Jurnal Minfo Polgan
Volume 14, Nomor 2, Januari 2026
DOI : https://doi.org/10.33395/jmp.v14i2.15576
e-ISSN : 2797-3298
p-ISSN : 2089-9424

Komparasi Kinerja dan Toleransi Kesalahan antara Cloud dan Fog Computing dalam Aplikasi IoT Terdistribusi

[1]Refi Riduan Achmad, [2]Muhammad Abil, [3]Muhammad Raihan Fadhilah, [4]Sandi
[1,2,3,4]Universitas Nahdlatul Ulama Kalimantan Timur
[1]refiriduanachmad@unukaltim.ac.id, [2]mhmmdabil828@gmail.com, [3]rehan.nuh80@gmail.com, [4]sndiaza12@gmail.com

Submit : 12 Nov 2025 | **Diterima** : 02 Jan 2026 | **Terbit** : 10 Jan 2026

ABSTRAK

Perkembangan Internet of Things (IoT) menuntut sistem komputasi yang mampu memproses data secara cepat dan andal. Komputasi awan (Cloud Computing) selama ini menjadi solusi utama dalam pengelolaan data IoT, namun arsitekturnya yang terpusat menyebabkan latensi tinggi dan ketergantungan pada jaringan. Untuk mengatasi hal tersebut, komputasi kabut (Fog Computing) dikembangkan dengan mendistribusikan proses lebih dekat ke sumber data. Penelitian ini bertujuan untuk membandingkan kinerja dan toleransi kesalahan antara arsitektur Cloud Computing dan Fog Computing dalam lingkungan IoT terdistribusi. Pengujian dilakukan dengan mensimulasikan sistem publisher–broker–consumer menggunakan protokol MQTT di lingkungan Docker. Parameter yang diuji meliputi rata-rata latency, maksimum, minimum, jitter, serta toleransi kesalahan saat terjadi gangguan broker. Hasil pengujian menunjukkan bahwa Fog Computing memiliki rata-rata latency sebesar 0.0019 detik, lebih rendah dibandingkan Cloud Computing sebesar 0.0022 detik. Namun, pada uji toleransi kesalahan, Cloud Computing menunjukkan stabilitas yang sedikit lebih tinggi dengan tingkat kehilangan pesan 55% dibanding Fog sebesar 56.67%. Temuan ini mengindikasikan bahwa Fog Computing lebih unggul untuk aplikasi IoT real-time karena latensi rendah dan pemulihan cepat, sedangkan Cloud lebih sesuai untuk pengolahan data berskala besar dan berkelanjutan. Penelitian selanjutnya dapat memperluas pengujian terhadap parameter bandwidth dan konsumsi energi untuk mendapatkan gambaran performa sistem yang lebih komprehensif.

Kata kunci: Internet of Things, Fog Computing, Cloud Computing, Latency.

PENDAHULUAN

Internet of Things (IoT) berkembang sebagai arsitektur teknologi yang menghubungkan beragam perangkat melalui jaringan internet, memungkinkan pertukaran data dan pelaksanaan tugas secara otomatis tanpa keterlibatan manusia secara langsung. Teknologi ini telah merambah berbagai sektor, mulai dari industri manufaktur, transportasi, kesehatan, pertanian, hingga sistem rumah pintar. Lonjakan jumlah perangkat yang aktif menciptakan lalu lintas data dalam skala besar yang menuntut sistem komputasi dengan kemampuan pemrosesan tinggi serta manajemen jaringan yang efisien (Suryadi, Octiva, Fajri, Nuryanto, & Hakim, 2024).

Selama beberapa tahun terakhir, komputasi awan (cloud computing) menjadi pilihan dominan untuk menangani pemrosesan dan penyimpanan data IoT. Namun, pendekatan terpusat ini memiliki keterbatasan saat dihadapkan pada kebutuhan aplikasi real-time dan mobilitas tinggi. Meskipun Cloud Computing menawarkan skalabilitas yang masif, ketergantungannya pada transmisi data jarak jauh menuju pusat data global menyebabkan masalah utama, yaitu latensi tinggi yang tidak dapat diterima oleh aplikasi sensitif waktu. Tingkat latensi yang relatif tinggi serta potensi kegagalan jaringan tersebut dapat menghambat performa sistem. Di sinilah muncul pendekatan komputasi kabut (fog computing), yang menawarkan pemrosesan lebih dekat ke sumber data untuk meningkatkan kecepatan respons dan toleransi terhadap kesalahan.

Komputasi awan (cloud computing) menawarkan skalabilitas tinggi, kapasitas

Jurnal Minfo Polgan
Volume 14, Nomor 2, Januari 2026
DOI : https://doi.org/10.33395/jmp.v14i2.15576
e-ISSN : 2797-3298
p-ISSN : 2089-9424

penyimpanan besar, serta kemudahan integrasi layanan (Ilahi et al., 2024). Namun, arsitektur cloud yang bersifat terpusat sering kali menyebabkan bottleneck pada jaringan, terutama ketika jumlah perangkat IoT semakin banyak dan tersebar. Sebaliknya, komputasi kabut (fog computing) menghadirkan pendekatan yang lebih terdistribusi dengan melakukan pemrosesan data di dekat sumbernya (Elektro & Telkom, 2020), sehingga dapat mengurangi latensi dan meningkatkan toleransi terhadap kegagalan jaringan.

Meskipun kedua pendekatan ini banyak digunakan dalam pengembangan sistem IoT, masih diperlukan analisis komprehensif untuk memahami perbedaan kinerja dan toleransi kesalahan di antara keduanya. Perbandingan ini penting terutama untuk menentukan pendekatan yang paling sesuai dengan karakteristik aplikasi IoT tertentu, seperti smart city, smart health, dan smart transportation, yang menuntut kecepatan dan keandalan tinggi.

Artikel ini bertujuan untuk melakukan komparasi kinerja dan toleransi kesalahan antara komputasi awan dan komputasi kabut dalam konteks aplikasi IoT terdistribusi. Analisis difokuskan pada parameter seperti latensi, kecepatan pemrosesan, ketersediaan sistem, serta kemampuan penanganan kegagalan. Hasil dari penelitian ini diharapkan dapat menjadi dasar pertimbangan dalam pemilihan arsitektur yang tepat untuk implementasi IoT skala besar.

TINJAUAN PUSTAKA

Internet of Things (IoT)

IoT adalah sistem perangkat yang saling terhubung lewat jaringan untuk bertukar data dan menjalankan fungsi secara otomatis. Dengan meningkatnya jumlah perangkat IoT, muncul kebutuhan pemrosesan data real-time dan manajemen sistem yang efisien. Dalam konteks ini, penggunaan platform cloud menjadi pilihan utama karena kapasitasnya yang besar dan kemudahan integrasi (Daharmi, Bhawiyuga, & Basuki, 2022). Sebagai contoh, dalam studi Pengembangan IoT Cloud Platform berbasis Serverless Computing menyajikan implementasi dan pengujian sistem IoT berbasis cloud di mana sistem mampu menangani permintaan hingga 82,42 permintaan per detik pada beban 150 permintaan secara simultan. Selain itu, implementasi IoT terus berkembang pada berbagai sektor, seperti pertanian, transportasi, dan sistem pemantauan lingkungan, yang memerlukan infrastruktur komunikasi andal dan pemrosesan data cepat (Selay et al., 2022).

