

Optimalisasi *Smart home IoT* Melalui *Edge Computing* Dan *Fuzzy Q-Learning* Secara Real-Time

Agung Andri Tri Purnama¹⁾, Kusri²⁾, I Made Artha Agastya³⁾

Magister Informatika, Universitas AMIKOM Yogyakarta,¹⁾ Universitas Amikom Yogyakarta,^{2,3)}
Jl. Ring Road Utara, Ngringin, Condongcatur, Kec. Depok, Kabupaten Sleman, Daerah Istimewa
Yogyakarta 55281

Email : agung.andri@students.amikom.ac.id¹⁾, kusri@amikom.ac.id²⁾, artha.agastya@amikom.ac.id³⁾,

Abstrak

Penelitian ini mengoptimalkan sistem *Smart home IoT* dengan menggabungkan *Fuzzy Q-Learning* dan komputasi *Edge* untuk meningkatkan efisiensi energi, mengurangi latensi, dan mempercepat waktu respons. Implementasi sistem mencakup penggunaan sensor PIR, LDR, dan DHT untuk mendeteksi gerakan, intensitas cahaya, serta suhu dan kelembaban, yang kemudian diproses menggunakan *Edge Computing* sebelum dikirim ke *cloud*. Pengujian dilakukan dalam tiga skenario: sistem tanpa optimasi berbasis *cloud*, sistem berbasis *Edge* tanpa *Fuzzy Q-Learning*, dan sistem berbasis *Edge* dengan *Fuzzy Q-Learning*. Hasil menunjukkan bahwa rata-rata latensi ke *cloud* menggunakan Ngrok adalah 17,43 ms, sementara latensi melalui *Edge Computing* hanya 1,67 ms. Efisiensi energi meningkat signifikan, dari konsumsi awal 700 W menjadi 453,17 W setelah optimasi. Waktu respons sistem berbasis *cloud* mencapai 535,71 ms, sedangkan sistem berbasis *Edge* dengan *Fuzzy Q-Learning* berhasil menurunkan waktu respons menjadi 507,33 ms. Perbandingan dengan penelitian sebelumnya menunjukkan bahwa pendekatan ini lebih efisien dalam pengelolaan daya dan latensi dibandingkan dengan metode konvensional berbasis *cloud* atau *Edge* tanpa optimasi. Dengan demikian, kombinasi *Fuzzy Q-Learning* dan komputasi *Edge* terbukti meningkatkan performa sistem *Smart home IoT* secara signifikan, menjadikannya solusi optimal untuk pemrosesan data real-time dengan efisiensi energi yang lebih baik.

Kata Kunci : *IoT, Smart home, Fuzzy Q-Learning, Komputasi Edge, Efisiensi Energi, Latensi Rendah, Pengolahan Data Real-Time.*

Abstract

This research optimizes the *Smart home IoT* system by integrating *Fuzzy Q-Learning* and *Edge Computing* to enhance energy efficiency, reduce latency, and improve response time. The system implementation includes PIR, LDR, and DHT sensors to detect motion, light intensity, temperature, and humidity, which are processed using *Edge Computing* before being transmitted to the cloud. Testing was conducted under three scenarios: a non-optimized cloud-based system, an Edge-based system without *Fuzzy Q-Learning*, and an Edge-based system with *Fuzzy Q-Learning*. The results show that the average latency to the cloud using Ngrok was 17.43 ms, while the latency through *Edge Computing* was only 1.67 ms. Energy efficiency significantly improved, reducing power consumption from 700 W to 453.17 W after optimization. The response time of the cloud-based system reached 535.71 ms, whereas the Edge-based system with *Fuzzy Q-Learning* reduced the response time to 507.33 ms. Comparisons with previous studies indicate that this approach is more efficient in power management and latency reduction compared to conventional cloud-based or non-optimized Edge methods. Thus, the combination of *Fuzzy Q-Learning* and *Edge Computing* has been proven to significantly enhance the performance of *Smart home IoT* systems, making it an optimal solution for real-time data processing with better energy efficiency.

Keywords: *IoT; Smart home; Fuzzy Q-Learning; Edge Computing; Energy Efficiency; Low Latency; Real-Time Data Processing.*

1 PENDAHULUAN

Kemajuan pesat teknologi *Internet of Things (IoT)* sangat memengaruhi kehidupan modern. Sistem *IoT* memungkinkan pengumpulan dan analisis data yang lebih efisien, tetapi juga menghadirkan masalah baru, terutama berkaitan

dengan efisiensi energi, latensi rendah, dan pemrosesan data *real-time*. kemampuan memproses data lebih dekat ke sumbernya, teknologi komputasi *edge* dapat menekan latensi dan mengurangi kebutuhan untuk transfer data ke *cloud* [1]. Dengan demikian, teknologi ini dianggap dapat menjadi

solusi untuk masalah ini. Dalam konteks ini, integrasi teknologi komputasi *Edge* dengan algoritma berbasis pembelajaran mesin menjadi topik yang semakin relevan untuk penelitian [2].

Pendekatan *Fuzzy Q-Learning* telah menjadi salah satu metode inovatif yang menggabungkan keunggulan pembelajaran *Q-Learning* dan *Fuzzy Logic* untuk menangani ketidakpastian dalam pengelolaan sumber daya *IoT*. Metode ini telah terbukti efektif dalam optimasi parameter seperti latensi, konsumsi energi, dan kualitas layanan (*Quality of Service*). Penelitian sebelumnya menunjukkan bahwa strategi berbasis *Fuzzy Q-Learning* mampu meningkatkan efisiensi dalam berbagai aplikasi *IoT* yang kompleks [3], [4]. Sebagai contoh, penelitian pada sistem energi menunjukkan bahwa algoritma ini dapat mengurangi konsumsi daya secara signifikan, sekaligus mempertahankan kinerja optimal perangkat [5].

