

DIAGNOSIS KERUSAKAN PADA SISTEM KOMPRESOR PRODUKSI GAS MENGGUNAKAN ALGORITMA GENETIKA

Amelia Vega¹, Rindu Restu Adinda^{1*}, Muhammad Iqbal Yusan¹, Ihad Rayyan Jhorrif¹,
Asepta Surya Wardhana¹

¹ Teknik Instrumentasi Kilang, Politeknik Energi dan Mineral Akamigas, Cepu, Blora, 58312

*E-mail: rinduadinda91@gmail.com

ABSTRAK

Sektor produksi gas sangat bergantung pada keandalan kompresor, kegagalan dapat mengakibatkan kerugian finansial dan operasional yang signifikan. Metode diagnosis kerusakan konvensional seringkali memiliki keterbatasan dalam menghadapi kompleksitas dan sifat non-linier sistem kompresor modern. Penelitian ini mengembangkan sistem diagnosis kerusakan kompresor produksi gas menggunakan Algoritma Genetika yang mengintegrasikan identifikasi jenis kerusakan, estimasi biaya perbaikan, dan durasi waktu pengerjaan. Sistem diimplementasikan dengan Python-PyQt5 dan memanfaatkan fungsi fitness multi-kriteria yang menggabungkan *coverage* (0,55), *precision* (0,15), *cost factor* (0,15), dan *simplicity* (0,15). Dengan parameter populasi 80 individu, laju *crossover* 0,8, dan mutasi 0,03, sistem mencapai konvergensi pada generasi ke-15 dengan *fitness* optimal 0,7000. Hasil diagnosis mengidentifikasi tiga kandidat kerusakan: *Pressure Transmitter Failure* (kecocokan 28,6%), *Flow Transmitter Failure* (16,7%), dan *Level Transmitter Failure* (0,0%). Total akumulasi estimasi biaya dari kedua kerusakan signifikan adalah Rp 41.700.000 dengan waktu perbaikan 5,9 jam. Nilai kecocokan 28,6% merupakan nilai tertinggi dibandingkan 14 kandidat lainnya dalam sistem *multi-fault* diagnosis dengan 80 parameter gejala, menunjukkan *confidence level* memadai untuk prioritas *maintenance*. Waktu komputasi rata-rata di bawah 5 detik memungkinkan implementasi real-time untuk pemeliharaan prediktif.

Kata Kunci: Diagnosis Kerusakan, Kompresor Produksi Gas, Algoritma Genetika, Pemeliharaan Prediktif, Optimasi

1. PENDAHULUAN

Kompresor adalah komponen krusial dalam sistem produksi gas, dan kerusakan pada unit, khususnya kompresor *reciprocating* dalam industri minyak dan gas lepas pantai yang beroperasi di lingkungan ekstrem, dapat menyebabkan kerugian finansial signifikan serta penurunan efisiensi operasional [1]. Oleh karena itu, diagnosis kerusakan yang cepat dan akurat sangat penting untuk meminimalkan *downtime* (waktu henti) dan menjaga keberlanjutan produksi [2]. Dalam konteks ini, Algoritma Genetika (AG) menawarkan solusi inovatif karena keunggulannya dalam mengoptimalkan pencarian parameter pada ruang solusi yang kompleks dan non-linier [1], terbukti efektif dalam klasifikasi kegagalan, optimasi parameter *Support Vector Machine* (SVM), dan seleksi fitur diagnostik seperti *envelope analysis* [3], memungkinkan penjadwalan pemeliharaan yang lebih optimal dan deteksi dini [4], serta meminimalkan fungsi biaya pemeliharaan preventif [5]. Kemampuan AG dalam menyeimbangkan multiple kriteria secara simultan menjadikannya tepat untuk keputusan pemeliharaan optimal [6].

Meskipun AG telah dieksplorasi untuk pemeliharaan prediktif dan deteksi kegagalan sistem [1],[2],[4], sebagian besar studi sebelumnya belum menyediakan estimasi komprehensif terhadap biaya perbaikan dan durasi waktu yang dibutuhkan secara terintegrasi dengan klasifikasi kerusakan [7]. Kesenjangan ini krusial mengingat karakteristik operasi lepas pantai: (1) biaya mobilisasi teknisi mencapai ratusan juta rupiah per mobilisasi, (2) keterbatasan

window maintenance bergantung kondisi cuaca, (3) kerugian produksi miliaran rupiah per hari akibat downtime tidak terencana, dan (4) keterbatasan spare part dan SDM di lokasi terpencil. Tanpa estimasi terintegrasi, operator tidak dapat membuat keputusan maintenance terencana, berpotensi pemborosan sumber daya atau kegagalan kritis dengan dampak operasional serius.

Penelitian ini bertujuan mengembangkan sistem diagnosis kerusakan kompresor berbasis AG yang mengidentifikasi jenis kerusakan dan memberikan estimasi biaya, durasi waktu, serta tingkat kecocokan diagnosis secara terintegrasi untuk memfasilitasi pengambilan keputusan pemeliharaan prediktif yang terencana dan ekonomis.

