

Analisis Latency Edge Computing dan Cloud Computing pada Monitoring IPAL Berbasis IoT

Eneng Susilistia Agustin¹, Eva Hendrawati²

^{1,2}, Program Studi Sistem Komputer - Universitas Pamulang Kampus Serang
Jl. Raya Serang-Jakarta, Kelodran, Walantaka, Serang, Banten - Indonesia

¹ dosen10009@unpam.ac.id, ² dosen10014@unpam.ac.id

Abstrak - Sistem monitoring kualitas air pada Instalasi Pengolahan Air Limbah (IPAL) berbasis Internet of Things (IoT) membutuhkan waktu respons yang rendah agar perubahan kondisi air dapat diketahui secara cepat. Penelitian ini bertujuan menganalisis dan membandingkan latency pada arsitektur Cloud Computing dan Edge Computing dalam sistem monitoring IPAL berbasis IoT. Sistem menggunakan sensor pH, suhu DS18B20, dan turbidity yang terhubung ke ESP32, sedangkan Raspberry Pi digunakan sebagai perangkat edge untuk melakukan pemrosesan data secara lokal. Pengujian dilakukan menggunakan 15 sampel yang terdiri atas air bersih, air limbah sabun, dan air limbah rumah tangga. Hasil pengujian menunjukkan bahwa rata-rata latency Cloud Computing sebesar 146,07 ms, dengan nilai minimum 138 ms dan maksimum 155 ms, sedangkan Edge Computing menghasilkan rata-rata latency sebesar 61,00 ms, dengan nilai minimum 58 ms dan maksimum 64 ms. Penerapan Edge Computing menurunkan latency sebesar 85,07 ms atau 58,24% dibandingkan Cloud Computing. Hasil tersebut menunjukkan bahwa pemrosesan data lebih dekat dengan sumber data dapat memperpendek jalur komunikasi dan mempercepat respons sistem monitoring IPAL. Temuan ini menjadi dasar bagi pengembangan sistem IPAL berbasis Edge AI pada penelitian lanjutan.

Kata Kunci - Edge Computing, Cloud Computing, Internet of Things, IPAL, latency.

Abstract - Internet of Things (IoT)-based wastewater treatment plant (WWTP) monitoring system requires low response time so that changes in water conditions can be detected quickly. This study aims to analyze and compare latency in Cloud Computing and Edge Computing architectures for an IoT-based WWTP monitoring system. The system uses pH, DS18B20 temperature, and turbidity sensors connected to an ESP32, while a Raspberry Pi serves as the edge device for local data processing. Testing was conducted using 15 samples consisting of clean water, soapy wastewater, and household wastewater. The results show that Cloud Computing achieved an average latency of 146.07 ms, with a minimum of 138 ms and a maximum of 155 ms, whereas Edge Computing achieved an average latency of 61.00 ms, with a minimum of 58 ms and a maximum of 64 ms. The implementation of Edge Computing reduced latency by 85.07 ms or 58.24% compared with Cloud Computing. These results indicate that processing data closer to the data source can shorten the communication path and improve the response time of the WWTP monitoring system. The findings provide a basis for further development of an Edge AI-based WWTP system.

Keywords - Edge Computing, Cloud Computing, Internet of Things, IPAL, latency.

1. PENDAHULUAN

Perkembangan penelitian beberapa tahun terakhir menunjukkan bahwa optimasi arsitektur komputasi menjadi salah satu pendekatan untuk meningkatkan performa sistem monitoring berbasis IoT. Ren mengembangkan sistem monitoring kualitas air dengan menerapkan *Edge Computing* pada stasiun monitoring sehingga sebagian proses pengolahan data dilakukan lebih dekat dengan sumber data untuk mendukung sistem peringatan dini. Penelitian tersebut menunjukkan bahwa pemrosesan pada perangkat *edge* dapat meningkatkan efisiensi pengelolaan data monitoring kualitas air, namun belum berfokus pada pengukuran perbandingan *latency* antara arsitektur *cloud* dan *edge*.

Instalasi Pengolahan Air Limbah (IPAL) merupakan salah satu infrastruktur penting dalam menjaga kualitas lingkungan melalui proses pengolahan air limbah sebelum dialirkan kembali ke lingkungan. Dalam proses operasionalnya, perubahan parameter kualitas air seperti derajat keasaman (pH), suhu, dan tingkat kekeruhan (*turbidity*) perlu dipantau secara berkala agar kondisi air dapat diketahui secara cepat dan mendukung pengambilan keputusan secara tepat. Pemanfaatan *Internet of Things* (IoT) memungkinkan proses akuisisi data sensor dilakukan secara *real-time* sehingga informasi kualitas air dapat ditampilkan melalui sistem monitoring secara berkelanjutan. Penelitian Indarto dan Agustini telah mengembangkan sistem monitoring IPAL berbasis IoT menggunakan ESP32 dengan sensor pH, *turbidity*, dan suhu untuk mendukung

pemantauan kualitas air secara *real-time*. Namun, sistem monitoring berbasis IoT masih memiliki tantangan terhadap waktu responss karena proses pengiriman dan pengolahan data dapat menimbulkan waktu tunda (*latency*) pada komunikasi data.

