

ALGORITMA GENETIKA BERBANTUAN MATLAB UNTUK PENETAPAN PARAMETER TIDAK KONSTAN PADA TURBIN AIR SKALA MIKRO

GENETIC ALGORITHM ASSISTED BY MATLAB FOR DETERMINING THE NOT CONSTANT PARAMETERS ON MICRO-SCALE WATER TURBINE

Fauzan Shulhan¹, Arief Goeritno^{*1}, Ridwan Arief Subekti²

¹ Electrical Engineering Study Program, Fakultas Teknik dan Sains, Universitas Ibn Khaldun Bogor
Jalan Sholeh Iskandar km.2 Kedungbadak, Tanah Sareal, Kota Bogor 16164

² P2 Telimek, Badan Riset dan Inovasi Nasional (BRIN)
Jalan Sangkuriang, Dago, Coblong, Kota Bandung 40135

Diterima: 12 April 2022

Direvisi: 17 Mei 2022

Disetujui: 13 Juni 2022

ABSTRAK

Metode Algoritma Genetika (AG) telah dikenal sebagai algoritma dengan teknik pencarian secara acak dan mampu dengan solusi akurat dan optimal, digunakan dalam aktivitas prediksi di berbagai bidang. Sejumlah parameter berpengaruh pada turbin air, meliputi gaya gravitasi bumi dan densitas air yang dapat dianggap konstan, sedangkan parameter tidak konstan meliputi tinggi jatuh air, debit air, dan efisiensi mekanis. Pencapaian efisiensi mekanis pada nilai optimum dilakukan dengan pendugaan berbantuan metode AG dengan aplikasi MATLAB terhadap tiga parameter berpengaruh bersifat tidak konstan. Dua hal pokok sebagai sasaran penelitian, meliputi perolehan *logical array* untuk dua struktur matriks berbeda, yaitu 5x3 dan 3x2. Metode pelaksanaan optimasi berupa (a) penyiapan aplikasi MATLAB dan (b) pelaksanaan simulasi dengan tahapan proses inialisasi, evaluasi, *crossover*, mutasi dan seleksi. Hasil kedua proses simulasi yang diperoleh sesuai dengan pendugaan, yaitu kesempurnaan pada nilai *logical array*. Berdasarkan hasil tersebut dapat disimpulkan, bahwa pendugaan dengan metode AG terhadap sejumlah parameter berpengaruh tidak konstan pada turbin air skala mikro sangat cocok untuk pencapaian efisiensi mekanis pada nilai optimum.

Kata kunci: algoritma genetika, parameter berpengaruh, turbin air skala mikro, efisiensi mekanis

ABSTRACT

The Genetic Algorithm (GA) method is known as an algorithm with a random search technique and is capable of accurate and optimal solutions so that it can be used in predictive activities in various fields. Parameters that affect the water turbine, including the earth's gravity and water density, can be considered constant, while the influential parameters that are not constant include the height of the waterfall, water flow, and mechanical efficiency. The achievement of mechanical efficiency at the optimum value was estimated using the AG method assisted by the MATLAB application on three parameters that were not constant. The two main things as research objectives, including the acquisition of logical arrays for two different matrix structures, i.e., 5x3 and 3x2. The optimization implementation methods are (a) the preparation of the MATLAB application and (b) the implementation of a simulation with the stages of the process of initialization, evaluation, crossover, mutation, and selection. Both simulation processes obtained results that match the predictions, i.e., perfection in the logical array values. Based on these results, it can be concluded that the estimation using the GA method on several of the not constantly influential parameters on micro-scale water turbines is very suitable to achieve mechanical efficiency at the optimum value.

Keywords: genetic algorithm, influential parameters, micro-scale water turbine, mechanical efficiency

PENDAHULUAN

*Corresponding Author :

Email: arief.goeritno@uika-bogor.ac.id

DOI: DOI: <http://dx.doi.org/10.37209/jtbbt.v12i1.259>

Implementasi algoritma genetika merupakan pencarian solusi yang cepat untuk proses optimasi. Algoritma Genetika (AG) berasal dari algoritma komputasi yang terinspirasi oleh teori evolusi Darwin. Salah satu aplikasi AG pada permasalahan optimasi kombinasi, diperoleh nilai solusi optimal terhadap permasalahan. Sebuah solusi terbangkitkan dalam AG disebut sebagai kromosom [1]. Kesederhanaan dan pencarian solusi yang cepat merupakan salah satu keunggulan dari AG. Keunggulan tersebut merupakan alasan AG dapat digunakan dalam berbagai bidang [2], sehingga metode AG digunakan untuk penemuan variabel rancangan kondisi optimal guna pencapaian kriteria bebas kejutan pada turbin air [3]. Variabel rancangan kondisi optimal tersebut digunakan untuk pembuatan model turbin air yang didasarkan kepada data diameter, lebar, jumlah sudu, sudut kelengkungan sudu, debit air, dan beban generator [4]. Penjelasan tersebut merupakan alasan terhadap pemilihan judul pada artikel ini, agar efisiensi mekanis turbin air skala mikro dapat diperoleh seoptimal mungkin melalui penetapan sejumlah parameter berpengaruh pada turbin air yang bersifat tidak konstan dan dipilih sebagai parameter acuan.

Sejumlah *state-of-the-art* penelitian sebelumnya yang berkaitan dengan penelitian ini meliputi (i) pemilihan simulasi berbasis metode AG berbantuan aplikasi MATLAB [5], (ii) tinjauan optimasi kinerja sebuah turbin air skala mikro yang dapat diuji dengan proses simulasi, agar diperoleh daya listrik pada kondisi maksimal [6], dan (iii) tinjauan efisiensi mekanis pada turbin air. Penetapan tahapan optimasi dengan bantuan AG dilakukan melalui beberapa tahapan. Nilai-nilai yang terbentuk dari hasil perhitungan berbasis metode AG nantinya menjadi sebuah program [7] yang dibaca dan diolah dengan aplikasi MATLAB [8]. Program tersebut digunakan untuk penentuan jumlah faktor yang menjadi parameter pencarian *fitness value* tidak hanya 1 atau 2 variabel saja, tetapi dapat dihasilkan yang bersifat *multivariable* [9]. Tinjauan pencapaian kondisi kinerja maksimal berdasarkan ketiga parameter berpengaruh tersebut pada turbin air skala piko tipe vorteks, berupa optimasi setiap sudu turbin untuk pemenuhan kriteria bebas kejutan.

Penelitian lain terkait penggunaan metode AG pada Pembangkit Listrik Tenaga Air (PLTA) tipe mini, yaitu untuk optimasi penempatan titik interkoneksi *Distributed Generation* (DG).

Optimasi tersebut bertujuan untuk perolehan nilai rugi-rugi daya seminimal mungkin. Hasil penggunaan metode AG mampu untuk penurunan nilai rugi-rugi daya secara signifikan melalui penentuan penempatan DG secara tepat [10]. Penggunaan metode AG pada PLTA tipe mikro berupa penyelesaian terhadap permasalahan kompleks pada waduk Logung sebagai waduk multifungsi. Permasalahan kompleks tersebut berupa optimasi pola operasi waduk Logung, sehingga dapat dihasilkan kapasitas sebesar 500 kW [11].

Penetapan sejumlah parameter berpengaruh yang bersifat tidak konstan untuk optimasi efisiensi mekanis pada turbin air, merupakan salah satu bentuk aktivitas untuk pelaksanaan proses prediksi atau pendugaan. Optimasi kinerja turbin air secara keseluruhan dipengaruhi oleh sejumlah parameter (variabel) penting (berpengaruh), yaitu (i) densitas air (ρ), (ii) gaya gravitasi bumi (g), (iii) tinggi jatuh air (h), (iv) debit air (Q), dan (v) efisiensi mekanis (η). Parameter densitas air (ρ) dan gaya gravitasi bumi (g), dapat diasumsikan bernilai konstan, sedangkan pada pembangkit listrik tenaga air (PLTA) untuk aliran sungai (skala mikro) hanya dipengaruhi oleh debit air (Q) dan efisiensi mekanis (η) [12]. Parameter debit air (Q) dan efisiensi mekanis (η) pada turbin air skala mikro merupakan penentu nilai kinerja sebuah turbin air [13], sehingga kinerja turbin air dapat dilakukan optimalisasi keberadaannya.