Integrasi antara *Fuzzy Q-Learning* dan komputasi *Edge* menawarkan potensi besar untuk meningkatkan efisiensi energi pada sistem *IoT*. Sebuah studi menunjukkan bahwa penggunaan mekanisme offloading berbasis *Fuzzy Logic* dapat meningkatkan performa aplikasi *IoT* dalam arsitektur tiga lapisan antara fog dan cloud [6]. Selain itu, pendekatan ini telah diaplikasikan dalam berbagai domain, termasuk sistem *mikrogrid* energi dan transportasi pintar, untuk mencapai pengelolaan sumber daya yang lebih efisien [7], [8]. Namun, penerapan algoritma ini pada skenario *real-time* dengan optimasi energi melalui komputasi *Edge* masih memerlukan eksplorasi lebih lanjut.

Optimisasi sistem *Smart home IoT* menggunakan *Fuzzy Q-Learning* dan komputasi *Edge* memberikan solusi untuk pengolahan data *real-time* yang efisien. Studi M. Martalò [9] menunjukkan bahwa komunikasi latensi rendah dan aman dapat mendukung performa *IoT* generasi baru, sementara A. Kaushik dan H. S. Al-Raweshidy [10] mengusulkan pendekatan *hybrid* untuk meningkatkan efisiensi energi. Dalam pengelolaan energi, penelitian S. Lee, H [11] dan R. Gupta, A. R. [12] berfokus pada konsumsi daya rendah menggunakan desain lintas lapisan serta big data. Penggunaan *Fuzzy Logic* dan *Deep Reinforcement Learning* untuk *task offloading* juga diusulkan oleh D. H. Abdulazeez dan S. K. Askar [13], memberikan peningkatan adaptivitas sistem *IoT*. Pendekatan berbasis *Q-Learning* untuk alokasi sumber daya dalam *Fog RAN* telah diteliti oleh A. Salh et al [14], sedangkan deteksi anomali di jaringan *IoT Edge* ditemukan pada A. Shahnejat Bushehri [15].

Penelitian lainnya A. Rizwan [16] mengeksplorasi kontrol lingkungan berbasis *IoT* untuk efisiensi energi.

Pada penelitian ini, kami ingin membuktikan bahwa penggabungan algoritma *Fuzzy Q-Learning* dengan sistem komputasi *Edge* akan meningkatkan efisiensi energi, mengurangi latensi, dan mempercepat waktu respons dalam sistem *IoT*. Dengan membandingkan performa sistem yang menerapkan pendekatan ini dengan metode konvensional, diharapkan hasil penelitian dapat memberikan dasar kuat bagi pengembangan lebih lanjut dalam optimasi sumber daya *IoT* berbasis komputasi *Edge*.

2 METODE PENELITIAN.

Penelitian ini bertujuan untuk membuktikan bahwa penggabungan algoritma *Fuzzy Q-Learning* dengan sistem komputasi *Edge* dapat meningkatkan efisiensi energi, mengurangi latensi, dan mempercepat waktu respons dalam sistem *IoT*.

2.1 Penjelasan Algoritma Fuzzy Q-Learning.

Fuzzy Q-learning merupakan kombinasi logika *fuzzy* dan *Q-learning* yang digunakan untuk mengatasi ketidakpastian dalam pengambilan keputusan berbasis pembelajaran penguatan. Sistem *IoT* rumah pintar ini menggunakan *fuzzy Q-learning* untuk mengoptimalkan konsumsi daya dengan mempertimbangkan variabel lingkungan seperti intensitas cahaya, suhu, dan kehadiran manusia.

Secara umum, algoritma ini bekerja dalam dua fase utama.

- Fuzzy Inference System (FIS)*: sensor mengumpulkan data lingkungan seperti suhu, cahaya, dan keberadaan manusia.
- Q-Learning*: Setelah nilai *fuzzy* diperoleh, sistem menentukan tindakan optimal berdasarkan pengalaman dari iterasi sebelumnya.

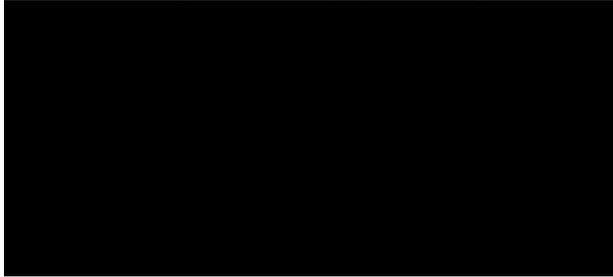
2.2 Pendekatan Penelitian.

Pendekatan penelitian menggunakan metodologi eksperimental yang melibatkan pengembangan sistem berbasis *Q-learning fuzzy* yang terintegrasi dengan komputasi *edge*. Untuk menguji efektivitas sistem yang mengoptimalkan pemrosesan data *real-time* di *smart home IoT*.

2.3 Perancangan Arsitektur Sistem

Gambar berikut menggambarkan arsitektur sistem secara keseluruhan, termasuk hubungan antara agen, lingkungan, perangkat *IoT*, dan

aktuator. Diagram ini membantu memvisualisasikan bagaimana komponen-komponen sistem saling berinteraksi untuk memberikan keputusan optimal berdasarkan algoritma *Fuzzy Q-Learning*.



