

# Analisis Cost-Performance Arsitektur Komputasi Edge, Cloud, dan Hybrid untuk Industri 4.0

**<sup>1</sup>Raihan Azhar Lapandu, <sup>2</sup>Nur Widiyasono, <sup>3\*</sup>Alam Rahmatulloh**

<sup>123</sup> Informatika, Universitas Siliwangi

E-mail: [raihanazhar05@gmail.com](mailto:raihanazhar05@gmail.com), [nur.widiyasono@unsil.ac.id](mailto:nur.widiyasono@unsil.ac.id), [\\*alam@unsil.ac.id](mailto:*alam@unsil.ac.id)

**Penulis Korespondens : Alam Rahmatulloh**

**Abstrak**— Pemilihan arsitektur komputasi *Edge*, *Cloud*, atau *Hybrid* menjadi tantangan krusial di Industri 4.0 karena perbedaan signifikan dalam performa dan biaya operasional. Berbeda dari studi yang hanya berfokus pada aspek teknis, penelitian ini menyajikan analisis *cost-performance* yang komprehensif. Menggunakan skenario simulasi *predictive maintenance*, *trade-off* antara metrik teknis (utilisasi *CPU*, *RAM*, latensi, dan penyimpanan) dievaluasi dengan estimasi biaya harian. Lebih lanjut, penelitian ini mengusulkan sebuah *decision matrix* sebagai alat bantu pengambilan keputusan. Hasil menunjukkan *Edge* unggul dalam latensi rendah (17,5 ms) dan efisiensi biaya. *Cloud* menawarkan skalabilitas superior dengan biaya lebih tinggi, sementara *Hybrid* menjadi kompromi seimbang meski memiliki *overhead* dan kompleksitas tertinggi. Temuan ini memberikan kontribusi aplikatif berupa panduan empiris bagi para praktisi untuk memilih arsitektur yang paling sesuai dengan prioritas operasional dan ketersediaan anggaran.

**Kata Kunci**— Arsitektur *Hybrid*, Efisiensi Biaya, Industri 4.0, Komputasi *Cloud*, Komputasi *Edge*

**Abstract**— Choosing *Edge*, *Cloud*, or *Hybrid* computing architecture is a crucial challenge in Industry 4.0 due to significant differences in performance and operational costs. Different from studies that only focus on technical aspects, this study presents a comprehensive cost-performance analysis. Using a predictive maintenance simulation scenario, the trade-off between technical metrics (*CPU* utilization, *RAM*, latency, and storage) is evaluated with daily cost estimation. Furthermore, this study proposes a decision matrix as a decision-making tool. The results show that *Edge* excels in low latency (17.5 ms) and cost efficiency. *Cloud* offers superior scalability at a higher cost, while *Hybrid* is a balanced compromise despite having the highest overhead and complexity. These findings provide an applicable contribution in the form of empirical guidance for practitioners to choose the most appropriate architecture based on operational priorities and budget availability.

**Keywords**— *Cloud Computing*, *Cost Efficiency*, *Edge Computing*, *Hybrid Architecture*, *Industry 4.0*

This is an open access article under the CC BY-SA License.



## I. PENDAHULUAN

Revolusi Industri 4.0 menandai pergeseran fundamental dalam sistem produksi, yang dicirikan oleh integrasi sistem siber-fisik (*Cyber-Physical Systems - CPS*), *Internet of Things (IoT)*, dan kecerdasan buatan (*AI*) untuk menciptakan ekosistem manufaktur cerdas [1]. Implementasi teknologi ini memungkinkan otomatisasi tingkat lanjut, pemantauan proses secara *real-time*, dan pengambilan keputusan berbasis data, yang pada akhirnya bertujuan untuk meningkatkan efisiensi dan daya saing industri. Jantung dari transformasi ini adalah *IoT*, sebuah jaringan yang memungkinkan perangkat-perangkat industri untuk saling terhubung dan berkomunikasi secara otonom, yang menjadi pilar utama dalam digitalisasi industri [2]. Sensor-sensor yang tertanam di

lantai produksi secara konstan memantau parameter kritis seperti suhu, tekanan, dan getaran menghasilkan aliran data dalam volume dan kecepatan yang masif (*Big Data*). Ledakan data ini, meskipun merupakan aset yang sangat berharga, menghadirkan tantangan utama: bagaimana mengelola, memproses, dan menganalisisnya secara efisien, cepat, dan hemat biaya untuk mendukung pengambilan keputusan yang tangkas [3].

