

Penerapan Arsitektur Edge Intelligence Untuk Reduksi Latensi Pada Sistem Pemantauan Internet of Things Lokal

Rudianto^{1*}

¹Fakultas Ilmu Komputer, Program Ilmu Komputer, Universitas Bina Bangsa, Serang, Indonesia

Email: 1*Rudianto@binabangsa.ac.id

(* Email Corresponding Author: Rudianto@binabangsa.ac.id)

Received: 12 Januari 2026 | Revision: 13 Januari 2026 | Accepted: 15 Januari 2026

Abstrak

Penerapan Internet of Things (IoT) pada sistem pemantauan real-time umumnya memanfaatkan arsitektur pemrosesan terpusat berbasis cloud computing. Meskipun pendekatan tersebut memberikan kemudahan dalam pengelolaan dan penyimpanan data, ketergantungan terhadap server pusat sering menimbulkan permasalahan latensi, khususnya pada lingkungan jaringan lokal dengan keterbatasan bandwidth dan konektivitas internet yang tidak stabil. Latensi yang tinggi dapat mengurangi kecepatan respons sistem serta menurunkan efektivitas pengambilan keputusan secara real-time. Penelitian ini bertujuan untuk menerapkan arsitektur edge intelligence sebagai solusi dalam mereduksi latensi pada sistem pemantauan berbasis Internet of Things lokal. Pendekatan yang digunakan adalah dengan memindahkan sebagian proses pemrosesan data dan pengambilan keputusan ke node edge yang berada dekat dengan sumber data. Data sensor diproses secara lokal menggunakan mekanisme analisis berbasis ambang batas sebelum dikirimkan ke server pusat untuk keperluan penyimpanan dan visualisasi. Evaluasi kinerja sistem dilakukan dengan membandingkan latensi dan waktu respons antara sistem IoT berbasis cloud penuh dan sistem IoT yang menerapkan edge intelligence. Hasil pengujian menunjukkan bahwa pemrosesan data di sisi edge mampu menurunkan latensi secara signifikan serta meningkatkan keandalan sistem, terutama pada kondisi jaringan lokal. Dengan demikian, arsitektur edge intelligence terbukti efektif untuk mendukung sistem pemantauan IoT yang membutuhkan respons cepat dan stabil.

Kata Kunci: Edge Intelligence, Internet of Things, Latensi, Sistem Pemantauan, Edge Computing

Abstract

Internet of Things (IoT) real-time monitoring systems commonly rely on centralized cloud-based architectures for data processing and management. Although cloud computing offers flexibility and scalability, its dependency on remote servers often leads to latency issues, especially in local network environments with limited bandwidth and unstable internet connectivity. High latency can negatively affect system responsiveness and reduce the effectiveness of real-time decision-making. This study aims to implement an edge intelligence architecture to reduce latency in local Internet of Things-based monitoring systems. The proposed approach shifts part of the data processing and decision-making tasks to edge nodes located closer to data sources. Sensor data are processed locally using threshold-based analysis before being transmitted to the central server for storage and visualization purposes. System performance is evaluated by comparing latency and response time between a fully cloud-based IoT system and an edge intelligence-based system. The experimental results indicate that local data processing at the edge significantly reduces latency and improves system reliability, particularly in local network conditions. Therefore, the edge intelligence architecture is considered an effective solution for enhancing the performance of IoT monitoring systems that require fast and reliable responses.

Keywords: Edge Intelligence, Internet of Things, Latency, Monitoring System, Edge Computing

1. PENDAHULUAN

Perkembangan teknologi Internet of Things (IoT) telah mendorong transformasi signifikan dalam cara sistem pemantauan dirancang dan diimplementasikan. IoT memungkinkan berbagai perangkat fisik yang dilengkapi sensor untuk saling terhubung dan bertukar data secara otomatis melalui jaringan [1]. Penerapan IoT banyak dijumpai pada sistem pemantauan lingkungan, industri manufaktur, pertanian cerdas, hingga infrastruktur bangunan. Sistem pemantauan berbasis IoT umumnya dirancang untuk bekerja secara real-time, sehingga mampu memberikan informasi kondisi lapangan secara cepat dan akurat kepada pengguna atau sistem pengambil keputusan.

Pada sebagian besar implementasinya, sistem IoT masih mengandalkan arsitektur pemrosesan terpusat berbasis cloud computing [2]. Dalam arsitektur ini, data yang dihasilkan oleh sensor dikirimkan ke server pusat untuk diproses, dianalisis, dan disimpan. Pendekatan tersebut menawarkan keunggulan dari sisi skalabilitas, kemudahan pengelolaan data, serta dukungan analisis dalam skala besar [3]. Namun, ketergantungan penuh terhadap cloud juga menimbulkan sejumlah permasalahan, terutama terkait latensi komunikasi data [4].