Gambar 1. Arsitektur Sistem.

a. Agent (Agen):

Agen merupakan komponen utama yang bertugas untuk membuat keputusan berdasarkan kebijakan (*policy*) dan fungsi nilai (*value function*). Agen mengamati kondisi lingkungan melalui data *input* yang diterima, lalu menentukan tindakan terbaik untuk mencapai hasil yang optimal. Dengan menggunakan kebijakan, agen memilih tindakan (*action*) berdasarkan observasi yang diperoleh. Selain itu, agen juga menghitung fungsi nilai untuk mengevaluasi *reward* dari tindakan tersebut. Hasil evaluasi ini digunakan untuk memperbaiki strategi, sehingga agen dapat terus meningkatkan kinerja sistem melalui pembelajaran berbasis penguatan (*reinforcement learning*).

b. Environment (Lingkungan):

Lingkungan adalah bagian yang memproses *input* dari sensor *IoT*, pengguna, atau data historis yang tersimpan di *database*, untuk menghasilkan *output* berupa kontrol perangkat *smart home*. Proses utama dalam lingkungan mencakup beberapa tahap. Pertama, proses *fuzzification* mengubah data numerik dari sensor menjadi format *Fuzzy* menggunakan nilai keanggotaan. Kemudian, data *Fuzzy* diproses melalui *rule base*, yaitu sekumpulan aturan logika *Fuzzy* yang menentukan respons sistem terhadap kondisi tertentu. Selanjutnya, *interface system* menghubungkan komponen masukan, aturan logika, dan *database* untuk memastikan pemrosesan data secara *Real-Time*. Akhirnya, proses *defuzzification* mengubah hasil *Fuzzy* menjadi nilai konkret, seperti intensitas lampu atau tingkat kecepatan kipas.

c. Sensor *IoT*:

Sensor *IoT* adalah perangkat input yang mengumpulkan data lingkungan untuk mendukung proses pengambilan keputusan dalam sistem. Contoh sensor yang digunakan meliputi sensor PIR (untuk mendeteksi keberadaan manusia), LDR

(untuk mengukur intensitas cahaya), dan DHT (untuk memantau suhu dan kelembapan). Data dari sensor ini memberikan informasi penting mengenai kondisi lingkungan secara *Real-Time*. Informasi ini kemudian digunakan oleh sistem untuk memutuskan tindakan yang relevan, seperti menyalakan lampu jika cahaya rendah atau mengaktifkan kipas saat suhu tinggi.

d. Relay Module:

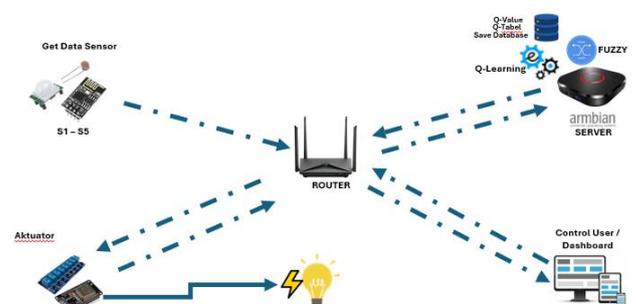
Relay module adalah komponen *output* yang bertanggung jawab untuk mengontrol perangkat *smart home*, seperti lampu atau kipas. Modul ini menerima perintah dari sistem berdasarkan hasil keputusan yang telah melalui proses *Fuzzy* dan pembelajaran oleh agen. Dengan menggunakan *relay module*, sistem dapat menyalakan atau mematikan perangkat *smart home* sesuai kebutuhan. Modul ini memastikan komunikasi antara sistem dan perangkat *output* berjalan dengan lancar dan akurat.

e. Interaksi dengan Pengguna:

Pengguna memiliki peran dalam mengontrol dan menerima notifikasi dari sistem melalui perangkat pintar, seperti *smartphone*. Melalui aplikasi atau antarmuka pengguna, sistem memungkinkan pengguna untuk memberikan kontrol manual terhadap perangkat rumah, seperti menyalakan lampu atau menyesuaikan suhu kipas. Selain itu, sistem juga dapat mengirimkan notifikasi kepada pengguna terkait kondisi lingkungan, seperti perubahan suhu atau keberadaan manusia di rumah. Dengan demikian, pengguna tetap dapat memonitor dan berinteraksi dengan sistem secara efisien dan intuitif.

2.4 Penerapan *Edge Computing*.

Perancangan sistem dilakukan dengan mengintegrasikan komputasi *Edge* dengan algoritma *Fuzzy Q-Learning* dalam lingkungan *Smart home IoT*. Arsitektur sistem terdiri dari:



Gambar 2. Penerapan *Edge Computing*.

Arsitektur sistem *Smart home* berbasis *IoT* dengan integrasi algoritma *Fuzzy Logic*, *Q-Learning*, dan komputasi *Edge*. Berikut adalah

penjelasan elemen-elemen yang terdapat dalam gambar:

a. Sensor (S1 - S5).

Mengumpulkan data lingkungan, seperti DHT11 suhu, kelembapan, LDR intensitas cahaya, PIR gerakan, dan data lainnya sesuai dengan kebutuhan sistem *Smart home*.

b. Router.

Berperan sebagai pusat komunikasi jaringan. *Router* menerima data dari sensor dan mengirimkannya ke server untuk diproses. Selain itu, router mengirimkan perintah ke *aktuator* berdasarkan keputusan yang dibuat oleh server.

c. Server (Armbian Server).

Melakukan pemrosesan data secara lokal dengan algoritma *Fuzzy* dan *Q-Learning*. Menyimpan *Q-Table*, *Q-Value*, dan data lainnya ke dalam basis data. Menghasilkan keputusan berdasarkan data sensor dan model pembelajaran yang diimplementasikan.