Arsitektur khas IoT terdiri dari tiga lapisan utama yaitu perangkat (device layer), jaringan (network layer), dan aplikasi/layanan (application layer). Setiap lapisan menghadapi tantangan tersendiri mulai dari keterbatasan daya dan kemampuan komputasi perangkat di lapisan pertama, hingga latensi, keamanan, dan interoperabilitas pada lapisan jaringan dan aplikasi (Reyana, Kautish, Alnowibet, Zawbaa, & Wagdy Mohamed, 2023). Karena itu, desain sistem IoT yang efektif harus mempertimbangkan distribusi beban pemrosesan, efisiensi komunikasi, serta integrasi layanan untuk mendukung skenario real-time seperti pemantauan kesehatan atau kendaraan otonom.

Lebih lanjut, skala besar dan heterogenitas perangkat IoT menciptakan tantangan signifikan dalam pengumpulan dan manajemen data. Seperti yang dipaparkan oleh (Elgazzar et al., 2022), IoT merupakan paradigma yang menghubungkan miliaran perangkat daring yang pada gilirannya menghasilkan volume data yang besar dan memerlukan orkestrasi yang efisien antara edge, fog, dan cloud. Oleh karena itu, efisiensi pemrosesan dan kecepatan respon menjadi aspek krusial dalam aplikasi IoT, dan ini memperkuat relevansi penelitian kamu yang fokus pada latensi serta toleransi gangguan di lingkungan terdistribusi.

Di sisi lain, aspek keandalan sistem juga menjadi perhatian utama dalam implementasi IoT. Faktor seperti kegagalan perangkat, gangguan jaringan, atau konektivitas intermittent dapat secara langsung mempengaruhi kontinuitas layanan IoT. Studi recent menekankan bahwa meskipun konektivitas dan pemrosesan makin cepat, sistem IoT masih rentan terhadap celah keamanan, keandalan, dan manajemen perangkat skala besar. Maka dari itu, pemilihan arsitektur—antara edge/fog atau cloud serta strategi mitigasi kesalahan harus menjadi bagian integral dalam desain sistem IoT yang handal (Kim, Yoo, & Kim, 2023).

Jurnal Minfo Polgan
Volume 14, Nomor 2, Januari 2026
DOI : https://doi.org/10.33395/jmp.v14i2.15576
e-ISSN : 2797-3298
p-ISSN : 2089-9424

Dengan demikian, dalam konteks penelitian ini, IoT bukan hanya sebagai kumpulan perangkat terhubung, melainkan suatu ekosistem yang menuntut kecepatan, keandalan, dan skalabilitas. Penelitian yang membandingkan arsitektur seperti Fog Computing dan Cloud Computing sangat relevan untuk menjawab kebutuhan ini. Karena pada akhirnya, efisiensi pengiriman data (latensi) dan ketahanan terhadap kegagalan (toleransi kesalahan) menjadi dua parameter kunci bagi implementasi IoT yang sukses di dunia nyata.

Komputasi Awan (Cloud Computing)

Komputasi awan merupakan paradigma komputasi yang menyediakan sumber daya komputasi dan penyimpanan yang sangat besar (massive computing and storage capacity) secara terpusat, beroperasi dari pusat data global yang masif. Keunggulan utamanya mencakup skalabilitas tinggi dan kemudahan manajemen sumber daya yang memungkinkan alokasi dan de-alokasi sumber daya secara elastis sesuai dengan permintaan, sehingga sangat ideal untuk menangani beban kerja yang fluktuatif (Islam et al., 2025). Melalui infrastruktur yang terdistribusi di berbagai lokasi geografis, pengguna dapat memanfaatkan berbagai layanan komputasi, mulai dari Infrastructure as a Service (IaaS) hingga Software as a Service (SaaS), tanpa harus berinvestasi pada penyediaan dan pemeliharaan perangkat keras secara mandiri. Pendekatan ini menjadikan cloud sebagai solusi yang efisien dan hemat biaya untuk berbagai kebutuhan yang menuntut pemrosesan data berskala besar (big data analytics), penyimpanan arsip jangka panjang, serta integrasi layanan berbasis web yang tersebar luas.

Meski demikian, dalam konteks aplikasi Internet of Things (IoT) yang memiliki sifat sangat sensitif terhadap waktu (real-time), pendekatan cloud mengalami keterbatasan fundamental, terutama pada aspek latensi dan ketergantungan terhadap kestabilan jaringan (Mulyana, Haryanti, Pradipta, Terapan, & Telkom, 2021). Ketika volume data besar dari perangkat IoT (sensor, kamera, aktuator) harus dikirim melintasi jarak fisik yang jauh menuju pusat data terpusat untuk diproses, waktu tunda komunikasi (end-to-end latency) sering kali meningkat secara signifikan. Hal ini menjadikannya tidak ideal untuk sistem kritis yang memerlukan respons cepat dalam hitungan milidetik, seperti pada aplikasi pemantauan lingkungan proaktif, sistem kontrol industri otomatis, atau layanan kesehatan real-time. Di sisi lain, isu keandalan dan toleransi kesalahan pada cloud juga terus menjadi fokus penelitian; misalnya, studi “Fault Tolerance of Cloud Infrastructure with Machine Learning” menyoroti bagaimana penerapan machine learning dapat secara prediktif meningkatkan keandalan cloud dengan mendeteksi potensi kegagalan dan mengaktifkan mekanisme toleransi kesalahan untuk meminimalkan downtime sistem (Kalaskar & Thangam, 2023).

Secara ringkas, sejumlah studi menunjukkan bahwa meskipun Cloud Computing unggul dan tidak tertandingi dari sisi kapasitas pemrosesan dan skalabilitas global, arsitektur yang terpusat ini cenderung kurang efisien dalam mendukung aplikasi IoT terdistribusi dengan kebutuhan pemrosesan data secara lokal yang tinggi (Mulyana et al., 2021). Oleh karena itu, keterbatasan utama cloud—yakni latensi tinggi dan beban bandwidth jaringan—menjadi katalisator munculnya paradigma komputasi terdistribusi baru seperti Fog Computing dan Edge Computing yang dirancang khusus untuk mengatasi kelemahan tersebut dengan memindahkan fungsi komputasi ke tepi jaringan.

Komputasi Kabut (Fog Computing)

Fog computing adalah ekstensi dari paradigma komputasi awan yang membawa sebagian beban pemrosesan lebih dekat ke perangkat pengguna (edge), sehingga mengurangi latensi dan beban jaringan. Dalam artikel Fog Computing in Next Generation Networks: A Review (2024), penulis mengulas arsitektur, tantangan, dan peluang fog computing sebagai jembatan antara cloud dan perangkat IoT (Farhan & Setiaji, 2023).

Begitu pula dalam edisi khusus Edge/Fog Computing Technologies for IoT Infrastructure II (2023), teknologi fog/edge dibahas sebagai elemen penting dalam pembangunan infrastruktur IoT dengan arsitektur terdistribusi (Kim et al., 2023).