Selanjutnya, penelitian *IoT-Edge Computing* untuk monitoring lingkungan menunjukkan bahwa penerapan *edge* mampu mengurangi *latency* dibandingkan arsitektur IoT konvensional melalui pemrosesan data yang dilakukan lebih dekat dengan sensor.

Penelitian lainnya dilakukan oleh Shahra dkk. yang mengembangkan kerangka *edge-cloud* untuk monitoring kualitas air dengan membandingkan beberapa skenario arsitektur, yaitu *Cloud Computing*, *Edge Computing*, dan *Hybrid Edge-Cloud*. Penelitian tersebut menunjukkan bahwa pemilihan lokasi pemrosesan data menjadi faktor penting dalam meningkatkan performa sistem monitoring kualitas air. Selain itu, penelitian mengenai penerapan *machine learning* pada perangkat *edge* berbiaya rendah menggunakan ESP32 menunjukkan bahwa model kecerdasan buatan dapat dijalankan secara langsung pada perangkat dengan sumber daya terbatas untuk mendukung prediksi kualitas air secara *real-time*. Sementara itu, penelitian terbaru mengenai sistem monitoring kualitas air berbasis ESP32 masih banyak memanfaatkan layanan *cloud* sebagai pusat penyimpanan dan visualisasi data, sehingga komunikasi dengan server tetap menjadi bagian penting dalam proses monitoring.

Berdasarkan penelitian-penelitian tersebut dapat diketahui bahwa pengembangan sistem monitoring kualitas air telah mengalami kemajuan dari sisi akuisisi data sensor, integrasi IoT, penerapan *Cloud Computing*, hingga implementasi *Edge Computing* dan kecerdasan buatan pada perangkat *edge*. Meskipun demikian, masih terdapat celah penelitian (*edge-cloud framework*) yang menunjukkan bahwa sebagian penelitian berfokus pada pengembangan sistem monitoring, sebagian menitikberatkan pada implementasi *edge-cloud framework*, sedangkan penelitian lainnya mengevaluasi performa model *machine learning* pada perangkat *edge*. Belum banyak penelitian yang secara khusus melakukan pengukuran kuantitatif terhadap perbandingan *latency* antara arsitektur *Cloud Computing* dan *Edge Computing* pada sistem monitoring IPAL berbasis IoT dengan menggunakan parameter kualitas air sebagai masukan sistem monitoring.

Untuk mengisi celah penelitian tersebut, penelitian ini menerapkan arsitektur *Edge Computing* pada sistem monitoring IPAL berbasis IoT dengan memanfaatkan ESP32 sebagai perangkat akuisisi data sensor dan Raspberry Pi sebagai perangkat pemrosesan lokal (*edge device*). Data yang diperoleh dari sensor pH, suhu DS18B20, dan *turbidity* dikirim melalui protokol MQTT menuju Raspberry Pi untuk diproses secara lokal sebelum hasil monitoring diteruskan ke *edge-cloud framework*. Sebagai pembanding, dilakukan pula pengujian menggunakan arsitektur *Cloud Computing* sehingga perbedaan waktu responss pada kedua arsitektur dapat diukur secara objektif berdasarkan nilai *latency*.

Pengujian dilakukan menggunakan 15 sampel yang terdiri atas air bersih, air limbah sabun, dan air limbah rumah tangga sebagai data masukan sistem monitoring. Fokus utama penelitian bukan pada pengembangan sensor maupun pembangunan sistem klasifikasi, melainkan pada analisis performa waktu responss sistem melalui pengukuran *latency* pada arsitektur *Cloud Computing* dan *Edge Computing*. Dengan demikian, penelitian ini memberikan kontribusi berupa evaluasi empiris mengenai pengaruh lokasi pemrosesan data terhadap kecepatan responss sistem monitoring IPAL berbasis IoT.

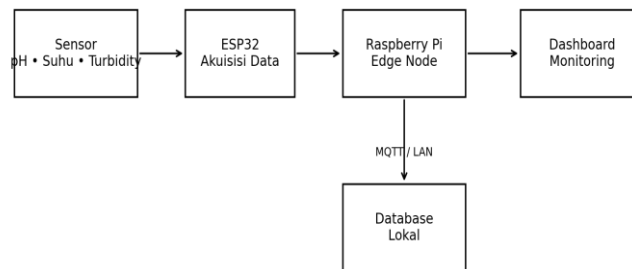
Tujuan penelitian ini adalah menganalisis dan membandingkan nilai *latency* pada arsitektur *Cloud Computing* dan *Edge Computing* pada sistem monitoring IPAL berbasis IoT serta menghitung besarnya pengurangan *latency* yang diperoleh melalui penerapan pemrosesan lokal pada perangkat *edge*. Hasil penelitian diharapkan dapat memberikan dasar teknis dalam pengembangan sistem monitoring IPAL yang memiliki waktu responss lebih cepat serta menjadi landasan bagi penelitian lanjutan mengenai integrasi *Edge Artificial Intelligence* untuk mengevaluasi keseimbangan antara performa klasifikasi dan kecepatan pengambilan keputusan secara *real-time*.

