berarti bahwa jarak yang ditempuh pada permasalahan Berlin52.TSP adalah semakin rendah. Pada pengujian dengan memvariasikan nilai probabilitas mutasi sebesar 0.1, 0.2, 0.3, dan 0.4 dan dikaitkan dengan nilai *best distance* dan nilai *average distance*. Nilai *best distance* merupakan nilai *distance* terendah yang diperoleh pada pengujian dengan melibatkan 51 kromosom dengan jumlah generasi sebanyak 100 generasi. Nilai *average distance* merupakan nilai rata-rata *distance* yang diperoleh pada pengujian dengan melibatkan 51 kromosom dengan jumlah generasi sebanyak 100 generasi.

Pembahasan

Berdasarkan hasil pengujian bahwa nilai probabilitas mutasi berpengaruh terutama terhadap nilai *best distance* yang diperoleh. Hal ini dapat dilihat bahwa terjadi peningkatan nilai *best distance* seiring dengan peningkatan nilai probabilitas mutasi. Bila dikaitkan dengan nilai rata-rata *average distance* maka nilai probabilitas mutasi tidak memiliki pengaruh yang signifikan dan hasil dapat berbeda-beda di dalam tiap pengujian.

Pengaruh nilai probabilitas mutasi terhadap nilai *best distance* meskipun tidak diikuti dengan pengaruh terhadap nilai *average distance* sudah menunjukkan bahwa proses mutasi memiliki pengaruh terhadap *performance* dari algoritma genetika.

Mutasi tidak berpengaruh besar terhadap nilai *average distance*, yang merupakan nilai rata-rata dari seluruh *distance* yang ada pada 100 generasi, dapat terjadi karena ada kalanya pertukaran posisi bit pada proses mutasi pada beberapa generasi dapat mengurangi kualitas gen pada *offspring*. Namun, secara umum semakin banyak pertukaran gen, yang ditandai dengan semakin tingginya nilai probabilitas mutasi akan dapat mencapai kualitas gen terbaik pada *offspring*. Hal ini ditandai dengan peningkatan kualitas *best distance* seiring dengan peningkatan nilai probabilitas mutasi.

5. KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan oleh peneliti maka beberapa kesimpulan yang dapat ditarik oleh peneliti adalah sebagai berikut.

1. Hasil penelitian menunjukkan bahwa proses mutasi berpengaruh terhadap *performance* dari algoritma genetika. Hal ini ditandai dengan terjadinya peningkatan *best distance* seiring dengan peningkatan nilai probabilitas mutasi. Peningkatan nilai probabilitas mutasi yang berarti peningkatan jumlah gen yang mengalami pertukaran ternyata dapat meningkatkan kualitas nilai *best distance* yang diperoleh.
2. Nilai *average best distance* tidak begitu terpengaruh dengan proses mutasi yang terjadi. *Average Distance* merupakan nilai rata-rata *distance* pada 100 generasi, dan ini menandakan bahwa dalam beberapa generasi semakin meningkatnya jumlah gen yang mengalami pertukaran, terkadang tidak dapat meningkatkan kualitas dari *offspring* yang dihasilkan.

5.2 Saran

Adapun saran yang dapat diberikan untuk pelaksanaan penelitian di masa mendatang adalah sebagai berikut.

1. Perlunya dilakukan pengujian dengan memvariasikan jumlah generasi. Hal ini dilakukan untuk mendapatkan gambaran apakah mutasi tetap berpengaruh pada jumlah generasi yang semakin besar.
2. Perlunya memvariasikan metode *crossover* dan juga nilai *probability crossover* sehingga akan diperoleh gambaran apakah mutasi tetap berpengaruh bila diubah proses *crossover* yang ada.

DAFTAR PUSTAKA

- [1] Biggs, N.L., Lloyd, E.K. and Wilson, R.J. 1976. *Graph Theory 1736-1936*. Clarendon Press: Oxford
- [2] Konar, Amit. 2005. *Computational Intelligence Principles, Techniques, and Applications*. Springer: Calcutta, India
- [3] Kumar, Rakesh and Jyotishree. 2012. Blending Roulette Wheel Selection & Rank Selection in Genetic Algorithms, *International Journal of Machine Learning and Computing*2(4): 365-370
- [4] Muzid, Syafiul. 2014. Dinamisasi Parameter Algoritma Genetika Menggunakan Population Resizing on Fitness Improvement Fuzzy Evolutionary Algorithm (PROFIFEA). *Prosiding SNATIF 2014*, pp. 471-478
- [5] Negnevitsky, Michael. 2005. *Artificial Intelligence-A Guide to Intelligent Systems*. Addison Wesley: Edinburg
- [6] Rabunal, Juan R. and Dorado, Julian. 2006. *Artificial Neural Networks in Real-Life Applications*, Ideal Group Publishing: Hershey, United States of America.
- [7] Schrempf, D & Hobolth, A. 2017. An alternative derivation of the stationary distribution of the multivariate neutral Wright-Fisher model for low mutation rates with a view to mutation rate estimation from site frequency data. *Theoretical Population Biology*114: pp. 88-94
- [8] Shaikh, Misba and Panchal, Mahesh. 2012. Solving Asymmetric Travelling Salesman Problem Using Memetic Algorithm, *International Journal of Emerging Technology and Advanced Engineering* 2(11): 634-639