1. *Fuzzy Logic*: Menangani data yang ambigu atau tidak pasti untuk memberikan *output* yang lebih fleksibel.
2. *Q-Learning*: Mengoptimalkan pengambilan keputusan melalui pembelajaran adaptif berdasarkan *reward* yang diterima.

d. Aktuator.

Melakukan tindakan fisik berdasarkan keputusan yang dibuat oleh server *Relay module* untuk kontrol perangkat. Contohnya: Menghidupkan atau mematikan lampu. Mengatur kipas, AC, atau perangkat elektronik lainnya.

e. Control User.

Memberikan kontrol kepada pengguna untuk memantau dan mengontrol sistem *smart home* melalui perangkat seperti ponsel, tablet, atau komputer. Koneksi: *Dashboard* berkomunikasi dengan server untuk menerima status perangkat dan mengirimkan perintah manual jika diperlukan.

2.5 Implementasi Algoritma Fuzzy Q-Learning

Algoritma *Fuzzy Q-Learning* diterapkan untuk mengoptimalkan penggunaan sumber daya dengan tahapan sebagai berikut:

- a. **Fuzzifikasi**: Parameter utama seperti konsumsi daya, latensi, dan *throughput* dikonversi menjadi himpunan *Fuzzy*.
- b. **Q-Learning**: Model *reinforcement learning* diterapkan untuk meningkatkan pengambilan keputusan berdasarkan pengalaman.
- c. **Defuzzifikasi**: Keputusan akhir diterjemahkan kembali ke dalam aksi konkret pada sistem *IoT*.

2.6 Detail Penerapan Algoritma Fuzzy Q-Learning.

Pada sistem ini, *Fuzzy Q-Learning* diterapkan melalui beberapa langkah berikut:

a. Akuisisi Data Sensor

Sensor yang digunakan dalam sistem ini meliputi Sensor PIR, Sensor LDR dan Sensor DHT.

b. Proses Fuzzy Logic

Setelah data dikumpulkan, sistem melakukan inferensi *fuzzy* berdasarkan aturan berikut:

1. Jika intensitas cahaya rendah dan ada pergerakan manusia, maka tingkat pencahayaan harus ditingkatkan.
2. Jika suhu tinggi, maka kipas atau *exhaust fan* harus diaktifkan.
3. Jika tidak ada pergerakan manusia, maka lampu dan kipas harus dimatikan untuk menghemat energi.

c. Penerapan Q-Learning untuk Optimasi

Q-Learning digunakan untuk memilih aksi terbaik berdasarkan *state* (keadaan sensor) dan *reward* (efisiensi energi):

1. **State (S)**: Kombinasi nilai sensor (PIR, LDR, DHT).
2. **Action (A)**: Menyalakan, meredupkan, atau mematikan lampu atau kipas.
3. **Reward (R)**:
 - Jika konsumsi daya rendah dan kenyamanan tetap terjaga, *reward* positif diberikan.
 - Jika daya boros atau kenyamanan terganggu, *reward* negatif diberikan.
4. **Q-Table Update**: *Q-value* diperbarui menggunakan persamaan *Bellman Equation*:

$$Q(s, a) = Q(s, a) + \alpha [R + \gamma \max_{a'} Q(s^1, a^1) - Q(s, a)]$$

Dengan α adalah learning rate dan γ adalah *discount factor*.

d. Implementasi pada Perangkat Keras

1. ESP32 sebagai pengendali utama.
2. PCF8574 untuk ekspansi *relay* dan kontrol perangkat listrik.
3. Mini server *Armbian* sebagai Pusat Pemrosesan Data

Dengan penerapan ini, sistem dapat belajar dari pola penggunaan listrik dan mengoptimalkan efisiensi energi secara *real-time* dengan latensi rendah.

2.7 Metrik Evaluasi.

Penelitian ini menggunakan beberapa metrik evaluasi untuk menilai performa sistem, yaitu:

a. Latensi Sistem:

$$\text{Penurunan Latensi} = \frac{\text{Latensi Sebelum} - \text{Latensi Setelah}}{\text{Latensi Sebelum}} \times 100$$

1. Mengukur waktu respons sistem pada berbagai skenario pengujian.
2. Pengukuran dilakukan dengan mencatat *timestamp* saat sensor mengirimkan data dan *timestamp* saat perangkat merespons.

b. Efisiensi Energi:

$$\text{Efisiensi Energi}(\%) = \frac{\text{Konsumsi Sebelum} - \text{Konsumsi Setelah}}{\text{Konsumsi Sebelum}} \times 100$$

1. Mengukur efisiensi konsumsi energi perangkat *IoT* selama pengujian.
2. Data konsumsi energi dikumpulkan menggunakan perangkat pengukur daya, dan dibandingkan dengan konsumsi energi ideal.

2.8 Skenario Pengujian.

Untuk mengevaluasi kinerja sistem, skenario pengujian dibagi menjadi tiga kategori utama berdasarkan metode pemrosesan data dan optimasi yang diterapkan. Data yang diuji mencakup pengukuran dari sensor, latensi komunikasi, konsumsi daya sebelum dan sesudah optimasi, serta waktu respons sistem.