Selain itu, ulasan di A Decade of Research in Fog computing menyoroti bahwa meskipun

Jurnal Minfo Polgan
Volume 14, Nomor 2, Januari 2026
DOI : https://doi.org/10.33395/jmp.v14i2.15576
e-ISSN : 2797-3298
p-ISSN : 2089-9424

riset fog computing cukup pesat, implementasi skala besar masih terbatas dan beberapa tantangan teknis, seperti manajemen sumber daya, keamanan, dan reliabilitas, masih belum tuntas (Srirama, 2024).

Meski model pelayanan terpusat pada cloud memungkinkan pemanfaatan sumber daya komputasi secara efisien dan mudah diakses dari mana saja, riset terbaru menunjukkan bahwa perubahan skala dan sifat latensi-sensitif dari aplikasi IoT menuntut pendekatan yang lebih adaptif. (Al-dulaimy et al., 2024) mencatat bahwa dalam "continuum komputasi" dari IoT ke cloud, beban jaringan yang meningkat dan rute komunikasi yang panjang menyebabkan keterbatasan performa untuk aplikasi yang memerlukan waktu tanggap rendah. Selain itu, aspek keamanan dan privasi juga menjadi tantangan tersendiri dalam ekosistem cloud-IoT, di mana (Singh & Buyya, 2025) menunjukkan bahwa kepedulian terhadap pengelolaan data, tanggung-jawab pihak ketiga, dan gangguan jaringan semakin penting dalam lingkungan cloud berbasis IoT. Dengan demikian, meskipun cloud tetap menjadi tulang punggung banyak sistem, tekanan terhadap latensi, bandwidth, dan reliabilitas saat digunakan dalam skenario IoT real-time mengarahkan penelitian ke arsitektur yang lebih dekat ke perangkat pengguna.

Perbandingan Cloud dan Fog dalam Aplikasi IoT

Sebagian besar penelitian sebelumnya lebih berfokus pada analisis performa Cloud Computing atau Fog Computing secara terpisah, tanpa melakukan perbandingan langsung antara keduanya dalam konteks aplikasi Internet of Things (IoT). Pendekatan komputasi awan memang menawarkan skalabilitas tinggi dan kapasitas pemrosesan besar yang ideal untuk analisis data skala global dan penyimpanan data terpusat dalam jangka panjang (Kim et al., 2023). Namun, arsitektur terpusat Cloud memiliki kelemahan krusial pada aspek latensi dan ketergantungan terhadap koneksi jaringan yang tinggi, menjadikannya kurang optimal untuk aplikasi IoT yang sensitif terhadap waktu (time-sensitive) dan membutuhkan respons instan, seperti pada sistem kendaraan otonom, kendali industri, atau pemantauan kesehatan real-time (Zainudin, Anisah, & Gulo, 2021).

Di sisi lain, Komputasi Kabut (Fog Computing) mampu mengatasi masalah latensi ini dengan mendistribusikan tugas penyimpanan dan pemrosesan data lebih dekat ke sumber data (perangkat tepi). Distribusi komputasi pada lapisan Fog dapat meminimalkan latensi secara signifikan; misalnya, dalam studi kasus pengolahan data cuaca berbasis IoT, penggunaan Fog Computing mampu membuat delay rata-rata menjadi dua kali lebih cepat dibandingkan skenario tanpa Fog (Mulyana et al., 2021). Penerapan arsitektur Fog pada jaringan bahkan berpotensi menurunkan latensi hingga 70% untuk aplikasi yang menuntut layanan real-time (Zainudin et al., 2021) .Meskipun demikian, Komputasi Kabut juga menghadapi tantangan serius, terutama terkait keterbatasan kapasitas sumber daya pada Fog Node lokal serta kompleksitas dalam manajemen beban dan orkestrasi layanan yang terdistribusi.

Selain itu, beberapa studi terdahulu juga cenderung menggunakan simulasi atau skenario pengujian terbatas, sehingga hasil yang diperoleh belum sepenuhnya mencerminkan kondisi nyata pada sistem IoT berskala besar. Aspek penting lain yang sering terabaikan adalah toleransi kesalahan (fault tolerance) dan efisiensi energi yakni kemampuan sistem untuk tetap berfungsi saat terjadi gangguan jaringan atau kegagalan node, serta kemampuan sistem untuk mengoptimalkan penggunaan daya. Padahal, arsitektur Fog yang terdistribusi memiliki potensi besar untuk meningkatkan toleransi kesalahan, dan bahkan ditemukan dapat mengurangi pengeluaran energi rata-rata secara signifikan dibandingkan Cloud konvensional (Reyana et al., 2023). Kondisi ini menunjukkan perlunya penelitian yang tidak hanya menilai kinerja (latensi dan throughput), tetapi juga mengkaji ketahanan sistem terhadap kegagalan dan optimasi energi dalam implementasi hybrid cloud dan fog secara lebih mendalam dan realistis.

Penelitian Terkait

Berdasarkan penelitian yang telah dilakukan oleh (Suryadi et al., 2024) mengenai optimasi kinerja sistem IoT menggunakan teknik Edge Computing, yang mana studi tersebut telah menunjukkan bahwa pendekatan komputasi yang lebih dekat ke sumber data (seperti Edge atau Fog Computing) sangat efektif dalam mengoptimalkan waktu respon dan mengurangi beban

Jurnal Minfo Polgan
Volume 14, Nomor 2, Januari 2026
DOI : https://doi.org/10.33395/jmp.v14i2.15576
e-ISSN : 2797-3298
p-ISSN : 2089-9424

jaringan pada sistem IoT terdistribusi. Penelitian yang kami lakukan ini mengembangkan pendekatan konseptual dari studi (Suryadi et al., 2024) dengan membandingkan secara langsung arsitektur Fog Computing dan Cloud Computing, yang merupakan dua paradigma utama dalam infrastruktur IoT. Pengembangan ini bertujuan untuk mengukur secara komprehensif perbedaan kinerja kedua arsitektur tersebut, khususnya dari sisi latensi yang krusial untuk aplikasi real-time, dan yang lebih penting, menambahkan dimensi toleransi kesalahan (fault tolerance) saat terjadi gangguan broker dalam lingkungan IoT terdistribusi.

Penelitian oleh (Mulyana et al., 2021) juga melakukan analisis komparatif yang serupa, berfokus pada perbandingan kinerja antara Fog Computing dan Cloud Computing dalam konteks implementasi pengolahan data cuaca berbasis IoT. Hasil dari studi tersebut menunjukkan bahwa penggunaan Fog secara signifikan dapat meningkatkan kinerja waktu tunda (delay) komunikasi. Penelitian kami memperkuat temuan tersebut dengan melakukan pengujian dalam lingkungan terdistribusi (simulasi publisher–broker–consumer menggunakan protokol MQTT di lingkungan Docker) dan memperluas parameter perbandingan tidak hanya pada latency, tetapi juga pada toleransi kesalahan (fault tolerance) saat terjadi gangguan broker, yang merupakan aspek krusial untuk keandalan sistem IoT.