Latensi merupakan waktu tunda yang terjadi sejak data dikirimkan dari perangkat IoT hingga hasil pemrosesan diterima kembali oleh sistem atau pengguna. Pada aplikasi IoT yang bersifat real-time, latensi yang tinggi dapat menyebabkan keterlambatan respons sistem dan menurunkan kualitas layanan [5]. Permasalahan ini semakin terasa pada lingkungan jaringan lokal yang memiliki keterbatasan bandwidth atau koneksi internet yang tidak stabil, seperti pada kawasan industri tertutup, gedung perkantoran, kampus, maupun wilayah terpencil [6]. Dalam kondisi tersebut, sistem pemantauan berbasis cloud sering kali tidak mampu memberikan respons yang cepat dan konsisten.

Sebagai solusi atas permasalahan latensi, konsep edge computing mulai banyak dikembangkan dalam sistem IoT [7]. Edge computing memindahkan sebagian proses komputasi dari server pusat ke perangkat yang berada lebih dekat

dengan sumber data. Dengan pendekatan ini, data tidak perlu selalu dikirimkan ke cloud untuk diproses, sehingga waktu transmisi dapat dikurangi secara signifikan [8]. Meskipun demikian, implementasi edge computing pada banyak penelitian masih terbatas pada pemrosesan dasar, seperti filtrasi data atau agregasi sederhana, tanpa adanya kemampuan analisis yang lebih cerdas.

Perkembangan selanjutnya dari edge computing melahirkan konsep edge intelligence, yaitu integrasi antara komputasi tepi dan kemampuan analisis cerdas pada node edge. Dalam arsitektur edge intelligence, node edge tidak hanya berfungsi sebagai perantara data, tetapi juga mampu melakukan analisis, klasifikasi, serta pengambilan keputusan secara lokal. Pendekatan ini memungkinkan sistem IoT untuk merespons kondisi tertentu secara langsung tanpa harus menunggu hasil pemrosesan dari server pusat [9]. Dengan demikian, edge intelligence berpotensi besar dalam mengatasi permasalahan latensi dan meningkatkan keandalan sistem pemantauan.

Beberapa penelitian dalam lima tahun terakhir telah mengkaji penerapan edge intelligence pada berbagai skenario IoT. Penelitian [10] menunjukkan bahwa pemrosesan data di sisi edge mampu menurunkan latensi secara signifikan pada sistem pemantauan industri. Penelitian lain melaporkan bahwa edge intelligence dapat meningkatkan efisiensi bandwidth dengan hanya mengirimkan data penting ke server pusat. Selain itu, beberapa studi juga mengungkapkan bahwa sistem berbasis edge intelligence memiliki tingkat keandalan yang lebih tinggi ketika terjadi gangguan koneksi jaringan ke cloud. Meskipun demikian, sebagian besar penelitian tersebut berfokus pada lingkungan berskala besar atau sistem berbasis jaringan luas, sementara implementasi pada sistem IoT lokal masih relatif terbatas.

Berdasarkan kajian penelitian terkait tersebut, dapat diidentifikasi adanya research gap dalam penerapan edge intelligence pada sistem pemantauan IoT berbasis jaringan lokal. Masih diperlukan kajian yang secara khusus mengevaluasi efektivitas edge intelligence dalam mereduksi latensi dan meningkatkan waktu respons sistem pada lingkungan lokal dengan keterbatasan konektivitas. Selain itu, diperlukan perancangan arsitektur sistem yang sederhana namun realistis agar dapat diimplementasikan secara langsung pada skenario nyata.

Oleh karena itu, penelitian ini berfokus pada penerapan arsitektur edge intelligence untuk mereduksi latensi pada sistem pemantauan berbasis Internet of Things lokal. Penelitian ini mengusulkan arsitektur sistem yang memanfaatkan node edge sebagai pusat pemrosesan data lokal dan pengambilan keputusan berbasis kondisi tertentu. Evaluasi kinerja sistem dilakukan dengan membandingkan latensi dan waktu respons antara sistem IoT berbasis cloud penuh dan sistem IoT yang menerapkan edge intelligence.

Tujuan dari penelitian ini adalah untuk menganalisis sejauh mana penerapan edge intelligence mampu meningkatkan performa sistem pemantauan IoT lokal, khususnya dalam hal reduksi latensi dan peningkatan keandalan sistem. Diharapkan hasil penelitian ini dapat menjadi referensi bagi pengembang dan peneliti dalam merancang sistem IoT yang lebih responsif, efisien, dan sesuai untuk lingkungan jaringan lokal.

