Jurnal Vokasi Teknik Mesin dan Fabrikasi Logam
Vol. 1, No. 1, Oktober 2022
e-ISSN: 2985-4547

PERANCANGAN APLIKASI PENERAPAN ALGORITMA GENETIKA UNTUK PENCARIAN SOLUSI PERSAMAAN LINIER

[1]Abdul Tahir, [2]Ahyar M, [3]Israkwaty
[1,2,3]Teknik Mesin, Akademi Teknik Soroako, Soroako, 92984, Indonesia
Email : [1]abdultahir0101@gmail.com , [2]ahyar@ats-sorowako.ac.id, [2]israwaty@ats-sorowako.ac.id

A B S T R A K

Algoritma genetika merupakan salah satu evolusi perkembangan dunia komputer dalam bidang kecerdasan buatan. Dalam penelitian ini algoritma genetika digunakan untuk mencari sejumlah solusi dari sebuah persamaan linier yang memiliki empat variabel. Dalam proses genetika keempat variabel tersebut merupakan representasi solusi yang juga dijadikan gen-gen penyusun ukuran kromosom. Sebanyak sepuluh kromosom dibentuk menjadi sebuah populasi yang akan berevolusi untuk mencari solusi permasalahan. Dalam proses berevolusi digunakan tiga buah operator algoritma genetika yaitu seleksi, penyilangan, dan mutasi. Metode seleksi yang digunakan adalah *Roulette wheel*, operator penyilangan dilakukan dengan memberikan crossover rate sebesar 40%, dan percobaan dilakukan dengan memberikan variasi mutation rate yaitu 0%, 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, 50%, dan 55%. Dalam setiap percobaan proses evoluasi dibatasi pada generasi ke-200. Dari percobaan yang dilakukan diperoleh hasil solusi pada variasi dengan mutation rate 25% mendapatkan solusi sebanyak 4 solusi dengan solusi tercepat diperoleh pada generasi ke-5. Hal ini menunjukkan bahwa perubahan mutation rate berpengaruh pada kinerja algoritma genetika.

Kata kunci: Algoritma, genetika, mutasi, penyilangan, seleksi, optimasi.

A B S T R A C T

Genetic algorithm is one of the evolutionary developments in the world of computers in the field of artificial intelligence. In this research, a genetic algorithm is used to find a number of solutions to a linear equation that has four variables. In the genetic process, these four variables represent solutions which are also used as genes that make up chromosome size. As many as ten chromosomes are formed into a population that will evolve to find solutions to problems. In the evolution process, three genetic algorithm operators are used, namely selection, crossover, and mutation. The selection method used is the Roulette wheel, the crossover operator is carried out by providing a crossover rate of 40%, and the experiment is carried out by providing variations in the mutation rate, namely 0%, 5%, 10%, 15%, 20%, 25%, 30%, 35 %, 40%, 45%, 50%, and 55%. In each experiment the evolution process is limited to the 200th generation. From the experiments conducted, the results obtained for the variation with a mutation rate of 25% obtained 4 solutions with the fastest solution obtained in the 5th generation. This shows that changes in the mutation rate affect the performance of the genetic algorithm.

Keywords: *Algorithm, genetics, mutation, crossover, selection, optimization.*

Submit:	Revised:	Accepted:	Available online:
30.09.2022	06.10.2022	26.10.2022	31.10.2022

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

PENDAHULUAN

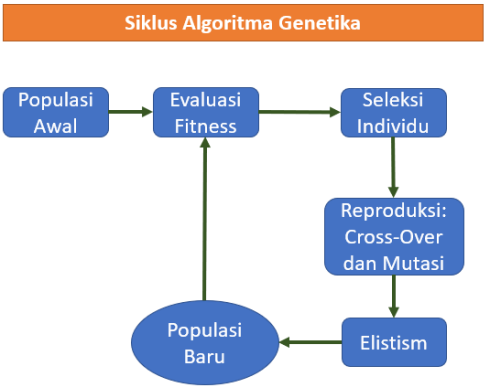
Algoritma Genetika (Genetic Algorithm, GA) salah satu cabang dari ilmu *Artificial Intelligence* (AI). John Holland (penemu algoritma genetika) mengatakan bahwa semua masalah dengan bentuk adaptasi (buatan atau alami) dapat diformulasikan dalam terminologi genetika [1]. GA juga sering digunakan pada penyelesaian masalah optimasi, pencarian solusi seperti pada kasus pencarian nilai akar dari suatu fungsi [2] dan nilai variable dari sebuah fungsi linier. Dalam masalah optimasi GA digunakan sebagai pembanding dengan metode lain seperti metode Newton-Raphson [3]. Penerapan GA dalam pemrograman linear dan nonlinear kasus memaksimumkan dan meminimumkan fungsi tujuan dapat juga memberikan hasil yang maksimal [4].

