

OPTIMALISASI *TRADE-OFF STORAGE* DAN LATENSI DALAM MANAJEMEN DATA INDUSTRI KILANG BERBASIS *GENETIC ALGORITHM*

Citra N. Dg. Tapala¹, Zahida Nabila Ramadhani¹, Lutfi Arya Pramanda¹, Andika Setya Nugraha¹, Asepata Surya Wardhana¹

¹Teknik Instrumentasi Kilang, Politeknik Energi dan Mineral Akamigas, Cepu, Blora, 58311

*E-mail: zahidanbl03@gmail.com

ABSTRAK

Pengelolaan data instrumen kilang menghadapi tantangan *trade-off* yang krusial antara efisiensi penyimpanan (*storage*) dan kecepatan akses (latensi/*Service Level Agreement*/SLA). Peningkatan granularitas data (*sampling_rate_hz*) dan ketersediaan (*replication_factor*) seringkali mengorbankan kapasitas penyimpanan yang terbatas. Untuk mengatasi konflik tersebut, penelitian ini menerapkan *Algoritma Genetika* (GA) sebagai pendekatan optimasi multikriteria. Kromosom GA merepresentasikan empat parameter utama, yaitu *sampling_rate_hz*, *compression_level*, *retention_days*, dan *replication_factor*. Proses evolusi menggunakan kombinasi operator seleksi (*Tournament* atau *Roulette*), *crossover* (*One-Point* atau *Two-Point*), mutasi acak, dan elitisme. Evaluasi dilakukan dengan fungsi fitness berbobot yang bertujuan meminimalkan biaya gabungan (kebutuhan *storage* dan *latency*) sambil memenuhi kendala granularitas data dan *availability* sistem. Hasil penelitian menunjukkan GA mampu menemukan serangkaian solusi Pareto optimal yang secara efektif menyeimbangkan kualitas data dan efisiensi sistem. Konvergensi solusi yang teramati pada setiap generasi menghasilkan estimasi penggunaan *storage* tahunan yang terkendali (misalnya, berkurang dari *baseline*) dengan *latency* yang tetap memenuhi standar SLA (misalnya, di bawah 50 ms). Berdasarkan temuan ini, *Algoritma Genetika* terbukti efektif sebagai metode optimasi *trade-off* dalam sistem manajemen data instrumen kilang, dan berpotensi berfungsi sebagai alat bantu pengambilan keputusan strategis pada pengelolaan data skala industri.

Kata kunci: algoritma, *availability*, granulasi, instrumen, optimasi

1. PENDAHULUAN

Pengelolaan data di sektor industri, khususnya kilang, kini menghadapi tingkat kerumitan yang belum pernah terjadi sebelumnya seiring dengan masuknya teknologi informasi dan komunikasi [1]. Kompleksitas ini tidak hanya bersumber dari besarnya volume data, tetapi juga dari keragaman asal data, kecepatan arus informasi, dan kebutuhan untuk memprosesnya secara *real-time* agar tetap relevan bagi pengambilan keputusan. Kondisi ini semakin diperumit oleh ledakan data yang dihasilkan melalui sistem *Internet of Things* (IoT) dan *Big Data*, yang berperan penting dalam pemantauan langsung serta optimalisasi kinerja operasional [2], [3].

Penerapan kedua teknologi tersebut memungkinkan terjadinya akuisisi data dalam jumlah masif dari berbagai titik operasional yang sebelumnya sulit dijangkau, mulai dari sensor di lapangan hingga peralatan produksi dan jaringan distribusi energi. Contohnya terlihat pada PT Pertamina yang berhasil mengintegrasikan sistem di lebih dari 5518 SPBU di Indonesia. Integrasi ini tidak hanya mendukung efisiensi logistik, tetapi juga digunakan untuk memperkuat karakterisasi reservoir, simulasi operasi, serta meningkatkan aspek keselamatan kerja di industri migas [2].

Transformasi digital tersebut memunculkan kebutuhan akan sistem manajemen data yang mampu menyeimbangkan aspek kapasitas penyimpanan, kecepatan pemrosesan, dan latensi

akses secara bersamaan [2], [4]. Tantangan ini menjadi semakin penting karena operasional kilang sangat bergantung pada data *real-time*. Informasi yang terus menerus dikumpulkan menjadi fondasi strategi optimalisasi efisiensi maupun pemeliharaan, serta memberi peluang bagi manajemen untuk segera melakukan tindakan korektif ketika terjadi kondisi abnormal [5].