Untuk mengatasi tantangan pemrosesan data tersebut, tiga arsitektur komputasi utama telah menjadi opsi strategis. *Cloud Computing* menyediakan sumber daya komputasi dan penyimpanan secara elastis dan terpusat melalui internet, menawarkan skalabilitas yang nyaris tak terbatas untuk analisis data besar. Meskipun ideal untuk melatih model *machine learning* yang kompleks, kekuatannya diimbangi oleh kelemahan berupa latensi jaringan yang tinggi dan ketergantungan konektivitas [4]. *Edge Computing* hadir sebagai pendekatan alternatif yang memindahkan proses komputasi lebih dekat ke sumber data, sehingga menekan latensi dan ideal untuk aplikasi sensitif waktu seperti *predictive maintenance* [5]. Perangkat *edge* memiliki keterbatasan kapasitas komputasi dan penyimpanan. Menjembatani keduanya, Arsitektur *Hybrid* mencoba menggabungkan latensi rendah dari *Edge* dengan kapasitas komputasi tinggi dari *Cloud*, meskipun dengan konsekuensi peningkatan kompleksitas dan biaya [6].

Pemilihan arsitektur yang tepat adalah keputusan krusial yang dipengaruhi oleh *trade-off* performa dan biaya. Tinjauan literatur mengungkap beberapa kesenjangan signifikan yang menjadi justifikasi penelitian ini. Tabel 1 di bawah ini merangkum analisis kesenjangan tersebut secara sistematis.

Tabel 1. Tabel Analisis Kesenjangan Penelitian

<b>Analisis Kesenjangan Penelitian</b>		
<b>Dimensi</b>	<b>Studi Sebelumnya</b>	<b>Kesenjangan yang Diisi</b>
Evaluasi Teknis	Banyak studi fokus pada perbandingan performa teknis antara <i>Edge</i> dan <i>Cloud</i> saja [7], [8].	Mengintegrasikan ketiga arsitektur ( <i>Edge</i> , <i>Cloud</i> , <i>Hybrid</i> ) dalam satu kerangka evaluasi yang sistematis dan simultan.
Pertimbangan Biaya Operasional	Aspek ekonomi atau estimasi biaya operasional riil jarang dibahas secara eksplisit dan empiris [9].	Memperkenalkan analisis <i>cost-performance trade-off</i> sebagai inti dari evaluasi, berbasis estimasi biaya nyata.
Pengambilan Keputusan Aplikatif	Mayoritas penelitian berhenti pada penyajian hasil teknis, tanpa menyediakan alat bantu keputusan praktis [10].	Menghasilkan sebuah <i>decision matrix</i> sebagai panduan aplikatif bagi praktisi untuk memilih arsitektur.
Konteks Studi Kasus	Banyak studi berfokus pada <i>use-case</i> umum seperti <i>streaming</i> atau <i>smart cities</i> [8], [11].	Berfokus secara spesifik pada skenario <i>predictive maintenance</i> di konteks manufaktur berbasis <i>IoT</i> .
Validasi Empiris	Beberapa studi masih bersifat konseptual atau hanya berbasis simulasi perangkat lunak [12].	Melakukan eksperimen langsung dengan implementasi perangkat keras dan layanan <i>cloud</i> yang sesungguhnya.

Dari analisis Tabel 1, terlihat jelas bahwa belum ada penelitian yang secara komprehensif membandingkan ketiga arsitektur secara empiris dengan mengintegrasikan analisis performa teknis dan efisiensi biaya operasional, serta menghasilkan sebuah alat bantu keputusan praktis untuk skenario *predictive maintenance*. Berdasarkan kesenjangan penelitian yang telah diidentifikasi, penelitian ini menyajikan pendekatan alternatif melalui analisis *cost-performance trade-off* untuk membantu pemilihan arsitektur komputasi yang tepat. Tujuan utama dari penelitian ini adalah memberikan kerangka evaluasi yang holistik dan praktis bagi pengambil keputusan di industri manufaktur. Secara spesifik, penelitian ini bertujuan untuk mengevaluasi dan membandingkan secara kuantitatif metrik performa teknis—mencakup utilisasi *CPU*, *RAM*, latensi, dan penyimpanan—from ketiga arsitektur. Selain itu, penelitian ini juga akan mengembangkan model estimasi biaya operasional harian untuk setiap arsitektur berbasis skenario simulasi, dan pada puncaknya, menghasilkan sebuah *decision matrix* sebagai panduan aplikatif untuk pemilihan arsitektur.