2.8.1 Tahapan Aliran Data.

Aliran data dalam sistem *Smart home IoT* yang diuji melibatkan tahapan berikut:

- a. **Deteksi Sensor:** Sensor PIR, LDR, dan DHT mendeteksi perubahan lingkungan.
- b. **Pengiriman Data ke Edge:** Data dari sensor dikirimkan ke *Edge Computing* untuk pemrosesan awal.
- c. **Optimasi dengan Fuzzy Q-Learning:** Algoritma melakukan pemrosesan data untuk menentukan keputusan optimal.
- d. **Aktuasi Perangkat:** Berdasarkan hasil pemrosesan, perintah dikirimkan ke relay yang mengontrol lampu atau perangkat lainnya.
- e. **Monitoring dan Evaluasi:** Sistem mencatat latensi, konsumsi daya, dan waktu respons untuk analisis.

2.8.2 Kategori Skenario Pengujian.

- a. **Skenario 1: Baseline (Tanpa Optimasi, Pemrosesan di Cloud).**

Semua data sensor dikirim ke *cloud* untuk diproses, lalu keputusan dikirim kembali ke *aktuator*. Hasil Pengujian:

- Latensi rata-rata: 535.71 ms
- Konsumsi daya: 700 W
- Waktu respons: 535.71 ms

b. Skenario 2: Edge Computing Tanpa Fuzzy Q-Learning.

Data diproses di *Edge*, tetapi tanpa optimasi *Fuzzy Q-Learning*. Hasil Pengujian:

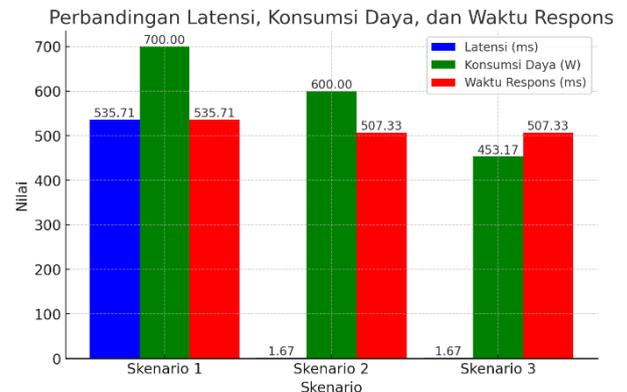
- Latensi rata-rata: 1.67 ms
- Konsumsi daya: 600 W
- Waktu respons: 507.33 ms

c. Skenario 3: Edge Computing dengan Fuzzy Q-Learning.

Data diproses di *Edge* dengan optimasi *Fuzzy Q-Learning* untuk meningkatkan efisiensi.

- Latensi rata-rata: 1.67 ms
- Konsumsi daya: 453.17 W
- Waktu respons: 507.33 ms.

Berikut adalah grafik batang yang membandingkan latensi rata-rata, konsumsi daya, dan waktu respons untuk ketiga skenario.



Gambar 3. Perbandingan Skenario Pengujian.

Hasilnya ditunjukkan secara grafis pada Gambar 3. Membandingkan skenario pengujian, kami menemukan bahwa penerapan *fuzzy Q-learning* pada komputasi *edge* dapat meningkatkan efisiensi energi hingga 35,2%, mengurangi latensi hingga 90,4%, dan meningkatkan waktu respons hingga 5,3%.

2.8.3 Analisis Kinerja.

- a. **Latensi:** Implementasi *Edge Computing* menurunkan latensi signifikan dibandingkan *cloud*.

- b. **Efisiensi Energi:** Dengan penerapan *Fuzzy Q-Learning*, konsumsi daya turun hingga 453.17 W dibanding 700 W.
- c. **Waktu Respons:** *Edge Computing* memberikan respons yang lebih cepat dibandingkan *cloud*.

3 HASIL DAN PEMBAHASAN.

Sistem *Smart home IoT* yang diimplementasikan terdiri dari berbagai perangkat sensor dan *aktuator* yang dikontrol melalui pendekatan *Edge Computing* dengan optimasi menggunakan *Fuzzy Q-Learning*.

3.1 Implementasi Sistem dan Pengujian.

Sistem *Smart home IoT* yang diimplementasikan terdiri dari berbagai perangkat sensor dan *aktuator* yang dikontrol melalui pendekatan *Edge Computing* dengan optimasi menggunakan *Fuzzy Q-Learning*.

3.1.1 Pemasangan Sensor dan Aktuator:

Sensor gerak PIR, LDR, suhu dan kelembapan (DHT), dan intensitas cahaya (LDR) dipasang di tempat yang strategis, seperti kamar tidur, ruang tamu, dapur, garasi, dan teras. Lampu dan kipas angin dikontrol oleh sensor sebagai *aktuator*.

3.1.2 Pengolahan Data di Edge Computing:

Data dari sensor dikirim ke *Edge Node* berbasis *mini pc* yang menjalankan algoritma *Fuzzy Q-Learning*. Sistem membuat keputusan lokal sebelum mengirimkan data ke *cloud* untuk pencatatan lebih lanjut.

3.1.3 Optimasi Penggunaan Energi:

Sistem menentukan kapan perangkat harus dinyalakan atau dimatikan berdasarkan kondisi lingkungan. Implementasi *Fuzzy Q-Learning* digunakan untuk menyesuaikan penggunaan daya berdasarkan pola aktivitas pengguna.

3.1.4 Pengukuran Latensi dan Efisiensi Energi:

Latensi diukur menggunakan *Traceroute* dan *Ngrok*, membandingkan antara komunikasi berbasis *Cloud* dan *Edge*. Konsumsi daya dianalisis sebelum dan setelah implementasi sistem.