Penggunaan data secara optimal berkontribusi besar terhadap peningkatan keandalan operasi. Misalnya, prediksi kerusakan peralatan dapat dilakukan dengan lebih akurat sehingga mengurangi *downtime* yang tidak terencana dan sekaligus menekan biaya pemeliharaan [7]. Digitalisasi berbasis IoT dengan pemantauan getaran mesin secara daring terbukti meningkatkan efektivitas pemeliharaan prediktif. Strategi ini memungkinkan deteksi dini kerusakan potensial dan mencegah berhentinya mesin secara tiba-tiba [7].

Lebih dari itu, data hasil digitalisasi memberi ruang bagi penerapan analitik prediktif dan pembelajaran mesin untuk mengidentifikasi pola tersembunyi, mengekstrak wawasan strategis, dan mendukung peningkatan berkelanjutan pada proses produksi. Dengan cara ini, pengambilan keputusan bisnis dapat dilakukan dengan lebih presisi, berbasis data, dan adaptif terhadap dinamika operasional kilang [6], [7].

Meski begitu, perkembangan tersebut tidak serta-merta menyelesaikan persoalan utama, yakni *trade-off* antara kebutuhan penyimpanan dan latensi akses. Pendekatan tradisional terbukti sering kali tidak cukup efektif karena sifat masalah ini multi-objektif dan memiliki ruang solusi yang luas serta kompleks. Hal ini menuntut adanya metode optimasi yang lebih luwes dan mampu beradaptasi. Urgensi penelitian ini terletak pada perlunya solusi yang mampu mengoptimalkan keseimbangan antara kapasitas penyimpanan dan latensi akses di industri kilang. Meskipun beberapa penelitian sebelumnya telah mengkaji optimasi dalam manajemen data, belum ada yang secara khusus mengintegrasikan aspek tersebut menggunakan metode yang dapat mengatasi masalah multi-objektif yang kompleks pada sistem manajemen data instrumen kilang.

Genetic Algorithm (GA) sebagai salah satu metode metaheuristik berbasis evolusi menjadi alternatif yang menjanjikan. GA dikenal mampu menangani persoalan optimasi yang non-linear, rumit, dan multi-objektif [10], [18]. Penerapannya di berbagai bidang industri telah menunjukkan efektivitas dalam menemukan solusi mendekati optimal dengan efisiensi komputasi yang tinggi. Oleh karena itu, tujuan utama penelitian ini adalah merancang dan mengimplementasikan model GA untuk mengoptimalkan *trade-off* antara kebutuhan penyimpanan (diwakili oleh *compression level* dan *retention days*) dan persyaratan latensi/ketersediaan (diwakili oleh *sampling rate* dan *replication factor*) dalam sistem manajemen data instrumen kilang. Dengan memanfaatkan GA, penelitian ini berupaya mencapai keseimbangan ideal antara efisiensi pemanfaatan sumber daya dan performa sistem secara keseluruhan.

2. METODE

Penelitian ini dirancang secara rasional, dengan landasan teori optimasi multi-objektif yang sesuai untuk menjawab persoalan *trade-off* antara kapasitas penyimpanan dan latensi [9]. Secara empiris, penelitian ini diuji melalui eksperimen komputasi berbasis parameter yang terukur. Sementara itu, aspek sistematis tercermin dari tahapan penelitian yang disusun berurutan, dimulai dari perumusan masalah, penerapan algoritma, hingga analisis terhadap hasil yang diperoleh. [10], [11]

Eksperimen dalam penelitian ini dilakukan dengan mensimulasikan penerapan GA pada sistem manajemen data industri kilang [12]. Pemilihan GA didasarkan pada keunggulannya dalam menyelesaikan persoalan optimasi yang bersifat kompleks, non-linear, dan multi-objektif [13]. Fokus penelitian meliputi pengujian parameter algoritma, penyusunan fungsi fitness, serta

evaluasi hasil optimasi berdasarkan indikator utama, yaitu kebutuhan penyimpanan, latensi akses, dan tingkat ketersediaan data [14].

A. Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimen komputasi dengan tujuan mengevaluasi efektivitas penerapan GA dalam mengoptimalkan *trade-off* antara kapasitas penyimpanan dan latensi pada manajemen data kilang [15]. Desain penelitian ini bersifat rasional, karena berangkat dari kerangka teori optimasi multi-objektif yang sesuai dengan karakteristik masalah; empiris, karena melibatkan simulasi langsung dengan data dan parameter terukur; serta sistematis, karena tahapan percobaan dilakukan secara runtut mulai dari formulasi masalah, implementasi algoritma, hingga analisis hasil [16].