2. METODOLOGI PENELITIAN

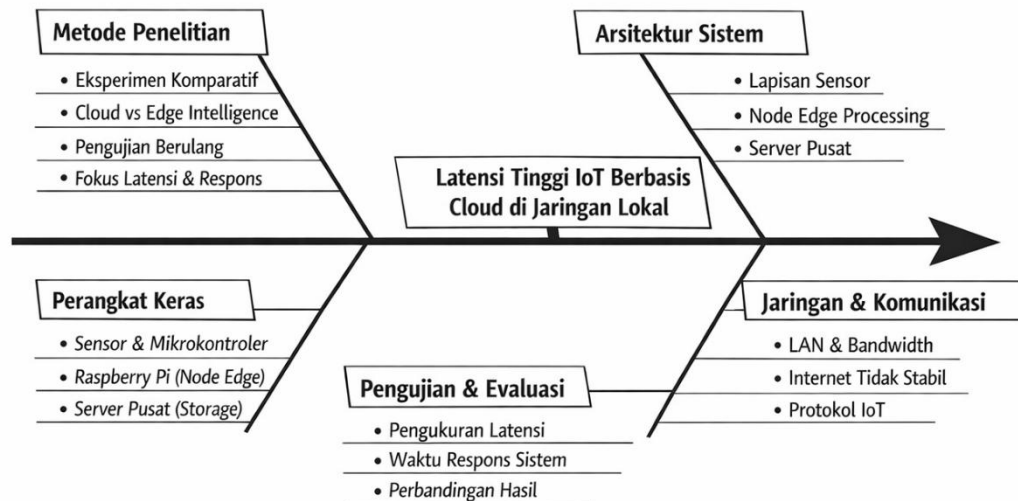
Penelitian ini menggunakan metode eksperimental komparatif, yaitu membandingkan kinerja sistem pemantauan Internet of Things (IoT) berbasis cloud computing dengan sistem pemantauan IoT yang menerapkan arsitektur edge intelligence. Fokus utama pengujian adalah pengukuran latensi dan waktu respons sistem pada lingkungan jaringan lokal. Metodologi penelitian disusun secara sistematis agar tahapan penelitian dapat dilakukan secara terstruktur dan hasil yang diperoleh dapat dipertanggungjawabkan secara ilmiah.

Tabel 1. Tahapan Penelitian

No	Tahapan Penelitian	Minggu ke-1	Minggu ke-2	Minggu ke-3	Minggu ke-4	Minggu ke-5	Minggu ke-6
1	Studi Literatur	✓	✓				
2	Analisis Kebutuhan		✓	✓			
3	Perancangan Sistem			✓	✓		
4	Implementasi Sistem				✓	✓	
5	Pengujian & Evaluasi					✓	✓

Fishbone diagram penelitian pada Gambar 1 dibawah ini digunakan untuk mengidentifikasi dan memetakan faktor-faktor utama yang memengaruhi permasalahan latensi pada sistem pemantauan Internet of Things (IoT) berbasis cloud di lingkungan jaringan lokal. Permasalahan utama yang dianalisis adalah tingginya latensi sistem, yang dipengaruhi oleh enam aspek, yaitu metode penelitian, perangkat keras, perangkat lunak, arsitektur sistem, jaringan dan komunikasi, serta pengujian dan evaluasi. Pada aspek metode penelitian, digunakan pendekatan eksperimen komparatif dengan membandingkan kinerja sistem IoT berbasis cloud penuh dan sistem IoT yang menerapkan arsitektur edge intelligence melalui pengujian berulang terhadap latensi dan waktu respons. Aspek perangkat keras meliputi penggunaan sensor,

mikrokontroler sebagai node sensor, Raspberry Pi sebagai node edge, dan server pusat sebagai media penyimpanan data, sedangkan aspek perangkat lunak mencakup pemanfaatan bahasa pemrograman Python, framework Flask, basis data MySQL, dan antarmuka berbasis web untuk visualisasi data. Aspek arsitektur sistem dirancang dalam tiga lapisan, yaitu lapisan sensor, lapisan edge intelligence, dan lapisan server pusat, dengan pemrosesan data lokal berbasis ambang batas (threshold-based analysis) pada node edge untuk menekan latensi komunikasi. Faktor jaringan dan komunikasi mencakup keterbatasan bandwidth jaringan lokal, koneksi internet yang tidak stabil, serta penggunaan protokol komunikasi IoT. Sementara itu, aspek pengujian dan evaluasi dilakukan melalui pengukuran dan perbandingan latensi serta waktu respons pada dua skenario sistem, sehingga fishbone diagram ini membantu peneliti menganalisis hubungan sebab-akibat secara



Gambar 1. Fishbone diagram penelitian

sistematis dan mendukung perumusan pendekatan metodologis yang tepat dalam mencapai tujuan penelitian.

2.1 Studi Literatur

Tahap studi literatur dilakukan untuk memperoleh pemahaman yang komprehensif mengenai konsep dan teknologi yang berkaitan dengan penelitian ini. Studi literatur mencakup pembahasan mengenai Internet of Things, cloud computing, edge computing, serta edge intelligence [11]. Sumber literatur diperoleh dari jurnal ilmiah, prosiding konferensi, dan publikasi ilmiah lainnya yang relevan dalam lima tahun terakhir. Hasil dari studi literatur digunakan untuk mengidentifikasi permasalahan utama pada sistem pemantauan IoT berbasis cloud, khususnya terkait latensi dan ketergantungan terhadap koneksi internet. Selain itu, studi literatur juga berfungsi untuk menemukan celah penelitian (research gap) yang menjadi dasar pengembangan arsitektur edge intelligence pada sistem pemantauan IoT lokal.