2. METODE PENELITIAN

A. Desain Penelitian

Penelitian ini menggunakan simulasi komputer dengan pendekatan kuantitatif untuk mengembangkan dan menguji AG sebagai metode prediksi kerusakan kompresor. Pemilihan AG karena kemampuannya menemukan solusi optimal dalam masalah kompleks dan non-linier [8]. Sumber Data: Basis pengetahuan dikompilasi dari: (1) data historis operasional nyata sistem kompresor di industri oil & gas, (2) manual operasional dan *maintenance log vendor kompresor reciprocating*, (3) *knowledge base* dari studi literatur jurnal teknik dan *handbook pemeliharaan*, dan (4) konsultasi dengan *expert engineer* instrumentasi. Sistem mengintegrasikan 80 jenis gejala mencakup anomali instrumentasi (*pressure, flow, level, temperature transmitter*), sistem kontrol (*valve, actuator, PID loop*), mekanik (*motor, pompa, bearing, seal*), elektrik (*power supply, grounding, kabel*), dan proses (*evaporator, steam system, filter, safety device*) yang telah divalidasi dengan data operasional aktual dan review praktisi *maintenance*.

B. Algoritma Genetika

Algoritma genetika AG merupakan metode heuristik berdasarkan prinsip evolusi biologis, di mana solusi direpresentasikan sebagai kromosom yang berevolusi melalui seleksi, *crossover*, dan mutasi untuk menyelesaikan masalah optimasi kompleks dan non-linier [8], karena dapat menjelajahi ruang solusi secara menyeluruh tanpa memerlukan turunan matematis. Pada penelitian ini, algoritma genetika diterapkan untuk mendeteksi dan mengidentifikasi jenis kerusakan pada sistem kompresor berdasarkan gejala yang diperoleh dari sistem instrumentasi dan sensor.

a. Inisialisasi Populasi Awal

Populasi awal dibentuk secara acak dan terdiri atas sejumlah kromosom, di mana setiap gen merepresentasikan satu jenis kerusakan (*fault*) dengan nilai biner (0 = tidak dipilih, 1 = dipilih) [9]. Sehingga satu kromosom dapat dituliskan pada Pers. (1):

$$X_i = [x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, \dots, x_{in}] \quad (1)$$

b. Evaluasi Nilai *Fitness*

Nilai *fitness* digunakan untuk mengukur seberapa baik suatu kromosom merepresentasikan solusi terhadap permasalahan yang dihadapi. Nilai *fitness* setiap kromosom dihitung berdasarkan tingkat kecocokan antara gejala input pengguna dan data gejala pada basis pengetahuan kerusakan. Evaluasi ini mempertimbangkan empat komponen utama, yaitu *coverage*, *precision*, *cost factor*, dan *simplicity*. Fungsi *fitness* yang digunakan dalam penelitian ini seperti pada Pers. (2) :

$$Fitness_i = 0.55 \times Coverage_i + 0.15 \times Precision_i + 0.15 \times Cost_i + 0.15 Simplicity_i \quad (2)$$

Pendekatan multi-kriteria fitness function ini sejalan dengan tren terkini dalam optimasi berbasis metaheuristik [10]. Bobot terbesar (0.55) dialokasikan pada *Coverage* pada Pers. (3) untuk memastikan prioritas tertinggi dalam diagnosis adalah cakupan maksimum gejala aktual oleh kromosom solusi, sementara *Precision* pada Pers. (4), *Cost* pada Pers. (5), dan *Simplicity* pada Pers. (6) berfungsi sebagai faktor penyeimbang.

Dengan:

$$Coverage_i = \frac{Jumlah\ gejala\ yang\ cocok}{Total\ gejala} \quad (3)$$

$$Precision_i = \frac{Jumlah\ kecocokan\ benar}{Total\ prediksi} \quad (4)$$

$$Cost_i = 1 - \frac{Biaya\ deteksi}{Biaya\ maksimum} \quad (5)$$

$$Simplicity_i = \frac{1}{1+nrules} \quad (6)$$

Kromosom dengan nilai *fitness* tertinggi menunjukkan tingkat kecocokan terbaik antara gejala input dan basis pengetahuan kerusakan.

c. Seleksi

Proses seleksi menggunakan metode Roulette Wheel Selection, dimana probabilitas setiap individu terpilih proporsional terhadap fitnessnya [11]. Probabilitas seleksi pada *roulette wheel selection* dapat dinyatakan seperti pada Pers. (7):

$$P_{sel}(i) = \frac{Fitness_i}{\sum_{j=1}^N Fitness_j} \quad (7)$$

d. Crossover

Crossover merupakan tahap penggabungan informasi genetik dari dua *parent* untuk menghasilkan individu baru, yaitu *offspring*. Proses ini berfungsi untuk menghasilkan kombinasi solusi baru yang berpotensi memiliki nilai *fitness* lebih tinggi dari induknya. Metode *crossover* yang digunakan pada penelitian ini adalah *one-cut point crossover*, yaitu penyilangan gen dengan satu titik potong acak sehingga sebagian gen dari kedua induk saling ditukar [12]. *Offspring* yang dihasilkan dari proses *one-cut point crossover* dapat diformulasikan seperti pada Pers. (8) dan (9):

$$Offspring_1 = [x_{p1}, 1, x_{p1}, 2, \dots, x_{p1}, k, x_{p2}, k+1, \dots, x_{p2}, n] \quad (8)$$

$$Offspring_2 = [x_{p2}, 1, x_{p2}, 2, \dots, x_{p2}, k, x_{p1}, k+1, \dots, x_{p1}, n] \quad (9)$$

