PENGENDALIAN *MUTATION RATE* PADA ALGORITMA GENETIKA

Rijois Iboy Erwin Saragih

Prodi Komputerisasi Akuntansi, Universitas Methodist Indonesia
Jl. Hang Tuah No. 8 Medan, Sumatera Utara

erwin_saragih@yahoo.com

Abstract

Genetic algorithm is heuristic searching algorithm which based on nature selection of mechanism and nature genetic. The basic concept that inspires the genetic algorithm is that evolution theory. In the process of searching the best solution on classic genetic algorithm often occurs optimum local. Optimum local is a common problem which is often occurs in genetic algorithm, and one of the reasons is that because of the population diversity. If population diversity is too low is led to optimum local, and if it is too high caused more times to look for the best solution. Mutation operator plays an important role in the process of genetic algorithm to manage that population diversity and an important element of mutation operator is mutation rate. On classic genetic algorithm that mutation rate is set in the beginning while the process of genetic algorithm depends on how many generations are. Therefore is needed to control mutation rate in generation. Controlling mutation operator, especially mutation rate based on Fuzzy Logic Controller (FLC) to manage population diversity, not too high or too low, in order to get optimal result. Evaluation is done 10 times execution by comparing the performance of standard genetic algorithm (GA), IAG and genetic algorithm based on Fuzzy Logic Controller (FLC) and experimental results show that there is an improvement on genetic algorithm performance based on FLC.

Keywords: *Genetic Algorithm, Fuzzy Logic Controller (FLC), Mutation Rate*

1. Pendahuluan

Algoritma genetika adalah kelas populasi berdasarkan teknik pencarian acak yang semakin banyak digunakan di sejumlah aplikasi praktis. Biasanya algoritma ini mempertahankan sejumlah solusi potensial untuk masalah yang sedang ditangani, yang dapat dilihat sebagai bentuk memori kerja ini dikenal sebagai populasi. Poin iteratif baru dalam ruang pencarian yang dihasilkan untuk evaluasi dan opsional dimasukkan ke dalam populasi (Smith, 2002).

Permasalahan umum pada algoritma genetika yang sering terjadi adalah lokal optima dan hal ini terjadi karena hilangnya perbedaan populasi (population diversity) awal dengan populasi selanjutanya (Zhu & Liu, 2004). Jika perbedaan populasi terlalu kecil akan memungkinkan terjadinya lokal optima, dan jika terlalu besar akan mengakibatkan lamanya waktu yang dibutuhkan algoritma genetika dalam menghasilkan solusi terbaik. Banyak penelitian yang telah dilakukan terkait dengan permasalahan diatas untuk menghindari lokal optima serta meningkatkan performansi algoritma genetika. Salah satu pendekatan yang dilakukan melalui perbaikan kinerja dari operator genetika itu sendiri; seperti operator seleksi, Persilangan dan mutasi.

Menurut Varnamkhasti et al (2012), performansi algoritma genetika dipengaruhi oleh operator genetika; operator seleksi, Persilangan dan mutasi secara umum. Secara khusus operator mutasi memegang peranan penting dalam proses evolusi dalam algortima genetika, permasalahan terjadi apabila hasil mutasi kromoson tidak berbeda dengan kromosom awal, dengan kata lain bahwa proses mutasi gagal menciptakan keturunan yang berbeda dari generasi sebelumnya. Mutasi yang efektif dalam algoritma genetika tercapai melalui membangun hubungan yang optimal diantara operator mutasi dan masalah pencarian itu sendiri.

Tedapat berbagai jenis operator mutasi pada algoritma genetika, seperti binary mutation (Holland, 1975), interchanging mutation, reversing mutation (Sivanandam & Deepa, 2008), Parity Encoding Mutation, Simple Sum Coding Mutation, Inversion Sum Coding Mutation, dan Cycle Sum Coding Mutation (Varnamkhasti et al, 2012). Dalam proses algortima genetika sering kali di dalam penentuan parameter genetika seperti mutation rate ditetapkan diawal, hal itu terjadi karna tidak adanya aturan baku yang mengaturnya. Keadaan tersebut menimbulkan ketidakpastian akan penentuan parameter tersebut, dan jika menentukan mutation rate yang tidak tepat maka akan membawa kepada local optima maupun performansi kinerja algoritma genetika tidak optimal.

Pada penelitian ini akan dilakukan pendekatan terhadap penentuan *mutation rate* yang tepat dalam menyelesaikan permasalahan kombinatorial optimasi seperti Knapsack Problem. Adapun maksud dari pendekatan tersebut adalah untuk men-design suatu teknik pemilihan *mutation rate* berdasarkan population diversity (perbedaan populasi) mengunakan *Fuzzy Logic Controller* (FLC). Perbedaan populasi diukur atas dasar karakteristik *genotype* dan *phenotype* dari kromosom.

