

Algoritma Genetika dan Penerapannya dalam Mencari Akar Persamaan Polinomial

Muhammad Abdy^{*1}, Maya Sari Wahyuni^{*2}, Nur Ilmi^{*3}

^{1,2,3}Jurusan Matematika, Universitas Negeri Makassar

e-mail: ^{*1}m.abdy@unm.ac.id, ^{*2}mayasari.wahyuni@gmail.com, ^{*3}nurilmi2392@gmail.com

Abstrak

Algoritma Genetika (AG) adalah algoritma pencarian untuk menyelesaikan masalah yang didasarkan pada evolusi genetika yang terjadi pada makhluk hidup. Masalah yang akan dibahas pada penelitian ini adalah pencarian akar persamaan polinomial dengan menggunakan algoritma genetika. Tujuan penulisan skripsi ini adalah menjelaskan implementasi algoritma genetika dengan metode seleksi roulette wheel untuk mencari akar persamaan polinomial. Langkah-langkah pencarian akar persamaan polinomial menggunakan algoritma genetika adalah membangkitkan populasi secara random, evaluasi fitness tiap individu, seleksi kromosom dengan metode roulette wheel, melakukan crossover pada kromosom yang terpilih, mutasi gen pada kromosom yang terpilih, menyusun populasi baru sampai memperoleh individu dengan nilai fitness optimum. Hasil implementasi algoritma genetika untuk mencari akar persamaan dari sebuah fungsi $p(x) = (-2) + x + 10x^2 - 5x^3 - 8x^4 + 4x^5$ pada interval $[0,3]$ didapat bahwa pada generasi pertama kromosom kedua belas telah dihasilkan fitness = 1,000, dengan kromosom $v_{12}'' = 10101010$ yang akarnya adalah $x_{12} = 2,0000$.

Kata Kunci: Algoritma Genetika, Akar Persamaan, Fungsi Polinomial

1. PENDAHULUAN

Algoritma Genetika (AG) adalah algoritma pencarian untuk menyelesaikan masalah yang didasarkan pada evolusi genetika yang terjadi pada makhluk hidup. Proses komputasi yang terjadi pada AG dianalogikan dengan proses seleksi makhluk hidup dalam sebuah populasi. Salah satu aplikasi AG adalah pada permasalahan optimasi, yaitu mendapatkan suatu nilai solusi optimal terhadap suatu permasalahan yang mempunyai banyak kemungkinan solusi. Penelitian ini membahas masalah optimasi yang lebih dikhususkan pada kasus pencarian nilai akar dari polinomial.

Secara matematis, perhitungan akar persamaan polinomial dilakukan untuk menemukan nilai sebuah fungsi yang sama dengan nol, namun hingga saat ini masih banyak ditemukan kesulitan-kesulitan dalam pencarian akar fungsi polinomial yang berderajat lebih tinggi secara analitik, juga tidak tersedia metode aljabar untuk mendapatkan penyelesaiannya, sehingga dibutuhkan suatu metode aproksimasi.

Penelitian ini bertujuan untuk dapat memperkuat dan menambah wawasan penulis serta pembaca tentang cara mengkaji dan mencari akar persamaan fungsi polinomial dengan menggunakan Algoritma Genetika serta aplikasinya dalam *software Microsoft Visual Basic 2010*.

2. TINJAUAN PUSTAKA

1. Akar Persamaan Polinomial

Suatu akar persamaan polinomial $p(z) = 0$ adalah sebuah bilangan z_i sedemikian sehingga $p(z_i) = 0$.

Teorema dasar aljabar mengatakan bahwa suatu polinomial $p(z)$ berderajat n buah akar yang beberapa diantaranya dapat digeneralisasikan. Sebagai contoh akar-akar persamaan polinomial dari $2 + x - 2x^2 + x^3 = (x - 2)(x + 1)(x - 1)$ adalah 2, -1, dan 1.