2.2 Analisis Kebutuhan

Tahap analisis kebutuhan bertujuan untuk menentukan spesifikasi sistem yang akan dikembangkan agar sesuai dengan tujuan penelitian. Analisis ini meliputi kebutuhan perangkat keras, perangkat lunak, serta kebutuhan jaringan. Perangkat keras yang dianalisis mencakup sensor, mikrokontroler, node edge, dan server pusat. Sementara itu, perangkat lunak yang dibutuhkan meliputi sistem operasi, bahasa pemrograman, protokol komunikasi, serta sistem basis data.

Selain itu, analisis kebutuhan juga mencakup penentuan parameter pemantauan yang akan diuji, seperti latensi transmisi data dan waktu respons sistem. Lingkungan pengujian difokuskan pada jaringan lokal untuk mensimulasikan kondisi nyata di mana koneksi internet memiliki keterbatasan. Hasil analisis kebutuhan menjadi dasar dalam tahap perancangan sistem.

2.3 Perancangan

Tahap perancangan dilakukan untuk menyusun arsitektur sistem pemantauan IoT berbasis edge intelligence. Arsitektur sistem dirancang dengan membagi sistem ke dalam tiga lapisan utama, yaitu lapisan sensor, lapisan edge, dan lapisan server pusat. Lapisan sensor bertugas mengumpulkan data dari lingkungan, sedangkan lapisan edge berfungsi sebagai pusat pemrosesan data lokal dan pengambilan keputusan. Lapisan server pusat digunakan untuk penyimpanan data jangka panjang dan visualisasi [12].

Pada tahap ini juga dirancang alur pengolahan data, mulai dari pengambilan data sensor, pemrosesan lokal di node edge, hingga pengiriman hasil pemrosesan ke server pusat. Perancangan dilakukan dengan mempertimbangkan efisiensi pemrosesan dan minimisasi latensi komunikasi data.

2.4 Perancangan

Implementasi merupakan tahap realisasi dari rancangan sistem yang telah dibuat. Pada tahap ini, sensor dihubungkan dengan mikrokontroler untuk mengumpulkan data secara periodik. Node edge diimplementasikan untuk menerima data dari sensor dan melakukan pemrosesan lokal menggunakan logika analisis berbasis kondisi tertentu.

Sistem komunikasi antara sensor, node edge, dan server pusat diimplementasikan menggunakan protokol komunikasi yang sesuai untuk IoT [13]. Data hasil pemrosesan di sisi edge kemudian dikirimkan ke server pusat untuk disimpan dan ditampilkan dalam bentuk visualisasi. Implementasi dilakukan pada lingkungan jaringan lokal sesuai dengan skenario pengujian yang telah ditetapkan.

2.5 Pengujian

Tahap pengujian bertujuan untuk mengevaluasi kinerja sistem yang telah diimplementasikan. Pengujian dilakukan dengan dua skenario utama, yaitu sistem IoT berbasis cloud penuh dan sistem IoT berbasis edge intelligence. Pada setiap skenario, dilakukan pengukuran latensi dan waktu respons sistem secara berulang untuk memperoleh hasil yang representatif.

Hasil pengujian kemudian dianalisis untuk mengetahui perbedaan performa antara kedua arsitektur sistem. Analisis ini digunakan untuk menilai efektivitas penerapan edge intelligence dalam mereduksi latensi dan meningkatkan keandalan sistem pemantauan IoT lokal.

3. HASIL DAN PEMBAHASAN

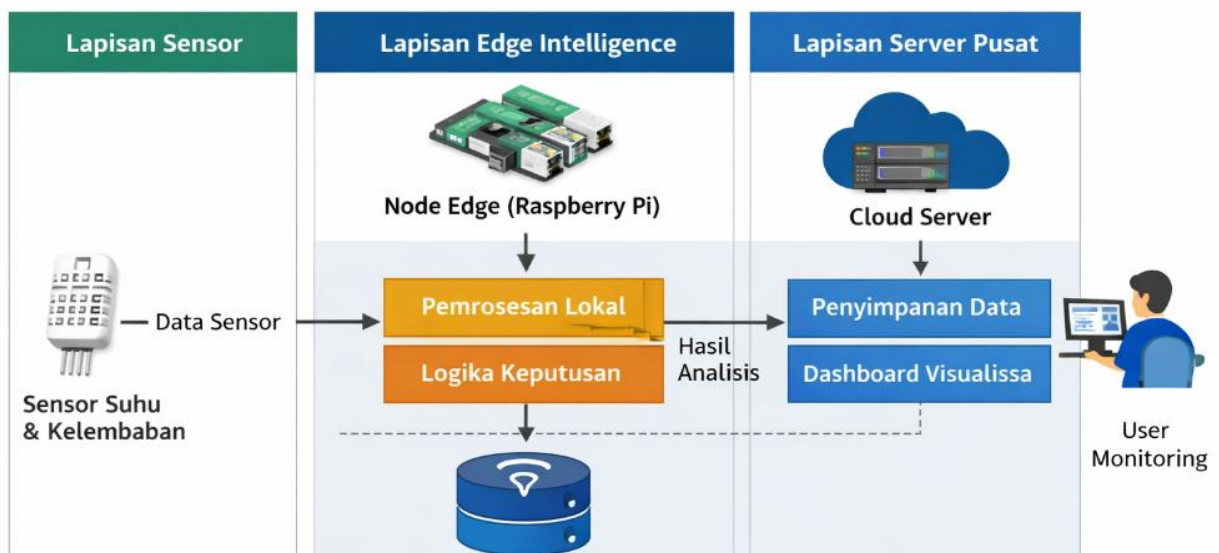
Bagian ini menyajikan hasil perancangan dan implementasi sistem pemantauan Internet of Things berbasis edge intelligence serta pembahasan terhadap kinerja sistem yang dikembangkan. Pembahasan difokuskan pada arsitektur sistem, perancangan antarmuka pengguna, hasil implementasi, dan analisis performa sistem berdasarkan hasil pengujian yang telah dilakukan.