Menurut Singh (2011), permasalahan Knapsack adalah suatu permasalahan optimasi kombinatorial. Sebagai contoh diberikan satu set item dengan berat dan nilai, kemudian dilakukan pemilihan dari item-item tersebut untuk dimasukan kedalam ransel (knapsack) dengan kapasitas terbatas. Jadi item-item yang dimasukan beratnya harus lebih kecil atau sama dengan kapasitas dari ransel tersebut, tetapi total nilai sebesar mungkin.

Haibo et al (2011) mengembangkan algoritma genetika untuk permasalahan knapsack problem dengan cara menjaga sekumpulan kromosom-kromoson terbaik maka solusi terbaik perlahan-

lahan dapat dicapai. Penerapan multi-point *crossover* dapat memberikan hasil yang lebih baik terhadap permasalahan knapsack problem melalui pembuatan scope pencarian lebih besar dan kondusif. Sebuah fungsi reulatte mendefinisikan perbandingan probabilitas Persilangan dan mutasi, yang nantinya menempatkan kromoson-kromoson pada posisi terdepan.

Pada penelitian (Varnamkhasti et al, 2012), dilakukan pendekatan penerapan *Fuzzy Logic Controller* (FLC) dalam pemilihan operator *crossover* dan probabilitasnya. Terdapat beberapa operator *crossover* pada algoritma genetika yang dapat dipilih namum jika operator yang digunakan tidak tepat terhadap permasalahan yang dihadapi maka timbul permasalahan umum algoritma genetika, yakni local optima. FLC berperan memilih operator *crossover* yang tepat terhadap permasalahan yang ada berdasarkan perbedaan populasi (population diversity).

Berdasarkan penelitian di atas maka penulis tertarik untuk melakukan penelitian bagaimana menerapkan *Fuzzy Logic Controller* (FLC) pada algoritma genetika dan melakukan analisis kinerjanya serta membandingkanya dengan algoritma genetika klasik.

2. Metode Penelitian

Pada algoritma genetika, penentuan operator mutasi yang dipengaruhi oleh mutation rate merupakan hal yang sangat penting, karena *mutation rate* yang tepat akan memberikan solusi yang optimal. Penentuan *mutation rate* menggunakan *Fuzzy Logic Controller* (FLC) diharapkan dapat memperbaiki kinerja algoritma genetika klasik sehingga diharapkan dapat menghindari lokal optima.

Data knapsack problem adalah suatu permasalahan optimasi kombinatorial. Data tersebut telah diuji oleh pelbagai algoritma yang berbeda untuk mencari solusi terbaik. Sebagai perbandingan juga digunakan data penelitian terkait dengan 50 item serta nilai weight dan value.

Table 1 Data Knapsack Problem

Item	Weight	Value
1	9	150
2	13	35
3	153	200
4	50	160
5	15	60
6	68	45
7	27	60
8	39	40
9	23	30
10	52	10
11	11	70
12	32	30
13	24	15
14	48	10
15	73	40
16	42	70
17	43	75
18	22	80
19	7	20
20	18	12
21	4	50
22	30	10

Inisialisasi Populasi

Pada populasi awal terdiri dari 6 kromosom yang dibentuk dari 10 item barang data knapsack problem pada tabel 3.1. Kesepuluh item barang tersebut membentuk sebuah kromosom atau individu, dimana kromosom tersebut disusun dari beberapa gen yang berisi nilai. Nilai gen ditentukan berdasarkan item barang yang dipilih atau tidak. Item barang yang dipilih diberi tanda 1 sedangkan item barang yang tidak dipilih diberi tanda 0. Kedua tanda tersebut dapat direpresentasikan seperti pada gambar 1

0	0	1	0	1	0	0	1	1	0
0	0	1	0	1	0	0	1	1	0
1	1	0	0	0	1	0	1	0	1
1	0	1	0	1	0	0	1	1	0
0	1	1	0	1	0	0	1	1	0
0	1	1	1	1	0	0	1	1	0