B. Variabel Penelitian

Adapun variabel keputusan yang dioptimalkan oleh GA membentuk kromosom individu, di mana setiap variabel direpresentasikan sebagai gen. Variabel-variabel tersebut meliputi *Sampling Rate* (Hz) yaitu frekuensi pengambilan sampel data dari sensor, *Compression Level* yaitu tingkat kompresi data yang digunakan, *Retention Days* yaitu durasi data disimpan dalam sistem, dan *Replication Factor* yaitu jumlah replika data untuk meningkatkan ketersediaan sistem [17]. Keempat variabel ini mewakili aspek yang berlawanan *Compression Level* dan *Retention Days* mengendalikan kebutuhan *storage*, sedangkan *Sampling Rate* dan *Replication Factor* mempengaruhi latensi dan ketersediaan sistem.

C. Peralatan dan Lingkungan Eksperimen

Eksperimen dilaksanakan menggunakan perangkat lunak Python 3.10+ dengan Thonny sebagai lingkungan pengembangan terintegrasi (IDE) karena kesederhanaannya serta kemampuannya menjalankan kode Python secara interaktif [18]. *Library* pendukung yang digunakan meliputi PyQt5 (untuk antarmuka grafis), Matplotlib (untuk visualisasi grafik dan pembuatan *flowchart*), serta *dataclasses* dan *typing* (untuk struktur data dan anotasi tipe) [19]. Selain itu, *library random*, *math*, dan *csv* dimanfaatkan untuk mendukung proses inisialisasi populasi, perhitungan fitness, serta penyimpanan riwayat *fitness* dalam format file.

Perangkat keras yang digunakan adalah komputer dengan spesifikasi minimal prosesor Intel Core i5, RAM 8 GB, dan sistem operasi Windows 10/11. Konfigurasi ini sudah memadai untuk menjalankan simulasi GA dengan visualisasi *real-time* tanpa hambatan berarti [20].

D. Prosedur Penelitian

Tahapan eksperimen dilakukan secara sistematis sesuai Gambar 1, sebagai berikut:

- Formulasi Fungsi *Fitness*

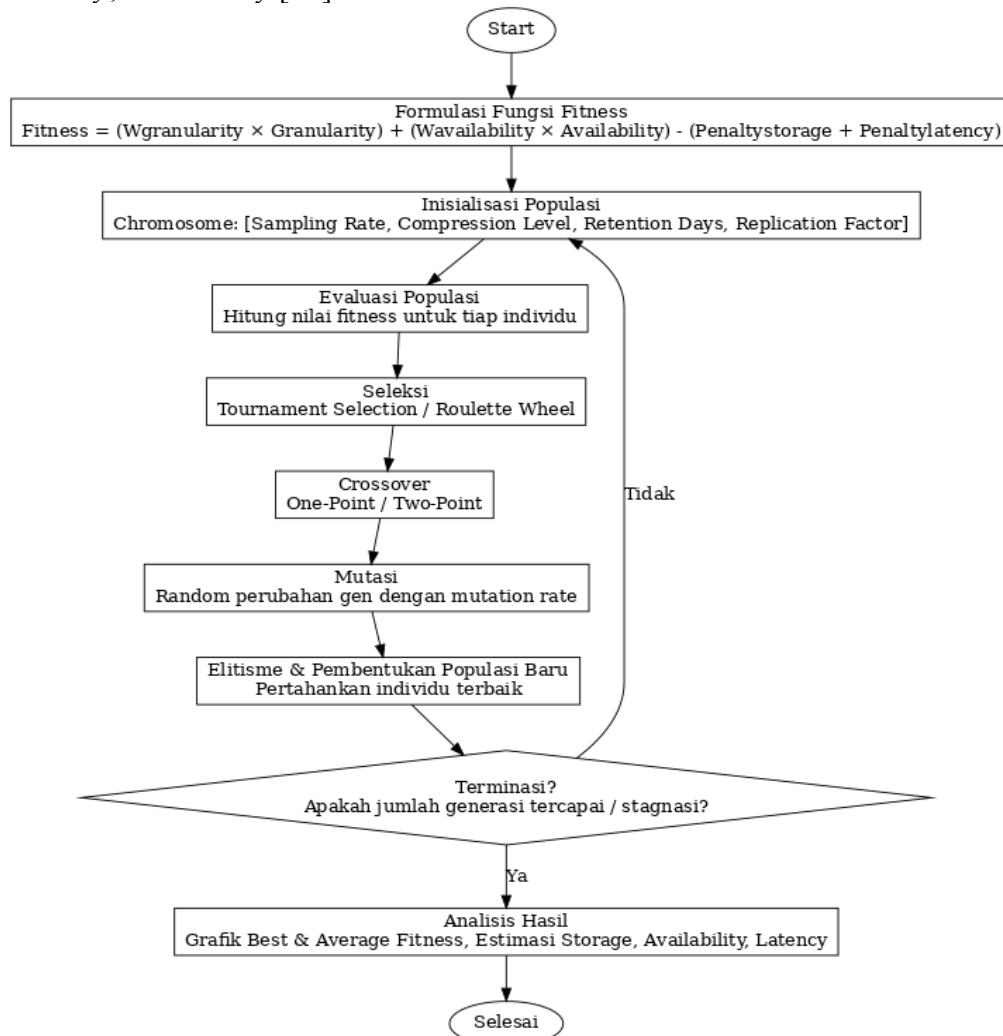
Fungsi fitness (F) dirancang sebagai fungsi Kualitas Solusi berbobot tunggal untuk memaksimalkan performa solusi. Fungsi ini mencerminkan keseimbangan antara Utilitas (diberi bobot positif untuk *granularitas* dan *availability*) dengan Penalti/Biaya (diberi bobot negatif untuk *storage* dan *latency*) [9]. Fungsi ini mengambil bentuk:

$$Fitness = (W_{granularity} \times granularity) + (W_{availability} \times availability) - (Penalty_{storage} + Penalty_{latency}) \quad (1)$$

- Inisialisasi Populasi

Populasi awal dibangkitkan secara acak, setiap kromosom berisi empat gen yang mewakili variabel keputusan [10].

- Evaluasi Populasi
Nilai fitness dihitung untuk seluruh individu berdasarkan fungsi *fitness* [11].
- Seleksi
Metode Seleksi: Untuk perbandingan efektivitas, dilakukan uji baik *Tournament Selection* maupun *Roulette Wheel Selection* [12].
- Crossover dan Mutasi
 - Crossover menggunakan *One-Point* atau *Two-Point Crossover* [13].
 - Mutasi dilakukan dengan probabilitas tertentu untuk menjaga keragaman solusi [14].
- Elitisme dan Pembentukan Populasi Baru
Individu terbaik dipertahankan agar kualitas solusi tidak menurun [15].
- Terminasi
Proses dihentikan jika jumlah generasi maksimum (N=200) tercapai atau jika stagnasi solusi terbaik terdeteksi selama sejumlah iterasi tertentu. [16].
- Analisis Hasil
Performa GA dipantau melalui grafik perkembangan *best fitness* dan *average fitness* per generasi, sementara solusi terbaik dievaluasi dalam hal estimasi *storage* tahunan, *availability*, dan *latency* [17].



Gambar 1. Diagram Blok Penelitian dengan *Genetic Algorithm* (GA)

3. HASIL DAN PEMBAHASAN

Penelitian ini bertujuan untuk mengoptimalkan *trade-off* antara penyimpanan data (*storage*) dan waktu akses data (latensi) dalam manajemen data industri kilang menggunakan GA [9]. Berdasarkan hasil uji coba yang dilakukan, GA terbukti efektif dalam menemukan solusi optimal yang memenuhi kebutuhan sistem manajemen data dengan memperhitungkan keterbatasan penyimpanan dan SLA latensi [12]. Berikut adalah pembahasan berdasarkan hasil eksperimen yang diperoleh [7].

A. Pengaruh Metode Seleksi dan Crossover terhadap Fitness

Hasil percobaan menunjukkan bahwa metode seleksi *Tournament* dengan *crossover One-Point* memberikan solusi terbaik dalam hal *fitness* [10]. Pada skenario "*Balance (Default)*", dengan pengaturan *Tournament* dan *One-Point crossover*, *fitness* terbaik yang diperoleh adalah 67.094, dengan penggunaan penyimpanan 432.59 GB dan latensi 15.00 ms [11]. Sebaliknya, pada pengaturan *Roulette* dengan *Two-Point crossover*, *fitness* sedikit menurun menjadi 65.760, dengan penyimpanan meningkat menjadi 446.49 GB, dan terjadi sedikit peningkatan pada latensi serta penurunan *availability* [13]. Gambar 2 menunjukkan bahwa seleksi *Tournament* lebih efektif dalam menemukan solusi optimal dibandingkan dengan metode *Roulette*, meskipun *Two-Point crossover* memberikan variasi solusi yang lebih banyak [14].