3.1 Arsitektur Sistem

Arsitektur sistem yang diusulkan dirancang untuk mendukung pemrosesan data secara lokal guna mengurangi latensi pada sistem pemantauan Internet of Things [14]. Sistem ini dibagi ke dalam tiga lapisan utama, yaitu lapisan sensor, lapisan edge intelligence, dan lapisan server pusat.

Lapisan sensor berfungsi sebagai pengumpul data lingkungan, seperti suhu dan kelembaban. Data yang diperoleh dari sensor dikirimkan ke node edge melalui jaringan lokal. Pada lapisan edge intelligence, data diproses secara langsung menggunakan logika analisis berbasis ambang batas (threshold-based analysis). Proses ini memungkinkan sistem untuk melakukan pengambilan keputusan awal tanpa harus menunggu pemrosesan di server pusat.

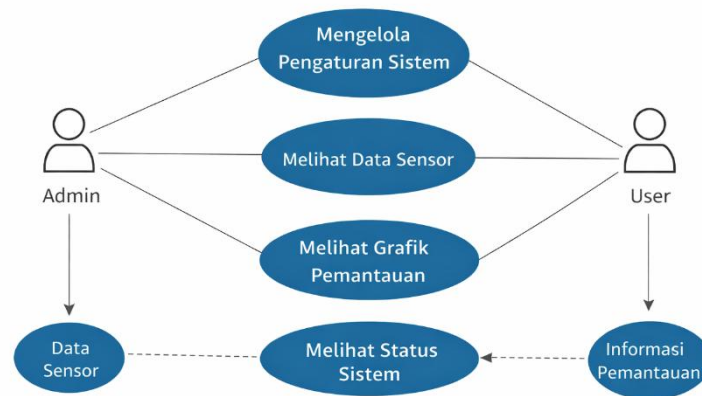
Lapisan server pusat berfungsi sebagai media penyimpanan data jangka panjang dan visualisasi [15]. Data yang dikirim ke server pusat merupakan hasil pemrosesan dari node edge, sehingga volume data yang dikirim menjadi lebih kecil dan efisien. Arsitektur ini dirancang untuk tetap dapat beroperasi meskipun terjadi gangguan koneksi internet, karena proses utama dilakukan di sisi edge.



Gambar 2. Arsitektur Sistem Pemantauan IoT Berbasis Edge Intelligence

3.1.1 Perancangan User Interface

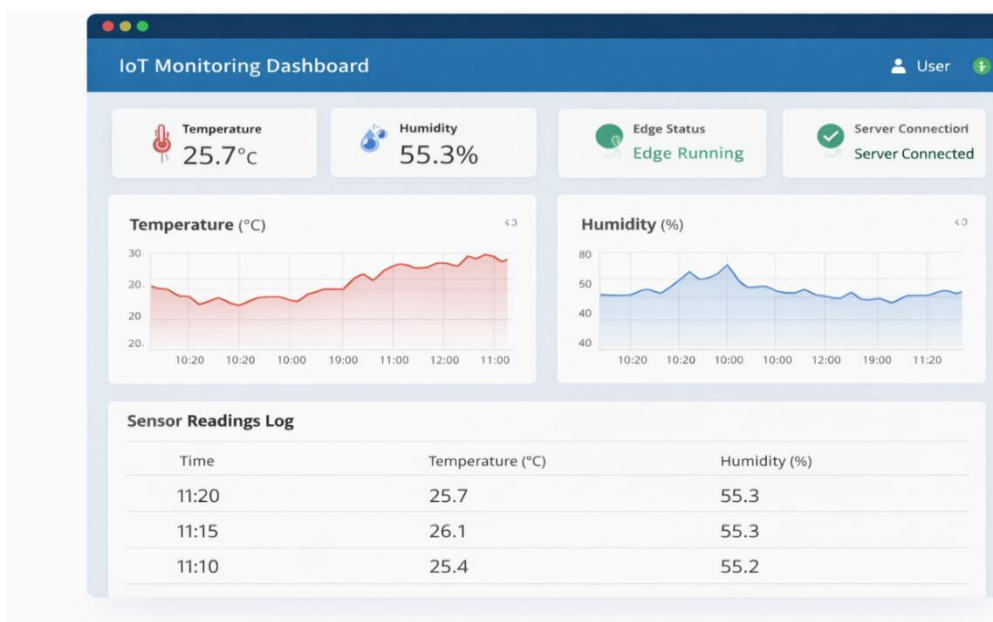
Perancangan User Interface (UI) dilakukan untuk memudahkan pengguna dalam memantau kondisi sistem secara real-time. Antarmuka pengguna dirancang berbasis web agar dapat diakses melalui perangkat komputer maupun perangkat bergerak. Informasi yang ditampilkan meliputi data sensor terkini, status sistem, dan riwayat data pemantauan. Dalam sistem ini terdapat dua aktor utama, yaitu Admin dan User. Admin memiliki hak akses penuh untuk mengelola sistem, seperti melihat seluruh data pemantauan dan melakukan pengaturan sistem. User hanya memiliki hak untuk melihat data pemantauan tanpa dapat mengubah konfigurasi sistem. Interaksi antara aktor dan sistem digambarkan dalam use case diagram. Use case diagram ini menunjukkan fungsi utama sistem, antara lain melihat data sensor, melihat grafik pemantauan, dan memantau status sistem secara real-time. Diagram ini membantu menggambarkan kebutuhan fungsional sistem secara jelas dan terstruktur.



