

## OPTIMASI FUNGSI NONLINEAR TIGA VARIABEL MENGUNAKAN ALGORITMA GENETIKA

### *OPTIMIZATION OF THREE VARIABLE NONLINEAR FUNCTIONS USING GENETIC ALGORITHM*

**Didin Adri<sup>1§</sup>**

<sup>1</sup>Universitas Muhammadiyah Buton, Kota Baubau, Indonesia [Email: [adri.didin@gmail.com](mailto:adri.didin@gmail.com)]

<sup>§</sup> *Corresponding Author*<sup>1</sup>

Received Januari 2021; Accepted Juni 2021; Published Juni 2021;

---

#### **Abstrak**

Pada penelitian ini akan ditentukan nilai minimum dari tiga buah fungsi yang berbeda menggunakan algoritma genetika. Pengkodean algoritma genetika pada penelitian ini menggunakan binary encoding. Metode crossover yang dipilih adalah uniform crossover dengan menggunakan parameter Jumlah Maksimum generasi ( $mIt$ ) = 100; Jumlah Populasi ( $nPop$ ) = 100; Probabilitas Crossover ( $pCro$ ) = 0,01; Probabilitas Mutasi ( $pMut$ ) = 0,02. Kemudian Program dijalankan sebanyak 20 kali untuk melihat konsistensi nilai yang dihasilkan dari kode program yang dibuat, kemudian diperoleh hasil sebagai berikut : Nilai minimum untuk fungsi pertama pada persamaan 1 adalah 1,0000, nilai minimum dari fungsi kedua persamaan (1) adalah 0 dan nilai minimum dari fungsi ketiga pada persamaan (1) adalah 49,0000.

**Kata Kunci:** Optimisasi, Fungsi nonlinear multi variabel, Algoritma Genetika.

#### **Abstract**

*In this research, the minimum value of three different functions will be determined using a genetic algorithm. The coding of the genetic algorithm in this research used binary encoding. The crossover method chosen is uniform crossover by using the parameter Maximum number of generations ( $mIt$ ) = 100; Total population ( $nPop$ ) = 100; Probability of Crossover ( $pCro$ ) = 0.01; Probability of Mutation ( $pMut$ ) = 0.02. Then the program is run 20 times to see the consistency of the values generated from the program code created, then the following results are obtained: The minimum value for the first function in equation 1 is 1.0000, the minimum value of the second function equation (1) is 0 and the value The minimum of the third function in equation (1) is 49,0000.*

**Keywords:** Optimization, Multi-variable nonlinear function, Genetic Algorithm

---

## 1. Pendahuluan

Solusi eksak untuk masalah optimisasi fungsi tiga variabel sangat susah untuk dicari, Solusi optimal dapat dicari dengan menggunakan pendekatan numerik. Pendekatan numerik untuk menyelesaikan masalah optimisasi para ahli sudah mengusulkan banyak sekali metode, salah satunya menggunakan Algoritma Genetika. Algoritma genetika memungkinkan untuk menemukan solusi untuk permasalahan optimisasi suatu fungsi. Algoritma ini didasarkan pada proses genetika yang ada di dalam makhluk hidup yaitu perkembangan generasi dalam sebuah populasi yang alami, secara lambat laun mengikuti proses seleksi alam atau siapa yang kuat dia yang akan bertahan (survive). Dengan meniru proses ini, Algoritma Genetika dapat digunakan untuk mencari solusi permasalahan-permasalahan optimisasi fungsi-fungsi multi variabel. (Beasley & Chu, 1994).

## 2. Landasan Teori

Pada penelitian ini akan menggunakan Algoritma Genetika untuk menyelesaikan masalah optimisasi pada fungsi-fungsi berikut:

$$\begin{aligned}
 f(x_1, x_2, x_3) &= (x_1 - 2)^2 + 2(x_2 - 1)^2 + 3(3 - x_3)^2 - \\
 &\quad 0,3 \cos(3\pi x_1) - 0,4 \cos(4\pi x_2) + 1,7 \\
 f(x_1, x_2, x_3) &= (x_1 + 2x_2 - 8)^2 + (x_1 + 3x_3 - 5)^2 + \\
 &\quad (2x_1 + x_2 - 7)^2 \\
 f(x_1, x_2, x_3) &= e^{-x_1^2 - x_2^2 - x_3^3} + (2 - x_1)^2 + (1 - 2x_2)^2 + \\
 &\quad (x_3 - 10)^2
 \end{aligned} \tag{1}$$

Dengan  $x_1, x_2, x_3 \in [-3, 3]$ . Kemudian akan ditentukan nilai  $x_1, x_2$  dan  $x_3$  agar fungsi mencapai nilai minimum.