Pada penelitian ini AG digunakan dalam mencari solusi sebuah kombinasi linier matematika dengan menerapkan metode operator AG dengan perumusan masalah sebagai berikut :

a. Bagaimana mencari sejumlah solusi *variable* a,b,c,d yang memenuhi persamaan 2a+3b -4c+5d = 40.
b. Bagaimana merancang aplikasi simulasi untuk dapat memperoleh solusi-solusi tersebut. Merancang aplikasi simulasi dengan menggunakan bahasa pemrograman Microsoft visual basic dan menerapkan teknik algoritma genetika untuk : (1) Merancangaplikasi untuk pencarian solusi dari kombinasi linier sebuah persamaan matematik 2a+3b-4c+5d =40, dimana akan dicari nilai a,b,c,d yang memenuhi persamaan tersebut. (2) Mencari pengaruh operator mutasi dengan memberikan variasi mutation rate (MR).

Algoritma Genetik (AG) merupakan evolusi/perkembangan dunia komputer dalam bidang kecerdasan buatan. AG adalah algoritma pencarian yang berdasarkan pada mekanisme sistem natural yakni genetik dan seleksi alam. Dalam aplikasi AG, variabel solusi dikodekan kedalam struktur string yang merepresentasikan barisan gen, yang merupakan karakteristik dari solusi problem.

AG dibentuk dari himpunan sejumlah solusi yang didapatkan secara acak, yang disebut populasi, sementara individu dalam populasi dinamakan kromosom yang menjadi representasi sebuah solusi. Dalam implementasinya sebuah kromosom dibuat secara acak yang biasanya disusunan dari sejumlah angka berbentuk binary atau real yang digenerate dan untuk dipilih. Nantinya kromosom ini akan berevolusi dalam beberapa tahap perulangan atau iterasi yang disebut dengan generasi. Setiap generasi baru atau *offsprings* disusun dengan sistim kawin silang atau *crossover* serta mutasi atau *mutation*. Proses crossoverterdiri dari pemecahan atau *splitting* terhadap dua kromosom, selanjutnya dilakukan kombinasi terhadap sebagian dari masing-masing kromosom dengan pasangan-pasangan lainnya. Sedangkan mutasi meliputi penggantian (*flipping*) satu bit (bagian) dari kromosom dengan satu bagian lain dari kromosom lain yang menjadi pasangannya. Kromosom-kromosom ini selanjutnya berevolusi dengan suatu kriteria kesesuaian (*fitness*) yang ditetapkan dan hasil terbaik akan dipilih sementara yang lainnya diabaikan. Selanjutnya, proses dilakukan berulangulang sampai didapatkan kromosom yang mempunyai kesesuaian terbaik *(best fitness*) yang akan diambil sebagai solusi terbaik dari permasalahan. Gambar 1 berikut ini adalah siklus AG dalam mencari sebuah solusi.



Gambar 1. *Siklus algoritma Genetika*

a. Representasi Kromosom

Cara merepresentasikan permasalahan dalam kromosom merupakan suatu hal yang penting dalam AG. Ada beberapa model representasi kromosom yang dapat dipergunakan untuk menyelesaikan suatu masalah, salah satunya adalah operation based representation. Prinsip dari Operation-Base Representation adalah semua operasi pada suatu pekerjaan akan dikodekan dengan simbol yang sama, kemudian diinterpretasikan menurut urutannya dalam sequence. Sehingga untuk problem dengan a-pekerjaan bmesin,kromosom akan terdiri dari a x b gen. Dimana setiap pekerjaan akan mengalami pengulangan sebanyak m kali.

b. Fungsi Evaluasi

Fungsi evaluasi merupakan dasar untuk proses seleksi. Langkah-langkahnyayaitu string dikonversi ke parameter fungsi, fungsi objective-nya dievaluasi, kemudian mengkonversi fungsi objektif tersebut ke dalam fitness. Dimana untuk maksimasiproblem, fitness sama dengan fungsi objective-nya [5]. Output dari fungsi fitnessdipergunakan sebagai dasar untuk menyeleksi individu pada generasi berikutnya.

c. Seleksi

Roulette whell (cakram rolet) merupakan salah satu metode seleksi untuk menyeleksi populasi baru dengan distribusi probabilitas yang berdasarkan nilai fitness [6]. Teknik seleksi ini diilustrasikan dimana setiap kromosom dalam populasi menempati suatu slot pada cakram rolet. Besarnya ukuran slot adalah sama dengan rasio antara nilai fitness suatu kromosom dengan total nilai fitness semua kromosom. Untuk menghasilkan sejumlah populasi, rolet tersebut diputar sebanyak ukuran populasi yang ada.

d. Operator Genetik

Operator genetik dipergunakan untuk mengkombinasi (modifikasi) individu dalam aliran populasi guna mencetak individu pada generasi berikutnya. Terdapat dua operator genetik pada AG yang umum digunakan yaitu *crossover* (penyilangan) dan *Mutation* (mutasi).