Kontribusi utama dari penelitian ini adalah menyediakan kerangka evaluasi yang lebih lengkap, yang tidak hanya mempertimbangkan siapa yang "tercepat" atau "terkuat", tetapi juga siapa yang "paling efisien secara biaya" sesuai dengan konteks operasional spesifik—sebuah pendekatan yang hingga saat ini belum banyak dibahas secara empiris dalam literatur.

## II. METODE

Penelitian ini menggunakan pendekatan kuantitatif melalui eksperimen berbasis simulasi terkontrol untuk membandingkan performa dan estimasi biaya dari tiga arsitektur komputasi utama: *Edge Computing*, *Cloud Computing*, dan *Hybrid Computing*. Fokus utama eksperimen adalah mengevaluasi efektivitas arsitektur dalam konteks *data analytics* untuk skenario *predictive maintenance*, yang merupakan *use-case* umum dalam implementasi Industri 4.0.

### A. Desain Eksperimen

Desain eksperimen ini melibatkan tiga konfigurasi infrastruktur yang berbeda. Untuk arsitektur *Edge*, sebuah perangkat *Raspberry Pi* 4 Model B (8GB) digunakan sebagai perangkat pemrosesan lokal. Untuk arsitektur *Cloud*, digunakan layanan dari *Amazon Web Services (AWS)*, khususnya *EC2 (Elastic Compute Cloud)* untuk pemrosesan komputasi dan *RDS (Relational Database Service)* untuk manajemen *database* terpusat. Arsitektur *Hybrid* mengombinasikan keduanya, di mana pemrosesan awal seperti *preprocessing* dan penyaringan data dilakukan di *Raspberry Pi*, sebelum hasil *preprocessing* dikirim ke *AWS* untuk inferensi utama dan penyimpanan. Sumber data untuk semua skenario berasal dari simulasi pengiriman data sensor industri secara real-time menggunakan protokol *MQTT* dengan *Quality of Service (QoS)* level 1. *Dataset* yang digunakan adalah *AI4I 2020 Predictive Maintenance Dataset*, yang dianalisis menggunakan model *machine learning decision tree* yang telah dilatih sebelumnya. Model ini dipilih karena dua alasan utama: efisiensi komputasi yang ringan, sehingga relevan untuk perangkat *edge* dengan sumber daya terbatas, dan tingkat interpretasi yang tinggi.

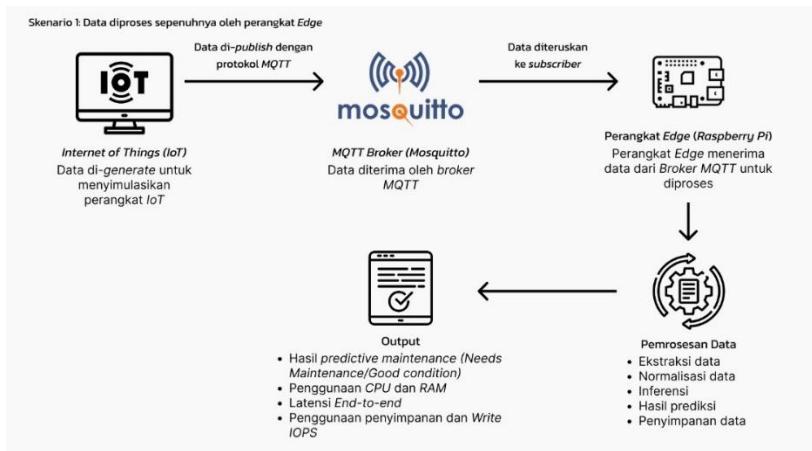
Untuk mengevaluasi setiap arsitektur secara komprehensif, serangkaian metrik kuantitatif diukur menggunakan instrumen yang sesuai. Kinerja Komputasi diukur melalui Utilisasi *CPU* (%) dan Penggunaan *RAM* (%). Utilisasi *CPU* mengindikasikan seberapa besar beban kerja yang ditanggung oleh prosesor, yang krusial untuk memahami efisiensi dan potensi *bottleneck*. Sementara itu, penggunaan *RAM* merefleksikan kebutuhan memori dari aplikasi. Kinerja Penyimpanan dievaluasi dari dua sisi: *Write IOPS (Input/Output Operations per Second)*, yang menunjukkan frekuensi operasi tulis ke disk dan merefleksikan intensitas beban kerja *database*, serta Penggunaan Ruang Disk (MB), yang mengukur jejak penyimpanan data dari waktu ke waktu. Selanjutnya, Responsivitas Sistem, sebagai metrik paling kritis untuk aplikasi real-time, diukur melalui Latensi *end-to-end (ms)*. Metrik ini mencatat total waktu yang dibutuhkan sejak pesan dikirim oleh simulator hingga hasil inferensi akhir tersedia, memberikan gambaran utuh mengenai kecepatan respons sistem. Pengukuran metrik teknis pada perangkat *Edge* dilakukan menggunakan *library Python psutil*, sedangkan pada platform *Cloud* menggunakan layanan pemantauan terintegrasi *AWS CloudWatch*.