Akar-akar suatu persamaan biasanya juga dikenal sebagai penyelesaian dari suatu persamaan yaitu nilai-nilai yang tidak diketahui yang membuat dua pernyataan menjadi sama. Suatu bilangan z_i disebut pembuat nol.

Jika untuk $z = z_i$ nilai $p(z)$ adalah 0. Pembuat nol $p(z)$ merupakan selesaian atau akar persamaan $p(z) = 0$. Pada persamaan linier $ax + b = 0$ dimana a dan b adalah konstanta tertentu dan $a \neq 0$ maka $x = -\frac{b}{a}$ adalah akarnya dan merupakan satu-satunya penyelesaian dari $ax + b = 0$. Contoh lain yang sederhana adalah persamaan kuadrat $ax^2 + bx + c = 0$ dimana a, b dan c merupakan bilangan bulat dan $a \neq 0$ maka terdapat dua akar yaitu $x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ dan $x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$ tentu saja jika $b^2 = 4ac$ maka kedua akar tersebut identik.

2. Probabilistik

Probabilitas didefinisikan sebagai peluang atau kemungkinan dari suatu kejadian, suatu ukuran tentang kemungkinan atau derajat ketidakpastian suatu peristiwa (*event*) yang akan terjadi di masa mendatang. Rentang probabilitas antara 0 sampai dengan 1. Jika probabilitas sebuah peristiwa adalah 0, maka peristiwa tersebut tidak mungkin terjadi atau nihil. Jika probabilitas sebuah peristiwa adalah 1 maka peristiwa tersebut pasti terjadi. Jumlah antara peluang suatu kejadian yang mungkin terjadi dan peluang suatu kejadian yang mungkin tidak terjadi adalah satu, jika kejadian tersebut hanya memiliki dua kemungkinan kejadian yang mungkin akan terjadi.

3. Frekuensi

Dalam statistik, "frekuensi" mengandung pengertian angka (bilangan) yang menunjukkan seberapa kali suatu variabel (yang dilambangkan dengan angka-angka itu) berulang dalam deretan angka tersebut; atau berapa kalikah suatu variabel (yang dilambangkan dengan angka itu) muncul dalam deretan angka tersebut. Sedangkan frekuensi kumulatif yaitu didapatkan dengan menambahkan total seluruh frekuensi dari kelas interval pertama sampai kelas interval yang dimaksud. Adapun rumus menghitung frekuensi kumulatif $fk_i - i$ dapat dilihat pada persamaan

$$fk_i = f_1 + f_2 + \dots + f_i \quad ; \quad i = 1, 2, 3, 4, \dots \tag{3.1}$$

Keterangan:

$$fk_i = \text{frekuensi kumulatif kelas ke } i$$

$$f_i = \text{frekuensi kelas ke } i$$

4. Algoritma Genetika

AG pertama kali ditemukan oleh John Holland, yang dapat dilihat dari bukunya yang berjudul *Adaption in Natural and Artificial Systems* pada tahun 1960-an dan kemudian dikembangkan bersama murid dan rekan kerjanya di Universitas Michigan pada tahun 1960-an sampai 1970-an. Kemudian dipopulerkan oleh David Goldberg pada tahun 1980-an yang kemudian dalam bukunya, David Golberg mendefinisikan AG sebagai algoritma pencarian yang didasarkan pada mekanisme seleksi alamiah dan genetika alamiah. AG merupakan metode pencarian yang disesuaikan dengan proses genetika dari organisme-organisme biologi yang berdasarkan pada teori evolusi Charles Darwin (Kusumadewi: 2003).

Pada proses evolusi alamiah, dikenal adanya seleksi alamiah. Di alam, kromosom yang terbaik akan bertahan hidup dan memiliki kemampuan untuk mempertahankan diri dalam populasi sehingga generasi berikutnya akan lebih baik karena kromosom pada generasi tersebut

diturunkan dari induk yang baik pula. Nilai optimum yang didapat dari AG disamakan dengan keturunan yang mempunyai kemampuan mempertahankan diri paling tinggi yang dicari dalam seluruh populasi dari awal hingga akhir (Zukhri: 2014).