METODE PENELITIAN

Perangkat dan Lingkungan Uji

Penelitian ini dilaksanakan menggunakan lingkungan pengujian berbasis virtualisasi container dengan memanfaatkan Docker Desktop versi 4.33.0 pada sistem operasi Windows 11. Setiap komponen utama sistem IoT terdistribusi, yaitu publisher, broker, dan consumer, dijalankan sebagai container terpisah di dalam jaringan virtual Docker bernama iot-net. Bahasa pemrograman yang digunakan adalah Python 3.11 dengan pustaka paho-mqtt sebagai protokol komunikasi, sedangkan broker yang digunakan adalah Eclipse Mosquitto. Pengujian dilakukan pada laptop dengan spesifikasi prosesor Intel Core i5-12500H, GPU RTX 4050, dan RAM 16 GB, untuk memastikan bahwa hasil uji tidak terpengaruh oleh keterbatasan perangkat keras. Lingkungan ini dipilih karena dapat mensimulasikan sistem terdistribusi dengan kontrol penuh terhadap konfigurasi jaringan dan latensi antar node.

Skenario pengujian dilakukan dengan dua konfigurasi arsitektur, yaitu Cloud Computing dan Fog Computing. Pada skenario Cloud Computing, broker ditempatkan secara terpusat untuk merepresentasikan pemrosesan data di server pusat, sedangkan pada skenario Fog Computing, broker dijalankan di node yang lebih dekat dengan publisher untuk mensimulasikan pemrosesan di tepi jaringan (edge). Komunikasi antar node dilakukan melalui protokol MQTT dengan topik yang sama untuk memastikan kesetaraan skenario uji. Setiap hasil pengujian disimpan dalam format CSV dan Excel, kemudian dianalisis menggunakan pustaka Pandas dan Matplotlib di Python untuk menghitung parameter kinerja seperti latency, throughput, serta packet loss. Dengan pengaturan ini, lingkungan uji mampu merepresentasikan kondisi sistem IoT terdistribusi secara realistis namun tetap terkontrol.

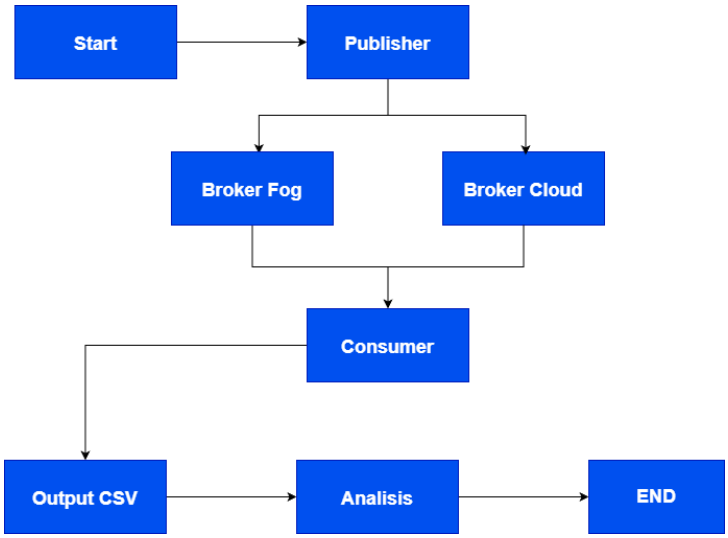
Desain dan Perancangan Sistem

Penelitian dimulai dengan perancangan arsitektur sistem IoT yang terdiri atas tiga komponen utama, yaitu publisher, broker, dan consumer. Komponen broker dibagi menjadi dua jenis, yaitu fog-broker dan cloud-broker, yang masing-masing mewakili arsitektur komputasi kabut dan komputasi awan. Setiap broker dijalankan menggunakan Docker container dengan konfigurasi jaringan yang sama (bridge network) agar komunikasi antar komponen berjalan stabil. Publisher berfungsi untuk mengirimkan data sensor secara periodik, sementara consumer bertugas menerima data dari broker dan mencatat waktu penerimaan untuk menghitung latency dan jitter. Meskipun pengujian dilakukan melalui simulasi berbasis Docker, arsitektur yang digunakan tetap merepresentasikan karakteristik dasar Cloud dan Fog Computing secara fungsional. Pendekatan ini memungkinkan analisis performa dilakukan dalam lingkungan yang terkontrol tanpa mengurangi validitas konseptual hasil penelitian.

Jurnal Minfo Polgan
Volume 14, Nomor 2, Januari 2026
DOI : https://doi.org/10.33395/jmp.v14i2.15576
e-ISSN : 2797-3298
p-ISSN : 2089-9424

Flowchart Penelitian

Flowchart pada Gambar 1 menunjukkan alur eksperimen penelitian ini, dimulai dari inisialisasi perangkat IoT, proses pengiriman data oleh publisher, pengelolaan pesan oleh broker (Fog dan Cloud), hingga penerimaan data oleh consumer serta pengukuran parameter latensi dan toleransi kesalahan.



Gambar 1. Flowchart Eksperimen Pengujian Fog dan Cloud Computing dalam Aplikasi IoT.

Flowchart pada Gambar di atas adalah alur kerja eksperimen yang dilakukan dalam penelitian ini. Setiap tahapan memiliki fungsi spesifik sebagai berikut:

1. Start

Tahap awal dimulai dengan inisialisasi seluruh komponen sistem, termasuk konfigurasi jaringan Docker, pembuatan container publisher, broker, dan consumer, serta penentuan parameter pengujian seperti jumlah pesan dan interval waktu pengiriman.

2. Publisher

Publisher bertugas mengirimkan data simulasi IoT dalam format JSON yang berisi ID pesan dan waktu pengiriman. Data ini dikirim secara periodik melalui protokol MQTT menuju broker. Proses ini merepresentasikan aktivitas sensor IoT yang secara rutin mengirimkan data ke sistem pusat.

3. Broker Fog dan Broker Cloud

Dua broker dijalankan untuk membandingkan dua arsitektur:

- Broker Fog dijalankan di node lokal (port 1885) untuk merepresentasikan pemrosesan di tepi jaringan (edge).
- Broker Cloud dijalankan secara terpusat (port 1884) sebagai simulasi server cloud. Keduanya berperan dalam menerima pesan dari publisher, menyimpannya sementara, dan meneruskannya ke consumer.

4. Consumer

Komponen ini berfungsi sebagai penerima data dari broker. Consumer mencatat waktu penerimaan tiap pesan dan menghitung latency berdasarkan perbedaan waktu kirim dan waktu terima. Data hasil penerimaan disimpan dalam file CSV untuk dianalisis.

5. Output CSV

Semua hasil pengujian otomatis direkam dalam dua file, yaitu results_fog.csv dan results_cloud.csv. File ini berisi data mentah hasil pengujian dari masing-masing arsitektur yang akan digunakan sebagai dasar analisis performa.

6. Analisis Data

Data dari file CSV kemudian diproses menggunakan skrip Python (analyze_latency.py). Analisis ini menghasilkan metrik seperti rata-rata latency, nilai maksimum–minimum, dan jitter, yang

Jurnal Minfo Polgan
Volume 14, Nomor 2, Januari 2026
DOI : https://doi.org/10.33395/jmp.v14i2.15576
e-ISSN : 2797-3298
p-ISSN : 2089-9424

menjadi parameter utama dalam evaluasi sistem.

7. End

Tahap akhir menandakan berakhirnya proses pengujian. Semua hasil pengukuran direkap, dibandingkan, dan diinterpretasikan untuk menentukan perbedaan kinerja antara arsitektur Fog Computing dan Cloud Computing.