2. METODE PENELITIAN

2.1 Tahapan Penelitian

Penelitian menggunakan pendekatan eksperimen kuantitatif dengan membandingkan dua arsitektur pemrosesan data, yaitu *Cloud Computing* dan *Edge Computing*. Variabel utama yang diamati adalah *latency* sistem. Data masukan berasal dari sensor kualitas air yang terdiri atas pH, suhu, dan *turbidity*. Pengujian dilakukan pada prototipe sistem monitoring IPAL yang telah dikembangkan dan kemudian digunakan sebagai dasar pengujian arsitektur *edge*. Pada arsitektur *Edge Computing*, sensor membaca parameter kualitas air dan ESP32 menerima data hasil pembacaan. Data dikirim melalui MQTT ke Raspberry Pi. Raspberry Pi melakukan pemrosesan secara lokal, kemudian hasil monitoring dikirim ke *edge-cloud framework* dan data

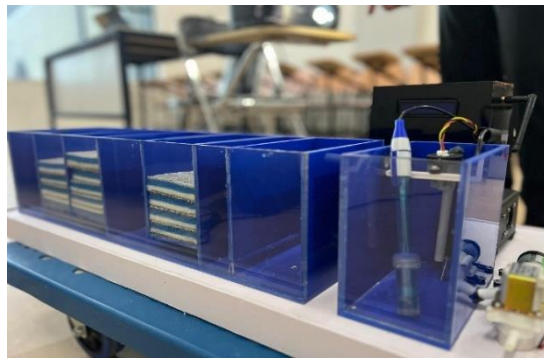
disimpan pada database lokal. Alur implementasi tersebut sesuai dengan rancangan pada laporan penelitian, yaitu sensor → ESP32 → MQTT → Raspberry Pi → hasil klasifikasi/*edge-cloud framework* → penyimpanan lokal.



Gambar 1. Arsitektur sistem monitoring IPAL berbasis *Edge Computing*

2.2 Tahapan Pengujian

Perangkat keras yang digunakan dalam penelitian ini sebagian memanfaatkan prototipe penelitian terdahulu yang telah memiliki sistem monitoring kualitas air berbasis IoT.