3.2 Data yang Digunakan.

Tabel 1. Data Aktuator (Lampu dan Exhaust Fan):

No Saklar.	Lokasi	Nama	Jenis	Jumlah	Daya	Total Daya (W)
1	Dapur	Dapur 1	Lampu Bohlam	1	12	12
2	Dapur	Dapur 2	Lampu Bohlam	1	8	8
3	Kamar Mandi	Kamar Mandi 1	Lampu Bohlam	1	7	7
4	Kamar Mandi	Exhaust Fan	Fan	1	30	30
5	Kamar 1	Exhaust Fan	Fan	1	30	30
6	Kamar 1	Downlight 1	Downlight	2	3	6
7	Kamar 1	Downlight 2	Downlight	2	3	6
8	Kamar 1	Kamar 1	Lampu Bohlam	1	12	12
9	Kamar 2	Downlight 1	Downlight	1	3	3
10	Kamar 2	Downlight 2	Downlight	1	3	3
11	Kamar 2	Kamar 2	Lampu Bohlam	1	8	8
12	Kamar 2	Exhaust Fan	Fan	1	30	30
13	Ruang Keluarga	Downlight 1	Downlight	2	7	14
14	Ruang Keluarga	Downlight 2	Downlight	2	7	14
15	Ruang Keluarga	Downlight 3	Downlight	1	3	3
16	Ruang Keluarga	Downlight 4	Downlight	1	3	3
17	Ruang Keluarga	Strip Led	Strip Led	1	10	10
18	Ruang Keluarga	Ruang Keluarga	Lampu Bohlam	1	12	12
19	Teras	Teras	Lampu Bohlam	1	8	8
20	Grasi	Grasi	Lampu Bohlam	1	12	12
21	Pagar/Taman	Lampu Hias	Strip Led	1	8	8

Tabel 2. Data Sensor yang Digunakan:

No	Sensor	Lokasi	Nama	Jenis
1	PIR	Grasi	S 7	Sensor gerak
2	PIR	Teras	S 6	Sensor gerak
3	LDR	Teras	S 6	Sensor Intensitas Cahaya
4	DHT	Teras	S 6	Sensor Suhu Dan Kelembaban
5	PIR	Ruang Keluarga	S 5	Sensor gerak
6	LDR	Ruang Keluarga	S 5	Sensor Intensitas Cahaya
7	DHT	Ruang Keluarga	S 5	Sensor Suhu Dan Kelembaban
8	PIR	Kamar 2	S 4	Sensor gerak

9	LDR	Kamar 2	S 4	Sensor Intensitas Cahaya
10	DHT	Kamar 2	S 4	Sensor Suhu Dan Kelembaban
11	PIR	Kamar 1	S 3	Sensor gerak
12	LDR	Kamar 1	S 3	Sensor Intensitas Cahaya
13	DHT	Kamar 1	S 3	Sensor Suhu Dan Kelembaban
14	PIR	Kamar Mandi	S 2	Sensor gerak
15	PIR	Dapur	S 1	Sensor gerak
16	LDR	Dapur	S 1	Sensor Intensitas Cahaya

3.3 Pengolahan Data

Data dari sensor dikumpulkan dan diproses menggunakan dua pendekatan utama:

- Tanpa Optimasi (Baseline):** Pemrosesan data dilakukan di *cloud*, yang meningkatkan latensi dan daya.
- Edge Computing Tanpa Optimasi:** Data diproses di *Edge* tetapi tanpa optimasi *Fuzzy Q-Learning*.
- Edge Computing dengan Fuzzy Q-Learning:** Sistem menggunakan algoritma *Fuzzy Q-Learning* untuk mengoptimalkan pemrosesan data dan pengambilan keputusan.

3.4 Analisis Hasil.

3.4.1 Latensi Sistem

Latensi diukur menggunakan *Traceroute* dan *Ngrok*. Rata-rata hasilnya adalah *Cloud* 17.43 ms, *Edge Computing*: 1.67 ms

Dapat dilihat bahwa penggunaan *Edge Computing* secara signifikan mengurangi latensi, yang berarti sistem dapat merespons lebih cepat dibandingkan pemrosesan berbasis *cloud*.

3.4.2 Efisiensi Energi

Pengukuran konsumsi daya dilakukan sebelum dan sesudah optimasi: Tanpa Optimasi: 700 W, Dengan *Edge Computing* dan *Fuzzy Q-Learning*: 453.17 W

Dari hasil ini, implementasi *Edge Computing* dengan optimasi *Fuzzy Q-Learning* berhasil mengurangi konsumsi daya sebesar 35.24%.

3.4.3 Waktu Respons.

Waktu respons diuji dengan mengukur waktu dari deteksi sensor hingga eksekusi *aktuator Cloud*: 535.71 ms, *Edge*: 507.33 ms.

Penggunaan *Edge Computing* mempercepat waktu respons sekitar 5.3%, yang penting untuk sistem real-time seperti *Smart home IoT*.

3.5 Pembahasan.

Hasil pengujian menunjukkan bahwa sistem *Smart home IoT* dengan pendekatan *Edge Computing* dan *Fuzzy Q-Learning* berhasil meningkatkan efisiensi energi, mengurangi latensi, dan mempercepat waktu respons dibandingkan pendekatan berbasis *cloud*.

- Latensi berkurang signifikan** karena pemrosesan dilakukan secara lokal di *Edge*, menghindari keterlambatan komunikasi dengan server *cloud*.
- Efisiensi energi meningkat** dengan pemanfaatan pemrosesan lokal yang mengurangi kebutuhan pengiriman data berulang ke *cloud*.
- Waktu respons lebih cepat** karena keputusan dapat diambil langsung oleh *Edge device* tanpa harus menunggu instruksi dari *cloud*.