Implementasi Sistem

Implementasi dilakukan menggunakan bahasa pemrograman Python dengan library paho-mqtt sebagai protokol komunikasi antar perangkat. Dua file utama digunakan, publisher.py, untuk mengirimkan data JSON berisi ID pesan, waktu pengiriman, dan identitas publisher. consumer.py, untuk menerima pesan dari broker, menghitung waktu tunda (latency), serta menyimpan hasil ke file CSV.

Broker fog dijalankan pada port 1885, sedangkan broker cloud menggunakan port 1884. Pengujian dilakukan dengan menjalankan publisher dan consumer secara bersamaan dalam jaringan Docker yang sama untuk memastikan konektivitas langsung.

Pengujian Kinerja (Latency)

Pengujian ini dilakukan untuk membandingkan rata-rata waktu tunda (latency) antara Fog Computing dan Cloud Computing dalam proses komunikasi data IoT. Simulasi dilakukan menggunakan MQTT dengan publisher–broker–consumer yang berjalan di Docker. Broker fog berjalan pada port 1885, sedangkan broker cloud berjalan pada port 1884. Publisher mengirimkan 60 pesan JSON berisi waktu pengiriman (t_pub) setiap 1 detik ke topik sensors/data. Consumer menerima setiap pesan dan mencatat waktu penerimaan (t_recv). Latency dihitung dengan rumus:

$$Latency = t_{recv} - t_{pub}$$

Data hasil pengukuran disimpan dalam dua berkas, results_fog.csv dan results_cloud.csv, yang kemudian dianalisis untuk memperoleh nilai rata-rata (mean latency), nilai maksimum dan minimum, serta variasi waktu tunda (jitter). Analisis ini bertujuan untuk menggambarkan kestabilan komunikasi data pada masing-masing arsitektur, sekaligus menilai seberapa besar perbedaan performa antara Fog Computing dan Cloud Computing dalam proses transmisi data IoT.

Pengujian Toleransi Kesalahan

Pengujian ini bertujuan untuk mengetahui kemampuan sistem dalam menghadapi gangguan jaringan (fault tolerance) pada arsitektur Fog dan Cloud Computing. Simulasi gangguan dilakukan dengan menghentikan layanan broker (Mosquitto) di tengah proses pengiriman data, tepatnya saat pesan ke-12 dari total 60 pesan yang dikirim oleh publisher. Pada kondisi tersebut, consumer otomatis berhenti menerima pesan baru hingga broker kembali aktif. Jumlah pesan yang berhasil diterima sebelum dan sesudah gangguan dibandingkan dengan total pesan yang dikirim untuk menghitung tingkat kehilangan data (packet loss). Nilai packet loss dihitung menggunakan perbandingan antara jumlah pesan yang hilang dengan total pesan yang dikirim dengan rumus:

$$PacketLoss(\%) = \frac{(Total\ Dikirim - Total\ Diterima)}{Total\ Dikirim}\ X\ 100\%$$

Hasil uji dilakukan baik pada lingkungan Fog maupun Cloud untuk melihat perbedaan ketahanan sistem terhadap gangguan jaringan atau broker.

Analisis Data

Analisis hasil dilakukan dengan membandingkan rata-rata latency, jitter, maksimum, dan

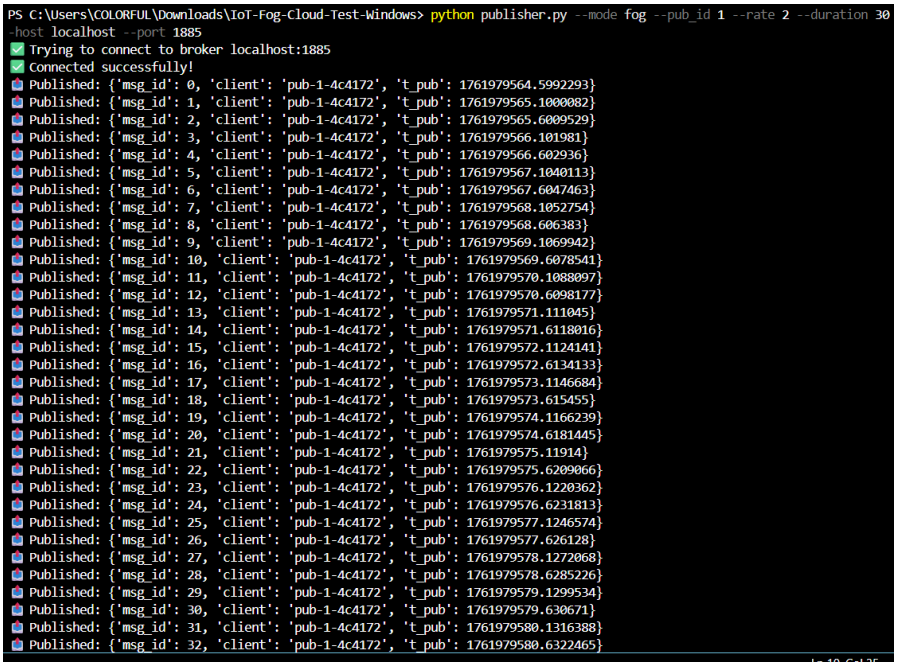
Jurnal Minfo Polgan
Volume 14, Nomor 2, Januari 2026
DOI : https://doi.org/10.33395/jmp.v14i2.15576
e-ISSN : 2797-3298
p-ISSN : 2089-9424

minimum dari kedua lingkungan. Selain itu, jumlah pesan yang hilang pada saat simulasi gangguan digunakan sebagai ukuran toleransi kesalahan. Nilai-nilai tersebut diinterpretasikan untuk menunjukkan keunggulan masing-masing arsitektur: Fog computing diharapkan unggul dalam hal latensi dan stabilitas, sementara Cloud computing memiliki keunggulan dalam skala dan kapasitas pemrosesan data.

HASIL DAN PEMBAHASAN

Implementasi dan Progres Pengujian

Proses implementasi dan pengujian dilakukan menggunakan Visual Studio Code dan Docker Compose sebagai lingkungan eksekusi. Tahapan pengujian terdiri atas dua komponen utama, yaitu publisher yang berfungsi mengirimkan data simulasi IoT, dan consumer yang berperan menerima serta mencatat data tersebut melalui koneksi broker MQTT. Gambar 2 memperlihatkan tampilan saat publisher berhasil mengirimkan data ke broker Fog Computing pada port 1885. Setiap pesan dikirim secara berurutan dengan waktu jeda tertentu sesuai pengaturan laju pengiriman data.