One-cut point *crossover* memberikan keseimbangan optimal antara eksplorasi ruang solusi dan konvergensi [13].

e. Mutasi

Mutasi merupakan tahapan yang bertujuan untuk menambah variasi genetik dalam populasi dengan cara mengubah nilai gen pada kromosom secara acak. Metode yang digunakan yaitu *bit-flip mutation*, di mana gen pada posisi tertentu diubah dari 0 menjadi 1 atau sebaliknya berdasarkan probabilitas mutasi (*mutation rate*). Tahap mutasi ini membantu algoritma agar

tidak terjebak pada solusi lokal dan memperluas pencarian terhadap solusi optimal. Proses *bit-flip mutation* dapat dinyatakan seperti pada Pers. (10), di mana P_m adalah *mutation rate*, biasanya bernilai kecil (misalnya 0.01–0.1):

$$x'_{ij} = \begin{cases} 1 - x_{ij}, & \text{dengan probabilitas } P_m \\ x_{ij}, & \text{dengan probabilitas } 1 - P_m \end{cases} \quad (10)$$

Penggunaan *bit-flip mutation* dengan probabilitas rendah terbukti efektif dalam mencegah konvergensi prematur pada masalah optimasi biner [14].

f. Kriteria Terminasi

Kriteria terminasi digunakan untuk menentukan kapan proses evolusi pada algoritma genetika harus dihentikan. Pada dasarnya, algoritma genetika bersifat iteratif dan akan terus menghasilkan generasi baru hingga kondisi tertentu terpenuhi. Proses evolusi akan berhenti apabila jumlah generasi telah mencapai batas maksimum yang telah ditentukan sebelumnya seperti pada Pers. (11):

$$(t \geq t_{max}) \quad (11)$$

Penentuan nilai t_{max} dilakukan berdasarkan kebutuhan tingkat ketelitian dan waktu komputasi yang tersedia. Selain itu, proses juga dapat dihentikan ketika nilai *fitness* terbaik antar generasi sudah tidak menunjukkan peningkatan yang signifikan. Kondisi ini disebut konvergensi, di mana perubahan nilai *fitness* antara dua generasi berturut-turut sudah lebih kecil dari ambang batas toleransi ϵ , sebagaimana dinyatakan pada Pers. (12):

$$|Fitness_{best}(t) - Fitness_{best}(t-1)| < \epsilon \quad (12)$$

Kriteria lain yang menjadi dasar penghentian adalah ketika populasi mengalami stagnasi, yaitu tidak adanya perubahan nilai *fitness* terbaik dalam beberapa generasi berturut-turut. Hal ini menandakan bahwa seluruh individu dalam populasi telah memiliki tingkat keseragaman genetik yang tinggi dan algoritma kemungkinan besar telah mencapai titik optimum lokal. Selain itu, algoritma genetika juga dapat dihentikan lebih awal apabila nilai *fitness* terbaik yang diperoleh telah mencapai atau melampaui nilai target tertentu seperti pada Pers. (13):