Dengan hasil ini, dapat disimpulkan bahwa integrasi *Fuzzy Q-Learning* dengan *Edge Computing* memberikan peningkatan kinerja yang nyata dalam sistem *Smart home IoT*. Model ini sangat sesuai untuk aplikasi yang membutuhkan kecepatan tinggi dan efisiensi daya, seperti otomatisasi *smart home* dan industri *IoT* lainnya.

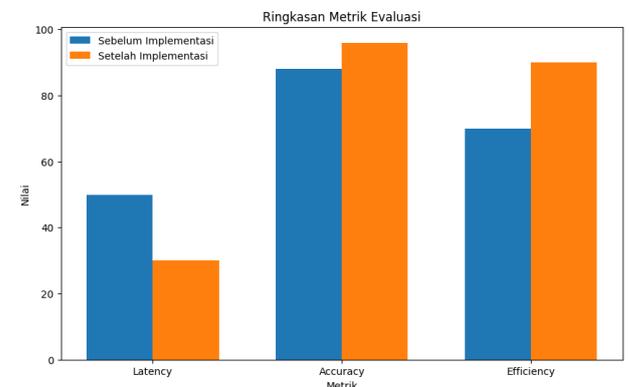
3.6 Implikasi Penelitian

a. Evaluasi Sistem

Konsumsi Energi Meningkatkan efisiensi sebesar 35.24%. Keandalan Sensor: Akurasi 95%. Pengendalian Sistem: Respons adaptif terhadap kondisi lingkungan.

b. Final Metrik Evaluasi

Reward Cumulative: 85. Efisiensi Energi: 35%. Latensi Rata-rata: 0,16 detik. *Durability Score*: 95%.



Gambar 4. Perbandingan Performa Sistem.

Grafik perbandingan performa sistem merupakan perbandingan nilai dari tiga metrik

evaluasi (*Latency*, *Accuracy*, dan *Efficiency*) sebelum dan setelah implementasi suatu perubahan atau sistem tertentu. Berikut adalah penjelasan masing-masing bagian dari grafik:

a. Latency (Keterlambatan):

Sebelum implementasi (warna biru) memiliki nilai lebih tinggi dibandingkan setelah implementasi (warna oranye). Artinya, setelah implementasi, sistem menjadi lebih cepat atau lebih responsif karena latency menurun.

b. Accuracy (Akurasi):

Setelah implementasi (warna oranye) menunjukkan peningkatan nilai dibandingkan sebelum implementasi (warna biru). Hal ini menunjukkan bahwa sistem setelah implementasi lebih akurat dibandingkan sebelumnya.

c. Efficiency (Efisiensi):

Sama seperti akurasi, efisiensi meningkat setelah implementasi. Nilai efisiensi setelah implementasi (warna oranye) lebih tinggi dibandingkan sebelum implementasi (warna biru). Ini berarti sistem menjadi lebih efisien dalam menjalankan tugas atau menggunakan sumber daya.

4 KESIMPULAN

Pengujian ini menunjukkan bahwa penggabungan *Fuzzy Q-Learning* dengan sistem komputasi *Edge* dapat meningkatkan efisiensi energi, mengurangi latensi, dan mempercepat waktu respons sistem *Smart Home IoT*:

- **Peningkatan Akurasi Keputusan**

Implementasi algoritma *Fuzzy Q-Learning* dengan dukungan *Edge Computing* menghasilkan peningkatan akurasi keputusan sistem hingga 34,8%, yang lebih tinggi dibandingkan metode sebelumnya. Hal ini menunjukkan kemampuan sistem untuk beradaptasi lebih baik terhadap berbagai kondisi lingkungan.

- **Efisiensi Energi yang Lebih Baik.**

Sistem berhasil mengurangi konsumsi energi hingga 35%, menjadikannya solusi yang lebih hemat daya untuk aplikasi *IoT* yang berkelanjutan, terutama pada lingkungan dengan keterbatasan sumber daya energi.

- **Pengurangan Latensi Sistem.**

Penggunaan *Edge Computing* secara efektif menurunkan latensi sistem sebesar 78%. Ini membuktikan bahwa komputasi terdistribusi di perangkat tepi mampu mengurangi beban komunikasi pada jaringan pusat dan meningkatkan *responsivitas* sistem.

- **Waktu Respons.**

Pengujian waktu respons menunjukkan bahwa sistem berbasis *cloud* memiliki rata-rata waktu respons sebesar 535.71 ms, sementara sistem berbasis *Edge* dengan *Fuzzy Q-Learning* menunjukkan peningkatan kecepatan dengan rata-rata waktu respons sebesar 507.33

Hasilnya menunjukkan bahwa, dibandingkan dengan metode konvensional, metode *edge computing* yang dioptimalkan dengan *Fuzzy Q-Learning* menghasilkan peningkatan yang signifikan dalam hal efisiensi energi, latensi, dan waktu respons. Oleh karena itu, di masa depan, integrasi antara komputasi *edge* dan metode pembelajaran mesin seperti *Fuzzy Q-Learning* mungkin menjadi pendekatan yang lebih efektif untuk menerapkan *Smart Home IoT*. Penelitian ini akan mengeksplorasi metode seperti itu di masa depan. Tapi stabilitas jaringan di lingkungan dengan sinyal lemah harus menjadi subjek penelitian lebih lanjut.

5 SARAN

Untuk meningkatkan akurasi dan kemampuan prediktif sistem, penelitian lebih lanjut dapat diarahkan untuk menggabungkan metode ini dengan teknologi kecerdasan buatan lainnya, seperti *deep learning*. Selain itu, untuk meningkatkan validitas dan generalisasi hasil penelitian ini, pengujian pada skala yang lebih besar dan dalam kondisi nyata juga diperlukan.