Gambar 1 Representasi Kromosom Pada Knapsack Problem

Nilai Fitness

Nilai finess dapat dihitung dengan menjumlahkan nilai semua barang yang dipilih, tetapi dibatasi berat maksimalnya. Nilai finess dan berat total dapat dihitung dengan menggunakan rumus:

$$F = \sum_{i=1}^{n} b_i \; v_i \qquad (1)$$

$$W_{tot} = \sum_{i=1}^{n} b_i \; w_i \qquad (2)$$

3. Hasil Dan Pembahasan

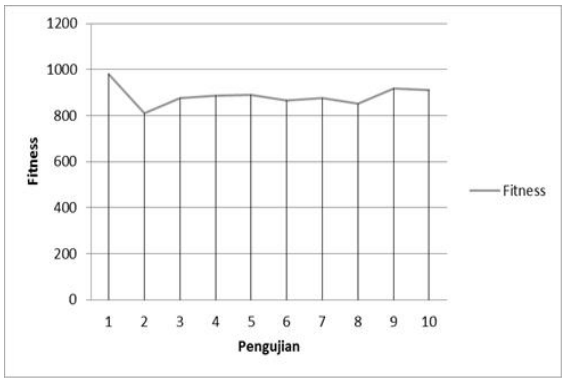
Pada penelitian ini ditampilkan hasil pengujian algoritma genetika klasik dan penerapan FLC pada algoritma genetika. Penilaian dilakukan terhadap nilai maksimum yang didapat dan kecepatan menemukan solusi terbaik pada generasi keberapa. Adapun hasil pengujian akan ditampilkan dalam bentuk tabel dan grafik.

Pada penerapan FLC dalam algoritma genetika nilai mutasi ditentukan atau dikendalikan oleh FLC. Parameter yang digunakan untuk penerapan FLC pada algoritma genetika adalah sebagai berikut:

1. Jumlah Individu = 10, 20, dan 50
2. Batas Generasi = 50, 80 dan 100
3. Probabilitas persilangan = 0.5
4. Probabilitas mutasi dikendalikan oleh FLC

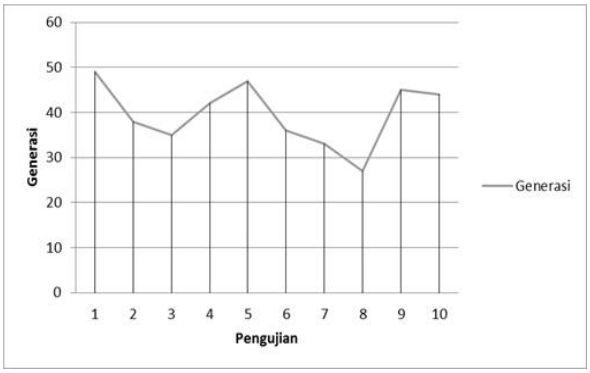
Tabel 2 Hasil pengujian pada 10 Individu dan 50 generasi

Pengujian	Generasi	Fitness
1	49	980
2	38	810
3	35	875
4	42	887
5	47	890
6	36	867
7	33	877
8	27	852
9	45	917
10	44	910



Gambar 2 Grafik pengujian untuk fitness

Pada tabel 2 terlihat bahwa nilai fitness tertinggi terjadi pada pengujian ke 1 sebesar 980 dan generasi ke 49. Sedangkan nilai terendah terdapat pada pengujian ke 5 sebesar 810 dan generasi ke 38. Dibandingkan dengan hasil pada tabel 3 menggunakan algoritma genetika klasik terjadi perubahan hasil nilai fitness setelah diterapkan FLC pada algoritma genetika.



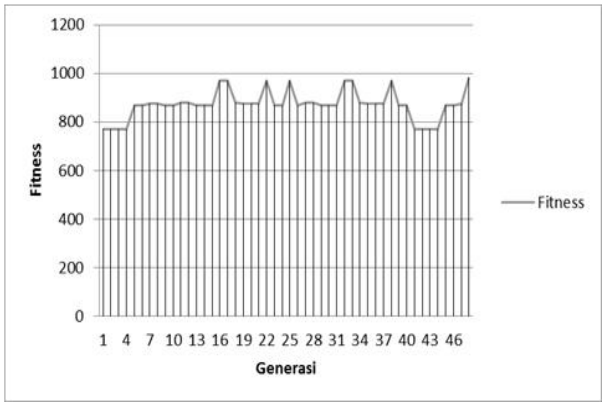
Gambar 3 Grafik pengujian untuk generasi

Berdasarkan nilai fitness terbaik pada tabel 2 yaitu sebesar 877 maka pada tabel 3.2 ditampilkan proses pencarian fitness terbaik per generasi serta solusi.

Tabel 3 Pencarian fitness terbaik per generasi

Generasi	Fitness	Solusi
1	770	10011010001110010011 10 1 3 5 7 11 12 13 16 19 20 21 (item barang terpilih)
2	770	
3	770	
4	770	
5	870	
6	870	
7	875	
8	875	
9	870	
10	870	
11	880	
12	880	
13	870	
14	870	
15	870	
16	970	
17	970	
18	880	
19	877	
20	877	
25	877	
30	970	
35	870	
40	870	
49	980	

Pada tabel 3 terlihat bahwa pencarian fitness terbaik terjadi pertama di generasi ke 49, meskipun generasi berikutnya bernilai fitness sama. Solusi merupakan item-item barang yang dipilih.