Berpedoman pada sejumlah *state-of-the-art* dari penelitian terdahulu, ditunjukkan bahwa turbin air skala mikro sangat dipengaruhi oleh dua parameter berpengaruh bersifat tidak konstan, yaitu debit air dan efisiensi mekanis. Simulasi ini dilakukan dalam dua tahapan pada penelitian ini. Pertama, lima parameter berpengaruh ditetapkan sebagai masukan, sedangkan tiga parameter berpengaruh bernilai tidak konstan ditetapkan sebagai keluaran. Tahap kedua dilakukan simulasi dengan penetapan tiga parameter berpengaruh bersifat tidak konstan sebagai masukan, sedangkan keluaran berupa dua parameter berpengaruh tidak konstan terhadap turbin air skala mikro, yaitu debit air dan efisiensi mekanis. Berdasarkan hal itu, pada penelitian ini dilakukan dua tahap simulasi dengan dua struktur matriks berbeda, yaitu 5x3 dan 3x2.

Studi prediksi ini ditandai dengan dua sasaran penelitian, yaitu melaksanakan simulasi berbasis metode AG berbantuan aplikasi MATLAB [5] untuk dua kategori simulasi, yaitu untuk matriks 5x3 dan 3x2. Penyiapan aplikasi

MATLAB [8] merupakan tahapan awal untuk proses simulasi berbasis AG, sehingga proses pendugaan terhadap pemilihan faktor berpengaruh pada kinerja turbin air skala mikro dapat dilaksanakan. Berpedoman kepada sasaran penelitian, maka perlu ditetapkan ruang lingkup penelitian yang berkaitan dengan sejumlah parameter berpengaruh yang bersifat konstan maupun tidak konstan pada turbin air berbasis algoritma genetika berbantuan MATLAB [5]. Perolehan kepastian terhadap penetapan sejumlah parameter yang bersifat tidak konstan tersebut untuk pencapaian optimasi kinerja turbin air skala mikro.

BAHAN DAN METODE

Bahan

Proses pengerjaan dalam penetapan sejumlah parameter berpengaruh untuk optimasi kinerja turbin skala mikro berbantuan AG dapat dilakukan dengan bantuan aplikasi MATLAB [14]. Aplikasi MATLAB adalah *platform* pemrograman berbasis matriks, biasa digunakan untuk analisis data, pembuatan algoritma, dan penciptaan permodelan dan pembuatan aplikasi MATLAB [8]. Pengoperasian metode AG berbantuan aplikasi MATLAB menjadikan fungsi AG prospektif dalam optimasi kinerja turbin air. Optimasi kinerja turbin air secara keseluruhan dipengaruhi oleh parameter bersifat konstan, yaitu (i) densitas air (ρ), (ii) gaya gravitasi bumi (g), sedangkan parameter bersifat tidak konstan meliputi (i) tinggi jatuh air (h), (ii) debit air (Q), dan (iii) efisiensi mekanis turbin (η). Penentuan nilai kinerja sebuah turbin air skala mikro [13], difokuskan kepada hasil simulasi dengan parameter debit air dan efisiensi mekanis turbin.

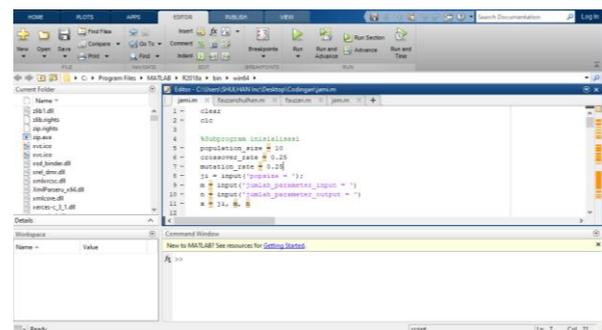
Metode

Penyiapan Aplikasi MATLAB

Penggunaan aplikasi MATLAB untuk simulasi, disiapkan dengan tahapan, yaitu i) penyusunan struktur *syntax* di *window*, (ii) pengoperasian atau pelaksanaan *running* terhadap struktur *syntax*, (iii) penampilan data hasil dari proses *running* terhadap struktur *syntax*, dan (iv) proses evaluasi terhadap struktur *syntax* sebagai bentuk pemeriksaan akhir. Pembentukan struktur *syntax* dilakukan untuk kemudahan proses simulasi penetapan urutan parameter berpengaruh terhadap optimasi kinerja turbin air.

Penyusunan Struktur *Syntax* di Window

Penyusunan struktur *syntax* dilakukan untuk pemberian perintah pengolahan nilai-nilai optimasi. Perintah pada MATLAB berdasarkan urutan proses yang disyaratkan pada metode AG, yaitu inisialisasi, evaluasi, *crossover*, mutasi dan seleksi. Tujuan pemberian perintah, yaitu untuk pembentukan matriks. Penyimpanan dan penamaan *file* struktur *syntax* dilakukan sesuai dengan proses dari setiap tahapan AG. Pemberian perintah dilakukan pada bagian “editor” dari aplikasi MATLAB [5]. Bagian “editor” berisikan struktur *syntax* masukan, seperti ditunjukkan pada Gambar 1.

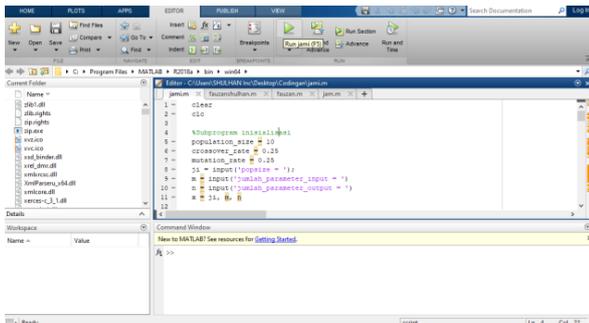


Gambar 1. Bagian “Editor” Berisikan Struktur *Syntax* Masukan

Berdasarkan Gambar 1 dapat dijelaskan, sejumlah baris struktur *syntax* yang merupakan bagian proses inisialisasi. Bermula dari struktur *syntax* tersebut, nantinya akan diperlihatkan sebuah matriks hasil dari pembangkitan bilangan secara acak.

Pengoperasian atau Pelaksanaan *Running* terhadap Struktur *Syntax*

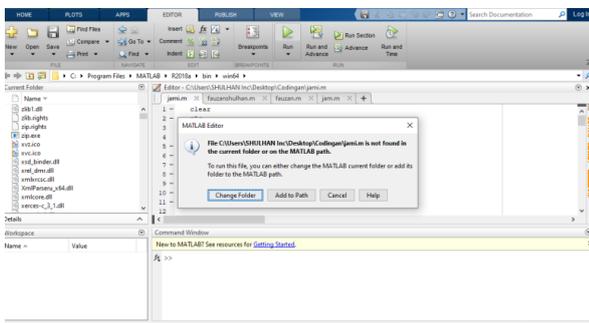
Setelah keterbentukan struktur *syntax* pada aplikasi MATLAB [5], selanjutnya dapat dilakukan *running* (dioperasikan). Pengoperasian struktur *syntax* bertujuan untuk pemeriksaan terhadap program dapat beroperasi atau tidak. Struktur *syntax* program yang beroperasi menampilkan data pada bagian “*Command*”, sedangkan program yang tidak beroperasi berwarna merah dan bertuliskan “*error*”. Untuk pengoperasian program pada aplikasi MATLAB dapat menggunakan bagian “*Run*” pada kolom “*Home*”. Tampilan proses pengoperasian, seperti yang ditunjukkan pada Gambar 2.



Gambar 2. Tampilan Proses Pengoperasian

Berdasarkan Gambar 2 dapat dijelaskan bahwa untuk penyusunan struktur *syntax* dapat digunakan bagian “Run” atau dengan tombol F5 pada papan ketik (*keyboard*) di *laptop*.

Saat penyusunan struktur *syntax*, terdapat perintah untuk penambahan *directory*. Tampilan perintah penambahan *directory*, seperti ditunjukkan pada Gambar 3.