Gambar 3. Use Case Diagram Sistem Pemantauan IoT

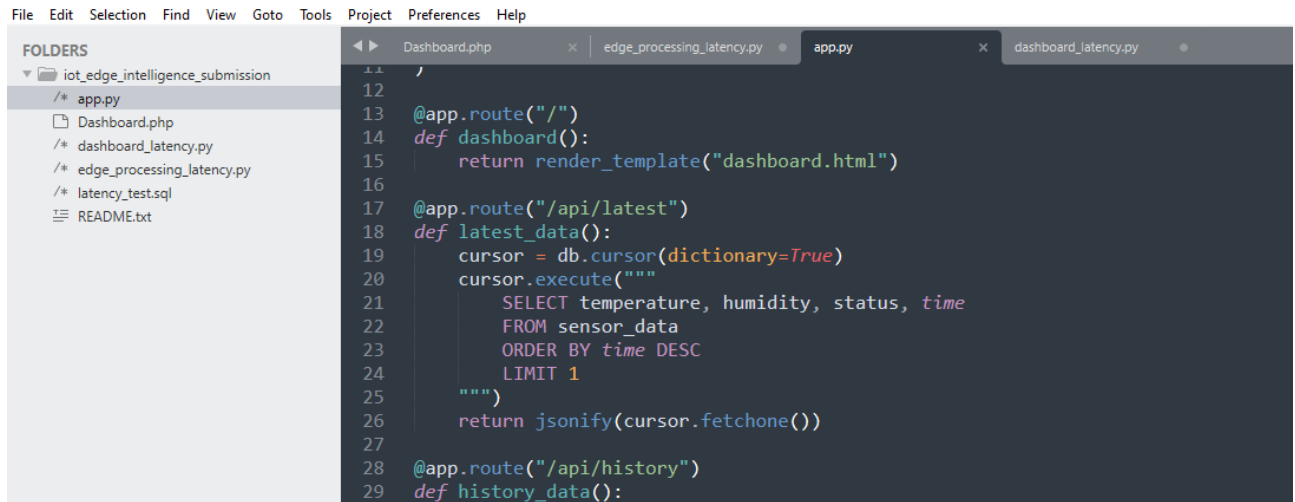
3.2 Implementasi

Hasil implementasi menunjukkan bahwa sistem pemantauan IoT berbasis edge intelligence dapat berjalan sesuai dengan rancangan yang telah ditetapkan. Sensor berhasil mengirimkan data ke node edge secara periodik, dan node edge mampu melakukan pemrosesan data lokal dengan baik. Antarmuka pengguna menampilkan data sensor secara real-time dalam bentuk angka dan grafik, sehingga memudahkan pengguna dalam memantau kondisi lingkungan. Selain itu, sistem mampu menampilkan status pemrosesan data di sisi edge dan status koneksi ke server pusat. Implementasi ini membuktikan bahwa pemrosesan data di sisi edge dapat mengurangi ketergantungan terhadap server pusat. Sistem tetap dapat menampilkan data pemantauan meskipun koneksi internet mengalami gangguan, selama jaringan lokal tetap aktif.