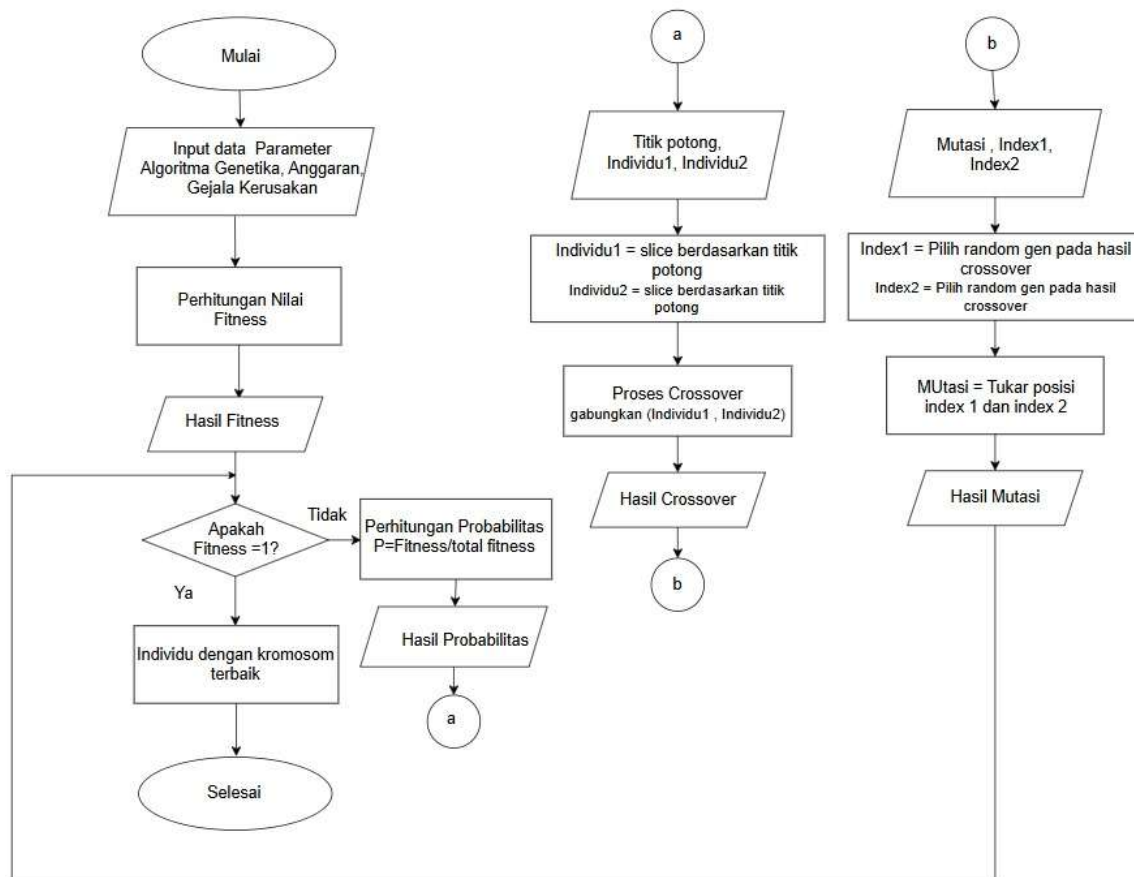
$$(Fitness_{best}(t) \geq Fitness_{target}) \quad (13)$$

Pers. (13) menunjukkan bahwa solusi yang dihasilkan sudah memenuhi kriteria optimalitas yang diinginkan. Secara umum, kondisi terminasi pada algoritma genetika dapat dirumuskan seperti pada Pers. (14):

$$\text{Hentikan jika } \begin{cases} t \geq t_{max}, \\ |Fitness_{best}(t) - Fitness_{best}(t-1)| < \epsilon, \\ (Fitness_{best}(t) \geq Fitness_{target}). \end{cases} \quad (14)$$

Dengan penerapan kriteria terminasi tersebut, algoritma genetika dapat berhenti secara efisien setelah menemukan solusi yang stabil atau mendekati optimal, sehingga waktu komputasi dapat diminimalkan tanpa mengorbankan kualitas hasil akhir. Kombinasi multiple stopping criteria ini mengikuti praktik terbaik dalam implementasi algoritma evolusioner modern [15].

C. Diagram Alir Metode



Gambar 1. Diagram Metode Algoritma Genetika

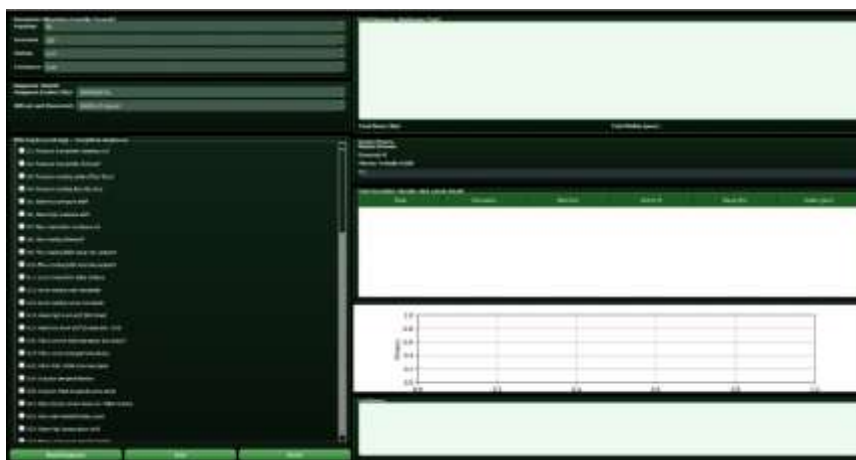
Gambar 1 menunjukkan diagram alir metode algoritma genetika. Proses diawali dengan *input* data parameter AG, meliputi data anggaran, gejala kerusakan, dan parameter utama. Selanjutnya, dilakukan perhitungan nilai *fitness* untuk mengevaluasi individu. Berdasarkan nilai *fitness* dan probabilitasnya, dilakukan seleksi menggunakan metode roulette wheel untuk menentukan *parent*. Tahap berikutnya adalah proses *crossover* dengan menentukan titik potong gen dan menghasilkan individu baru, dilanjutkan dengan proses *mutasi* untuk menukar posisi gen pada indeks acak. Individu hasil mutasi akan dievaluasi kembali hingga diperoleh kromosom dengan nilai *fitness* terbaik sebagai solusi optimal.