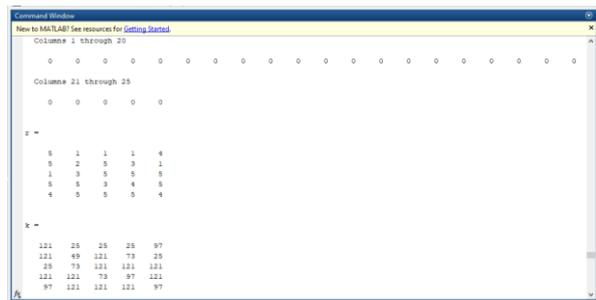


Gambar 3. Tampilan Perintah Penambahan Directory

Berdasarkan Gambar 3 dapat dijelaskan, bahwa penambahan direktori dilakukan guna pembacaan perintah pada struktur *syntax* terprogram pada aplikasi MATLAB. Data hasil penyusunan struktur *syntax* tertampilkan setelah struktur *syntax* berhasil di-*running*. Data tertampilkan pada bagian “Command Windows” di aplikasi MATLAB.

Penampilan Data Hasil dari Proses *Running* terhadap Trukstur *Syntax*

Hasil pengoperasian terhadap struktur *syntax* pada aplikasi MATLAB tertampilkan pada bagian “Command Windows”. Tampilan data hasil pengoperasian terhadap struktur *syntax*, ditunjukkan pada Gambar 4.

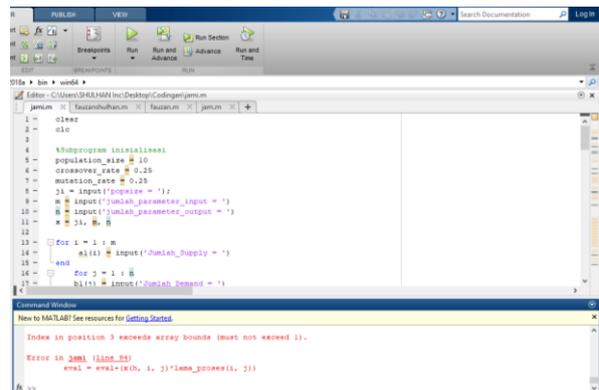


Gambar 4. Tampilan Data Hasil Pengoperasian terhadap Struktur *Syntax*

Berdasarkan Gambar 4 terlihat, bahwa hasil pengoperasian MATLAB pada bagian “Command Windows”.

Proses Evaluasi terhadap Struktur *Syntax* sebagai Bentuk Pemeriksaan Akhir

Tahapan setelah data tertampilkan pada bagian “Command Windows”, selanjutnya struktur *syntax* program dievaluasi. Evaluasi struktur *syntax* program berguna untuk perbaikan perintah yang tidak beroperasi, sehingga nilai tidak tertampilkan pada bagian “Command”. Tampilan hasil evaluasi, seperti ditunjukkan pada Gambar 5.

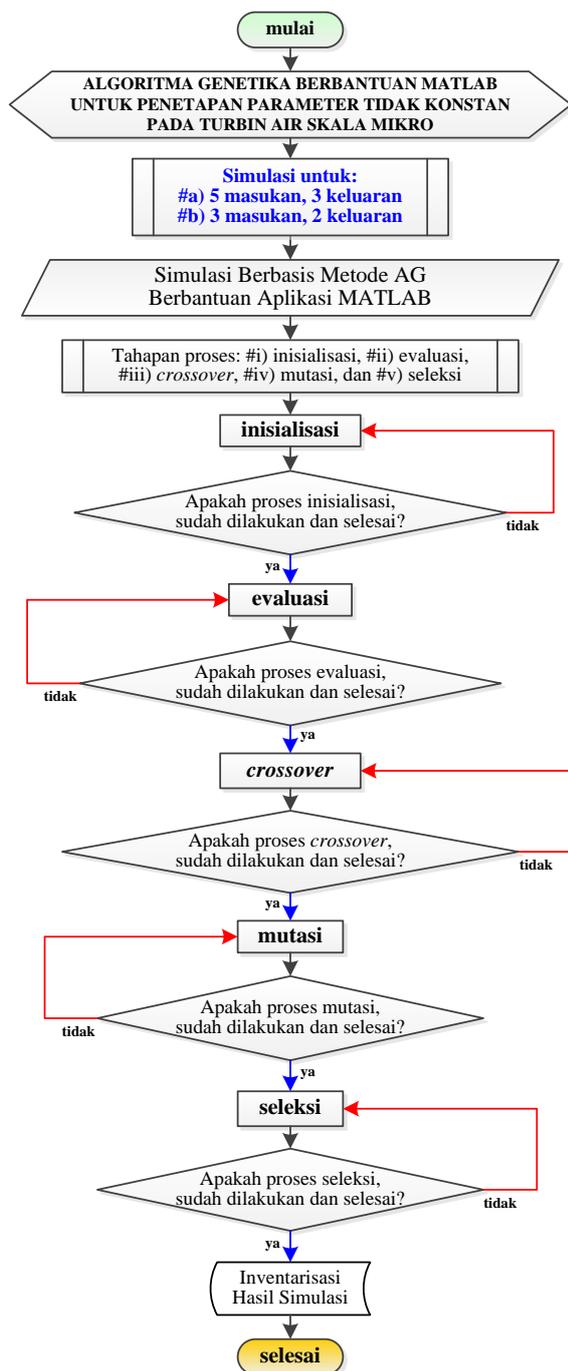


Gambar 5. Tampilan hasil evaluasi

Berdasarkan Gambar 5 dapat dijelaskan, bahwa terdapat hasil baris *syntax* berwarna merah yang merupakan fungsi dari *syntax* program yang tidak beroperasi, sehingga hasil tertampilkan tidak sesuai perintah.

Tahapan-tahapan Pelaksanaan Proses Simulasi Berbasis Metode AG Berbantuan Aplikasi MATLAB

Metode penelitian merupakan langkah-langkah untuk pencapaian sasaran penelitian berbentuk diagram alir (*flow chart*) [15-19] melalui batasan masalah. Diagram alir metode penelitian, seperti ditunjukkan pada Gambar 6.



Gambar 6. Diagram Alir Metode Penelitian

Berdasarkan Gambar 6 dapat dijelaskan, bahwa di dalam diagram alir metode penelitian terkandung dua sasaran penelitian. Tahapan-tahapan pada penyiapan aplikasi MATLAB, meliputi (i) penyusunan struktur *syntax* di *window*, (ii) pengoperasian atau pelaksanaan *running* terhadap struktur *syntax*, (iii) penampilan data hasil dari proses *running* terhadap struktur *syntax*, dan (iv) proses evaluasi terhadap struktur *syntax* sebagai

bentuk pemeriksaan akhir. Tahapan-tahapan pelaksanaan proses simulasi berbantuan metode AG berbantuan aplikasi MATLAB [5], [8], meliputi inisialisasi, evaluasi, *crossover*, mutasi, dan seleksi.

Tahap pertama dalam pengoperasian algoritma genetika, adalah pembuatan kromosom yang berisi jumlah parameter masukan (*input*) untuk perolehan parameter keluaran (*output*). Kromosom berbentuk matriks $m \times n$ (parameter *input* x parameter *output*), seperti yang ditunjukkan pada persamaan (1).

$$X_p = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1n} \\ X_{21} & X_{22} & \dots & X_{2n} \\ \dots & \dots & \dots & \dots \\ X_{m1} & X_{m2} & \dots & X_{mn} \end{bmatrix} \quad (1),$$

X_p merupakan kromosom ke- p , p merupakan jumlah populasi yang digunakan, dan X_{mn} merupakan gen pada parameter *input* (m) dan parameter *output* (n).