1) Crossover (penyilangan)

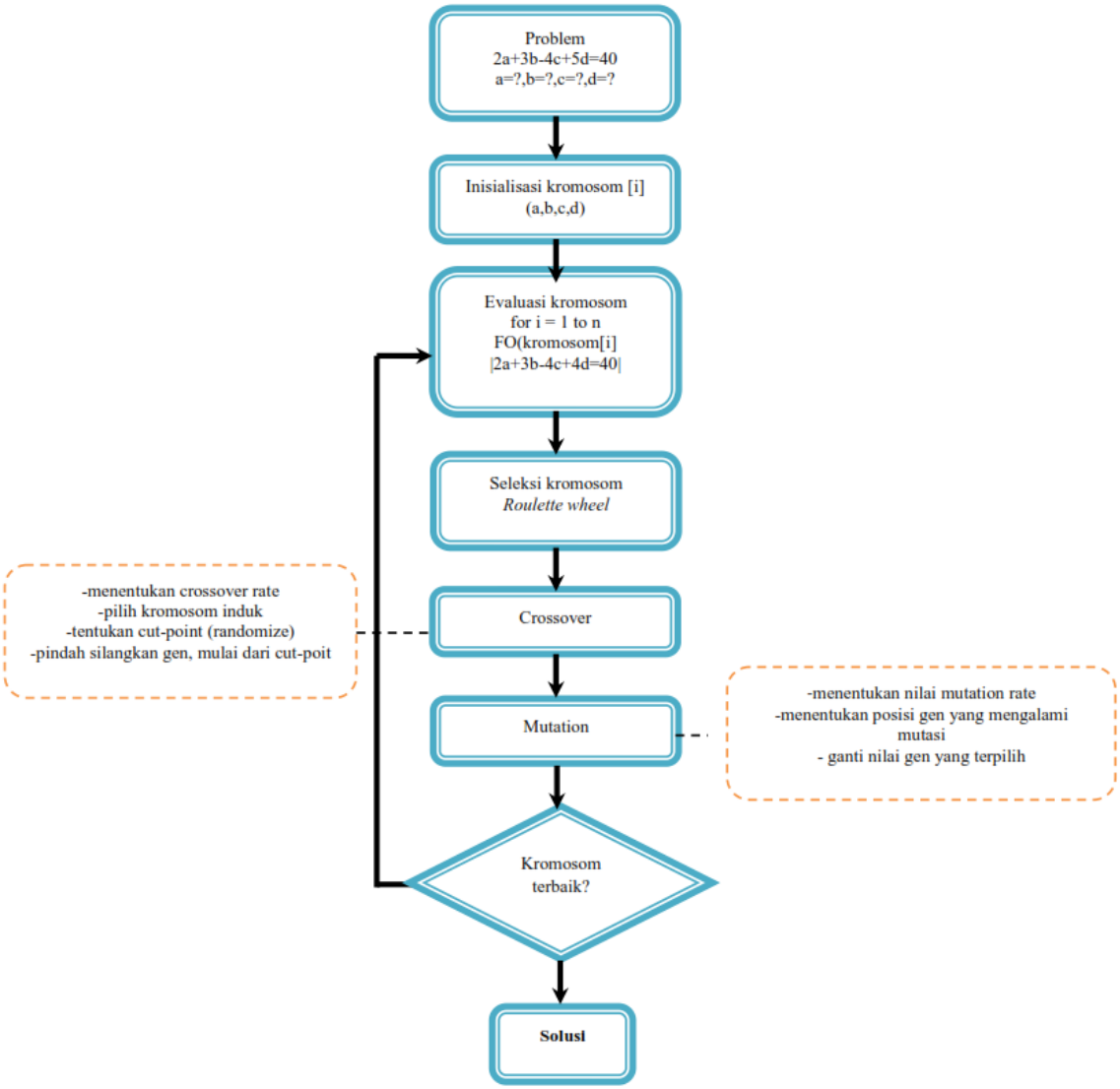
Operator penyilangan adalah operator primer yang bekerja pada sepasang kromosom induk untuk menghasilkan dua kromosom anak dengan menukarkan beberapa elemen (gen) yang dimiliki masing masing kromosom induk. Holland (1975) memperkenalkan operator penyilangan yang disebut penyilangan satu titik . Sejalan dengan perkembangan penelitian muncul beberapa teknik penyilangan terbaru seperti *partially-mapped-crossover* (PMX), *order crossover* (OX), *cycle-crossover* (CX), *position-based crossover* dan *heuristic-crossover* .