3. PEMBAHASAN

A. Implementasi Sistem

Sub bab ini menyajikan deskripsi implementasi sistem diagnosis kerusakan kompresor produksi gas menggunakan Algoritma Genetika (AG) yang telah dirancang. Implementasi dilakukan menggunakan bahasa pemrograman Python dengan antarmuka grafis (GUI) yang dikembangkan berbasis pustaka PyQt5. Sistem ini berfungsi sebagai *front-end* yang memungkinkan pengguna memasukkan parameter Algoritma Genetika, memilih gejala kerusakan yang teramati pada kompresor (*symptom checklist*), serta menampilkan hasil diagnosis yang terintegrasi, termasuk *fitness* optimal, peringkat jenis kerusakan, dan estimasi biaya. Sistem dimple-

mentasikan menggunakan Python dengan GUI PyQt5 yang memungkinkan input parameter AG, pemilihan dari 80 gejala kerusakan, dan menampilkan hasil diagnosis terintegrasi ditunjukkan pada gambar 2.



Gambar 2. Tampilan Awal Sistem

Gambar 2. menampilkan antarmuka utama program diagnosis kerusakan kompresor berbasis Algoritma Genetika (GA) yang dikembangkan menggunakan bahasa pemrograman Python dengan pustaka *PyQt5*. Antarmuka ini terdiri atas tiga area utama, yaitu input parameter algoritma, pemilihan gejala kerusakan, dan hasil diagnosis. Pada sisi kiri, pengguna dapat memasukkan parameter GA seperti ukuran populasi, probabilitas *crossover* dan *mutation*, serta jumlah generasi maksimum. Di bawahnya terdapat daftar gejala kerusakan (*symptom checklist*) yang mencakup indikator umum pada kompresor, seperti tekanan discharge abnormal, kenaikan suhu bearing, dan getaran berlebih. Bagian kanan antarmuka menampilkan hasil proses optimasi, meliputi nilai *fitness* total, tabel hasil diagnosis dengan tingkat kecocokan tiap individu, serta grafik evolusi *fitness* terhadap jumlah generasi. Tampilan ini memberikan visualisasi performa algoritma dalam mencapai solusi optimal terhadap jenis kerusakan yang paling mungkin terjadi. Desain antarmuka menggunakan tata letak terstruktur dan kontras warna hijau gelap untuk meningkatkan keterbacaan serta efisiensi interaksi pengguna dalam proses diagnosis.

B.Implementasi Metode Hasil Percobaan

Implementasi metode diagnosis kerusakan pada sistem kompresor dilakukan dengan mengintegrasikan algoritma genetika (GA) ke dalam antarmuka sistem berbasis *PyQt5*. Tujuannya adalah untuk mencari kombinasi kerusakan (*fault*) yang paling sesuai dengan gejala input berdasarkan nilai *fitness* tertinggi. Metode ini bekerja dengan parameter yang diatur melalui GUI, antara lain jumlah populasi (80 individu), jumlah generasi (100), probabilitas mutasi (0,03), dan probabilitas *crossover* (0,80). Setiap individu dalam populasi direpresentasikan sebagai *array biner* yang menunjukkan kondisi suatu komponen 1 menandakan komponen mengalami kerusakan, dan 0 menandakan kondisi normal.

a. Mekanisme Perhitungan Fitness

Nilai *fitness* dihitung dalam program dengan mempertimbangkan kesesuaian antara gejala yang dipilih pengguna dan *database* gejala tiap jenis kerusakan. Perhitungan ini menggunakan fungsi *fitness* multi-kriteria sebagaimana didefinisikan pada Pers. (2) di Bab 2. Kriteria yang

dipertimbangkan meliputi *coverage*, *precision*, *cost factor*, dan *simplicity*. Nilai *fitness* tertinggi menandakan kombinasi kerusakan yang paling sesuai dengan pola gejala aktual sekaligus memberikan pertimbangan biaya dan kompleksitas yang optimal.

b. Estimasi Waktu dan Biaya Perbaikan

Estimasi waktu dan biaya merupakan aspek penting dalam evaluasi performa sistem, terutama untuk menentukan efisiensi perawatan dan efektivitas proses identifikasi kerusakan pada sistem kompresor. Analisis ini dilakukan dengan mempertimbangkan waktu operasional sistem, frekuensi kerusakan, serta biaya perbaikan yang dikeluarkan. Menghitung estimasi waktu dan biaya untuk tiap kandidat kerusakan berdasarkan data parameter komponen yang telah ditentukan dalam sistem. Nilai waktu dan biaya dihitung ulang pada setiap generasi, dan hasil akhirnya diambil dari individu dengan *fitness* tertinggi pada generasi terakhir. Perhitungan dilakukan menggunakan pendekatan proporsional terhadap tingkat kecocokan (*match percentage*) seperti pada Pers. (15) dan (16):

$$W_i = W_{nom} \times Match\% \quad (15)$$

$$C_i = C_{nom} \times Match\% \quad (16)$$

Nilai *Match Percentage* ini diperoleh dari komponen *Coverage* Pers. (3) yang merupakan bagian dari perhitungan *fitness* multi-kriteria Algoritma Genetika.