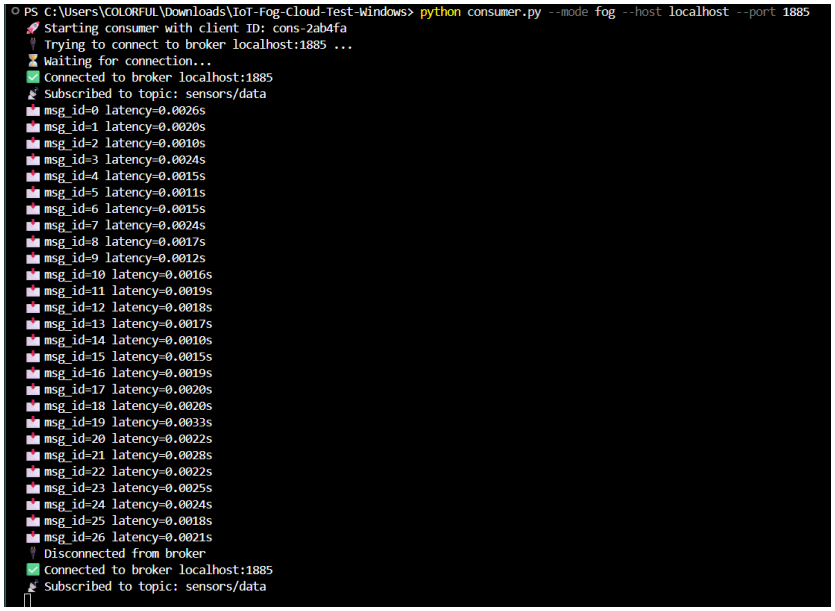


Gambar 2. Tampilan publisher saat mengirimkan data ke broker Fog Computing.

Gambar 2 menunjukkan tampilan terminal saat publisher berhasil mengirimkan data ke broker Fog Computing melalui port 1885. Terlihat bahwa proses pengiriman berlangsung secara berurutan dengan kecepatan sesuai parameter pengujian.

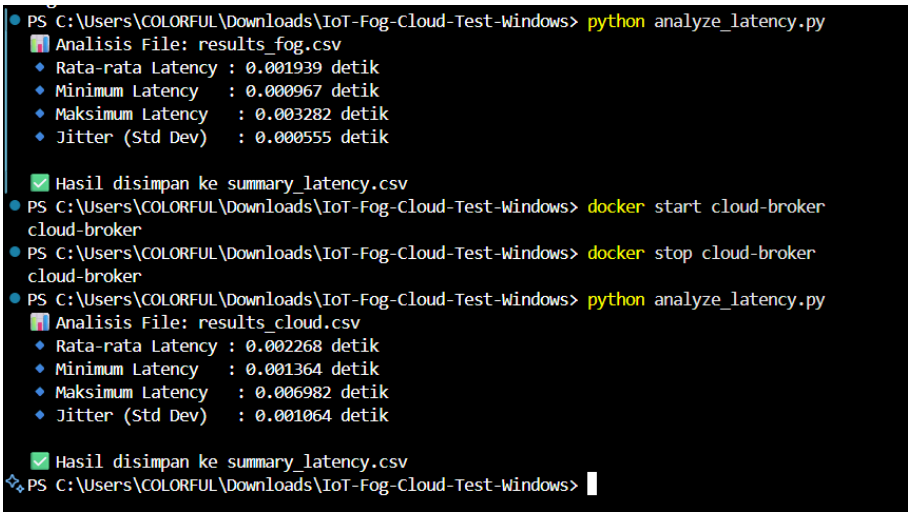
Selanjutnya, Gambar 3 menunjukkan tampilan terminal consumer yang terhubung ke broker Fog melalui port 1885. Pada terminal tersebut terlihat proses penerimaan data secara real-time dari publisher, ditandai dengan output berisi ID pesan (msg_id) dan nilai latency setiap paket yang diterima. Data hasil penerimaan ini kemudian disimpan secara otomatis ke dalam file hasil (results_fog.csv).

Jurnal Minfo Polgan
Volume 14, Nomor 2, Januari 2026
DOI : https://doi.org/10.33395/jmp.v14i2.15576
e-ISSN : 2797-3298
p-ISSN : 2089-9424



Gambar 3. Tampilan consumer saat menerima data dari broker Fog Computing.

Selanjutnya, dilakukan analisis hasil pengujian menggunakan skrip analyze_latency.py untuk menghitung rata-rata, minimum, maksimum, dan jitter dari hasil pengiriman data pada kedua lingkungan. Gambar 4 menampilkan hasil analisis kinerja Fog dan Cloud Computing di terminal Visual Studio Code menggunakan skrip analyze_latency.py, yang menunjukkan perbedaan nilai latency dan jitter dari masing-masing lingkungan.

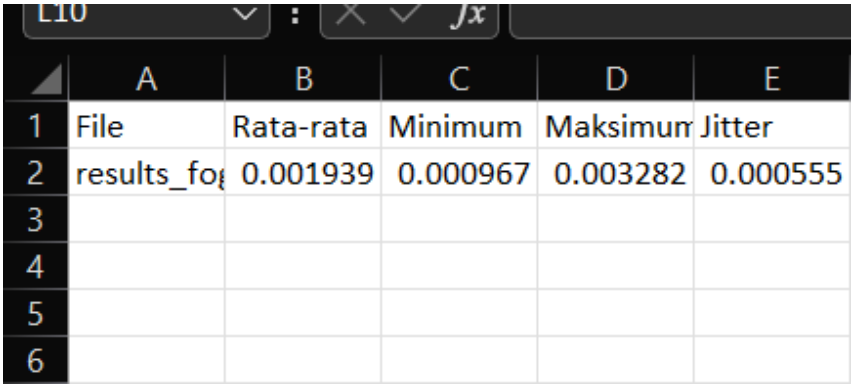


Gambar 4. Tampilan consumer saat menerima data dari broker Fog Computing

Berdasarkan hasil analisis di terminal pada Gambar 4, terlihat bahwa sistem berhasil menghasilkan metrik performa dari masing-masing arsitektur. Nilai-nilai tersebut kemudian direkap secara otomatis ke dalam file summary_latency_fog.csv dan summary_latency_cloud.csv untuk dianalisis lebih lanjut.

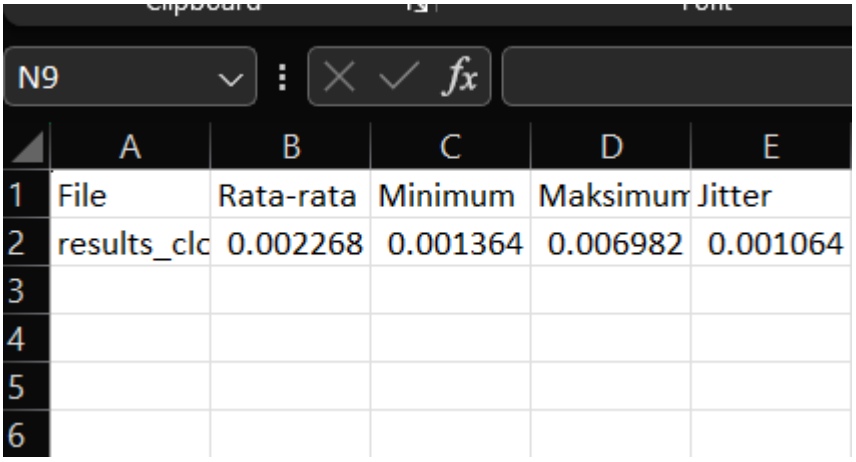
Lalu Gambar ke 5 dibawah menunjukkan hasil analisis Fog Computing yang tersimpan dalam file summary_latency_fog.csv, sedangkan Gambar 6 menampilkan hasil analisis Cloud Computing pada file summary_latency_cloud.csv. Kedua hasil ini menjadi dasar dalam perbandingan kinerja antara arsitektur Fog dan Cloud.