Pemetaan proses alamiah ke dalam proses komputasi AG dapat dirangkum dalam tabel

Tabel 1. Pemetaan Proses Alamiah ke Proses Komputasi

Proses Alamiah	Proses Komputasi
Individu	Penyelesaian masalah
Populasi	Himpunan penyelesaian
Fitness / kebugaran	Kualitas penyelesaian
Kromosom	Kode / representasi penyelesaian
Gen	Bagian dari representasi penyelesaian
Pertumbuhan	Pendekodean representasi penyelesaian
Penyilangan	Operator genetika
Mutasi	Operator genetika
Seleksi alam	Menyeleksi penyelesaian masalah (sementara) berdasarkan kualitasnya

(Sumber: Zukhri, 2014)

3. METODOLOGI PENELITIAN

Jenis penelitian yang digunakan adalah kajian teori sekaligus terapan/aplikasi, yaitu penelitian yang bertujuan untuk menyelesaikan masalah yang ada dengan terlebih dahulu menyusun konsep-konsep yang diperlukan dalam mencari akar persamaan polinomial dengan menggunakan AG. Penelitian ini selain bersifat kajian literatur dalam artian mengumpulkan informasi dari beberapa referensi buku, jurnal, dan internet. Secara garis besar peneliti memfokuskan kajiannya pada penggunaan AG dalam mencari akar persamaan polinomial.

Prosedur penelitian yang diterapkan dalam penulisan ini guna mencapai tujuan penelitian meliputi langkah-langkah umum yang dilakukan adalah sebagai berikut:

1. Mengumpulkan sumber-sumber informasi/materi yang dibutuhkan dalam penelitian, kemudian dilakukan kajian kepustakaan. Setelah itu dilanjutkan dengan penelaahan isi sumber pustaka tersebut dalam kaitannya dengan masalah-masalah yang ditemui dalam kegiatan perkuliahan. Dari kajian yang dilakukan, muncul suatu gagasan yang penting serta menjadi landasan utama untuk melakukan penelitian tentang penerapan AG dalam mencari akar persamaan polinomial.
2. Mengambil teori dan konsep yang dibutuhkan dalam penyelesaian optimasi dengan menggunakan AG.
3. Merumuskan masalah. Membuat rancangan terlebih dahulu mengenai penerapan AG dalam mencari akar persamaan polinomial.
4. Melakukan studi literatur atau kajian kepustakaan yang berkaitan dengan masalah yang dirumuskan. Kajian pustaka dilakukan dengan cara mengumpulkan berbagai literatur yang berhubungan dengan permasalahan yang akan dibahas serta membaca dan memahaminya. Dalam hal ini, literatur yang digunakan berupa buku-buku dan jurnal-jurnal yang berkaitan dengan AG yang terkhusus pada masalah optimasi yaitu pada kasus pencarian akar-akar dari suatu fungsi.
5. Menyelesaikan masalah optimasi dengan menggunakan AG.

6. Menyusun laporan penelitian dari masalah tersebut.
7. Merumuskan kesimpulan, pengkajian, dan penyelesaian dari masalah tersebut.

4. HASIL DAN PEMBAHASAN

AG merupakan metode yang memandu proses pencarian yang diadaptasi dari proses evolusi yang terjadi di alam, serta mengikuti prinsip “siapa yang kuat, dia yang bertahan”. AG adalah algoritma yang dikembangkan dari proses pencarian solusi menggunakan pencarian secara *random*, hal ini terlihat pada proses pembangkitan populasi awal yang menyatakan sekumpulan solusi yang dipilih secara *random*.