2) Mutation

Operator Mutasi adalah operator penunjang atau pendukung dalam proses AG. Operator ini berperan mengubah benyuk kromoson dengan tiba-tiba. Perubahan tersebut akan mengakibatkan terbentuknya sebuah kromosom baru yang secara genetik berbeda dari kromosom sebelumnya. Operator mutasi bekerja pada satu kromosom, tidak pada sepasang kromosm seperti yang dilakukan pada operator penyilangan. Tujuan dari operator mutasi ini adalah : (1) mengembalikan gen gen yang hilang pada generasi generasi sebelumnya, dan (2) memunculkan gen gen baru yang belum pernah muncul pada generasi generasi sebelumnya. Dalam beberapa tahun belakangan ini para peneliti telah memperkenalkan beberapa jenis operator mutasi seperti mutasi inversi, mutasi insersi, displacement mutation dan reciprocal exchange mutation .

METODE PENELITIAN

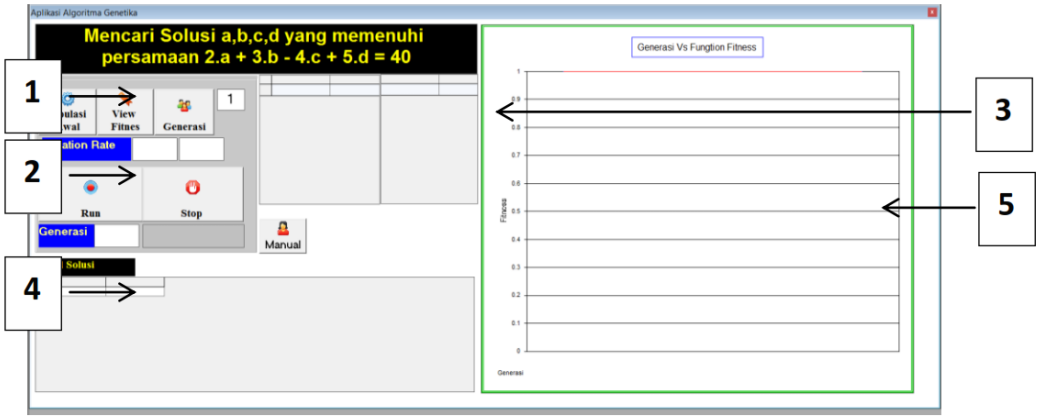
Metode yang digunakan pada penelitian ini adalah mengimplementasikan aturan dan teknik teknik algoritma genetika dengan urutan seperti yang digambarkan berikut ini.



Gambar 2. *Diagram Alur Metode Penelitian*

Penjelasan dari gambar 2 diatas adalah sebagai berikut :

1) Menetapkan permasalahan, yakni bagaimana mencari solusi nilai a,b,c,d yang memenuhi persamaan 2a+3b-4c+5d = 40, dimana 0<|a,b,c,d|<41.
2) Menentukan ukuran kromosom dan populasi, membangkitkan populasi awal, populasi awal ini dibangkitkan secara random sehingga didapatkan solusi awal.Populasi itu sendiri terdiri dari sepuluh kromosom yang merepresentasikan solusi yangdiinginkan.
3) Membentuk generasi baru, dalam membentuk generasi baru digunakan tiga operator yaitu seleksi dengan Roulette wheel, penyilangan dan mutasi. Proses seleksi dilakukan untuk memilih kromosom yang mempunyai fungsi objectif kecil, dalam hal ini kromosom yang memiliki fungsi objectif kecil akan memiliki peluang yang besar untuk terpilih karena memiliki probabilitas yang tinggi. Adapun penetapan fungsi fitness adalah fitness = 1/(1+fungsi_objectif). Fungsi objektif perlu ditambah 1 untuk menghindari error saat program dijalankan karena ada kemungkinan terjadi pembagian oleh bilangan nol. Setiap perubahan generasi akan dilakukan evaluasi untuk menghitung nilai fitness setiap kromosom sampai terpenuhi kriteria berhenti.
4) Perancangan Antarmuka(*Interface*), Dalam mengimplementasi alagoritma genetika untuk mencari solusi dari permasalahan, Penulis merancang antarmuka dengan pemrograman *Visual Basic* seperti gambar 3 dibawah ini :