Gambar 2. *Tournament, One-Point, Balance (Default)*

B. Pengaruh Skenario Trade-Off terhadap Storage dan Latency

Pada pengujian skenario *trade-off* yang melibatkan berbagai fokus (*Balance*, *Granularity* Fokus, *Availability* Fokus, dan *Penalty* Fokus), hasil percobaan menunjukkan bahwa setiap skenario menghasilkan konfigurasi yang unik dan *trade-off* yang berbeda antara *storage*, *latency*, dan *availability* [14].

Skenario "*Granularity Fokus*"

Dalam skenario ini, bobot pada granularitas data ditingkatkan, mendorong sistem untuk mengoptimalkan *Sampling Rate* yang lebih tinggi dengan tingkat kompresi yang lebih rendah [15]. Meskipun pengoptimalan granularitas data menjadi fokus utama, *fitness* keseluruhan justru menurun dibandingkan *baseline "Balance"*. *Fitness* terbaik yang diperoleh pada skenario ini adalah 55.509, jauh lebih rendah dari *baseline* (67.094), dengan *storage* 432.59 GB dan

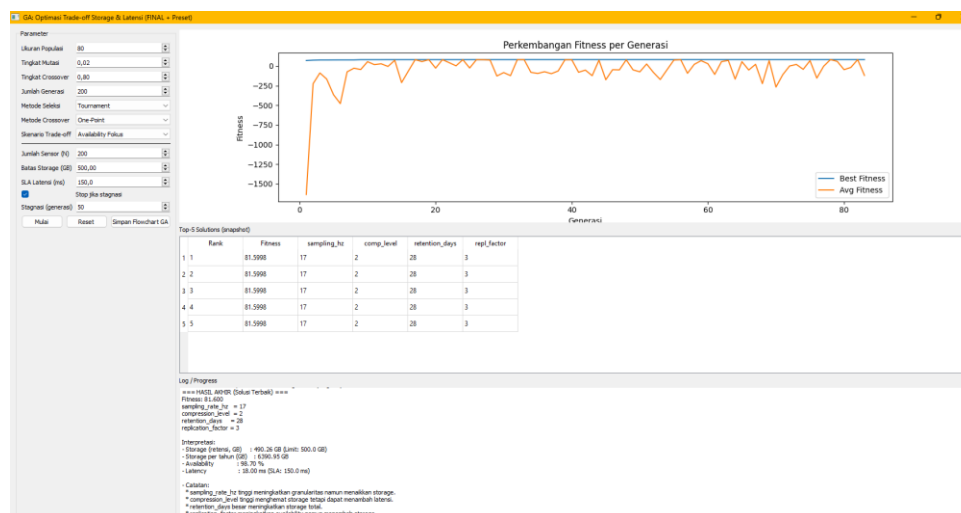
latensi 15 ms [9]. Penurunan *fitness* seperti tampak pada Gambar 3 ini terjadi karena *sampling rate* yang tinggi mengakibatkan penurunan signifikan pada *availability* sistem, menunjukkan bahwa pengorbanan ketersediaan demi kualitas data tidak efektif secara keseluruhan.



Gambar 3. Tournament, Two-Point, Granularity Fokus

Skenario "Availability Fokus"

Dalam skenario ini, algoritma menekankan peningkatan *availability*, terutama dengan menambah *replication factor* [16]. Skenario ini menghasilkan solusi terbaik secara keseluruhan karena mencapai *fitness* tertinggi, yaitu 81.600, dengan *storage* meningkat menjadi 490.26 GB [17]. Tingginya nilai *fitness* (jauh di atas *baseline* 67.094) mengindikasikan bahwa bobot utilitas pada *availability* dan *granularitas* jauh lebih dominan daripada penalti biaya *storage* tambahan. Meskipun *storage* meningkat, *availability* berhasil mencapai 98.70% seperti tampak pada Gambar 4. Hal ini menunjukkan bahwa untuk sistem kilang, menjamin ketersediaan data memiliki nilai operasional yang sangat tinggi dan dapat membenarkan peningkatan penggunaan *storage*.