Proses terbentuknya populasi awal secara *random* yang terdiri dari beberapa individu dalam satu populasi. Individu dalam hal ini merupakan solusi dari permasalahan, yang direpresentasikan dari sebuah kromosom berbentuk string biner yang berisi susunan gen-gen secara *random* sampai N bit. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol (kode), sehingga perlu dilakukan pendekodean. Untuk populasi awal dibangkitkan secara *random* sedangkan untuk populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan fungsi *fitness* yang diturunkan dari fungsi polinomial $p(x)$ yang akan dicari akar persamaannya. Nilai *fitness* suatu kromosom menunjukkan kualitas kromosom dalam populasi tersebut. Seleksi kemudian dilakukan untuk memilih kromosom-kromosom yang akan melakukan *crossover* dan mutasi. Proses seleksi menggunakan metode *Roulette Wheel* yang pada langkah-langkahnya yaitu dengan menghitung total *fitness* ($TotFitness$) pada populasi, menghitung *fitness* relatif (p_k) tiap kromosom, menghitung *fitness* kumulatif (q_k), dan membangkitkan bilangan *random* $[0,1]$ sebanyak 20 bilangan yang kemudian dibandingkan dengan *fitness* kumulatif untuk memilih induk yang akan menjadi kandidat *crossover*.

Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai *fitness* dari kromosom induk (*parent*) dan nilai *fitness* dari kromosom anak (*offspring*), serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik (Kusumadewi, Sri: 2003).

Metode Algoritma Genetika untuk Pencarian Akar Persamaan Polinomial

1. Bangkitkan populasi secara *random*

Diketahui polinomial $p(x) = a_0 + a_1x^1 + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5$ yang memiliki batas atas dan batas bawah (interval) pada $x_a \leq x \leq x_b$. Terdapat tiga kelompok data yang diperlukan pada bagian masukan (*input*) yaitu koefisien ($a_0, a_1, a_2, a_3, a_4, a_5$), batas bawah x_a dan batas atas x_b serta maksimum generasi.

Teknik pengkodean yang digunakan adalah pengkodean biner, dimana satu kromosom yang berukuran 8 bit ($N = 8$) akan merepresentasikan satu individu. Jika ukuran kromosom bit, maka pada interval $x_a \leq x \leq x_b$ akan terdapat 2^N kemungkinan kromosom yang merupakan representasi dari nilai x .

Proses membangkitkan populasi awal, dimana sejumlah kromosom akan dibangkitkan secara *random* dengan cara tertentu sebanyak $PopSize \times N$ yang terdiri dari bilangan integer $[0,1]$. Setelah didapatkan beberapa kromosom untuk populasi awal, maka kromosom-kromosom tersebut yang masih dalam bentuk kode biner kemudian didekodekan dengan menggunakan persamaan

$$v_j = x_a + (x_b - x_a) \frac{\sum_{i=1}^N b_i \cdot 2^{N-i}}{2^N - 1} \quad (1.1)$$

Keterangan:

$v_j =$ kromosom ke $-j$; $j = 1,2,3, \dots, PopSize$

$x_a =$ batas bawah

$x_b =$ batas atas

$N = \text{ukuran kromosom}$

$$b_i = \text{gen ke } - i \quad ; \quad i = 1,2,3, \dots, N$$

2. Evaluasi Fitness

Ada dua hal yang harus dilakukan dalam melakukan evaluasi kromosom yaitu evaluasi fungsi objektif (fungsi tujuan) dan konversi fungsi objektif kedalam fungsi *fitness*. Secara umum fungsi *fitness* diturunkan dari fungsi objektif dengan nilai yang tidak negatif. Apabila ternyata fungsi objektif memiliki nilai negatif, maka perlu ditambahkan suatu konstanta C agar nilai *fitness* yang terbentuk menjadi tidak negatif (Kusumadewi, Sri : 2003).