Total estimasi dihitung seperti pada Pers. (17) dan (18):

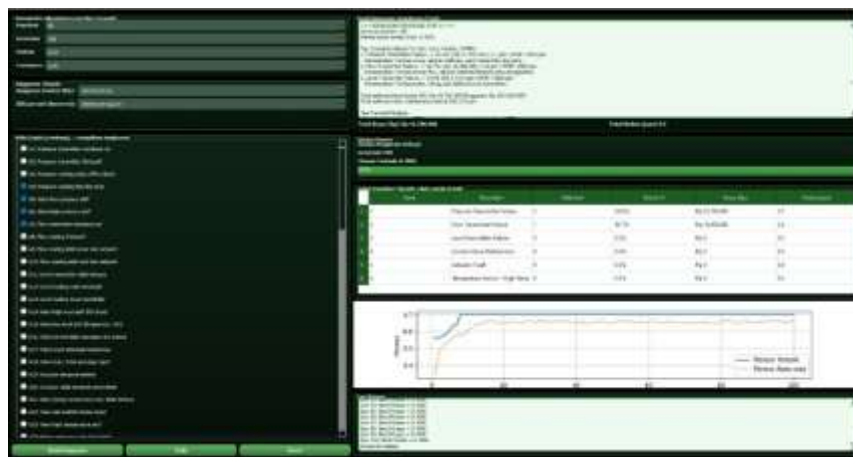
$$W_{Total} = \sum_{i=1}^n W_i \quad (17)$$

$$C_{Total} = \sum_{i=1}^n C_i \quad (18)$$

Nilai waktu dan biaya dihitung pada setiap generasi untuk individu terbaik, dan hasil akhir diambil dari individu dengan *fitness* tertinggi pada generasi terakhir. Pendekatan proporsional ini memastikan bahwa estimasi biaya dan waktu mencerminkan tingkat keyakinan sistem terhadap diagnosis kerusakan tersebut.

C. Hasil Eksperimen Algoritma Genetika

Proses diagnosis dilakukan dengan parameter algoritma genetika yang terdiri atas populasi 80 individu, 100 generasi, laju mutasi sebesar 0,03, dan laju *crossover* sebesar 0,8. Nilai-nilai ini dipilih berdasarkan hasil pengujian awal untuk mencapai keseimbangan antara eksplorasi dan konvergensi solusi. Pada Gambar 3 batas anggaran sistem ditetapkan sebesar Rp 100.000.000, dengan konversi unit Rp 100.000/unit sebagai parameter biaya nominal per jenis kerusakan. Setelah dilakukan 100 iterasi, sistem menghasilkan nilai *fitness* terbaik sebesar 0,7000, sebagaimana ditunjukkan pada grafik konvergensi di Gambar 3. Nilai ini menunjukkan tingkat kesesuaian gejala yang diberikan dengan data pada basis pengetahuan kerusakan kompresor.



Gambar 3. Hasil Uji Coba Algoritma Genetika

D.Hasil Diagnosa dan Estimasi Parameter

Proses diagnosis sistem kompresor dilakukan dengan menggunakan algoritma genetika (AG) yang telah dikonfigurasi dengan parameter populasi sebanyak 80 individu, 100 generasi, laju mutasi sebesar 0,03, dan *crossover* sebesar 0,8. Hasil keluaran sistem ditampilkan dalam bentuk peringkat jenis kerusakan (*fault ranking*) beserta nilai kecocokan (*match percentage*), estimasi biaya perbaikan, dan waktu pengerjaan untuk setiap jenis kerusakan yang teridentifikasi. Dari hasil eksperimen dengan input gejala aktual dari sistem kompresor, diperoleh hasil diagnosis sebagaimana ditunjukkan pada Tabel 1. Sistem mengevaluasi 15 kandidat jenis kerusakan yang tersimpan dalam basis pengetahuan dan menghasilkan tiga kandidat dengan nilai kecocokan tertinggi.

Tabel 1. Hasil Uji Coba Algoritma Genetika

Rank	Jenis Kerusakan	Match(%)	Biaya (Rp)	Waktu (Jam)
1	Pressure Trnsmitter Failure	28,6%	25.700.000	3.1
2	Flow Transmitter Failure	16,7%	16.000.000	2.8
3	Level Transmitter Failure	0.0%	0	0.0

Pada tahap diagnosis, sistem menghitung nilai kecocokan antara gejala input dengan basis pengetahuan menggunakan fungsi *fitness* multi-kriteria dari algoritma genetika. Semakin tinggi nilai kecocokan, semakin besar kemungkinan jenis kerusakan tersebut sesuai dengan kondisi sistem aktual. *Pressure Transmitter Failure* memperoleh nilai kecocokan tertinggi sebesar 28,6%, yang berarti 28,6% dari seluruh gejala yang diinput cocok dengan pola kerusakan jenis ini. Nilai kecocokan 28,6% ini merupakan hasil yang signifikan dalam konteks sistem *multi-fault diagnosis*.