Untuk memastikan transparansi dan reproduksibilitas, estimasi biaya operasional harian dihitung berdasarkan model dan asumsi yang terperinci. Semua biaya akhir disajikan dalam Rupiah (IDR) dengan asumsi kurs \$1 = Rp 16.000.

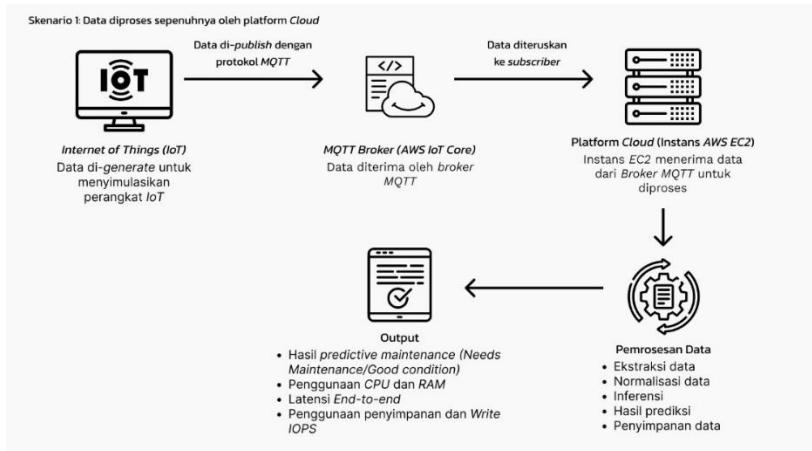
- 1) Model Biaya Arsitektur *Edge*: Biaya dihitung dengan menjumlahkan biaya amortisasi perangkat keras (*Capital Expenditure*) dan biaya konsumsi listrik (*Operational Expenditure*).
  - a. Amortisasi Perangkat: Biaya pembelian awal *Raspberry Pi 4* beserta aksesoris (diasumsikan Rp 1.500.000) disusutkan selama masa pakai ekonomis 3 tahun (1.095 hari).
  - b. Biaya Listrik: Konsumsi daya rata-rata *Raspberry Pi* saat beban eksperimen (diukur menggunakan *USB power meter* sebesar ~5 Watt) dikalikan dengan tarif listrik industri yang berlaku selama 24 jam.
- 2) Model Biaya Arsitektur *Cloud*: Biaya diestimasi berdasarkan harga publik dari layanan *AWS* yang digunakan (*on-demand pricing*). Ini mencakup biaya per jam untuk *instance EC2* dan *RDS* yang aktif, biaya penyimpanan per GB, serta estimasi biaya transfer data keluar (data *egress*).
- 3) Keterbatasan Model Biaya: Perlu dicatat bahwa model ini difokuskan pada perbandingan biaya infrastruktur langsung dan tidak mencakup biaya tersembunyi seperti biaya sumber daya manusia untuk instalasi/pemeliharaan, lisensi perangkat lunak, atau biaya kerugian akibat *downtime*.