### Pengkodean (Encoding)

Pada penelitian ini digunakan metode pengkodean biner (Binary Encoding) untuk merepresentasikan kromosom dan panjang gen (mj) yang dihitung dengan menggunakan rumus:

$$2^{m_i-1} < (b_i - a_i)10^k < 2^{m_i} - 1 \tag{2}$$

Kemudian untuk mengkonversi dari nilai biner ke desimal untuk variabel,  $i = 1, 2, 3$  digunakan rumus (Fadlisyah, dkk. 2009):

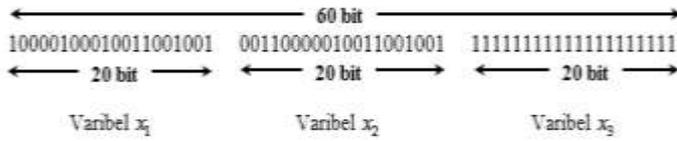
$$x_i = a_i + c_i \cdot \frac{b_i - a_i}{2^{m_i} - 1} \tag{3}$$

dalam hal ini

- $m_i$  = panjang kode bit (Panjang Kromosom)
- $b_i$  = nilai batas atas dari variabel  $x_i$
- $a_i$  = Nilai batas bawah dari variabel  $x_i$
- $k$  = jumlah angka dibelakang koma (Akurasi)
- $c_i$  = nilai desimal dari  $m_i$  bit kromosom

Pada kasus di atas, karena  $x_1, x_2, x_3 \in [-3, 3]$  maka diperoleh  $a_1 = a_2 = a_3 = -3$  dan  $b_1 = b_2 = b_3 = 3$ , kemudian pada perhitungan ini digunakan tingkat akurasi 5 (lima) angka di belakang koma atau dengan kata lain dipilih  $k = 5$ , sehingga dengan menggunakan persamaan (2) diperoleh  $m_i = 20$ , untuk setiap  $i = 1, 2, 3$ . Sehingga diperoleh panjang total dari kromosom adalah  $m = m_1 + m_2 + m_3 = 20 + 20 + 20 = 60$  gen (bit). 60 gen (bit) tersebut dapat direpresentasikan

sebagai berikut.



Gambar 1. Representasi Kromosom

Nilai desimal untuk masing-masing Variabel dan pada Gambar 1 akan ditunjukkan pada tabel berikut:

Variabel	Nilai Biner	Nilai Desimal
$x_1$	10000100010011001001	541897
$x_2$	00110000010011001001	197833
$x_3$	11111111111111111111	1048575

Kemudian, dengan menggunakan persamaan (3) akan dihitung nilai variabel  $x_1, x_2$  dan  $x_3$  dalam bentuk nilai desimal.

$$x_1 = -3 + 541897 \cdot \frac{3 - (-3)}{2^{20} - 1} = 0,10076$$

$$x_2 = -3 + 197833 \cdot \frac{3 - (-3)}{2^{20} - 1} = -1,86799$$

$$x_3 = -3 + 1048575 \cdot \frac{3 - (-3)}{2^{20} - 1} = 3,00000$$

**Parameter**

Sebelum memulai proses algoritma genetika, terlebih dahulu harus diinisialisasi nilai parameter-parameter awal yang diperlukan, yaitu sebagai berikut:

Jumlah Maksimum generasi (mIt) = 100;

Jumlah Populasi (nPop) = 100;

Probabilitas Mutasi (pMut) = 0,02.

**Alur Algoritma Genetika**

1. Inisialisasi Populasi
2. Seleksi *Parent* and *Crossover*
3. Mutasi *offsprings*
4. Mengabungkan populasi dan *offspring*
5. Evaluasi, pengurutan, dan pemilihan berdasarkan nilai fitness
6. Ulangi langkah 2 sampai kriteria berhenti terpenuhi (maIt = 200)

**Inisialisasi Populasi**

Pada tahap ini akan dibentuk populasi yang terdiri dari nPop buah Kromosom secara acak.