Flow Transmitter Failure memperoleh kecocokan 16,7%, menunjukkan bahwa kerusakan ini juga memiliki kemiripan pola dengan gejala input, namun dengan tingkat keyakinan yang lebih rendah. *Level Transmitter Failure* memperoleh kecocokan 0,0%, mengindikasikan bahwa gejala input tidak menunjukkan karakteristik kerusakan jenis ini [16] berikut:

Pada hasil diagnosis:

Pressure transmitter failure → Rp 25.700.000

Flow transmitter failure → Rp 16.000.000

Sehingga total estimasi biaya dan estimasi waktu:

$$C_{total} = 25.700.000 + 16.000.000 = \text{Rp } 41.700.000$$

$$W_{total} = 3.1 + 2.8 = 5.9 \text{ Jam}$$

Total akumulasi biaya Rp 41.700.000 dan waktu 5,9 jam merupakan penjumlahan dari dua kerusakan teridentifikasi (*Pressure dan Flow Transmitter Failure*), bukan satu kerusakan saja. Sistem mencapai *fitness* optimal 0,7000 pada generasi ke-15, menunjukkan keseimbangan optimal antara empat kriteria evaluasi. Nilai ini memadai untuk pengambilan keputusan lapangan karena: (1) konvergensi stabil menunjukkan solusi optimal tercapai, (2) *trade-off optimal* antara akurasi dan pertimbangan ekonomis, dan (3) identifikasi kandidat dengan prioritas jelas.

Kecocokan 28,6% signifikan dalam konteks sistem *multi-fault* dengan 80 gejala karena: (1) kompleksitas ruang solusi sangat luas, (2) gejala dapat *overlap* antar jenis kerusakan (*non-exclusive symptoms*), (3) *multi-domain coverage* (instrumentasi, mekanik, elektrik, proses), dan (4) nilai komparatif 71% lebih tinggi dibanding kandidat kedua (16,7%), menunjukkan *confidence level* kuat untuk prioritas diagnosis.

E. Analisis Konvergensi Nilai *Fitness*

Implementasi algoritma genetika dilakukan menggunakan bahasa Python dengan antarmuka GUI berbasis *PyQt5*. Parameter utama yang digunakan meliputi jumlah populasi sebanyak 80 kromosom, probabilitas *crossover* 0,8, probabilitas mutasi 0,03, dan jumlah generasi 100. Tujuan utama algoritma adalah memaksimalkan fungsi *fitness* multi-kriteria untuk menentukan kombinasi gejala yang paling sesuai dengan jenis kerusakan pada sistem kompresor. Nilai *fitness* dihitung berdasarkan fungsi multi-kriteria seperti yang ditunjukkan pada Pers. (2).

Nilai *fitness* multi-kriteria ini dihitung dengan mempertimbangkan empat komponen utama, yaitu *coverage*, *precision*, *cost factor*, dan *simplicity*, dengan pembobotan sesuai pada Pers. (2). Nilai *fitness* yang semakin mendekati 1 menandakan tingkat kesesuaian yang semakin tinggi antara data gejala dan pola kerusakan aktual sekaligus telah melewati tahap optimasi biaya dan kompleksitas. Proses seleksi dilakukan menggunakan metode *roulette wheel selection* untuk memberikan peluang lebih besar pada individu dengan *fitness* tinggi. Kombinasi gen dilakukan menggunakan *single-point crossover* untuk membentuk keturunan baru yang membawa sebagian karakteristik induk terbaik. Mutasi diterapkan secara acak pada gen tertentu dengan peluang 3% guna mempertahankan keragaman populasi dan mencegah konvergensi prematur. Hasil simulasi menunjukkan peningkatan nilai *fitness* hingga stabil di 0,7000 setelah generasi ke-15 (Gambar 4), mengindikasikan efisiensi algoritma dalam menemukan kombinasi diagnosis optimal tanpa iterasi berlebihan.



Gambar 4. Grafik Konvergensi Nilai *Fitness* terhadap Jumlah Generasi

Gambar 4. Merupakan interpretasi hasil diagnosis menunjukkan bahwa kromosom dengan *fitness* tertinggi mewakili kombinasi gejala yang paling sesuai dengan pola kerusakan tertentu pada sistem kompresor, seperti penurunan tekanan *discharge* akibat kebocoran katup atau kerusakan piston *ring*. Hal ini mengindikasikan bahwa algoritma genetika mampu mengenali pola hubungan nonlinier antar gejala dan mengonvergensi menuju hasil diagnosis yang akurat berdasarkan data masukan aktual.

4. SIMPULAN

Implementasi algoritma genetika untuk diagnosis kerusakan pada sistem kompresor produksi gas menunjukkan penelitian berhasil mengembangkan sistem diagnosis kerusakan kompresor menggunakan AG yang mengidentifikasi jenis kerusakan, mengestimasi biaya, durasi waktu, dan tingkat kecocokan secara terintegrasi sesuai tujuan penelitian. Implementasi AG dengan parameter populasi 80, laju *crossover* 0,8, dan *mutation rate* 0,03 mencapai *fitness* optimal 0,7000 pada generasi ke-15. Fungsi *fitness* multi kriteria berhasil menyeimbangkan akurasi dengan pertimbangan ekonomis. Operator genetika *roulette wheel selection*, *single-point crossover*, dan *bit-flip mutation* efektif mempertahankan keragaman populasi.