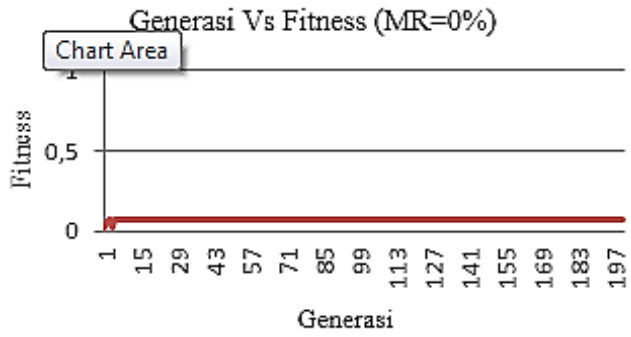
Gambar 3. *Antarmuka Aplikasi*

Keterangan gambar :
1. Tombol untuk pembangkit populasi awal kromosom
2. Tombol Perintah proses pembentukan generasi baru yang telah melalui proses genetika
3. Tabel untuk menampilkan populasi,kromosom,gen, fungsi objektif, dan fitness
4. Tabel untuk menampilkan Solusi
5. Grafik yang menampilkan hubungan Generasi dengan Fitness

HASIL DAN PEMBAHASAN

Setelah melakukan perancangan aplikasi dengan menerapkan aturan dan metode AG, selanjutnya dilakukan percobaan percobaan untuk mencari nilai variable a,b,c,d yang merupakan solusi dari persamaan 2a+3b-4c+5d = 40, dimana 0 < |a,b,c,d| < 41. Percobaan dilakukan dengan memberikan variasi pada mutation rate (MR) dengan maksud untuk mengetahu pengaruh dari MR tersebut. Dari percobaan yang dilakukan MR divariasikan yaitu : 0%, 5%, 10%, 15%, 20%, 25%, 35%, 40%, 45%, 50%, dan 55%, Prosentasi MR diambil dari rasio jumlah gen mutasi dibagi dengan total gen dalam populasi.

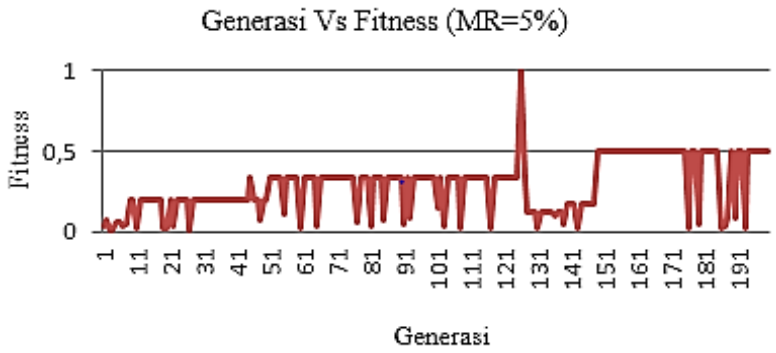
Pada percobaan 1, pencarian solusi dengan mutation rate = 0 (tidak ada proses mutasi) dan berhenti pada generasi ke 100. Hasilnya seperti grafik dibawah.



Gambar 4. *Generasi vs Fitness dengan MR = 0%*

Pada percobaan MR=0% artinya proses mutase tidak dilakukan. Sampai pada generasi ke -200 tidak ada solusi yang didapat.

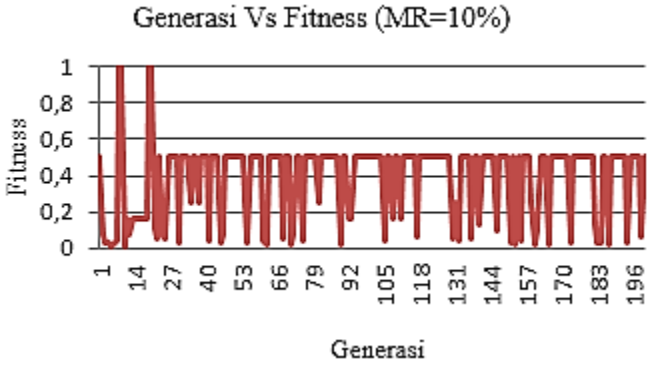
Pada percobaan 2, pencarian solusi dengan *mutation rate* = 5% dan berhenti pada generasi ke 100. Hasilnya seperti grafik dibawah.



Gambar 5. *Generasi vs Fitness dengan MR = 0%*

Pada percobaan dengan MR = 5% terlihat fitness menjadi lebih baik dari percobaan 1 dan pada generasi ke-126 telah didapatkan solusi.Hasil pada percobaan dengan MR = 5% : Jumlah solusi = 1, yaitu pada generasi ke-126.