Gambar 4. Dashboard Sistem

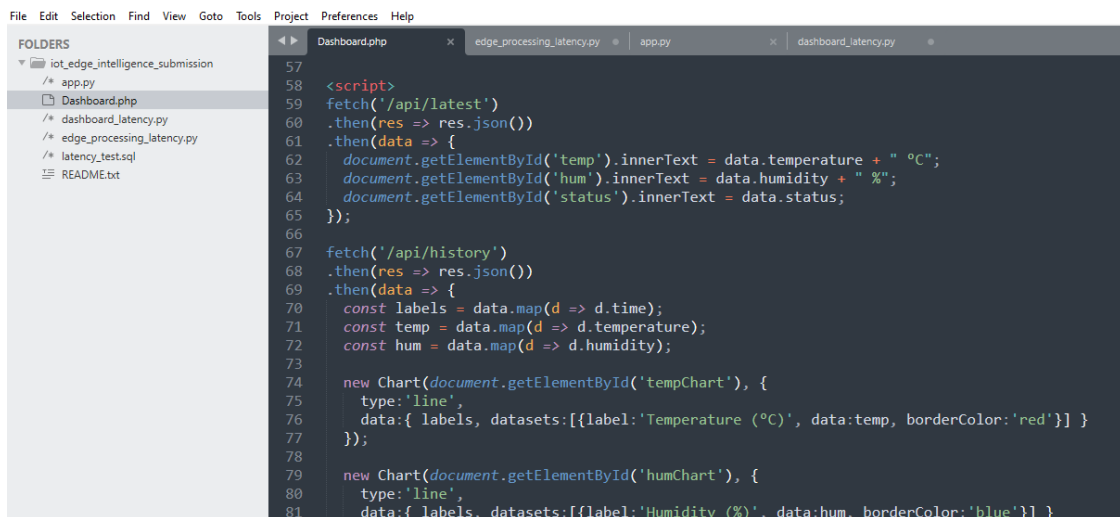
Berkas app.py berfungsi sebagai komponen backend pada sistem pemantauan Internet of Things berbasis edge intelligence. Aplikasi ini dikembangkan menggunakan kerangka kerja Flask untuk menghubungkan basis data MySQL dengan antarmuka pengguna. app.py bertugas mengambil data hasil pemrosesan edge intelligence dari tabel sensor_data, termasuk nilai suhu, kelembaban, status sistem, dan waktu pencatatan data. Data tersebut disediakan melalui layanan API sehingga dapat diakses secara dinamis oleh antarmuka web, sekaligus memastikan proses penyajian data berlangsung secara real-time dan konsisten dengan kondisi aktual sistem.



```
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  ▼ iot_edge_intelligence_submission
    /* app.py
    Dashboard.php
    /* dashboard_latency.py
    /* edge_processing_latency.py
    /* latency_test.sql
    README.txt
Dashboard.php x edge_processing_latency.py app.py dashboard_latency.py
12 /
13 @app.route("/")
14 def dashboard():
15     return render_template("dashboard.html")
16
17 @app.route("/api/latest")
18 def latest_data():
19     cursor = db.cursor(dictionary=True)
20     cursor.execute("""
21         SELECT temperature, humidity, status, time
22         FROM sensor_data
23         ORDER BY time DESC
24         LIMIT 1
25     """)
26     return jsonify(cursor.fetchone())
27
28 @app.route("/api/history")
29 def history_data():
```

Gambar 5. Kode untuk Python

Berkas dashboard.html merupakan komponen frontend yang menampilkan hasil pemantauan sistem dalam bentuk antarmuka berbasis web. Halaman ini menyajikan informasi suhu dan kelembaban terkini, status node edge, grafik historis perubahan data, serta log pembacaan sensor dalam bentuk tabel. Seluruh data yang ditampilkan diambil langsung dari basis data melalui API yang disediakan oleh app.py, sehingga dashboard mampu merepresentasikan kinerja sistem pemantauan secara aktual tanpa menggunakan data dummy. Antarmuka ini dirancang untuk memudahkan pengguna dalam melakukan pemantauan kondisi lingkungan secara real-time.



```
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  ▼ iot_edge_intelligence_submission
    /* app.py
    Dashboard.php
    /* dashboard_latency.py
    /* edge_processing_latency.py
    /* latency_test.sql
    README.txt
Dashboard.php x edge_processing_latency.py app.py dashboard_latency.py
57
58 <script>
59 fetch('/api/latest')
60 .then(res => res.json())
61 .then(data => {
62     document.getElementById('temp').innerText = data.temperature + " °C";
63     document.getElementById('hum').innerText = data.humidity + " %";
64     document.getElementById('status').innerText = data.status;
65 });
66
67 fetch('/api/history')
68 .then(res => res.json())
69 .then(data => {
70     const labels = data.map(d => d.time);
71     const temp = data.map(d => d.temperature);
72     const hum = data.map(d => d.humidity);
73
74     new Chart(document.getElementById('tempChart'), {
75         type: 'line',
76         data: { labels, datasets: [{label: 'Temperature (°C)', data: temp, borderColor: 'red'}] }
77     });
78
79     new Chart(document.getElementById('humChart'), {
80         type: 'line',
81         data: { labels, datasets: [{label: 'Humidity (%)', data: hum, borderColor: 'blue'}] }
```