Dalam penelitian ini, akan dicari akar persamaan dimana fungsi objektif harus memenuhi $p(x) = 0$ sehingga fungsi *fitness*nya adalah

$$\text{Fitness} = F(x) = \frac{1}{1 + |p(x)|} \tag{2.1}$$

3. Seleksi

Untuk memperoleh solusi terbaik, harus dilakukan penyeleksian solusi yang mempunyai nilai *fitness* yang berkategori tinggi. Seleksi yang digunakan adalah metode *roulette wheel*. Pada metode ini, individu-individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen memiliki ukuran yang sama dengan ukuran *fitness*nya. Sebuah bilangan *random* dibangkitkan dan individu yang memiliki segmen dalam kawasan bilangan *random* tersebut akan terseleksi. Proses ini diulang hingga diperoleh sejumlah individu yang diharapkan.

Algoritma *roulette wheel* (Kusumadewi, Sri :2003):

1) Hitung total *fitness* (F):

- $TotFitness = \sum F_k \quad ; k = 1,2,3, \dots, PopSize$

2) Hitung *fitness* relatif tiap individu:

- $p_k = \frac{F_k}{TotFitness}$ (3.1)

3) Hitung *fitness* kumulatif:

- $q_1 = p_1$ (3.2)
- $q_k = q_{k-1} + p_k \quad ; k = 2,3,4, \dots, PopSize$ (3.3)

4) Pilih induk yang akan menjadi kandidat untuk *dicrossover* dengan cara

- Bangkitkan bilangan *random*
- Jika $q_k \leq r$ dan $q_{k+1} > r$ maka pilih kromosom ke- $(k + 1)$ sebagai kandidat induk.

4. Crossover

Crossover dilakukan terhadap dua kromosom untuk menghasilkan kromosom anak (*offspring*). Metode *crossover* yang akan dilakukan adalah $1 - \text{point crossover}$. Pada metode ini pilih bilangan integer $[1, N - 1]$ secara *random* untuk menentukan posisi persilangan. Kemudian tukarkan bagian kanan titik potong dari kedua induk kromosom tersebut menghasilkan kromosom anak (Kusumadewi, Sri: 2003).

Misalkan:

$$v'_1 = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ \hline v'_2 = & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ \hline \end{array}$$

Apabila posisi titik potong yang terpilih secara *random* adalah 3, maka kromosom anak (*offspring*) yang terbentuk adalah

$$v''_1 = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ \hline v''_2 = & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ \hline \end{array}$$

5. Mutasi

Mutasi merupakan operator genetika yang hanya melibatkan satu kromosom saja, yang prosesnya memungkinkan untuk memodifikasi informasi gen-gen pada suatu kromosom. Pada tahap ini dipilih bilangan *random* untuk menentukan posisi gen yang akan dimutasi. Kemudian mengganti *bit 1* dengan *bit 0* atau mengganti *bit 0* dengan *bit 1*.

Misalkan:

$$v'_1 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Terkena mutasi pada gen ke- 6, diperoleh:

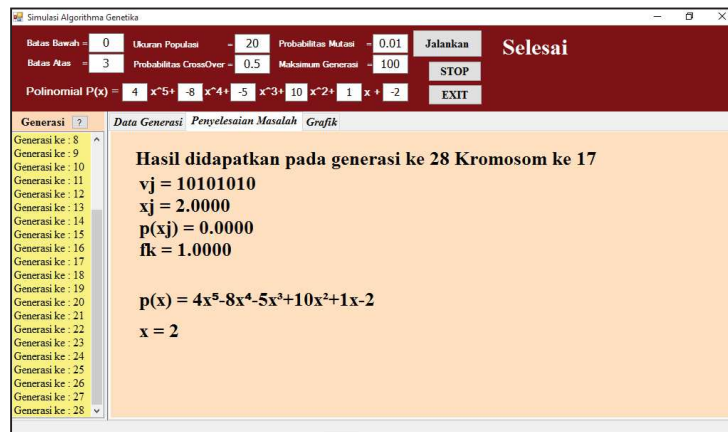
$$v_1^* = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Probabilitas mutasi menunjukkan persentasi jumlah total gen pada populasi yang akan mengalami mutasi. Untuk melakukan mutasi, terlebih dahulu harus menghitung jumlah total gen pada populasi tersebut. Kemudian bangkitkan bilangan *random* yang akan menentukan posisi mana yang akan dimutasi (gen beberapa pada kromosom beberapa).