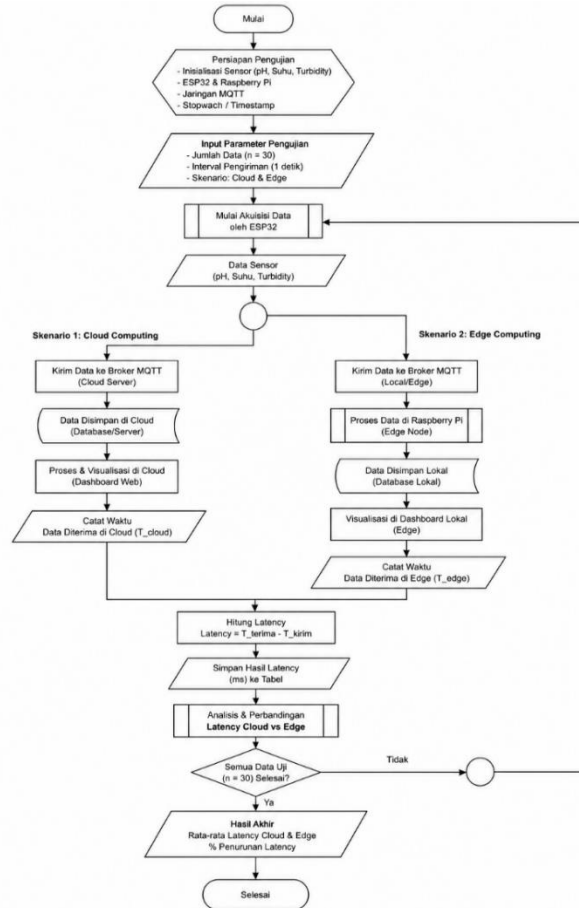


Gambar 2. Prototipe sistem IPAL berbasis IoT

1. Prototipe Sistem IPAL IoT, Penelitian ini menggunakan prototipe Instalasi Pengolahan Air Limbah (IPAL) berbasis *Internet of Things* (IoT) yang telah dikembangkan pada penelitian sebelumnya oleh Indarto dan Agustini. Prototipe tersebut telah mampu melakukan monitoring kualitas air secara *real-time* menggunakan beberapa sensor utama.
 - a. Sensor pH digunakan untuk mengukur tingkat keasaman atau kebasaan air limbah. Data pH digunakan sebagai salah satu parameter utama dalam proses klasifikasi kualitas air menggunakan TinyML.
 - b. Sensor *turbidity* digunakan untuk mengukur tingkat kekeruhan air limbah sebagai indikator tingkat pencemaran air.
 - c. Sensor suhu digunakan untuk mengukur temperatur air limbah yang dapat memengaruhi kondisi kualitas air.
 - d. ESP32 digunakan sebagai mikrokontroler utama yang berfungsi untuk: membaca data sensor, mengirim data, serta menjalankan model TinyML ringan untuk proses klasifikasi kualitas air secara *real-time*.
2. Raspberry Pi, digunakan sebagai perangkat *Edge Computing* untuk: pengolahan data lokal, preprocessing data, monitoring sistem, dan pengelolaan komunikasi data antara perangkat IoT dan *edge-cloud framework* monitoring.
3. Smartphone atau Laptop, digunakan untuk monitoring sistem melalui aplikasi Blynk secara *real-time*.

Parameter sensor digunakan sebagai input sistem, sedangkan *latency* digunakan untuk mengukur waktu respons sistem sejak data sensor diterima hingga keputusan berhasil dihasilkan oleh perangkat *Edge Computing*. Pengukuran *latency* dilakukan untuk membandingkan performa sistem berbasis *Edge Computing* dengan sistem berbasis *Cloud Computing*.

Pengujian dilakukan dengan dua skenario. Skenario pertama menggunakan arsitektur Cloud Computing, yaitu data dikirim ke server cloud untuk diproses dan responss dikembalikan ke sistem. Skenario kedua menggunakan Edge Computing, yaitu pemrosesan dilakukan secara lokal pada Raspberry Pi sehingga pengiriman data ke cloud tidak menjadi bagian dari jalur utama pengambilan keputusan. Pada pengujian cloud computing dan edge computing asing-asing dilakukan lima belas kali percobaan waktu request dan response. Hasil latency dihitung sebagai selisih waktu responss terhadap waktu request.



Gambar 3. Flowchart pengujian latency sistem IPAL

Latency dihitung menggunakan selisih waktu antara saat permintaan dikirim dan saat responss diterima. Rumus latency adalah:

$$Latency = t_{response} - t_{request}$$

Keterangan:

1. $t_{request}$ = waktu data dikirim
2. $t_{response}$ = waktu respons diterima

Rata-rata latency dihitung dengan membagi jumlah latency seluruh percobaan dengan jumlah percobaan. Untuk membandingkan kedua arsitektur, persentase pengurangan latency dihitung dengan cara menghitung persentase penurunan latency digunakan rumus:

$$Penurunan Latency = \frac{Latency_{Cloud} - Latency_{Edge}}{Latency_{Cloud}} \times 100\%$$

Keterangan:

3. $Latency_{Cloud}$ = rata-rata waktu respons sistem berbasis Cloud Computing (ms).
4. $Latency_{Edge}$ = rata-rata waktu respons sistem berbasis Edge Computing (ms).
5. Penurunan Latency (%) = persentase peningkatan kecepatan sistem setelah menggunakan Edge Computing.

Semakin kecil nilai latency yang diperoleh, semakin cepat sistem dalam melakukan pengolahan data dan menghasilkan keputusan secara real-time.

Interpretasi hasil *latency* menggunakan kategori yang tercantum pada laporan penelitian, yaitu kurang dari 50 ms dikategorikan sangat cepat, 50–100 ms cepat, 100–200 ms sedang, dan lebih dari 200 ms lambat. Kategori tersebut digunakan sebagai interpretasi tambahan, bukan sebagai pengganti nilai *latency* hasil eksperimen.

Tabel 1. Interpretasi Kecepatan Sistem

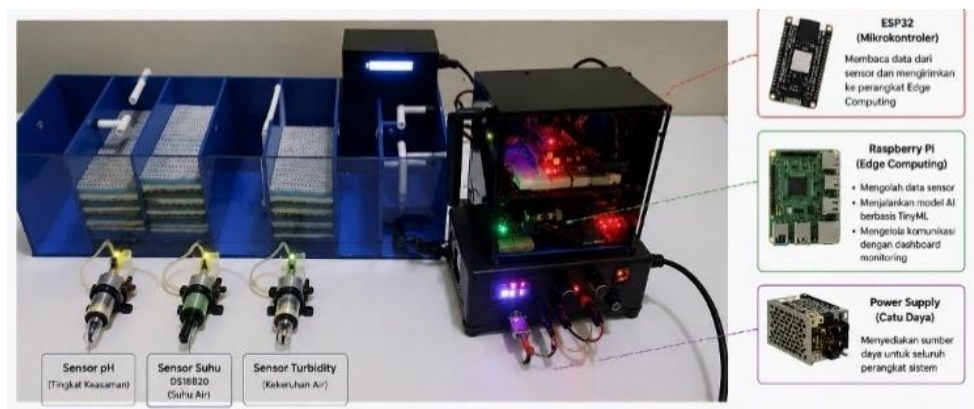
Nilai <i>Latency</i>	Kategori
< 50 ms	Sangat cepat
50–100 ms	Cepat
100–200 ms	Sedang
> 200 ms	Lambat