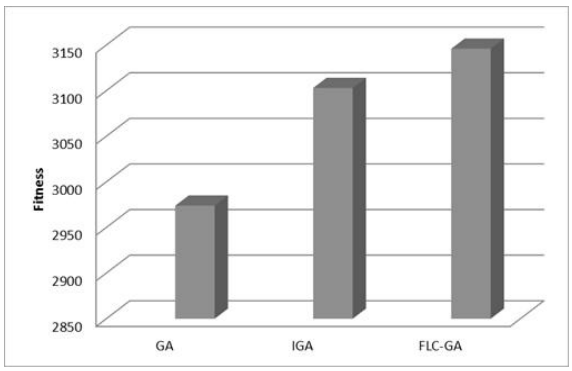
Gambar 4 Grafik fitness per generasi

Berikut ini akan ditampilkan hasil pengujian dalam tabel 4.

Tabel 4 Perbandingan Hasil Pengujian

Data	GA	IGA	FLC-GA
Knapsack Problem Haibo et al (2011) (50 items)	2974	3103	3146

Pada tabel 4 terlihat bahwa hasil pengujian pada algoritma genetika klasik (GA), Improvement of Genetic Algorithm (IGA) dan algoritma genetika berbasis *Fuzzy Logic Controller* (FLC) menggunakan data penelitian Haibo et al (2011) maka ada peningkatan nilai fitness atau nilai value item yang dapat dimasukkan ke dalam ransel (knapsack problem). Pada Gambar 5 menunjukkan hasil pengujian pada GA, IAG dan FLC-GA



Gambar 5 Grafik perbandingan hasil pengujian

4. Kesimpulan

Berdasarkan pembahasan serta pengujian yang dilakukan pada penelitian ini, maka kesimpulan yang dapat diambil adalah:

1. Dari hasil penelitian yang telah terlihat adanya peningkatan kinerja algoritma genetika setelah diterapkan *Fuzzy Logic Controller* (FLC) pengendalian *mutation rete* algoritma genetika serta terhindar dari lokal optima
2. Berdasarkan pengujian yang telah dilakukan pada algoritma genetika klasik (GA), Improvement of Genetic Algorithm (IAG) dan algoritma genetika berbasis *Fuzzy Logic Controller* (FLC-GA) sebanyak 10 kali percobaan, maka hasil percobaan GA dengan nilai fitness atau value item yang dapat dimasukkan ke dalam knapsack sebesar 2974, IAG sebesar 3103 dan FLC-GA sebesar 3146.
3. Algoritma genetika berbasis FLC dalam penelitian ini terlihat memberikan pengaruh dalam hal waktu proses pencarian solusi terbaik.

5. Referensi

[1] De Falco, Della, A., Ciopa. & Tarantino, E. 2002. Mutation-based genetic algorithm: performance evaluation. Applied Soft Computing1:285-299

[2] Gen, M. & Cheng, R. 1997. Genetic Algorithms & Engineering Design. Wiley-Interscience: New York.

[3] Haibo, Z., Liwen, C., Shenyong, G., Jianguo, C., Feng, Y., & Daqing, L. 2011. Improvements of Genetic Algorithm to the Knapsack Problem. ICAIC. Part I: 202-206

[4] Holland, J.H. 1975. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, University of Michigan Press, Ann Arbor, Mich: USA.

[5] Katayama, K. & Sakamoto, H. 2000. The efficiency of Hybrid Mutation Genetic Algorithm for the Travelling Salesmen Problem. Mathematical and Computer Engineering 31:197-203

[6] Mühlenbein, H. 1992. How Genetic Algorithms Really Work I. Mutation and Hillclimbing. In Proceedings of the 2ndConference on Parallel Problem Solving from Nature, pp. 15–29.

[7] Passino, K.M, Yurkovich, S. 1998. Fuzzy Control. Edison Wesley Longman Inc: Ohio.

[8] Sakawa, M. 2002. Genetic Algorithm and Fuzzy Multiobjective Optimizaton. Springer: Jepan.

[9] Singh, R.P. 2011. Solving 0-1 Knapsack Problem Using Genetic Algorithm. IEEE. 11:591-585

[10] Sivanandam, S.M, &Deepa, S.N. 2008. Introduction to Genetic Algorithms, Springer, Berlin:Germany.

[11] Smith, J.E. 2002. Handbook of Global Optimization. Volume 2, 275-362