Tahap evaluasi merupakan tahap perhitungan parameter *input* terhadap parameter *output* berdasarkan jumlah variabel. Tahapan evaluasi berfungsi untuk perolehan nilai fungsi tujuan, yaitu minimasi dan/atau maksimasi. Prosedur evaluasi menggunakan tahapan, seperti persamaan (2).

$$\frac{(X_{11} * d_{11}) + (X_{12} * d_{12}) \dots (X_{1n} * d_{1n}) * m}{Eval. Total (X_p)} + \quad (2)$$

X_p merupakan kromosom ke p , p merupakan jumlah populasi yang digunakan, X_{mn} merupakan gen yang ada pada parameter *input* (m) dan parameter *output* (n), d_{mn} merupakan lama proses, m merupakan parameter *input*, dan *Eval. Total* (X_p) merupakan total biaya pada proses evaluasi.

Penentuan kromosom pada tahap *crossover* dilakukan secara acak dan dipilih berdasarkan kromosom yang terbentuk dari tahap inisialisasi. Perolehan dua kromosom gen yang disilangkan ditentukan melalui tiga tahapan, yaitu:

Tahap-1. Pembuatan dua matriks baru dari matriks awal yang telah ditentukan dengan cara seperti pada persamaan (3) dan (4).

$$d_{ij} = (x^l_{ij} + x^r_{ij})/2 \quad (3),$$

$$r_{ij} = (x^l_{ij} + x^r_{ij}) \bmod 2 \quad (4),$$

dengan:

d_{ij} merupakan bentuk baru dari matriks D dan (r_{ij}) merupakan bentuk baru dari matriks R .

Tahap-2. Pembagian terhadap matriks R menjadi dua matriks baru, seperti ditunjukkan pada persamaan (5).

$$R = R^1 + R^2 \quad (5).$$

Tahap-3. Diperoleh dua matriks baru (*offspring*) berdasarkan dua persamaan, seperti ditunjukkan pada persamaan (6) dan (7).

$$X^1 = D + R^1 \quad (6),$$

$$X^2 = D + R^2 \quad (7),$$

dengan:

X^1 dan X^2 merupakan matriks baru yang terbentuk dari penjumlahan antara matriks D dan matriks R^1 dan antara matriks D dan R^2 .

Proses mutasi dilakukan untuk pembentukan submatriks yang telah dihasilkan dari proses *crossover* melalui tiga tahapan, yaitu:

Tahap-1. Pembuatan submatriks Y ($p \times q$) dari kromosom dengan status *parent* X ($m \times n$) yang dipilih secara acak, dengan p dan q ditentukan secara acak dengan ketentuan $2 \leq p \leq m$ dan $2 \leq q \leq n$.

Tahap-2. Pelaksanaan relokasi nilai matriks *parent* ke submatriks dengan pengaturan:

$$a_i^y = \sum_i \in \{i_1 \dots i_q\} Y_{ij} \quad \text{dengan: } i = i_1, i_2, \dots, i_p$$

$$b_j^y = \sum_j \in \{j_1 \dots j_q\} Y_{ij} \quad \text{dengan: } j = j_1, j_2, \dots, j_p$$

Tahap-3. Penggantian nilai matriks *parent* dengan nilai baru dari submatriks melalui tahapan:

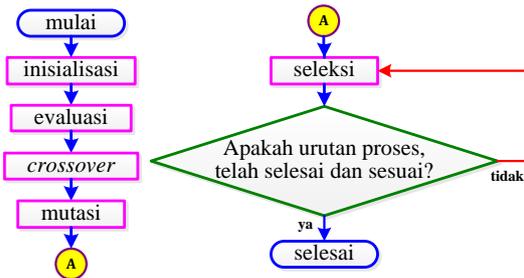
- Ilustrasi kromosom *parent* yang dipilih secara acak;
- Ilustrasi submatriks Y ($p \times q$);
- Ilustrasi kromosom hasil proses mutasi, diperoleh: dan,
- Ilustrasi kromosom *parent* yang dipilih secara acak.

Proses terakhir dari urutan pada algoritma genetika berupa proses seleksi untuk perolehan kromosom dengan nilai lebih besar.

HASIL DAN PEMBAHASAN

Pengoperasian algoritma genetika dalam pemecahan proses optimasi, bertujuan untuk pembelajaran penerapan AG terhadap suatu kendala dan kesesuaiannya terhadap struktur data pada sebuah masalah yang dipecahkan. Pemecahan masalah ini dicirikan dengan fungsi

tujuan untuk memaksimalkan *output*. Tahapan-tahapan penyelesaian berbasis AG, seperti ditunjukkan pada Gambar 7.



Gambar 7. Tahapan-tahapan Penyelesaian Berbasis AG

Berdasarkan Gambar 7 dapat dijelaskan, bahwa tahapan penyelesaian AG dimulai dari proses inisialisasi, evaluasi, *crossover*, mutasi, dan seleksi.

Simulasi terhadap Ukuran Matriks 5x3

Matriks 5x3 dimaknai sebagai lima masukan dan tiga keluaran. Lima masukan merupakan lima parameter berpengaruh pada turbin air (bersifat konstan dan tidak konstan), sedangkan tiga keluaran merupakan tiga parameter berpengaruh bersifat tidak konstan pada turbin air.

Inisialisasi untuk matriks 5x3

Tampilan subprogram inisialisasi, seperti ditunjukkan pada Gambar 8.

```

Editor - C:\Users\SHULHAN\inc\Desktop\Codingan\Untitled.m
Untitled.m
16 subprogram evaluasi
17 c = input('lama Proses = ');
18
19 a_1 = input('nilai input = ');
20 eval_11 = a_1 * c;
21 a_2 = input('nilai input = ');
22 eval_12 = a_2 * c;
23 a_3 = input('nilai input = ');
24 eval_13 = a_3 * c;
25 a_4 = input('nilai input = ');
26 eval_14 = a_4 * c;
27 a_5 = input('nilai input = ');
28 eval_15 = a_5 * c;
29
30 eval_total = eval_11 + eval_12 + eval_13 + eval_14 + eval_15
31 disp('Nilai Evaluasi = '), disp(eval_total);
    
```

Gambar 8. Tampilan subprogram inisialisasi

Berdasarkan struktur *syntax* subprogram untuk proses inisialisasi, diperoleh kromosom hasil *running* terhadap subprogram inisialisasi. Penetapan waktu tersebut berkaitan dengan proses kemunculan bilangan acak pada aplikasi MATLAB untuk keterbentukan matriks kromosom berjumlah lima buah matriks, yaitu:

$$X_1 = \begin{bmatrix} 0.9797 & 0.5949 & 0.1174 \\ 0.4389 & 0.2622 & 0.2967 \\ 0.1111 & 0.6028 & 0.3188 \\ 0.2581 & 0.7112 & 0.4242 \\ 0.4087 & 0.2217 & 0.5079 \end{bmatrix}$$

$$X_2 = \begin{bmatrix} 0.0855 & 0.7303 & 0.9631 \\ 0.2625 & 0.4886 & 0.5468 \\ 0.8010 & 0.5785 & 0.5211 \\ 0.0292 & 0.2373 & 0.2316 \\ 0.9289 & 0.4588 & 0.4889 \end{bmatrix}$$

$$X_3 = \begin{bmatrix} 0.6241 & 0.0377 & 0.2619 \\ 0.6791 & 0.8852 & 0.3354 \\ 0.3955 & 0.9133 & 0.6797 \\ 0.3674 & 0.7962 & 0.1366 \\ 0.9880 & 0.0987 & 0.7212 \end{bmatrix}$$

$$X_4 = \begin{bmatrix} 0.1068 & 0.9037 & 0.0305 \\ 0.6538 & 0.8909 & 0.7441 \\ 0.4942 & 0.3342 & 0.5000 \\ 0.7791 & 0.6987 & 0.4799 \\ 0.7150 & 0.1978 & 0.9047 \end{bmatrix}$$

$$X_5 = \begin{bmatrix} 0.6099 & 0.1829 & 0.1679 \\ 0.6177 & 0.2399 & 0.9787 \\ 0.8594 & 0.8865 & 0.7127 \\ 0.8055 & 0.0287 & 0.5005 \\ 0.5767 & 0.4899 & 0.4711 \end{bmatrix}$$

Jumlah kromosom terbentuk berdasarkan jumlah variabel terbesar, baik variabel masukan atau keluaran, kemudian hasil proses inisialisasi diteruskan menuju proses *crossover*.