Gambar 4. Tournament, One-Point, Availability Fokus

Skenario "Penalty Fokus"

Pada skenario ini, *penalty storage* ditingkatkan secara signifikan untuk memprioritaskan pengurangan penggunaan penyimpanan [18]. Hasilnya, *storage* berhasil ditekan menjadi 465.03 GB, tetapi hal ini berdampak negatif pada kinerja sistem secara keseluruhan seperti tampak pada Gambar 5. *Fitness* yang diperoleh adalah 65.925, yang lebih rendah dari *baseline*, dan disertai dengan penurunan *availability* serta peningkatan *latensi* [19]. Skenario ini menunjukkan bahwa terlalu menekankan pada penghematan *storage* memaksa GA memilih konfigurasi dengan *sampling rate* atau *replication factor* yang terlalu rendah, sehingga mengorbankan utilitas dan keandalan sistem demi efisiensi biaya penyimpanan.



Gambar 5. Tournament, Two-Point, Penalty Fokus

C. Trade-Off antara Storage dan Latensi

Berdasarkan hasil uji coba, *trade-off* antara penggunaan *storage* dan *latensi* dapat dimitigasi dengan GA, namun sensitif terhadap bobot kriteria. Pada skenario "*Balance (Default)*" (yang menghasilkan *fitness* 67.094), GA berhasil mengidentifikasi konfigurasi di mana *storage* dikendalikan pada 432.59 GB, sementara *latensi* tetap berada dalam batas optimal, yaitu 15.00 ms [13]. Hal ini menunjukkan bahwa GA efektif dalam mencari solusi optimal yang meminimalkan *storage* tanpa melanggar *SLA latensi*.

Sebaliknya, skenario "*Granularity Fokus*" (dengan *fitness* 55.509) memperburuk *trade-off*. Meskipun *latensi* tetap 15.00 ms (tidak berubah dari *baseline*), penurunan drastis pada *fitness* yang disebabkan oleh *availability* yang rendah, bersamaan dengan peningkatan *storage* menjadi 432.49 GB, mengindikasikan bahwa pengoptimalan granularitas yang berlebihan tidak menghasilkan solusi yang *viable* secara operasional [14]. Peningkatan ini memperburuk keseimbangan antara kedua parameter, membuktikan bahwa keseimbangan *latensi-storage* sangat bergantung pada *availability* data dalam konteks manajemen data kilang [10].

Solusi terbaik (*fitness* 81.600 pada skenario "*Availability Fokus*") menunjukkan bahwa, jika *availability* dijadikan prioritas, GA akan memilih *replication factor* tinggi yang meningkatkan *storage*. Peningkatan *storage* ini adalah harga yang harus dibayar untuk performa sistem yang optimal. Ini merupakan ilustrasi nyata dari *trade-off*: peningkatan *storage* diterima sebagai kompensasi untuk memaksimalkan utilitas sistem secara keseluruhan.

D. Evaluasi Hasil dan Implikasi

Hasil uji coba pada Tabel 1 menegaskan bahwa GA adalah metode yang efektif untuk memecahkan persoalan *trade-off* multi-objektif antara *storage* dan *latency* dalam manajemen data industri kilang. Temuan menunjukkan bahwa *Availability Fokus* menghasilkan solusi Pareto optimal terbaik (Fitness 81.600) karena nilai operasional tinggi dari ketersediaan data mendominasi biaya *storage* tambahan. Ini memberikan implikasi praktis bahwa dalam manajemen data kilang, investasi *storage* tambahan untuk menjamin *availability* adalah pilihan optimal.

Meskipun demikian, masih terdapat tantangan dalam menemukan konfigurasi yang optimal secara simultan untuk semua parameter [12]. Sebagai penelitian lanjutan, eksperimen dapat dilakukan dengan menyesuaikan tingkat mutasi yang lebih dinamis, menggunakan metode crossover hibrid, atau menambahkan lebih banyak kendala (*constraint*) pada masalah optimasi untuk eksplorasi ruang solusi yang lebih mendalam [18]. Dengan demikian, GA dapat digunakan sebagai alat bantu pengambilan keputusan strategis untuk mengoptimalkan sistem manajemen data industri kilang, serta memberikan gambaran mendalam tentang solusi terbaik dalam berbagai konfigurasi bobot operasional [16].