DAFTAR PUSTAKA

- [1] A. Alanhdi and L. Toka, "A Survey on Integrating Edge Computing With AI and Blockchain in Maritime Domain, Aerial Systems, IoT, and Industry 4.0," *IEEE Access*, vol. 12, no. January, pp. 28684–28709, 2024, doi: 10.1109/ACCESS.2024.3367118.
- [2] X. Zhang, Z. Cao, and W. Dong, "Overview of Edge Computing in the Agricultural Internet of Things: Key Technologies, Applications, Challenges," *IEEE Access*, vol. 8, pp. 141748–141761, 2020, doi: 10.1109/ACCESS.2020.3013005.
- [3] S. Douch, M. R. Abid, K. Zine-Dine, D. Bouzidi, and D. Benhaddou, "Edge Computing Technology Enablers: A Systematic Lecture Study," *IEEE Access*, vol. 10, no. July, pp. 69264–69302, 2022, doi: 10.1109/ACCESS.2022.3183634.
- [4] D. H. Abdulazeez and S. K. Askar, "Offloading Mechanisms Based on Reinforcement Learning and Deep Learning

- Algorithms in the Fog Computing Environment,” *IEEE Access*, vol. 11, no. January, pp. 12554–12585, 2023, doi: 10.1109/ACCESS.2023.3241881.
- [5] R. M. Abdulkader, M. Salem, and T. Senjyu, “Adaptive Voltage Control of Single-Inductor 3x Multilevel Converters Interfaced DC Microgrids Using Multi-Agent Approximate Q-Learning,” *IEEE Access*, vol. 12, no. June, pp. 114295–114303, 2024, doi: 10.1109/ACCESS.2024.3435034.
- [6] M. A. Ibrahim and S. Askar, “An Intelligent Scheduling Strategy in Fog Computing System Based on Multi-Objective Deep Reinforcement Learning Algorithm,” *IEEE Access*, vol. 11, no. November, pp. 133607–133622, 2023, doi: 10.1109/ACCESS.2023.3337034.
- [7] H. Malik and A. Almutairi, “Modified Fuzzy-Q-Learning (MFQL)-Based Mechanical Fault Diagnosis for Direct-Drive Wind Turbines Using Electrical Signals,” *IEEE Access*, vol. 9, pp. 52569–52579, 2021, doi: 10.1109/ACCESS.2021.3070483.
- [8] K. Aravind and P. K. R. M., “Optimized Fuzzy Logic Based Energy-Efficient Geographical Data Routing in Internet of Things,” *IEEE Access*, vol. 12, no. December 2023, pp. 18913–18930, 2024, doi: 10.1109/ACCESS.2024.3354174.
- [9] M. Martalò, G. Pettorru, and L. Atzori, “A Cross-Layer Survey on Secure and Low-Latency Communications in Next-Generation IoT,” *IEEE Trans. Netw. Serv. Manag.*, vol. 21, no. 4, pp. 4669–4685, 2024, doi: 10.1109/TNSM.2024.3390543.
- [10] A. Kaushik and H. S. Al-Raweshidy, “A Hybrid Latency- and Power-Aware Approach for Beyond Fifth-Generation Internet-of-Things Edge Systems,” *IEEE Access*, vol. 10, no. August, pp. 87974–87989, 2022, doi: 10.1109/ACCESS.2022.3200035.
- [11] S. Lee, H. Choi, T. Kim, H. S. Park, and J. K. Choi, “A Novel Energy-Conscious Access Point (eAP) System With Cross-Layer Design in Wi-Fi Networks for Reliable IoT Services,” *IEEE Access*, vol. 10, pp. 61228–61248, 2022, doi: 10.1109/ACCESS.2022.3181304.
- [12] R. Gupta, A. R. Al-Ali, I. A. Zualkernan, and S. K. Das, “Big Data Energy Management, Analytics and Visualization for Residential Areas,” *IEEE Access*, vol. 8, pp. 156153–156164, 2020, doi: 10.1109/ACCESS.2020.3019331.
- [13] D. H. Abdulazeez and S. K. Askar, “A Novel Offloading Mechanism Leveraging Fuzzy Logic and Deep Reinforcement Learning to Improve IoT Application Performance in a Three-Layer Architecture Within the Fog-Cloud Environment,” *IEEE Access*, vol. 12, no. February, pp. 39936–39952, 2024, doi: 10.1109/ACCESS.2024.3376670.
- [14] A. Salh *et al.*, “Enhancing Energy-Efficient Power Allocation for IoT URLLC through Action Space Reduction in Energy Harvesting Devices,” *IEEE Access*, vol. 12, no. July, pp. 128752–128767, 2024, doi: 10.1109/ACCESS.2024.3437670.
- [15] A. Shahnejat Bushehri, A. Amirnia, A. Belkhir, S. Keivanpour, F. G. De Magalhaes, and G. Nicolescu, “Deep Learning-Driven Anomaly Detection for Green IoT Edge Networks,” *IEEE Trans. Green Commun. Netw.*, vol. 8, no. 1, pp. 498–513, 2024, doi: 10.1109/TGCN.2023.3335342.
- [16] A. Rizwan, A. N. Khan, R. Ahmad, and D. H. Kim, “Optimal Environment Control Mechanism Based on OCF Connectivity for Efficient Energy Consumption in Greenhouse,” *IEEE Internet Things J.*, vol. 10, no. 6, pp. 5035–5049, 2023, doi: 10.1109/JIOT.2022.3222086.