Evaluasi untuk Matriks 5x3

Tampilan subprogram evaluasi ditunjukkan pada Gambar 9.

Gambar 9. Tampilan Subprogram Evaluasi

Berdasarkan Gambar 9 dapat dijelaskan, bahwa:

```
Lama Proses = 5
m = 5
nilai tinggi jatuh = 1
a_1 = 1
eval_11 = 5
nilai debit air = 1
a_2 = 1
eval_12 = 5
nilai efisiensi mekanis = 5
a_3 = 5
eval_13 = 25
```

```
nilai densitas air = 1
a_4 = 1
eval_14 = 5
nilai gaya gravitasi bumi = 1
a_5 = 1
eval_15 = 5
eval_total = eval_11 + eval_12 +
eval_13 + eval_14 + eval_15
eval_total = 45
Nilai Evaluasi = 45
```

Nilai evaluasi berfungsi untuk perolehan nilai fungsi tujuan yang dihasilkannya, sebagai nilai untuk pembuktian *logical array*.

Crossover untuk Matriks 5x3

Tampilan struktur *syntax* subprogram *crossover* untuk matriks 5x3, ditunjukkan pada Gambar 10.

Gambar 10. Tampilan Struktur Syntax Subprogram *Crossover* untuk Matriks 5x3

Gambar 10 menjelaskan, bahwa terhadap subprogram *crossover* dilakukan *running*, sehingga diperoleh kromosom hasil *crossover* sebagai berikut:

Kromosom terpilih secara acak, yaitu:

$$X_5 = \begin{bmatrix} 0.6099 & 0.1829 & 0.1679 \\ 0.6177 & 0.2399 & 0.9787 \\ 0.8594 & 0.8865 & 0.7127 \\ 0.8055 & 0.0287 & 0.5005 \\ 0.5767 & 0.4899 & 0.4711 \end{bmatrix}$$

Pemilihan terhadap kromosom X_5 , karena pemilihan kromosom secara acak dengan proses perulangan sebanyak lima kali sesuai dengan jumlah iterasi yang telah ditentukan.

Matriks baru dari kromosom terbentuk dengan status *parent*, yaitu:

$$R = \begin{bmatrix} 2.0000 & 0.6342 & 0.6641 \\ 2.0000 & 0.5201 & 2.0000 \\ 2.0000 & 2.0000 & 2.0000 \\ 2.0000 & 0.9427 & 2.0000 \\ 2.0000 & 0.0202 & 0.0578 \end{bmatrix}$$

$$D = \begin{bmatrix} 1.1099 & 0.6829 & 0.6679 \\ 1.1177 & 0.7399 & 1.4787 \\ 1.3594 & 1.3865 & 1.2127 \\ 1.3055 & 0.5287 & 1.0005 \\ 1.0767 & 0.9899 & 0.9711 \end{bmatrix}$$

Matriks baru yang terbentuk dari matriks *R*, menggunakan persamaan (5), sehingga diperoleh:

$$R^1 = \begin{bmatrix} 1.0000 & 0.3171 & 0.3321 \\ 1.0000 & 0.2601 & 1.0000 \\ 1.0000 & 1.0000 & 1.0000 \\ 1.0000 & 0.4714 & 1.0000 \\ 1.0000 & 0.0101 & 0.0289 \end{bmatrix}$$

$$R^2 = \begin{bmatrix} 1.0000 & 0.3171 & 0.3320 \\ 1.0000 & 0.2600 & 1.0000 \\ 1.0000 & 1.0000 & 1.0000 \\ 1.0000 & 0.4713 & 1.0000 \\ 1.0000 & 0.0101 & 0.0289 \end{bmatrix}$$

Kromosom kondisi *offspring* diperoleh setelah digunakan persamaan (6) dan (7), yaitu:

$$X^1 = D + R^1 \text{ dan } X^2 = D + R^2.$$

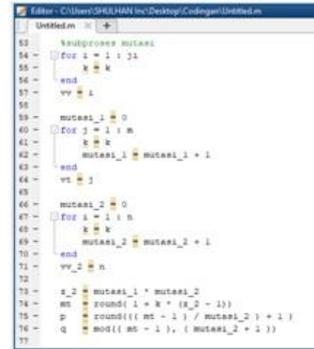
$$X^1 = \begin{bmatrix} 2.1000 & 1.0000 & 1.0000 \\ 2.1177 & 1.2600 & 2.4787 \\ 2.0000 & 2.3865 & 2.2127 \\ 2.3055 & 1.0001 & 2.0005 \\ 2.0767 & 1.0000 & 1.0000 \end{bmatrix}$$

$$X^2 = \begin{bmatrix} 2.1099 & 1.0000 & 0.9999 \\ 2.1177 & 0.9999 & 2.4787 \\ 2.0000 & 2.3865 & 2.2127 \\ 2.3055 & 1.0000 & 2.0005 \\ 2.0767 & 1.0000 & 1.0000 \end{bmatrix}$$

Kromosom hasil proses *crossover*, selanjutnya diteruskan ke proses mutasi untuk pembentukan submatriks.

Mutasi untuk Matriks 5x3

Tampilan struktur *syntax* subprogram mutasi untuk matriks 5x3, ditunjukkan pada Gambar 11.



Gambar 11. Tampilan Struktur *Syntax* Subprogram Mutasi untuk Matriks 5x3

Gambar 11 menunjukkan, bahwa hasil proses berdasarkan subprogram mutasi yang telah dilakukan dengan proses *running*, menghasilkan:

$$R = \begin{bmatrix} 2.0000 & 0.6342 & 0.6641 \\ 2.0000 & 0.5201 & 2.0000 \\ 2.0000 & 2.0000 & 2.0000 \\ 2.0000 & 0.9427 & 2.0000 \\ 2.0000 & 0.0202 & 0.0578 \end{bmatrix}$$

Kromosom *R* merupakan kromosom dengan status *parents* yang dihasilkan melalui tahap *crossover*. Submatriks *Y* dihasilkan atas proses pengoperasian matriks *offspring* yang terbentuk melalui proses *crossover*. Pembuatan submatriks *Y* ($p \times q$) dari kromosom dengan status *parent* *X* ($m \times n$) yang dipilih secara acak, dengan p dan q ditentukan secara acak dengan ketentuan $2 \leq p \leq m$ dan $2 \leq q \leq n$, diperoleh *Y* ($p \times q$), yaitu:

$$\text{Matriks: } Y = \begin{bmatrix} 2.0000 & 2.0000 \\ 0.9427 & 2.0000 \\ 0.0202 & 0.0578 \end{bmatrix}$$

Submatriks *Y* setelah relokasi nilai, menjadi

$$Y = \begin{bmatrix} 2.0000 & 2.0000 \\ 0.8427 & 2.1000 \\ 0.0578 & 0.0202 \end{bmatrix}$$

Kromosom hasil proses mutasi, menghasilkan:

$$X = \begin{bmatrix} 2.0000 & 0.6342 & 0.6641 \\ 2.0000 & 0.5201 & 2.0000 \\ 2.0000 & 2.0000 & 2.0000 \\ 2.0000 & 0.8427 & 2.1000 \\ 2.0000 & 0.0578 & 0.0202 \end{bmatrix}$$

Kromosom hasil mutasi terbentuk melalui hasil pengoperasian terhadap kromosom dengan status “*parents*” yang dihasilkan oleh proses *crossover*, sehingga diperoleh nilai yang sama ($p \times q$) atau matriks ukuran 5 x 3 dengan proses relokasi.

Seleksi untuk Matriks 5x3

Tampilan struktur *syntax* subprogram seleksi untuk matriks 5x3, ditunjukkan pada Gambar 12.

```

78 %subprogram seleksi
79 so = 0
80 switch 2
81     case('k')
82         y = k
83     case('k + 1')
84         y = k + 1
85     otherwise y = k
86     sprintf('\nChromosome Hasil Seleksi = \n', k, y)
87 end
88
89 so = so + 1
90 prob = 1 / so
91 pb = 0
92 pb = pb + prob
93 komprob = pb
94 if R <= komprob
95     else R > (komprob - 1)
96 end
    
```

Gambar 12. Tampilan Struktur *Syntax* Subprogram Seleksi untuk Matriks 5x3

Perolehan tersebut didasarkan pada ketentuan:

- Pengurutan kromosom (minimasi dan/atau maksimasi),
- Penyusunan *score* dan fungsi distribusi probabilitas,
- Penyusunan fungsi distribusi kumulatif, dan
- Pembangkitan bilangan acak.