Jurnal Minfo Polgan
Volume 14, Nomor 2, Januari 2026
DOI : https://doi.org/10.33395/jmp.v14i2.15576
e-ISSN : 2797-3298
p-ISSN : 2089-9424



	A	B	C	D	E
1	File	Rata-rata	Minimum	Maksimum	Jitter
2	results_fo	0.001939	0.000967	0.003282	0.000555
3					
4					
5					
6					

Gambar 5. File Hasil analisis latency lingkungan Fog Computing.



	A	B	C	D	E
1	File	Rata-rata	Minimum	Maksimum	Jitter
2	results_clc	0.002268	0.001364	0.006982	0.001064
3					
4					
5					
6					

Gambar 6. File Hasil analisis latency lingkungan Cloud Computing.

Berdasarkan hasil pada Gambar 5 dan Gambar 6, nilai rata-rata latency pada Fog Computing adalah 0.001939 detik dengan jitter 0.000555 detik, sedangkan pada Cloud Computing sebesar 0.002268 detik dengan jitter 0.001064 detik. Nilai ini memperkuat bahwa Fog Computing memiliki waktu tanggap lebih cepat dan variasi delay lebih rendah dibandingkan Cloud Computing.

Hasil Uji Kinerja (Latency)

Hasil Pengujian kinerja menunjukkan bahwa Fog Computing memiliki waktu tunda (latency) lebih rendah dibandingkan Cloud Computing. Selama pengiriman 60 pesan melalui protokol MQTT, rata-rata latency pada Fog sebesar 0.0019 detik, sedangkan pada Cloud sebesar 0.0022 detik. Perbedaan ini disebabkan oleh lokasi pemrosesan data pada Fog yang lebih dekat ke sumbernya, sehingga waktu propagasi dan beban jaringan lebih kecil.

Tabel 1: Perbandingan Kinerja Sistem Fog Computing Dan Cloud Computing

Lingkungan	Rata-rata Latency (s)	Maksimum (s)	Minimum (s)	Jitter (s)
Fog	0.001939	0.000967	0.003282	0.000555
Cloud	0.002268	0.001364	0.006982	0.001064

Dari hasil tersebut terlihat bahwa Fog Computing memberikan waktu tanggap lebih cepat dibandingkan Cloud Computing karena proses komunikasi terjadi lebih dekat dengan perangkat IoT. Sebaliknya, arsitektur Cloud memerlukan jalur komunikasi lebih panjang yang meningkatkan latensi dan jitter. Temuan bahwa Fog unggul dalam latensi (0.0019 s dibanding 0.0022 s) sejalan dengan penelitian (Suryadi et al., 2024), yang menunjukkan bahwa pendekatan komputasi dekat sumber data, seperti Edge atau Fog Computing, mampu mengoptimalkan waktu respon dan

Jurnal Minfo Polgan
Volume 14, Nomor 2, Januari 2026
DOI : https://doi.org/10.33395/jmp.v14i2.15576
e-ISSN : 2797-3298
p-ISSN : 2089-9424

mengurangi beban jaringan pada sistem IoT terdistribusi.

Hasil Uji Toleransi Kesalahan

Hasil uji toleransi kesalahan menunjukkan bahwa kedua arsitektur, baik Fog Computing maupun Cloud Computing, mengalami kehilangan pesan akibat gangguan broker, dengan tingkat kehilangan yang hampir sebanding. Meskipun Cloud Computing sedikit lebih stabil dalam mempertahankan koneksi karena manajemen broker yang terpusat, Fog Computing menunjukkan potensi pemulihan yang lebih cepat berkat pemrosesan yang lebih dekat dengan sumber data. Kondisi ini menegaskan bahwa Fog Computing lebih tangguh untuk aplikasi yang menuntut respons real-time, sementara Cloud Computing lebih cocok untuk skenario yang memerlukan keandalan transmisi data secara berkelanjutan.

Tabel 2: Simulasi Gangguan Sistem Fog Computing Dan Cloud Computing

Lingkungan	Pesan Dikirim	Pesan Diterima	Kehilangan Pesan	Packet Loss(%)
Fog	60	26	34	56.67%
Cloud	60	27	33	55.00%

Kedua arsitektur menunjukkan kerentanan terhadap kegagalan broker, tetapi Cloud sedikit lebih stabil dalam mempertahankan koneksi karena manajemen broker yang terpusat. Meski demikian, Fog Computing memiliki waktu pemulihan lebih cepat dan tetap unggul dari sisi latency, sehingga lebih sesuai untuk aplikasi IoT yang membutuhkan respons real-time meskipun toleransi kesalahannya sedikit lebih rendah.

Peningkatan Kinerja Melalui Pengurangan Latensi

Hasil penelitian menunjukkan bahwa arsitektur Fog Computing secara signifikan menurunkan nilai latensi dibandingkan Cloud Computing. Hal ini disebabkan oleh mekanisme pemrosesan data yang dilakukan di node yang lebih dekat dengan sumber data IoT, sehingga waktu propagasi data ke server menjadi lebih singkat. Pengurangan latensi ini sangat penting dalam aplikasi IoT real-time seperti sistem pemantauan lalu lintas, pemrosesan video, atau pengendalian perangkat industri yang membutuhkan respon cepat. Dengan demikian, pendekatan Fog Computing memberikan peningkatan kinerja pada sisi kecepatan respon sistem dan mendukung kebutuhan real-time decision making dalam lingkungan IoT.

Analisis Toleransi Kesalahan

Pada pengujian gangguan, ketika broker Fog maupun Cloud dihentikan secara mendadak, jumlah pesan yang berhasil diterima oleh konsumen menurun signifikan. Meskipun keduanya mengalami kehilangan pesan, Cloud Computing menunjukkan tingkat kehilangan pesan (packet loss) sedikit lebih rendah (55,00%) dibanding Fog (56,67%). Hal ini menandakan bahwa sistem cloud memiliki manajemen koneksi dan message buffering yang lebih stabil, sehingga lebih toleran terhadap kegagalan sementara. Namun, fog memiliki keunggulan dalam pemulihan cepat (quick recovery) ketika koneksi kembali aktif karena topologinya yang lebih dekat dengan edge devices.

Tantangan dan Peluang Implementasi Fog Computing

Tantangan dan Peluang Implementasi Fog Computing Meskipun hasil menunjukkan keunggulan fog dalam aspek latensi, tantangan utama terletak pada keterbatasan sumber daya komputasi dan kompleksitas manajemen node. Setiap node fog memerlukan pengaturan keamanan, sinkronisasi data, serta pemantauan performa agar tidak menimbulkan inkonsistensi.