Gambar 6. System Coding Dashboard

4. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa penerapan arsitektur edge intelligence pada sistem pemantauan Internet of Things (IoT) lokal mampu meningkatkan kinerja sistem secara signifikan, khususnya dalam mereduksi latensi dan meningkatkan waktu respons. Pemrosesan data yang dilakukan secara langsung pada node edge memungkinkan sistem mengambil keputusan awal tanpa ketergantungan penuh terhadap server pusat, sehingga waktu tunda akibat transmisi data ke cloud dapat diminimalkan. Implementasi sistem yang memanfaatkan mikrokontroler sebagai node sensor dan Raspberry Pi sebagai node edge menunjukkan bahwa pemrosesan lokal dapat berjalan secara stabil dan konsisten, dengan data sensor diproses secara real-time serta disimpan

ke dalam basis data untuk keperluan pemantauan historis. Dashboard berbasis web yang dikembangkan mampu menampilkan data aktual, grafik pemantauan, dan log pembacaan sensor secara dinamis sehingga merepresentasikan kondisi sistem yang sebenarnya. Hasil pengujian latensi menunjukkan bahwa sistem berbasis edge intelligence memiliki latensi yang lebih rendah dan keandalan yang lebih baik dibandingkan sistem berbasis cloud computing penuh, terutama ketika terjadi gangguan koneksi ke server pusat selama jaringan lokal masih tersedia. Dengan demikian, edge intelligence terbukti sebagai solusi yang efektif untuk mendukung sistem pemantauan IoT yang membutuhkan respons cepat, efisien, dan stabil, serta membuka peluang pengembangan lanjutan melalui penerapan algoritma analisis yang lebih kompleks pada node edge dan pengujian pada skala sistem yang lebih besar untuk meningkatkan generalisasi hasil penelitian.

REFERENCES

- [1] Y. Mao, C. You, J. Zhang, K. Huang, dan K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Commun. Surv. & Tutorials*, vol. 19, no. 4, hal. 2322–2358, 2017, doi: 10.1109/COMST.2017.2745201.
- [2] M. Chiang dan T. Zhang, "Fog and IoT: An Overview of Research Opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, hal. 854–864, 2016, doi: 10.1109/JIOT.2016.2584538.
- [3] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, dan M. Chen, "In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication," *IEEE Netw.*, vol. 33, no. 5, hal. 156–165, 2019, doi: 10.1109/MNET.001.1800456.
- [4] W. Shi, J. Cao, Q. Zhang, Y. Li, dan L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet Things J.*, vol. 3, no. 5, hal. 637–646, 2016, doi: 10.1109/JIOT.2016.2579198.
- [5] N. Abbas, Y. Zhang, A. Taherkordi, dan T. Skeie, "Mobile Edge Computing: A Survey," *IEEE Internet Things J.*, vol. 5, no. 1, hal. 450–465, 2018, doi: 10.1109/JIOT.2017.2750180.
- [6] M. Satyanarayanan, "The Emergence of Edge Computing," *Computer (Long Beach, Calif.)*, vol. 50, no. 1, hal. 30–39, 2017, doi: 10.1109/MC.2017.9.
- [7] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, dan D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Architecture," *IEEE Commun. Surv. & Tutorials*, vol. 19, no. 3, hal. 1657–1681, 2017, doi: 10.1109/COMST.2017.2705720.
- [8] X. Xu, H. Liu, Y. Zhang, L. Qi, dan W. Dou, "Edge Intelligence: Empowering Intelligence to the Edge of Network," *IEEE Access*, vol. 7, hal. 128–144, 2019, doi: 10.1109/ACCESS.2019.2912473.
- [9] J. Gubbi, R. Buyya, S. Marusic, dan M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, hal. 1645–1660, 2013, doi: 10.1016/j.future.2013.01.010.
- [10] P. Mach dan Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Commun. Surv. & Tutorials*, vol. 19, no. 3, hal. 1628–1656, 2017, doi: 10.1109/COMST.2017.2682318.
- [11] A. Yousefpour dan others, "All One Needs to Know about Fog Computing and Related Edge Computing Paradigms," *J. Syst. Archit.*, vol. 98, hal. 289–330, 2019, doi: 10.1016/j.sysarc.2019.02.009.
- [12] L. N. Ifani, R. Harimurti, N. Kholis, dan A. B. Santosa, "Pengembangan Media Trainer Arduino Uno Berbasis Iot Sebagai Media Pembelajaran Pada Mata Pelajaran Sistem Pengendali Elektronik Di Smkn 1 Jetis," *J. Pendidik. Tek. Elektro*, vol. 10, no. 03, hal. 357–365, 2021, doi: 10.26740/jpte.v10n03.p357-365.
- [13] R. Roman, J. Lopez, dan M. Mambo, "Mobile Edge Computing, Fog et al.: A Survey and Analysis of Security Threats and Challenges," *Futur. Gener. Comput. Syst.*, vol. 78, hal. 680–698, 2018, doi: 10.1016/j.future.2016.11.009.
- [14] A. Botta, W. De Donato, V. Persico, dan A. Pescapé, "Integration of Cloud Computing and Internet of Things: A Survey," *Futur. Gener. Comput. Syst.*, vol. 56, hal. 684–700, 2016, doi: 10.1016/j.future.2015.09.021.
- [15] P. Bellavista, A. Zanni, dan L. Foschini, "Scalable and Context-Aware IoT Service Provisioning via Edge Computing," *IEEE Internet Things J.*, vol. 6, no. 2, hal. 2599–2612, 2019, doi: 10.1109/JIOT.2018.2879205.