Sumber: Ajit Singh & Nirmalya Thakur. (2024). *Edge Artificial Intelligence: Convergence of Edge Computing and Artificial Intelligence*. Springer.

3. Hasil Dan Pembahasan

3.1 Pembahasan

Data kualitas air dikumpulkan menggunakan sensor pH, DS18B20, dan *turbidity* yang terpasang pada prototipe IPAL. menggunakan sensor pH, DS18B20, dan *turbidity* yang terpasang pada prototipe IPAL. Sampel pengukuran pada laporan penelitian mencakup air bersih, air limbah sabun, dan air limbah rumah tangga. Data tersebut digunakan sebagai masukan sistem klasifikasi dan monitoring. Implementasi *Edge Computing* menempatkan Raspberry Pi sebagai titik pemrosesan lokal sehingga data yang berasal dari ESP32 dapat diproses tanpa terlebih dahulu melewati server cloud.



Gambar 4. Implementasi Perangkat Keras

Pada tahap ini dilakukan pengumpulan data kualitas air menggunakan sensor pH, sensor suhu DS18B20, dan sensor *turbidity* yang terpasang pada prototipe Instalasi Pengolahan Air Limbah (IPAL). Pengumpulan data dilakukan secara *real-time* untuk memperoleh dataset yang digunakan dalam proses pengujian. Sampel yang digunakan terdiri atas tiga jenis air, yaitu air bersih, air limbah sabun, dan air limbah rumah tangga. Data hasil pembacaan sensor kemudian disimpan dan digunakan sebagai masukan pada sistem klasifikasi kualitas air.

Tabel 2. Data Pengujian menggunakan prototype IPAL

Jenis Sampel	Percobaan ke-	Nilai pH	Indikator (pH)	Keterangan Aktual
Air bersih	1	7,10	6–9	Layak
Air bersih	2	7,18	6–9	Layak
Air bersih	3	7,25	6–9	Layak
Air bersih	4	7,12	6–9	Layak
Air bersih	5	7,20	6–9	Layak
Air limbah sabun	1	9,21	6–9	Tidak Layak

Air limbah sabun	2	9,35	6–9	Tidak Layak
Air limbah sabun	3	9,41	6–9	Tidak Layak
Air limbah sabun	4	9,28	6–9	Tidak Layak
Air limbah sabun	5	9,33	6–9	Tidak Layak
Air limbah rumah tangga	1	5,82	6–9	Tidak Layak
Air limbah rumah tangga	2	5,74	6–9	Tidak Layak
Air limbah rumah tangga	3	5,91	6–9	Tidak Layak
Air limbah rumah tangga	4	5,68	6–9	Tidak Layak
Air limbah rumah tangga	5	5,79	6–9	Tidak Layak

Hasil pengukuran nilai pH air bersih memenuhi standar kualitas air. Nilai *turbidity* yang rendah menunjukkan tingkat kejernihan air yang baik sehingga seluruh sampel dikategorikan layak. Sedangkan hasil pengukuran air limbah sabun memiliki nilai pH di atas batas baku mutu serta tingkat kekeruhan yang tinggi sehingga dikategorikan tidak layak. Dan nilai *turbidity* yang melebihi ambang batas menunjukkan bahwa air limbah rumah tangga masih mengandung partikel tersuspensi sehingga belum memenuhi standar kualitas air.

3.1.1 Hasil Pengujian *Cloud Computing*

Pada arsitektur *Cloud Computing* diperoleh 15 hasil pengukuran *latency*. Nilai *latency* berada pada rentang 138–155 ms. Rata-rata *latency* yang diperoleh adalah 146,07 ms.