Untuk kondisi dimana bilangan acak lebih kecil dari fungsi distribusi probabilitas, maka kromosom tersebut dipertahankan dan diperoleh:

```

score = 5
probabilitas = 0.6000
komprob = 0.6000
    
```

Nilai *score*, probabilitas, dan distribusi kumulatif diperoleh berdasarkan hasil dari pengoperasian pada tahapan seleksi, yaitu kemunculan matriks ukuran 5 x 3 sebagai *logical array*, yaitu:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Susunan matriks akhir yang terbentuk pada tahapan seleksi pada algoritma genetika berupa matriks dengan nilai satu (benar).

Logical array merupakan sebuah sistem pengubah dari sebuah elemen atau nilai pada aplikasi MATLAB menjadi nilai logika. Elemen atau nilai yang bukan bernilai 0 (nol) diubah menjadi 1 (benar) dan elemen atau nilai dengan 0 (nol) diubah menjadi 0 (salah). Elemen atau nilai

kompleks tidak dapat diubah menjadi nilai logika, sehingga menjadi *error*. Berdasarkan kondisi *logical array* yang bernilai benar keseluruhannya, maka penetapan seluruh parameter berpengaruh (gravitasi bumi, densitas air, tinggi jatuh, debit air, dan efisiensi mekanis) sebagai masukan dan penetapan parameter berpengaruh bersifat tidak konstan (tinggi jatuh, debit air, dan efisiensi mekanis) sebagai keluaran, telah sesuai dalam hal penetapannya. Keberadaan *logical array* untuk matriks berukuran 5x3 merupakan syarat untuk pelaksanaan simulasi lanjutan untuk matriks berukuran 3x2. Tiga masukan merupakan tiga parameter berpengaruh bersifat tidak konstan pada turbin air (tinggi jatuh, debit air, dan efisiensi mekanis), sedangkan dua parameter sebagai keluaran merupakan dua parameter berpengaruh bersifat tidak konstan pada turbin air skala mikro, yaitu debit air dan efisiensi mekanis turbin.

Simulasi terhadap Ukuran Matriks 3x2

Matriks 3x2 dimaknai sebagai tiga masukan dan dua keluaran. Penjelasan terhadap penetapan matriks berukuran 3x2 didasarkan kepada jumlah parameter berpengaruh bersifat tidak konstan yang meliputi tinggi jatuh air, debit air dan efisiensi mekanis sebagai masukan, sedangkan dua keluaran ditetapkan sebagai parameter berpengaruh tidak konstan, yaitu parameter debit air dan efisiensi mekanis. Penetapan turbin air skala mikro, merupakan implementasi turbin yang relatif tidak dipengaruhi oleh tinggi jatuh air, sehingga parameter berpengaruh bersifat tidak konstan menjadi tersisa dua, sehingga proses simulasi lanjutannya berupa tiga masukan dan dua keluaran.

Inisialisasi untuk Matriks 3x2

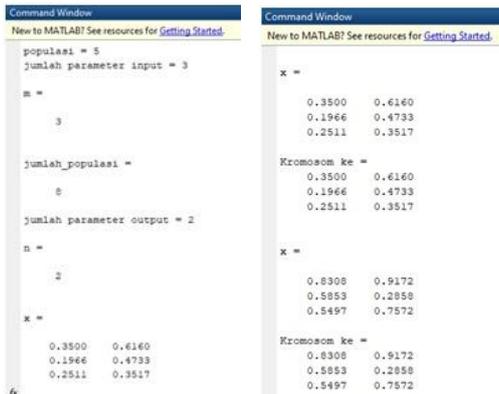
Tampilan subprogram inisialisasi matriks 3x2, seperti ditunjukkan pada Gambar 13.

```

1 - clear
2 - clc
3
4 %subprogram inisialisasi
5 j1 = input('populasi = ');
6 m = input('jumlah parameter input = ');
7 jumlah_populasi = 2^m
8 n = input('jumlah parameter output = ');
9
10 for i = 1 : m
11     x = rand(m, n)
12     disp('Kromosom ke = '), disp(x)
13 end
14
    
```

Gambar 13. Tampilan Subprogram Inisialisasi Matriks 3x2

Berdasarkan Gambar 13, diperoleh ilustrasi kromosom hasil inisialisasi, yang ditunjukkan pada Gambar 14.



Gambar 14. Ilustrasi Kromosom Hasil Inisialisasi

Gambar 14 menunjukkan bahwa jumlah kromosom terbentuk berdasarkan jumlah variabel terbesar, baik variabel masukan atau keluaran, sehingga ditampilkan tiga kromosom.

$$X_1 = \begin{bmatrix} 0.3500 & 0.6160 \\ 0.1966 & 0.4733 \\ 0.2511 & 0.3517 \end{bmatrix}$$

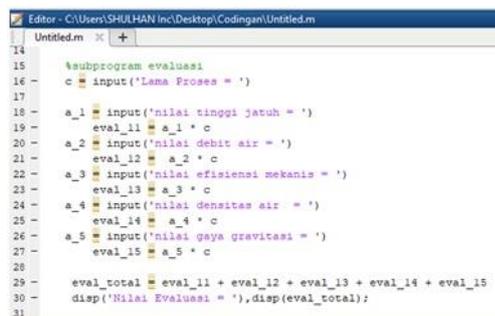
$$X_2 = \begin{bmatrix} 0.8308 & 0.9172 \\ 0.5853 & 0.2858 \\ 0.5497 & 0.7572 \end{bmatrix}$$

$$X_3 = \begin{bmatrix} 0.3804 & 0.0540 \\ 0.5678 & 0.5308 \end{bmatrix}$$

Tahapan lanjutan, hasil proses inisialisasi diteruskan menuju proses *crossover*.

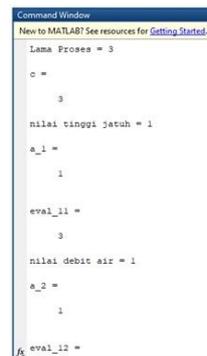
Evaluasi untuk Matriks 3x2

Tahapan evaluasi berfungsi untuk perolehan nilai minimasi dan/atau maksimasi dengan penggunaan persamaan (2). Tampilan subprogram evaluasi matriks 3x2, ditunjukkan pada Gambar 15.



Gambar 15. Tampilan Subprogram Evaluasi Matriks 3x2

Berdasarkan Gambar 15, diperoleh ilustrasi hasil evaluasi, seperti ditunjukkan pada Gambar 16.



Gambar 16. Ilustrasi Hasil Eksekusi pada Subprogram Evaluasi

Berdasarkan Gambar 16 dapat dijelaskan, bahwa

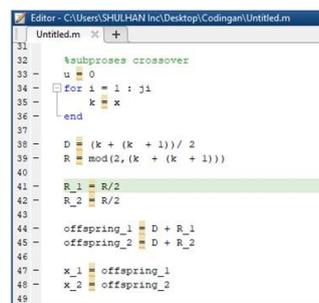
```
Lama Proses = 3
c = 3
nilai tinggi jatuh = 1
a_1 = 1
eval_11 = 3
nilai debit air = 1
a_2 = 1
eval_12 = 3
nilai efisiensi mekanis = 5
a_3 = 5
eval_13 = 15

eval_total = eval_11 + eval_12 +
eval_13 = 3 + 3 + 15
eval_total = 21
Nilai Evaluasi = 21
```

Nilai evaluasi berfungsi untuk perolehan nilai fungsi tujuan yang dihasilkannya, sehingga sebagai nilai untuk pembuktian *logical array*.