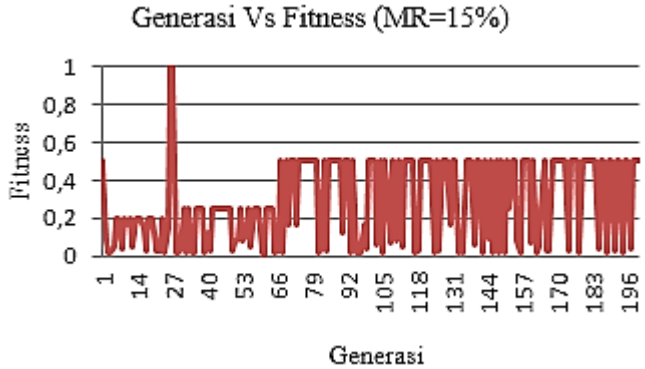
Pada percobaan 3, pencarian solusi dengan *mutation rate* = 10% dan berhenti pada generasi ke 100. Hasilnya seperti grafik dibawah.



Gambar 6. *Generasi vs Fitness dengan MR = 0%*

Hasil pada percobaan dengan MR = 10% : Jumlah solusi = 2, yaitu pada generasi ke-5, dan ke-25.

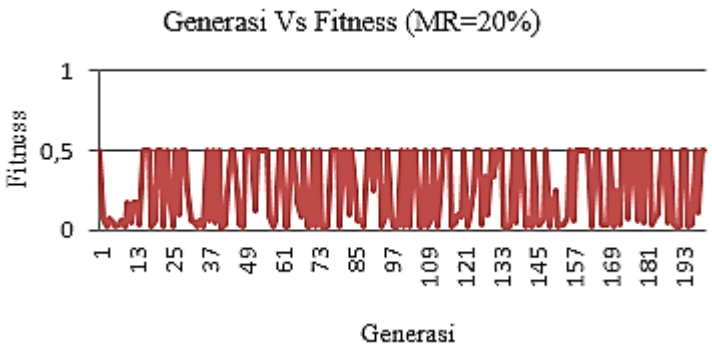
Pada percobaan 4, pencarian solusi dengan *mutation rate* = 15% dan berhenti pada generasi ke 100. Hasilnya seperti grafik dibawah.



Gambar 7. *Generasi vs Fitness dengan MR = 15%*

Hasil pada percobaan dengan MR = 15% : Jumlah solusi = 1, yaitu pada generasi ke-26.

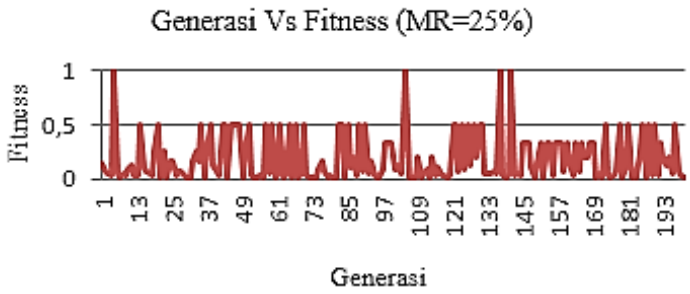
Pada percobaan 5, pencarian solusi dengan *mutation rate* = 20% dan berhenti pada generasi ke 100. Hasilnya seperti grafik dibawah.



Gambar 8. *Generasi vs Fitness dengan MR = 20%*

Pada percobaan dengan MR = 20% ,tidak ada solusi yang didapat.

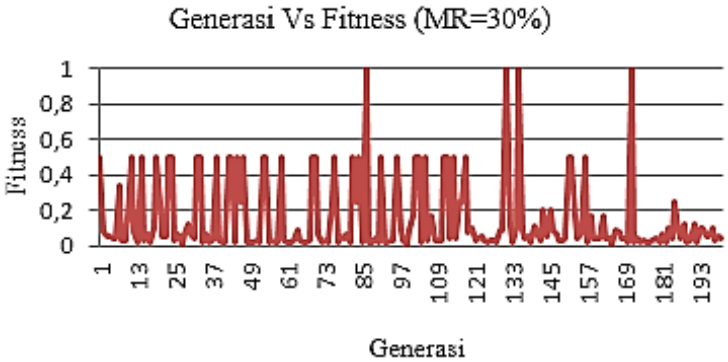
Pada percobaan 6, pencarian solusi dengan *mutation rate* = 25% dan berhenti pada generasi ke 100, seperti grafik dibawah.



Gambar 9. *Generasi vs Fitness dengan MR = 25%*

Hasil pada percobaan dengan MR = 25% : Jumlah solusi = 4, yaitu pada generasi ke-5, ke-105, ke -137, ke -147.