Tabel 3. Pengujian *Latency Cloud Computing*

Jenis Sampel	Percobaan Ke-	Nilai pH	t_request (ms)	t_responsse (ms)	Latency (ms)
Air bersih	1	7,10	0	145	145
Air bersih	2	7,18	3.200	3.351	151
Air bersih	3	7,25	7.050	7.188	138
Air bersih	4	7,12	11.200	11.349	149
Air bersih	5	7,20	15.400	15.543	143
Air limbah sabun	1	9,21	20.100	20.255	155
Air limbah sabun	2	9,35	24.350	24.490	140
Air limbah sabun	3	9,41	28.700	28.847	147
Air limbah sabun	4	9,28	33.250	33.402	152
Air limbah sabun	5	9,33	37.500	37.641	141
Air limbah rumah tangga	1	5,82	41.900	42.048	148
Air limbah rumah tangga	2	5,74	46.150	46.300	150
Air limbah rumah tangga	3	5,91	50.700	50.844	144
Air limbah rumah tangga	4	5,68	55.300	55.446	146
Air limbah rumah tangga	5	5,79	59.650	59.792	142

Perhitungan Rata-rata *Latency Cloud Computing*:

$$\bar{L}_{cloud} \frac{2191}{15} = 146,07 \text{ ms}$$

Latency Cloud: 146,07 ms (millisecond) = 0,14607 detik menunjukkan waktu yang dibutuhkan sistem *Cloud Computing* untuk menerima data sensor, mengirimkan data ke server cloud, melakukan proses pengolahan data, kemudian mengirimkan responss kembali ke sistem. Nilai *latency* sebesar 146,07 ms menunjukkan adanya waktu tunda dalam proses komunikasi antara perangkat IoT dan server cloud. Berdasarkan klasifikasi *latency* yang digunakan dalam penelitian, nilai tersebut termasuk dalam kategori sedang karena berada pada rentang 100–200 ms.

3.1.2 Hasil Pengujian *Edge Computing*

Pengujian pada arsitektur *Edge Computing* menghasilkan rata-rata *latency* sebesar 61 ms. Nilai ini diperoleh ketika proses pemrosesan dilakukan secara lokal pada Raspberry Pi.

Tabel 4. Pengujian *Latency Edge Computing*

Jenis Sampel	Percobaan Ke-	Nilai pH	t_request (ms)	t_responsse (ms)	Latency (ms)
Air bersih	1	7,10	0	59	59
Air bersih	2	7,18	3.200	3.263	63
Air bersih	3	7,25	7.050	7.111	61
Air bersih	4	7,12	11.200	11.260	60
Air bersih	5	7,20	15.400	15.462	62
Air limbah sabun	1	9,21	20.100	20.158	58
Air limbah sabun	2	9,35	24.350	24.414	64
Air limbah sabun	3	9,41	28.700	28.761	61
Air limbah sabun	4	9,28	33.250	33.310	60
Air limbah sabun	5	9,33	37.500	37.562	62
Air limbah rumah tangga	1	5,82	41.900	41.959	59
Air limbah rumah tangga	2	5,74	46.150	46.213	63
Air limbah rumah tangga	3	5,91	50.700	50.761	61
Air limbah rumah tangga	4	5,68	55.300	55.360	60
Air limbah rumah tangga	5	5,79	59.650	59.712	62

Perhitungan Rata-rata *Latency Edge Computing*:

$$\bar{L}_{edge} \frac{915}{15} = 61,00 \text{ ms}$$

Latency Edge: 61 ms (millisecond) = 0,061 detik menunjukkan waktu yang dibutuhkan sistem *Edge Computing* untuk menerima data sensor, melakukan pemrosesan data secara langsung pada perangkat edge di dekat sumber data, kemudian mengirimkan hasil responss kembali ke sistem. Nilai *latency* sebesar 61 ms menunjukkan bahwa pemrosesan pada edge dapat mengurangi waktu tunda karena data tidak perlu dikirim terlebih dahulu ke server cloud. Berdasarkan klasifikasi *latency* yang digunakan dalam penelitian, nilai tersebut termasuk dalam kategori cepat karena berada di bawah 100 ms.

Tabel 5. Perbandingan *latency Cloud Computing* dan *Edge Computing*

Metode	Jumlah Sampel	Latency Minimum (ms)	Latency Maksimum (ms)	Rata-rata Latency (ms)
<i>Cloud Computing</i>	15	138	155	146,07
<i>Edge Computing</i>	15	58	64	61,00

Perhitungan Pengurangan *Latency*:

Selisih *Latency* = 146,07–61,00=85,07 ms

Persentase = $146,07 / 146,07 - 61,00 \times 100\% = 58,24\%$

Hasil tersebut menunjukkan bahwa implementasi *Edge Computing* mampu menurunkan *latency* sebesar 58,24% dibandingkan *Cloud Computing*. Penurunan ini terjadi karena proses pengolahan data dilakukan secara lokal pada Raspberry Pi tanpa harus mengirim data ke server cloud terlebih dahulu. Dengan demikian, waktu transmisi data menjadi lebih singkat sehingga respons sistem menjadi lebih cepat.