Tabel 1 Rekapitulasi Hasil Optimasi Sistem Manajemen Data Berdasarkan Fitness dan Parameter Operasional

<i>Fitness</i>	<i>Sampling Rate & Compression</i>	<i>Retention & Replication</i>	<i>Storage (GB)</i>	<i>Avail. & Latency</i>
64.574	20 Hz / <i>Compression Level 3</i>	<i>Retention 30 / Replication 3</i>	463.49 / 5639.08	95.55 % / 21.00 ms
55.509	20 Hz / <i>Compression Level 1</i>	<i>Retention 12 / Replication 3</i>	370.79 / 11278.15	101.85 % / 15.00 ms
81.600	17 Hz / <i>Compression Level 2</i>	<i>Retention 28 / Replication 3</i>	490.26 / 6390.95	98.70 % / 18.00 ms

4. SIMPULAN

Penelitian ini berhasil mengimplementasikan GA sebagai metode yang efektif untuk mengoptimalkan persoalan *trade-off* multi-objektif antara kapasitas penyimpanan (*storage*) dan waktu akses data (latensi) dalam manajemen data industri kilang. GA terbukti mampu menemukan serangkaian solusi yang menyeimbangkan *granularitas* dan *availability* sistem terhadap biaya *storage* dan *penalty* latensi.

Hasil paling signifikan menunjukkan bahwa solusi optimal terbaik ditemukan pada skenario *Availability Fokus*, mencapai nilai *fitness* maksimum sebesar 81.600 (jauh lebih tinggi dari *baseline* 67.094). Temuan ini memberikan implikasi krusial bagi operasional kilang, peningkatan *storage* (sebesar 490.26 GB) untuk menjamin *availability* sistem (98.70%) merupakan konfigurasi yang paling optimal, menunjukkan bahwa nilai operasional ketersediaan jauh melebihi biaya tambahan penyimpanan. Adapun, konfigurasi operator optimal dicapai dengan menggunakan Seleksi *Tournament* dan *Crossover One-Point*.

Meskipun efektivitas GA telah terbukti, tantangan *trade-off* tetap sensitif terhadap bobot kriteria yang digunakan. Oleh karena itu, penelitian lanjutan disarankan untuk mengeksplorasi optimasi *Pareto Front* (*Multi-Objective GA*) secara eksplisit atau menyesuaikan parameter GA seperti tingkat mutasi yang lebih dinamis dan penerapan kendala (*constraints*) yang lebih ketat untuk mencapai keseimbangan *storage* dan *latensi* yang lebih adaptif.