## B. Konfigurasi dan Skenario Pengujian

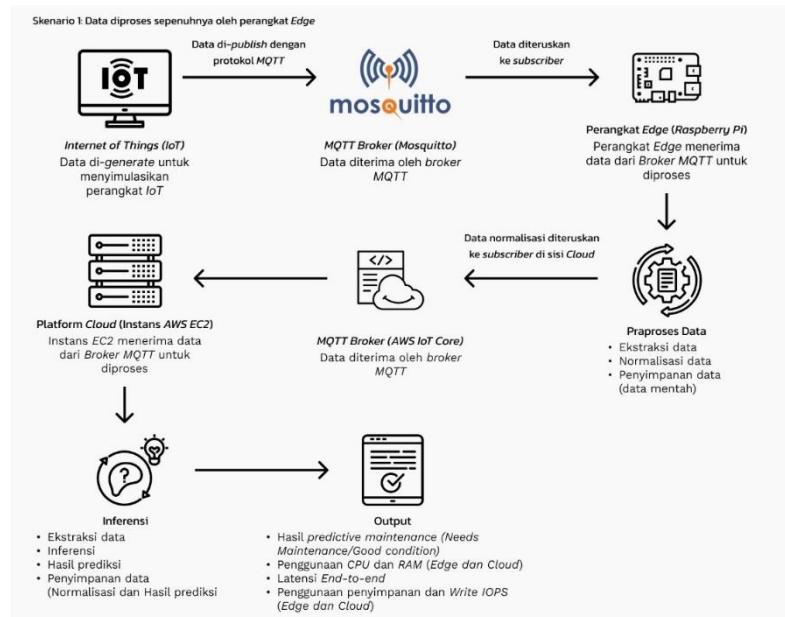
Eksperimen dijalankan melalui tiga skenario pengujian yang berbeda, masing-masing diuji selama 3 jam non-stop untuk setiap iterasi (3 kali iterasi). Pada setiap iterasi, sebanyak 216.450 pesan sensor disimulasikan, dikirim dengan kecepatan dasar 20 pesan/detik dan diselingi lima lonjakan beban sebanyak 30 pesan/detik untuk menguji ketahanan sistem. Skema pengujian untuk setiap skenario diilustrasikan pada Gambar 1, 2, dan 3.



Gambar 1. Skema Pengujian Skenario 1



Gambar 2. Skema Pengujian Skenario 2



Gambar 3. Skema Pengujian Skenario 3

Eksperimen ini dijalankan melalui tiga skenario alur kerja yang berbeda untuk menguji setiap arsitektur. Skenario pertama mengimplementasikan arsitektur *Edge* murni, di mana seluruh proses mulai dari penerimaan data MQTT, inferensi model prediktif, hingga penyimpanan hasil dieksekusi secara mandiri dan sepenuhnya lokal di perangkat *Edge*, seperti diilustrasikan pada Gambar 1. Berbeda dengan itu, skenario kedua mengadopsi pendekatan berbasis *Cloud* di mana seluruh beban kerja didelegasikan ke platform *Cloud*. Sebagaimana terlihat pada Gambar 2, data dari simulator langsung dikirim ke broker MQTT di *Cloud* untuk selanjutnya diproses dan disimpan di sana. Adapun skenario ketiga, yang alurnya divisualisasikan pada Gambar 3, menguji model *Hybrid* dengan membagi tugas komputasi; proses *preprocessing* data awal yang ringan dilakukan di *Edge*, kemudian hasilnya dikirim ke *Cloud* untuk menjalankan inferensi model yang lebih kompleks dan penyimpanan data akhir.

Untuk menjaga validitas dan reliabilitas hasil, serangkaian kontrol eksperimen diterapkan. Waktu pengujian diseragamkan pada kondisi jaringan dan lokasi yang sama untuk mengendalikan variabel eksternal. Pengukuran metrik dilakukan secara otomatis setiap 60 detik dengan logging yang sistematis untuk objektivitas. Selain itu, setiap skenario dijalankan ulang sebanyak tiga kali. Prosedur repetisi ini bertujuan untuk memastikan konsistensi hasil dan memungkinkan perhitungan nilai rerata serta standar deviasi, sehingga mengurangi dampak anomali sesaat. Setelah data terkumpul, analisis data dilakukan menggunakan metode statistik deskriptif (mean, standar deviasi) untuk mengolah dan merangkum temuan dari setiap iterasi. Analisis ini berfokus pada perbandingan performa rata-rata dan stabilitas dari setiap arsitektur. Hasil interpretasi data kuantitatif ini kemudian digunakan sebagai input fundamental dalam penyusunan matriks perbandingan dan *decision matrix* akhir, yang secara visual memetakan *trade-off* antara performa dan biaya operasional dari masing-masing arsitektur.