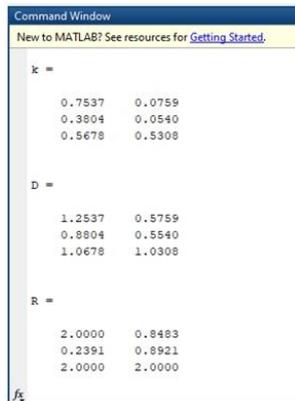
Crossover untuk Matriks 3x2

Tampilan struktur *syntax* subprogram *crossover* untuk matriks 3x2 ditunjukkan pada Gambar 17.



Gambar 17. Tampilan Struktur *Syntax* Subprogram *Crossover* untuk Matriks 3x2

Berdasarkan Gambar 17, diperoleh ilustrasi kromosom hasil *crossover*, ditunjukkan pada Gambar 18.



Gambar 18. Ilustrasi Kromosom Hasil *Crossover*

Gambar 18 dapat dijelaskan bahwa: Kromosom terpilih secara acak, yaitu:

$$X_3 = \begin{bmatrix} 0.7537 & 0.0759 \\ 0.3804 & 0.0540 \\ 0.5678 & 0.5308 \end{bmatrix}$$

Matriks baru yang terbentuk dari kromosom *parent*, yaitu:

$$D = \begin{bmatrix} 1.2537 & 0.5759 \\ 0.8804 & 0.5540 \\ 1.0678 & 1.0308 \end{bmatrix}$$

$$R = \begin{bmatrix} 2.0000 & 0.8483 \\ 0.2391 & 0.8921 \\ 2.0000 & 2.0000 \end{bmatrix}$$

Matriks baru yang terbentuk dari matriks *R*, yaitu:

$$R = \begin{bmatrix} 1.0000 & 0.4241 \\ 0.1196 & 0.4460 \\ 1.0000 & 1.0000 \end{bmatrix}$$

Kromosom *offspring*, yaitu:

$$X = \begin{bmatrix} 2.2537 & 1.0000 \\ 1.0000 & 1.0000 \\ 2.0678 & 2.0308 \end{bmatrix}$$

Kromosom hasil proses *crossover*, selanjutnya diteruskan ke proses mutasi untuk pembentukan submatriks.

Mutasi untuk Matriks 3x2

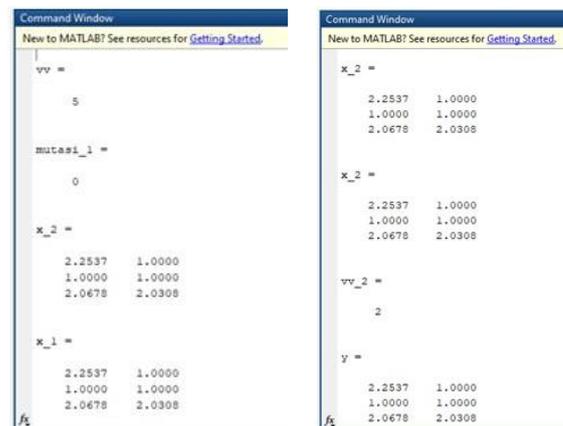
Tampilan struktur *syntax* subprogram mutasi untuk matriks 3x2, ditunjukkan pada Gambar 19.

```

50 subproses mutasi
51 vv = 1
52
53 mutasi_1 = 0
54 for j = 1 : m
55     x_2 = x_2
56     x_1 = x_1
57 end
58 vt = j
59
60 mutasi_2 = 0
61 for i = 1 : n
62     m_2 = m_2
63 end
64 vv_2 = n
65
66 switch 2
67     case('k_2')
68         y = x_2
69         case('n_2 + 1')
70             y = offspring_2 + 1
71         otherwise y = offspring_2
72         fprintf('\nChromosome Hasil Relokasi = %s', offspring_2, y)
73     end
74
    
```

Gambar 19. Tampilan Struktur *Syntax* Subprogram Mutasi untuk Matriks 3x2

Berdasarkan Gambar 19, diperoleh ilustrasi kromosom hasil mutasi, yang ditunjukkan pada Gambar 20.



Gambar 20. Ilustrasi Kromosom hasil Mutasi

Berdasarkan Gambar 20 dapat dijelaskan, bahwa berdasarkan subprogram mutasi yang telah dilakukan dengan proses *running*, diperoleh:

- Ilustrasi kromosom *parent* yang dipilih secara acak

$$X = \begin{bmatrix} 2.2537 & 1.0000 \\ 1.0000 & 1.0000 \\ 2.0678 & 2.0308 \end{bmatrix}$$

- Submatriks *Y* (*p* x *q*)

$$Y = \begin{bmatrix} 1.0000 & 1.0000 \\ 2.0678 & 2.0308 \end{bmatrix}$$

- Submatriks *Y* setelah relokasi nilai, menjadi:

$$Y = \begin{bmatrix} 0.9000 & 1.1000 \\ 2.0308 & 2.0678 \end{bmatrix}$$

- Kromosom hasil mutasi, diperoleh:

$$X = \begin{bmatrix} 2.2537 & 1.0000 \\ 0.9000 & 1.1000 \\ 2.0308 & 2.0678 \end{bmatrix}$$

Seleksi untuk Matriks 3x2

Tampilan struktur *syntax* subprogram seleksi untuk matriks 3x2, seperti ditunjukkan pada Gambar 21.

```

74
75 %subprogram seleksi
76 sc = 0
77
78 sc = sc + j
79 prob = i / sc
80 pb = 0
81 pb = pb + prob
82 komprob = pb
83 if R <= komprob
84 else R > (komprob - 1)
85 end
    
```

Gambar 21. Tampilan Struktur *Syntax* Subprogram Seleksi untuk Matriks 3x2

Berdasarkan Gambar 21, diperoleh ilustrasi kromosom hasil seleksi, yang ditunjukkan pada Gambar 22.

```

Command Window
New to MATLAB? See resources for Getting Started.

pb =
    0

pb =
    0.6667

komprob =
    0.6667

ans =
    3x2 logical array

    1    1
    1    1
    1    1

f3 >>
    
```

Gambar 22. Ilustrasi Kromosom Hasil Seleksi

Dari Gambar 22 dapat dijelaskan bahwa berdasarkan subprogram mutasi yang telah dilakukan dengan proses *running*, diperoleh:

```

sc = 0
sc = 3
prob = 0.6667
pb = 0
pb = 0.6667
komprob = 0.6667
    
```

Hasil untuk 3x2 *logical array*, yaitu:

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Susunan matriks akhir yang terbentuk pada tahapan seleksi pada algoritma genetika berupa matriks dengan nilai satu (benar).

Logical array merupakan sebuah sistem pengubah dari sebuah elemen atau nilai pada aplikasi MATLAB menjadi nilai logika. Elemen atau nilai yang bukan bernilai 0 (nol) diubah menjadi 1 (benar) dan elemen atau nilai dengan 0 (nol) diubah menjadi 0 (salah). Elemen atau nilai kompleks tidak dapat diubah menjadi nilai logika, sehingga menjadi *error*. Berdasarkan kondisi *logical array* yang bernilai benar keseluruhannya, maka penetapan sejumlah parameter berpengaruh tidak konstan sebagai masukan (tinggi jatuh, debit air, dan efisiensi mekanis) dan penetapan parameter bernilai tidak konstan sebagai keluaran (debit air dan efisiensi mekanis), telah sesuai dalam hal penempatannya.

KESIMPULAN DAN SARAN

Kesimpulan

Simulasi pendugaan untuk optimasi nilai daya keluaran dari turbin air yang berdasarkan pada lima parameter berpengaruh, telah dilakukan. Lima parameter berpengaruh terbagi ke dalam dua kelompok, yaitu kelompok bersifat konstan dan tidak konstan. Kelompok parameter berpengaruh bersifat konstan, meliputi gaya gravitasi bumi dan densitas air, sedangkan kelompok parameter berpengaruh bersifat tidak konstan, meliputi tinggi jatuh air, debit air, dan efisiensi mekanis turbin. Parameter berpengaruh bersifat tidak konstan sebagai penentu optimasi daya keluaran dari turbin air.