3.1.3 Analisis Pengurangan Latency

Perbandingan menunjukkan bahwa penerapan *Edge Computing* menurunkan *latency* sebesar 85,07 ms atau 58,24% dibandingkan *Cloud Computing*. Penurunan ini terjadi karena proses pengolahan data dilakukan secara lokal pada Raspberry Pi tanpa harus mengirim data ke server cloud terlebih dahulu. Dengan demikian, waktu transmisi data menjadi lebih singkat sehingga respons sistem menjadi lebih cepat. Hasil ini menunjukkan bahwa perubahan lokasi pemrosesan memberikan pengaruh yang nyata terhadap waktu respons.

Penurunan *latency* dapat dijelaskan oleh perbedaan jalur komunikasi. Pada *Cloud Computing*, data harus dikirim melalui jaringan menuju server cloud, diproses, kemudian respons dikirim kembali. Pada *Edge Computing*, pemrosesan dilakukan pada Raspberry Pi yang berada dekat dengan sumber data. Dengan demikian, tahapan komunikasi ke cloud tidak berada pada jalur utama proses pengambilan keputusan.

3.1.4 Implikasi terhadap Sistem Monitoring IPAL

Nilai 61 ms menunjukkan bahwa *Edge Computing* memberikan respons yang lebih cepat dibandingkan *Cloud Computing* pada skenario pengujian. Dalam konteks monitoring IPAL, waktu respons yang lebih rendah penting ketika perubahan parameter kualitas air perlu segera diketahui. Selain *latency*, pemrosesan lokal juga berpotensi mengurangi jumlah data mentah yang harus ditransmisikan karena *edge-cloud framework* dapat menerima hasil pemrosesan atau data penting yang diperlukan untuk monitoring.

Hasil penelitian juga memperlihatkan hubungan yang logis antara penelitian sebelumnya dan penelitian ini. Sistem IPAL berbasis IoT yang dikembangkan sebelumnya telah menyediakan akuisisi parameter pH, *turbidity*, dan suhu serta monitoring *real-time*. Penelitian ini melanjutkannya dengan menguji pemindahan pemrosesan ke edge dan mengukur dampaknya terhadap *latency*. Dengan demikian, *Edge Computing* dapat diposisikan sebagai lapisan optimasi arsitektur, bukan sebagai pengganti fungsi sensor dan monitoring. Berdasarkan hasil pengujian dan analisis yang telah dilakukan, implementasi *Edge Computing* memberikan beberapa keunggulan dibandingkan *Cloud Computing*, yaitu:

1. Mampu menurunkan *latency* hingga 58,24%.
2. Mempercepat proses pengambilan keputusan secara *real-time*.
3. Mengurangi waktu respons sistem monitoring kualitas air berbasis IoT.

Dengan demikian, penerapan *Edge Computing* pada skenario pengujian menunjukkan peningkatan performa dari sisi waktu respons sistem dibandingkan *Cloud Computing*. Integrasi dengan *Artificial Intelligence* berbasis *TinyML* merupakan peluang pengembangan pada penelitian selanjutnya untuk menguji performa klasifikasi kualitas air pada perangkat *edge*.

3.1.5 Keterbatasan Penelitian

Penelitian ini berfokus pada *latency* dan tidak dirancang untuk membandingkan seluruh aspek performa jaringan secara komprehensif. Oleh karena itu, penelitian lanjutan diperlukan untuk melakukan pengujian dengan jumlah sampel lebih besar, variasi beban jaringan, dan evaluasi statistik yang lebih mendalam. Penelitian lanjutan juga dapat mengevaluasi *accuracy*, *precision*, dan *recall* model AI yang berjalan pada edge agar kecepatan sistem dapat dianalisis bersama kualitas hasil klasifikasi.

4 KESIMPULAN

Berdasarkan hasil pengujian dan analisis yang telah dilakukan terhadap penerapan *Edge Computing* pada sistem monitoring Instalasi Pengolahan Air Limbah (IPAL) berbasis IoT, dapat disimpulkan beberapa hal sebagai berikut:

1. Penerapan *Edge Computing* pada sistem monitoring IPAL berbasis IoT mampu memberikan waktu respons yang lebih rendah dibandingkan dengan *Cloud Computing* dilihat berdasarkan 15 kali percobaan menggunakan data sensor pH, suhu DS18B20, dan *turbidity*, diperoleh hasil:
 - a. *Cloud Computing* menghasilkan rata-rata *latency* sebesar 146,07 ms, dengan nilai minimum 138 ms dan maksimum 155 ms.
 - b. *Edge Computing* yang menggunakan Raspberry Pi sebagai perangkat pemrosesan lokal menghasilkan rata-rata *latency* sebesar 61,00 ms, dengan nilai minimum 58 ms dan maksimum 64 ms.
2. Hasil perbandingan menunjukkan bahwa penggunaan *Edge Computing* mampu mengurangi *latency* sebesar 85,07 ms atau 58,24% dibandingkan dengan *Cloud Computing*. Pengurangan *latency* tersebut menunjukkan bahwa pemrosesan data secara lokal pada Raspberry Pi dapat mengurangi tahapan

komunikasi dengan server cloud pada jalur utama pengambilan keputusan, sehingga menghasilkan respons sistem yang lebih cepat.