**Regenerasi (Evolusi)**

1. Seleksi (Selection)

Pada tahap ini akan dipilih dari dua buah kromosom sebagai *parent*. Metode yang digunakan adalah *roulette-wheel selection* (Roda Roulette). Seperti permainan roda roulette, Kromosom ditempatkan pada masing-masing potongan lingkaran secara proposional sesuai dengan nilai fitnessnya. Kromosom yang memiliki nilai fitness yang lebih besar memiliki peluang yang lebih besar untuk terpilih sebagai *parent*. Prinsip kerja dari *roulette-wheel selection* sebagai berikut (Adriana, dkk, 2015):

- Hitung nilai Fitness untuk setiap kromosom:

$$eval(v_k) = f(x_1, x_2, x_3), k = 1, 2, \dots, nPop$$

Jika mencari nilai minimum maka:

$$eval(v_k) = \frac{1}{f(x_1, x_2, x_2)}, k = 1, 2, \dots, nPop$$

- Hitung nilai total fitness untuk populasi:

$$F = \sum_{k=1}^{nPop} eval(v_k)$$

- Hitung Probabilitas seleksi  $p_k$  untuk setiap kromosom

$$p_k = \frac{eval(v_k)}{F}, k = 1, 2, \dots, nPop$$

- Hitung probabilitas kumulatif  $q_k$  untuk setiap kromosom

$$q_k = \sum_{j=1}^k p_j, k = 1, 2, \dots, nPop$$

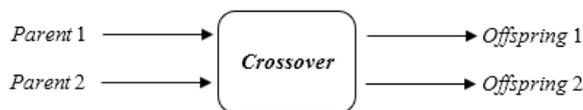
- Selanjutnya, akan diputar roda roulettes sebanyak jumlah individu (nPop). Proses memutar roda roulette indentik dengan membangkitkan sebuah bilangan acak  $r_k$  yang nilainya berada pada interval [0, 1]. Proses seleksi menggunakan persamaan berikut:

$$q_{k-1} \leq r_k \leq q_k \tag{4}$$

Jika persamaan (4) terpenuhi, maka akan dipilih kromosom ke- $k$  ( $v_k$ ) sebagai *parent*.

## 2. Crossover

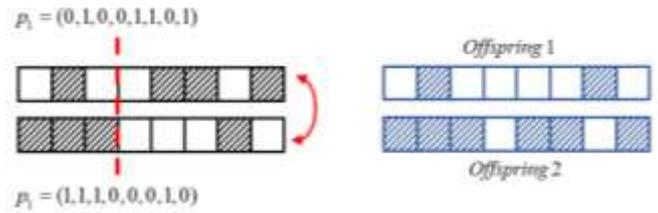
*Crossover* (kawin Silang) merupakan metode mengawinkan dua kromosom yang terpilih sebagai *parent* dengan tujuan untuk mendapatkan *offspring* yang lebih baik.



ada banyak metode yang digunakan untuk *crossover* diantaranya yaitu:

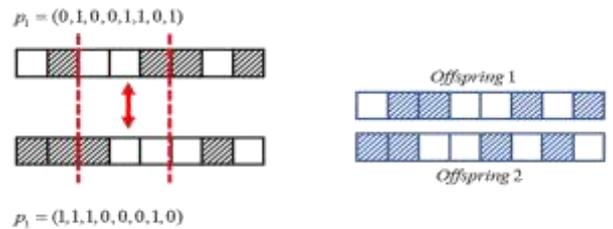
- *Single - Point Crossover*

Pada metode ini akan dipilih satu titik potong secara acak kemudian bagian pertama dari *parent1* digabungkan dengan bagian kedua dari *parent2*.



- *Two - Point Crossover*

Pada metode ini akan dipilih dua titik potong secara acak kemudian bagian tengah dari *parent 1* ditukarkan dengan bagian tengah dari *parent 2*.



- *Uniform Crossover*

Pada metode ini, pertama dipilih bilangan  $r_k$  bernilai 0 atau 1 secara acak sebanyak jumlah gen pada kromosom. Kemudian jika nilai  $r_k = 0$  maka nilai gen ke- $k$  pada *parent 1* akan ditukar nilai gen ke- $k$  pada *parent 2*. Secara umum nilai baru dari gen akan mengikuti rumus berikut:

$$h_{1k} = r_k g_{1k} + (1 - r_k) g_{2k}$$

$$h_{2k} = r_k g_{2k} + (1 - r_k) g_{1k}$$

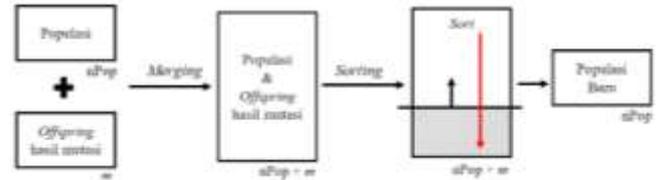
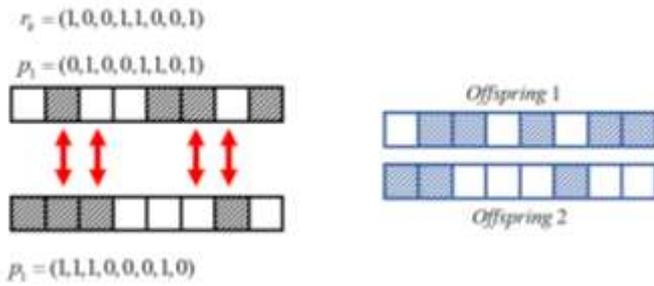
dalam hal ini

$h_{1k}$  adalah nilai gen ke- $k$  yang baru pada *parent 1*

$h_{2k}$  adalah nilai gen ke- $k$  yang baru pada *parent 2*

$g_{1k}$  adalah nilai gen ke- $k$  pada *parent 1*

$g_{2k}$  adalah nilai gen ke- $k$  pada *parent 2*



**Gambar 2.** Evaluasi, pengurutan, dan pemilihan berdasarkan nilai fitness

3. Mutasi

Mutasi adalah proses untuk mengubah nilai gen pada kromosom untuk membuat perbedaan dalam populasi. Mutasi dilakukan dengan tujuan agar terjadi *prematurn convergen*, yaitu suatu kondisi dimana kromosom-kromosom mengalami konvergen disaat nilai fitnessnya belum mencapai optimal. Semua gen dalam suatu kromosom memiliki peluang yang sama untuk mengalami mutasi. Suatu gen mengalami mutasi jika bilangan acak yang dibangkitkan kurang dari nilai probabilitas mutasi (pMut) maka nilai gen akan diubah nilai kebalikannya (0 diubah menjadi 1 dan 1 ubah menjadi 0).

3. Hasil Dan Pembahasan

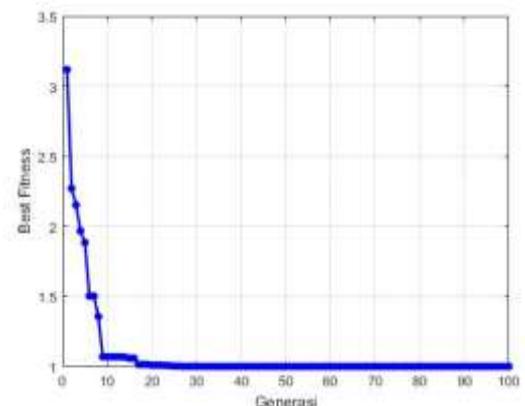
Proses Algoritma genetika untuk mencari nilai optimal fungsi tiga variabel dijalankan menggunakan parameter yang telah disebutkan di atas kemudian dilakukan run program sebanyak 20 kali. Untuk menjalan proses algoritma dalam penelitian ini menggunakan bahasa pemrograman MATLAB 2016. Kode Program telah disediakan pada lampiran penelitian ini. Setelah program di run sebanyak 20 kali, kemudian dihitung rata-rata nilai hasil run program. Hasil Running Program ditunjukkan pada tabel 1 berikut.

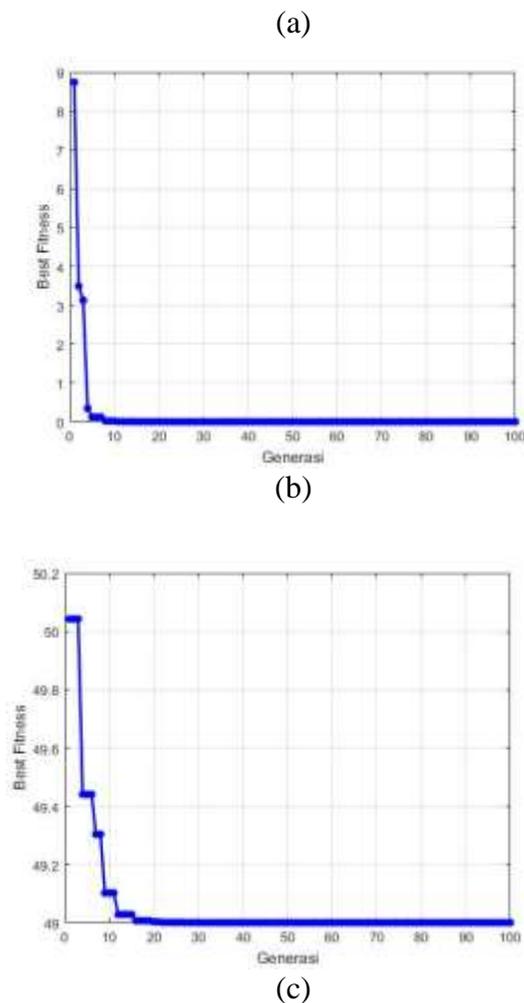
4. Evaluasi, pengurutan, dan pemilihan berdasarkan nilai fitness