### **III. HASIL DAN PEMBAHASAN**

#### A. Hasil Pengujian

Tabel 2. Hasil Pengujian Seluruh Skenario

<b>Hasil Pengujian Seluruh Skenario</b>			
<b>Parameter</b>	<b>Skenario 1: Edge</b>	<b>Skenario 2: Cloud</b>	<b>Skenario 3: Hybrid</b>
CPU Usage (Rata-rata)	7.18%	9.20%	Edge: 5.28% Cloud: 7.62%
RAM Usage (Rata-rata)	48.11%	46.46%	Edge: 46.28% Cloud: 47.55%
Write IOPS (Rata-rata)	10.6	4.02	Edge: 9.74 Cloud: 3.94
Disk Usage	20 MB	17.55 MB	Edge: 15 MB Cloud: 18.55 MB
End-to-End Latency (Rata-rata)	17.53 ms	116.12 ms	146.89 ms
Skalabilitas	Terbatas	Tinggi	Tinggi (Cloud), Terbatas (Edge)
Biaya Harian (Estimasi)	~Rp 2.000	~Rp 15.000	~Rp 20.000

Bagian ini menyajikan dan membahas hasil kuantitatif dari eksperimen yang dilakukan. Data performa dan estimasi biaya yang dihasilkan dari setiap skenario dirangkum dalam Tabel 2, yang menjadi dasar bagi analisis teoritis dan kualitatif. Pembahasan akan difokuskan pada interpretasi temuan-temuan kunci, implikasinya, serta hubungannya dengan konsep teoretis dan penelitian lain yang relevan.

### B. Komputasi dan Penyimpanan

Temuan paling signifikan dari analisis kinerja adalah *trade-off* antara efisiensi sumber daya lokal dan *overhead* platform terdistribusi. Seperti terlihat pada Tabel 3, arsitektur *Cloud* menunjukkan beban *CPU* tertinggi (9.20%). Secara teoritis, ini dapat dijelaskan bukan hanya oleh beban pemrosesan data, tetapi juga oleh *overhead* inheren dari platform virtualisasi, manajemen jaringan, dan lapisan keamanan yang kompleks di sisi *Cloud*. Temuan ini memperkuat argumen bahwa komputasi *Cloud*, meskipun kuat, memiliki "biaya" komputasi tak terlihat. Sebaliknya, arsitektur *Hybrid* berhasil mendistribusikan beban komputasi, namun hal ini berimplikasi pada peningkatan kompleksitas manajemen sistem. Dari sisi penyimpanan, efisiensi operasi tulis (IOPS terendah) pada arsitektur *Cloud* menegaskan keunggulan sistem *database* terkelola yang telah teroptimasi. Argumen ini menyoroti bahwa untuk beban kerja dengan intensitas tulis yang tinggi, platform *Cloud* yang matang menawarkan solusi yang lebih efisien daripada *database* lokal sederhana. Namun, efisiensi ini harus ditebus dengan penggunaan total *disk* terbesar pada skenario *Hybrid* (33.55 MB), yang merupakan konsekuensi dari strategi redundansi data. Implikasi aplikatifnya jelas: untuk sistem yang memprioritaskan ketahanan data (*data resiliency*), duplikasi data pada model *Hybrid* adalah fitur, bukan kelemahan.

### C. Latensi Sistem

Data latensi secara kuantitatif mengonfirmasi hipotesis dasar penelitian. Latensi ultra-rendah pada arsitektur *Edge* (17.53 ms) membuktikan kelayakannya untuk aplikasi industri yang menuntut respons real-time. Temuan yang lebih penting untuk didiskusikan adalah sumber dari latensi tinggi pada arsitektur *Cloud* dan *Hybrid*. Latensi *Cloud* (116.12 ms) tidak hanya disebabkan oleh jarak fisik, melainkan akumulasi penundaan dari berbagai *network hop* dan proses enkripsi/autentikasi. Fenomena yang paling menonjol adalah latensi tertinggi pada arsitektur *Hybrid* (146.89 ms). Temuan ini sejalan dengan konsep *latency stacking* yang diidentifikasi oleh Silva et al. (2021), namun penelitian ini melangkah lebih jauh dengan membedah komponen penyebabnya secara konseptual. Akumulasi latensi ini berasal dari proses serial yang mencakup: *overhead* pemrosesan awal di *Edge*, proses serialisasi data, penundaan dari *network hop* tambahan ke *Cloud*, dan akhirnya seluruh penundaan yang juga dialami oleh arsitektur *Cloud*. Hal ini berimplikasi bahwa penerapan arsitektur *Hybrid* tanpa desain alur kerja paralel dapat secara kontra-intuitif menghasilkan sistem yang paling lambat, sebuah argumen penting bagi para arsitek sistem.