5. DAFTAR PUSTAKA

- [1] T. P. Raptis, A. Passarella, and M. Conti, "Data management in industry 4.0: State of the art and open challenges," *arXiv*, 2019. [Online]. Available: <https://doi.org/10.48550/ARXIV.1902.06141>
- [2] R. Amanda, S. A. Putri, Y. N. M. Arifan, R. Hidayat, and M. Ikaningtyas, "Optimalisasi proses operasional dengan menggabungkan teknologi IoT dan big data: Studi kasus pada PT Pertamina dalam industri minyak dan gas," 2024.
- [3] N. Freitas, A. D. Rocha, and J. Barata, "Data management in industry: Concepts, systematic review and future directions," *J. Intell. Manuf.*, vol. 36, no. 5, pp. 129–139, 2025. [Online]. Available: <https://doi.org/10.1007/s10845-025-02570-z>
- [4] H. Patel, D. Prajapati, D. Mahida, and M. Shah, "Transforming petroleum downstream sector through big data: A holistic review," *J. Petrol. Explor. Prod. Technol.*, vol. 10, no. 6, pp. 2601–2612, 2020. [Online]. Available: <https://doi.org/10.1007/s13202-020-00889-2>
- [5] M. C. McCorry, K. F. Reardon, M. Black, C. Williams, G. Mirica, S. Boormeester, A. Underhill, B. Sumona, J. M. Halpern, S. Sarkar, N. S. Swami, and K. A. Biofabrikasi, "Teknologi sensor untuk kontrol kualitas dalam produksi jaringan rekayasa," 2023.
- [6] F. O. Onyeke, O. Odujobi, F. E. Adikwu, and T. Y. Elele, "The role of data-driven insights in industrial control systems: Advancing predictive maintenance and operational efficiency in refinery processes," *Eng. Sci. Technol. J.*, vol. 5, no. 12, pp. 3266–3274, 2024. [Online]. Available: <https://doi.org/10.51594/estj.v5i12.1775>
- [7] A. Romdhoni and Tamaji, "Efektivitas digitalisasi IoT vibration online monitoring dalam peningkatan akurasi predictive maintenance pada main fuel oil pump mesin PLTG Mitsubishi MW-701," 2024.
- [8] S. Mahdizadeh, et al., "Latency and cost-aware genetic-based framework for task scheduling," *arXiv*, 2025. [Online]. Available: <https://arxiv.org/pdf/2507.12445.pdf>
- [9] M. V. Shkvaryliuk, "The use of genetic algorithms for multicriteria optimization," *CEUR-WS*, 2021. [Online]. Available: <https://ceur-ws.org/Vol-3048/paper13.pdf>
- [10] UMY Journal, "Genetic algorithm optimization for a novel dynamic lot sizing model," *J. UMY*, 2023. [Online]. Available: <https://journal.umy.ac.id/index.php/jrc/article/download/20667/8830/77633>
- [11] Aidic, "Development of optimisation model based on genetic algorithms for palm oil refining industry," *Aidic Conference Series*, vol. 17, no. 56, pp. 125, 2017. [Online]. Available: <https://www.aidic.it/cet/17/56/125.pdf>
- [12] M. T. Sembiring, "Production optimization design in supply chain crude palm oil using genetic algorithm," *J. Eng. Technol.*, vol. 5, no. 12, pp. 3266–3274, 2020. [Online]. Available: <https://idjpcr.usu.ac.id/jet/article/download/3621/2573/11169>
- [13] R. A. C. Leuveano, "Genetic algorithm optimization for a novel dynamic lot sizing model in perishable product manufacturing," *J. UMY*, 2023. [Online]. Available: <https://journal.umy.ac.id/index.php/jrc/article/download/20667/8830/77633>
- [14] F. Syalsabila, "Sizing optimization using genetic algorithm to achieve minimal jacket structure design," *Trunojoyo Journal*, 2022. [Online]. Available: <https://journal.trunojoyo.ac.id/rekayasa/article/view/15102>
- [15] A. Norouzi, et al., "Genetic algorithm application in optimization of wireless sensor networks," *Wiley Online Library*, 2014. [Online]. Available: <https://onlinelibrary.wiley.com/doi/pdf/10.1155/2014/286575>
- [16] A. Norouzi, et al., "Genetic algorithm application in optimization of wireless sensor networks," *Wiley Online Library*, 2014. [Online]. Available: <https://onlinelibrary.wiley.com/doi/pdf/10.1155/2014/286575>
- [17] H. Alghamdi, et al., "Multi-objective genetic algorithm and machine learning optimization framework," *ScienceDirect*, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2590123024008284>

- [18] R. Muliono, N. A. Silviana, and N. Novita, "Involvement of various selection methods for genetic algorithms in production scheduling," *J. Ind. Optimiz.*, 2024. [Online]. Available: <https://joiv.org/index.php/joiv/article/viewFile/2632/1181>
- [19] C. Panggabean, "Optimized cloud resource allocation using genetic algorithm," *arXiv*, 2025. [Online]. Available: <https://arxiv.org/pdf/2504.17675.pdf>

Daftar Simbol

- S (*Sampling Rate*): Frekuensi pengambilan sampel data dari sensor (Hz).
- C (*Compression Level*): Tingkat kompresi data yang diterapkan pada data yang dikumpulkan.
- R (*Retention Days*): Durasi data disimpan dalam sistem (Hari).
- D (*Replication Factor*): Jumlah replika data untuk meningkatkan ketersediaan sistem.
- F (*Fitness*): Nilai fungsi tujuan yang mengukur kualitas solusi (Kualitas Solusi yang Dimaksimalkan).
- W_g (Bobot *Granularity*): Bobot kepentingan relatif untuk granularitas data dalam fungsi *Fitness*.
- W_a (Bobot *Availability*): Bobot kepentingan relatif untuk Ketersediaan (*Availability*) sistem dalam fungsi *Fitness*.
- P_s (*Penalti Storage*): Komponen penalti dalam fungsi *Fitness* yang terkait dengan penggunaan *storage* yang tinggi.
- P_l (*Penalti Latency*): Komponen penalti dalam fungsi *Fitness* yang terkait dengan pelanggaran SLA *latensi*.
- L (*Latency*): Waktu yang dibutuhkan untuk mengakses data dalam sistem (ms).
- A (*Availability*): Persentase ketersediaan sistem.
- G (*Granularity*): Ukuran data yang diambil dari sensor dalam setiap pengambilan sampel.
- GA (*Genetic Algorithm*): Algoritma metaheuristik berbasis evolusi untuk optimasi.
- SLA (*Service Level Agreement*): Standar yang ditetapkan untuk memastikan kinerja sistem.