Sebaliknya, Cloud masih menjadi pilihan utama untuk pengelolaan data besar dan analisis jangka panjang. Oleh karena itu, pendekatan hybrid antara Cloud dan Fog Computing berpotensi menjadi solusi optimal, di mana fog menangani proses real-time sementara cloud bertanggung jawab pada penyimpanan dan analitik berskala besar (Nurcahya, Karimah, & Mugitama, 2023)

Jurnal Minfo Polgan
Volume 14, Nomor 2, Januari 2026
DOI : https://doi.org/10.33395/jmp.v14i2.15576
e-ISSN : 2797-3298
p-ISSN : 2089-9424

Batasan Studi dan Ancaman terhadap Validitas

Penelitian ini memiliki beberapa batasan yang dapat memengaruhi generalisasi hasil. Pertama, pengujian dilakukan dalam lingkungan simulasi berbasis Docker yang terkontrol, sehingga kondisi jaringan, beban trafik, dan jumlah node belum sepenuhnya mencerminkan lingkungan IoT berskala besar di dunia nyata. Kedua, parameter kinerja yang diuji hanya berfokus pada latensi dan toleransi kesalahan, tanpa mempertimbangkan konsumsi energi dan bandwidth yang juga berperan penting dalam implementasi IoT. Ketiga, gangguan jaringan disimulasikan secara manual dengan penghentian broker, yang mungkin berbeda dari kegagalan sistem alami di lapangan. Meskipun demikian, rancangan uji ini tetap memberikan gambaran yang valid mengenai perbedaan mendasar antara arsitektur Cloud dan Fog dalam konteks kinerja dan ketahanan sistem.

KESIMPULAN

Penelitian ini membandingkan kinerja dan toleransi kesalahan antara arsitektur Fog Computing dan Cloud Computing dalam lingkungan IoT terdistribusi. Hasil pengujian menunjukkan bahwa Fog Computing memiliki latensi rata-rata lebih rendah (0.0019 detik) dibandingkan Cloud Computing (0.0023 detik), yang menunjukkan keunggulan fog dalam pemrosesan data real-time karena kedekatannya dengan sumber data. Namun, pada uji toleransi kesalahan, Fog Computing kehilangan 34 pesan (56,67%) sedangkan Cloud Computing kehilangan 33 pesan (55%) dari total 60 pesan ketika terjadi gangguan broker. Hal ini menunjukkan bahwa kedua arsitektur memiliki tingkat ketahanan yang relatif sebanding, meskipun cloud sedikit lebih stabil dalam kondisi gangguan. Secara keseluruhan, penelitian ini menegaskan bahwa Fog Computing lebih unggul dalam efisiensi waktu respon, sementara Cloud Computing lebih konsisten dalam penanganan gangguan, sehingga penerapan pendekatan hybrid antara keduanya dapat menjadi solusi optimal untuk aplikasi IoT yang membutuhkan keseimbangan antara kinerja dan keandalan sistem. Penelitian selanjutnya disarankan untuk memperluas pengujian dengan mempertimbangkan parameter lain seperti penggunaan bandwidth, konsumsi energi, dan skenario multi-node untuk memperoleh gambaran yang lebih komprehensif terhadap performa sistem IoT terdistribusi.

REFERENSI

Al-dulaimy, A., Jansen, M., Johansson, B., Trivedi, A., Iosup, A., Ashjaei, M., … Papadopoulos, A. V. (2024). Internet of Things The computing continuum : From IoT to the cloud. *Internet of Things*, *27*(January), 101272. https://doi.org/10.1016/j.iot.2024.101272

Daharmi, R. R., Bhawiyuga, A., & Basuki, A. (2022). Pengembangan IoT Cloud Platform berbasis pada Layanan Serverless Computing. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *6*(12), 5905–5914.

Elektro, F. T., & Telkom, U. (2020). *Migrasi Mikroservis pada Fog Computing untuk Mendukung Kinerja Komputasi Robot dengan Cakupan Pergerakan yang Luas*.

Elgazzar, K., Khalil, H., Alghamdi, T., Badr, A., Abdelkader, G., Elewah, A., & Buyya, R. (2022). *Revisiting the internet of things : New trends , opportunities and grand challenges*. (November), 1–18. https://doi.org/10.3389/friot.2022.1073780

Farhan, N. M., & Setiaji, B. (2023). Indonesian Journal of Computer Science. *Indonesian Journal of Computer Science*, *12*(2), 284–301. Retrieved from http://ijcs.stmikindonesia.ac.id/ijcs/index.php/ijcs/article/view/3135

Ilahi, E. N., Saripudin, M., Nughraha, M. A., Dwight, G., Cardinsyah, A., Hikmatullo, M. F., & Encep, M. (2024). *Mengungkap Potensi Luar Biasa dan Tantangan Menantang Cloud Computing di Era Digital*. *3*, 2197–2206.

Islam, I. N., Setiawan, A. D., Jaenul, A., Mulyono, S., Hertin, D., Wibowo, A. S., … Wibisono, S. K. (2025). *Analisis AWS , Microsoft Azure , dan GCP terhadap Latency , Throughput , IAM , dan Firewall*. *6*(2), 93–97.

Kalaskar, C., & Thangam, S. (2023). Fault Tolerance of Cloud Infrastructure with Machine Learning. *Cybernetics and Information Technologies*, *23*(4), 26–50. https://doi.org/10.2478/cait-2023-0034

Kim, T., Yoo, S. E., & Kim, Y. (2023). Edge/Fog Computing Technologies for IoT Infrastructure II. *Sensors*, *23*(8), 2–4. https://doi.org/10.3390/s23083953

Mulyana, A., Haryanti, T., Pradipta, R. F., Terapan, F. I., & Telkom, U. (2021). *ANALISIS KOMPARASI FOG COMPUTING - CLOUD COMPUTING DALAM IMPLEMENTASI PENGOLAHAN DATA CUACA BERBASIS IOT COMPARISON ANALYSIS OF FOG COMPUTING AND CLOUD COMPUTING IN CONCERN DATA PROCESSING OF WEATHER*. *8*(2), 1127–1137.

Nurcahya, D., Karimah, S. A., & Mugitama, S. A. (2023). *Performance Analysis of Scheduling Algorithms on Fog Computing using YAFS*. *7*(3), 1677–1686.

Reyana, A., Kautish, S., Alnowibet, K. A., Zawbaa, H. M., & Wagdy Mohamed, A. (2023). Opportunities of IoT in Fog Computing for High Fault Tolerance and Sustainable Energy Optimization. *Sustainability (Switzerland)*, *15*(11). https://doi.org/10.3390/su15118702

Selay, A., Andgha, G. D., Alfarizi, M. A., Bintang, M. I., Falah, M. N., Khaira, M., & Encep, M. (2022). *Karimah Tauhid, Volume 1 Nomor 6 (2022), e-ISSN 2963-590X*. *1*, 860–868.

Singh, N., & Buyya, R. (2025). *Securing Cloud-Based Internet of Things : Challenges and Mitigations*. 1–45.

Srirama, S. N. (2024). A decade of research in fog computing: Relevance, challenges, and future directions. *Software - Practice and Experience*, *54*(1), 3–23. https://doi.org/10.1002/spe.3243

Suryadi, D., Octiva, C. S., Fajri, T. I., Nuryanto, U. W., & Hakim, M. L. (2024). Optimasi Kinerja Sistem IoT Menggunakan Teknik Edge Computing. *Jurnal Minfo Polgan*, *13*(2), 1456–1461. https://doi.org/10.33395/jmp.v13i2.14102

Zainudin, A., Anisah, I., & Gulo, M. M. (2021). *IMPLEMENTASI FOG COMPUTING PADA APLIKASI SMART HOME BERBASIS INTERNET OF THINGS*. *6*(1), 127–132.