### D. Implikasi Strategis dari Biaya, Skalabilitas, dan Konteks Adopsi

Integrasi analisis biaya dengan performa teknis memberikan dimensi strategis yang krusial. Arsitektur *Edge*, dengan biaya terendah dan skalabilitas terbatas, merepresentasikan model *Capital Expenditure* (*CapEx*) yang cocok untuk lingkungan terkontrol dengan kebutuhan yang

terdefinisi jelas, seperti pada lini produksi spesifik atau untuk adopsi oleh UKM. Sebaliknya, arsitektur *Cloud* menawarkan model *Operational Expenditure (OpEx)* yang fleksibel dengan skalabilitas tinggi, selaras dengan kebutuhan perusahaan dalam fase pertumbuhan cepat yang memprioritaskan kelincahan. Biaya tertinggi pada arsitektur *Hybrid* adalah cerminan dari nilainya sebagai investasi dalam fleksibilitas dan ketahanan sistem. Implikasi teoretisnya adalah bahwa "arsitektur optimal" bukanlah konsep absolut, melainkan fungsi dari profil risiko dan tujuan bisnis suatu organisasi. Untuk industri kritis di mana biaya downtime jauh melampaui biaya infrastruktur, model *Hybrid* menjadi pilihan yang paling rasional.

#### E. Decision Matrix

Untuk menerjemahkan temuan kuantitatif dan diskusi kualitatif di atas menjadi sebuah panduan yang dapat ditindaklanjuti, penelitian ini menghasilkan sebuah matriks keputusan (*decision matrix*). Matriks pada Tabel 3 ini memetakan prioritas operasional yang berbeda dengan rekomendasi arsitektur yang paling sesuai, beserta justifikasi ringkas berdasarkan hasil eksperimen.

Tabel 3. Decision Matrix

Decision Matrix		
Prioritas Utama	Rekomendasi	Penjelasan
Arsitektur		
Kecepatan Respons ( <i>Edge</i> (Latensi Ultra-Rendah))		Latensi terendah (17.53 ms), pemrosesan dilakukan di sumber data, menghindari penundaan jaringan. Ideal untuk kontrol proses real-time.
Efisiensi Biaya Maksimal ( <i>Edge</i> )		Biaya operasional harian paling rendah (~Rp 2.000), menggunakan perangkat keras dengan konsumsi daya minimal dan tanpa biaya langganan.
Skalabilitas & <i>Cloud</i> Pertumbuhan Cepat		Infrastruktur elastis yang dapat disesuaikan dengan permintaan. Cocok untuk menangani volume data yang besar dan fluktuatif tanpa investasi perangkat keras.
Ketahanan Sistem & <i>Hybrid</i> Redundansi Data		Menggabungkan penyimpanan lokal ( <i>Edge</i> ) dan terpusat ( <i>Cloud</i> ), memastikan data tetap aman dan operasional jika salah satu platform mengalami kegagalan.
Fleksibilitas & <i>Hybrid</i> Keseimbangan Performa		Mampu menjalankan tugas sensitif latensi di <i>Edge</i> dan analisis data berat di <i>Cloud</i> . Menawarkan solusi paling seimbang meskipun dengan biaya dan kompleksitas tertinggi.

Decision Matrix pada Tabel 3 berfungsi sebagai panduan praktis untuk menerjemahkan hasil temuan penelitian ke dalam rekomendasi strategis. Matriks ini menegaskan bahwa jika prioritas utama sebuah organisasi adalah kecepatan respons yang instan (latensi ultra-rendah) atau efisiensi biaya yang maksimal, maka arsitektur Edge adalah pilihan yang paling tepat. Hal ini didasarkan pada keunggulannya dalam latensi terendah (17.53 ms) dan biaya operasional harian yang minimal. Jika prioritasnya adalah skalabilitas dan kemampuan untuk mendukung pertumbuhan cepat, maka arsitektur Cloud menjadi rekomendasi utama karena infrastrukturnya yang elastis mampu menangani volume data yang besar dan fluktuatif. Untuk kebutuhan yang lebih kompleks seperti ketahanan sistem melalui redundansi data atau fleksibilitas untuk menyeimbangkan performa, maka arsitektur Hybrid adalah pilihan yang paling sesuai. Meskipun memiliki biaya dan kompleksitas tertinggi, model ini mampu menjalankan tugas sensitif latensi di Edge sambil memanfaatkan kekuatan analisis data berat di Cloud, menawarkan solusi yang paling seimbang untuk industri kritis.