Tahap ini dilakukan untuk menjaga agar kromosom yang memiliki nilai fitness terbaik tidak hilang selama proses evolusi. Sebab Seleksi dilakukan secara acak, sehingga tidak ada jaminan bahwa kromosom yang memiliki nilai fitness yang terbaik yang akan dipilih, bisa juga nilai fitness dari kromosom tersebut nilai fitnessnya akan menurun karena proses crossover atau mutasi. Secara umum prosedur tahap ini ditunjukkan pada gambar berikut.

**Tabel 1.** Hasil Run Program AG fungsi pada Persamaan (1)

Fungsi	Mean of Solution in 20 times Run			Mean Function values
	x1	x2	x3	
1	2,000000	1,00002	3,00000	1,00000
2	2,000000	3,00000	1,00000	0
3	2,000000	0,50000	3,00000	49,00000





**Gambar 3.** Hasil Run Program satu kali (a) fungsi 1 (b) fungsi 2 (c) fungsi 3.

Pada Gambar 3 (a) Nilai Fitness konvergen ke titik minimal pada generasi ke 20. Kemudian pada Gambar 3 (b) Nilai Fitness konvergen ke titik minimal pada generasi ke 10 dan Pada Gambar 3 (c) Nilai Fitness konvergen ke titik minimal pada generasi ke 20.

#### Matlab Code

```
Main.m
clear;
clc;
close all;
%% Problem Definition
problem.CostFunction = @(x)
fungsiCost4(x);
problem.nVar = 3;
problem.VarMin = [-3 -3 -3];
problem.VarMax = [ 3  3  3];
```

```
%% GA Paramaters
params.MaxIt = 100;
params.nPop = 100;
params.beta = 1;
params.pC = 1;
params.mu = 0.02;
params.sigma = 0.1;
params.gamma = 0.1;

%% Run GA
% trial = 20;
% for i = 1:trial
    out = RunGA(problem,params);
%     f(i) = out.bestsol;
% end

%% Result

figure;
plot(out.bestcost, 'b.-',
'LineWidth', 2);

xlabel('Generasi');
ylabel('Best Fitness');
grid on;
```

#### 4. Kesimpulan Dan Saran

Dari hasil pembahasan, dapat disimpulkan bahwa algoritma genetika yang telah dijalankan dapat menentukan nilai optimal dari fungsi multivaribel. Nilai minimum untuk fungsi pertama pada persamaan 1 adalah 1,0000, nilai minimum dari fungsi kedua persamaan (1) adalah 0 dan nilai minimum dari fungsi ketiga pada persamaan (1) adalah 49,0000.

Dalam Penelian ini, pemrograman menggunakan bahasa pemrograman matlab. Sehingga banyak fungsi-fungsi bantu yang dapat mempersingkat penulisan program. Penelitian selanjutnya akan diteliti masalah-masalah lain yang dapat diselesaikan oleh Algoritma Genetika.

## Daftar Pustaka

- [1] Adri F dan Emerensye, 2020, *Elitisme Algoritma Genetika Pada Fungsi Nonlinear Dua Peubah*, Jurnal Komputer dan Informatika, Universitas Nusa Cendana.
- [2] Adri F dan Fadly, 2015, *Algoritma Generika dan Penerapannya*, Teknosain, Yogyakarta.
- [3] Intan B dan Miftahol, 2010, *Teknik-Teknik Optimisasi Heuristik*, Graha Ilmu, Yogyakarta.
- [4] Mostapha KH. 2020. *Practical Genetic Algorithms*, online media (Youtube) [youtube.com/list=PLgH3sgdvgO4TQagjfi1FmPcdBjIlmk5s9](https://youtube.com/list=PLgH3sgdvgO4TQagjfi1FmPcdBjIlmk5s9), diakses tanggal 24 Desember 2020.
- [5] Suyanto, 2007, *Algoritma Genetika Dalam Matlab*, Penerbit Andi, Jakarta.
- [6] Xinjie Y dan Mitsuo Gen, 2010, *Introduction to Evolutionary Algorithms*, Springer-Verlag, London.