Sistem mengintegrasikan 80 jenis gejala komprehensif. Hasil diagnosis mengidentifikasi *Pressure Transmitter Failure* (28,6%) dan *Flow Transmitter Failure* (16,7%), dengan total akumulasi estimasi biaya Rp 41.700.000 dan waktu 5,9 jam dari kedua kerusakan. Waktu komputasi < 5 detik memungkinkan implementasi real-time untuk pemeliharaan prediktif terencana. Integrasi estimasi biaya dan waktu memberikan informasi krusial untuk perencanaan maintenance pada operasi lepas pantai dengan biaya mobilisasi tinggi dan *window maintenance* terbatas.

5. DAFTAR PUSTAKA

- [1] K. Salim, R. S. A. Hebri, and S. Bisma, "Classification Predictive Maintenance Using XGboost with Genetic Algorithm," *Rev. d'Intelligence Artif.*, vol. 36, no. 6, pp. 833–845, 2022, doi: 10.18280/ria.360603.
- [2] A. Farhana, S. Sabeer, A. Siddiqua, and A. Anjum, "Enhancing Production System Performance: Failure Detection and Availability Improvement with Deep Learning and Genetic Algorithm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 12, pp. 716–726, 2023, doi: 10.14569/IJACSA.2023.0141274.
- [3] M. Ahmed, F. Gu, and A. Ball, "Feature selection and fault classification of reciprocating compressors using a genetic algorithm and a probabilistic neural network," *J. Phys. Conf. Ser.*, vol. 305, no. 1, 2011, doi: 10.1088/1742-6596/305/1/012112.
- [4] D. S. Donoriyanto, A. S. Anam, and E. W. Pudji, "Application of genetic algorithm method on machine maintenance," *J. Phys. Conf. Ser.*, vol. 953, no. 1, 2018, doi: 10.1088/1742-6596/953/1/012225.
- [5] I. Maatouk, I. Jarkass, E. Châtelet, and N. Chebbo, "Preventive maintenance optimization and comparison of genetic algorithm models in a series-parallel multi-state system," *J. Intell. Syst.*, vol. 28, no. 2, pp. 219–230, 2019, doi: 10.1515/jisys-2017-0096.
- [6] S. Erpolat Tasabat and O. Aydin, "Using Long-Short Term Memory Networks with Genetic Algorithm to Predict Engine Condition," *Gazi Univ. J. Sci.*, vol. 35, no. 3, pp. 1200–1210, 2022, doi: 10.35378/gujs.937169.
- [7] K. Zhang *et al.*, "Compressor fault diagnosis system based on PCA-PSO-LSSVM algorithm," *Sci. Prog.*, vol. 104, no. 3, pp. 1–16, 2021, doi: 10.1177/00368504211026110.
- [8] X. Fan, W. Sayers, S. Zhang, Z. Han, L. Ren, and H. Chizari, "Review and Classification of Bio-inspired Algorithms and Their Applications," *J. Bionic Eng.*, vol. 17, no. 3, pp. 611–631, 2020, doi: 10.1007/s42235-020-0049-9.

- [9] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimed. Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, 2021, doi: 10.1007/s11042-020-10139-6.
- [10] K. Rajwar, K. Deep, and S. Das, "An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges," *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 13187–13257, 2023, doi: 10.1007/s10462-023-10470-y.
- [11] A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri, and V. B. S. Prasath, "Choosing mutation and crossover ratios for genetic algorithms-a review with a new dynamic approach," *Inf.*, vol. 10, no. 12, 2019, doi: 10.3390/info10120390.
- [12] D. Wahyuningsih and E. Helmud, "Penerapan Algoritma Genetika Untuk Optimasi Penjadwalan pada MTS Negeri 1 Pangkalpinang," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 3, pp. 435–441, 2020, doi: 10.32736/sisfokom.v9i3.994.
- [13] H. Armanto, H. A. Rosyid, M. Muladi, and G. Gunawan, "Evolutionary Algorithm in Game – A Systematic Review," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 2, pp. 4–11, 2023, doi: 10.22219/kinetik.v8i2.1714.
- [14] Š. Hubálovský, M. Hubálovská, and I. Matoušová, "A New Hybrid Particle Swarm Optimization–Teaching–Learning–Based Optimization for Solving Optimization Problems," vol. 9, no. 1. 2024. doi: 10.3390/biomimetics9010008.
- [15] A. Slowik and H. Kwasnicka, "Evolutionary algorithms and their applications to engineering problems," *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12363–12379, 2020, doi: 10.1007/s00521-020-04832-8.
- [16] S. P. Collins *et al.*, "Penerapan Artificial Intelligence (AI) dalam Teknik Industri menuju Industrial 5.0." 2021.