Simulasi pertama dilakukan berkaitan dengan turbin air skala besar, sedang, dan kecil, sehingga ditetapkan lima masukan dengan seluruh parameter berpengaruh sebagai masukan (densitas air, gaya gravitasi bumi, tinggi jatuh air, debit air, dan efisiensi mekanis), sedangkan tiga parameter berpengaruh bersifat tidak konstan sebagai keluaran (tinggi jatuh air, debit air, dan efisiensi mekanis). Hasil tertampilkan lima kromosom sebagai “parent” dan *logical array* bernilai 1 (benar) di akhir proses, keseluruhan berbentuk matriks ukuran 5x3. Simulasi kedua dilakukan berkaitan dengan turbin air skala mikro, sehingga ditetapkan tiga parameter berpengaruh bersifat tidak konstan sebagai masukan (tinggi jatuh air, debit air, dan efisiensi mekanis) dan dua parameter berpengaruh bersifat tidak konstan sebagai keluaran (debit air dan efisiensi mekanis). Hasil tertampilkan tiga kromosom sebagai “parent” dan *logical array* bernilai 1 (benar) di akhir proses dan merupakan hasil yang sesuai dengan kriteria, keseluruhan berbentuk matriks ukuran 3x2.

Saran

Untuk penelitian lanjutan, perlu dilakukan simulasi lebih lanjut terkait dengan diferensiasi terhadap subparameter pada parameter berpengaruh bersifat tidak konstan (terutama parameter efisiensi mekanis), agar dapat ditentukan subparameter untuk peningkatan daya keluaran pada turbin air, yaitu subparameter “mana saja” sebagai masukan (m) terpilih dan subparameter “mana saja” sebagai keluaran (n) terpilih.

DAFTAR PUSTAKA

- [1] Aziz, A., Prihandono, B., and Ilhamsyah, I. Algoritma Genetika pada Pemrograman Linear dan Nonlinear, [e-journal] *Bimaster: Buletin Ilmiah Matematika, Statistika dan Terapannya*, 5(3), pp. 265-274, 2016, <http://jurnal.untan.ac.id/index.php/jbmstr/article/view/16935/14505>
- [2] Fauzan, M., Setiawan, B., and Indriati, I. Algoritma Genetika Untuk Optimasi Fuzzy Time Series Dalam Memprediksi Debit Air (Studi Kasus: PDAM Indramayu). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 1, pp. 831-835, 2019, <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/4235>
- [3] Soesanto, Q.M.B., Widiyanto, P., Susatyo, A., and Yazid, E. Cascade optimization of an axial-flow hydraulic turbine type propeller by a genetic algorithm,” *International Journal Technolog*, 10(1), 200–211, 2019, <https://dx.doi.org/10.14716/ijtech.v10i1.1744>
- [4] Juni, J., Yusva, N.A., Pandiangan, I.O.M., and Nugroho, H. 2022. Penerapan Metode Particle Swarm Optimization dan Genetic Algorithm pada Optimisasi Sudut Kelengkungan Turbin Air Banki untuk Mendapatkan Efisiensi Daya Optimal, [e-journal] *Energi dan Kelistrikan: Jurnal Ilmiah*, 14(1), pp. 82-89, 2022, <https://stt-pln.e-journal.id/energi/article/view/1636>
- [5] Widiyatun, F., Sumarni, R.A., and Okyranida, I.Y. GUI MATLAB untuk Pembuatan Kalkulator Fisika Gerak Media Belajar Mahasiswa Teknik Informatika, *Seminar Nasional Sains*, 2(1), pp. 435–442, 2021, <http://proceeding.unindra.ac.id/index.php/sinasis/article/view/5381/1174>
- [6] Fauzi, F., and Hermanto, A. Analisis Kelayakan Ekonomi dan Finansial Pembangkit Listrik Tenaga Air (PLTA) dengan Menggunakan Software Visual Basic for Applications (VBA), [e-journal] *Barometer*, 6(2), pp. pp. 335–342, 2021, <https://dx.doi.org/10.35261/barometer.v6i2.4887>
- [7] Rustandi, D., Firman, L.O.M., and Harahap, S. Perancangan dan Optimasi Desain Turbin Francis pada Pembangkit Listrik Tenaga Minihidro di Bendungan Jatibarang Kota Semarang, *Teknobiz: Jurnal Ilmiah Program Studi Magister Teknik Mesin*, 9(3), pp. 20–28, 2019, <https://journal.univpancasila.ac.id/index.php/teknobiz/article/view/1152>
- [8] MATLAB. no-year. Create an Executable Notebook, [online], Available: <https://www.mathworks.com/products/matlab/live-editor>
- [9] Ratulangi, D.R.G. Penerapan Algoritma Genetika Untuk Optimasi Penawaran Biaya Pekerjaan Konstruksi Dengan Bantuan Software Matlab, *Jurnal Ilmiah Media Engineering*, 9(1), pp. 2087–9334, 2019, <https://ejournal.unsrat.ac.id/index.php/jime/article/view/31321>
- [10] Putra, I.K.W.T., Utama, N.P.S., and Setiawan, I.N. 2019. Optimasi Penempatan Titik Interkoneksi Distributed Generation (DG) PLTM Muara pada Penyulang Panji dengan Metode Genetic Algorithm (GA), *J. SPEKTRUM*, 6(2), pp. 141-147, 2019, <https://dx.doi.org/10.24843/spektrum.2019.v06.i02.p19>
- [11] Harjanti, W.N., and Darsono, S. Optimasi Lepasn Waduk Logung dengan Algoritma Genetik, *SIKLUS: Jurnal Teknik Sipil*, 6(1), pp. 38–48, 2020, <https://dx.doi.org/10.31849/siklus.v6i1.3693>
- [12] Zulmahendra, and Derta, S. Penjadwalan Kuliah Otomatis Menggunakan Algoritma Genetika Program Studi PTIK IAIN Bukittinggi, *JSI: Jurnal Sistem Informasi*, 12(2), pp. 2128–2137, 2020, <https://dx.doi.org/10.36706/jsi.v12i2.12097>
- [13] Candra, E.N., Cholissodin, I., and Wihandika, R.C. Klasifikasi Status Gizi Balita menggunakan Metode Optimasi Random Forest dengan Algoritme Genetika (Studi Kasus: Puskesmas Cakru), *Jurnal*

- Pengembangan Teknologi Informasi dan Ilmu Komputer*, 6(5), pp. 2188–2197, 2022, <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/11010>
- [14] Cynthia, E.P., Ismanto, E., and Alrian, R. Integrasi Geographical Information System dan Algoritma Genetika Dalam Penentuan Lokasi Objek Wisata, Seminar Nasional Teknologi Informasi Komunikasi dan Industri (SNTIKI), 11, 83–92, 2019, <http://ejournal.uinsuska.ac.id/index.php/SNTIKI/article/view/7843/4435>
- [15] Goeritno, A., Nurmansyah, D., and Maswan, M. Safety instrumented systems to investigate the system of instrumentation and process control on the steam purification system, *International Journal of Safety and Security Engineering*, 10(5), pp. 609-616, 2020, <https://doi.org/10.18280/ijssse.100504>
- [16] Goeritno, A., Nugraha, I., Rasiman, S., and Johan, A. Injection Current into the Power Transformer as an Internal Fault Phenomena for Measuring the Differential Relay Performance, *Instrumentation Measure*
- Métrologie*, 19(6), pp. 443-451, 2020, <https://doi.org/10.18280/i2m.190605>
- [17] Goeritno, A., Setyawibawa, I., Suhartono, D. 2021. Designing a microcontroller-based half duplex interface device drove by the touch-tone signal, *J. INFOTEL*, 13(4), pp. 205-215, 2021, <https://dx.doi.org/10.20895/infotel.v13i4.712>
- [18] Goeritno, A. and Setyawibawa, I. An Electronic Device Reviewed by Diagnosing on the Module Embodiment, *International Journal of Electronics and Communications Systems*, 1(2), pp. 41-55, 2021, <http://ejournal.radenintan.ac.id/index.php/IJECS/article/view/10383/pdf>
- [19] Prayudyanto, M.N., Goeritno, A., Ikhsan, S.H.A., and Taqwa, F.M.L. Designing a model of the early warning system on the road curvature to prevent the traffic accidents, *International Journal of Safety and Security Engineering*, 12(3), pp. 291-298, 2022, <https://doi.org/10.18280/ijssse.120303>