3. Berdasarkan kategori latency yang digunakan dalam penelitian:

- a. Rata-rata latency Cloud Computing sebesar 146,07 ms termasuk kategori sedang.
- b. Rata-rata latency Edge Computing sebesar 61,00 ms termasuk kategori cepat.

Dengan demikian, pada skenario pengujian yang dilakukan, Edge Computing menunjukkan performa yang lebih baik dari sisi waktu respons untuk mendukung monitoring kualitas air secara real-time.

Secara keseluruhan, hasil penelitian membuktikan bahwa lokasi pemrosesan data berpengaruh terhadap latency sistem monitoring IPAL berbasis IoT. Pemindahan proses pengolahan data dari cloud ke perangkat edge di dekat sumber data memberikan peningkatan performa pada aspek waktu respons. Penelitian ini masih berfokus pada latency dan belum mengevaluasi performa sistem secara menyeluruh maupun kualitas klasifikasi AI. Oleh karena itu, hasil penelitian ini dapat menjadi dasar untuk pengembangan sistem monitoring IPAL berbasis Edge AI atau TinyML pada penelitian selanjutnya.

DAFTAR PUSTAKA

- [1] C. T. H. Manurung, J. Arifin, F. T. Syifa, and R. A. Rochmanto, "Pemanfaatan ESP32 Sebagai Sistem Pemantauan Kualitas Air Keran Siap Minum Secara Real-Time Menggunakan Aplikasi," *Journal of Telecommunication Electronics and Control Engineering*, vol. 4, no. 2, 2022, doi: 10.20895/jtece.v4i2.535.
- [2] A. Fahmi and I. Kurniawan, "Studi Kinerja Transmisi Data Menggunakan ESP32 dan Raspberry Pi Pico Berbasis Simulasi Wokwi," *Journal Zetroom*, vol. 7, no. 1, 2025, doi: 10.36526/ztr.v7i1.5102.
- [3] M. Tomas, Z. Zaini, Zulkarnaini, A. A. Yulianto, and S. Ismail, "Edge-Enabled Digital Twin for Autonomous Low-Latency Dissolved Oxygen Control for Partial Nitrification-Anammox Reactor," *Applied Science and Engineering Progress*, 2026.
- [4] "Implementation of IoT Edge Computing for Control and Monitoring System of Hydroponic Plant Water Quality Using Raspberry Pi," 2024.
- [5] "A Study on Edge Computing through Machine Learning for IoT Devices," in *2021 International Conference on Forensics, Analytics, Big Data, Security (FABS)*, 2021, doi: 10.1109/FABS52071.2021.9702557.
- [6] "Intelligent Delay-Aware Partial Computing Task Offloading for Multiuser Industrial Internet of Things Through Edge Computing," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 2954–2966, 2023, doi: 10.1109/JIOT.2021.3123406.
- [7] "IoT-based edge computing (IoTEC) for improved environmental monitoring," *Sustainable Computing: Informatics and Systems*, vol. 38, p. 100870, 2023, doi: 10.1016/j.suscom.2023.100870.
- [8] J. M. Jasim, F. M. Adress, and A. A. Fadhil, "Evaluating Reliability of Communication in IoT based Water Monitoring Using ESP32," *Kyzylorda Scholarly Review*, 2026.
- [9] B. R. Rambani and A. Suhendi, "Sistem Pemilah Limbah Organik dan Anorganik Berbasis IoT untuk Bank Sampah di SMP Terpadu Al-Qudwah," *Journal Information & Computer (JICOM)*, vol. 4, no. 2, pp. 84–92, 2026.
- [10] Muarif, A. Sumaedi, and Mardiansyah, "Monitoring Suhu Ruangan Server Jaringan Hostpot Berbasis Mikrokontroler NodeMCU ESP8266 Dengan Thingspeak," *Journal Information & Computer (JICOM)*, vol. 2, no. 1, pp. 17–27, 2024, doi: 10.32493/jicomisc.v2i1.38638.
- [11] F. Firdaus and A. Suhendi, "Perancangan Sistem Otomatisasi Berbasis ID Card dan Internet of Things (IoT) pada Pintu Kamar Kost Menggunakan Mikrokontroler ESP32," *Journal Information & Computer (JICOM)*, vol. 4, no. 2, pp. 62–67, 2026.
- [12] K. Fadhilah, M. S. Lestari, and S. Devanti, "Monitoring dan Otomatisasi Sistem Aquaponik Budidaya Lele dan Kangkung Berbasis IoT Menggunakan ESP32 dan Aplikasi Blynk," *Prosiding Seminar Kecerdasan Artifisial, Sains Data, dan Pendidikan Masa Depan*, vol. 3, no. 2, pp. 34–40, 2025.