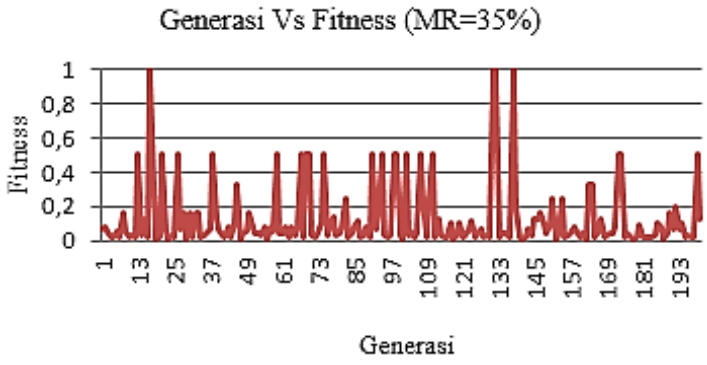
Pada percobaan 7, pencarian solusi dengan *mutation rate* = 30% dan berhenti pada generasi ke 100, seperti grafik dibawah.



Gambar 10. *Generasi vs Fitness dengan MR = 30%*

Hasil pada percobaan dengan MR = 30% : Jumlah solusi = 4, yaitu pada generasi ke-86, ke-131, ke -135, ke -171.

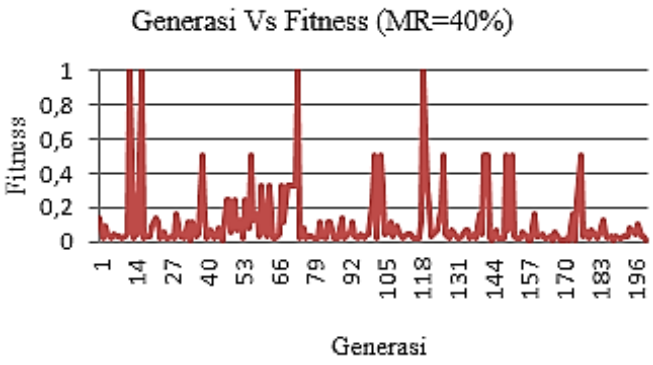
Pada percobaan 8, pencarian solusi dengan *mutation rate* = 30% dan berhenti pada generasi ke 100, seperti grafik dibawah.



Gambar 11. *Generasi vs Fitness dengan MR = 35%*

Hasil pada percobaan dengan MR = 35% : Jumlah solusi = 3, yaitu pada generasi ke-17, ke-131, ke -178.

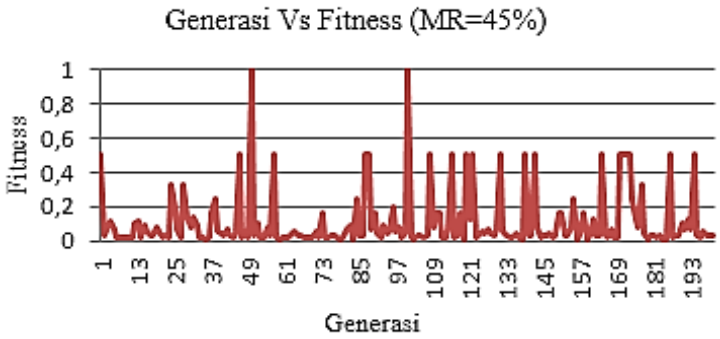
Pada percobaan 9, pencarian solusi dengan *mutation rate* = 40% dan berhenti pada generasi ke 100, seperti grafik dibawah.



Gambar 12. *Generasi vs Fitness dengan MR = 40%*

Hasil pada percobaan dengan MR = 40% : Jumlah solusi = 4, yaitu pada generasi ke-12, ke-16, ke -73, dan ke -119.

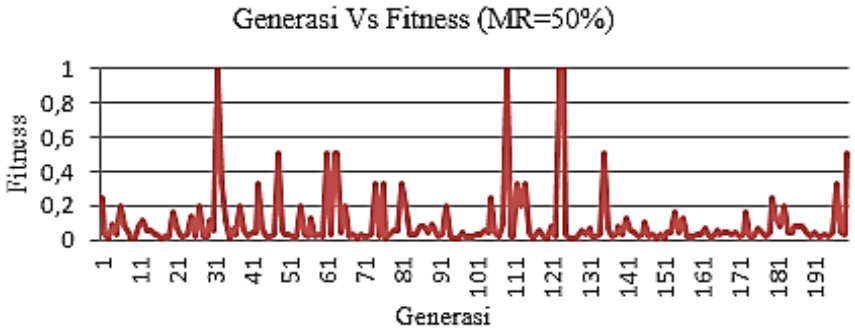
Pada percobaan 10, pencarian solusi dengan *mutation rate* = 45% dan berhenti pada generasi ke 100, seperti grafik dibawah.