#### IV. KESIMPULAN

Penelitian ini menjawab tujuan utama untuk merumuskan strategi pemilihan arsitektur komputasi dengan menyimpulkan bahwa tidak ada arsitektur tunggal, baik *Edge*, *Cloud*, maupun *Hybrid* yang secara universal superior untuk semua skenario data analytics di Industri 4.0. Pemilihan arsitektur yang optimal terbukti merupakan sebuah fungsi dari *trade-off* yang fundamental antara metrik performa teknis, seperti latensi dan skalabilitas, dengan efisiensi biaya operasional. Peningkatan utama yang ditawarkan penelitian ini pada bidang teknik industri dan sains komputasi adalah pengenalan kerangka evaluasi holistik yang mengintegrasikan dimensi ekonomi ke dalam analisis teknis. Pendekatan *cost-performance* dan *decision matrix* yang dihasilkan menyediakan sebuah metodologi terstruktur untuk pengambilan keputusan berbasis data, yang sebelumnya lebih sering didasarkan pada pertimbangan teknis semata. Berdasarkan temuan tersebut, saran praktis yang dapat diberikan adalah: untuk aplikasi yang menuntut respons real-time dan sensitif terhadap biaya, seperti pada UKM, arsitektur *Edge* direkomendasikan. Sebaliknya, organisasi yang memprioritaskan skalabilitas tinggi untuk mengelola volume data yang besar dan fluktuatif disarankan untuk mengadopsi arsitektur *Cloud*. Sementara itu, untuk sistem industri kritis yang membutuhkan ketahanan data (data resiliency) dan fleksibilitas, arsitektur *Hybrid* menjadi pilihan strategis meskipun memerlukan investasi dan kompleksitas yang lebih tinggi.

#### DAFTAR PUSTAKA

- [1] H. Prasetyo and W. Sutopo, “Perkembangan Keilmuan Teknik Industri Menuju Era Industri 4.0,” *Angewandte Chemie International Edition*, 6(11), 951–952., vol. 13, no. 2, 2019.
- [2] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, “Industrial internet of things: Challenges, opportunities, and directions,” *IEEE Trans Industr Inform*, vol. 14, no. 11, 2018, doi: 10.1109/TII.2018.2852491.
- [3] I. H. Khan and M. Javaid, “Role of Internet of Things (IoT) in Adoption of Industry 4.0,” *Journal of Industrial Integration and Management*, vol. 7, no. 4, 2022, doi: 10.1142/S2424862221500068.
- [4] M. Ghobaei-Arani, A. Souri, and A. A. Rahamanian, “Resource Management Approaches in Fog Computing: a Comprehensive Review,” 2020. doi: 10.1007/s10723-019-09491-1.
- [5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” *IEEE Internet Things J*, vol. 3, no. 5, 2016, doi: 10.1109/JIOT.2016.2579198.

- [6] F. C. Andriulo, M. Fiore, M. Mongiello, E. Traversa, and V. Zizzo, “Edge Computing and Cloud Computing for Internet of Things: A Review,” Dec. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/informatics11040071.
- [7] S. Gautam, “Comparison of Edge and Cloud Computing Technology for Industry 4.0 Perspective on the Future,” *Interantional Journal of Scientific Research in Engineering and Management*, vol. 07, no. 07, 2023, doi: 10.55041/ijssrem24555.
- [8] P. Silva, A. Costan, and G. Antoniu, “Investigating Edge vs. Cloud Computing Trade-offs for Stream Processing,” 2021.
- [9] A. Ali-Eldin, B. Wang, and P. Shenoy, “The Hidden cost of the Edge: A Performance Comparison of Edge and Cloud Latencies,” Apr. 2021, [Online]. Available: <http://arxiv.org/abs/2104.14050>
- [10] T. Küfner, S. Schönig, R. Jasinski, and A. Ermer, “Vertical data continuity with lean edge analytics for industry 4.0 production,” *Comput Ind*, vol. 125, 2021, doi: 10.1016/j.compind.2020.103389.
- [11] D. Loghin, L. Ramapantulu, and Y. M. Teo, “Towards Analyzing the Performance of Hybrid Edge-Cloud Processing,” 2019.
- [12] B. Bajic, I. Cosic, B. Katalinic, S. Moraca, M. Lazarevic, and A. Rikalovic, “Edge computing vs. Cloud computing: Challenges and opportunities in industry 4.0,” in *Annals of DAAAM and Proceedings of the International DAAAM Symposium*, 2019. doi: 10.2507/30th.daaam.proceedings.120.