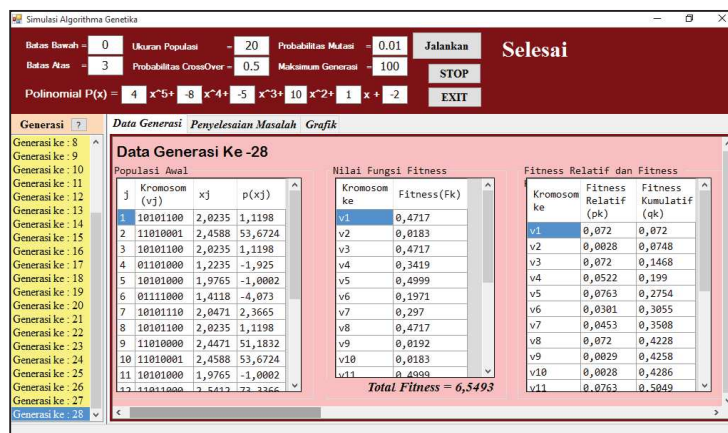
6. Kondisi Selesai

Setelah proses mutasi maka terbentuklah populasi baru dan telah diselesaikan satu iterasi dalam AG atau disebut dengan satu generasi. Proses selanjutnya adalah mengevaluasi kembali populasi baru yang terbentuk setelah melalui tahapan-tahapan. Apabila nilai *fitness* telah mencapai *fitness = 1* atau *fitness ≈ 1* dan telah mencapai maksimum generasi maka pencarian dihentikan, jika tidak maka kembali ke langkah 3.

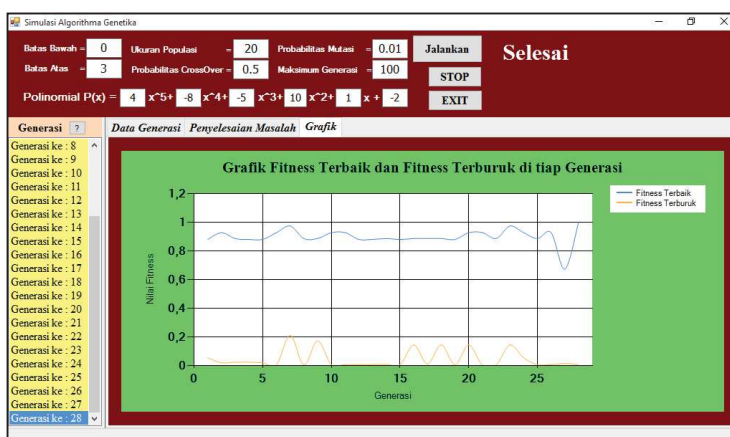
Contoh: Diketahui persamaan polinomial $p(x) = (-2) + x + 10x^2 - 5x^3 - 8x^4 + 4x^5$ dengan interval $[0,3]$, akan dicari akar persamaan $p(x)$ dengan menggunakan algoritma genetika. Berikut merupakan output yang dihasilkan untuk mencari akar dari suatu persamaan fungsi $p(x) = (-2) + x + 10x^2 - 5x^3 - 8x^4 + 4x^5$ menggunakan algoritma genetika:



Gambar 1 Tampilan penyelesaian masalah



Gambar 2 Tampilan Data Generasi



Gambar 3 Tampilan Grafik

Dari hasil implementasi menjalankan program algoritma genetika dengan menggunakan *software Microsoft Visual Basic 2010* untuk mencari nilai dari akar persamaan $p(x)$ didapat bahwa persamaan polinomial $p(x) = (-2) + x + 10x^2 - 5x^3 - 8x^4 + 4x^5$ memiliki akar persamaan $x = 2$ yang didapatkan pada generasi ke-28 Kromosom ke-17, dengan kromosom $v_j = 10101010$.