Gambar 13. Generasi vs *Fitness dengan MR = 45%*

Hasil pada percobaan dengan MR = 45% : Jumlah solusi = 2, yaitu pada generasi ke-50, dan ke-101.

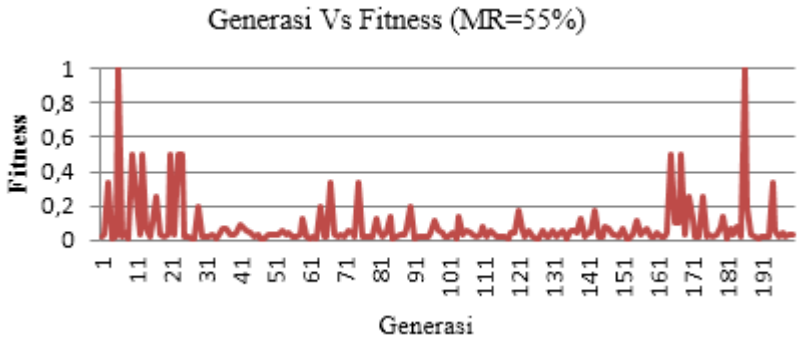
Pada percobaan 11, pencarian solusi dengan *mutation rate* = 50% dan berhenti pada generasi ke 100, seperti grafik dibawah.



Gambar 14. *Generasi vs Fitness dengan MR = 50%*

Hasil pada percobaan dengan MR = 50% : Jumlah solusi = 4, yaitu pada generasi ke-32, ke-109, ke-123, dan ke-124.

Pada percobaan 12, pencarian solusi dengan *mutation rate* = 55% dan berhenti pada generasi ke 100, seperti grafik dibawah.



Gambar 15. *Generasi vs Fitness dengan MR = 55%*

Hasil pada percobaan dengan MR = 55% : Jumlah solusi = 2, yaitu pada generasi ke-6, dan ke-186.

Dari percobaan yang telah dilakukan didapatkan hasil sebagaimana tabel 1 dibawah ini:

Tabel 1 hasil percobaan variasi MR

No	Mutation rate (%)	Jumlah Solusi	Solusi Pertama pada generasi ke - ?
1	0	0	-
2	5	1	126
3	10	2	5
4	15	1	26
5	20	0	-
6	25	4	5
7	30	4	86
8	35	3	17
9	40	4	12
10	45	2	50
11	50	4	6
12	55	2	126

Dari hari percobaan seperti pada tabel 1 diatas, dapat dilihat bahwa penggunaan *mutation rate* 25% memberikan kinerja yang lebih baik dimana pada generasi ke-5 sudah menghasilkan solusi pertama dan jumlah solusi yang didapatkan setelah ber-iterasi selam 200 kali sebanyak 4 solusi.

KESIMPULAN

Setelah melakukan percobaan percobaan tentang penerapan algoritma genetika dalam mencari solusi yang memenuhi persamaan 2a+3b-4c+5d = 40, maka dapa ditarik kesimpulan sebagai berikut :

1. Penarapan algoritma genetika dapat digunakan untuk mencari solusi sebuah kombinasi linier fungsi matematika.
2. Operator seleksi, penyilangan dan mutasi berperan penting dalam meningkatkan kinerja algoritma genetika dalam mendapatkan solusi permasalahan kombinasi linier fungsi matematika.
3. Kinerja algoritma genetika terbaik diperoleh pada variasi *mutation rate* 25%.

REFERENSI

[1] Holland, J. H., *Genetic Algorithm.* USA: Scientific American.,1992

[2] Yusuf Akhmad dan Soesanto Oni, "Algoritma Genetika Pada Penyelesaian Akar Persamaan Sebuah Fungsi ", *Jurnal Matematika Murni dan Terapan",* Vol.6 No.2 Desember 2012 : 47-56

[3] Kania Evita Dewi, "Perbandingan Metode Newton-Raphson Dan Algoritma Genetik Pada Penentuan *Implied Volatility* Saham ", *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)",* Vol.1 No.2 Oktober 2012 : 9-16

[4] Kania Evita Dewi, "Algoritma Genetika Pada Pemrograman Linear dan Nonlinear", *Buletin Ilmiah Mat. Stat. dan Terapannya (Bimaster),* Vol.5 No.3, 2016 : 265-274

[5] Gen, M., & Cheng, R., *Genetic Algorithm and Engineering Design.*Willey & Song, Inc.,1997

[6] E.Goldberg, D., *Genetic Algorithm in Search, Optimazion, and Machine Learning.* USA: Addison-Wesley, Reading,MA., 1989