5. KESIMPULAN

Berdasarkan hasil penelitian di atas maka pencarian akar persamaan polinomial menggunakan AG dapat diselesaikan. Langkah-langkah dalam pencarian akar persamaan polinomial yaitu membangkitkan populasi awal secara *random*, mencari nilai dari fungsi *fitness* (evaluasi *fitness*), melakukan seleksi pada kromosom, melakukan *crossover* pada induk kromosom yang telah terpilih untuk melakukan *crossover*, kemudian melakukan mutasi pada kromosom yang terpilih untuk melakukan mutasi untuk menghasilkan kromosom yang lebih variatif, mengevaluasi kembali kromosom-kromosom baru yang terbentuk setelah melalui beberapa tahapan, apabila nilai *fitness* telah mencapai $fitness = 1$ atau $fitness \approx 1$ dan telah mencapai maksimum generasi maka pencarian dihentikan, jika tidak maka kembali ke langkah melakukan seleksi pada kromosom.

Proses pencarian akar persamaan polinomial menggunakan AG berupa iterasi yang biasa disebut generasi, sehingga membutuhkan proses komputasi yang memudahkan dalam pengerjaannya. Mencari akar persamaan polinomial dengan menggunakan AG serta bantuan program yang telah dibuat pada *software Microsoft Visual Basic 2010* lebih efisien.

DAFTAR PUSTAKA

- Apriani, R., 2012, *Algoritma Genetika untuk Menyelesaikan Permasalahan Penjadwalan Perkuliahan dan Praktikum*, Skripsi: Universitas Sumatera Utara.
- Desiani, A. dan Arhami, M., 2006, *Konsep Kecerdasan Buatan*, Penerbit Andi: Yogyakarta.
- Hartono, J., 2005, *Pengenalan Komputer*, Penerbit Andi: Yogyakarta.
- Kusumadewi, S., 2003, *Artificial Intelligenci (Teknik dan Aplikasinya)*, Penerbit Graha Ilmu: Yogyakarta.
- Lee, C., 2014, *Buku Pintar Pemrograman Visual Basic 2010*, Penerbit PT Elex Media Komputindo: Jakarta.
- Munir, R., 2006, *Metode Numerik*, Bandung: Informatika.
- Nurpadillah, 2014, *Algoritma Genetika dalam Penyelesaian Travelling Salesman Problem (TSP)*, Skripsi: Universitas Negeri Makassar.
- Purnomo, H.D., 2014, *Cara Mudah Belajar Metode Optimasi Metaheuristik Menggunakan Matlab*, Graha Media: Yogyakarta.
- Suryaningsih, 2012, *Komparasi Metode Muller dan Metode Bairstow dalam Menentukan Akar-Akar Persamaan Polinomial*, Skripsi: Universitas Negeri Makassar.
- Suyanto, 2005, *Algoritma Genetika dalam MATLAB*, Penerbit Andi: Yogyakarta.
- Tiro, M.A., 2008, *Dasar-Dasar Statistika*, Penerbit Andira Publisher: Makassar.
- Verberg, D. dan Purcell, E. dan Rigdon, S., 2007, *Terjemahan Calculus Ninth Edition Verberg, Purcell, Rigdon*, Penerbit Erlangga: Jakarta.
- Yusuf, A. dan Soesanto, O., 2012, *Algoritma Genetika pada Penyelesaian Akar Persamaan Sebuah Fungsi*, Jurnal: Universitas Lambung Mangkurat, Banjarbaru.
- Zukhri, Z., 2014, *Algoritma Genetika Metode Komputasi Evolusioner Untuk Menyelesaikan Masalah Optimasi*, Penerbit Andi: Yogyakarta.